CDC 2R051

Maintenance Management Analysis Journeyman

Volume 2. Maintenance Information Systems

Air Force Career Development Academy

Air University

Air Education and Training Command

2R051 02 1704, Edit Code 07

AFSC 2R051

Author: MSgt Clifton Yelverton
363rd Training Squadron
82nd Training Group (AETC)
363 TRS/TRR
520 Missile Road
Sheppard Air Force Base, Texas, 76311-2261
DSN: 736-2146
E-mail address: clifton.yelverton@us.af.mil;
363TRSCDCWriters@us.af.mil

Instructional Systems Specialist: Todd Knowles

Editor: Evangeline K. Walmsley

Air Force Career Development Academy (AFCDA)
Air University (AETC)
Maxwell AFB, Gunter Annex, Alabama 36114-3107

Material in this volume is reviewed annually for technical accuracy, adequacy, and currency. For SKT purposes the examinee should check the *Enlisted Promotions References and Requirements Catalog* to determine the correct references to study.

This CDC 2R051, *Maintenance Management Analysis Journeyman,* volume 2, takes us into the functional concept and capabilities of maintenance information systems. Managing these databases requires us to know how they operate. We are going to look at the essential technical and administrative responsibilities we perform in maintaining IMDS.

Unit 1 starts by explaining the administration and coordination process needed to operate IMDS on a daily basis. Next, we'll delve into additional data systems that interface with the MIS, followed by the appropriate actions required in the event of an aircraft accident.

Unit 2 explains the database concepts that make up the IMDS central database, concepts of data modeling and data normalization, as well as the security features of IMDS. This unit is important because it provides the in-depth knowledge needed to understand the physical structure of a database, how it operates, and what safeguards are in place to ensure only authorized individuals have access to only the data required to perform their job.

Unit 3 introduces you to the data retrieval programs used to query the database. Existing reports provide much of the data we use each day, but quite often you'll need specific data or reports that can only be accessed if you write a retrieval from scratch. This section teaches you the skills needed to custom tailor reports that help save valuable time and provide specifically targeted data; ultimately making you more efficient at your job.

A glossary is included for your use.

Code numbers on figures are for preparing agency identification only.

The use of a name of any specific manufacturer, commercial product, commodity, or service in this publication does not imply endorsement by the Air Force.

To get a response to your questions concerning subject matter in this course, or to point out technical errors in the text, unit review exercises, or course examination, call or write the author using the contact information on the inside front cover of this volume.

NOTE: Do not use Air Force Instruction (AFI) 38−402, *Airmen Powered by Innovation,* to submit corrections for printing or typographical errors. For Air National Guard (ANG) members, do not use Air National Guard Instruction (ANGI) 38−401, *Suggestion Program."*

If you have questions that your supervisor, training manager, or education/training office cannot answer regarding course enrollment, course material, or administrative issues, please contact Air University Educational Support Services at http://www.aueducationsupport.com. Be sure your request includes your name, the last four digits of your social security number, address, and course/volume number.

This volume is valued at 18 hours and 6 points.

NOTE:

In this volume, the subject matter is divided into self-contained units. A unit menu begins each unit, identifying the lesson headings and numbers. After reading the unit menu page and unit introduction, study the section, answer the self-test questions, and compare your answers with those given at the end of the unit. Then complete the unit review exercises.

Page

Unit 1. Maintenance Information System Administration

AS A MAINTENANCE MANAGEMENT ANALYST with database manager (DBM) responsibilities, you are accountable for monitoring Integrated Maintenance Data System (IMDS) operation, retrieving IMDS data, and maintaining IMDS data files. This, to say the least, is quite a job because it requires you to become knowledgeable of all aspects of maintaining an efficient and accurate database.

This unit begins with an overview of the system operation of IMDS. We will then discuss some critical administrative functions for using an operational database. Then, you will learn how the Air Force Life Cycle Management Center (AFLCMC)/Business Enterprise Systems (BES) Directorate communicates with the field concerning problems discovered. Next, we look at other systems with which IMDS interfaces. Further, we will cover the general causes of failures and errors. We will also cover online programs used for IMDS/Reliability and Maintainability Information System (REMIS) reconciliation program and a DBM tool known as the Universal Data System (UDS) monitor. Finally, we learn what to do in the event of an aircraft accident.

1–1. Integrated Maintenance Data System Administration and Coordination

As a DBM, you are responsible for coordinating with functional users and affected agencies concerning loading of new system releases, special programs, and changes to existing programs. You must inform users every time the Automated Information System (AIS) is to go down because of programs that must be run immediately or that are scheduled to run. You need to develop a schedule if there are programs to run on a regular basis. Understanding the main functions of IMDS is important to your job. We begin this section with an overview of system operation. Then we look at some critical IMDS system administration procedures, particularly in the area of scheduled and unscheduled downtime and off-line processing. Lastly, we cover the various forms used to report problems and inform users of system changes and updates.

201. System operation

The IMDS is the standard Air Force base-level automated maintenance information management system. This means that users enter maintenance-related data from their maintenance work centers at the base where they work. IMDS is vital for effective and efficient management of weapon systems worldwide. This system is designed to support aircraft communications-electronics (C-E) and equipment maintenance activities at worldwide operating bases, including those at Air National Guard (ANG)/Air Force Reserve (AFR) sites, and selected North Atlantic Treaty Organization (NATO) locations.

Purposes

IMDS has many purposes. They are discussed in the following paragraphs.

Provides communication

IMDS provides the capability for maintenance personnel to communicate with a central, base-level computer via remote terminals in virtually all maintenance work areas. All remote terminals, using personal computer (PC) desktops or portable PCs, are located in the maintenance work centers and offices where IMDS is used.

Automates maintenance data collection and work order systems

IMDS automates maintenance data collection and maintenance work order systems through online remote terminals connected to a single standard base-level computer (SBLC) system throughout the maintenance complex. IMDS updates information for other users to use. Data entered in one program from one work center automatically updates the database, where another user from another work center or agency can inquire about that data.

Automates aircraft information

In addition, IMDS automates aircraft history, aircraft scheduling, aircrew debriefing processes, and provides a common interface for entering base-level maintenance data into other standard logistics management systems. Users from other organizations using other information systems related to aircraft information interface with IMDS. For instance, supply, operations, engine management systems, and command level information systems all interface with IMDS.

Enhances design and increases readiness

Finally, IMDS enhances the front end design of new weapon systems and increases the readiness and sustainability of existing weapon systems. A centralized database system improves the availability, accuracy, and flow of essential maintenance, operational, and supply information. This is critical to the wartime readiness and operational support of aircraft, C-E, missile maintenance, and support equipment.

The following table provides a quick overview as to the IMDS capabilities.

Current IMDS Capabilities		
Automated Debriefing	Training/Personnel	Automated Scheduling
IMDS/Supply interface	Comprehensive Engine Management System (CEMS)	Inspection/Time Change
Job Data Documentation	Status and Inventory reporting	Asset Research
Specialist Dispatch	Operational Events	Configuration Management

General functions

The three general functions IMDS performs include updating the database, retrieving information from the database for local use, and reporting data required for higher headquarters.

Updates the database

The database maintenance functions provide the capabilities to enter new data, change existing data, and delete erroneous and obsolete data from the database. Extensive editing of input transactions is accomplished programmatically to ensure that only accurate data is entered into the database.

Retrieves information from the database for local use

The retrieval information function provides for two types of retrieval. These are the processing of small volume retrievals online and the preparation of reports and listings by background batch programs that process independently of the online system.

Online retrievals

Online retrievals are processed at the time the transaction is input. The output is returned to the requesting remote terminal.

Background products

Background products are processed under the control of the Unisys 2200 Executive System. System saturation, the amount of data to be extracted, and communications traffic all determine how quickly the products are processed. Background products are output on the high-speed printer at the Defense Enterprise Computing Center (DECC) or on the unit's Enterprise output manager printer.

Reports data required by higher headquarters

Data required by higher headquarters is collected as a by-product of the normal base-level operation of IMDS. It is then automatically extracted and transmitted via file transfer protocol (FTP) to interfacing systems (e.g., REMIS) on a periodic basis by IMDS interface programs. Data such as job data documentation (JDD) and debrief data is extracted and transmitted daily. Other data, such as that relating to aircraft status, inventory and utilization or configuration changes, is transmitted even more frequently (hourly).

Operating concept

IMDS is an event-oriented system. In most cases, data is entered to update the database because of an activity that occurred in the maintenance environment. Retrieval of information from the database is dictated by the needs of the functional user. This operating concept is implemented through online processing of data and the use of a networked database structure.

Online processing capability

Online processing capability is provided by access to IMDS via remote terminals located in the work areas. The user may enter data or retrieve information as the need arises. As a result, the database is maintained in a current state at nearly all times and information retrieval reflects up-to-date conditions.

Networked database

Use of a networked database enhances the online processing of the data by permitting data that has been recorded one time to be accessed and used by multiple subsystems. This reduces the amount of data the user would otherwise be required to maintain in the system. Furthermore, since commonly used data occurs only one time in the database, the accuracy and data integrity are improved.

The user may interface with IMDS in the following three ways:

- In day-to-day operation, the user normally interfaces with the system via the remote terminals.
- Those users supported by the Remote Job Entry Terminal System (RJETS) enter transactions via their IMDS remotes. Products produced by background programs are routed to the unit via RJETS.

- Background batch programs may be processed by submitting Executive Control Language (ECL) instructions in DEMAND mode rather than processing the transaction identification code (TRIC) through a remote in an online mode.

202. System procedures

This lesson covers the importance of critical system administrative procedures, scheduling batch programs, downtime, notifications, off-line processing, and recovery processing.

Batch programs

Batch programs process large amounts of data for transfer, update, or retrieval purposes. Running a batch program is known as batch processing or off-line processing. Batch processing usually takes a long time to run, depending on the amount of data involved. These batch programs require that online processing, called transaction interface package (TIP) mode, be down for most of the time to run efficiently. Other programs may be run while the system is on line; however, this slows down online processing and inconveniences the user.

Programs

Batch programs come in different forms. They are classified based on the functions they perform. Some of these functions include updating the system, copying and transferring the database to tape or other medium, and retrieving large amounts of data stored on the database.

Update

System releases can contain batch programs that update the IMDS database with changes to IMDS programs or data. They are usually run by the DECC in coordination with the AFLCMC and the host DBM. Special programs are also developed for special cleanup purposes such as interactive query utility (IQU) or query language processor (QLP) programs.

Transfer

Some utility programs are designed to transfer the database to another medium, usually to a tape. These types of programs are the save programs.

Retrieve

IMDS background programs are designed to retrieve data for the user. Generated runstream (GENRUN) programs are modified background reports. IQU or QLP programs are also primarily used for retrieving specific data from the database.

Scheduling batch processing

As the DBM, you coordinate with the system monitor at the DECC and with the base functional users and affected agencies concerning the operation of IMDS whenever it impacts the user. You must inform users every time IMDS online processing, or TIP mode, is to go down because of batch programs that must be run.

Impact to user

Develop a schedule if there are batch programs to run on a regular basis, especially if the program affects the users' computer processing. The main consideration when you produce a schedule to run batch programs is to allow the least impact to the users. Developing a year-long schedule for regular programs enables you to prioritize the monthly utility programs required to keep an efficient database. Then, break down the schedule for monthly, weekly and daily runs. Coordinate this schedule with the DECC because sometimes its batch processing takes precedence over an individual base's schedule. Reach a compromise with the DECC without sacrificing the mission of your IMDS users.

Factors

You should consider many factors when building your batch program schedule. Evaluating these factors should balance customer needs with requirements of the IMDS system and mainframe. Issues such as system traffic and required system maintenance impact the availability of system resources.

Consider the factors in the following table when developing a schedule:

Factor	Explanation
Shifts	Are your users on a 24-hour shift operation seven days a week? What shift uses the least on-line processing?
Time of day	After you have determined shift operation, determine what time of day users process on line the most. This is sometimes referred to as prime time. Experience shows that personnel process less during shift turnover (shift change) and lunch breaks. Running a batch program during these times certainly affects fewer users.
Batch program	Its purpose (update, transfer, retrieve) and scope of transaction. What type or types of data will it process? Know the features of the program. Can you abort the program and suspend the run; then continue later from the point where you stopped it? You may be able to run the program in several parts.
DECC processing and preventive maintenance schedule	Our mainframe resides in the DECC. The computer operators run multiple batch programs—some can run concurrently. Check with the DECC concerning the programs you desire to run regularly. Ensure that they don't conflict with other programs and scheduled preventive maintenance.
Size of database	This determines the amount of time to process. If you have a large database, expect the program to run several hours.
Support	Is there sufficient support from the base network control center (BNCC), the DECC, or the field assistance service (FAS)? Arrange for a standby support if these agencies are normally available only on weekdays.

System downtime

One of the most important areas for constant analysis is the time your mainframe computer equipment is down. When monitoring computer maintenance, direct your effort primarily toward the analysis of scheduled and unscheduled downtime. Computer maintenance affects your IMDS processing schedule. You will have to schedule online and batch processing time around scheduled downtime and revise existing schedules when unscheduled downtime occurs. Consequently, you and the DECC must devote special attention to both scheduled and unscheduled downtime to ensure uninterrupted flow of information to the functional user (the customer).

Notification

As an IMDS DBM, you have the overall responsibility of coordinating with your functional users on all information concerning IMDS. This means you must ensure your users are kept up to date on all IMDS issues, such as availability and downtime. Also, you must make sound decisions concerning system operation at a moment's notice. Do not take these responsibilities lightly. This lesson discusses the use of a checklist to demonstrate notification procedures and how to implement decisions affecting system operation. Your knowledge of notification procedures and sound decision-making concerning system operation has a major impact on your credibility as a DBM.

Develop notification checklist

There are times when problems develop in IMDS or the system is unavailable for use by your functional users. Problems can range from incorrect record pointers to software coding mistakes. System non-availability could be due to bad weather, hardware or software failure, database

maintenance, the loading of a system release, etc. Whatever the reason may be, you must notify the functional users when you are aware of current or future system problems and/or non-availability. The best way to ensure users are promptly notified is to develop and implement a checklist. By developing a notification checklist, you can be assured that the correct information is relayed to each user you support. For example, if the system suddenly goes down because of lightning in the area, and the only person in the office at the time is a newly arrived airman, how are your users going to know what is happening if the notification procedures are not written down somewhere and easily accessible to any available analyst?

Notify functional users

Ensure the checklist includes the address and phone number of each user who must be notified of scheduled or unscheduled downtime. Your notification checklist should include each subsystem manager. Normally, there is a primary and alternate point of contact for each unit analysis office or work center. The host DBM determines the IMDS method of system recovery based on length of downtime and volume of transactions. The host DBM then notifies each unit DBM of the recovery method to be used and notifies functional users in the unit. Each unit DBM notifies their work centers of the selected recovery method.

Notify senior maintenance leaders

The IMDS manual tells you who and what to do in events like this. Refer to Air Force Computer Systems Manual (AFCSM) 21–556, *Integrated Maintenance Data System (IMDS) Central Database (CDB),* Volume 2, *Team Scheduling*. This manual states that the senior maintenance leaders (operations group and logistics group commanders and staff, if affected) must be informed. Major command (MAJCOM) instructions also direct the host DBM to inform the MAJCOM DBM of system downtime exceeding 24 hours. Find out if there are local operating instructions (OI) concerning this matter.

Record time and date contacted

Make sure the checklist also includes a place to record the time and date you notified each point of contact. You may be called by a user asking why they weren't contacted when the system went down. If you recorded the name, time, and date of your initial contact, you can tell the caller whom you contacted. The caller can converse with their local point of contact to see why the caller was not notified. This also *proves* you contacted specific users.

Contact Defense Enterprise Computing Center

Coordinate with the DECC as to how long the system is to be down and the nature of the problem. By doing this, you are able to provide your users with the nature of the problem and a sound estimate as to when the system may become available. If the scheduled period of downtime is more than four hours, notify the system monitor to save the IMDS database before shutdown.

Indicate the nature of the problem

If possible, identify the nature of the problem to your users. By doing so, your users feel they are being kept informed of what's going on and able to make better management decisions regarding system downtime. This also boosts your credibility with them.

203. Monthly utility programs

Maintaining a clean and updated database is a constant process. It is similar to routine preventive maintenance performed on vehicles. To prevent major problems that can shut down or slow down the system, it is important that you run programs that check for errors and correct those errors or problems while they are still considered minor.

In this lesson, we introduce you to the different utility programs used to maintain the database. You need to understand the characteristics of these programs so you may use them effectively to maintain your database with minimal problems.

Database utility programs

There are several utility programs you run to keep your database effective. Some are programs that automatically check and repair errors within the structure of the database; others we use to look for data errors and correct them. Although these programs can be run anytime, some are directed by MAJCOM instructions to be run monthly to ensure a smooth operating database. Running any of these utility programs may also require coordination with the online functional users, DECC, and the AFLCMC. They can slow down online processing and may interrupt batch programs being run by the DECC.

The following tables describe some commonly employed utility programs used to maintain the database. We describe them in terms of their key features, advantages and disadvantages in order to assist you in using them appropriately and effectively. We separate the tables based on their primary functions–to identify errors and to repair errors.

Database error verification utilities

The following table lists two utility programs used to identify and isolate database errors: the NDA500 and the Data Management Utility, also known as DMU Verify. Both are database verification programs the enterprise database administrator (DBA) runs on a monthly basis to ensure smooth operation, or as needed when problems are suspected. Applicable MAJCOM Air Force instructions (AFI) may specify the frequency and conditions of executing these programs.

Program	Key Features	Advantages	Disadvantages
NDA500	• Screen driven program. The enterprise DBA runs it for each ELC. • Used in DEMAND mode. • Processes as a batch run. • Verifies a set, a group of sets, or all sets; checks for records with errors within sets. • Recommends patches to correct the pointer errors that it encounters.	• Recommends patches to correct the pointer errors that it encounters. • Can be run concurrently with the database up. • Perform database verification on all or selected sets. • It only allows access to those sets with errors. • It also places the patches into a temporary work file to ensure accuracy before applying them to the on-line database. • Can run concurrently with a database save.	• Running this utility while the database is up may cause it to identify false errors if a particular record is currently in use. • Limited correction; only creates patches for those records in the chain identified, it may verify a set it cannot patch, or it may overlook an orphan set (a set with a record or group of records that have no owner record). • Cannot apply patch within program; must use data management utility to apply correction.
DMU Verify	• Initiated by the enterprise DBA. • Response-driven interactive	• More thorough than NDA500; provides more information	• Offline processing; database must be down.

Program	Key Features	Advantages	Disadvantages
	processing facility (IPF) subroutine procedure. • Identifies pointer errors. • Used in DEMAND mode. • Checks area linkage throughout the database. • Checks record sets individually to determine if pointer/linkage problems exist in a database. • Checks proper placement of random records throughout the database.	about the error. • Can verify pointer errors initially identified by NDA500.	• No batch processing allowed.

Database query and repair programs

These programs allow the user to view the database records and their relationship. Considered higher level utility programs than NDA500 and DMU Verify, they view and manipulate records by the use of commands. The enterprise DBA will run these programs and forward the report to each ELC DBM for review.

Program	Key features	Advantage	Disadvantage
Database Editor (DBE)	• Must have access to a special account in order to use DBE. • Used in DEMAND mode. • Allows the user to browse through the database. • Only utility to view pointers. • Acts as a batch/demand application. • Fix pointer errors.	• Used to fix pointer errors when other utilities (e.g. NDA500, DMU verify) do not suggest patches for the error. • Used in DEMAND mode.	• May corrupt database if used by inexperienced users.
QLP	• Retrieval language program used to extract data. • Two modes of operation. • Uses syntax structure. • Writes program that can automatically extract and manipulate data. • Used in DEMAND mode.	• May be run while database is up.	• May slow down online processing if changes are made to database.
IQU	• Retrieval language program used to extract data. • Must have access to a special account in order to use IQU. • Two modes of operation.	• Similar but more powerful than QLP. • Considered an advanced retrieval processor. • May be run while	• Greater potential to corrupt database by inexperienced users. • May slow down online processing if changes are made to database.

Program	Key features	Advantage	Disadvantage
	• Uses syntax structure. • Writes program that can automatically extract and manipulate data. • Used in DEMAND mode.	database is up. • Syntax less restrictive than QLP.	

Personnel update

The monthly man-hour summary (WAH) and the work-hour update (WHC) are IMDS programs used to maintain an accurate account of maintenance man-hours for an organization. The complete instructions for running these two related programs are found in AFCSM 21–571, *Integrated Maintenance Data System (IMDS) Central Database (CDB)*, Volume 2, *Database Management*. Let's briefly discuss each program.

Monthly man-hour summary

The monthly man-hour summary (NFS910, TRIC: WAH) program updates the current month's assigned available man-hours (WAHM–363) and creates the monthly man-hour summary (AFS91P) print file. This program is run on or about the first working day of each month. The host DBM is responsible for coordinating the running of WAH between all units and DECC because processing is for all units. The host DBM is also responsible for preparing the appropriate WAH transaction. Before executing WAH, the host DBM must ensure that the database has been saved, and units have processed all required maintenance personnel record (MPR) load, change, and delete options.

NOTE: For clarification, these type of software updates begin with the acronym NFS (Network File System), and the last three digits identify the specific program being discussed. For example, in the preceding paragraph, the numbers 910 identify the specific program within the NFS.

Work-hour update

We use the work hour update (NFSO40, TRIC: WHC) program to update the WAH when the man-hours previously input were incorrect or a special requirement exists where AFR or ANG units require more hours than are automatically assigned for the current month. When using this program, exercise caution and use it only after the unit requiring the change has coordinated with the host DBM. This program can adjust current or previous hours by adding or subtracting by unit, organization, or work center.

Database manager responsibility

When you are assigned the duties of the DBM, you are expected to know and be proficient with these programs. We have only touched briefly on the different programs and their impact on the database when executed. Many IQU and QLP programs that enhance your database maintenance are written by the AFLCMC. These programs look for records that are known to cause database problems. Being familiar with these two programs, for example, enables you to understand how the program runs in repairing your database, and also prepares you to check for unexpected problems in case these programs abort. You must contact the DECC to give you access to your database using these programs: DBE, QLP with update and IQU with update. In this career development course (CDC), we will only focus on QLP. There are advanced courses available to increase your knowledge and build your confidence in running these programs. After you qualify, it is up to you to select the best solution to your database problems and situation.

204. Special programs

There are three programs that require special mentioning, specifically the JDD delete history, equipment transfers, and the annual standard reporting designator (SRD) reconciliation. As an IMDS DBM, it is your responsibility to oversee these programs. We'll discuss each in this lesson.

Job data documentation delete history

With the arrival of the IMDS central database (CDB), many of the programs base level DBMs used to run for their respective bases were changed to run from the enterprise level (entire database). As a result, the Enterprise DBA shop at Maxwell Air Force Base (AFB), Gunter Annex controls when these programs run. One such program is NFS760, TRIC (SHM), JDD delete history.

SHM takes the history data associated with maintenance documentation, inserts the records into the Relational Data Management System (RDMS) historical database and then deletes the data from the Data Management System (DMS) active database. While IMDS is on-line, the program starts automatically every Wednesday and retains no more than 92 days of history on the DMS. The number of days left on the active database is scalable, which means the number of days can be adjusted, and is used to balance the overall number of active database records versus its impact on overall IMDS response times. By having fewer records in the DMS and storing the history in the RDMS, IMDS operates at peak performance.

To verify SHM runs each week, use the program NFSRU0, screen 730, TRIC RUN, Scheduled Runs. From the graphical user interface (GUI), choose the "Inquiry" transaction and check the "All Scheduled Runs" and "System Level Runs" boxes. The output lists the frequency of the run, next scheduled date and last started/completed dates. You can also select "Unfinished or Errored Runs." Contact the DECC DBA at Tinker AFB or Field Assistance Service at Maxwell AFB, Gunter Annex if you encounter any problems. Detailed procedures for SHM and screen 730 are in AFCSM 21–571, Volume 2.

Equipment transfer process

As a DBM, one of your goals is to use your expert knowledge of IMDS capabilities to help when you can. Quite often, items tracked in IMDS are physically transferred from one base to another. If personnel at either base have to manually load or delete the associated records in IMDS, an enormous number of man-hours are wasted. Fortunately, there is a program that will accomplish an equipment record transfer and save valuable time for everyone involved.

Program NFS7U0, TRIC Transfer between ELC (TBE), is used to transfer the following item's equipment records from one ELC to another ELC within the IMDS CDB:

- Aircraft.
- Installed engines.
- Uninstalled engines.
- C-E.
- Aerospace ground equipment (AGE).
- Support equipment-identification (ID) and all installed parts.
- Time compliance technical orders (TCTO).
- Maintenance events and inspections.
- Time changes.
- Uninstalled part serial numbers.

Program NFS7U0 is executed via an IMDS screen or from a demand terminal using a GENRUN created runstream. There are three options, as presented in the following table:

Option	Description
Check	Performs all edits on the losing and gaining ELC databases and produces a list of major errors/differences, but does not copy the equipment to the gaining ELC. Must be executed prior to executing the "Transfer" option.
Transfer	Performs all edits on the losing and gaining ELC databases and copies the equipment to the gaining ELC if no major errors were found. If any major errors were found, they must be corrected and the transfer option re-executed.
Job standards (JST) Card	Performs numerous edits on job standards and TCTOs to include associated items such as automated history records, open jobs, configuration tables and debrief data. These are a few of the items the program checks. Review the extensive list in AFCSM 21–567, *Integrated Maintenance Data System (IMDS) Central Database (CDB)*, Volume 2, *Equipment/Personnel Transfer,* prior to running this option.

Work with local personnel to ensure errors identified on the respective reports are corrected. It is imperative you coordinate with the gaining base to ensure a timely transfer and the appropriate duty sections are involved. At a minimum, contact the gaining unit's host database management section and maintenance management production office. Detailed procedures for the equipment transfer process are in AFCSM 21–567, Volume 2.

Annual standard reporting designator reconciliation

Many AISs interface with and are dependent on IMDS. One of the most important is the Standard Base-Level Supply System (SBSS). The SBSS is used by maintenance and supply personnel to order parts for all aircraft, missiles, C-E, support equipment and a myriad of other equipment in the Air Force inventory. Supply personnel are required to annually conduct an SRD verification, also known as a reconciliation.

SRDs are three position codes that identify a specific system or weapons system. Examples are SRD AKH, which represents an F–16C fighter and SRD BHM, a satellite radio set. An accurate SRD table is essential for the SBSS to maintain the appropriate spare parts and order items not readily in stock. Supply maintains its SRD table based on a disk file provided by the IMDS host DBM.

Program NFS840, TRIC QCC, screen 126, Maintenance Code Listing Inquiry, is used to create the disk file. Supply personnel contact the IMDS host DBM and request the SRD table. When contacted, process screen 126, using the "Create Standard Reporting Designator Reconciliation File for Use by SBSS" option. Once you process the screen, you are done; however, it's important to understand the entire process in case you need to troubleshoot a problem. Take a look at the following items to help better understand the process:

- IMDS host DBM processes screen 126, reconciliation file option.
- The IMDS file is automatically sent to all supply platforms.
- Supply processes program NGV567, SBSS/IMDS SRD Reconciliation.
- NGV567 reads the IMDS file and conducts the reconciliation.
- Based on the reconciliation, an SBSS file is created with expired SRD delete transactions.
- Supply processes the SBSS file with the delete transactions during SBSS end of day processing.

Once supply personnel process the final SBSS file and their data base is updated, the reconciliation is complete. It's important to note that the local SRDs (those with an L or Z in the first position of the SRD, except ZZZ) are not reconciled because they are unique to a particular base.

Detailed procedures for TRIC QCC are in AFCSM 21–571, Volume 2. Air Force Manual (AFMAN) 23–122, *Materiel Management Procedures*, and Air Force Handbook (AFH) 23–123, Volume 2, Part 2, *Integrated Logistics System-Supply (ILS-S), ILS-S, Standard Base Supply System Operations*, Chapter 6, *As-Required Reports*, are also very useful to troubleshoot any problems.

205. Off-line processing

Your unit's flying and maintenance mission will not stop or be slowed simply because IMDS is down or has lost connectivity. The Air Force mission is just too important. Therefore, in times of Maintenance Information System (MIS) non-availability, whether short or long term, maintenance personnel must have an alternate means of documenting maintenance. In this lesson, we discuss different means of processing maintenance data when IMDS is unavailable, or off-line processing.

System non-availability

Even though the mainframe is an efficient, reliable system, there will be times when the base-level computer is not available for use by either the maintenance organization or you, the DBM.

Some possible illustrations of mainframe non-availability include the following:

- Removal of classified data from system.
- Performing preventive maintenance.
- Correcting hardware or software malfunctions.
- Performing computer housekeeping requirements.
- A terminal connected to the mainframe computer is not available.

Two additional causes for non-availability include the inability to connect to the Internet and a problem with network security.

Inability to connect to the Internet

There will be times when IMDS will be up and running perfectly, but your unit cannot connect to the Internet. Though GUI screens have made IMDS processing much easier and more user friendly, the user still has to have connectivity through servers and communications equipment. Not all of this equipment is owned or managed by the Air Force. A downed civilian contractor communications server in Texas can cause IMDS connectivity problems between Nellis AFB, Nevada and Tinker AFB, Oklahoma. An email server problem at your base can affect Internet access at a base across the country.

Network security issues

Network security is another issue that can affect your user's access to IMDS. Since the IMDS GUI is accessed through the Internet, it is subject to the same security protocols as any other system using the Internet. Ongoing efforts to keep Air Force information technology safe and secure make coordination between DBMs, workgroup managers, the BNCC, and the MAJCOM network operations and security centers (NOSC) more critical now than ever. Today's IMDS DBM must be aware of how changes in information security can affect their customers' access to the MIS.

Alternative data documentation

Whenever system non-availability occurs, the host DBM must provide alternate data documentation and processing plans in case the mainframe is not available for use for an extended period of time. There are several methods you can implement for off-line processing and eventual on-line recovery. Upon notification that IMDS is going offline, the affected work centers and agencies implement local backup procedures. To keep track of all transactions needing entry during system downtime, you

should implement one or more of the following basic methods: have users make and keep blank copies of their commonly used IMDS screens, create input text files, and create manual forms.

Copy integrated maintenance data system screens

Have users make and keep blank copies of their commonly used IMDS screens. These hard copies of screens allow your users to simulate using the system and recording the appropriate information with less chance of erroneous data being input when the system becomes available.

Input text file

Create an input text file on a PC or a data file in demand mode to record transactions for pseudo processing. Make sure users keep all transactions in chronological order. This way, transactions can be entered in the system as they occurred while the system was unavailable.

Create manual forms

You may also manually create your own form in the absence of any standard Air Force or Department of Defense (DOD) form that meets your needs. The form may follow the format of an IMDS program (screen) you normally use. During this time, maintenance documentation must continue by using manual backup procedures.

If IMDS is down for a considerable amount of time, the controlling agency for each unit may manually assign job control numbers (JCN). Once IMDS becomes available, the use of manual JCNs will be permitted to enter discrepancies that were discovered while IMDS was off-line.

Work centers may use AFTO Form 349, Maintenance Data Collection Record, to track data collection information for input until IMDS returns to on-line processing. Functional users may format all transactions as they occur during the downtime periods on applicable forms or other source documents that ensure the accuracy and integrity of the system.

Detailed procedures can be found in AFCSM 21–556, *Integrated Maintenance Data System (IMDS) Central Database (CDB),* Volume 2, *Introduction to IMDS CDB.*

Recovery processing

The host DBM, in coordination with tenant users, determines the method of updating the database once the system is back on line. The DBM determines the recovery procedure based on the length of downtime and how much volume of transaction is needed. The two basic procedures for data recovery, once the mainframe comes back on line, are manual remote and pseudo remote processing.

Manual remote processing

You can implement manual remote processing when the system is unavailable for only a short period of time and only when a relatively small amount of transactions require processing. Once IMDS is available, users simply input the transactions they recorded on paper into the proper IMDS screens.

Pseudo remote processing

Implement pseudo remote processing when the system is unavailable for a long time, or if you need to process a large amount of transactions. Pseudo processing can be performed either on line or off line. Pseudo files must be in chronological order, separated into one day’s transactions at a time. This facilitates correction of rejects and execution of background reports. Pseudo remote processing is done while IMDS is in file update (FUD) mode. While processing file updates, subsystem monitor representatives assist the DBM to collect rejects and correct them, if possible, at the same time.

Contingency processing

You must have a plan when IMDS is to be unavailable for an indefinite period of time. Be prepared for this condition because the processing of data doesn’t stop or get put on hold when IMDS is

unavailable. You continue to document important data and then send it as soon as possible to REMIS when the capability to use IMDS has been restored. This calls for a plan for contingency IMDS processing that you, as a DBM, must prepare and execute.

Alternate site

Whenever possible, select an alternate site for continuous IMDS processing when your home site (the main base) loses its capability to transmit on-line data to the mainframe for an extended time (weeks or months). This can happen if a catastrophic disaster, whether natural or man-made, such as sabotage, cuts off your online processing capability. The terrorist attack on the World Trade Center in New York City on September 11, 2001 was an unforgettable lesson in contingency planning. The alternate site must have a similar communication infrastructure (e.g., a local area network [LAN] or digital phone lines) with access to your mainframe at your DECC. Use the Defense Data Network (DDN) to establish communication between the alternate site and the DECC. The DECC may also use an alternate site for this function so regular processing at the main DECC facility is not interrupted.

This procedure also applies if the main DECC becomes incapacitated and may proceed contingency processing at the DECC alternate site. For this lesson, we discuss the procedures when your home base becomes unavailable for IMDS processing and the main DECC is intact. The type of facilities for an alternate site can be another nearby military base, an auxiliary field, an off-station tenant, or government organization. An agreement and a contingency plan must be established between your organization and the owner of the alternate facility.

Processing at alternate site

There must be full site support with a work area for IMDS personnel. Remote terminals and all peripherals must be provided. When normal communications can be established, use the DDN to process data on your home or alternate site database. You must provide to the base-level reporting activities (the functional users) the DECC number and routing indicator at the alternate site. Per AFCSM 21–556, Vol 2, the alternate site must provide for the continuous and uninterrupted processing and reporting of AFI 21–103, *Equipment Inventory, Status, and Utilization Reporting* for REMIS-reportable JDD data. Contingency processing should provide the full processing at the alternate site as is accomplished at the home site.

User action

If maintenance activities are still carried out at the home site, then the user hand-carries the manually documented data to the alternate site for processing. Users should follow the locally established contingency plan for manually documenting IMDS CDB data, if an extended period of manual processing is expected.

206. Notices and system releases

In this lesson, you will learn about the various advisory notices and system releases you review from AFLCMC to convey IMDS related information to functional users. At times this can get hectic, so you must stay current on all advisory notices and system releases.

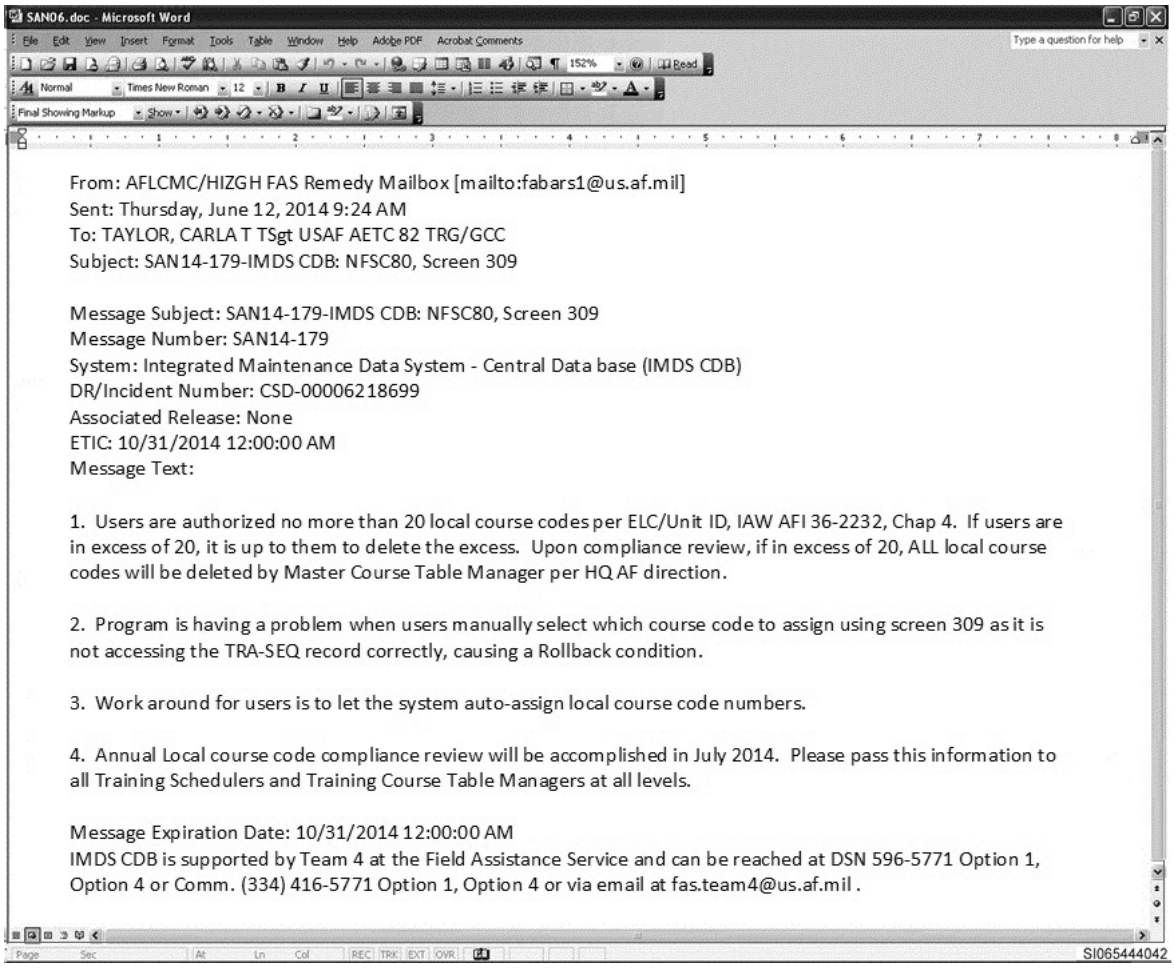
Advisory notices

Advisory notices are timely communications used by AFLCMC to provide the DECC and functional users with the necessary interim guidance and procedures required for continued system operation or to inform the user that continued system operation is not possible until corrections are received. An advisory notice is processed from AFLCMC to all DECCs, and in turn, to the host DBM at base level, where the DBM determines who is affected and what action a user must take.

The primary purpose of advisory notices is to provide functional users with technical solutions, interim guidance, and procedures for continued system operation and heads-up information. An advisory notice usually provides a temporary fix for a specific problem in a data system. It gives a description of the problem and identifies the agencies affected, with specific instructions for processing until a permanent fix is developed. Some advisory notices are transmitted along with a corrective action. If a corrective action is not provided, the repair may require in-depth research and programming by AFLCMC. Even though a temporary corrective action may not be included, advisory notices are AFLCMC's method of acknowledging that a problem does exist and to inform the affected users that the problem is being worked. The two types of advisory notices commonly used in our job include system advisory notice (SAN) and heads-up message (HUM).

System advisory notices

SANs pertain to system problems Air Force wide and are transmitted to all sites. SANs provide technical solutions or interim guidance of a technical nature to all users/locations who utilize standard software. The system advisory notice can be found on the Enterprise Database Management SharePoint site. You can also receive SANs automatically by registering online by contacting the FAS. Any new SAN is sent to you automatically by email. Figure 1–1 is a sample of a SAN sent by email.


SAN06.doc - Microsoft Word

From: AFLCMC/HIZGH FAS Remedy Mailbox [mailto:fabars1@us.af.mil]
Sent: Thursday, June 12, 2014 9:24 AM
To: TAYLOR, CARLA T TSgt USAF AETC 82 TRG/GCC
Subject: SAN14-179-IMDS CDB: NFSC80, Screen 309

Message Subject: SAN14-179-IMDS CDB: NFSC80, Screen 309
Message Number: SAN14-179
System: Integrated Maintenance Data System - Central Data base (IMDS CDB)
DR/Incident Number: CSD-00006218699
Associated Release: None
ETIC: 10/31/2014 12:00:00 AM
Message Text:

1. Users are authorized no more than 20 local course codes per ELC/Unit ID, IAW AFI 36-2232, Chap 4. If users are in excess of 20, it is up to them to delete the excess. Upon compliance review, if in excess of 20, ALL local course codes will be deleted by Master Course Table Manager per HQ AF direction.

2. Program is having a problem when users manually select which course code to assign using screen 309 as it is not accessing the TRA-SEQ record correctly, causing a Rollback condition.

3. Work around for users is to let the system auto-assign local course code numbers.

4. Annual Local course code compliance review will be accomplished in July 2014. Please pass this information to all Training Schedulers and Training Course Table Managers at all levels.

Message Expiration Date: 10/31/2014 12:00:00 AM
IMDS CDB is supported by Team 4 at the Field Assistance Service and can be reached at DSN 596-5771 Option 1, Option 4 or Comm. (334) 416-5771 Option 1, Option 4 or via email at fas.team4@us.af.mil .

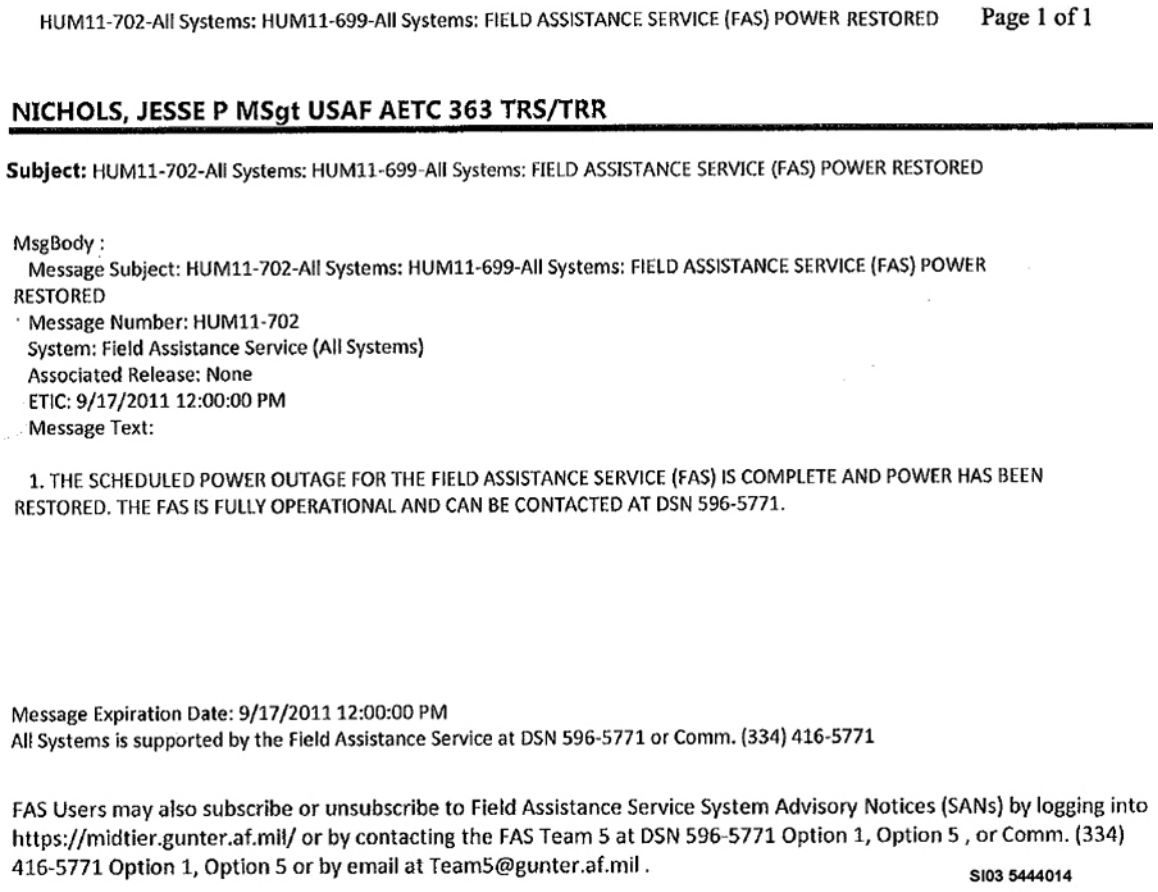
SI065444042

Figure 1–1. Sample, system advisory notice.

Heads-up messages

Along with general information, HUMs provide data processing agencies, like the DECC, knowledge of potential problems for which the fix or work around is not currently known. HUMs are also used as tools for the FAS to communicate with field agencies. They are sent as soon as possible after a problem is identified or when an advisory cannot be generated within 24 hours. HUMs may also be sent by email when you register online by contacting the FAS.

Let's say, for instance, that the FAS had a scheduled power outage and the power has been restored and they need to notify all users. Since it takes several days coordination time before a SAN can be sent out and all affected sites need to be notified right away, FAS uses a HUM. They would send out a HUM to all sites informing them the power has been restored and that a SAN is forthcoming. While a HUM is not a replacement for a SAN, a HUM is not limited to the system you are using, as shown in figure 1–2.



HUM11-702-All Systems: HUM11-699-All Systems: FIELD ASSISTANCE SERVICE (FAS) POWER RESTORED Page 1 of 1

NICHOLS, JESSE P MSgt USAF AETC 363 TRS/TRR

Subject: HUM11-702-All Systems: HUM11-699-All Systems: FIELD ASSISTANCE SERVICE (FAS) POWER RESTORED

MsgBody :
Message Subject: HUM11-702-All Systems: HUM11-699-All Systems: FIELD ASSISTANCE SERVICE (FAS) POWER RESTORED
Message Number: HUM11-702
System: Field Assistance Service (All Systems)
Associated Release: None
ETIC: 9/17/2011 12:00:00 PM
Message Text:

1. THE SCHEDULED POWER OUTAGE FOR THE FIELD ASSISTANCE SERVICE (FAS) IS COMPLETE AND POWER HAS BEEN RESTORED. THE FAS IS FULLY OPERATIONAL AND CAN BE CONTACTED AT DSN 596-5771.

Message Expiration Date: 9/17/2011 12:00:00 PM
All Systems is supported by the Field Assistance Service at DSN 596-5771 or Comm. (334) 416-5771

FAS Users may also subscribe or unsubscribe to Field Assistance Service System Advisory Notices (SANs) by logging into https://midtier.gunter.af.mil/ or by contacting the FAS Team 5 at DSN 596-5771 Option 1, Option 5 , or Comm. (334) 416-5771 Option 1, Option 5 or by email at Team5@gunter.af.mil .

SI03 5444014

Figure 1–2. Sample, heads-up message.

System release

System releases identify changes, corrections, and updates for IMDS. These system releases are provided by AFLCMC to each DECC and all functional users worldwide in order to have the release files placed on their mainframes. Typically the next day AFLCMC initializes the files within IMDS. They list SANs that have been cleared, as well as changes to the IMDS manuals.

Determining the effects of system releases

System releases are automated reports, released through each DECC, that list corrective actions and the systems affected. There is a narrative that describes the rationale for each change. This document is known as release rationale. If you are registered with a DECC for email communication, you will also receive the system rationale by email. System releases also have a section that lists all advisory notices that will be cleared after the system release is processed.

Coordinating system release load

System releases are loaded immediately by each DECC. There are instances where coordination is done with the ANG. Such coordination is usually attempted to avoid an ANG weekend as a courtesy; however, it is not a requirement. As the DBM, you are responsible for informing the users of the corrections and changes implemented in the release.

Online inquiries

You can view systems releases online using the IMDS program NFSBP0, *Online Inquiry of Release Rationale* (Screen 899, TRIC CHG), where CHG means "change." This program provides a narrative description of the changes that have been made to programs, TRICs, or screens as described on the release document, Air Force Form 636, Systems Change Release Document. This screen is in TIP mode and therefore available to all IMDS users. Within this program you can find information on multiple releases for changes made to TRICs, screens, and programs. You can also find the information on the most recent release.

You can view advisory notices and system releases online at the IMDS home page on the Internet. From the IMDS website, you can be linked to the FAS webpage. They maintain a file of all active SANs and releases. If your MAJCOM requires you to maintain your local file, you may devise your own system for local files; however, you must obtain internet access online from the FAS.

207. Difficulty reports and Information Technology Systems Requirements Document

IMDS problems at a unit are one of two types; namely, IMDS is not operating as designed or lacks a specific capability that a user requires. All subsequent changes to IMDS are made based on difficulty reports (DIREP) to correct operation problems or Information Technology Systems Requirements Document (ITSRD) to introduce a new feature the system user requires, which we will address within this lesson.

Difficulty reports

Software DIREPs are documented instances where software fails to perform as designed or documented. DIREPs are used by the AFLCMC to identify IMDS software problems and provide updated software to correct the problem. Due to the importance of maintaining updated and properly working software, this section covers the procedures to submit a DIREP.

Submission and evaluation

A DIREP is used when IMDS does not operate as designed. When an IMDS user suspects a problem, such as a TRIC not operating properly, the problem must be referred immediately to the DBM. The host DBM will decide if the problem can be corrected or resolved at the unit or base level, or if it should be referred to the AFLCMC through the FAS to be fixed.

Checklist

Before calling the FAS, always contact the DECC; your system monitor may be able to resolve your problem. To save you and any affected agencies time, check the following items before calling the FAS:

- Review user inputs for possible errors.
- Make sure it is not a software interface problem (Internet Explorer or Java).

- Make sure it is not a network problem.
- Review all SANs and release documentation and manuals for possible fixes to the problem.
- Coordinate with the system monitor at the DECC to see if the problem can be resolved.
- Make sure all releases that may affect the problem are loaded.

New or old problem

When the problem is beyond the ability of the DBM and the DECC to fix, the DBM calls the FAS. The FAS agent will search the troubleshooting database to determine if a similar problem has been called in and corrected, or if a DIREP exists similar to your problem.

If the problem has a temporary workaround, the DBM is given instructions on how to do the temporary workaround. If it is a new problem, the FAS agent contacts the IMDS functional manager for technical support.

Three stages

If IMDS functional managers cannot provide workarounds to the specified problems, they submit DIREPs. The submitted DIREPs go through three stages–initial review, evaluation and validation.

Stage	Explanation
Initial review	During the initial appraisal, the IMDS functional manager obtains more information from the DBM, such as error codes, error listings, printouts, etc. This information is needed to recreate the problem.
Evaluation	When the problem is verified, it is evaluated with possible solutions.
Validation	When a solution is found and verified to fix the problem, then it is validated and will be included in a future software release.

Eventually, a corrected version of the problem will be loaded to the database and the problem resolved. Review all release documentation sent by the DECC system monitor for the DIREP number assigned when the DIREP was created. If you find it, contact all affected users and let them know the problem is resolved.

Information Technology Systems Requirements Document

If a user feels that a current IMDS program does not satisfy a specific need to meet a unit's or Air Force mission, they can request to add or change the capabilities in IMDS by submitting an ITSRD.

When a user wants to change or add capabilities to IMDS, the user must submit an ITSRD through the host DBM section. Your job as the DBM is to evaluate the nature of the request and review it to ensure it is adequately described prior to submission to the FAS Remedy Mid-Tier website. The current link to the website for the FAS can be found in AFCSM 21–556, Volume 2. The MAJCOM representative will then review the requirement and forward it to the User Requirements Working Group (URWG). The URWG is responsible for review and approval of new and Legacy Systems requirements.

Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

201. System operation

1. How does IMDS automate maintenance data collection and maintenance work order systems?

2. What are the three general functions IMDS performs?

3. Explain the concept of an event-oriented system.

202. System procedures

1. What does the DBM develop if there are batch programs to run on a regular basis?

2. Why must the DECC and the DBM devote special attention to both scheduled and unscheduled maintenance?

3. What is considered the best way to ensure your users are notified of system non-availability?

4. Who determines the method of system recovery based on length of downtime?

5. What do you do if the system is to be down for more than four hours?

203. Monthly utility programs

1. What two programs are used to identify and isolate database errors?

2. What program allows you to browse through the database?

3. What are the two retrieval language programs we use to extract data from the database?

4. When do you run the monthly WAH program?

5. What two tasks must the host DBM ensure have been accomplished prior to executing the WAH?

6. When do you use the work-hour update?

204. Special programs

1. Which IMDS program deletes JDD history?

2. How often does JDD delete history run?

3. How many days of history does TRIC SHM retain on the database?

4. Which program is used to transfer equipment records from one ELC to another ELC?

5. What are the three options for program NFS7U0?

6. How often are supply personnel required to conduct an SRD verification?

7. What screen does the DBM process for supply during an SRD verification?

205. Off-line processing

1. What are the basic procedures for processing transactions during system downtime?

2. When would you implement manual remote processing?

3. When would you implement pseudo remote processing?

4. What type of facilities can we use for contingency processing?

5. What do you use to establish communication between the alternate site and the DECC?

6. Why do you need an alternate IMDS processing site?

206. Notices and system releases

1. What is an advisory notice?

2. Upon receipt of an advisory notice, what action would the DBM take?

3. What is the primary purpose of advisory notices?

4. How does the FAS use a HUM?

5. What is the purpose of a system release?

207. Difficulty reports and Information Technology Systems Requirements Document

1. When is a DIREP used?

2. Who initiates the submission of the DIREP?

3. What are the three stages a submitted DIREP goes through?

4. How can an IMDS user add or change the capabilities of IMDS?

1–2. System Interfaces

Do you know what systems or subsystems interface with IMDS on a daily and monthly basis? Do you know what IMDS data goes to higher headquarters for their review? This section covers the ways in which IMDS interfaces with selected systems and subsystems.

208. Interactive communication interface

Interactive communication interface (ICI) is a software package that controls the interface between two AISs. ICI is the standard system interface package for the SBLC. IMDS uses ICI to communicate with SBSS, REMIS, and others. In this lesson, the function of ICI that we will be concerned with is the interface between IMDS and SBSS.

Supply uses SBSS to control and route materials, parts, tools and equipment used in the Air Force. Most of the aircraft maintenance activity involves ordering parts to support the flying mission. IMDS

uses program NFSZ80, *Timer Interrupt Switch* (TRIC TIS). ICI provides message routing, recovery, and utility functions to IMDS and Supply system users. ICI routes data from IMDS to Supply (or vice versa), determines the validity of the interface request, and ascertains the status of the destination AIS and/or the communication link to the destination host. It also provides for message storage capability when the requested AIS is not in a mode to receive the message or when the communication link between IMDS and Supply is not available. A status code is returned to the sending terminal to identify the action that was taken. The host DBM is responsible for initiating the ICI software package (via screen 891) for the IMDS and Supply system interface, as well as ICI software packages for other interfacing programs that use ICI.

Initializing

IMDS screen 891, Automated Data System (ADS) terminal open, is the screen required to initiate ICI, which may only be executed from the ADS remote terminal. When this screen is displayed, the host DBM inputs a Y in the field termed DO YOU WANT TO START ICI? Once you enter this letter, the ICI software is initialized. To turn off ICI, enter an N instead.

Monitoring

There will be times when IMDS and Supply may be up and running smoothly, but ICI is not. For these reasons, the host DBM must notify unit DBMs as well as their own functional users on the status of ICI at all times. This prevents frustration on the part of the functional users, especially the maintainers, who might be ordering parts online through IMDS. If ICI is unavailable, the users have to contact supply to request that supply personnel order the part directly via the SBSS.

209. Reliability and maintainability information system interface programs

Earlier we discussed how to initialize ICI to provide for an IMDS and Supply interface. In this lesson, you will learn how certain IMDS programs are used to send and receive information from REMIS. As a DBM, you should be familiar with these programs.

There are several programs in IMDS that facilitate the transmission of information to and from REMIS. You should concern yourself with REMIS programs that pertain primarily with maintenance and administration information related to your base. Several REMIS reference tables go to the CDB – these tables contain reference data that supports maintenance documentation. Although we might not be directly responsible for correcting these errors, understanding how these programs work helps us to check for problems when certain REMIS information doesn’t arrive in a timely manner. We show several commonly used IMDS-REMIS programs in this lesson. You can find more IMDS-REMIS interface program in AFCSM 21–571, Volume 2.

Transmission of IMDS and REMIS data

IMDS and REMIS communicate to each other on a 24-hour basis. There are certain data that IMDS must send to REMIS and vice-versa. Several programs in IMDS were created to automatically send and receive these data in the form of files. IMDS data files that go to REMIS are known as KFS files; REMIS data files that go to IMDS are known as KRE files. One IMDS-REMIS program executes another IMDS-REMIS program automatically, usually when TIP processing comes online at the beginning of day or immediately after midnight. Some interface programs require manual input because they are concerned with reporting errors. Therefore, IMDS screens were provided to allow reporting or updates to certain information. There are also background programs available to send large volumes of data to REMIS. The two tables (online and background) summarize the transaction processing and description of these programs.

Online Programs			
TRIC	**Program and Title**	**Screen Input or Transaction**	**Description**
DDN	NFSEN0, DDN Address Record	Screen 881	This program loads the base user-ID and password for DDN transmission between IMDS and REMIS. It also reschedules REMIS inventory reconciliations.
REM	NFSEM0, REMIS Error Correction Menu	Screen 877	This program is the REMIS error correction menu.
REI	NFSEL0, REMIS Error Inquiry	Screen 876	This program reviews errors that REMIS returned to IMDS for correction.
TPR	NFSKI0, Table Validation Problem Reporting	Screen 077	This program is used to report REMIS table problems to the office of primary responsibility (OPR) at the appropriate Air Logistics center (ALC).
RIN	NFSGF0, REMIS Incoming DDN	Automatically executed by program NFS3Q0.	This program catalogs REMIS pushdown files into a DDF–145 record for follow-on program usage.
RFS	NFS3I0, REMIS File Separator	Automatically executed by program NFSGF0.	This program separates file data in DDF–145 records into individual files for future use.
RIR	NFS3P0, REMIS Inventory Reconciliation	Automatically executed by program NFSGF0.	This program provides REMIS with requested reconciliation transactions.
RFM	NFS3O0, REMIS Error Files Maintenance	Automatically executed by program NFS3I0.	This program extracts data from the AFS30D file and creates error records to be corrected and retransmitted to REMIS.
RFO	NFS3N0, REMIS File Output	Automatically executed every hour by other programs.	This program provides up channel reporting to REMIS for all JDD, status, and utilization transactions created in the IMDS database.
IFM	NFS3Q0, REMIS File Manager	Automatically executed by program NFSG20.	This program polls the IMDS 2200 system and catalogs all REMIS pushdown files into a DDF–145 record on each active database, which is stored for follow-on program usage.
DFS	NFS4Q0, REMIS Inventory File Output	Automatically executed by other programs.	This program transfers files from the IMDS system to the systems designated in the database records in REMIS. The database record for that file is modified to reflect that the file has been successfully transferred.
DMB	NFS0F0, Debriefing MOA Builder	Executed daily by the FSA. Manual input option is made through a DS remote terminal.	This program creates debriefing transactions, which are sent to REMIS.

Background Programs			
TRIC	**Program and Title**	**Transaction**	**Description**
OTR	NFS1X0, Operating Time Reinitialization	Run in DEMAND mode.	This program runs against the entire ELC database and reinitializes REMIS (via 8554 memorandum of agreement [MOA]) with the operating time for installed aircraft and missile part number/serial number (PNSN) items, installed engines, and PNSN items which have been installed on them.
JSM	NFS0N0, JDD Send MOA	Automatically executed by program NFS0E0.	This program selects, formats, and reports all JDD transactions (daily and monthly) to REMIS.

Manual online programs

Certain online programs require manual entries through the program screen. You encounter these programs often in your job as a DBM. It is to your benefit to understand their significance.

Defense data network address record

DDN address record (Program NFSEN0, TRIC: DDN—screen 881) loads a base-level user-ID and password that authorizes DDN transmission between IMDS and a host ID at a REMIS site. This program also loads an alternate base-level user-ID and password, start date, and start time. The system monitor at DECC provides both the primary and alternate user-ID and password to the host IMDS DBM. This program also changes the scheduled start date and clock hour for REMIS scheduled inventory reconciliations. This program allows communication between the base level IMDS and REMIS. NFSEN0 also provides a list of all files, which have been received from or transmitted to REMIS within the last 72 hours.

REMIS error inquiry

REMIS error inquiry (NFSEL0, TRIC: REI – screen 876) allows users to review maintenance data record errors that have been returned to IMDS from REMIS for correction. There are times when information being transferred from IMDS into REMIS is rejected by REMIS because of a probable error existing in the record. REI can be used to review JDD, inventory, status, and utilization-type-records. For example, REMIS may not recognize a work unit code that was used by a maintenance technician who input JDD documentation. Even though IMDS did not reject the input, REMIS would if it doesn’t recognize the code against its work unit code table. Once errors are reviewed and identified, they can be corrected by using TRIC REM, REMIS error correction menu.

REMIS error correction menu

REMIS error correction menu (NFSEM0, TRIC: REM— screen 877) allows users to either review or correct errors that have been returned to IMDS from REMIS. Only errors that have been recorded by REMIS within the last seven days from the date they were rejected by REMIS can be corrected. IMDS errors that have been returned fall into the four categories (options) listed in the table:

Category	Explanation
Job data documentation	These errors can be selected for review or correction by work center and employee number (optional).
Inventory	Accessed by entering type of equipment and start date.
Status	Accessed by entering the type of equipment and start date.
Utilization	Accessed by entering the desired type of equipment and start date.

Processing

TRIC REM can be processed by any IMDS remote, which means that any IMDS user has the capability to review maintenance data errors. However, in some cases, errors identified by REMIS may pertain to data that only the unit DBM can change. In this case, the database condition that caused the error must be corrected by the unit DBM. Once the errors are located and corrected in IMDS, the records are ready for resubmission to REMIS.

Table validation problem reporting

Table validation problem reporting (NFSKI0, TRIC: TPR – screen 77) provides the user the capability to contact and report problems with different tables directly to the end-item manager for the SRD. This program creates a record (RCS–376) that contains reporting information read by a report regeneration type program in IMDS. The RCS–376 records are a temporary holding area of large volume reporting data and are updated by several programs. Another program (NFS3N0) processes the RCS–376 record and sends it to REMIS within an hour. Problem reporting is only allowed for a specific SRD with a corresponding SRD–122 record.

Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

208. Interactive communication interface

1. What does ICI provide to IMDS and Supply System users?

2. Who is responsible for initiating ICI software for IMDS and Supply System users?

3. What screen do we use to initiate ICI?

209. Reliability and maintainability information system interface programs

1. What is the difference between KFS and KRE files?

2. Match each program description in column A with the corresponding TRIC code in column B. Each program description in column B may be used only once.

Column A	*Column B*
____ (1) This program runs against the entire ELC database and reinitializes REMIS (via 8554 MOAs).	a. DDN.
____ (2) This program selects, formats, and reports all JDD transactions to REMIS.	b. DFS.
____ (3) This program creates debriefing transactions, which are sent to REMIS.	c. DMB.
____ (4) This program is used to report REMIS table problems to the OPR at the appropriate ALC.	d. IFM.
____ (5) This program catalogs REMIS pushdown files into a DDF–145 record for follow-on program usage.	e. JSM.
____ (6) This program separates file data in DDF–145 records into individual files for future use.	f. OTR.
____ (7) This program loads the base user-ID and password for DDN transmission between IMDS and REMIS.	g. REI.
____ (8) This program transfers files from the IMDS system to the systems designated in the database records in REMIS.	h. REM.
____ (9) This program provides REMIS with requested reconciliation transactions.	i. RFM.
____ (10) This program provides up channel reporting to REMIS for all JDD, status, and utilization transactions created in the IMDS database.	j. RFO.
____ (11) This program extracts data from the AFS30D file and creates error records to be corrected and retransmitted to REMIS.	k. RFS.
____ (12) This program reviews errors that REMIS returned to IMDS for correction.	l. RIN.
____ (13) This program is the REMIS error correction menu.	m. RIR.
____ (14) This program polls the IMDS 2200 system and catalogs all REMIS pushdown files into a DDF–145 record on each active database.	n. TPR.

1–3. Database Errors, Monitoring and Reconciliation

When users start telling you that their IMDS program is not working, you know there is a failure somewhere in the system. When calls start to come one after another, you know there is something wrong with the system. Being able to recognize failures is a step towards fixing errors in the database. This section teaches you about system failures and errors and the use of a program you can use to look for errors before they cause failure.

210. Failures and errors

Database failures are the result of errors within the programs running on the mainframe. You need to distinguish online and offline program failures to give you a direction on what kinds of errors occur and the action you need to do. In this lesson, we will look at failures, types of errors, and abnormal disk conditions.

Failures

In a Unisys environment, the host DBM, in conjunction with the system monitor at the DECC, classifies the type of failure that occurs. System failure is when the mainframe computer goes down and communication to the computer is lost. Program failure is when a program does not successfully reach completion and error messages are displayed. The data management routine (DMR) generally "rolls back" all updates when a program failure occurs. If you follow proper procedures after a system failure, the database will be restored to a usable state. The DBM must be familiar with the two modes that affect users: online and offline. The online mode is TIP processing which is the majority of IMDS users. Offline mode is the environment with which the DBM is more familiar.

Online programs

System and program failures affect the user who is running IMDS online. The DBM can determine the appropriate action to restore processing once failure is determined. The following table shows the two types of failures and corresponding recommended actions.

Online Program Types of Failures	
Failure	**Recommended Action**
(1) System Failure	An online program that has not reached program completion is restarted automatically once the database has been restored by the DMR. No input action is required by the user.
• Invalid input	If the program finds invalid input during execution, the database is not updated. Correct the input transaction and reenter.
• Database rolled out or a corrupt database	When the program cannot access the database for the above reasons, the program exits and displays an error message. When an error message is received, notify the database management section. Upon notification that the database has been recovered, resubmit the input that caused this error.

Background batch programs

When a program failure occurs while running a background or batch program, there are three possible kinds of failure. The following table may guide you and the user should these kinds of failures occur.

Types of failures occurring during background or batch program operation	
Failure	**Recommended actions**
(1) Invalid input	Correct the input transaction and reenter.
(2) Database retrieval/store of record	The database is restored to a state prior to the program execution that corrupted the database. The error message and keys displayed by the program should be sent to the DBM.
(3) Unrecoverable error	This condition occurs when a program can no longer process. Examples of an unrecoverable error are an interrupt guard mode (IGDM), input/output error, and file error. These errors should be sent to the DBM to be reviewed.

Record errors

The two types of record errors that can be found in the IMDS database are data and pointer.

Data errors

Data errors occur when records store information incorrectly. This error could result from a user inputting information on the wrong screen or documenting information incorrectly. Another possible cause is if a TRIC is malfunctioning and storing information in the wrong records. The user's first indication of a data error is a TRIC inquiry or background report output that displays the wrong information.

Pointer errors

Pointer errors are normally errors created by DEMAND users, intentionally or unintentionally. TIP users, other than the DBM, cannot create pointer errors. Pointer errors sometimes can occur due to the following:

1. Power outages.
2. Hardware failure.
3. Lightning strikes.
4. Data stored on bad disks.
5. Refresh or short recovery executed improperly.
6. Long recovery executed improperly.

Categories

Records in the IMDS database are physically connected using links called pointers. As mentioned previously, errors can occur in these pointers. They can be likened to links on a webpage that point to the wrong file, point to a file that doesn't exist, or even point to no file at all. These types of pointer errors can further break down into two categories—structural and logical set errors.

Structural

Structural errors occur when the information or data does not follow the schema rules; for example, when a number of characters in a field within a record exceed the maximum allowed.

Logical set

Logical set errors can either be a structural or data error. One example would be an EVT–370 record shown as a member of both the EVT completed set (395) and the EVT scheduled set (394). Logical set errors usually occur in manual sets. As a DBM, you must have a general understanding of these types of errors and how to resolve problems when they become apparent. As rules of thumb when interpreting error conditions, determine the following:

1. If the user exercised correct input procedures.
2. The type of error—data or pointer.
3. What records the error condition may involve.
4. What sets the error condition may involve.
5. What method to use to pinpoint the error.

Abnormal disk conditions

An abnormal disk is an error in the system database that is stored on disk and that is unresolvable by the functional system analyzer (FSA). An abnormal disk is a reject condition and is identified by the reject phrase 99. The causes of an abnormal disk are many. Some errors do not indicate file problems, yet they could result from conditions that existed at the time the input was made. You might encounter one of these problems when you run a utility program.

When an abnormal disk occurs, the computer initiates a rollback (RB). This could result in the termination of the program you're running. If you are monitoring your program while it's running, you might be able to see the error codes on your screen. The system sends a print file to your remote when a program aborts. This gives you an idea of what went wrong when you analyze the problem.

Rollback

An RB simply erases the effects of input. In other words, if an input transaction, sometimes called a RUN-UNIT, made any changes to the database before it rejected with an abnormal disk, those changes would be rolled back. The DMR would return the database to its previous condition prior to the offending input. An RB applies to a single job. When an abnormal disk situation and rollback occurs, you receive a series of codes on the terminal. The computer provides the following three codes to help you identify and correct the error:

- Rollback error code.
- Function code.
- The DMR error code.

AFCSM 21–571, Volume 2 provides narratives for each of these codes. When you encounter an abnormal disk condition, go to Attachment 2 for a detailed explanation of each code as you refer to figure 1–3. An online version of the attachment is also found at the IMDS Online Database Help System, which we discussed previously.

Rollback error code

A rollback error code is a two-position code that pinpoints the actual cause of a problem. This code tells you why the DMR performed a rollback. One of the most often encountered rollback codes is 02. AFCSM 21–571, Volume 2, provides detailed explanation of the codes associated with abnormal disk conditions. Attachment 2 of the volume tells you that RB code 02 equals a deadlock between run units. You recall that a run-unit is an input transaction; in other words, two inputs tried to access the same data at the same time and deadlocked between each other. We sometimes call RB code 02 the busy signal because it is very much like the busy signal of a telephone. The user need only retransmit the input to correct this condition. Another common rollback error code is 14, which means you attempted to access a downed area. This error often occurs the first thing in the morning, normally indicating an aborted save (dump) from the night before.

DMR external errors

The DMR external error is made up of the function code and the error code. The rollback code only tells you that something went wrong in a general area when it encountered a problem. The DMR external error analyzes the abnormal disk condition and pinpoints what the DMR is executing at the time when it detected a problem. It is up to you then to troubleshoot further and verify that the error code confirmed the problem. Figure 1–4 shows the first page of the "Common DMR External Errors" chart. The top row contains the function codes. It shows the particular activity of the DMR run. The first column describes briefly the error condition for each row. The dots mean that there is no applicable condition. When a number exists, match up the number with the function code.

Database Management

A2.1.2 COMMON DMR EXTERNAL ERRORS.

This subsection lists common DMR external errors.

Common DMR External Errors

	01 CLOSE	02 DELETE	03 FIND/ FETCH	04 FREE	05 GET	06 KEEP	07 INSERT	08 MODIFY	09 OPEN	11 REMOVE	12 STORE	13 DEPART	14 IF	15 IMPART	16 MOVE	17 LOG	19 ACQUIRE
Area not open or or incorrect usage mode.	01	01	01	•	•	•	01	01	•	01	01	•	•	•	•	•	•
If condition evaluated legitimately as false.	•	•	•	•	•	•	•	•	•	•	•	•	01	•	•	•	•
Area-key violates within clause.	•	•	•	•	•	•	•	02	•	•	02	•	•	•	•	•	•
If condition error. Record not defined in schema as owner/member of set.	•	•	•	•	•	•	•	•	•	•	•	•	02	•	•	•	•
Occurs depending item count is out of range.	•	•	03	•	03	•	•	03	•	•	03	•	•	•	•	•	•
Maximum of 10 CALC keys exceeded.	•	•	•	•	•	•	•	04	•	•	04	•	•	•	•	•	•
Violation of duplicates not allowed clause.	05	•	05	•	•	•	05	05	•	•	05	•	•	•	•	•	•
No current record of area/record/set name.	•	•	06	•	•	•	06	06	•	06	06	•	06	•	06	•	06
End of set/area/ chain.	•	•	07	•	•	•	•	07	•	•	07	•	•	•	•	•	07
End of PA or IPA.	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	07
Record not defined in set/area, or record does not contain specified keys.	•	•	08	•	•	•	08	08	•	08	08	•	08	•	•	•	•
Improper area usage.	•	09	09	•	09	•	09	09	•	09	09	•	•	•	09	•	•
Attempted an inter-area modify.	•	•	•	•	•	•	•	10	•	•	•	•	•	•	•	•	•
Find param out of range.	•	•	11	•	•	•	•	•	•	•	•	•	•	•	•	•	•
No record or set occurrence satisfies criteria.	•	•	13	•	•	•	•	13	•	13	13	•	13	•	•	•	•

Figure 1–3. Rollback error codes, attachment 2.

Database Management

A2.1.2 COMMON DMR EXTERNAL ERRORS.

This subsection lists common DMR external errors.

Common DMR External Errors

	01 CLOSE	02 DELETE	03 FIND/FETCH	04 FREE	05 GET	06 KEEP	07 INSERT	08 MODIFY	09 OPEN	11 REMOVE	12 STORE	13 DEPART	14 IF	15 IMPART	16 MOVE	17 LOG	19 ACQUIRE
Area not open or or incorrect usage mode.	01	01	01	•	•	•	01	01	•	01	01	•	•	•	•	•	•
If condition evaluated legitimately as false.	•	•	•	•	•	•	•	•	•	•	•	•	01	•	•	•	•
Area-key violates within clause.	•	•	•	•	•	•	•	02	•	•	02	•	•	•	•	•	•
If condition error. Record not defined in schema as owner/member of set.	•	•	•	•	•	•	•	•	•	•	•	•	02	•	•	•	•
Occurs depending item count is out of range.	•	•	03	•	03	•	•	03	•	•	03	•	•	•	•	•	•
Maximum of 10 CALC keys exceeded.	•	•	•	•	•	•	•	04	•	•	04	•	•	•	•	•	•
Violation of duplicates not allowed clause.	05	•	05	•	•	•	05	05	•	•	05	•	•	•	•	•	•
No current record of area/record/set name.	•	•	06	•	•	•	06	06	•	06	06	•	06	•	06	•	06
End of set/area/ chain.	•	•	07	•	•	•	•	07	•	•	07	•	•	•	•	•	07
End of PA or IPA.	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	07
Record not defined in set/area, or record does not contain specified keys.	•	•	08	•	•	•	08	08	•	08	08	•	08	•	•	•	•
Improper area usage.	•	09	09	•	09	•	09	09	•	09	09	•	•	•	09	•	•
Attempted an inter-area modify.	•	•	•	•	•	•	•	10	•	•	•	•	•	•	•	•	•
Find param out of range.	•	•	11	•	•	•	•	•	•	•	•	•	•	•	•	•	•
No record or set occurrence satisfies criteria.	•	•	13	•	•	•	•	13	•	13	13	•	13	•	•	•	•

SI065444049

Figure 1–4. Common data management routine external errors page from attachment 2.

Function code

A function code is a two-digit code that identifies the function or operation the computer was performing when the abnormal disk occurred (i.e., open, or close). There are 17 function codes (01 – open, 02 – delete, etc...) that refer to the DMR activity. When the function code is used in conjunction with the DMR error code, the resultant code reveals the reason for the abnormal-disk condition.

DMR error code

The last two digits are the DMR error code—error number. This numerical code, when used in conjunction with the function code, pinpoints the problem that most likely produced the specific rollback error and/or abnormal disk condition. Three common DMR error codes are listed in the following table:

Code	Meaning
60	60—Unable to find starting point in set.
92	92—Database Key (DBK) found vacancy entry.
98	98—Invalid area code/page number.

There is no one particular way to correct abnormal disk conditions. The correction procedures depend on the condition revealed by the combination of the DMR rollback code, function code, and DMR error code. Many abnormal disk conditions are simply RB code 02, which only requires the user to reinput, while other abnormal disk conditions may indicate pointer problems or perhaps a more serious problem. If an abnormal disk condition exists that you are not sure how to handle, consult with your supervisor, your host DBM, your system monitor, or the FAS at Air Force Materiel Command (AFMC) AFLCMC Enterprise Services (HICS), located at Maxwell AFB, Gunter Annex in Montgomery, Alabama.

211. Program monitoring

The universal data system monitor (UDSMON) is used to monitor runs (user programs or jobs) that are being processed with the Universal Data System. It provides centralized functions of thread control, locking and queuing, system table control (like file description tables), database I/O, cache, storage management, and data recovery. The UDSMON can be used on the DMS and RDMS databases. UDSMON can be a very helpful tool in troubleshooting batch or application programs that aborted.

Uses

UDSMON is another tool to view the status of any run or application that you are processing, especially batch runs. The UDSMON provides for an integrated software environment for control and maintenance of user databases. Let’s say, for example, that you are updating the information on your database with a batch program provided by the AFLCMC (e.g. aircraft data cleansing). With UDSMON, you can view how the program runs with the basic activity such as finding, fetching, and changing database records. When a particular program aborts, it gives you an idea how far into the program the abort occurred.

When a program aborts, UDSMON enables you to monitor if the program rolls forward or rolls back. When a program rolls back, UDSMON assists you with troubleshooting in pinpointing the exact point where it aborted. It could point to a corrupted record or a conflict of database records.

Components

UDSMON contains many different program components used by system monitors, programmers, developers, and DBMs to monitor and troubleshoot programs running on the mainframe. The following is a list of major software components, or products:

- Exec system software.
- Integrated recovery utility.
- Logical data manager.
- Universal Data System control.
- Data Dictionary System.
- Data Management System.

To enter the UDSMON, input the following:

@UDS$$SRC*UTIL$.UDSMON APPL01

Upon execution, you are prompted with the main menu shown in figure 1–5. Listed are all the options you have available to monitor and review various runs on the system. However, only a few of them are useful to you as a DBM, and are listed in the following table. The other options are used by programmers.

```
UDS Monitor Menu.  Alias: APPL01, Appl name: APPL01, Appl no.: 1

 A:Status of all runs                    B:Buffer Space Usage
 C:Global Status of individual run       D:Any block of core
 E:File Usage Information                F:Altered Pages by run
 G:Rollback Error History                ?:Help with use of UDSMON
 I:DMR Schemas & Subschemas              J:DMR Status of individual run
 K:RDM Status of individual run          L:SFS Status of individual run
 M:Bank Fragmentation                    N:Retention File Information
 H:Terminate the Monitor


For each Monitor, there are predefined default functions. These are:
 'H'alt (or F1) to terminate the Monitor program,
 'M'enu (or F2) to return to this menu screen,
 'F'aster (or F3) to halve the sampling period (min = 100ms),
 'S'lower (or F4) to double the sampling period (max = 30000ms),
 'R'efresh (or F5) to re-paint the existing screen,
 'C'opy (or F6) to save the current screen to 'appnam*MONITOR$COPY.'.
-----------------------U-D-S-M-O-N----L-E-V-E-L--6-R-4---------------------
Enter Desired Monitor Mnemonic                                    S/985439044
```

Figure 1–5. Universal Data System monitor main menu screen.

Option	Explanation
A: Status of all runs	Gives you the status of all runs that are processing or waiting to process in the DMR.
E: File usage information	Tells you what database files and schemas are being used and number of pages being used by each file.
F: Altered pages by run	Displays the same information as option A plus the index (INDX) number (area code) and page number for each page the run ID has changed.

I: DMR schemas & subschemas	Lists the schemas currently being used and basic sizing information about each one.
J: DMR status of individual run	Displays what the DMR is doing with the run specified. It also tells you the current record, DBK, and command being performed.

Main menu

The main menu also shows you what keys to use to change the program's default settings. For each monitor, there are predefined default functions. They are described (briefly) on the bottom line of each screen. Those functions described by a single character (e.g., M for Menu), can either be input as that character, or as the equivalent function key (i.e., the 3rd function is equivalent to F3). Any other input, usually a 1 to 6 digit value, has no equivalent function key. There are six common inputs for each monitor, as shown in the following table.

Input	What it does
H (or F1)	Halts the program execution.
M (or F2)	Halts the current monitor and goes to the menu.
F (or F3)	Makes the sampling rate faster (by half, min = 100 millisecond [ms]).
S (or F4)	Makes the sampling rate slower (by double, max = 30000ms).
R (or F5)	This redisplays the screen (in case of any typing accidents).
C (or F6)	This copies an image of the screen to an alternate print file (catalogued, and +1 cycled) for later perusal.

212. Reconciliation

IMDS/REMIS reconciliation (NFS5B0, TRIC: RIS) is a highly dynamic program to correct REMIS errors for the C-E community. This program reconciles the IMDS and REMIS databases. It also identifies and/or corrects errors within the IMDS database. Unsuccessful REMIS initializations in the past highlighted the need for this program. The reasons for initialization problems were many, including software deficiencies and the lack of clearly defined software requirements in both IMDS and REMIS. Another problem was a lack of procedures and guidance in the field.

Functions

Many significant improvements came about from identifying all these difficulties, and the IMDS C-E subsystem has improved as a result. The NFS5B0 has the following three functions.

Identifies and/or corrects

It identifies and/or corrects C-E status and inventory problems in the IMDS database. The first time it is run, approximately one-third of all errors found are auto-correctable.

NOTE: The following is an example of an automatic correction scenario:

A report level for a piece of equipment is loaded incorrectly as "**R**" when it should have been "**Y**." This program identifies this error to the output product and automatically corrects the report level.

Reconciles

It reconciles the IMDS and REMIS databases. When you choose this option, the program builds a file of all reportable equipment and their reportable statuses and transmits that file to REMIS. You may also use the auto-fix capability of this program without sending transactions to REMIS.

Deletes and clears

It deletes and clears all REMIS errors on the IMDS database. The output report specifies whether or not the program can correct the error. 5B0 has replaced the 87X series screens for C-E error correction.

Using the program

This program can be processed by straight-line input or scheduled to run at a regular interval using screen 730. It is designed to run on the current ELC and the output is sent to the default printer. When processed by straight-line input or through screen 730, its progress can be monitored by background status inquiry and remote line printer. You can find complete instruction about NFS5B0 in AFCSM 21–560, *Integrated Maintenance Data System (IMDS) Central Database (CDB)*, Volume 2, *C-E Equipment Status and Inventory Reporting*.

Reportable equipment

TRIC RIS processes all REMIS reportable equipment and statuses not found in error; those automatically corrected are sent to REMIS regardless of whether they are gained, lost, opened, or closed. If the program finds a C-E inventory error that it cannot correct, it will change the report level of that piece of equipment to a "?" and the equipment inventory information will not be sent to REMIS until you correct all errors.

Correcting errors

After the program changes the report level for the equipment to a "?", every piece of REMIS reportable equipment that has a "?" for report level can be viewed in screen 800. The error may reside in the EQPP–158 record or one of its member records. You get an output printout of all the errors and their corresponding recommended corrective action. The unit DBM assigned to the C-E organization or the host DBM (if there is no unit DBM) corrects the error manually by going to the "culprit" record and does the necessary correction. Once these are corrected, some of the other errors are also cleared. It might take some time to correct the errors, depending on the errors the program found and the level of difficulty the DBM encounters during the correction process.

After you correct the errors, you may re-execute the NFS5B0 with the auto-fix on. The program then updates the corrected statuses by changing the "?" to the appropriate report. The status of the equipment is based on the mission capable (MICAP) indicator as set forth in the SRD–122 table. If it is MICAP reportable equipment, then NFS5B0 assigns a "Y." If it is non-MICAP reportable equipment, then NFS5B0 assigns a "P."

Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

210. Failures and errors

1. What is the difference between a system and a program failure?

2. How do you restart an online program when the database is restored after a system failure?

3. What does it mean when an unrecoverable error occurs while the system is running a batch program?

4. How does a user know when a data error may exist?

5. What type of users cannot create pointer errors?

6. Where do logical errors usually occur?

7. What is meant by a rollback?

8. Where would you find detailed explanations of the codes associated with an abnormal disk condition?

9. What is rollback error code 02 and what corrective action would you first try for an RB code 02?

10. What error code is used in conjunction with the function code to assist in analyzing an abnormal disk condition?

11. When you encounter an abnormal disk condition that you are unsure how to correct, to whom do you turn for assistance?

211. Program monitoring

1. What type of centralized functions are provided by the UDSMON?

2. What option tells you what database files and schemas are being used?

3. What option of the UDSMON provides the current record, DBK, and command being performed?

212. Reconciliation

1. What are the three functions of TRIC RIS?

2. What kind of C-E equipment is processed by TRIC RIS?

3. Where are instructions for using NSB5B0 located?

1–4. Aircraft Mishap Procedures

Every day the Air Force flies hundreds of sorties. Added up over a year, the number climbs to a staggering amount. During peace-time, most missions are either training or airlift related. During times of conflict, aircrews are the front line fighters for the AF. It is amazing, considering the demanding and often dangerous nature of flying, how good the Air Force's overall safety record is. Our pilots are highly skilled and our maintainers are well versed at providing safe, reliable aircraft. Unfortunately, no matter how careful everyone is, accidents happen. As an IMDS or G081 DBM, you play an important role when aircraft are involved in a mishap.

Regardless of which database you support, your quick response is important to secure the maintenance records associated with the aircraft that encountered the mishap. Your first priority is to ensure those records cannot be altered. An investigation board will convene and a crucial part of determining what happened will be their review of all maintenance and associated actions performed (or not performed) prior to the mishap. In this section, you will learn the minimum steps to take and the minimum reports to run depending on the database you support.

213. Aircraft mishap Integrated Maintenance Data System procedures

Your role after an aircraft mishap is pivotal to the investigation that follows. Securing the IMDS database and providing maintenance history records is crucial to attempting to reconstruct the root cause of the accident. The following are the five major steps you'll need to accomplish for IMDS:

1. Secure the database.
2. Process a database save.
3. Get specific aircraft details.
4. Process reports and inquiries.
5. Allow users back into the database.

Step 1: Secure the database

Initially, the most important step is to secure the database and ensure limited access. Immediately accomplish the following when you receive notification of a mishap:

Task	TRIC	Screen	Program ID	Reference
1. Place IMDS in FUD mode, check box on screen 891	TIS	891	NFSZ80	AFCSM 21–571, Vol. 2.
2. Enter User IDs of personnel allowed to process transactions while the system is in FUD. Limit this to DBMs or schedulers that might need to run products.	MIK	992	NFSZ70	

FUD allows users to login to the database, but they are not able to do anything. Screen 992 allows processing by only the personnel you authorize. Be very selective to limit the amount of authorizations you allow, if any.

Step 2: Process database save

Next, contact the DECC Unisys Team to initiate a database save. Normal saves expire after 14 days, so request the tape expiration date be changed to 1 year and annotate the tape number(s) for future reference. After the save is done, ensure the system is still in FUD by checking screen 891.

The database save is important if, at a later date, the mishap investigators need to review data not on the reports or inquiries. This valuable tool allows you to reload an exact copy of the database as of the day the mishap occurred. Contact the DECC for assistance if you are asked to do so. Make sure you tell them this is an old save and not to reload over the current database.

Step 3: Get specifics

Once you secure and save the database, the next critical step is to gather the specific details needed to run reports and do inquiries. At a minimum, you will need following information, if not provided when you were initially notified:

- Mission design series (MDS).
- Tail number/engine number(s).
- Serial number.
- Squadron and owning work center.
- Time and date.
- Name, number and title of the person notifying you.

The exact nature of the mishap and what was affected will dictate the specific items you require. Do not be afraid to ask for more information if needed.

Step 4: Process reports and inquiries

Using the information obtained earlier, you'll need to run the following background reports and on-line inquiries:

Background Reports				
Screen	**Program Title**	**TRIC**	**Program ID**	**AFCSM**
105	Maintenance History Report	QMH	NFS350	21–563, Vol 2
510	Significant Historical Data	SHD	NFS700	21–566, Vol 2
429	Equipment Transfer-Format 1	TRE	NFS140	21–567, Vol 2
396	Planning Requirements	PRA	NFS180	21–566, Vol 2
418	Automated Records Check for Equipment-IDs	ARC	NFS340	21–561, Vol 2
181	Deviation Detail Listing	DDL	NFS3B0	21–574, Vol 2
533	Serial Number Detail Listing Format #1, Input for Equipment-IDs	STL	NFS280	21–568, Vol 2
185	Code 3 Fix Time Report	FTR	NFS2T0	21–574, Vol 2
179	Pilot Reported Discrepancies Report	PRD	NFSIS0	21–574, Vol 2
Inquiries				
Screen	**Program Title**	**TRIC**	**Program ID**	**AFCSM**
380	Documented Maintenance List	DOM	NFSJ20	21–559, Vol 2
460	Summarized/Detailed Status for Equipment-ID	EST	NFSI30	21–564, Vol 2
726	Shop Equipment Operational Inquiry	SAE	NFSG50	21–565, Vol 2

Background Reports				
Screen	**Program Title**	**TRIC**	**Program ID**	**AFCSM**
810	Parts Tracked Inquiry	PTI	NFSB00	21–559, Vol 2
174	Debriefing Sortie Recap	DRC	NFSAG0	21–574, Vol 2
538	781 Automated Forms Discrepancies Print	AFP	NFSS90	21–572, Vol 2
824	Equipment Record Print For MD, TMSM, WUC, PSN, and C-E	ERP	NFSW30	21–564, Vol 2

Download the reports and direct the inquiry output to data files. Once someone from the investigation team contacts you, determine if printed copies are desired, a compact disk with the files written to it or both. Also ask if additional products are required that are not on the previously mentioned lists.

Step 5: Allow users back into the database

Finally, IMDS is ready to return to normal processing. Execute the straight line input “TISX1” to take the system out of FUD.

214. Aircraft mishap G081 procedures

In the unlikely event there is a mishap, the G081 DBM, like the IMDS DBM, must immediately secure the aircraft records. The main difference between G081 and IMDS is how the records are secured. G081 locks the aircraft records and IMDS secures the entire database. In G081, perform the following four major steps:

1. Get specific aircraft details.
2. Lock the aircraft records.
3. Run history reports.
4. Unlock the aircraft records.

Step 1: Get specifics

When you are notified of an incident, request the following information:

- MDS.
- Aircraft serial number/engine number(s).
- Date and time.
- Name, number and title of the person notifying you.

As with IMDS, the exact nature of the mishap and what was affected may dictate the specific items you require.

Step 2: Lock the aircraft records

Use screen 9012 to lock the aircraft records. Process a “LOCK” action for the aircraft serial number.

Step 3: Process history reports

At a minimum, you’ll want to run a 67–117 Maintenance Documentation History report and the following screens:

Screen	Title
8005	General Aircraft Data
8038	Flight Status Data Report

Screen	Title
8040	TCTO Report
9035	Serially Controlled Component History Report
9188	Dash 6 Report

Keep in mind, these are the minimum reports you should run. Speak with your squadron supervision, scheduling and quality assurance offices to include additional items your base may want to see.

Step 4: Unlock the aircraft records

The aircraft records remain locked during the course of an investigation. Do not unlock the records until the maintenance group commander gives the approval. If approved, use screen 9012 to unlock the aircraft records by processing an “UNLOCK” action for the aircraft serial number.

Self-Test Questions

213. Aircraft mishap Integrated Maintenance Data System procedures

1. In IMDS, what are the five major steps to accomplish if there is an aircraft mishap?

2. What do you immediately do when you are notified of a mishap?

3. How long do you request the database save tape expiration date be changed to?

214. Aircraft mishap G081 procedures

1. For a mishap, how do G081 and IMDS differ in the way their records are secured?

2. What are the four major steps to accomplish if there is an aircraft mishap?

3. What screen is used to LOCK and UNLOCK aircraft records?

4. Who gives the approval to UNLOCK aircraft records?

Answers to Self-Test Questions

201

1. Through online remote terminals connected to the SBLC system throughout the maintenance complex.
2. Updating the database, retrieving information from the database for local use, and reporting data required for higher headquarters.
3. Data is entered to update the database because of an activity that occurred in the maintenance environment. Retrieval of information from the database is dictated by the needs of the functional user.

202

1. A schedule.
2. To ensure uninterrupted flow of information to functional user (the customer).
3. By developing and implementing a checklist.
4. The host DBM.
5. Notify the system monitor to save the IMDS database before shutdown.

203

1. NDA500 and DMU Verify.
2. DBE.
3. QLP and IQU.
4. On or about the first working day of each month.
5. The database has been saved and units have processed all required MPR load, change and delete options.
6. When the man-hours previously input were incorrect or a special requirement exists where AF Reserve or Air National Guard units require more hours than are automatically assigned for the current month.

204

1. NFS760, TRIC SHM, JDD delete history.
2. Every Wednesday.
3. No more than 92 days.
4. NFS7U0, TRIC TBE.
5. Check, transfer and “JST” card.
6. Annually.
7. Screen 126.

205

1. Have users make and keep blank copies of their commonly used IMDS screens, create input text files, and create manual forms.
2. When the system is unavailable for only a short period of time and only when a relatively small amount of transactions require processing.
3. When the system is unavailable for a long period of time, or if you need to process a large amount of transactions.
4. The alternate site can be another nearby military base, an auxiliary field, an off-station tenant, or government organization.
5. DDN.
6. For the continuous reporting of REMIS-reportable JDD.

206

1. A timely communication used by AFLCMC to provide DECC and functional users with the necessary interim guidance and procedures required for continued system operation or to inform the users that continued system operation is not possible until needed corrections are received.

2. Determines who is affected and what action a user must take.
3. To provide functional users with technical solutions, interim guidance, and procedures for continued system operation and heads-up information
4. As a tool to communicate with field agencies.
5. To identify changes, corrections, and updates for IMDS to each DECC and all functional users worldwide.

207

1. When IMDS does not operate as designed.
2. The IMDS functional manager.
3. Initial review, evaluation, and validation.
4. Submit an ITSRD through the host DBM section.

208

1. Message routing, recovery, and utility functions.
2. Host DBM.
3. Screen #891.

209

1. IMDS data files that go to REMIS are known as KFS files; REMIS data files that go to IMDS are known as KRE files.
2. (1) f.
   (2) e.
   (3) c.
   (4) n.
   (5) l.
   (6) k.
   (7) a.
   (8) b.
   (9) m.
   (10) j.
   (11) i.
   (12) g.
   (13) h.
   (14) d.

210

1. System failure is when the mainframe computer goes down and communication to the computer is lost. Program failure is when a program does not successfully reach completion and error messages are displayed.
2. It is restarted automatically once the database is restored by the DMR. No input action is required by the user.
3. This condition occurs when a program can no longer process.
4. By a TRIC inquiry or background report output that displays the wrong information.
5. TIP users (other than the DBM).
6. In manual sets.
7. Simply erases the effects of input; if an input transaction made any changes to the database before it rejected with an abnormal disk, those changes would be rolled back. The DMR would return the database to its previous condition prior to the offending input.
8. AFCSM 21–571, Volume 2, Attachment 2.

9. Essentially, it is a deadlock between run units; in other words, two inputs tried to access the same data at the same time and deadlocked between each other. The user need only retransmit the input to correct this condition.
10. The DMR error code.
11. Your supervisor, your host DBM, your system monitor, or the FAS at AFMC AFLCMC/HICS.

211

1. Thread control, locking and queuing, system table control (like file description tables), database I/O, cache, storage management, and data recovery.
2. File usage information.
3. DMR status of individual run.

212

1. (1) Identifies and/or corrects C-E status and inventory problems in IMDS.
   (2) Reconciles the IMDS and REMID databases.
   (3) Deletes and clears all REMIS errors on the IMDS database.
2. REMIS reportable equipment.
3. When all errors are corrected.

213

1. (1) Secure the database.
   (2) Run a database save.
   (3) Get specific aircraft details.
   (4) Run reports and inquiries.
   (5) Allow users back into the database.
2. Place IMDS in FUD mode and enter the UserIDs of personnel allowed to process transactions while the system is in FUD.
3. One year.

214

1. G081 locks the aircraft records and IMDS secures the entire database.
2. (1) Get specific aircraft details.
   (2) Lock the aircraft records.
   (3) Run history reports.
   (4) Unlock the aircraft records.
3. 9012.
4. The maintenance group commander.

Complete the unit review exercises before going to the next unit.

Unit Review Exercises

Note to Student: Consider all choices carefully, select the *best* answer to each question, and *circle* the corresponding letter. When you have completed all unit review exercises, transfer your answers to the Field Scoring Answer Sheet.

Do not return your answer sheet to Air Force Career Development Academy (AFCDA).

1. (201) Which subsystem process does the Integrated Maintenance Data System (IMDS) automate?
   a. Aircraft status.
   b. Aircraft location.
   c. Aircrew debriefing.
   d. Personnel inprocessing.

2. (201) What system *best* describes the operating concept of Integrated Maintenance Data System (IMDS)?
   a. User-oriented.
   b. Event-oriented.
   c. Database-centered.
   d. Information-centered.

3. (202) The database manager (DBM) *must* inform users when a batch program is scheduled to be run
   a. for a long time.
   b. in the middle of the day.
   c. by the Defense Enterprise Computing Center.
   d. requiring the Integrated Maintenance Data System (IMDS) online processing to be down.

4. (202) When monitoring computer maintenance as a database manager (DBM), your efforts *must* be directed toward
   a. requesting hardware and software.
   b. updating the capabilities on an as needed basis.
   c. the analysis of unscheduled and scheduled downtime.
   d. technical maintenance of Integrated Maintenance Data System (IMDS) database equipment.

5. (202) The Integrated Maintenance Data System (IMDS) recovery procedure is based on length of downtime and
   a. number of remote terminals.
   b. volume of transactions.
   c. number of users.
   d. time of day.

6. (203) The Integrated Maintenance Data System (IMDS) work-hour update (WHC) program is used to
   a. update monthly man-hour summary (WAH) when a special requirement exists where Air National Guard units require more hours.
   b. update man-hours for an active duty unit, or work center that are different from what is automatically assigned.
   c. adjust man-hours for reserve personnel established by monthly WAH.
   d. adjust incorrect man-hours prior to executing monthly WAH.

7. (204) Which program is used to verify program network file system (NFS)760 runs each week?
   a. NFS760.
   b. NFS910.
   c. NFSO40.
   d. NFSRU0.

8. (204) How many options are there for program network file system (NFS)7U0?
   a. 2.
   b. 3.
   c. 4.
   d. 5.

9. (204) What does the database manager (DBM) create for supply to use during an annual standard reporting designator (SRD) reconciliation?
   a. Printout.
   b. Disk file.
   c. Error report.
   d. Spare parts list.

10. (205) Nonavailability of the mainframe can be caused by
    a. pseudo remote processing.
    b. removal of classified data.
    c. a user's computer that won't boot up.
    d. inputting of job data documentation into the Integrated Maintenance Data System.

11. (205) Which is an option for off-line processing?
    a. Creating input text files.
    b. Running a database save.
    c. Updating the audit trail tape.
    d. Memorizing every transaction made.

12. (206) Whenever an advisory notice is sent out to the field, who determines the affected users at base level?
    a. Air Force Life Cycle Management Center (AFLCMC).
    b. Defense enterprise computing center (DECC).
    c. Combat Systems Support Help Desk.
    d. Host database manager (DBM).

13. (206) What document is used by the Field Assistance Service (FAS) to communicate with field agencies?
    a. System advisory notice (SAN).
    b. Heads-up messages (HUM).
    c. Printed newsletter.
    d. System release.

14. (206) What does the Air Force Life Cycle Management Center (AFLCMC) send out to the field right away if a software problem is discovered but a system advisory notice (SAN) might be delayed?
    a. Wireless email.
    b. Instant text messaging.
    c. Heads up message (HUM).
    d. Facsimile message (FAX).

15. (207) A difficulty report (DIREP) is submitted when the
    a. Integrated Maintenance Data System (IMDS) functional manager cannot provide a workaround for the problem.
    b. field assistance branch receives the problem from the database manager (DBM).
    c. Defense Enterprise Computing Center (DECC) approves the submission.
    d. DBM calls the Field Assistance Service (FAS) to submit one.

16. (207) An Integrated Maintenance Data System (IMDS) user can request, through the host database manager (DBM), to add or change the capabilities of IMDS by submitting a
    a. Reliability and Maintainability Information System (REMIS) modification request.
    b. Information Technology Systems Requirements Document (ITSRD).
    c. Defense Enterprise Computing Center (DECC) update request.
    d. IMDS trouble ticket.

17. (208) The Interactive Communication Interface (ICI) software package provides Integrated Maintenance Data System (IMDS) and supply users with
    a. utility, inquiry, and recovery functions.
    b. recovery, setup, and validation capabilities.
    c. message routing, recovery, and utility functions.
    d. message routing, inquiry, and data validation functions.

18. (208) The responsibility for initializing Interactive Communication Interface (ICI) via screen #891 for the Integrated Maintenance Data System (IMDS) and Supply System interface lies with
    a. the host data base manager (DBM).
    b. the system monitor.
    c. unit level DBMs.
    d. functional users.

19. (209) Integrated Maintenance Data System (IMDS) data files transmitted to Reliability and Maintainability Information System (REMIS) are known as
    a. KRE files.
    b. KFS files.
    c. DDN files.
    d. MOA files.

20. (209) Which Reliability and Maintainability Information System (REMIS) interface program provides a list of all files which have been received from or transmitted to REMIS within the last 72 hours?
    a. Debriefing Memorandum of Agreement (MOA) Builder.
    b. Defense Data Network (DDN) Address Record.
    c. Table Validation Problem Reporting.
    d. REMIS Error Inquiry.

21. (210) What type of Integrated Maintenance Data System (IMDS) database errors occur when the data does *not* follow the schema rules?
    a. Data.
    b. Pointer.
    c. Structural.
    d. Logical Set.

22. (211) The Universal Data System monitor (UDSMON) provides for an integrated software environment for control and maintenance of
    a. the data management utility (DMU) processor.
    b. executive system software.
    c. Air Force data systems.
    d. user databases.

23. (211) What option of the Universal Data System monitor (UDSMON) provides the current record, database key (DBK), and command being performed?
    a. Data management routine (DMR) file usage.
    b. DMR status of individual run.
    c. File usage information.
    d. Status of all runs.

24. (212) When is a communications-electronics (C-E) equipment inventory information sent from the Integrated Maintenance Data System (IMDS) to the Reliability and Maintainability Information System (REMIS)?
    a. After you run the auto-fix option in the first run.
    b. After you correct all the errors not fixed by the program.
    c. The first time you run the IMDS/REMIS reconciliation program.
    d. When requested by the C-E major command (MAJCOM) representative.

25. (213) Initially, what is the most important Integrated Maintenance Data System (IMDS) step when notified of an aircraft mishap?
    a. Run a database save.
    b. Secure the database.
    c. Run reports and inquiries.
    d. Get specific aircraft details.

26. (214) What must you immediately do in G081 if there is an aircraft mishap?
    a. Run history reports.
    b. Secure the database.
    c. Run a database save.
    d. Secure the aircraft records.

Please read the unit menu for unit 2 and continue ➔

Student Notes

Unit 2. Integrated Maintenance Data System Central Database

INTEGRATED MAINTENANCE DATA SYSTEM is a large dynamic, on-line system used at base level to manage Air Force resources—equipment, supplies, and personnel. IMDS is the standard Air Force system for maintenance production support and the collection and processing of equipment maintenance information. All information supporting maintenance functions must be accessible to IMDS for collection, storage, and dissemination of critical data for repair and improvement of our weapon systems and equipment. Eventually, IMDS will function as a single logical database to accommodate historical and legacy data currently stored in other databases. The IMDS design is flexible to support changes in logistics infrastructure size, quantity, and mission orientation whether at home base or deployed. Understanding the process of how data is collected and retrieved is important for the maintenance analyst who performs the job of a database manager.

This unit aims to increase your level of understanding of the database, its structure, and operation. It is equally important, too, to protect the database since the data therein is essential logistics information that is vital to the Air Force mission.

2–1. Centralized Database Operation

IMDS utilizes a powerful central database system to reduce redundancy of data and speed up information processing. IMDS consists of computer database systems that provide automated inventory control and management information for base-level maintenance managers allowing for efficient utilization of their resources in mission performance. In this section, we will discuss the two basic database systems that comprise the central database operation; in addition, we will cover the concept of operation, or an overview of how a system operates.

215. Network versus relational database

A database is a collection of interrelated data. There are two different types of database systems that IMDS employs: the network and the relational databases. The network database is classified as a non-relational database. IMDS uses the network DMS 2200. The RDMS 2200 is also included with the DMS to compose what is called the CDB. Let's look at the basic difference between these two database systems.

Network database

The network database is a complex web of data that are tied together in an organized fashion. Larger databases use this because it can accommodate different types of data based on the users' need.

Basic structure

The basic structures of a network database include records and sets. The records are composed of fields (data items) and are arranged into sets. Each set has an owner type record and a member type record. The primary difference of the network model is that it allows a member record to have no owner record, one owner record, or many owner records. Using the example of units and organizations is that you may have several units that are supported by several of the same organizations.

The network model supports the many-to-many relationship. It also allows the user to access records at many different levels. An advantage of this model is that it offers an internal file structure known as a loop. This structure allows records of the same type to be strung together, which improves processing time.

Physical pointers are used to establish relationships between records. Because the network model is composed of loops and physical pointer chains, these data elements must be maintained as the database is updated. Although these improve performance when retrieving data, there is a significant cost when updating. Physical pointers can be cumbersome to reorganize. Such a model requires a high level of technical skill to implement and maintain a network database. As a result, the evolution of this type of database is slow and is only suitable if the structure of the system is relatively stable. Also, as with all non-relational databases, the network model still uses record-at-a-time processing; this can be slow and tedious.

Retrieval process

The DMS database has used a number of programming languages, including common business oriented language (COBOL), QLP, and IQU, to name a few. Retrieval programs follow a syntax structure to pull information from the database. Rigid procedures or rules to extract data make it difficult to the user to develop.

Relational database

What makes a relational database different from non-relational ones is that the models on which they are based were developed after the physical structure had been defined. For this reason, network database structures can be converted into relational structures. This process is known as normalization and is the process of eliminating duplicate or redundant data. Relational databases use a single data language for all data operations.

Basic structure

A relational database is a set of tables from which data can be accessed and analyzed in many different ways without needing to reorganize tables or cut and paste data. Unique identifiers for each row, called primary keys, are what allow relational databases to work. In a customer database, for example, a column containing unique customer ID numbers would keep customers—including the multiple John Smiths—distinct. These unique customer IDs, rather than all the contact data, would be used by other databases to keep track of information related to each customer.

Relational databases consist of one or more two-dimensional tables of data values that form relations (more often referred to as tables). The tables are made up of rows and columns. Each row represents one occurrence of the entity represented by the table. The occurrence is called a record. A column contains what is called an attribute, which we describe as one characteristic of an entity or relation. An entity in real life becomes a table in a relational database.

Retrieval process

Structured Query Language (SQL) is the programming language used for building, retrieving and updating relational databases. SQL enables queries from databases and gathers data for custom reports. The most distinguishing characteristic of a relational query language is its non-procedural language.

Relational databases are more effective than non-relational databases for various reasons: the use of two-dimensional structures, use of simple and powerful standard query language, and the use of logical pointers.

Two-dimensional structures make it much easier to represent data than with network models. The format of existing columns may be changed in tables or new columns added with relative ease. The use of a simple and powerful query language makes it easy to create, add, delete, or update tables with a few lines of commands. The use of logical pointers makes the relational database more data independent (do not have to restructure data around a hierarchy). This also makes for ease of design and maintenance.

We cover more about the structure and functions of the two database systems in the next section.

216. Concept of operation

In this lesson, we will discuss the concept of operation, which is an overview of how the system operates, including its different functions. The concept of operation for the CDB is a move towards integration and efficient use of data and internetworking of all Air Force information systems.

Software configuration

IMDS is a functional system program (FSP) that is composed of a FSA and background batch programs. The computer hardware and the programming standards dictate the relationship between the FSA and its background batch programs. For details on the FSA and background batch programs, the reference is AFCSM 21–556 Volume 1, *Integrated Maintenance Data System (IMDS) Central Database),* Attachment 1.

The data system designator (DSD) for IMDS is G054, and the System Code is FS, in which “FS” is the qualifier filename that refers to the IMDS 7R1 schema, or simply the schema name. Consequently, the software version for the IMDS Unisys CDB is 7R1. The IMDS Unisys Central Database programming language is Cobol 85. The IMDS Unisys CDB application uses the Web Transaction Server (WebTS) GUI application software as the presentation layer.

Single physical location

The IMDS Unisys CDB operates on one Unisys computer at the DECC Oklahoma City (OKC). An alternate site, located at DECC Ogden, acts as a backup and mirrors the same database and operational capacity in case of a contingency. There will be at least 130 separate physical databases,

each with the unique database for each base, on one Unisys platform. Each base will be a separate ELC. The system will contain one schema (or database blue print) for all of the ELCs, as shown in figure 2–1. There is a roll up area that will emulate an ELC in terms of its structure.

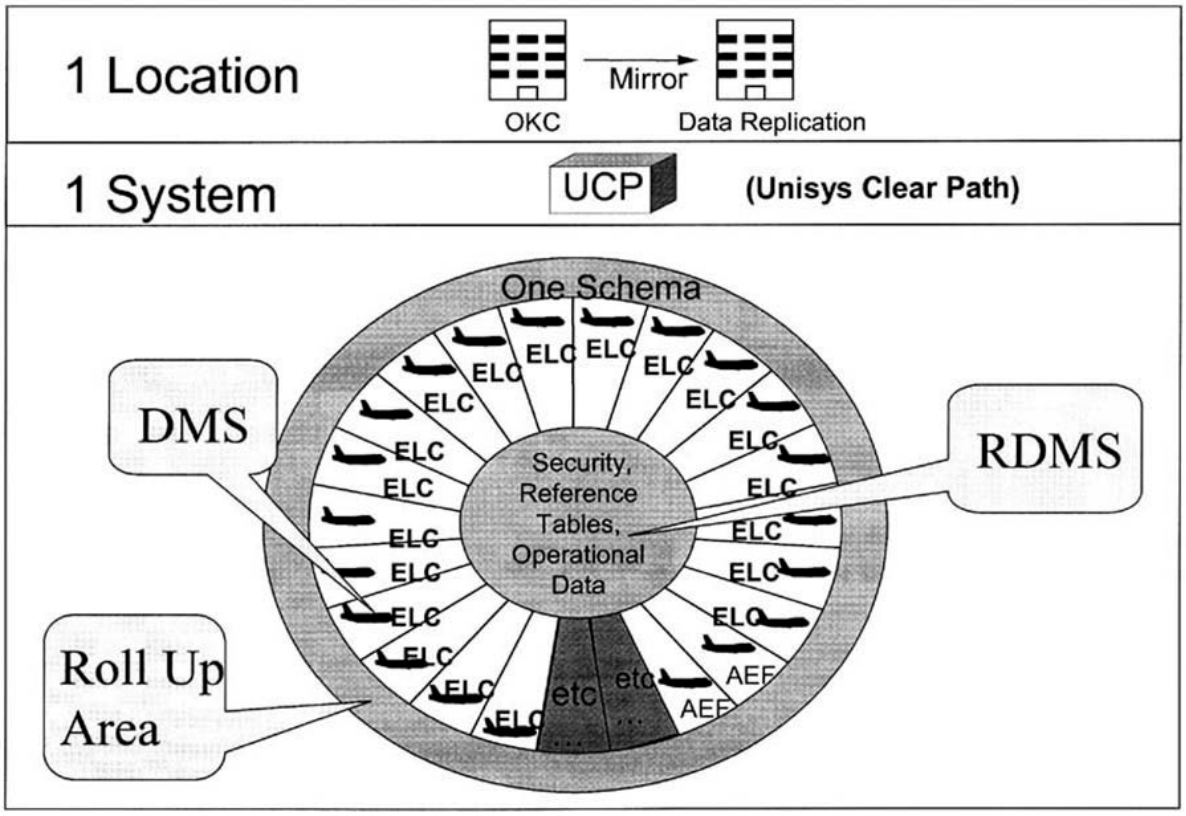


Figure 2–1. Integrated maintenance data system central database.

Major database sections

The three primary sections within this central database include the DMS, roll-up, and RDMS, and are discussed in the following paragraphs.

Data management system

The DMS section at the base level contains data from integration and is where your ELC daily work is accomplished and ELC data stored. Transactional type data from your ELC remains in the Unisys DMS database.

Roll up area

The roll up area stores and uses aggregate data for reporting that supports total asset visibility (TAV), AF Portal, and Air & Space Expeditionary Force (AEF) requirements. This section contains reject code and narrative records and general-purpose records. Capabilities include the ability to report on maintenance performance (e.g., scheduling effectiveness, abort rates, repeat/recur rates, etc.), configuration, and asset type for enterprise wide visibility. The roll up area will essentially look and feel like an ELC. The roll-up area follows the DMS in terms of its database structure.

Relational data management system

This section is the RDMS section. The RDMS includes the security portion for the single sign-on across the enterprise. Reference data, security data, validation tables, and JDD history are located here. Reference tables provided by REMIS include equipment and aircraft data. These reference tables include a D043 table; work unit codes (WUC), how malfunctioned (HOWMAL) code tables,

action taken (AT) codes, and other reference data tables. In addition, there is also AEF information and portal data in the RDMS.

The key point in moving to the RDMS is that it substantiates the migration to the Global Combat Support System - Air Force (GCSS-AF) framework. It provides a transition midpoint to support the GCSS-AF migration strategy to a one network Air Force.

Advantages of central data base

There are clear advantages of the centralized structure. Having all data reside on one machine and a shared common reference data will provide the users several key advantages. Just a few of these advantages include the following:

- Reduces data duplication—this will save storage space and ensures everyone is using the same set of reference data.
- Provides capability for enterprise view and increased information sharing—with all data sources residing on one platform, the capability will exist to create programs that can walk across the enterprise to gather data and provide senior leadership a look at all relevant data. It will also allow users access to like equipment data for comparison on other systems.
- Equipment/personnel transfer—will facilitate in automating equipment and personnel transfers between databases. Significantly reduces amount of manual input performed at losing and gaining bases.
- AEF reporting—will allow equipment and personnel to be linked to an AEF, so that AEF specific reporting can be accomplished without having to pull from multiple sources and manually calculate data.
- Improved security—will allow users to be assigned profiles based on the roles that they perform. This will restrict users to only being able to process those screens that relate to the work they do.
- Improve access to historical data—All ELCs will have access to historical data stored in RDMS. This will allow users to look at historical data for any aircraft regardless of when it was gained.

Self-Test Questions

215. Network versus relational database

1. What is a database?

2. What are the basic structures of a network database?

3. What are used to establish relationships between records in a network database?

4. What kind of database is made of two-dimensional tables of data values?

5. What is the programming language used for building, retrieving, and updating a relational database?

216. Concept of operation

1. How many schemas will the IMDS central database have?

2. What database structure does the roll-up area follow?

3. Where are the reference data, security data, validation tables, and JDD history located?

2–2. Data Management System

In the previous section, we learned about the two basic database models that comprise the IMDS central database. We also discussed the concept of operation that describes in general how those two databases operate. This section of the 7R1 database will contain all the records stored in your ELC. We will explore the actual structure of the DMS database and discuss how it is put together.

217. Database structure

The physical structure of the DMS database depicts the actual characteristics of the database and how it is constructed. The schema and subschema define the database, and are not the database itself. The actual data is stored elsewhere on disk.

Schema

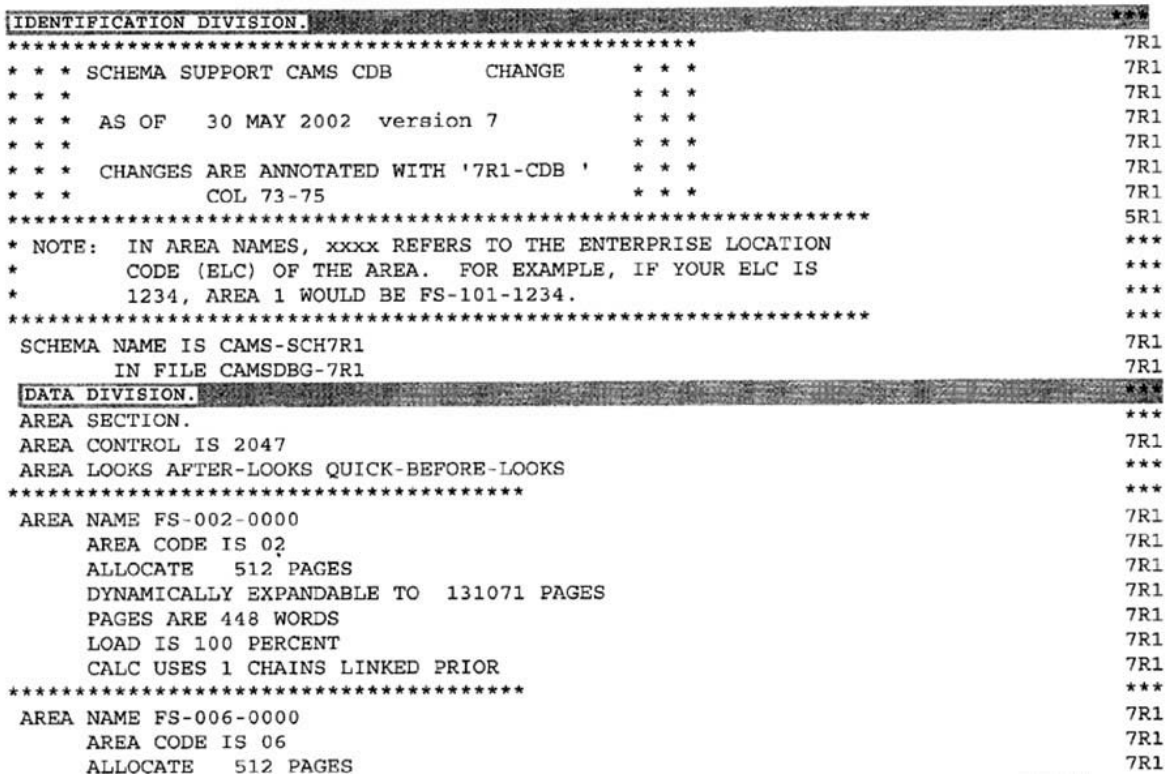
The schema is not part of the database files, but a separate entity. It describes the exact physical layout of every type of record that can be stored in the database. It also describes the logical relationships that exist between each type of record. When a program requests access to a certain record, a utility program called the DMR uses the schema like a road map to find the requested record. The two divisions of the schema are identification and data (fig. 2–2).

Identification division

The syntax of the schema begins with an identification division. This division supplies the schema name and the TIP file number.

Data division

The data division is next in the schema syntax. It actually describes the data within the database. There are four sections in the data division—data name, areas, record, and set.



```
IDENTIFICATION DIVISION.                                                ***
*************************************************                       7R1
* * * SCHEMA SUPPORT CAMS CDB       CHANGE     * * *                     7R1
* * *                                         * * *                     7R1
* * *  AS OF   30 MAY 2002  version 7          * * *                     7R1
* * *                                         * * *                     7R1
* * *  CHANGES ARE ANNOTATED WITH '7R1-CDB '   * * *                     7R1
* * *          COL 73-75                      * * *                     7R1
*****************************************************************       5R1
* NOTE:  IN AREA NAMES, xxxx REFERS TO THE ENTERPRISE LOCATION           ***
*        CODE (ELC) OF THE AREA.  FOR EXAMPLE, IF YOUR ELC IS            ***
*        1234, AREA 1 WOULD BE FS-101-1234.                              ***
*****************************************************************       ***
 SCHEMA NAME IS CAMS-SCH7R1                                             7R1
       IN FILE CAMSDBG-7R1                                              7R1
DATA DIVISION.                                                          ***
 AREA SECTION.                                                          ***
 AREA CONTROL IS 2047                                                   7R1
 AREA LOOKS AFTER-LOOKS QUICK-BEFORE-LOOKS                              ***
*****************************************                               ***
 AREA NAME FS-002-0000                                                  7R1
      AREA CODE IS 02                                                   7R1
      ALLOCATE   512 PAGES                                              7R1
      DYNAMICALLY EXPANDABLE TO  131071 PAGES                           7R1
      PAGES ARE 448 WORDS                                               7R1
      LOAD IS 100 PERCENT                                               7R1
      CALC USES 1 CHAINS LINKED PRIOR                                   7R1
*****************************************                               ***
 AREA NAME FS-006-0000                                                  7R1
      AREA CODE IS 06                                                   7R1
      ALLOCATE   512 PAGES                                              7R1
```

Figure 2–2. Schema divisions.

Data name

The data name section is used to define any data items that will be used throughout the schema. Data names are like variables and must be initialized before they are used.

Area

The area section contains information about each area in the database. It contains things such as the area name, number of pages in that area, size of the pages, and type of the pages.

Record

The record section describes the records that will be placed in those areas. The syntax describes the name by which the record will be referenced, an internal code used by the DMR, and the method used to store/retrieve the record in/from the database.

Set section

Finally, the record relationships are specified via the syntax contained in the set section. This section is used by the DMR to locate records or chain them into various set occurrences. An application program to locate a record via the linkage may use the set name.

Subschema

One or more subschemas may exist within a single schema. The subschema describes how different users view the database. Depending on which subschema is being accessed, the user will have access to a different subset of the records in the database. All users of different subschemas are able to

access common records. Users may also have other records that only they are authorized to access. In this way, data security and continuity is maintained. Our IMDS schema is, in reality, a subschema in that it limits your view of the database to only your areas, pages, and sets. This subschema file is stored on disk along with the actual database.

The syntax of the subschema is similar to the syntax of the schema. Again, there are two divisions: identification and data. These two divisions contain the same types of information as their corresponding divisions in the schema. As a general rule, the sections in the data division define what portions of the schema this subschema will use. For instance, if the schema defines eight records in the record section, the IMDS subschema record section might only define three of those original eight records. Not all sections are the same. The QLP section defines the paths that the system will use to access the data.

Data definition language

You now understand that the physical and logical structures of a database are not defined in the programs that use them, but they are actually defined by the schema. Since multiple users will access the same database concurrently. There is also a need to control access to the different types of records so that each record can be accessed only by those users with a need to know. This control is established by the data definition language (DDL). The schema source statements are written in DDL and processed by the DDL processor in order to define the database to the computer. DDL is a high-level computer language used to design and create schemas, just as a programmer might use COBOL to write a software program. However, DDL is a schema language, not a programming language, and is independent of any programming language.

218. Data Management System schema

Schema source statements are written in DDL and processed by the DDL processor in order to define the database to the computer. This DDL source code is divided into several entries—such as areas, records, and sets—that name and describe the schema. These entries describe the exact physical layout of the database. Think of the schema as a large map that guides the DBM through system processing. This lesson covers schema sections and the DDL clauses that describe each element of the schema.

Schema sections

The schema of the database is divided into three sections: area, record, and set sections. These three sections are described in the IMDS schema printout. In the text printout, the codes for each section (area, record, set) are arranged numerically from the lowest number to the highest, although not necessarily in successive sequence.

On the schema printout, shown in figure 2–2, notice that the column on the right indicates what modifications to the original schema have been added since it first came out (i.e. “7R1”, “6R1”, “5R1”, “Y2K-STD”, etc.).

NOTE: The online 7R1 schema for IMDS is known as “7R1 DMS Schema Book,” a subsection of the schema layout of “The IMDS Database Help System.” This database help system is found on the A4QJ Maintenance Analysis SharePoint site under Air Combat Command (ACC) A4 Directorate of Logistics.

Area

The area section appears first in the schema and defines each area used by the database, to include clauses such as the area name and code, after looks (types of looks for DMU recovery, audit trail related), and sizing information (number of pages allocated, and the size of each page). Figure 2–3 shows how the area section is displayed on the 7R1 schema.

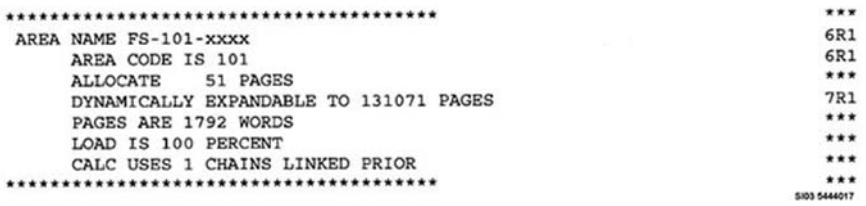


Figure 2–3. Sample area from area section.

The key elements in this section are shown in the following table:

Clause	Meaning
Name	The name of the area is FS–101-XXXX, where XXXX is the ELC code for the specific ELC database.
Area code	Code 101 corresponds to the area name
Allocate	Allocation of 51 pages—dynamically expandable to 131071 pages.
Pages	These data items reflect the data size for this area, describe the current size, and indicate its expansion ability. This value is always in multiples of 28 words.
Load	From 100 percent, indicates how much space is left over. In this case, the area will always be full.
CALC	Show one chain linked prior; the maximum value in this schema is 4. This indicates how many CALC chains can be on each prime page of that area. CALC chain allows entry point access.

An area is a database file consisting of a group of records, and is also the highest subdivision of the database. Areas are different sizes, depending on what data is contained therein. The body represents the various areas within the IMDS schema. A storage area is associated with a physical file, either Exec or TIP. Records of one or several record types can be stored in a single area. This physical file determines which areas are parts of the physical database design.

Base level area

Areas are identified by area names within a specific schema. In 7R1, each ELC database has 11 areas, known as the "***base level areas.***" Each of the areas has its own unique area name. The areas are named FS–101-XXXX where the XXXX represents the ELC. *For example:* FS–101–8031 is area 1 on ELC 8031. The following table shows the 7R1 ELC areas.

Area Number	Area Name
1	FS–101-XXXX
2	FS–102-XXXX
3	FS–103-XXXX
4	FS–104-XXXX
5	FS–105-XXXX
8	FS–108-XXXX
9	FS–109-XXXX
10	FS–110-XXXX
12	FS–112-XXXX

Area Number	Area Name
13	FS–113-XXXX
14	FS–114-XXXX

Roll-up area

The roll-up area contains records that supply common information to the ELCs. These areas are also known as "***enterprise-wide or global roll-up areas.***"

There are two roll-up areas and both of their ELC codes are "0000," as in the following example:

FS–002–0000
FS–006–0000

Record

The record section appears second and defines every record in the database. A record is a collection of related data items. An example of a record extracted from the 7R1 schema is shown in figure 2–4. A text narrative describes the full name of the record and its primary purpose. The clauses that define the elements in this section are record names, record code (numbers), location modes CALC, direct, Via Set), record modes (type of characters, IMDS record mode is American Standard Code for Information Interchange [ASCII]), the area where the record is stored, and the picture clauses required to define all data items in every record. Just like each individual record is identified by a DBK, each type of record also has its own name. Record names can be up to 30 alphanumeric characters long, 26 possible alpha character record names, followed by a dash and three numeric record IDs.

A detailed description of all the records can be found in AFCSM 21–571, Volume 2, *Integrated Maintenance Data System (IMDS) Central Database (CDB), Database Management.* If you have reviewed that manual, you know that the alpha name and numeric record ID are both unique entities. A record is referred to by its alpha name or the numeric ID. The record section displays the following data for each record (fig. 2–4).

Clause	Explanation
Record name	ISP–168 is the name of this particular record. In this case, this is the inspection (ISP) record.
Record code	Numeric value assigned to the record name. Record code for this example record is 168.
Location mode	Via EQPP-TO-HISTORY SET. This is the access method you'll use to see the record within an area. The additional information within indicates that this record is available through the areas indicated.
Within	Identifies the area where the record is placed. The ISP–168 record is found in area 5 (FS–105-XXXX).
Reserve	Indicates the number of pointer locations for records that can participate in manual sets. In this case, the record ISP–168 can have up to 5 pointer locations. **NOTE:** the DDL automatically allocates the pointers required for automatic set participation.
Record mode	All IMDS records use ASCII to define the types of characters used.
Data identifier	The remaining parts of this record section are called the data identifiers (e.g. ISP-JOB-STD-NR). Data identifiers define areas where the records can be found plus the picture (PIC) clauses to define all data items in the record.

Clause	Explanation
Level number	Identifies a data item or unused portion of a record with a database identifier and establishes its position relative to the hierarchical data item structure within a record. The numbers 05, 10, 15, and successive increments of 5 indicate the hierarchy of the data identifier in the IMDS schema. The 05 is the first level. Where there is a breakdown of a data identifier into specific data items, the sub-data items take the next level. Ex. 05 ISP-WUC-DATA 10 ISP-WUC-LCN ISP-WUC 15 ISP-WUC-FILLER

```
*****************************************                                    ***
* Inspection Record (ISP-168)                                               ***
*    This record contains maintenance inspection information about a particular ***
*    equipment.                                                             ***
*****************************************                                    ***
 RECORD NAME IS ISP-168                                                     ***
     RECORD CODE IS 168                                                     ***
     LOCATION MODE IS VIA EQPP-TO-HISTORY SET                               ***
     WITHIN FS-105-xxxx                                                     ***
     RESERVE 06 POINTERS                                                    ***
     RECORD MODE IS ASCII                                                   ***
     05  ISP-JOB-STD-NR                    PIC 9(05).      1-5    = 5     ***
     05  ISP-TYPE-INTERVAL                 PIC X(01).      6-6    = 1     ***
     05  ISP-FREQUENCY                     PIC 9(06).      7-12   = 6     ***
     05  ISP-DUE-DATE-UJ7                  PIC 9(07).     13-19   = 7     Y2K-STD
*    05  ISP-DUE-DATE-UJ7-X                                               SUBSCHEMA
*        10  ISP-CC                        PIC 9(02).                     SUBSCHEMA
*        10  ISP-YY                        PIC 9(02).                     SUBSCHEMA
*        10  ISP-U-DDD                     PIC 9(03).                     SUBSCHEMA
     05  ISP-MULTI-IND                     PIC 9(01).     20-20   = 1     ***
     05  ISP-ESR-REPT                      PIC 9(01).     21-21   = 1     ***
     05  ISP-ACT-IND                       PIC 9(01).     22-22   = 1     ***
     05  ISP-POS                           PIC X(04).     23-26   = 4     ***
     05  ISP-TYPE-INSP                     PIC X(01).     27-27   = 1     ***
     05  ISP-NBR                           PIC 9(02).     28-29   = 2     ***
     05  ISP-STAT-SYM                      PIC X(01).     30-30   = 1     ***
     05  ISP-WUC-DATA                                                     6R2-GUI
         10  ISP-WUC-LCN                                                  6R2-GUI
             15  ISP-WUC                   PIC X(05).     31-35   = 5     6R2-GUI
             15  ISP-WUC-FILLER            PIC X(10).     36-45   = 10    6R2-GUI
         10  ISP-WUC-LCN-FILLER            PIC X(17).     46-62   = 17    6R2-GUI
     05  ISP-OP-HRS                        PIC 9(08)V9.   63-71   = 9     5R3
     05  ISP-CYC-ST                        PIC 9(09).     72-80   = 9     5R3
     05  ISP-TYPE-RCD                      PIC X(01).     81-81   = 1     ***
     05  ISP-CATALOG-NBR                   PIC 9(02).     82-83   = 2     ***
     05  ISP-ENGINE-IDENT                  PIC X(02).     84-85   = 2     ***
     05  ISP-LST-PERFORMED                 PIC 9(09).     86-94   = 9     5R3
     05  ISP-DT                                                           ***
         10  ISP-CHANGED-DATE-ZJ5          PIC 9(05).     95-99   = 5     Y2K-STD
         10  ISP-TYPE-TIME-IND             PIC X(01).    100-100  = 1     Y2K-STD
         10  ISP-CHANGED-TIME-Z4           PIC 9(04).    101-104  = 4     Y2K-STD
         10  ISP-CHANGED-BY-PROG-ID        PIC X(02).    105-106  = 2     Y2K-STD
     05  ISP-SPARE                         PIC X(06).    107-112  = 6     6R2-GUI
*****************************************                                    ***
*END ISP-168 - total characters (9 bit) = 112    28 words                   ***
*                                                                           ***
```

Figure 2–4. Sample record from record section.

The set section appears last and defines every set in the database. It defines the logical relationship that exists between each type of record and defines every set user in the database. Sets are not part of the physical structure—only logical relationships. The clauses that define the elements in this section are set names, set codes, linkage modes, set placement order, set owner, and member(s) records, set

participation of each member, and set selection. We are again pulling a sample set from the set section (SET 370) as shown in figure 2–5.

Figure 2–5. Set section.

Using our sample set in the following table, refer to figure 2–5 as we look at what elements the set section defines:

Clause	Explanation
Name	Set name is EVT-TO-RECORDS.
Code	Code 370 corresponds to the set name
Mode	Chain linked prior identifies the method of linkage order.
Order	Identifies record placement.
Owner	Identifies the owner record the set is attached to. In this case, the EVT–370 is the owner of this set.
Member	Lists all the member records attached to the owner record through this set (CITS–368, EVN–371, EVNCOS–222, EVTU–380, and DEBD–215).
Set selection	Identifies the database key of the last record for a set accessed by a run unit, regardless of location or record type.

219. Page and record structure

In the previous lesson, we talked about the schema, which we compared to a map. Remember the actual physical design of the DMS database is organized into three components: areas, pages, and records. The areas hold the pages; the pages hold the records. Conversely, records are within pages; pages are within areas. We could compare this to the postal mailing address where the house address reveals the exact location of a house.

You have already learned that areas are the highest subdivision of the physical structure of a database. You will now learn about the other two components of the physical database—pages and records.

Page structure

Areas are further subdivided into one or more physical blocks called pages. Pages are the basic unit of input/output between the area files and the internal buffers maintained in the database. These are the units that a program can gain exclusive use of when updating the database. A page can contain one or more records. Pages within the same area are the same size, as specified in the subschema. That is, the pages in area 1 are all the same size. However, they are not necessarily the same size as pages in another area and vice versa.

Page allocation should be large enough to hold the expected volume of data. In the event additional space is needed, the data management routine (DMR) provides the needed space on an as-required

basis by placing additional pages beyond the highest page currently allocated. Pages cannot be extracted in page format from the system. Each record on the page can be individually extracted or displayed, but the page itself cannot be seen. Figure 2–6 depicts a typical page.

There are three types of pages available in DMS–2200. You can have data pages, index pages, and pointer pages. These are defined using the MODE verb in DDL. Every page type has a 10-word page header on the top of the page containing identification, pointers, overflow pages, and information about page usage. Then, under the page header, there is vacant space for records. The number of records to be stored varies for each page. Data and pointer pages have slots at the bottom of their pages. Slots are pointers to records used by the DMR to locate records on a page. Number of slots also varies—one for each record. The CALC chain headers are used whenever a CALC chain is declared for CALC records. Only one word is used.

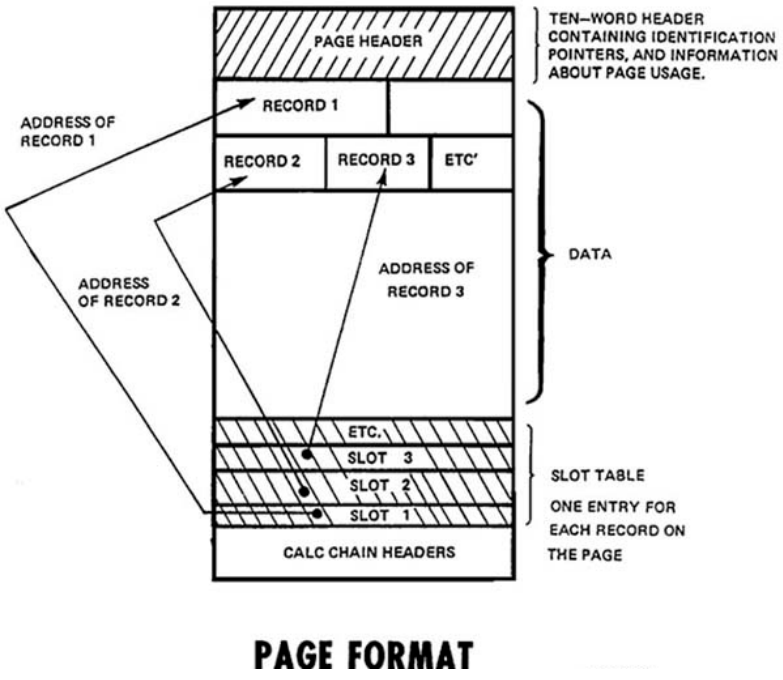


Figure 2–6. Page format.

Page usage

The DMR records all unused space on a page and reorganizes a page to accommodate a new record if at all possible. Ordinarily, new records are stored sequentially in the record area of a page. If the DMR finds insufficient space remaining in the record area to store a new record, it checks for a vacancy entry large enough for the new record. The DMR shifts records in the page and accumulates vacant entries to make space for the new record.

Record structure

Pages are further subdivided into records. Since a record is a collection of related data items, they are grouped together according to the data they contain. Records of the same type are identical in size and structure. Each record is identified by its DBK.

Record placement

Record placement within the database depends on association, distribution, and volatility. Association refers to records that are tied closely together or are related to one another. This relationship is based on record usage or record content. Distribution refers to where in any given area a particular record is

placed. Direct records are previously assigned to specific locations within an area. CALC records are assigned a location based on an algorithm. Volatility refers to usage based on volume (record count) or by the number of times accessed.

Types of information

Every record in the database is comprised of two types of information—control and data. A record's control information helps determine how the DMR manipulates the record. A record's data is the information supplied by the user, such as maintenance or supply.

Control

The control information is further broken down into a couple of parts as shown in figure 2–7. First, each record has a record header. The record header indicates the record type, length and a special status code. Second, each record has pointers. These pointers are used to link the records into the sets in which they participate. The number of pointers a record has depends on the number of sets with which a record is associated. The DMR automatically assigns values to these pointers. Finally, the record has system pointers, which are special-purpose pointers for DMR use.

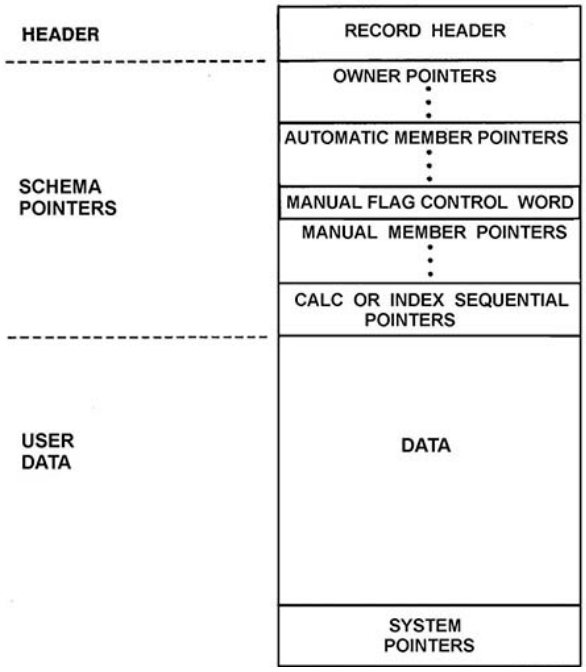


Figure 2–7. Record format.

Data

In addition to all this control information, the record has its user-supplied data. The size of user data depends on how the record is defined in the schema.

Database key

Each record in a database is uniquely identified by the area and page in which it resides, and its record number on the page. The area code, page number, and record number combined together form the DBK for a record. The DBK is in the form of a 13-digit character and is used to address records in a database directly. Any record can be addressed by its DBK, which is not physically located in a record, but can be compared to a relative location within a given area. A DBK contains three parts: area code (first three digits will always be preceded by a zero), page number (next six digits), and record number (last three digits), respectively. It could be displayed in either decimal or octal format.

The format only changes for the area code; the number system is converted automatically by the system when requested. Refer to the following DBK table as an example.

DBK 101000001001 (decimal format)			DBK 145000001001 (octal format)		
Area code	**Page number**	**Record number**	**Area code**	**Page number**	**Record number**
101	0000001	001	145	0000001	001

Database pointer

Along with DBK, every record is associated with a database pointer (DBP). Each DBP uniquely identifies a record on the page. A pointer is a single link between two records that indicates the physical location of the other record on a page. We call it a physical pointer. The DBP also has three parts: area code, page number, and slot number. It also takes the same format as the DBK. The DBP takes into consideration CALC chain record headers located at the bottom of each page (only if they are requested).

You could say the DBP represents the relative position of a record on a page and the DBK represents the absolute location of that record. The DBP is used by many of the utility programs to access data in a database. When you look for a record initially, you would use a DBK. When you have found your record and want to find a record linked to the former one, you use the DBP. You will understand how to use pointers later when we talk about record linkages.

Data items

A data item is the smallest unit of data stored within a record. A data item can be subdivided into sub-data items. Although this could be done for local usage, it would not affect the actual storage of data on a record.

Words

A word is a unit of measure of data storage space. One word is equal to four ASCII (or six data field) characters. A word is also equal to 12 octal digits. (Our pointers are represented using the OCTAL number system. You can see their significance later.) Whenever a record is requested, the DMR moves a page into its buffer area and then passes the specified record to the data management communication area. This allows the run unit to look at the data without physically acquiring the actual space where the record is stored.

220. Record types

As we mentioned earlier, a record is a collection of related data items. In this lesson, we are going to talk about the different types of records and their characteristics.

Common characteristics

A record is the basic storage and logical access unit. It is not restricted to a specific length. The only requirement is that records of the same type be identical in size and structure. Regardless of type, all records contain a 12-character date-time field that also contains the date and time the record was last updated. Each record has a DBK. This is a unique identifier associated with each record. The DBK contains the necessary information to identify the area, page, and record location. Although this is not an address, it resembles one in that it gives the location of a record within a database. When a program requests that a certain record be retrieved from the database, the DMR copies the page, which contains that record into its memory buffer. The DMR then passes the complete record to the program. Once the program has acquired the record, it can select from the various fields, or data items, contained in the record.

Types of records

There are different ways of storing records on the database. The DMR uses the same method for storing and accessing a record for each type. This is called the record placement strategy. There are three types of records based on the record placement strategy we use for the DMS—direct, CALC, and Via Set.

Direct

A direct record is one that is accessed via the DBK. When a record is created and stored in the database, it becomes a record occurrence. Any number of record occurrences may physically exist in a database for each record type from zero to the maximum capacity of the area file designated for that record type. Closely related occurrences of a record type are usually placed on the same physical page by the DMR to improve the efficiency of the system. The DBK for the direct records is a specific key assigned by the database administrator at the time the database is designed. In the case of IMDS, they are assigned and controlled by the AFLCMC. Direct records are known as entry point records. They are the fastest records to store and access.

CALC

A CALC (calculation) record in the DMS–2200 is a record that can be accessed directly by providing key data items in the record and defined in the schema as CALC keys. These keys are normally unique; however, in special instances, duplicates are allowed. The CALC records are in three basic groups based on usage.

1. One group provides an entry point for stand-alone tables.
2. Another group facilitates the many-to-many relationship between records, thus cutting down on lengthy area or set walks.
3. The remainder primarily provides an efficient entry point to a chain to eliminate unnecessary chain or area walks. For example, the DMR takes the employee number and changes it into a numeric value that it places into a mathematical formula to generate a DBK.

CALC records serve as entry points to the IMDS database. DBMs usually use CALC keys to enter the database and look for other records.

Via Set

Via Set records comprise the majority of the IMDS database. They are not entry point records like direct or CALC records. They can normally be accessed through an entry point record and from an owner record. With one exception, they are no more than three levels deep. The exception is line replaceable unit data record (LRUD)–163, which is four levels deep. These records are referred to as indirect because they are only accessed by walking from a direct or CALC record.

221. Record relationships

In the UNISYS environment, records are stored at various levels. This is referred to as the hierarchical approach. The hierarchy is established by record sets using owner and member records. A set is a named collection of records with a hierarchical relationship to each other. Sets are used to establish hierarchical relationships between different records that exist in the database. Sets and records compose the logical structure of the database. Sets show what records are related (or linked) to each other. We have mentioned the term hierarchical relationship a couple of times now, so it is only fair to explain the structure of the database where the records exist at different levels.

Set type

Database records are logically related to one another via the set mechanism. A set is a named collection of record types having a hierarchical relationship to each other. A set is also the description or the relationship that can exist between two record types. Figure 2–8 should help you understand the concept of a set type. In a set relationship, one type is defined as the owner of the set and the other

type is the member of the set. This begs the question, which one is which? By convention, the one that is drawn above the other, the one on top, is defined as the “owner.” Notice the arrow always comes from the owner of the set and points to the member. Bear in mind, at this point, you do not know how many sets you have or how many members each set has. Basically, those different levels take the form of two types of records: owner and member.

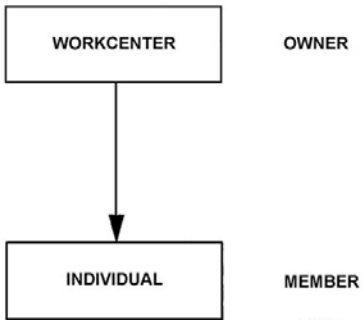


Figure 2–8. Set type.

Owner/member records

Owner records are at the highest level of a set with member records linked at a lower level to their respective owner. In order for a record to be an owner there must be records below it. This relationship allows member records to be owners of a set, and owner records to be members of a higher set. Owner and member records have several defining characteristics which include the following:

- A record set can have only one owner and the members of that set are accessed through it.
- Member records can be the same types of records or several different types.
- Members at one level may be owners at the next lower level.
- A set can have several members, and those members may be a single record type or may be several different types of records.

Chain

Linkages between records are known as chains. The mode definition, in the set section of the subschema, identifies the linkage that applies to the set. Whenever two or more records are linked, a set occurrence is established.

Set occurrence

A specific group of related record occurrences is called a set occurrence. A slightly more complex definition is the actual presence of the described record types on the database and the presence of the necessary linkage that confirms the implied relationship. The key to set occurrences is that they physically exist. Although the set is a logical concept, it is the pointer that makes it a physical reality. Pointers are 36-bit words that are an integral part of each record occurrence. They contain the database key of the next logical record in the set. Record occurrences of a set are logically linked together using pointers to point from one record to another. The use of pointers allows records in a set to be positioned anywhere in a database; they need not be physically contiguous. The two types of linkages we cover are chain next and chain linked prior.

Chain next

This type of linkage is referred to as single links, as shown in figure 2–9. The chain goes in one direction only to the next record occurrence. Every record in a set is logically linked to the next record. The owner record points to the first member in the chain, and that record points to the next, and so on through the set with the last member pointing to the owner. The owner will also point back

to the last member of the set. The physical linking is established by the database pointers (the DBK in this case). This is also called a *one-way circular linkage.* This is the default linkage.

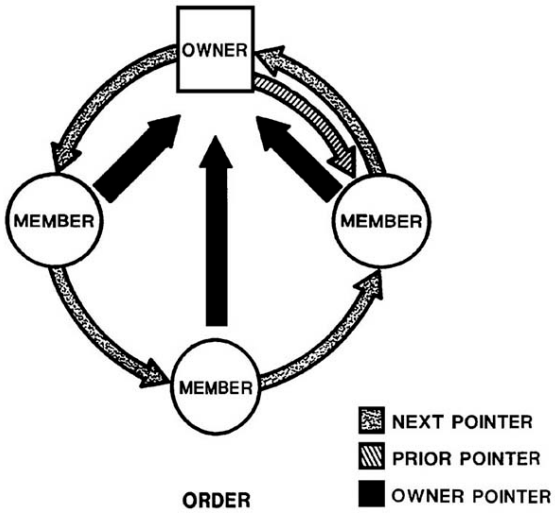


Figure 2–9. One-way circular linkage.

Chain linked prior

If database designers think it may improve access time to the database by allowing the DBM to follow a set in reverse order, they may specify that certain record types have prior pointers. Chain linked prior type linkages exist where every record of a set also points to the record that precedes it (fig. 2–10). In this type of linkage, the owner points to the first member in the chain (the same as in chain linkage). The owner also points to the last, the last points to the previous, the previous points to the first, which points back to the owner. This type of linkage is referred to as double or bidirectional linking. In addition, each member points back to the owner of the set. This is also known as *two-way circular linkage* and is defined in the set section of the schema by the clause *linked prior*. It is written in the subschema as chain linked prior. However, since each pointer takes up to 36-bit words of storage space, and there may be thousands of record occurrences using prior and owner pointers to improve access time costs storage space. The designers of a database must thoughtfully weigh the value of faster access time with cost of additional storage space.

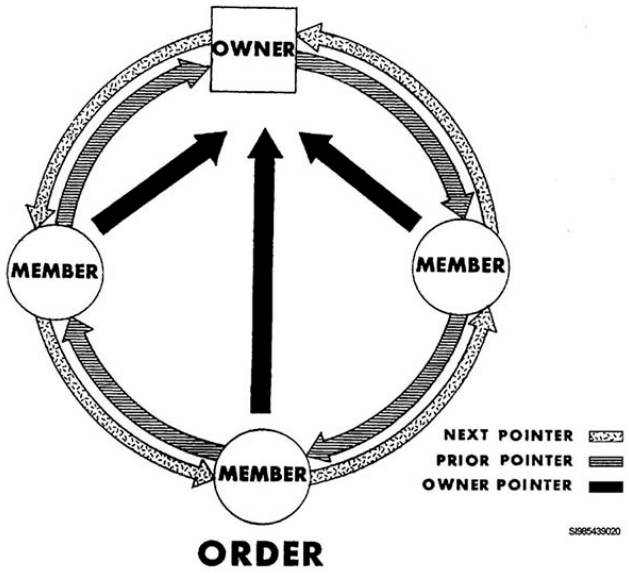


Figure 2–10. Two way circular linkage.

Record placement

When new member records are added to an existing set, the logical placement is determined by the order clause in the set section of the schema.

Placement Method	Explanation
First	Places the newest records closest to the owner, placing the chain in descending order—last in, first out.
Last	Places the newest record farthest from the owner, placing the chain in ascending order—last in, last out.
Next	Is used when the newest record is required to follow a specific record previously existing in the same chain.
Prior	Is used when the newest record is required to precede a specific record previously existing in the same chain.
Sorted	Is used when the newest record needs to be in a specific order based on the value of a key field that is in the member's chain.

Set participation

We are concerned with two types of set participation—automatic and manual.

Automatic

Member/owner linkages are established by the system when a member record is created in the database. The DMR is responsible for proper record association and linkage, as defined in the schema.

Manual

Member/owner linkages are established on an explicit program request. No manual linkages are established when records are stored in the database. Programs are responsible for proper association and linkages as necessary. A record is manually inserted to establish relationship.

222. Reading data management schematics

When you look for a place in a city, let's say a house or a restaurant, you would normally ask for directions. Because directions can get confusing, using a map gives you an idea of the direction and distance of the place you're looking for. Your reference is usually the starting point, which is your current location. A map might not show the specific physical location of the place, but it can guide you there. This is a logical approach to search for a place. A database is just as complicated as a city; however, it is organized in a logical structure. A database diagram therefore also helps us in looking for a type of record. It does not show us the location of a specific record. We only use it as a guide.

Schematic

The DMS database diagram is called a schema diagram. It is also referred to as schematic. It is a diagram of the logical structure of the DMS schema—composed of areas, records, and sets. It shows the relationship of records and sets in a particular database area. It also shows the types of records and some basic characteristics of the record.

The diagram shown in figure 2–11 is a sample general layout of the DMS schematic. This is not the actual DMS schematic. Figure 2–11 is used for illustration only. Using figure 2–11, we explain what the boxes, arrows, and codes (numbers) mean. You need to know the meaning of the symbols and elements in a schematic. This will aid you in understanding the current schematic in use. The actual schematic consists of several pages and serves as a road map to help you explore the database.

Linking

If the schematic can consist of many pages, there must be some way of linking them together. Looking at figure 2–11, the two-circled numbers—4 and 12—represent areas. For the 7R1 DMS schematic, they are the ELC areas.

Off-page connector

Notice the circled numbers: these are called off-page connectors. An arrow pointing from a circled number 4 to a record indicates an off-page connector from another area, whereas an arrow pointing to a circled number 12 from a record indicates it links to another area. Next to the circled number is the referenced record. If an arrow points from a record to another area, you must locate the reflected area to continue your search. From our example, you can see that record UNIT–052 leads to area 12 Via Set code 222, where it leads to record MMON–219. You also can see that the EVT–370 record in area 4 leads to the ISP–168 record.

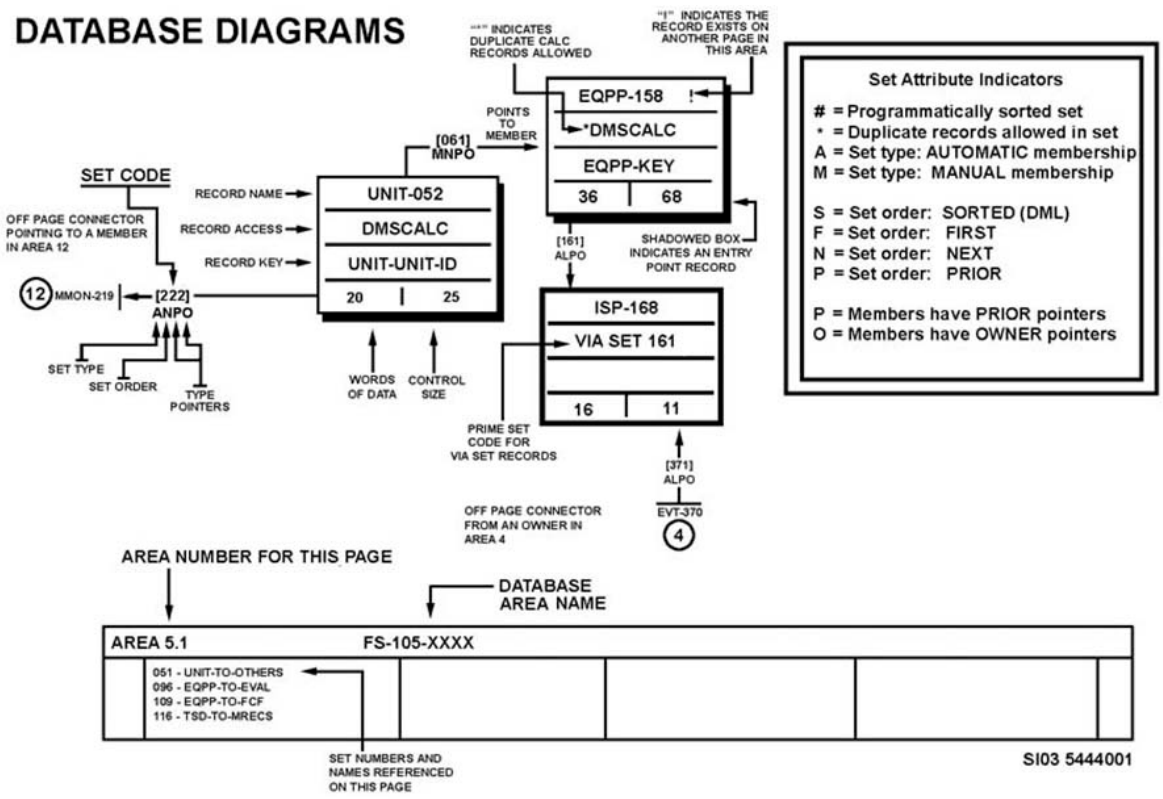


Figure 2–11. The data management system schematic.

Records

The large three rectangle boxes represent records. Each illustrated record is divided into five segments. Examine each section, starting from the top. The record name is located in the top section. Underneath the record name is the record access method: DMSCALC, Via Set, or direct. If a record has a CALC key, it will be located below the record access method. Only CALC records will have a CALC key. Other records are accessed by their respective DBKs or set codes. Of course, you would use the DBK to access a direct record.

But what about a Via Set record? Set codes are located between some records. Notice, for example, set 161 links one of the three records shown. You might use the set code when building QLP retrieval or when using database look (DBLOOK). The set attributes associated with a record indicate how records are linked and show whether they are members or owners. The bottom row is split. The left half reflects the record size in words; the right half is the control size.

Set code

In the middle of the lines between records is the numeric set code within brackets. If there are several member records to one owner record, the link between the two records is still shown with one set code.

Area number

At the bottom of the page is a long rectangle box, indicating the area number. The upper left column is where the area number is placed (e.g. “area 5.1”). The area name is also shown (e.g. “FS–105-XXX”). A whole number is used when an area only consists of one schematic page**.**

Multiple pages

If an area cannot be placed on the same page, there will be multiple pages. The primary area page starts with N.1 (where N is a whole number). If area 5.1 is shown, this means that the other pages that follow for the same area are in successive decimal numbers (i.e., 5.2, 5.3), depending on how many pages were utilized in the diagram for the same area. Since multiple pages are required for area 5, the first page begins with 5.1, a decimal; the second would be 5.2, etc.

Set number and name

In the box under the area number and the area name are the set numbers and name. Listed here are the sets associated with this area shown in this particular page. In this example, sets 051, 096, 109, and 115 are the set codes you will find on the page.

Online schema diagram

The online schema diagram for IMDS can be viewed at the “The IMDS Database Help System.” This online resource can be accessed from the A4QJ Maintenance Analysis SharePoint site under ACC A4 Directorate of Logistics. It can also operate on a PC or server as a standalone system (when loaded independently). It is very user friendly, especially with the help function. The diagrams are found under “Schema Diagrams.” We are not giving you the website since they always change. Since it is an interactive (“point and click”) online diagram, it is much easier to navigate to the different areas through the “Area Map Navigator.”

Figure 2–12 shows the screen snapshot of the online schema diagram. It depicts the 7R1 DMS and RDMS schema, as shown on the dark menu on the left hand side. For the DMS schema, we are just concerned with the roll-up and ELC areas (upper two sections of the menu). On the dark menu, you can see the rollup areas 2 and 6 and ELC areas 1 through 14. The “ELC area 1” was selected; thus, you see area 1 (area name is Area 101) on the sample screen. You can see a full view of the area on your computer monitor. We are going to talk about the RDMS areas (lower section of menu) next.

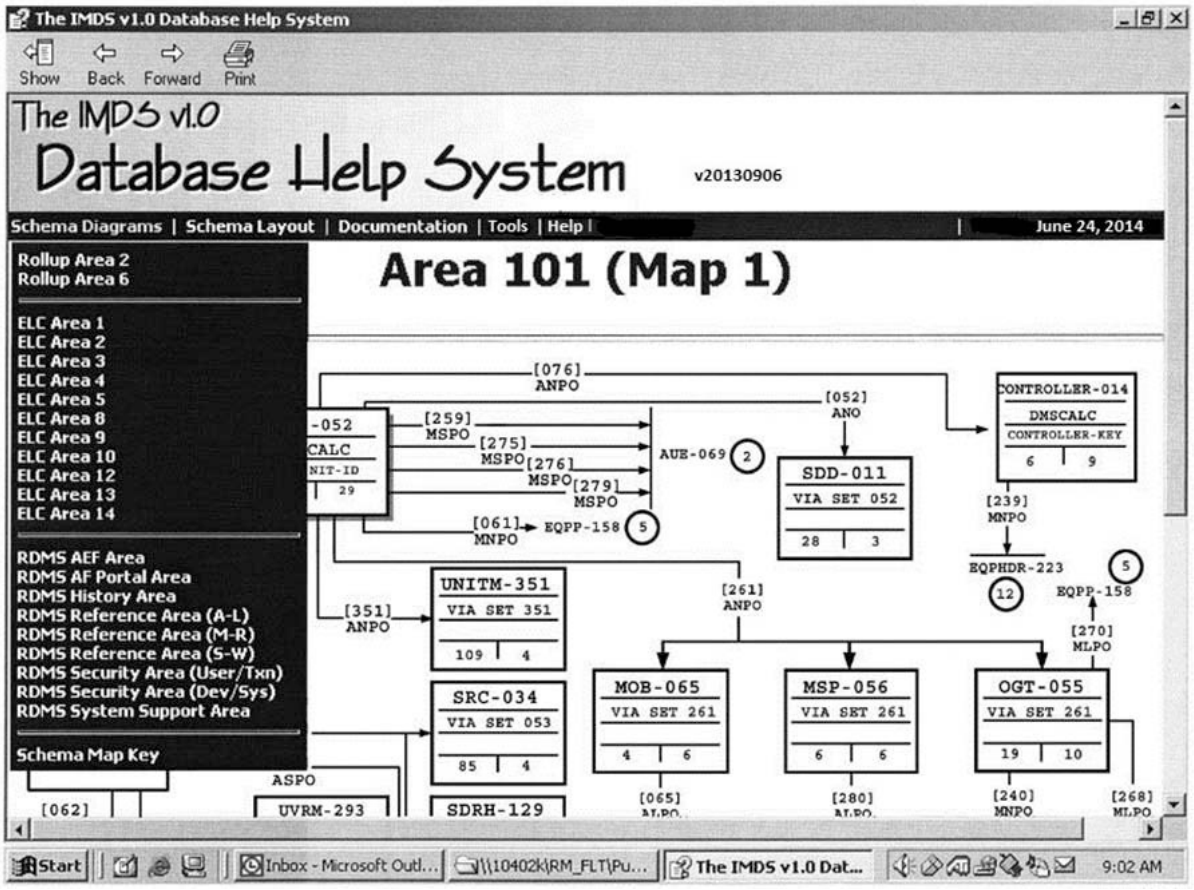


Figure 2–12. Online schema diagram for the data management system.

The next figure (fig. 2–13) is a screen snapshot of a full page of the 7R1 online schematic. On the top is the area name. This full page shows all the records, sets codes and set names and all the page connectors. The interactive area map navigator links the current page to the applicable area to which you wish to go. In this case the circled number on the page translates to the area name on the navigator (circled 2- 102, circled 5 – 105). When you want to go to the circled 5, you point to “105” on the navigator and click on it. It will take you to another webpage that shows area 5.

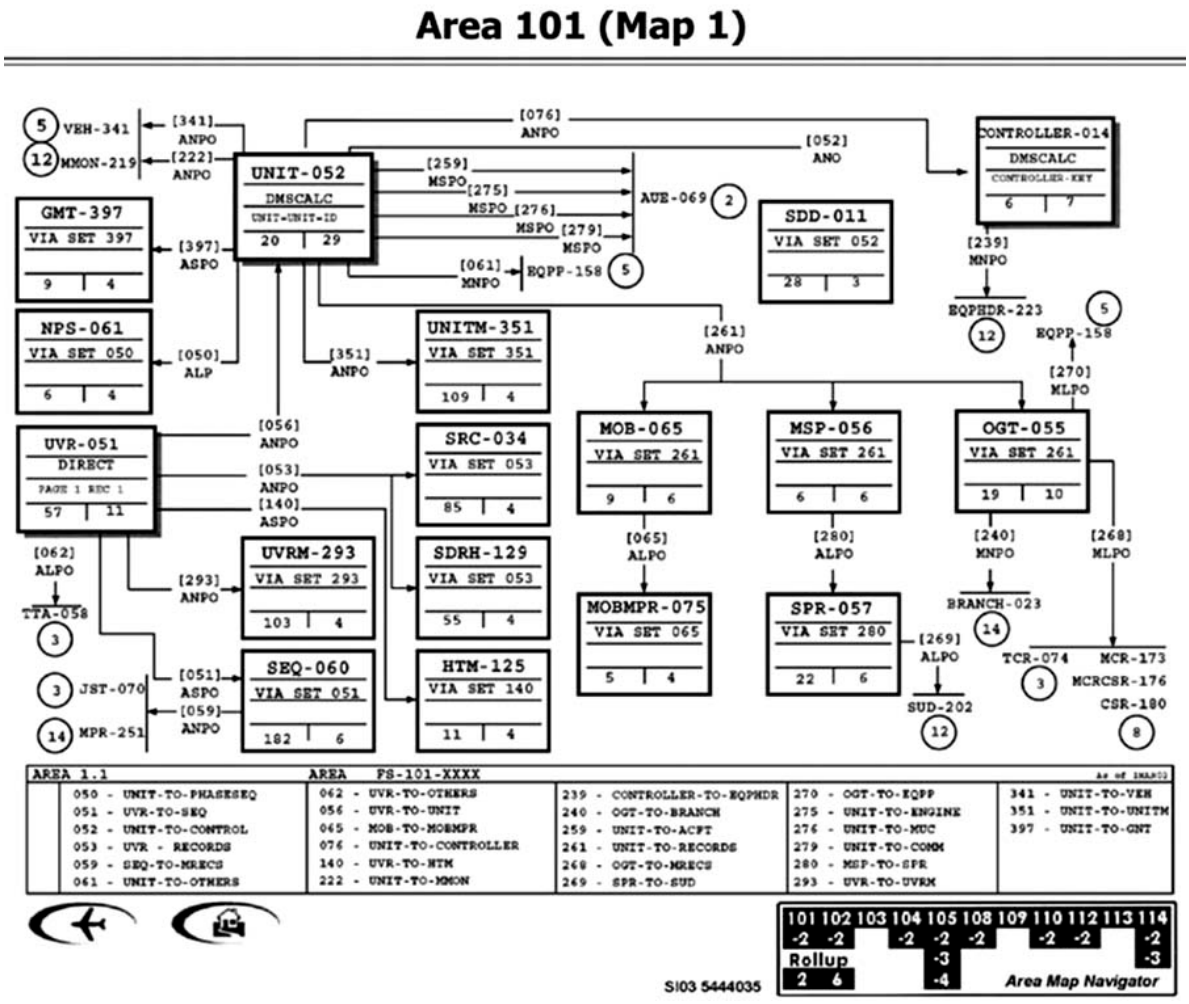


Figure 2–13. Actual IMDS online schema diagram page.

This online help system is easy to use. We will again look at the other features of the help system as we go along in our lessons.

Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

217. Database structure

1. What is the schema?

2. What are the four sections of the data division within the schema and what are their purposes?

3. What does the subschema do?

4. What is the DDL?

218. Data Management System schema

1. When reading the area section of the subschema, what does sizing information mean?

2. What do the last four characters in the area name represent?

3. How many roll-up areas are there in the 7R1 DMS schema?

4. What purpose does the record section of the subschema serve?

5. What is defined in the set section of the subschema?

219. Page and record structure

1. What is the highest subdivision of a database?

2. What is a page?

3. What are the three parts of a DBK?

4. What is the difference between a DBP and DBK?

220. Record types

1. What is a record placement strategy?

2. What are the three types of records based on the record placement strategy?

3. How are direct records accessed?

4. What type of records can be accessed using key data elements identified in the schema?

5. What type of records comprise the majority of the IMDS database?

221. Record relationships

1. What are the owner/member record relationships?

2. How can you describe one-way circular linkage?

3. What is meant by linked prior?

4. Describe the record placement method *first*.

5. What is meant by automatic set participation?

222. Reading data management schematics

1. What kind of structure does the DMS database diagram, or schematic, describe?

2. How are off page connectors from another area reflected?

3. How are off page connectors to another area reflected?

4. What does a rectangle box represent?

5. What does the online schema diagram depict?

2–3. Data Modeling

To understand the operation of the RDMS part of IMDS better, we first need to discuss some concepts of data modeling and relational databases. As a maintenance analyst, you cannot limit your gathering of data to just one MIS. There are many different electronic sources available: IMDS, G081, and REMIS, just to name a few. Not to mention the enormous amount of data in your maintenance complex that is not stored in a database. All this data is stored and accessed differently, whether electronically or on paper. How can you easily compare data you've taken from one source to data from another source? Programs such as Monarch are readily available to strip unneeded data quickly from huge reports run from an MIS, leaving you with only the data you need. But what do you do with the data? Here we'll discuss using data modeling to help you structure and organize your data so it is useful and meaningful, and give you a basic understanding of building a relational database such as the RDMS.

223. Introduction to data modeling

A data model is simply a description of data structures that are required by a database. These structures include data objects, relationships between data objects, and rules that govern the manipulation of the data objects. Data models focus more on the structure and organization of the data than operations to be performed on the data; in other words, a data model is sort of a building blueprint for a database. Data models represent data as the user would see it rather than the database itself and are not strictly defined by hardware or software. A data model is used to design the relational tables used in relational databases.

The goal in building a data model is to ensure all data required by the database is represented completely and accurately. A data model must be simple enough for the user to understand the structure required by the database while detailed enough to design an efficient physical structure for the database. An effective model eliminates redundant data, and can be adapted easily to changing requirements. The model is used to define relational tables, primary and foreign keys, and procedures for manipulating the stored data. Without careful planning, your database may omit required data, produce incorrect or incomplete queries, and suffer a loss of data integrity.

Planning and analysis

Database design is defined as "the logical and physical structure of one or more databases to accommodate the information needs of the user in an organization for a defined set of applications." The first step in building an effective data model is the planning and requirements analysis. During this stage, you (the modeler) collect information about the requirements of the database, as well as collect data that will make up the records in the database.

You can gather this information in a variety of ways. Review existing documents, reports, and such to determine how the data was previously structured and presented. Talk to your customers to determine how they need to view the data. Review existing automated systems to determine similarities in the data that can be used to build relationships.

Planning and analysis provide your roadmap to building a successful data model. Your plan should define the goals of the database and provide the path to reach those goals. Your analysis should determine the requirements of the database. You should develop a basic model representing data elements and relationships. Then add the details by including data attributes, manipulation and integrity rules.

Requirements analysis

Requirements analysis should determine the following:

- Data requirements of the database in terms of raw data elements.
- Classification and description of these elements.

- Relationships among the objects.
- Types of transactions performed on the elements.
- Rules governing data integrity.

Requirements analysis is normally accomplished simultaneously with the data modeling. As information is collected, data elements are classified as entities, attributes, or relationships. They are assigned names and defined using terms familiar to the data user.

224. Creating a data model

The first step in creating a data model is the identification of data elements and relationships. As with other database systems, a relational system has structures and rules you must know to understand how it operates. A relational database structure consists primarily of tables. We illustrate the structure with simple diagrams that represent these characteristics, known as a data model. The diagrams that accompany this lesson do not represent the actual physical structure of the RDMS tables; they only show similarities. For you to understand the structure, you must learn some terms that describe the unique characteristics of RDMS structure. Let's take a look at these items.

Relational tables

Relational databases consist of one or more two-dimensional structure of data values that form relations. These structures are known as relational tables or simply, tables. The two dimensions that make up a table are vertical columns and horizontal rows. Figure 2–14 illustrates the two dimensions of a table. This is not a sample of the IMDS relational table. Each table has a name. Each table has a specified number of named columns and any number of unnamed rows.

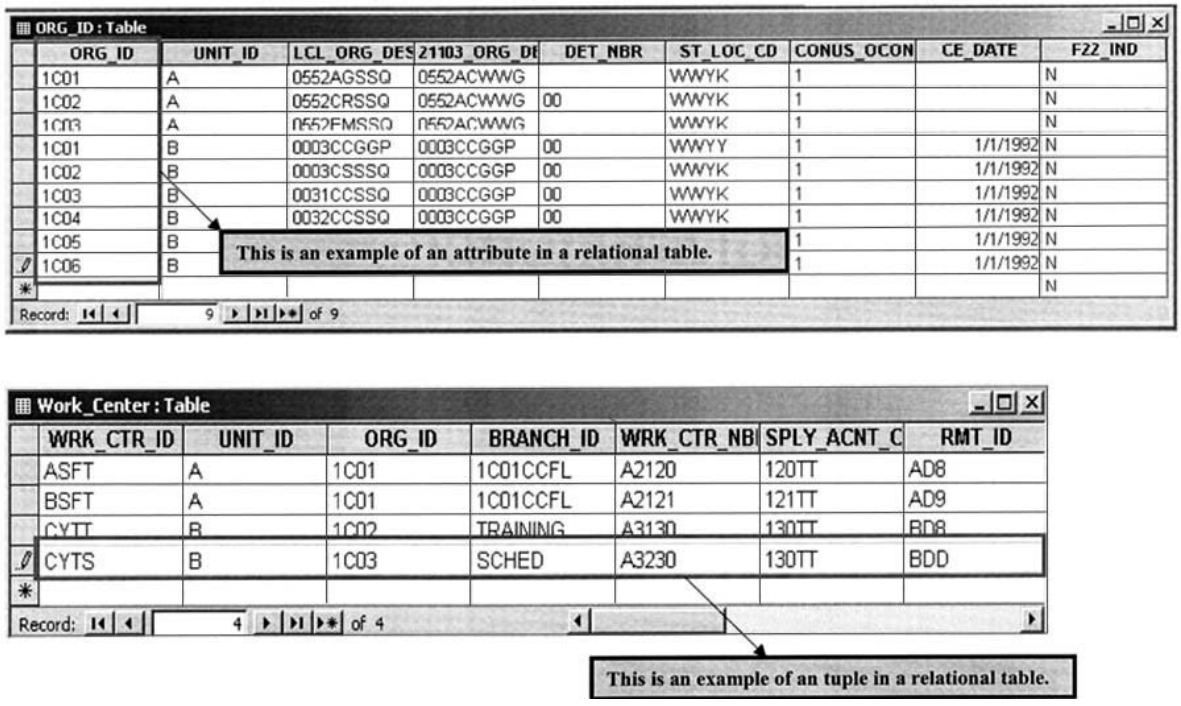


Figure 2–14. Two-dimensions of a table.

Column

Each column is defined to hold a specific type of data. The columns are called attributes. An attribute is described as one characteristic of an entity or relation. Attributes describe one characteristic of the data stored within the table. An entity in real life becomes a table in a relational database.

Row

A table is a row. Rows contain different types of information about one thing. Each row represents one occurrence of the entity represented by the table. This occurrence is called a record or a tuple.

In figure 2–14, there are two separate hypothetical tables – one is the "ORG_ID" table (top part) and the other is the "Work_Center" table. In the ORG_ID table, the boxed column represents organization ID codes as the attribute for that column. The heading was appropriately labeled "ORG_ID." Only organization ID codes are to be placed in that column. On the other hand, on the Work_Center table, the boxed row represents the occurrence of one entity. This occurrence has some data related to that entity throughout the row as specified by each attribute. The entire row, then, is considered one record occurrence of the data represented by this table.

Values

A value is a numeric, alphabetic or alphanumeric character string stored for an entity (such as an attribute) at the address allocated for that entity. It is also known as data value. The intersection of a column and a row is a data item. The specific value of a data item is a data value. Generally, a value is the data stored in a record. These values must be *atomic* (in other words indivisible). In a relational database, a value cannot be broken down further into several parts.

Null values

If an item is missing or irrelevant, a null value is inserted in the field. A zero (0) in a numeric field is a specified value; a null value has no value. An entire row cannot be empty; if it were, there would be no occurrence. However, unless it is defined as Not Null, a column in a row may be empty (fig. 2–15).

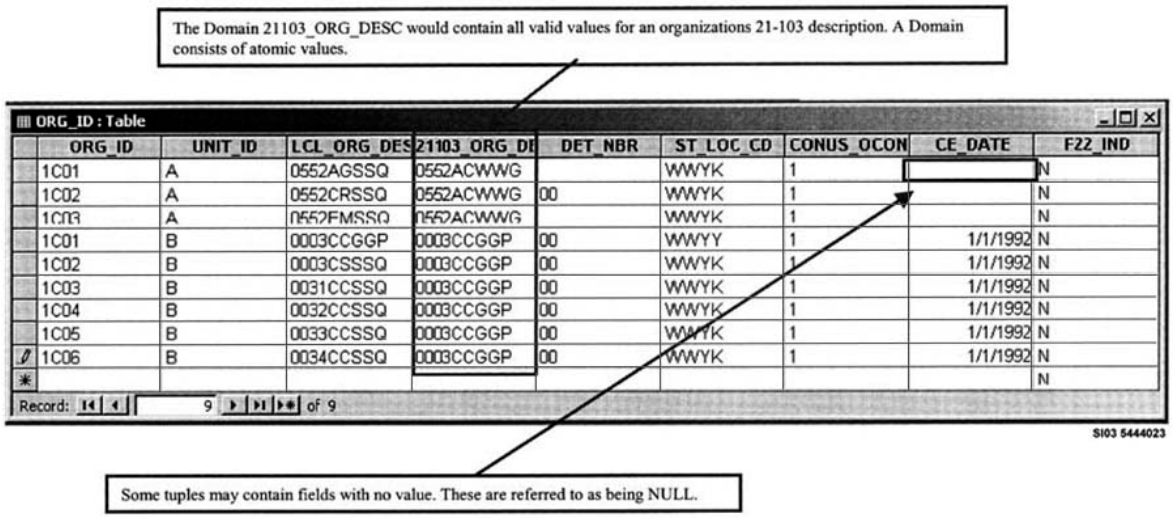


Figure 2–15. Domains and null.

Domains

The value for a column is drawn from a domain. A domain is a set of atomic values. For example, look at figure 2–15. The domain "21130_ORG_DESC" contains all valid organization description values.

Keys

A key is a column (or combination of columns) in a relational table that uniquely identifies a row in a relation. It is also called an identifier. You can find a specific row (record) through the key.

Candidate key

A candidate key is any column that contains values in a table that can be used to identify a row uniquely. When we say that a value uniquely identifies a row, that value is used only once for that column. A table may have more than one column that qualifies for this. If you look at figure 2–16, the darkened column “LCL_ORG_DES” is a candidate key.

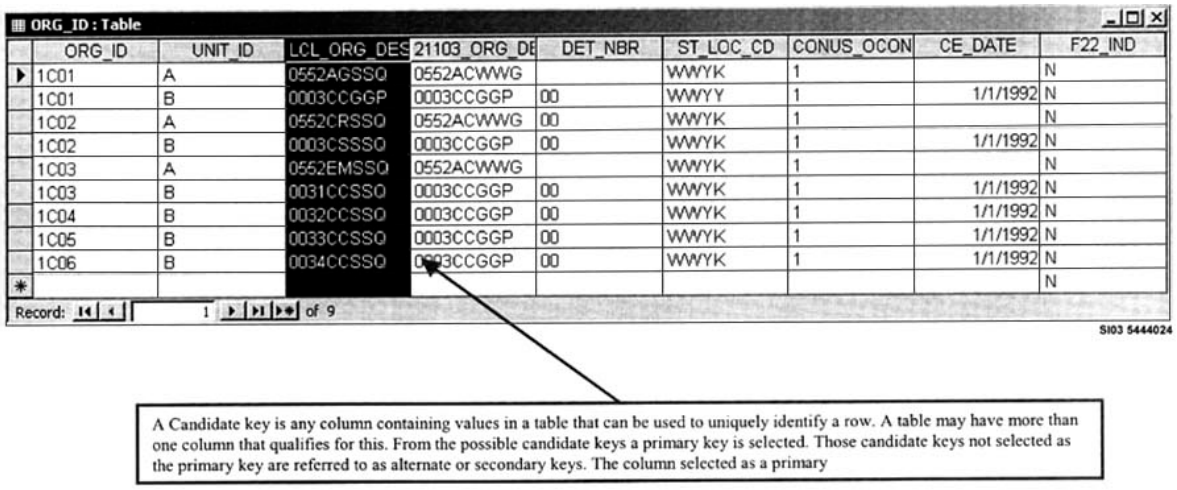


Figure 2–16. Candidate and primary key.

Primary key

The primary key is the key chosen from among the candidate keys to act as the unique identifier for the table in a relational database. Candidate keys not chosen are referred to as alternate or secondary keys. The column chosen as the primary key may not contain null values. Additionally, the candidate key can also be chosen as the primary key.

Concatenated key

When a table does not contain any single column that can be used to uniquely identify a row, it may be necessary to use a combination of columns. This is called a concatenated key (fig. 2–17). The darkened columns in figure 2–17 “ORG_ID and UNIT_ID” serve as the concatenated key. The combination of their respective values identifies the row.

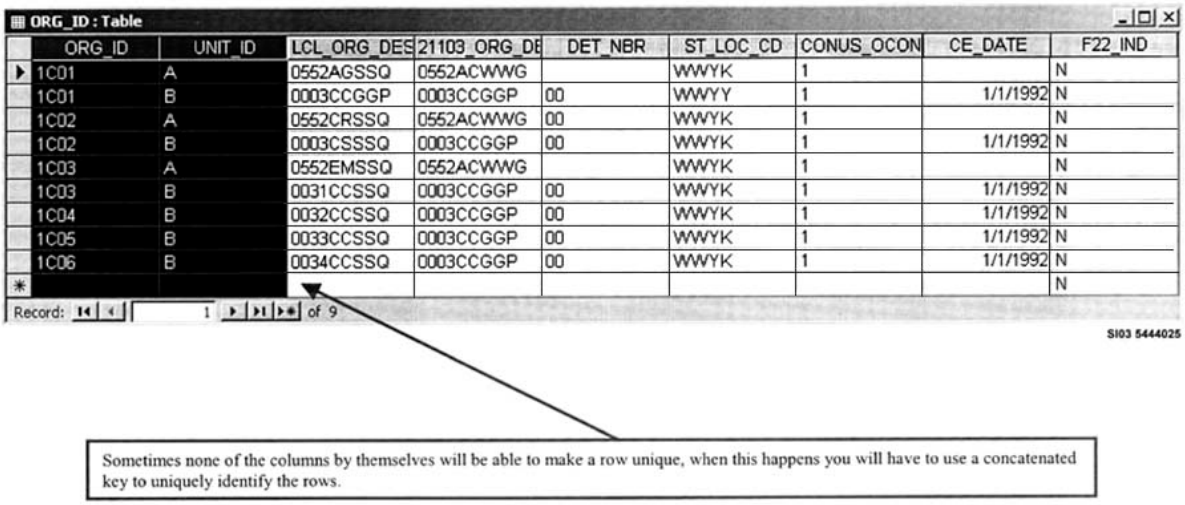


Figure 2–17. Concatenated key.

Foreign key

RDMS uses foreign keys to establish the relationship with records in one table to records in another table. This type of key is nothing more than the same data being stored in the two different tables. The common data provides a logical link between the two tables. Foreign keys are useful when there is certain hierarchy, or levels, established between the tables, similar to a parent-child relationship. The foreign key relates the table to its parent table. The column(s) comprising a foreign key from the child table must be unique in the parent table. As an example, let's refer back to figure 2–16. You have these two tables in your RDMS database: one is an organization table "ORG_ID" (parent), and the other one is a work center table "Work_Center" (child). Let's say that the primary key in the organization table is the combination of the ORG_ID and UNIT_ID columns. The primary key of the Work Center table is the WRK_CTR_ID column; however, this table also contains a UNIT_ID column and an ORG_ID column. These two columns in the Work Center table can be used to establish a logical relationship between the two tables. These two columns would be referred to as the foreign key in the Work_Center table.

Data integrity rules

In terms of relational databases, data integrity means that you can correctly and consistently navigate and manipulate the tables within a database. There are two basic rules to ensure data integrity: entity integrity and referential integrity.

Entity integrity

This rule states that the value of a table's primary key can never be a null value. Remember, a null value has no value and is not the same as a blank or a 0. The primary key must always be specified because it is used to uniquely identify a row in a table. Any inserts, updates, or deletes must maintain the uniqueness of all primary keys.

Referential integrity

Referential integrity means that if a table has a foreign key, every value of that foreign key must be null or match the values in the table where it is a primary key. Columns UNIT-ID and ORG-ID of the Work_Center table must match the ORG_ID and UNIT_ID columns in the ORG_ID table.

225. Establish relationships

A data relationship is simply an association between two or more data items. Relationships are also established between tables. They are expressed in the data values of the primary and foreign keys. Navigation in a relational database depends on the primary key uniquely identifying specific rows or records in a table. Let's discuss some characteristics of data relationships.

Types of relationships

After defining data elements with attributes, you must determine if a relationship exists between the data. A relationship is a two-directional significant association between two data elements. Each is defined with a name, whether it is optional or mandatory, and a degree (how many). They are described in real terms. For example, a person must have only one Social Security number, and a Social Security number can be for only one person. We will discuss three types of data relationships; one-to-one, one-to many, and many-to-many.

One-to-one

In a one-to-one relationship, each row in a table is related to a single row in another table. In other words, a primary key will appear only once in a related table. This type of relationship is the least occurring of the three types. An example would be a relationship between personnel and IMDS user IDs. Each user that accesses IMDS must have one and only one IMDS user ID. The relationship between users and IMDS User IDs is a mandatory existence. Each IMDS user ID can be assigned to one and only one user. However, not all user IDs will be assigned to a user at a given time. This condition may reflect that some user IDs are kept for spares or reserved for TDY personnel.

Therefore, the relationship between IMDS user IDs and users is an optional existence. This type of relationship sometimes can be used to break down a complicated table into more usable data elements. You may be able to use one-to-one relationships if you have attributes (columns) in a table that pertain to only a few records (rows) in your table.

One-to-many

The one-to-many relationship is the most widely used in relational databases. A one-to-many relationship allows records in one table to easily relate to an unspecified number of records in another table without adding redundant attributes or limiting the number of attributes in a single table. Let's suppose you are building a database to represent what type and how many parts each work center has ordered. Your first draft has columns for Part1, Part2, and Part3, with the corresponding Quantity1, Quantity2, and Quantity3 fields.

Work_Center	Mnemonic	Order #	Part1	Quan1	Part2	Quan2	Part3	Quan3

As you can see, if a shop only orders three different parts, this works. What happens if a shop orders more than three parts? Any time you start adding similar fields like this to the same table, you need to break the table into two or more related tables using a one-to-many relationship. Using one-to-many relationships in a relational database will reduce the size of the database and increase the flexibility and performance of queries performed on the data.

In our shop ordering example, we can break our table into an order table and item table.

Order Table		
OrderID	Work Center	Mnemonic
1	Analysis	MX0A

Order Item Table			
Order ItemID	OrderID	Item	Quantity
100	1	Pencils	100

As you can see, the two tables are related using the OrderID field. The contents of any order in the Order table can be found by finding all the items with a matching value in the OrderID field. The tables are independent and can easily be modified. If we wanted to add a national stock number (NSN) to the Order Item table, we simply add a single column, instead of adding NSN1, NSN2, and so on. If we wanted to order additional items for the Analysis work center, we simply add a "1" in the OrderID field to for the item to designate it as an item ordered for Analysis.

Order Item Table			
Order ItemID	OrderID	Item	Quantity
100	1	Pencils	100
200	1	Notepad	20

Many-to-many

The third type of relationship we will discuss is the many-to-many relationship. Many-to-many relationships are more complex than one-to-many. This type of relationship means that multiple occurrences of one data element are related to one occurrence of another, and vice versa. For instance, you are building a database to store information about trainers and trainees. One trainee can have many different trainers teaching different tasks. On the other hand, one trainer can train many different trainees. How do you depict this data without resorting to repeating attributes such as this?

Stud_#	Name	Trainer1	Trainer2	Trainer3	Trainer4	Trainer5
1170	Raison, Kenneth	MW	JB	DN	HA	CG

AND

Trainer_Id	Name	Stud1	Stud2	Stud3	Stud4	Stud5
MW	Wussow, Mike	KR	NC	CR	MJ	JM

As you can see, there can be an indeterminate amount of student and trainer fields. This type of relationship cannot be used in your data model because they cannot be represented in a relational database. Therefore, they must be resolved early in the modeling process.

To resolve many-to-many relationships, you must create a third table, called a linking table, to join the two tables together and create two one-to-many relationships. This table will contain two columns, the primary keys from the other tables. For example, we'll call the linking table Trainer_Student:

Table 1		Table 2		Table 3-Linking Table Trainer_Student	
Trainer_ID	Name	Stud_#	Name	Trainer_ID	Stud_#
MW	Wussow, Mike	1170	Raison, Kenneth	MW	1170
JB	Boyd, John	1620	Owens, Jeff	JB	1620

Using a linking table, we've moved all redundant data from the Student and Trainer tables to the linking table, which is made up of only two columns of integers. If we decide to make changes to the structure of the main tables, the changes are independent of their relationship. For instance, we can add a Date_Trained column to the Student table without disturbing the Trainer table or their relationship. We could also add a Course_Code column to the linking table and link it to Courses table without disturbing the first two tables.

Logical relationship between tables

Unlike DMS databases where records' relationships are maintained through the use of physical database pointers, in relational databases the relationships between records are logical. This logical relationship is created and maintained through the use of key fields. These key fields are the common data between two or more tables and serve as the logical pointers. Refer to figure 2–18. You can see that both tables use "UNIT_ID" and "ORG_ID" for their attributes. They can serve as the logical pointer or link between these two tables because the values they contain are the same for both tables.

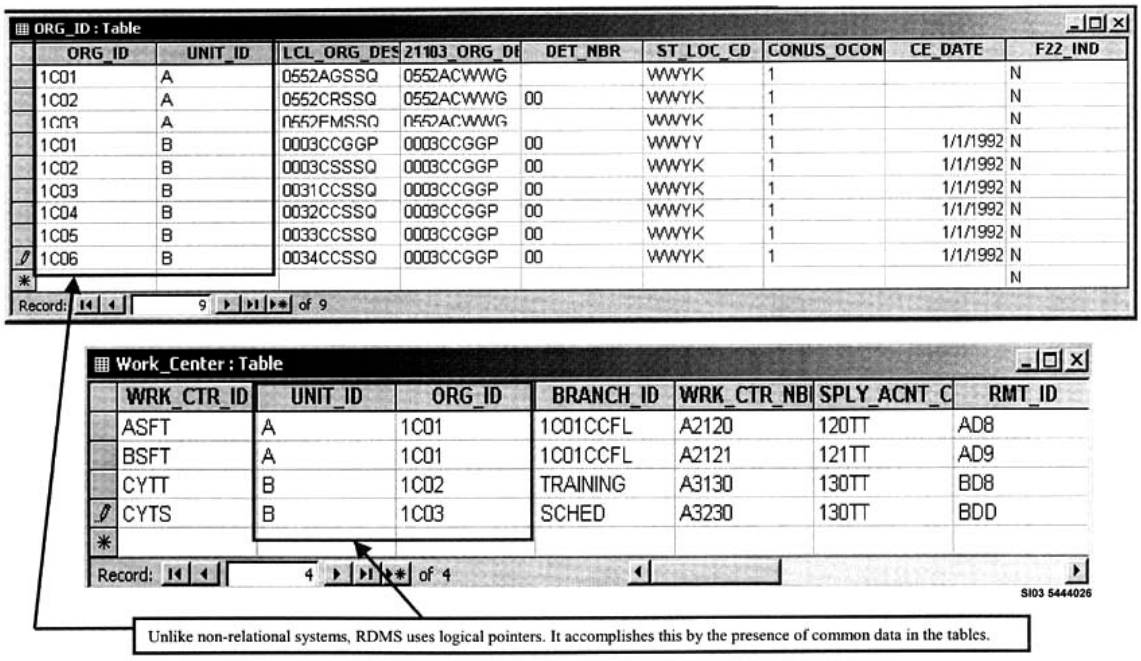


Figure 2–18. Logical pointers between tables.

Properties of relational tables

Relational tables have several properties:

1. Every table in the database must have a unique name—no two tables within the same database may be named exactly the same.
2. Values are atomic—columns in a relational table are not repeating. This meets the first normal form (1NF). This is one of the cornerstones of a relational data model.
3. Column values are of the same kind; that is, the values in a column are from the same domain—in other words, columns only contain specific values, not other information such as comments, status flags, or data from a different scale (monthly versus weekly).
4. Each row is unique—no two rows in a relational table are identical. At least one column of volumes must uniquely describe each row or record. See primary key above.
5. Sequence of columns and rows is insignificant—order of the columns or rows makes no difference. Allows user to share tables without regard to table organizations.
6. Each column has a unique name—columns must be referenced by name and not by position, such as in a spreadsheet. Column names need not be unique within the entire database, only in its table.
7. A primary key field may not be null—null is not the same as being equal to zero or blank. Both zero and blank are considered to be values. Null means that the value is equal to nothing.

Indexed fields

Relational databases use indexed fields to improve query performance and efficiency. Primary key fields are automatically indexed and others may be indexed. Write your queries to include indexed columns in your conditions; otherwise, full table scans are required. A full table scan practically goes to each table in the database, which on large tables can take a great deal of time to accomplish.

226. Normalizing data relationships

Data normalization consists of optimizing table structures and organizing data in the tables so the data will always be correct and precise, so that your database contains less redundant data, is more efficient, and is more adaptable to changing requirements. The goal of normalization is to create a set of relational tables that are free of redundant data and with records that can be easily queried and correctly updated, added, or deleted. A table is said to be normalized if all non-key items are mutually independent and fully dependent on the table's primary key. In this lesson, we will discuss three levels of normalization: first, second, and third normal forms. The first two are intermediate steps to achieve the goal of all tables being in the third normal form. A higher form of normalization cannot be achieved unless the previous level has been satisfied.

First normal form

Suppose you were creating a database containing personnel in your analysis office, what projects they are assigned to, and the time they've work on each task. We'll call it "Analysts_Projects_Time," and it might look something like this:

Analysts_Projects_Time			
Emp_ID	**Name**	**Task**	**Time**
001	Blow, Joe	1,3,5	0.25, 0.40, 0.50
002	Smith, James	1,3,5	0.40, 0.25, 0.75
003	Brown, Jane	2,4,6	0.05, 0.35, 0.60
004	Snuffy, Bill	2,4,6	0.90, 0.75, 0.20

The following rules must be met to normalize a data model to the 1NF:

- Values in each column of a table are atomic—no sets of values within a column.
- Eliminate repeating groups in individual tables.
- Create a separate table for each set of related data.
- Identify each set of related data with a primary key.

In our preceding example, we have a set of data in the name column, a first and last name. These are in fact separate values. If they are not separated, it can become difficult to perform a sort on the last name. Notice the task and time columns also contain multiple pieces of data. There is no way to know which allocation of time belongs to which task. Let's add columns so that each field contains only one piece of data.

Analysts_Projects_Time								
Emp_ID	**Last_Name**	**First_Name**	**Project1**	**Time1**	**Project2**	**Time2**	**Project3**	**Time3**
001	Blow	Joe	1	0.25	3	0.40	5	0.50
002	Smith	James	1	0.40	3	0.25	5	0.75
003	Brown	Jane	2	0.05	4	0.35	6	0.60
004	Snuffy	Bill	2	0.90	4	0.75	6	0.20

Now, what if an analyst was assigned a fourth task? You would have to add two new fields to the table. Remember that similar items belong in the same column with a single value in each row. To normalize our database to 1NF, we must eliminate repeating groups of data from the table. We do that by separating similar data into, in this case, two different tables, and join these tables using its primary and/or foreign keys.

*Emp_ID	Last_Name	First_Name
001	Blow	Joe
002	Smith	James
003	Brown	Jane
004	Snuffy	Bill

Our table is now in the 1NF. Notice there are no repeating columns in the tables. However, our table does contain redundant data, which in turn causes what are called update anomalies. These are problems that arise when data is inserted, deleted, or updated. Notice that to update Emp_ID 002 with a new address, multiple rows would have to be updated. These anomalies are addressed when we continue to normalize our table to the next level.

Projects_Analysts_Time		
*Task_Num	*Emp_ID	Time1
1	001	0.25
1	002	0.40
2	003	0.05
2	004	0.90
3	001	0.40
3	002	0.25
4	003	0.35
4	004	0.75
5	001	0.50
5	002	0.75
6	003	0.60
6	004	0.20

Functional dependency

Before we can move on to the next normal form, we must first discuss the concept of functional dependency. Functional dependency is a relationship between fields such that the value in Field A determines the value in Field B, and there can be only one value in Field B. We would say Field B is functionally dependent on Field A. Look at the following table:

Unit	Base
362nd Training Squadron	Sheppard AFB, TX
363rd Training Squadron	Sheppard AFB, TX
552nd Air Control Wing	Tinker AFB, OK
57th Fighter Wing	Nellis AFB, NV
422nd Test Squadron	Nellis AFB, NV

Each unit has a unique name and can only be headquartered at one base. Therefore, the Base column is functionally dependent on the Unit column. The value in the Unit field determines what value can go in the Base field, and there can only be one value in the Base field. This does not have to work in

reverse. Each base can have more than one unit, so the Unit field is not functionally dependent on the Base field. You will sometimes see this written in the following format:

Determinant Field(s) >> Functionally Dependent Field

Second normal form

While the first normal form deals with atomicity of data, or each field containing the smallest meaningful value, the second normal form (2NF) deals with relationships between composite key columns and non-key columns. Remember, normalizing data is a progressive process. Your table must already be in the first normal form to achieve the second normal form. In the 2NF, any non-key columns must depend directly on the entire primary key. Notice the example in the following table:

Analysts				
***Emp_ID**	**Last_Name**	**First_Name**	***Task_Num**	**Task_Title**
001	Blow	Joe	1	9302 Report
002	Smith	Jane	2	Continuity Book
001	Blow	Joe	3	Analysis Webpage
003	Snuffy	Bill	4	F–16 Analysis
001	Blow	Joe	5	Health of Fleet
002	Smith	Jane	6	Training Monitor
004	Hammer	Jack	7	Mx History DB
003	Snuffy	Bill	3	Analysis Webpage
002	Smith	Jane	1	9302 Report
004	Hammer	Jack	5	Health of Fleet
003	Snuffy	Bill	2	Continuity Book

In the preceding table, the asterisks indicate the columns that make up the primary key. Since neither the Emp_ID nor the Task_Num columns contain data to uniquely identify a row, we must use a concatenated key; in this case, a combination of the Emp_ID and Task_Num columns.

Looking at this table, what if you wanted to add a new Task_Num and Task_Title, but not assign an analyst to it until later? We learned previously that a primary key cannot contain a null value. If we tried to add a task, so you couldn't add the task. If a Task_Title changed, you would have to update several different fields. There are several possible update anomalies that have to be dealt within our table.

Notice the repeating values in the Emp_ID, name, and project columns. Repeated values exist in the Last_Name, First_Name, and Task_Title columns because these columns are dependent on only part of the primary key. The field Emp_ID determines what goes in the Last_Name column, but has nothing to do with the Task_Num column. The Task_Num column determines the Task_Title data, but the Emp_ID column does not. The non-key columns relate to only part of the primary key, which is a combination of the Emp_ID and Task_Num columns. They are not functionally dependent on the entire primary key. This is not an efficient way to store data. The only data that should be duplicated is in key fields used to connect tables. Key fields will rarely change, while data in non-key fields may frequently change.

The following rules must be met to normalize a data model to the 2NF:

- Table must already be in 1NF.
- Any non-key columns must depend on the entire primary key.

To bring this table into the 2NF, we have to break the table into smaller tables where all non-key columns depend on the entire primary key. We have to determine which columns determine the values in other columns, and create a table for those columns. If necessary, we'll have to add a foreign key, or a junction table. You'll find that creating more, simpler tables is the solution to most problems you'll see in database normalization. After applying these techniques, our new tables should look something like the following tables:

Analysts		
***Emp_ID**	**Last_Name**	**First_Name**
001	Blow	Joe
002	Smith	Jane
003	Snuffy	Bill
004	Hammer	Jack

Analysts_Projects	
***Emp_ID**	***Task_Num**
001	1
002	2
003	3
003	4
001	5
002	6
004	7
003	2
001	3
002	1
004	5

Projects	
***Task_Num**	**Task_Title**
1	9302 Report
2	Continuity Book
3	Analysis Webpage
4	F–16 Analysis
5	Health of Fleet
6	Training Monitor
7	Mx History DB

In this case, we had to create a junction table for the sets of values that apply to multiple records (rows). Since we had a concatenated key as a primary key, we did not have to add a foreign key to our tables.

Third normal form

We've discussed normalizing data tables to 1NF and 2NF. First normal form deals with eliminating repeating groups. The second normal form deals with eliminating redundant data. The third normal form (3NF) requires that all columns in a relational table are dependent only on the primary key. For instance, examine the following table:

Projects			
*Task_Num	Task_Title	Task_Mgr	Phone
1	9302 Report	Smith	2756
2	Continuity Book	Jones	2954
3	Analysis Webpage	West	2846
4	F–16 Analysis	Gordon	3102
5	Health of Fleet	Jones	2954
6	Training Monitor	West	2846
7	Mx History DB	Smith	2756

This table has a single primary key (Task_Num), where "Num" represents "number," and repeating values in non-key fields (Phone and Task_Mgr), and where "Mgr" represents "manager." In addition, Mx represents "maintenance," while DB represents "database." The phone number repeats each time a task manager's name is repeated. That is because the Phone column is dependent on the Task_Mgr column, which in turn is dependent on the primary key, the Task_Num column. The Phone column does not depend directly on the Task_Num column. This is known as a transitive dependency, which is a type of functional dependency in which the value in a non-key field is determined by the value in another non-key field, and that field is not a candidate key.

The data in this table is in 2NF. However, what if a task was re-assigned to another manager? Or the phone number of one of the managers changed? In this case, every instance of the manger's phone number would have to be changed manually. Depending on the size of the table, this could be a huge task. If an update were missed, you would have a task manager record existing with two different phone elements. This would be an update anomaly.

The following rules must be met to normalize our data table to 3NF:

- The table must already be in 2NF.
- There must be no transitive dependencies.

To bring this table into the 3NF, we have to break the table into smaller tables where all non-key columns depend only upon the primary key. Again, we have to determine which columns determine the values in other columns, and create a table for those columns. If necessary, we'll have to add a foreign key, or a junction table. The table should have no columns that depend on any column other than the primary key. After applying these techniques, our new tables should look something like the following:

Projects		
*Task_Num	Task_Title	Task_Mgr
1	9302 Report	Smith

Projects		
2	Continuity Book	Jones
3	Analysis Webpage	West
4	F–16 Analysis	Gordon
5	Health of Fleet	Jones
6	Training Monitor	West
7	Mx History DB	Smith

Projects	
***Task_Mgr**	**Phone**
Smith	2756
Jones	2954
West	2846
Gordon	3102

Notice there are no repeated values when primary keys are used and each row of fields are dependent on only one unique primary key (not part of the primary key), and all the columns are dependent on the primary key, and only the primary key. No foreign key was necessary here.

Having your tables normalized to the 3NF eliminates redundant data, saving space and reducing data manipulation anomalies. Consequently, queries on data will run much faster and more efficient. There are higher forms, such as the Boyce-Codd Normal form and fourth normal form. However, in most cases, the third normal form is sufficient enough to ensure your database is properly normalized.

While complete data normalization is desirable, but in practice you may find that complete normalization may add unwanted complexity to your database. More tables mean more data management and manipulation operations, which can lead to decreased performance. You should strive to find a balance where the first, second, and third normal forms are generally met without creating an unnecessarily complicated database schema.

Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

223. Introduction to data modeling

1. What is a data model?

2. What is the purpose of data model?

3. Define database design.

4. What is the first step in building an effective data model?

5. What should a requirements analysis determine?

224. Creating a data model

1. What are the two dimensions that make up the structure of a relational table?

2. What is the difference between a data item and a data value?

3. Which type of table uses the foreign key, parent or child?

225. Establish relationships

1. What is a data relationship?

2. What two elements define a data relationship?

3. What are the three types of data relationships?

4. What are two advantages of using one-to-many relationships in a relational database?

5. Why are many-to-many relationships not allowed to be used in a relational database?

6. How do you resolve many-to-many relationships?

7. How are the logical relationships in a relational database created and maintained?

8. List the seven properties of relational tables.

9. Why do relational databases use indexed fields?

226. Normalizing data relationships

1. What are four advantages to data normalization?

2. What is the goal of data normalization?

3. Name the four rules for data normalization to 1NF.

4. Define functional dependency.

5. What does 2NF deal with in regards to data relationships?

6. Name the two rules for data normalization to 3NF.

7. What are some advantages to normalizing data to 3NF?

2–4. Relational Data Management System

RDMS stores common access data tables with information used by all ELCs. Earlier, we have shown the general differences between the two kinds of database systems. You have learned the database structure of the DMS and all its characteristics. Now let's discuss some of the specifics of the RDMS. In this section, we will consider using Relational Data Management System tables, their schema, and how to read the schema diagrams.

227. Using Relational Data Management System tables

The RDMS tables used to support the 7R1 schema are broken down into seven categories: History, AEF, Portal, Reference, Security, System Support, and Interface Support. We look at an overview of these tables, their primary functions, and how to use them.

Historical data

The historical data in these tables contain equipment and maintenance information. The tables hold up to five years of data. These tables are broken down further into three areas:

- JDD.
- Status and inventory.
- Automated history (AHE).

Job data documentation

These tables store all the JDD data for each ELC. These tables offer the users up to five years of historical JDD data that can be readily accessed by certain background reports or QLP programs using SQL.

Status and inventory

These tables hold historical status and inventory data that is deleted by NFS120. The data is also for each ELC. These tables offer the users up to five years of historical status and inventory data that can be readily accessed by certain background reports or QLP programs using SQL.

Automated history

These tables hold AHE data that is deleted when the equipment is put in to permanent transfer status. The data is also for each ELC. The tables hold up to five years' worth of previous AHE data that can be accessed by certain background reports or QLP programs using SQL. This way the next base that gains the equipment will be able to utilize the RDMS to have immediate access to this historical data.

Air & Space Expeditionary Force tables

These tables provide the necessary structure to load AEFs to the database and assign equipment and personnel to the AEFs. This data is then used to provide some limited AEF reports. As the AEF organization and mission changes, more functionality is added to subsequent maintenance releases.

Portal tables

Portal tables hold all data necessary to populate the Air Force portal Fleet Asset Status gadget. The use of these tables has replaced the current process of running individual files for database and sending them to the Portal for display. Instead, the Portal connects to the CDB and grabs the information it needs from these tables.

Reference tables

Reference tables hold all REMIS supplied reference data. This eliminates the duplication of reference tables on every database.

Security tables

Security tables are used to control access to the IMDS CDB, including the different levels of access. These tables replace those records previously stored in security area (area 7) of the DMS database. Security tables hold information pertaining to terminal IDs, TRIC codes and their options, user IDs, profiles, profile deltas, and ELCs (or unit data, the previous unit variable record [UVR] records).

System support tables

System support tables hold information needed by the system, such as data about background runs, print products, reject codes and narratives, and GUI GAG messages.

Interface support tables

Interface support tables contains records needed for IMDS CDB to establish connections with different the interfaces such as the SBSS, Enhanced Maintenance Operations Center (EMOC), and Integrated Maintenance Information System (IMIS). Some interfaces may not use all the tables in this area.

228. Relational Data Management System schema

We learned earlier about the general structure of a relational database. We have mentioned that the RDMS consists primarily of tables. We look at how the 7R1 RDMS schema is organized and describe

what the tables look like. Remember, the schema is only a description of the database. Again, we look at the characteristics and different sections of the RDMS database.

RDMS schema book

The schema can be viewed through the online IMDS database help system. This is the same source for viewing the DMS schema we learned from previous lessons. It was created to be viewed a page (webpage) at a time. You can find the tables through the menu under "Schema Layout." The selection is called the "7R1 RDMS Schema Book." The RDMS schema is composed of tables arranged alphabetically. A complete list of the tables can be found when you select "RDMS Table Definitions."

Understanding the RDMS table

When you start using the RDMS tables, you go to a screen that shows pertinent information about the table. We first go to the page layout that describes the different parts of the page. Then we go to the table syntax. Figure 2–19 shows the page containing the "Reject_Code" table. Look at the figure as we describe the different parts.

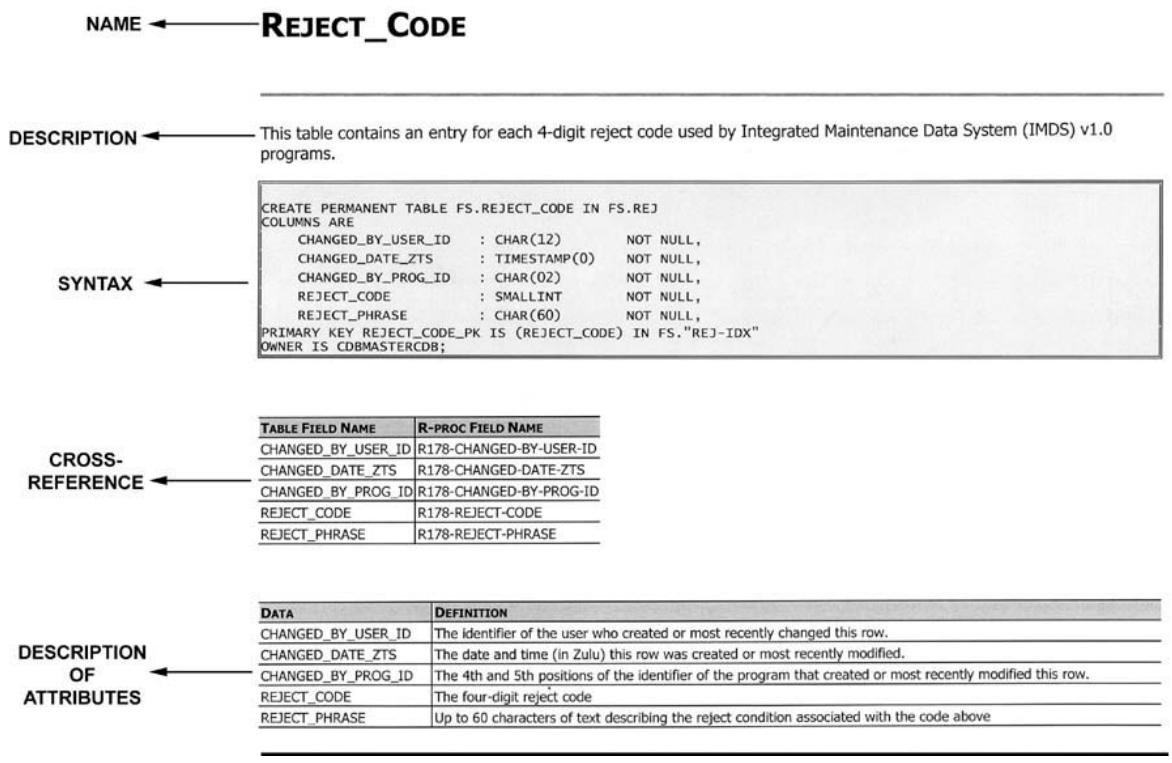


Figure 2–19. Reject code table.

Table name and description

The top part of the page is the table name. Right below it is the general description of the table and its purpose.

Table syntax

The box below the table description contains the different elements of the table itself—the table syntax with the actual column fields and key specifications.

Cross-reference

Next is the table cross reference chart used mainly by SQL programmers. The column on the left (TABLE FIELD NAME) relates to the table attributes. The column on the right (R-PROC FIELD NAME, also known as the "R-PROCs") relates to the SQL procedures used by programmers to access the table columns. When there are three columns in the cross-reference section, the middle column (REMIS MOA [I-PROC] FIELD NAME) is cross referenced to the REMIS interface files, which are also in table format. Some of these RDMS reference tables are populated by REMIS data.

Column description

The bottom section is the description of the column attributes. The column on the left (DATA) relates again to the table attributes. The column on the right describes the values allowed for the entries.

The format for all the table pages are organized this way. Some are large tables, which can take many pages to scroll though. Some tables are small (such as the REJECT_CODE table) enough to contain enough data for reference use.

Table syntax

A relational table is built in SQL using the CREATE TABLE statement. A table follows a syntax or format in the way it is structured. We show you first a table from the 7R1 RDMS as an example (fig. 2–20). The SQL statements are in bold. As we go through the syntax, we will explain each primary clause that describes the statement.

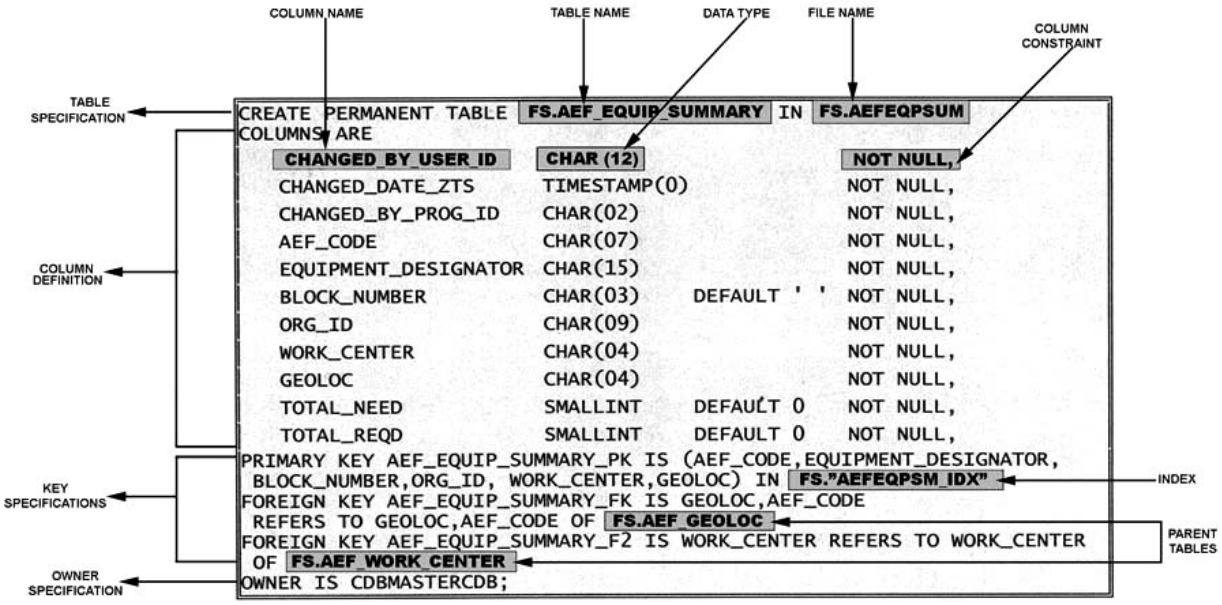


Figure 2–20. Table structure.

Syntax statement	Description of clause and keywords
CREATE PERMANENT TABLE FS.AEF_EQUIP_SUMMARY IN FS.AEFEQPSUM	This is the *table-specification*. FS.AEF_EQUIP_SUMMARY is the table name. It follows the CREATE TABLE command. PERMANENT is an optional clause. FS is the qualifier filename that refers to the IMDS 7R1 schema, or simply the schema name. This is the *storage-specification* where the table is stored in the master file directory. Storage specification uses EXEC files; thus, FS.AEFEQPSUM is an EXEC filename. The EXEC qualifier "FS" usually follows the schema name (FS).

Syntax statement	Description of clause and keywords
COLUMNS ARE CHANGED_BY_USER_ID CHAR (12) NOT NULL	This is the *column-definition.* These are the name, data type, size or number of characters, and scale of each column in the table. Follows the statement COLUMNS ARE. TIMESTAMP is the absolute value of the date and time in the format: yyyy-mm-dd hh:mm:ss. In this particular table, the date and time is not required. NOT NULL means that the field must have a value assigned to it when data is entered. SMALLINT is used when there is a minimum and maximum numeric value assigned. In this case, zero is the minimum number.
PRIMARY KEY AEF_EQUIP_SUMMARY_PK IS (AEF_CODE,...etc.) IN FS."AEFEQPSM-IDX"	This is the *primary-key-specification* The primary key(s) for the table are within the parenthesis, known as the *sort-column-list*. The RDMS tables are indexed to allow for faster search and retrieval. The index files are assigned to store primary (for the primary keys) and secondary index (columns used other than primary keys). When the keyword INDEX is used on the statement, another column name will be specified. In this particular table there is no indexed column specified.
FOREIGN KEY AEF_EQUIP_SUMMARY_FK IS GEOLOC, AEF_CODE REFERS TO GEOLOC, AEF_CODE OF FS.AEF_GEOLOC	This is called the *standard-foreign-key-specification.* Foreign keys relate these columns, known as the *column-name-list*, by their column names from this table to its parent table. GEOLOC and AEF_CODE of this table (FS.AEF_EQUIP_SUMMARY) has the same data as the GEOLOC and AEF_CODE columns of the parent table, FS.AEF_GEOLOC. The columns from child and parent tables must correspond one-to-one. Each FOREIGN KEY statement refers to one parent table. Since this table has two FOREIGN KEY statements, it relates to two different parent tables.
OWNER IS CDBMASTERCDB	This is the *owner-specification.* For programmers' use only. Identifies the user ID of the owner/creator of the table. "CDBMASTERCDB" is the user ID of the administrator of the 7R1 RDMS (AFLCMC -Gunter).

229. Reading Relational Data Management System schema diagrams

The 7R1 RDMS schema diagrams can be viewed on the online IMDS database help system. In this lesson, we describe the different parts of the schema diagrams and how they relate to one another.

RDMS areas

Earlier, we mentioned that there are seven categories of tables: History, AEF, Portal, Reference, Security, and System Support. However, the RDMS areas on the online schema diagram are slightly organized in a different manner. The Portal tables are combined in one area with aircraft, engine, and part information tables to avoid duplication of data across the different ELCs. This is known as Equipment Support. The seven areas on the RDMS schema, as arranged on the menu, are the following:

- AEF.
- Equipment Support.
- History (Sub-areas are automated history, JDD history and status and utilization history).

- Interface Support Area.
- Reference.
- Security.
- System Support.

Figure 2–21 shows a screen snapshot of the schema diagram drop-down menu showing the RDMS areas with the AEF area selected.

Table box

The tables that comprise an RDMS area are drawn in boxes to show relationship to each other. It is more or less like a block diagram. This gives us a quick overview of related and non-related tables. We use some areas to describe the different characteristics found in the diagram. We start with the AEF area page, as shown in figure 2–22.

Each box represents a table. The upper part, in black shade, shows the name of the table. The lower part, in a lighter color (the actual color is yellow), contains one or more primary keys for that table. Only the primary keys are shown on the box.

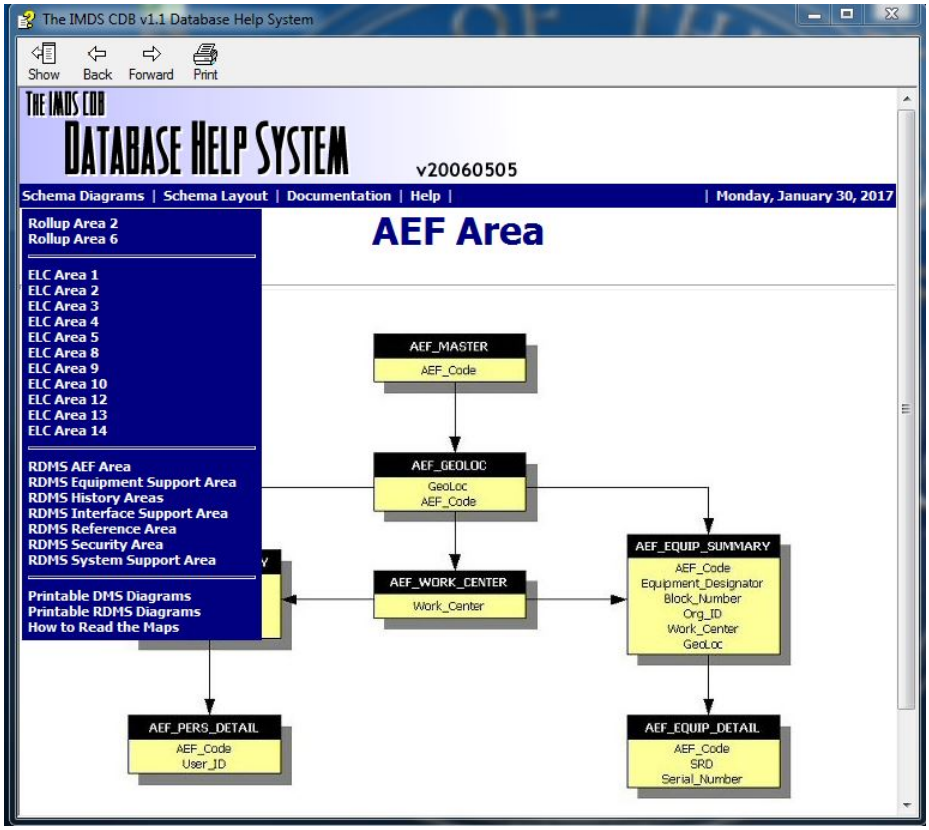


Figure 2–21. Relational Data Management System drop-down menu.

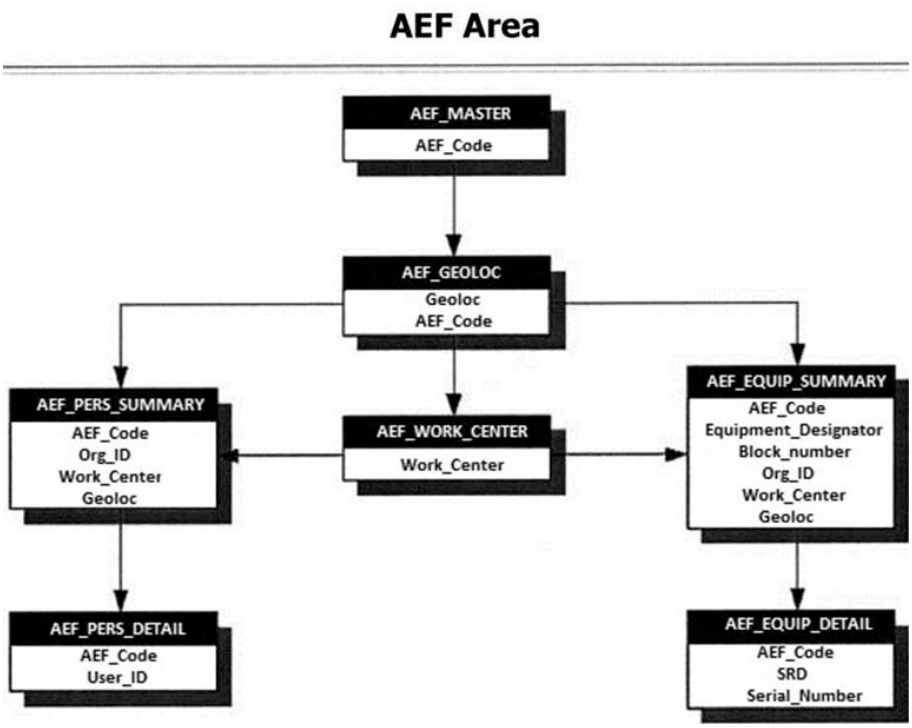


Figure 2–22. Air & Space Expeditionary Force area tables.

Defined foreign key

Solid arrow lines, connecting two tables, show that a defined foreign key relationship exists—from parent table to child table. The arrow points to the table with a defined foreign key or keys. That means at least one foreign key is specified in the table schema of the child table. The foreign keys are not shown on the box but are defined in the schema. Some of the primary keys may also act as a foreign key to its corresponding parent table. The parent table is the one from which the arrow directly points.

The AEF area table shows the AEF_GEOLOC and AEF_WORK_CENTER tables are the parent tables of the AEF_MASTER. The AEF_MASTER table is ***not*** a parent table of the AEF_EQUIP_SUMMARY table. If you recall from the previous lesson on the table schema, the foreign keys defined in the AEF_EQUIP_SUMMARY table points the keys to its parent table. The parent table is clearly visible when an arrow is shown. You still have to look up the defined foreign key in the table schema.

Implied foreign key

Some tables show a broken arrow (dotted line) between them. A broken arrow shows an *implied* foreign key relationship between the tables. This means that the table schema does not mention a foreign key (also known as defined), but a foreign key exists for the SQL programmers to use. This is the case of the Utilization History tables (upper portion) in the Status and Utilization History sub-area. Figure 2–23 is an extract taken from that area. Notice that some of the primary keys between two tables are the same. These keys can be used as a foreign key to link the child table to the parent table. However, if you pick one of the tables, let's say UTILIZATION, and look at the table schema, there is no foreign key specified, although from the area diagram we know a foreign key exists that links to the parent table (UTILIZATION_DEVIATION_XREF table).

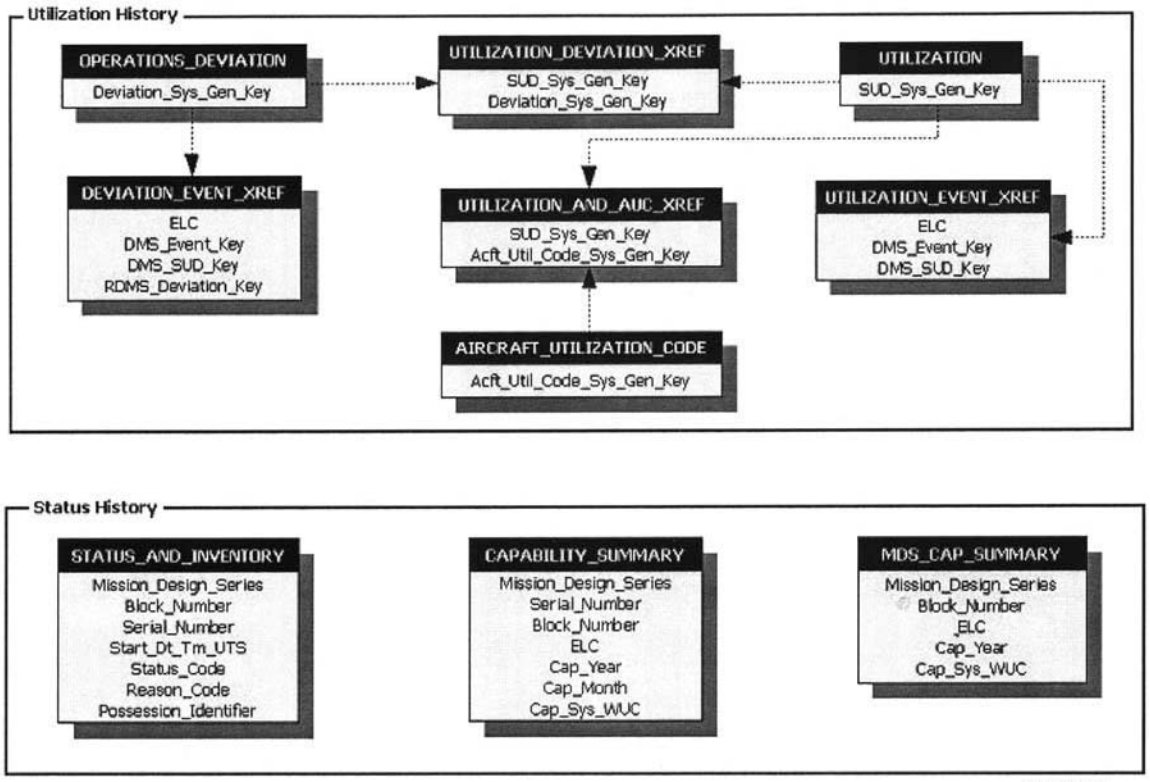


Figure 2–23. Utilization and status history tables.

Table groupings

Borders around tables show table groupings. This occurs in an area where certain tables are not many enough to warrant their own category or area confusing. This is the case of the tables that compose the Equipment Support Area, as shown in figure 2–24. You can see that there are two borders containing tables. These tables are grouped together within a solid line border under a heading that generally describes the table contents.

These two groups within the Equipment Support area are as shown in the following table:

Heading Title	Tables
Part Number Integrity	AWAITING_ITEM_MANAGER ENTERPRISE_PN_SN DUPLICATE_PN_SN
Air Force Portal	AIRCRAFT_DOC_NUM AIRCRAFT_MAINTENANCE AIRCRAFT_STATUS

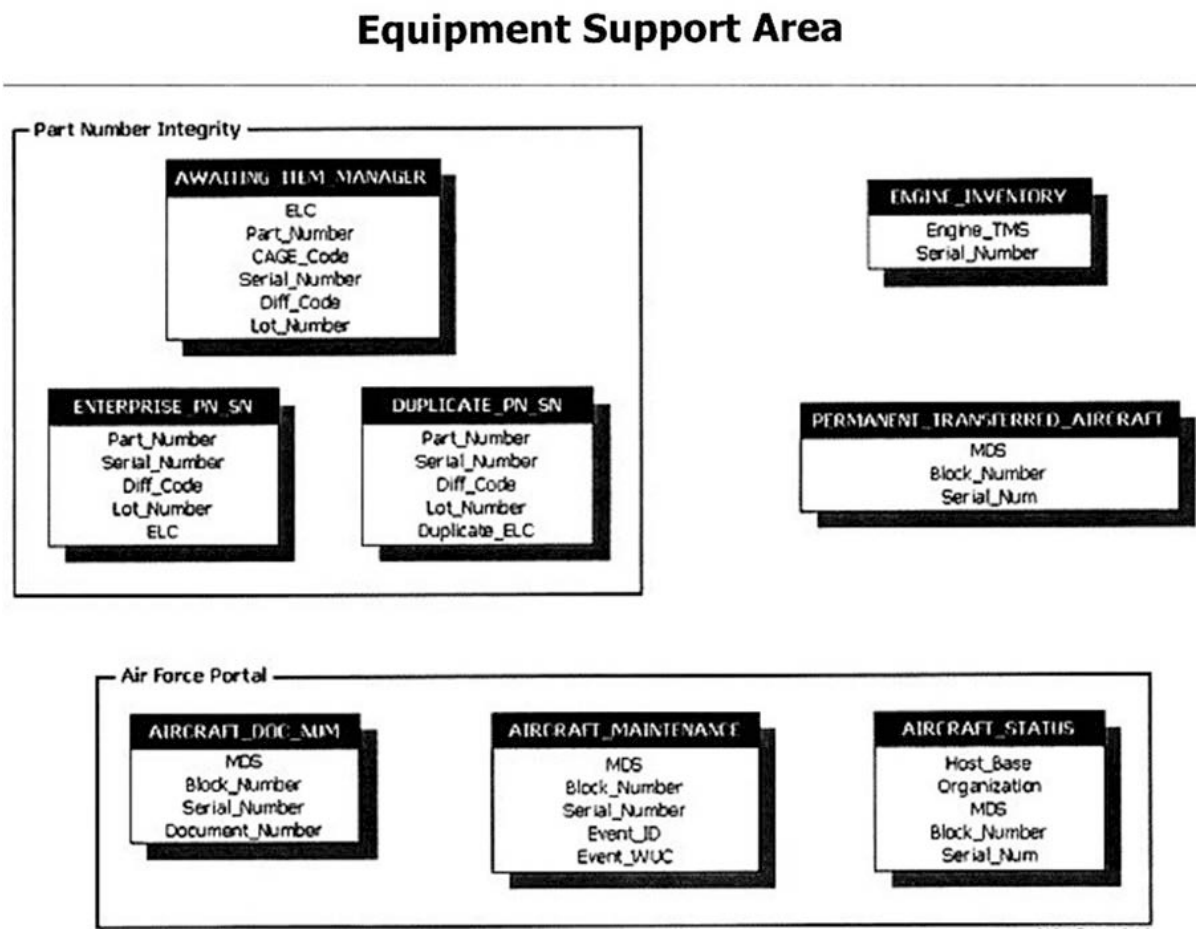


Figure 2–24. Equipment support area.

Independent tables

When there are no borders enclosing a table or group of tables, those tables are considered independent tables. Tables with no arrows between them are also considered independent. On the equipment support table, the independent tables are the ENGINE_INVENTORY and PERMANENT_TRANSFERRED_AIRCRAFT tables.

Expansion of areas

As new tables are created, more are included within the appropriate table grouping. Therefore, the area diagrams can expand with additional tables.

Link to table schema

The table box also serves another function. Each table box on the area diagram serves as a hyperlink to the table schema. By placing your cursor over the box and "clicking" your mouse, the page jumps to the table schema for that specific table.

Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

227. Using Relational Data Management System tables

1. List the seven categories of the 7R1 RDMS tables.

2. How many years of JDD data can the historical tables hold?

3. In what tables can you find REMIS supplied reference data?

4. In what tables can you find information such as background runs and print products?

228. Relational Data Management System schema

1. How are the tables in the RDMS schema arranged?

2. Who are the primary users of the table cross reference chart?

3. What SQL statement is used to build a relational table?

4. What type of keys do the primary index files store?

229. Reading Relational Data Management System schema diagrams

1. Describe the two parts of a table box.

2. What does it mean when a solid arrow line connects two tables?

3. How would you know if an implied foreign key exists?

4. What is another function of the table box on the area diagram?

2–5. System Security

Security is an extremely important issue when it comes to computer software, especially databases. Since a database can contain sensitive or classified information, you must take steps to restrict and control access to it. IMDS is a sensitive but unclassified (SBU) information system because it contains sensitive information about personnel as well as weapon systems status and condition data. There are programs exclusively used by functional users who protect the data from manipulation or corruption. TRIC security is applied in these cases to allow authorized users.

Make sure that only people with a valid need for this data are allowed access privileges. All security lies within the RDMS in 7R1. The security subsystem in IMDS provides a single sign-on capability that is based on a role based access clearance (RBAC). This section discusses TRIC security, user ID access, and user profiles. The lessons cover the program screens and their features: purpose, description, and available options. They also cover their unique functions.

230. Transaction identification code security

In the 7R1 environment, when a user is loaded into the database, that individual has access to all programs except for those programs that are restricted. In the 7R1 environment, security is based on permissions instead of restrictions. Users only have access to those programs for which they have been granted permission. These programs are identified by the use of TRICs. Within this lesson, we will discuss TRIC access, user profiles, security inquiry, and their descriptions. You need to understand TRIC security in order to know the level and type of access you can grant to users. As a DBM, there will be several programs you will use on a consistent basis to maintain a high level of security for your IMDS database.

Understanding transaction identification code access

For you to understand how TRIC security works within IMDS, you need to know the extent of programs TRICS control and how TRIC access is controlled by IMDS through the RDMS security tables.

TRIC and programs

A TRIC is a three-character code that the computer uses to determine the program or routine requested. The computer we are referring to is the mainframe that houses IMDS. It is like a key that the utility program uses to go to a program in IMDS and run it. IMDS programs have TRICs used to identify both online as well as background programs.

A TRIC can have one or more programs associated with it. With one TRIC, you can run these programs through their respective screens. For example, the TRIC MIK controls five programs (screens 631, 632, 633, 635, 636, 992), whereas, the TRIC PRB controls only one program (screen 897). However, the TRICs have two levels of access, referred to as the main options and sub-options. Through these TRIC levels, access to screens within the TRIC can be also denied.

Master transaction and transidentification option tables

TRIC security in 7R1 involves two main tables. The first is the Master_Transaction table, while the second is the Trans_ID_Option table.

Master_Transaction table

This table has the transaction, whether it is enabled or not; the columns are for the “main” and “sub” options (if applicable) in the program.

Trans_ID_Option table

This table lists all of the acceptable option/sub-option combinations for each transaction. It is the table that is allowing us to change from a “you can do everything except what we say you can’t” process to a “you can do nothing except what we say you can” process.

Cooperation between tables

As a result, the application program is not executed if the option/sub-option combination does not match an entry in this table. If a valid TRIC option/sub-option is not in this table, it rejects. Think of the Master_Transaction table as the “where,” and the Trans_ID_Option table as the “what.”

Screen transaction reject

If a screen has type transactions of (L)oad, (C)hange, (D)elete, and (I)nquiry, and a blank type transaction is transmitted, you get the security option reject (5136 - “You are not authorized this transaction ID or option”). This is a reject from the security tables, not from the program. What this means is that you are not allowed to run what was just transmitted. Since certain programs have option security defined and a blank option is valid, the security layer cannot give a different reject for a blank option. If there is not a blank in the Trans_ID_Option table, it is not allowed.

Determining user access to TRICs

When you are trying to decipher what options you need to add or deny from a user ID, screen 633 and screen 635 both show exactly what options are valid for a particular TRIC. The Inquiry option is the only available option at ELC level for screen 633 and 635 (Trans_ID_Option and Master_Transaction tables). That is because these tables sit in the RDMS and apply to every IMDS database (the ELCs).

TRICs and user profiles

In 7R1, individuals are assigned profiles that control their TRIC permissions. A profile is a list of TRICs to which the user is allowed access based on the user’s role (job). Since there are hundreds of programs in IMDS, TRICs are predefined to profiles. When a user is loaded in IMDS to have access to certain TRICs, the programs tied to those can be run. However, there are still certain privileges within the programs, which we call scenarios.

There are four possible scenarios or ways a user can have access to TRICs. These scenarios are based initially on the user profile plus additional access or restriction to the options associated with the TRIC.

Scenario	Explanation
TRIC allowed	TRIC is allowed. The individual can process the TRIC based on profile or when authorized in addition to the profile.
TRIC not allowed	Entire TRIC is restricted. The individual cannot process the TRIC at all, either not part of the user’s profile or restricted from the TRIC.

Scenario	Explanation
All options allowed	All options of the TRIC are allowed. The individual can process the TRIC based on profile or when authorized in addition to the profile.
Some options allowed	Some options of the TRIC are allowed. The individual can process specific options only based on profile or when restricted.

For example, if a user only has permissions for TRIC CME, options 4 and 5, and TRIC CUP, these are the only TRICs and options they are allowed to perform.

Likewise, you, as the DBM, may not run screen 636 even though you have access to TRIC MIK. This screen is for the exclusive use of the programmers at the AFLCMC. The program, TRIC Security Inquiry (screen 896), shows how the different TRIC levels of access are given.

TRIC security inquiry

The TRIC security inquiry program (NFSEQ0, TRIC: SEC, screen 896) inquires the Allowed_Profile and the User_Profile delta tables to see what access user IDs have been granted. The online TRIC security inquiry lists all authorized users for a TRIC and its options for a specific user ID and/or TRIC. User ID or TRIC must be entered to prevent GAG-file limit being exceeded. This program shows everything the user ID has access to. Either user ID or a TRIC must be input.

Description

This online TRIC security inquiry lists all authorized users for a TRIC and its options for a specific user ID and/or TRIC. User ID or TRIC must be entered to prevent GAG-file limit being blown. This program shows everything the user ID has access to. Either user ID or a TRIC must be input.

The output format for this program displays the following information:

- If only the user ID is entered, NFSEQ0 lists every TRIC, main, and/or sub-options to which the user ID has access.
- If the user ID, TRIC, main, and/or sub-options are entered, NFSEQ0 lists whether or not the user ID has access to that TRIC, main, and/or sub-options combination.
- If the user ID, TRIC and */* are entered, NFSEQ0 lists whether or not the user ID has access to that TRIC and any main/sub-option combination.
- If the TRIC and MAIN/SUB options are entered, NFSEQ0 lists every user ID that has access to that TRIC main/sub-option.
- If the TRIC and */* are entered, NFSEQ0 lists every user ID that has access to that TRIC and every main/sub-option combination.

231. User identification and profiles

In the previous lesson, we talked about TRIC security. Users only have access to those programs, through TRIC access, for which they have been granted permission. When a user is given access to IMDS, that individual is assigned a user ID. Along with a password, access to IMDS programs is given to that user ID through profiles. Security is therefore based upon user ID. Let’s talk about how user ID and profiles are assigned.

Profiles for individual roles

We mentioned earlier that the security subsystem provides a single sign-on capability that is based on a RBAC. You have the capability to move from one ELC to another without signing off and signing back on. With one User ID, you can access your own ELCs database plus the other ELCs, too. The enterprise recognizes your user ID and allows you the appropriate access.

General characteristics of a profile

User profiles are created using IMDS TRIC code security. User profiles utilize the least privilege principle (i.e., users cannot access the system unless they are authorized to do so by a higher office of responsibility). Users only have access to those TRICs main/sub-options for which they have been granted permission to use through the use of master profiles. Every user is given an IMDS master profile named Generic_User_Profile. This profile contains the basic TRICs that all users need. Each profile is assigned a profile ID in IMDS.

Another type of master profile equates to a role that is performed by an individual. There are approximately 38 master profiles in the CDB. New profiles are created as requirements are identified from the field.

Multiple profiles

A single user is normally assigned one profile related to the user's primary job (e.g. crew chief). A user may have multiple profiles assigned to meet the different roles the individual fills. This is related to their job position or additional duties given. Individuals may have up to eight master profiles loaded to them depending on their duties.

For example, consider a user who is a crew chief and has the additional duty of training monitor for his or her work center. This user would most likely be assigned two profiles: Flight Line Technician and Work Center Training Monitor.

Levels of profiles

User profiles govern what ELCs, AEFs and/or global/roll up areas users are allowed to access. Before we go to these profiles, we need to understand the two areas that a profile falls in: functional user and administrators. Functional users refer to those individuals who use IMDS online in their daily transactions related to their mission—whether aircraft maintenance, maintenance training, etc. Administrators are functional users whose main responsibilities are the administration of programs related to IMDS system operation: security access, database management, support of the functional users, etc.

Levels of operation

In the central database, the two levels of operation include enterprise and ELC level. The enterprise level is managed by enterprise database managers at Maxwell AFB, Gunter Annex, Alabama. Host DBMs at the base level manage the operation of the ELC level.

1. Enterprise level—where most programs govern the database operations that affect all ELCs. DBAs normally work at this level. In the CDB, they have access to both the DMS (all ELCs) and the RDMS. The enterprise level is also referred to as the global area.
2. ELC level—where users conduct their daily IMDS processing at their respective bases. The main population of users works at the ELC level. This includes the DBM assigned at that ELC. The AEF function of IMDS is considered ELC level.

Some users (MAJCOM, Air Staff, etc.) only require access to the global/roll up areas and do not require direct access to any ELCs.

Levels of administration

This is a multi-leveled user profile administration. Administrators at each level manage those profiles at their level within their span of control, as well as, and those managed by subordinate administrators. The levels of administration deal primarily with the privileges the user has in running programs that control the profiles. The CDB DBA is responsible for the following:

1. Administers' user access, master profiles, and profiles for users above base level (i.e., Air Staff, MAJCOMs, Numbered Air Force [NAF], etc.).
2. Creates and modifies master profiles.

3. Grant access to rollup area-only users (MAJCOM, Air Staff, etc.).
4. Can manage user access and profiles at all levels.
5. Manages RDMS tables.
6. Performs database global and rollup area saves.
7. Manages all database reloads and recoveries.
8. Runs NDA500 and DMU verifies.
9. Performs database repairs.
10. Grants batch-processing privileges as required.

Executive control language database management

The host DBM assumes the role of the ELC DBM in IMDS. The host DBM administers user access and profiles for their particular ELC, as well as access to their ELC by other ELC users. In addition to administering profiles, the host DBM can copy and modify user profiles within their ELC. The host DBM also manages database saves for their ELC.

Unit database management level

The unit DBM is responsible for administering user access and unit-ID level user profiles for his or her particular unit. When authorized by the ELC DBM, the unit DBM can also copy and modify user profiles for use in their unit. This level of administration is the lowest level of profile administration and is optional.

Assigning user identification and profiles

When an individual is loaded in IMDS, he or she is loaded to the home ELC. The individual is initially assigned an employee number. The next step is to create a user ID for the user. The host DBM, as the ELC DBM, is responsible for loading the user ID in IMDS. This function may be delegated, depending upon the authority given by the unit.

There are two types of user IDs that can be loaded: TIP User ID and DEMAND User ID. The TIP user ID is generally loaded to all functional users. The DEMAND User ID is reserved for administrators at the ELC level—the ELC DBM and other users authorized.

Home executive control language employee number/user identification maintenance

The program, Home ELC Employee Number/User ID Maintenance (NFSDW0, TRIC: HES, screen 887), is used for the initial user ID load. The program updates the Home_User and Demand_User tables. NFSDW0 loads, disables/deletes, and inquires Tip and Demand User IDs. It is also used to set the capability for background report processing, restrict to work center, and other unit access for the *home* ELC.

Purpose

This program updates the Home_User and Demand_User tables. NFSDW0 loads, disables/deletes, and inquires Tip and Demand user IDs. It is also used to set the capability for background report processing, restrict to work center, and other unit access for the *home* ELC.

Description

Screen 887 creates or deletes/disables user information. The Home_User table contains a user ID, an ELC, the user's default unit ID within that ELC, a list of other units to which the user is authorized access, and whether the user is authorized administration privileges in the default unit. The table also contains whether the user is authorized to start background runs, and a lists the units, other than the default, on which the user is authorized administration privileges. The application programs, outside of the system-wide security layer, check the administrative privileges and background-starting permissions.

Inquiry

For the inquiry option, enter the user ID or employee number. The program displays all the applicable information stored in the HOME_USER or Demand_User tables.

Load

For the load option, process inquiry option against the employee number first, then load applicable user information. TRIC HES generates a new Tip and Demand User ID for the new user. The Demand User ID is optional. At the time you load a user, the background report processing, restrict to work center, and other unit access switches must be set.

- Background report processing defaults to N. You must change this indicator to Y in order for the user to process Background reports. The user must also possess the background TRIC in either their profile or delta table.
- Restrict to work center is used by program NFSES0, TRIC: USL. This switch allows the DBM to authorize work center supervisors to use TRIC USL. The work center supervisor is only able to authorize individuals within their work center to process certain TRICs.

The user is only authorized to process for the units that are entered in this field. Other units' access was removed from screen 709 and put on screen 887.

A Demand User ID is created if a Y is entered in the Demand field. The Demand User ID is the same as the Tip User ID, but is prefixed with a D. The Demand User ID is stored in the Demand_User table.

Change

The change option is processed when a user has permanently changed station or when the need arises to update what access the user currently has.

Disable

Disable removes all profiles and deltas loaded to a user ID. The disable option also removes the user from any allowed ELCs that they have been authorized. The user ID is still in the Home_User and Demand User Table, but the user will not be able to process any transactions.

232. Profiles related to database management

In 7R1, the creation of the Allowed_ELC, Allowed_Profile, and User_Profile_Delta tables enables the DBM to grant access to any user who has a need to access ELC's other than their own. These tables identify what other ELCs an individual can access and what type of access they have within those allowed ELCs.

Allowed executive control language concept

This concept gives DBMs the ability to allow a user from another base access to the DBM's database without having to be loaded in the database within IMDS. This allows users to get comparative data on like MDSs, document JDD for deployed equipment, and gives MAJCOMs enterprise wide view.

For example, let's say that an analyst at Base A wants to run a comparative report on a like mission design series (MDS) at Base B. Base B's DBM can grant the analyst at Base A an allowed ELC for the data with an inquire only profile.

Let's consider another example to help provide further clarification. In this example, crew chief from Base A deploys to Base B. The Base B DBM can grant the Base A crew chief an allowed ELC for the base that gives that individual the ability to update maintenance actions on Base B's equipment.

In the CDB environment, a user may be allowed to switch from the home ELC to one or more other allowed ELCs. This access is called an allowed ELC. To view those ELCs and access them, you need to know how to use the *ELC straight-line command in TIP.

View allowed enterprise location codes

To pull up a list of ELCs to which your user ID has been granted access, you need to process an inquiry with ***ELCI.** You enter **"*ELCI"** at the bottom entry block of the screen. A screen appears with a list of the ELCs to which you have access. Figure 2–25 shows a sample "Allowed ELC" screen that came up after "*ELCI" was entered.

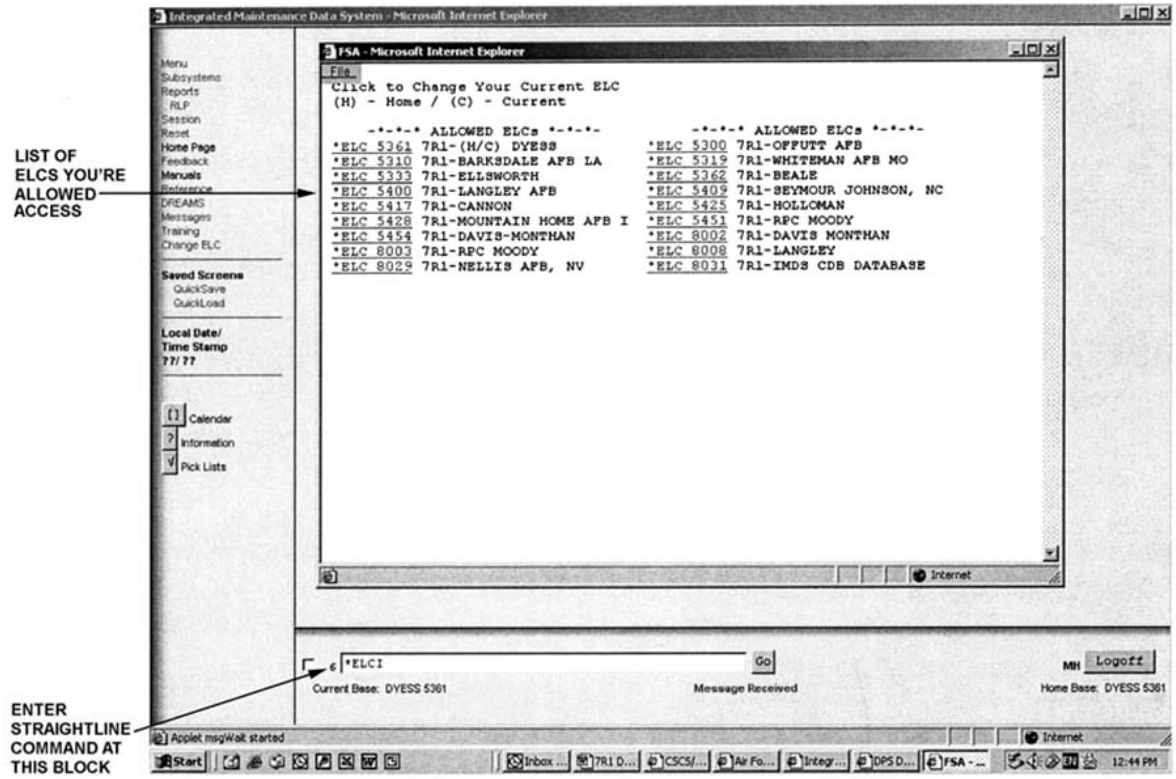


Figure 2–25. Allowed enterprise location code screen.

Switch to an allowed enterprise location codes

In order to change to an allowed ELC from the screen, click on the hyper link of the ELC you wish to go to and the system switches to the database of the ELC you selected. You can also switch to an allowed ELC by entering the following straight-line input:

***ELC ####** (where #### is the ELC to which you want to switch)

Delta tables

Along with the implementation of the master profile concept, delta tables have been designed to augment the master profiles and override what are in a user's master profiles. This process is called creating a *delta*. The delta is the adding or subtracting of a TRIC with its respective options and sub-options out of a loaded profile on an individual. Each user can have a User_Profile_Delta row(s) associated with their user ID. The delta table allows the DBM or work center supervisors to add or remove TRIC main/sub-options from users.

To allow a DBM or work center supervisor the ability to customize a profile loaded to an individual, the new security system allows individual TRICs with their options and sub-options to be restricted under that individual.

For example, let's say the DBM loads the flight line technician profile to a new crew chief, but does not want that individual to be able to delete JCNs. The DBM or work center supervisor would then create subtraction deltas for TRIC CME options 6, 7, and 8.

Master Profile Manager

This Master Profile Manager program (NFSES0, TRIC USL, screen 902) inquiries, loads, and deletes TRIC(s) and their options (delta's), either by user ID or a work center grouping.

Description

Deltas are new in the 7R1 security subsystem. The deltas augment the Master Profile that is loaded to a user ID. TRIC, main, and/or sub-options may be plused or minused to or from a user ID. For example, if the user has Master Profile "Database Manager" and that profile has all options of TRIC CME within it, and you would like to restrict the user from processing CME option 6, you would minus TRIC CME option 6 from that user ID.

Inquiry

The inquiry option displays all TRICs, main, and/or sub-options that have been plused or minused to or from a user ID.

Load

The load option pluses or minuses specified TRICs, main, and/or sub-options to or from a user ID.

Delete

The delete option removes the specified TRICs, main, and/or sub-options from a user ID.

Other profile privileges

In addition, a profile can be given other privileges. To switch units, for example, the *UNIT command has been added, which operates like the *ELC command, and has two options.

View allowed units

To obtain a list of authorized units to which your user ID has access, you must process *UNITI. For example, enter **"UNITI**" at the bottom entry block of the screen, the same way you would when you want to view your allowed ELCs. This returns a list of all available units you have access to in your current ELC. To change units, just click on the hyperlink for the unit you want to view.

Change to another authorized unit identification

You can use the process described above to change the unit ID you are on or you can type in the following command:

***UNIT ?** (Where "?" equals any unit ID you are authorized to process on.)

Special program privileges

There are three special program privileges that may be granted to users. The CDB DBM normally grants these privileges to the ELC users, other than the ELC DBM. These privileges relate to processing (1) in file update mode, (2) green screen transactions, and (3) background products.

File update

In 7R1 you are no longer able to use screen 891 to switch processing to another unit. This screen is restricted to only those individuals with TRIC ADS in their profile and can only be used to put the system in FUD mode, turn on the timer, and bring the ICI up.

When you grant permission to process while the system is in FUD mode, you must identify those user IDs you want to give access to process in FUD. Once this is done, you can use screen 992 to load those user IDs to process in FUD.

Green screens

With the Air Force directing the mandatory use of the GUI screens of IMDS, a requirement was initiated to control those users who are allowed to process in green screen. Green screens refer to those systems still utilizing a text-based screen transaction using green text and black background screens. Some of these systems are the Comprehensive Engine Management System (CEMS) and the

Reliability, Availability and Maintainability for Pod (RAMPOD) System. A program was created for the 7R1 to specifically identify those users of other maintenance AISs (such as CEMS and RAMPOD) who are granted permission to process green screen transactions. DBMs may also be granted permission to process in green screen, but this should only be used to troubleshoot problems for the authorized users just mentioned. Use screen 992 to load the user IDs of those who are approved for green screen processing.

Background processing

The number of people authorized to run background programs on your databases is a part of the profile based security. Those backgrounds needed for a certain job are included in the profile designed for it. This alone though does not give the person the ability to process the background programs. They also have to have the “Can Generate Reports” switch set on screen 887 for a user on your database or screen 681 for an allowed ELC user.

Master instruction card file update/”green screen” authorizations

The master instruction card FUD/“green screen” authorizations (NFSZ70, TRIC: MIK, screen 992) provides part of the program updates to the Home_User Table. This screen allows the DBM to grant authorization to process in FUD and authorize green screens access by user ID.

Description

This section of the program maintains access to green screens for our screen scrape interface partners and allows the green screen capability for a selected group of users.

Load

The load option loads user IDs that are authorized to put the system into FUD and loads user IDs that are authorized to process green screens.

Delete

The delete option removes user ID authorization to put the system in FUD and delete user ID authorization to process green screen authorization.

Print

Print user IDs with FUD or green screen authorizations. If there are more than 55 user ID, this comes back as traffic, sorted by user ID. If there are 55 or fewer, it is returned in the screen.

Controlling the functional system analyzer

With all ELCs processing under the same FSA, it was necessary to develop a way to bring down the FSA for a single ELC at a time for the running of saves, NDA500s, and so forth. To do this, two new options were added to TRIC TIS. TISXU brings up the FSA for the ELC it is processed on while TISXD brings down the FSA for the ELC it is processed.

233. Profiles management

In the preceding lessons about TRIC security and user profiles, we mentioned several screens that relate to TRICs and profiles. You, as the DBM, must be familiar with several more programs. We begin with the security subsystem menu, which lists the screens related to security.

Security subsystem menu

The security subsystem menu, or screen 38, allows users to easily and quickly access the security subsystem. The menu lists all applicable security subsystem programs. The programs listed on the menu screen can be accessed by clicking on the link as well as typing the screen number on the command line. The following lists the options available for the security subsystem.

Purpose

The purpose of screen 38 is to allow users to quickly access the security subsystem screens.

Description

Menu screen 38 lists all applicable security subsystem programs. The options of the screen are provided in the following table:

Screen	Program Title	Program ID	TRIC
887	Home ELC Employee Number/User ID Maintenance	NFSDW0	HES
681	Allowed ELC Employee Number/User ID Maintenance	NFSDX0	AES
901	Master Profile Manager	NFSES0	USL
871	Master Profile Manager (Copy/Merge Profile)	NFSES0	USL
897	Master Profile Report (Background)	NFS3R0	PRB
886	ELC Profile Manager by User ID	NFSEG0	EPU
882	ELC Profile Manager by Profile ID	NFSEH0	EPP
902	Delta Table Manager	NFSES0	USL
896	TRIC Security Inquiry	NFSEQ0	SEC
872	Delta Table Report (Background)	NFS3T0	DTB
635	Master Transaction Maintenance	NFSZ70	MIK
633	Transaction ID/Option Maintenance	NFSZ70	MIK
632	Transaction On/Off For ELC/Unit	NFSZ70	MIK
631	Device Maintenance	NFSZ70	MIK
636	Enterprise Settings	NFSZ70	MIK
992	Master Instruction Card FUD/"GREEN SCREEN" Authorizations	NFSZ70	MIK

Master transaction maintenance

The master transaction maintenance program (NFSZ70, TRIC: MIK, screen 635), updates the Master_Transaction table. This table is a reference table of all authorized transactions (TRIC) within IMDS. This screen maintains enterprise-wide configuration settings for IMDS.

Description

This screen identifies the transaction ID, its status (enabled/disabled), whether the transaction is a background run, whether security may be specified down to the option/sub-option level or not, the run priority (for background runs), whether the transaction is capable of being run through (PSEUDO processor for batch) PSUPRB (for online transactions), and whether the transaction is subject to security constraints. Additionally, for transactions that are able to be restricted, it holds the column of the option (which is always assumed to be one character) and the starting column and length of the sub-option.

Inquiry

The inquire option lists the settings and other information for a particular TRIC. To list all transactions for a particular TRIC, as well as a brief summary of the settings for each, simply enter "ALL." This option is useful to find all the options for a TRIC.

Load

The load options loads a new TRIC to the Master_Transaction table. This option is used by the enterprise database manager to create new programs for use IMDS.

Change

The change option of TRIC MIK changes information about a TRIC in the Master_Transaction table. This option will only be processed by the enterprise database manager.

Delete

MIK's delete option deletes a TRIC from the Master_Transaction table. As is the case with the change option, this option of MIK is only authorized for the enterprise database manager.

Transaction identification/option maintenance

The Transaction ID/Option Maintenance (NFSZ70, TRIC: MIK, screen 633) is part of the program that updates the Trans_ID_Option table. This screen maintains enterprise-wide configuration settings for IMDS.

Description

This screen is a reference that lists all possible combinations of options and sub-options, including a short narrative as to what this option combination does, for a TRIC.

Inquires

The inquiry option of screen 633 is used to inquire and display all possible combinations of options and sub-options, including a short narrative.

Loads

The load option of screen 633 loads a TRIC and all possible combinations of options and sub-options, including the short narrative to the Trans_ID_Option table. This option of MIK is not authorized for the whole enterprise.

Delete

The delete option on this screen processes several transactions. This option of MIK is not authorized for the whole enterprise. This option removes the TRIC from the Trans_ID_Option table. It also deletes any occurrences of the deleted TRIC, including any options/sub-options that are in any master profiles, and displays a list of how many of each the program deleted. The delete option also removes the delta from any user IDs that possess this TRIC, option/sub-option, and displays how many of each were deleted. This option of MIK is not authorized for the whole enterprise.

Transaction on/off for enterprise location code/unit

The transaction on/off for ELC/unit (NFSZ70, TRIC: MIK, screen 632) part of the program updates the Trans_ID_Off Table. This table enables TRICs to be turned off at the ELC/unit level.

Description

This screen consists of a transaction ID, an ELC, and a list of up to 24 unit IDs. If a transaction ID is in this table for a given ELC and unit ID, the transaction is not allowed to process. If the list of unit IDs is blank, the transaction ID is off for the entire ELC.

Inquiry

The inquiry option of screen 632 inquires and displays whether a TRIC is on or off for a particular ELC.

Change

The change option changes the on/off status for a TRIC for a particular ELC. If you leave all the unit fields blank, it's disabled for all units. If you put in a unit ID, it's only turned off for that unit.

Device maintenance

The device maintenance screen (NFSZ70, TRIC: MIK, screen 631) provides for inquiries, loads, changes, and deletion of data in the Master_Device table. This screen maintains enterprise-wide configuration settings for IMDS.

The table is a reference of all authorized devices within the enterprise. It consists of the device ID and the device type; the ELC, unit, organization, and work center to which the device is assigned; a 25-character space for a device narrative; and, for terminal-type devices, the maximum input time between inputs before a new login is forced. It also includes the alternate device ID for this device (if it is unavailable), and the device ID of the default forms printer.

Inquiry

Assigned organization, ELC, and PID are only returned on inquiries. For an inquiry, enter either device ID or remote ID.

Load

For a load, all fields marked with * must be input (remote IDs are mandatory for type T (terminal) devices).

Change

Changes are only made within fields that are filled in. All changes apply to the input device ID.

Delete

For a delete, enter only the device ID.

Device identification process

Device identification, or terminal IDs, are issued differently with the new centralized database. You, as a host DBM, load the terminal ID to this screen (631). You must load the location, work enter, and so forth. Terminal IDs are assigned at the work center level. Loading the terminal IDs does not actually grant access to the individual; for access, the following must also happen:

- DISA (Defense Information Systems Agency) must be notified.
- There are two files on the system that control what devices have access to the system. 0FS00000*AFS1RCUNDD10 is the SILAS PID file, and WEBTS$*TTW-CONFIG.PIDS is the WebTS PID file. When a device is loaded and we notify DISA, they can do a dynamic change to the SILAS and WebTS configuration.

NOTE: If either of these processes stop and restart, the dynamic changes are lost. That is where NFS1R0 comes in; NFS1R0 reads the Master_Device table and rebuilds a complete configuration in these two files. That way, if SILAS or WebTS are restarted, the additions or deletions that had been added dynamically are now part of the master configuration.

Currently, there is no naming convention with the terminal IDs. The system will accept anything the DBM enters as long as it is unique and alphanumeric. Although there is no naming restriction within IMDS, DISA STIG still has restrictions. We recommend that you use the same naming convention as we use today (*e.g.*, a terminal ID at Luke, AFB will look similar to: LU0001).

Enterprise settings

The enterprise settings (NFSZ70, TRIC MIK, screen 636) part of the program updates the FSA_Control table. This screen maintains enterprise-wide configuration settings for IMDS.

Description

The FSA_Control table identifies whether the system is up or down, whether the DMS rollup area is available for transaction processing, and whether ICI is available. It also counts the number of times the system has been brought up. Additionally, there are three fields relating to background processing. The first limits the number of background runs a given ELC/unit may have processing at any given time (a zero indicates no restriction by unit). The second limits the number of background runs a given ELC may have processing at any given time (a zero indicates no restriction by ELC). If a background run is requested, and either of these two limits is exceeded, the run will be queued, and started based on run priority. The third limits the number of background runs that may be processing

enterprise-wide (a zero in this field disables background processing and informs the user of this unavailability).

Inquiry

An inquiry shows all enterprise settings for maximum number of background runs that can be active for each unit, each ELC, and enterprise wide; GUI dynamic XML element purge threshold; background report purge threshold; and history parameters.

Change

Changes affect the whole enterprise. This option of MIK is not authorized for the whole enterprise.

Allowed enterprise location code Employee Number/User Identification maintenance

The allowed ELC Employee Number/User ID maintenance (NFSDX0, TRIC: AES, screen 681) program updates the Allowed_ELC Table. This program allows users to access ELC's other than their home ELC.

Description

This program creates or disables user information for an allowed ELC. This table identifies what other ELCs an individual can access. It inquires, loads, and deletes the capability for background report processing and other unit access at the allowed ELC location. This table contains a user ID, an ELC, the user's default unit ID within that ELC, a list of other units the user is authorized to access, whether the user is authorized administration privileges in the default unit, whether the user is authorized to start background runs, and a list of units, other than the default, on which the user is authorized administration privileges. The application programs, outside of the system-wide security layer, check the administrative privileges and background-starting permissions.

Inquiry

To process an inquiry, you must know the user ID. The inquiry option lists information about the user ID, including the ELC authorized, default unit ID, authorized administrative privileges for the default unit, any other units the user ID is authorized administrative privileges, as well as authorization to run background products.

Load

To process a load, process inquiry options against user ID first, and then load applicable information. NFSDX0 stores the new user info in the Allowed ELC table. Up to 24 units may be loaded to an individual.

Change

To process a change, inquire on user ID first, and then fill in the fields that require updates. This updates the Allowed ELC table for the individual.

Delete

To delete, input user ID and process. The delete option removes the user ID from the Allowed_ELC table.

Master Profile Manager (screen 901)

The Master Profile Manager (NFSES0, TRIC: USL, screen 901) part of the program maintains master profiles in the Master_Profile and Master_Profile_Narrative tables. This screen maintains enterprise-wide configuration settings for IMDS.

Description

This part of the program loads, deletes, and changes information stored in the Master_Profile and Master_Profile_Narrative tables. The main objective of this part of the program is to manipulate profiles within the RDMS. NFSES0 allows users to process a TRIC and certain options of it.

Load

The load option loads new master profiles to the RDMS for use at all ELCs.

Inquiry

There are several different types of inquiries that may be processed using screen 901.

- If only a profile ID is entered, NFSES0 produces an output displaying all TRICs, options and sub-options loaded to the Profile ID.
- If a profile ID and one or more specific TRIC codes and/or options are entered, NFSES0 prefills the screen with all TRICs, options and sub-options loaded to that profile for the specified TRIC.
- If there is not a profile ID entered, but one or more specific TRIC codes and/or options are entered, NFSES0 sends a management notice back to the user with all the profile ID(s)/narrative(s) of the master profiles that contain the requested TRIC and its options.

Change

The change option only allows the narrative to be changed.

Delete

The delete option removes a master profile from the RDMS.

Master Profile Manager (screen 871)

The Master Profile Manager (NFSES0, TRIC: USL, screen 871) part of the program also maintains master profiles in the Master_Profile Table Profile and Master_Profile_Narrative tables. This screen maintains enterprise-wide configuration settings for IMDS.

Description

This part of the program copies or merges master profiles. The main objective of this part of the program is to copy or merge master profiles. This screen was created to speed up the process of creating master profiles.

Copy

The copy option copies one master profile to another. Enter a “C” in the type transaction field, enter an existing/valid master profile name in the first field of the Profile ID field and a New Master Profile ID in the first field of the To Profile ID field. This makes a copy of the profile with a new name.

Merge

The merge option merges one profile to another. Enter an “M” in the Type Transaction field, enter an existing/valid Profile ID in the first field of the From Profile ID and enter another existing/valid Profile ID in the second field of the From Profile ID. Afterwards, enter the new profile name in the To Profile ID field and put in a narrative, or enter the same name as one of the profiles in the From Profile ID field.

NOTE: Remember, the original profile IDs remain unless you name the new profile the same name as one of the profiles merged.

Master Profile Report (background)

This Master Profile Report (NFS3R0, TRIC PRB, screen 897) is a new security background report that displays what is in Master Profiles and which master profiles are attached to user IDs.

Description

This security background report displays what is attached to the Master Profile or user ID in three formats.

Format one

Format one retrieves and display all Master Profiles in the RDMS with all TRIC main and sub-options sorted by profile-ID. Output is sorted by profile-ID.

Format two

Format two retrieves and displays Master Profiles in use at ELC with all TRIC main and sub-options. The report may be compiled by unit or work center. This report may be sorted on unit ID, user ID, or work center.

Format three

Format three retrieves and displays the Profiles in use at ELC without Options Sorted by user ID.

Enterprise location code Profile Manager by User Identification

The ELC Profile Manager by User ID program (NFSEG0, TRIC: EPU, screen 886), loads profiles to a user ID and updates the Allowed_Profile table. This program inquiries, loads, and deletes up to eight master profiles at a time to and from a single user ID. The Allowed_Profile table contains a user ID, a profile key, and an ELC. This table may identify one or more profiles for a user within an ELC. A user may have different profiles on different ELCs.

Inquiry

NFSEG0 identifies what master profiles are loaded to the user ID. This program displays all master profiles for every ELC that the user ID has loaded to it.

Load

Enter a user ID and master profile(s) you would like to load and transmit. This program can load up to eight master profiles at a time to a single user ID.

Delete

This program can remove up to eight Master Profiles at a time from a single user ID.

Enterprise Location Code Profile Manager by Profile Identification

This ELC Profile Manager by Profile ID program (NFSEH0, TRIC EPP, screen 882), can inquire, load, and delete up to 25 user IDs to a single Master Profile and update the Allowed_Profile table.

Description

This program can inquire, load, and delete up to 25 user IDs at a time to and from a single Master Profile. The Allowed_Profile contains a user ID, a profile key, and an ELC. This table may identify one or more profiles for a user ID within an ELC. A user may have different profiles on different ELCs.

Inquiry

NFSEH0 identifies which user IDs are loaded to the Master Profile. This program displays all user IDs for every ELC to which the Master Profile is attached.

Load

This option can load up to 25 user IDs to a single Master Profile.

Delete

This option can delete up 25 user IDs from a single Master Profile.

NOTE: Remember that the new security subsystem is based on user IDs, not work centers. The load option, using the work center field, grabs all user IDs (at that current time) under a work center and load deltas to all the user IDs in that work center. The work center field is a way of grouping, if a new member is added to a work center after the delta was plused or minused, that new member does not have the same restrictions as the existing members.

NOTE: If you do not possess the TRIC and its options you are trying to load in your profile/deltas, you may not load the profile/delta to another individual.

NOTE: Enter "*" in Main and/or Sub fields to include all options for TRIC.

Delta Table Report

This Delta Table Report program (NFS3T0, TRIC: DTB, screen 872), displays user information and deltas that have been loaded to user IDs. This is a background program.

Description

This program can retrieve and display the user ID, name, rank, type user, unit, branch, work center, background report processing, green screen authorization, and other unit's access along with the deltas loaded for user IDs.

Output

If user ID, Org ID and Work_Center are left blank, an ELC level report is produced. This repot may be processed by unit, Org ID or work center.

234. Site management

Site management (SIMAN) is the primary security processor for the mainframe computers used at the DECC. It is used for creating and maintaining security and resource control in a Unisys environment. This is the basic security software for the operating system used in mainframes. This software is designed for security officers, site administrators besides the database administrator, and users who maintain access control records.

Function

SIMAN provides an overall set of computer-enforced control protection features, catalogued file security information, and account maintenance features. SIMAN also establishes and defines user IDs for demand, batch, and TIP users for the entire mainframe. A user's password, run mode, clearance level, default account and project ID, terminal time-out, CONS mode level and key-ins are handled by SIMAN. The DBM's privileges in EXEC are controlled in SIMAN. The security manager establishes different accounts for running batch runs in demand. This includes special accounts to run QLP with update, IQU, DBE, and SQL.

Users

The security manager at DECC mainly uses SIMAN. Limited functions are given to the BNCC. A DBM may be given access to view the program but not to alter any data. SIMAN has its own internal security based on administrative levels that correspond to the appropriate user.

Modes of operation

SIMAN has two modes of operation—screen and batch. The screen mode uses a full-screen display for presenting information. The screen mode is *menu-driven*. Batch mode, on the other hand, is *keyword-oriented*. It operates by using keyword commands.

Screen mode

To enter screen mode, you input "@SIMAN" from demand mode. This takes you to the master menu where you select from various options available to perform various database management tasks. This mode is especially effective for making precise entries or for use by less experienced DBMs. Since the screens do extensive checks for valid entries, there is little chance of entering data in error.

Screen directives

On any given screen in SIMAN, all directives are entered in the upper left-hand corner, or home position. This is known as the directive line. When transmitting, the cursor can be in any position on the screen, as it automatically positions itself in the lower right corner of the screen.

Basic directives

OMIT and COMMIT are the two basic directives you may use in processing SIMAN screens during screen mode. OMIT permits you to disregard all inputs made during the processing mode and returns you to the menu screen of the screen set you are working. OMIT can be entered from any non-menu screen to avoid finalizing something entered by mistake. COMMIT processes all input and applies it to the corresponding system databases. Updates to SIMAN screens are made using COMMIT. Let's say, for instance, you accidentally overtyped someone's password; simply type OMIT in the upper left-hand corner of the screen and transmit. Once this is accomplished, all changes made since the last COMMIT command are ignored.

Exiting site management

You exit out of SIMAN by entering the option "X" after OMIT or COMMIT on the directive line on any screen.

Command strings

Commands can be strung together by the use of commas. For instance, you may wish to enter "@SIMAN" and then enter a "5" on the master menu and then enter a "1" on the catalogued file security submenus in order to display the security information on a file without inputting the options on each screen. You could also enter "@SIMAN 5, 1" that allows you to bypass the menus for a faster response time. A space is inserted between "@SIMAN" and the "5", with a comma used after the 5. This also works from any screen.

Batch mode

SIMAN batch mode looks different than the batch run mode in demand. However, it does the same function of displaying, updating and loading large amounts of data as a batch job—primarily related to system security such as accounts and user ids. It's an interactive interface with SIMAN without the use of screens. To enter batch mode, you input "@SIMAN, B" from in demand mode. The "B" option invokes the keyword syntax interface. It is the keywords that allow the execution of batch runs from SIMAN. When an error occurs in a keyword syntax command, SIMAN ignores the entire command and you must reenter the correct command. Examples of keywords used in the user ID maintenance are: (1) DIS for display; (2) USE for user ID; and (3) SYN for syntax (which generates the user ID parameters).

For example, if we want to display the parameters of a certain user ID called RAMO, the syntax looks like the following:

```
DIS USE = RAMO SYN;
```

The output shows all the parameters given to that user ID. Every keyword syntax command must end with a semicolon (;) or it will cause an error.

Batch mode is ideal for processing runstreams released from the AFLCMC at Maxwell AFB, Gunter Annex, Alabama, because it provides for almost all of the functions available in screen mode. For the regular user, use screen mode whenever possible because there are practically no edits and no OMITS options in batch mode to prevent accidental updates.

Span of control

SIMAN interfaces with many areas of the system: User ID maintenance, account maintenance, the SIMAN environment itself, and the overall system environment. These areas are generally controlled within the SYS$*TSS$FILE, SYS$*ACCOUNT$R1 and SYS$*SACRD$ files.

Self-Test Questions

230. Transaction identification code security

1. What is a TRIC?

2. What are the two levels of TRIC access?

3. On what is TRIC access based?

4. What is the purpose of the program TRIC Security Inquiry, screen 896?

231. User identification and profiles

1. Describe the single sign-on capability.

2. What does the master profile Generic_User_Profile contain?

3. What are the two levels of operation in the central database?

4. What is the lowest level of profile administration?

232. Profiles related to database management

1. Explain briefly the allowed ELC concept.

2. What is a delta?

3. What does a green screen refer to regarding computer systems?

4. What else is granted to allow users to run background programs even though their profile allows access to background TRICs?

233. Profiles management

1. What IMDS screen number is known as the security subsystem menu?

2. Match each IMDS security action in column A with its security subsystem program in column B. Items in Column B are used only once.

Column A	*Column B*
____ (1) Updates the Master_Transaction table.	a. Device Maintenance, Screen 631.
____ (2) Updates the Trans_ID_Option table.	b. Transaction On/Off For ELC/Unit, Screen 632.
____ (3) Updates the Trans_ID_Off table.	c. Transaction ID/Option Maintenance, Screen 633.
____ (4) Inquires, loads, changes, and deletes data in the Master_Device table.	d. Master Transaction Maintenance, Screen 635.
____ (5) Updates the FSA_Control table.	e. Enterprise Settings, Screen 636.
____ (6) Updates the Allowed_ELC table.	f. Allowed ELC Employee Number/User ID Maintenance, Screen 681
____ (7) Maintains master profiles in the Master_Profile and Master_Profile_Narrative tables.	g. Master Profile Manager, Screen 871.
____ (8) Maintains master profiles in the Master_Profile Table Profile and Master_Profile_Narrative tables.	h. Delta Table Report, Screen 872.
____ (9) Displays what is in Master Profiles and what Master Profiles are attached to user IDs.	i. ELC Profile Manager by Profile ID, Screen 882.
____ (10) Loads profiles to a user ID and updates the Allowed_Profile table.	j. ELC profile Manager by User ID, Screen 886.
____ (11) Inquires, loads, and deletes up to 25 user IDs to a single Master Profile and updates the Allowed_Profile table.	k. Master Profile Report, Screen 897.
____ (12) Displays user information and deltas that have been loaded to user IDs.	l. Master Profile Manager, Screen 901.

234. Site management

1. What is SIMAN?

2. When is the OMIT directive used?

3. What character option do you enter after the OMIT or COMMIT directives while in screen mode to exit out of SIMAN?

4. What do we use to string commands together?

5. How is a run executed in batch mode of SIMAN?

Answers to Self-Test Questions

215

1. A collection of interrelated data.
2. Records and sets.
3. Physical pointers.
4. A relational database.
5. SQL.

216

1. One.
2. The DMS database.
3. In the RDMS database.

217

1. As a separate entity from a database file, it describes the exact physical layout of every record in the database. It also describes the logical relationship between each record.
2. (1) Data name section—defines any data items used throughout the schema,
   (2) Area section—contains information about each area in the database.
   (3) Record section—describes the records to be placed in each area.
   (4) Set section—used by the DMR to locate records or chain them into various set occurrences; an application program to locate a record via the linkage may use the set name.
3. Describes how different users view the database.
4. A high-level computer schema language used to design and create schemas, and is independent of any programming language.

218

1. Number of pages allocated and the size of each page.
2. The ELC.
3. Two.
4. Defines every record in the database.
5. Every set in the database and every set user in the database.

219

1. An area.
2. Subdivision of an area, and the basic unit of input/output between the area files and internal buffers maintained in the database.
3. Area code, page number, and record number.
4. A DBP represents the relative position of a record on a page and a DBK represents the absolute location of that record.

220

1. A method for storing and accessing a record for each type.
2. Direct, CALC, and Via Set.
3. By the DBK.
4. CALC records.
5. Via Set.

221

1. Owner records are at the highest level of a set with member records linked at a lower level to their respective owner. In order for a record to be an owner there must be records below it.

2. The owner record points to the first member in the chain, and that record points to the next, and so on through the set with the last member pointing to the owner.
3. Where every record of a set also points to the record that precedes it.
4. It places the newest record closest to the owner.
5. Member/owner linkage is established by the system when a record is created in the database.

222

1. Logical.
2. With an arrow pointing from a circled number to a record.
3. With an arrow pointing to a circled number.
4. A record.
5. 7R1 DMS and RDMS schema.

223

1. A description of data structures that are required by a database.
2. To define relational tables, primary and foreign keys, and procedures for manipulating the stored data.
3. The logical and physical structure of one or more databases to accommodate the information needs of the user in an organization for a defined set of applications.
4. The planning and requirements analysis.
5. Determine the following:
   (1) Data requirements of the database in terms of raw data elements.
   (2) Classification and description of these elements.
   (3) Relationships among the objects.
   (4) Types of transactions performed on the elements.
   (5) Rules governing data integrity.

224

1. Vertical columns and horizontal rows.
2. The intersection of a column and a row is a data item. The specific value of a data item is a data value.
3. Child table.

225

1. An association between two or more data items or relational tables.
2. A data relationship is defined with a name, whether it is optional or mandatory, and a degree (how many).
3. One-to-one, one-to many, and many-to-many.
4. Reduce the size of the database and increase the flexibility and performance of queries performed on the data.
5. Because they cannot be represented in a relational database.
6. You must create a third table, called a linking table, to join the two tables together and create two one-to-many relationships.
7. Through the use of key fields.
8. The properties are as follows:
   (1) Every table in the database must have a unique name. No two tables within the same database may be named exactly the same.
   (2) Values are atomic—columns in a relational table are not repeating. This meets the 1NF. This is one of the cornerstones of a relational data model.
   (3) Column values are of the same kind; that is, the values in a column are from the same domain. In other words, columns only contain specific values, not other information such as comments, status flags, or data from a different scale (monthly versus weekly).

(4) Each row is unique—no two rows in a relational table are identical. At least one column of volumes must uniquely describe each row or record. See primary key above.

(5) Sequence of columns and rows is insignificant—order of the columns or rows makes no difference. Allows user to share tables without regard to table organizations.

(6) Each column has a unique name—columns must be referenced by name and not by position, such as in a spreadsheet. Column names need not be unique within the entire database, only in its table.

(7) A primary key field may not be null. Null is not the same as being equal to zero or blank. Both zero and blank are considered to be values. Null means that the value is equal to nothing.

9. To improve query performance and efficiency.

226

1. Data will always be correct and precise, so that your database contains less redundant data, is more efficient, and is more adaptable to changing requirements.
2. To create a set of relational tables that are free of redundant data and with records that can be easily queried and correctly updated, added, or deleted.
3. The rules are as follows:
   (1) Values in each column of a table are atomic—no sets of values within a column.
   (2) Eliminate repeating groups in individual tables.
   (3) Create a separate table for each set of related data.
   (4) Identify each set of related data with a primary key.
4. A relationship between fields such that the value in Field A determines the value in Field B, and there can be only one value in Field B.
5. It deals with relationships between composite key columns and non-key columns.
6. Table must already be in 2NF and there must be no transitive dependencies.
7. It eliminates redundant data, saving space and reducing data manipulation anomalies. Consequently, queries on data will run much faster and more efficiently.

227

1. (1) History.
   (2) AEF.
   (3) Portal.
   (4) Reference.
   (5) Security.
   (6) System Support.
   (7) Interface Support.
2. Up to five years.
3. Reference tables.
4. System support tables.

228

1. Alphabetically.
2. SQL programmers.
3. CREATE TABLE.
4. Primary.

229

1. Upper part, in black shade, contains the table name; lower part, in the color yellow, contains one or more primary keys.
2. It shows that a defined foreign key relationship exists—from parent table to child table.
3. A broken arrow dotted line between the tables.

4. A hyperlink to the table schema.

230

1. A three-character code that the computer uses to determine the program or routine requested.
2. Main option and sub-option.
3. A user's role or job.
4. This program inquires the Allowed_Profile and the User_Profile_Delta tables to see what access user IDs have been granted.

231

1. You have the capability to move from one ELC to another without signing off and signing back on.
2. The basic TRICs that all users need.
3. Enterprise and ELC.
4. Unit DBM.

232

1. Gives DBMs the ability to allow a user from another base access to the DBMs' database without having to be loaded in the database.
2. The adding or subtracting of a TRIC with its respective options and sub-options out of a loaded profile on an individual.
3. Those systems still utilizing a text-based screen transaction using green text and black background screens.
4. They will also have to have the "Can Generate Reports" switch set on screen 887 for a user on your database or screen 681 for an allowed ELC user.

233

1. Screen 38.
2. (1) d.
   (2) c.
   (3) b.
   (4) a.
   (5) e.
   (6) f.
   (7) l.
   (8) g.
   (9) k.
   (10) j.
   (11) i.
   (12) h.

234

1. The primary security processor for the mainframe computers used at the DECC.
2. When you desire to disregard all inputs made during the processing mode.
3. X.
4. Commas.
5. By using keywords.

Complete the unit review exercises before going to the next unit.

Unit Review Exercises

Note to Student: Consider all choices carefully, select the *best* answer to each question, and *circle* the corresponding letter. When you have completed all unit review exercises, transfer your answers to the Field Scoring Answer Sheet.

Do not return your answer sheet to AFCDA.

27. (215) The basic structures of a network database include
    a. data and fields.
    b. records and sets.
    c. input and output.
    d. loops and pointers.

28. (215) How many dimensions do tables have in a relational database?
    a. One.
    b. Two.
    c. Three.
    d. Four.

29. (215) The most distinguishing characteristic of a relational query language, like Structured Query Language (SQL), is
    a. its nonprocedural language.
    b. it's the oldest retrieval language.
    c. the use of a complex data language.
    d. the use of an exact procedural language.

30. (216) Within the Relational Data Management System (RDMS) section of the central database (CDB), reference tables that include equipment and aircraft data are provided by
    a. general-purpose records.
    b. transactional data from enterprise location code (ELC).
    c. the Reliability and Maintainability Information System (REMIS).
    d. traditional legacy Core Automated Maintenance system (CAMS) data.

31. (217) What does the schema of a database describe?
    a. The user of the database.
    b. The location of the database.
    c. Physical relationships between every type of area.
    d. Physical layout of records and logical relationships between records.

32. (217) The two divisions of the schema are identification and
    a. set.
    b. area.
    c. data.
    d. record.

33. (217) Which section of the schema is used by the Data Management Routine (DMR) to locate records?
    a. Set.
    b. Area.
    c. Record.
    d. Data Name.

34. (218) Which element under the area section of the Integrated Maintenance Data System (IMDS) schema indicates how much space is left over?
    a. Load.
    b. Allocate.
    c. Area Size.
    d. Area Maps.

35. (218) How many base level areas does each enterprise location code (ELC) database have?
    a. 10.
    b. 11.
    c. 12.
    d. 13.

36. (218) Which clause in the record section of the Integrated Maintenance Data System (IMDS) schema defines the areas where the records can be found?
    a. Record code.
    b. Record name.
    c. Data identifier.
    d. Location mode.

37. (219) What is the basic unit of input/output between the area files and the internal buffers in the database?
    a. Record.
    b. Header.
    c. Page.
    d. Slot.

38. (219) Record placement in the Integrated Maintenance Data System (IMDS) database depends on association, distribution, and
    a. data.
    b. volatility.
    c. page size.
    d. relationship.

39. (219) The three parts of an Integrated Maintenance Data System (IMDS) record database key (DBK) are area code, page number, and
    a. database pointer.
    b. record number.
    c. data item.
    d. words.

40. (220) You can access Via Set records *normally* through
    a. an entry point record and an owner record.
    b. key data items in the record.
    c. a database pointer.
    d. a database key.

41. (221) Which type of linkage would be present when the owner record points to the first member in the chain, the first member record points to the next member, and the last member points back to the owner?
    a. Prior.
    b. Relationship.
    c. One-way circular.
    d. Two-way circular.

42. (221) What clause in the set section determines the logical placement of a new member record added to a set?
    a. Code.
    b. Mode.
    c. Order.
    d. Set selection.

43. (222) Which element of the database structure is *not* used by the Integrated Maintenance Data System (IMDS) Data Management System (DMS) schematic?
    a. Set.
    b. Area.
    c. Page.
    d. Record.

44. (222) When viewing the rectangle box on an Integrated Maintenance Data System (IMDS) schematic, the area consisting of one schematic page is denoted by
    a. a whole number.
    b. a first digit of the set number.
    c. the first digit of the area code.
    d. the transaction interface package (TIP) number.

45. (223) When building a data model, all data required by the database should be represented
    a. by a sample of the entire population.
    b. completely and accurately.
    c. by data modeling symbols.
    d. graphically and logically.

46. (223) What is the *first step* in building an effective data model?
    a. Schema design.
    b. Adding key attributes to the model.
    c. Planning and requirements analysis.
    d. Adding non-key attributes to the model.

47. (223) Adding details to a data model includes adding data attributes, manipulation and
    a. integrity rules.
    b. classification rules.
    c. calculation formulas.
    d. transaction to be performed.

48. (224) The two-dimensional structure of data values that form relations to one another in a relational database is known as a
    a. row.
    b. key.
    c. table.
    d. column.

49. (224) In a relational database, an atomic value means that it is
    a. destructive.
    b. indivisible.
    c. explosive.
    d. invisible.

50. (224) In a relational database
    a. a row can be duplicated in a table.
    b. each row can have more than one value for each column.
    c. two tables can have exactly the same name within the same database.
    d. the column selected as the primary key field may not contain null values.

51. (225) Multiple occurrences of one data element are related to one occurrence of another, and vice versa, representing which type of data relationship?
    a. One-to-one.
    b. Three-to-one.
    c. One-to-many.
    d. Many-to-many.

52. (226) A relationship between fields such that the value in Field A determines the value in Field B, and there can only be one value in Field B, describes
    a. operational dependency.
    b. functional dependency.
    c. relational dependency.
    d. hierarchy dependency.

53. (226) In a relational table, data may change frequently within a
    a. key field.
    b. non-key field.
    c. data entry field.
    d. table identification field.

54. (226) A functional dependency in which the value in a non-key field is determined by the value of another non-key field is called
    a. an inactive dependency.
    b. a transitive dependency.
    c. a proactive dependency.
    d. a redundant dependency.

55. (227) Which table is part of the Relational Data Management System (RDMS) history tables?
    a. Portal.
    b. Reference.
    c. System support.
    d. Status and inventory.

56. (228) In what order are the tables arranged within the Relational Data Management System (RDMS) schema?
    a. Chronological.
    b. Alphabetical.
    c. Numerical.
    d. Topical.

57. (228) Which statement do we use in Structured Query Language (SQL) to build relational tables?
    a. CREATE TABLE.
    b. START TABLE.
    c. FORM TABLE.
    d. USE TABLE.

58. (228) Relational Data Management System (RDMS) tables are indexed
    a. to increase table capacity.
    b. for logical placement of data.
    c. to arrange data alphabetically.
    d. for faster search and retrieval.

59. (229) In the Relational Data Management System (RDMS) schema diagram, the tables in an area are drawn in
    a. boxes.
    b. circles.
    c. oblong.
    d. triangles.

60. (230) How many levels of access does the transaction identification code (TRIC) have?
    a. Four.
    b. Three.
    c. Two.
    d. One.

61. (231) The Integrated Maintenance Data System (IMDS) master profile generic user profile contains
    a. all the transaction identification codes (TRIC) in IMDS.
    b. the basic TRICs for the maintenance analysts only.
    c. all the enterprise location codes (ELC) in IMDS.
    d. the basic TRICs all users need.

62. (231) What profile level of operation allows database administrators to access all enterprise location codes (ELC) and the Relational Data Management System (RDMS)?
    a. Unit.
    b. ELC.
    c. Base.
    d. Enterprise.

63. (231) Managing the Relational Data Management System (RDMS) tables is the responsibility of the
    a. unit database manager (DBM).
    b. Enterprise location code (ELC) DBM.
    c. Air & Space Expeditionary Force (AEF) user.
    d. central database (CDB) database administrator (DBA).

64. (231) What are the two types of user identification (user ID) in the Integrated Maintenance Data System (IMDS)?
    a. Database administrator (DBA) and database manager (DBM).
    b. Enterprise location code (ELC) and central database (CDB).
    c. Transaction interface package (TIP) and DEMAND.
    d. Online and background.

65. (232) The allowed enterprise location codes (ELC) concept gives the ELC database managers (DBM) the ability to allow
    a. another Integrated Maintenance Data System (IMDS) user from another base access to their database.
    b. a new user ID to be created temporarily for the ELC.
    c. another computer system to switch to IMDS.
    d. an additional ELC for the same database.

66. (232) The adding or subtracting of a transaction identification code (TRIC) with its respective options and sub-options out of a loaded profile on an individual is known as
    a. an *ELCI on Integrated Maintenance Data System (IMDS).
    b. an allowed ELC.
    c. a file update.
    d. a delta.

67. (232) What *best* describes a “green screen?”
    a. Computers with green monitors.
    b. Monitors with a green screen effect.
    c. Systems still using a text-based screen transaction.
    d. New systems using graphical user interface (GUI)-based transaction.

68. (233) Which table identifies whether the data management system (DMS) rollup area is available for transaction processing?
    a. Functional System Analyzer Control (FSA Control).
    b. Transaction_Identification_Off (Trans_ID_Off).
    c. Master_Transaction.
    d. Master_Device.

69. (233) Which program can you use to display green screen authorizations?
    a. Delta Table Report.
    b. Device Maintenance.
    c. Master Profile Report.
    d. Master Profile manager.

70. (234) Site management (SIMAN) establishes and defines user identifications (user ID) for demand, batch, and transaction interface package (TIP) users for
    a. just for Integrated Maintenance Data System (IMDS).
    b. _DS remotes (applicable terminal remote unit-IDs).
    c. the entire mainframe.
    d. the supply system.

71. (234) What are the two modes of operation for site management (SIMAN)?
    a. Batch and transaction interface package (TIP).
    b. Screen and demand.
    c. Demand and batch.
    d. Screen and batch.

72. (234) What site management (SIMAN) mode is especially effective for making precise entries or for use by less experienced database managers (DBM)?
    a. Transaction interface package (TIP).
    b. Command.
    c. Screen.
    d. Batch.

73. (234) What command in site management (SIMAN) is used to update screens in screen mode?
    a. OMIT.
    b. APPLY.
    c. COMMIT.
    d. REFRESH.

Unit 3. Database Retrieval Languages

THE QUERY LANGUAGE PROCESSOR is a high-level language found on Unisys mainframes that allows you to retrieve and manipulate data stored in your computer system. QLP can access both DMS and RDMS databases. However, due to the procedures in QLP, there are very limited commands it can use to retrieve from the RDMS database. For the RDMS, you use SQL. One of the most notable features of QLP is its real-time processing. QLP is useful in instances such as investigating a deficient maintenance area or pulling data for a maintenance supervisor immediately. This unit begins with a discussion of the syntax and structure, along with using conversational QLP. Then, we look at how to create short macros and learn the procedures, as well as how to prepare a customized report using report writer. The last section introduces you to SQL basics and how to use QLP reports with SQL in the RDMS environment, and use of SQL within the IPF.

3–1. Conversational Structured Query Language

Conversational SQL, as its name suggests, is used to inquire directly from the DMS database. Conversational SQL gives the user a quick way to pull up a list of records, count a number of records, make a change to records, delete records, and so forth. In this section, you will learn the basics of QLP. This will include the terms you need to be familiar with, the syntax rules for entering characters, and initialization and termination procedures. Together, we develop QLP programs using the conversational commands COUNT and LIST. As you will see in this section, these two are primarily the basic commands you use for extracting data.

235. Syntax elements

The elements we discuss in this lesson are those we frequently encounter when using QLP, including character string, commands, words, literals, user defined terms, data identifiers, clause, verb, operators, expressions, and syntax errors.

Character string

A character string is a set of continuous characters. One example would be IMDS-QLP2.

Commands

A command consists of one or more words starting with the command name, which is usually followed by one or more clauses. Commands direct QLP to perform a function, such as listing the personnel in a unit. All QLP syntax requires that you form commands from left to right. For example, in using the LIST command, it would look like the following:

LIST MPR-NAME-PERS.

Words

In conversational QLP, words are divided into two categories: reserved and key.

Reserved words

Reserved words are words that have syntactical restrictions. As such, they cannot be used as data identifiers. They are used only within a report definition.

Key words

Key words have special meanings at certain points within the syntax. A key word may be used as a data identifier in any context where it is not interpreted as a key word. In our example, LIST is a key word that instructs QLP to list the chief master sergeants assigned to unit A. In the second example, LIST is not considered a key word; it is part of a heading on the message page. Here are the two examples.

1. LIST MPR-NAME-PERS WHERE MPR-UNIT-ID = ‘A’ AND MPR-GRD = ‘009’.
2. MESSAGE PAGE IS LINE 10 COL 1 ‘MPR LIST’.

Literals

A literal is a character string with an implied value that is determined by its composition. There are three types of literals: numeric, integer, and alphanumeric.

Type	Explanation
Numeric literal	Is a character string selected from the digits 0 through 9. It may contain an optional leading plus (+) or minus (–) sign and a floating decimal (.). 12.4; +4.0
Integer literal	Is a numeric character without a decimal. It may also be referred to as a whole number. 3; 520
Alphanumeric literal	Is a character string enclosed by single quotes (‘). The character string may include any characters in the international syntactical operator (ISO) character set. ‘ANALYSIS REPORT 2’ A quotation mark within an alphanumeric literal is specified by two consecutive single quotation marks: ‘COMMANDER’S REPORT’

User defined terms

A user-defined term is a sequence of one or more words, literals, or combination of both which must be supplied by the user to satisfy the syntactical requirements of a QLP statement. Some examples are file names, data identifiers, conditional qualifying expressions, and report names.

Data identifiers

A data identifier is a user-defined term that defines to QLP, a data item, a table in the QLP subschema, or an occurrence of a data item. Each data identifier must exist in the subschema defining

the data being queried. An example of a data ID is WCR-WK-CTR-ID. This data identifier represents a work center ID in the WCR–360 record in the QLP schema.

Clause

A clause is an ordered sequence of words and literals that communicates pieces of information, such as the particular order in which data is to be sorted. Conditional and picture clauses are two types of these.

Conditional clause

A conditional clause is a string of relational expressions, keywords, literals, and a path name that defines the qualification and path selection criteria for record selection. A conditional clause is also called a WHERE clause.

Picture clause

A picture clause describes the size and type of field for a data item or derived name. It specifies whether the field contains alphanumeric or numeric data and how many characters are in that field. An example, using a picture clause, would be DETAIL LINE PLUS 2 COL 1 OAR-CURR-HRS PIC IS 9(05). In this statement, the picture clause is PIC IS 9(05). The 9 indicates that the field contains numeric data and the (05) indicates that the field is 5 characters long. An alphanumeric picture may be written as X(05), with the X indicating alphanumeric data. The field length—in the picture clause, not the data identifier—is two positions. Any value less than 10 has a leading 0.

Verb

The verb is a key word that specifies the action to be performed for a given clause. Primary and secondary action verbs are the two types used with QLP.

Primary action verb

A primary action verb defines the primary action to be performed by a command. In other words, all of these commands must start with a primary action verb. Some examples of primary action verbs include LIST, COUNT, DELETE, PURGE, and CHANGE.

Secondary action verb

A secondary action verb defines the secondary action to be performed by a command. Secondary action verbs must appear after the primary action verb and before the condition clause. An example of a secondary action verb would include SORTED.

The following statement is an example of the use of both primary and secondary action verbs, as well as a conditional clause. In this case, LIST is the primary action verb, while SORTED is the secondary action verb, and WHERE MPR-UNIT-ID= ‘C’ is the conditional clause:

```
LIST MPR-NAME-PERS, MPR-GRD SORTED ON DESCENDING MPR-GRD WHERE MPR-UNIT-ID = ‘C’
```

Operators

Three types of operators used with QLP include arithmetic, relational, and Boolean.

Arithmetic

Arithmetic operators are key words or special characters that communicate to QLP a request for an arithmetical operation to be performed. The arithmetical operations are addition (+), subtraction (–), multiplication (*), and division (/). For subtraction, the minus sign must be offset with spaces between the two data identifiers. The arithmetic operators are addition, plus or “+”; subtraction, minus or “–”; multiplication, multiply or “*”; and division, divide or “/.”

Relational

A relational operator is a key word or a special character that communicates to QLP a request to perform a relational operation between data items and literals. The relational operators are equality, inequality, greater than, less than, greater than or equal to, less than or equal to, within range, and outside range. All relational operators may be preceded with NOT.

Boolean

Boolean operators are key words that link relational operators in a logical sequence. The Boolean operators are *AND* and *OR,* with the *AND* having precedence over *OR.*

Expressions

QLP uses four types of expressions:

- Operand.
- Relational.
- Boolean.
- Conditional.

Operand

The operand expression is a data identifier, literal, or derived name (user-defined variable). It may be followed by one or more series of arithmetic operators, data identifiers, literals, or derived names. Some examples of operand expressions are as follows:

```
EQPP-OPERATING-TIME
AQ01 + AQ02
WAHM-CUR-ASSG MULT WAHM-CUR-HRS
```

Relational

A relational expression is two operand expressions separated by a relational operator. Let's use the relational expression ESH-ENG-FLY-HRS < '01000'. Here is an example of how you would enter it into the system:

```
LIST ESH-MDS WHERE ESH-ENG-FLY-HRS < '01000'
```

Boolean

A Boolean expression is one or more relational expressions separated by a Boolean operator. Let's say we wanted to use the Boolean expression MPR-DAFSC-FST = '391' or MPR-DAFSC-FST = '392'. Here is an example of how they would be entered:

```
SELECT RECORD WHERE MPR-DAFSC-FST = '391'
OR
MPR-DAFSC-FST = '392'.
```

Conditional

The conditional qualifying expression specifies the data conditions that must be met in order to retrieve or update data. In this statement the conditional qualifying expression is WHERE EVT-TYPE-EVT = 'MT' AND EVT-START WITHIN RANGE '03200' '03231'. The following is an example of this:

```
SELECT RECORD WHERE EVT-TYPE-EVT = 'MT' AND EVT-START WITHIN RANGE '03200' '03231'.
```

Syntax error

When you execute a command, QLP checks your input and displays a message if you make a syntax error. For commands, other than single keyword commands such as **HOLD**, an unrecognized word at the end of the string causes a syntax error for that command even if it could be a complete command. Consider the following example:

LIST MPR-NAME-PERS WERE MPR-UNIT-ID = ‘A’

This input is marked as in error and is not executed because WERE is not recognized as part of the LIST command or the start of a new command.

236. Syntax skeleton

In QLP, the structure of a command and the statements that fall under it are displayed in the form of a syntax skeleton. The user determines what statements can be used under a QLP command and what statements are optional by looking at a syntax skeleton and the notational symbols used in it. In other words, the syntax skeleton provides a QLP statement construction requirement. This lesson teaches you some important notation symbols and their meanings.

Notation convention

To interpret QLP syntax properly, you must be aware of its notational conventions. These are the words and symbols commonly used in a QLP statement. A QLP command uses several of these conventions. A QLP syntax skeleton is broken down like this:

LIST [BY COLUMN]

$$\left\{\begin{array}{l}\left\{\begin{array}{l}TOTAL\\MINIMUM\\MAXIMUM\\AVERAGE\\DATABASE-KEY\ OF\end{array}\right\} data-identifier-1\ [edit-clause-1]\\ operand-expression-1\ [edit-clause-2]\end{array}\right\},\ldots$$

$$\left[SORTED\left\{ON\left[\begin{array}{l}ASCENDING\\DESCENDING\end{array}\right]data-identifier-2\right\},\ldots\right]$$

$$[USING\ POINTER\ TO\ data-identifier-3]$$

$$[WHERE\ conditional-qualifying-expression-1]$$

$$[VIA\ path-name-1\ PATH]$$

Uppercase words

Words and characters that appear in uppercase (capital letters) represent key words. If these uppercase words or characters are underlined, they are mandatory and must be used in each occurrence of that statement. If uppercase words or characters are not underlined, they are optional.

Lowercase italic words

Words and characters that appear in lowercase italics are generic terms, names, or literal types that must be replaced with actual names or literal types in that statement. Most lowercase italic words have a number attached to avoid confusion, in case two terms are of the same type. For example, *data-identifier–1* and *data identifier–2* in the same syntax skeleton indicates that two data identifiers must be supplied.

Space

In a command, spaces must be used where indicated by the skeleton. Consecutive spaces are treated as one space.

Special characters

Like spaces, special characters (such as =, *, $) must also be used where they are indicated in a skeleton.

Brackets

Statements enclosed by brackets ([ ]) are optional. If there is more than one statement enclosed in a set of brackets, the user may choose only one of those statements.

Braces

Statements enclosed by braces ({ }) are mandatory. If more than one statement is enclosed by a set of braces, the user must choose only one of those statements.

Double bars

Statements enclosed by double bars (|| ||) are optional. If there is more than one statement enclosed by a set of double bars, each statement may be used once.

Ellipsis

An ellipsis (...) indicates that the preceding construct (part of a skeleton) may be repeated indefinitely as long as a space separates each repetition.

Comma/ellipsis

Comma/ellipsis (,...) means that the preceding construct may be repeated any number of times, but each repetition must be separated by a comma.

Continuation character

Most commands require only one line of text. If a command will extend beyond one line, skip one space, enter a semicolon (;), and press the XMIT key to continue on the next line. In QLP, the semicolon serves as the continuation character, as in the following example:

```
LIST COL EQPP-UNIT-ID,EVT-CURR-IND,EQPP-SRD,EQPP-INDICATOR,EQPP-CURRENT, ;
EQPP-TRANS-EQP-IND,EQPP-STATUS,EVT-KEY,EQPP-KEY ;
SORT DESC EQPP-KEY,EVT-KEY ;
WHERE EQPP-UNIT-ID WRG ‘A’ ‘M’ AND EQPP-KEY MASK -* = ‘A’ AND EVT-KEY ;
MASK *—* WRG ‘97’ ‘99’ ;
VIA P158–395–370
```

Parentheses

Parentheses () may be used in numeric operand expressions to specify the order in which elements are to be evaluated. Expressions within parentheses are evaluated first and are computed from left to right. For example, the computer multiplies, divides, adds, and subtracts from left to right—in that order. Therefore, use parentheses to avoid ambiguity in logic.

Separator

When two consecutive literals are used, a separator must offset them. The legal separators are a space, comma, left parenthesis, right parenthesis, single quotation mark, arithmetic operator, and a comparative operator. Use single quotation marks to denote the beginning and end of an alphanumeric literal.

237. Initialization and termination

This lesson covers QLP session, initialization and termination procedures.

Session

You gain access to the DMS database and perform manipulation tasks within a QLP session. A typical QLP session consists of the tasks shown in order from top to bottom on the following table.

Command	Description
@QLP	Initializes the QLP processor.
INVOKE	Invokes the QLP subschema in the DMS schema for a specific database
OUTPUT (optional)	Directs the session output to a specified device (like your terminal or printer) or file.
MODE (optional)	Modifies QLP session default settings.
Retrieval and Update Commands (LIST, COUNT, GENERATE, etc.)	Use command sequence necessary to perform the purpose of the session.
EXIT	Terminates the session.

Initialization procedures

QLP is initialized through DEMAND mode. After you sign on in demand mode, enter QLP as shown in the following table.

Step	Enter the following:
1	@QLP
2	INVOKE FS-QLP-XXXX DMS*CAMSDBG–7R1 (XXXX = ELC number)
3	MODE IS NO ECHO (The default is MODE IS ECHO)
4	USE *RAMOS* AS USER

Enter Query Language Processor

The actual entry into QLP is through the use of @QLP. After you log in DEMAND, QLP is initialized by the command @QLP. At this point, you are now in the QLP environment. However, QLP still does not know which particular database you want to access. Sometimes QLP terminates when trying to compile a report, format, procedure, and so forth. When this occurs, you must reenter QLP beginning with the @QLP command.

Next step

The next step is to INVOKE the DMS database subschema for your ELC. After initialization, you must link to a specific subschema (section of the database) via the INVOKE command. For example, if you wanted to link QLP to 7R1 DMS subschema and your ELC code is 4321, your input might look like this: INVOKE FS-QLP–4321 DMS*CAMSDBG–7R1. This input would link QLP to the DMS subschema.

QLP *does not* allow you to issue any further commands after the @QLP statement until you issue an INVOKE. The INVOKE statement tells QLP which particular database (or subschema) you wish to use. The number 4321 indicates you are accessing the DMS database for your home ELC (4321). INVOKE establishes a link between the user and a particular subschema (or database) or files to be accessed. More than one INVOKE statement may be entered in one session; however, a second or subsequent INVOKE statement automatically overrides the previous INVOKE statement.

Third step

The third step is MODE. When the commands to initialize QLP are input, the input is echoed back. The computer repeats each input after it is entered. This can be stopped with the command MODE IS NO ECHO. Selecting either ECHO or NO ECHO enables or disables QLP's echoing of your most recent input. When the mode is ECHO, the monitor displays each line entry once you press the XMIT key. The default setting of the echo function is ON. This means that every time you enter QLP, your inputs are redisplayed on the monitor, unless MODE IS NO ECHO is specified. It's your preference whether you want the echo function to be ON or OFF.

Fourth step

The fourth step is to establish a user savefile. The USE command allows the user to specify a user savefile to which reports, macros, procedures, formats and requests can be saved. The savefile is very similar to having your own library of reports. The savefile can be created by entering USE RAMOS FOR USER. The word RAMOS is the name of the user savefile. Once the savefile is created, the name is no longer required during your current session. QLP then recognizes the user savefile (RAMOS) when you refer to it as USER. There are two more savefiles that are established automatically when QLP is initialized: temporary and permanent. All of the reports, etc., are placed in the temporary savefile. This file is abolished when the user signs off from QLP. If you desire to retain a report, format, procedure, etc., you can save it to a permanent savefile. This savefile is like a public library in that anyone can review its content.

Termination

When finished with the QLP, the user may terminate the session. EXIT terminates QLP session and releases the link between the user and the database (subschema) or data files and terminates processing by QLP, but leaves you in DEMAND. It also purges all formats, procedures, reports, and macros that were not saved by use of the SAVE command.

Terminal sign off

If you want to leave the DEMAND mode and completely sign off from your terminal, then you perform additional commands to permit it. Any control statement (beginning with @) acts as an EXIT command. The command @FIN signals the computer you are finished and provides a time accounting statement for the user-ID; @@TERM/$$CLOSE closes the session; and $$SOFF deactivates the terminal.

So, to completely sign off your terminal from a QLP session, the procedure is as follows:

Step	Enter the following:
1	EXIT
2	@FIN
3	@@TERM/$$CLOSE
4	$$SOFF

238. Using inquiry commands

The inquiry commands perform an action on specified items in the database or data files established by their respective WHERE clauses. A WHERE clause specifies the criteria to be used for selection of data from the database.

Count

COUNT tallies the number of occurrences of the data item specified in the COUNT command action clause. It may also be defined as the number of records containing the data items specified. COUNT does not extract data from the database. The tally takes place only for that data qualified by the WHERE clause. If there is no WHERE clause, all items specified in the action clause of the command are extracted. Use a WHERE clause to establish your selection criteria. This clause tells the computer to operate on the data and select just the records where one or more data item meets a certain criteria. It does this by comparing the data items with some possible value. Use COUNT if you want to know the number of occurrences of specific records that met a certain criteria.

First input

The first input in the following example will go through the database and give you a count of all the MPR–251 records, where GRD represents the personnel's "pay-grade," and WRG represents "within range." If you do not want to get a tally of all records of one type, you can specify a WHERE clause, as indicated in the second input. The WHERE clause is always defined with a conditional-qualifying expression by using combinations of relational and Boolean expressions.

```
COUNT MPR–251
COUNT MPR–251 WHERE MPR-GRD = '009'
COUNT MPR–251 WHERE MPR-GRD WRG '007' '009' AND MPR-UNIT-ID = 'A'
```

More than one

If you attempt to count more than one data item from the same record, the computer will only count each data item once and give you a warning that you have tried to count the same record type more than once.

List

LIST prints (displays) the content of specific records, which may range from a specific data item to an entire record. The basic command for getting a list of the data you want is the word LIST followed by the name of the data item or items that you want. If you want to see a list of all personnel records loaded in IMDS, your input would be as follows.

```
LIST MPR–251
```

Not limited

Your LIST clause is not limited to one data item. You can also provide a list of the values of those data items determined by the WHERE clause. If there is no WHERE clause, all occurrences of the specified identifier are printed. Your inquiry may be extended, as indicated here:

```
LIST MPR-NAME-PERS, MPR-GRD WHERE MPR-GRD WRG '007' '009' AND; MPR-UNIT-ID = 'A'
```

Your response would be similar to the following abbreviated response:

```
MPR-NAME-PERS = ACTON DEAN C
MPR-GRD = 007
MPR-NAME-PERS = ARMSTRONG CALVIN C
MPR-GRD = 007
```

Vertical row

Notice that your response is listed in a vertical row. That is, the data identifiers are on the left and the results of your inquiry are listed to the right, as indicated previously. Two disadvantages are noticed immediately. First, if the output is long, it's not easy to read. Second, the data identifier is repeated for each output. In the next lesson, you'll learn how to improve your output.

List by column

LIST BY COLUMN makes your output more readable and presentable by putting it in a columnar format. Your output is listed in columns with each column headed by its corresponding data item name. Shown in a straight line, the output data is not based on the size but rather the picture clause of the record being retrieved. MPR-NAME-PERS is always X(18) long (JOHN + 14 spaces thereafter). Using LIST BY COLUMN, compare the following input with the previous input, referencing the personnel's primary Air Force specialty code (PAFSC):

```
LIST BY COLUMN MPR-NAME-PERS, MPR-GRD, MPR-PAFSC ;
WHERE MPR-GRD WRG '007' '009' AND MPR-UNIT-ID = 'A'
MPR-NAME-PERS MPR-GRD MPR-PAFSC
ACTON DEAN C 007 39170
ARMSTRONG CALVIN C 007 43171
JOEL BILLY 007 43171
KEATON ALEX P 007 39170
THOMPSON HUNTER S 009 39300
```

Sort

The SORT clause causes the output data to be arranged alphabetically, numerically, or both. The first data identifier denotes the major key of the sort. All remaining data identifiers are sorted in sequence as minor keys. Notice that a sort in ascending order is assumed unless SORT DESC (descending from highest to lowest) is used, or no order is specified. Examine the following input:

```
LIST BY COLUMN MPR-NAME-PERS, MPR-GRD, MPR-PAFSC ;
SORT ON MPR-NAME-PERS ;
WHERE MPR-GRD WRG '007' '009' AND MPR-UNIT-ID = 'A'
```

Alphabetized

The resulting output will be a list of all personnel, alphabetized from A to Z. If you want an inquiry that will be list ranked in descending order (from highest to lowest), the input would look something like this:

```
LIST BY COLUMN MPR-NAME-PERS, MPR-GRD, MPR-PAFSC ;
SORT DESC MPR-GRD, ASC MPR-NAME-PERS ;
WHERE MPR-GRD WRG '007' '009' AND MPR-UNIT-ID = 'A'
```

Grade and name

In the previous example, personnel are sorted by grade in descending order and by name in ascending order. As you see in this example, ASCENDING and DESCENDING can be abbreviated to ASC and DESC respectively. Also, notice that you can specify multiple data identifiers in the SORT clause. If you want to alphabetize the assigned personnel in descending order, your inquiry could be formatted in the following way:

```
LIST BY COLUMN MPR-NAME-PERS, MPR-GRD, MPR-PAFSC ;
SORT DESC MPR-GRD MPR-NAME-PERS ;
WHERE MPR-GRD WRG ‘007’ ‘009’ AND MPR-UNIT-ID = ‘A’
```

Mask

There are times when only part of a data item is important to you. For example, you might want only the work center record (WCR)–360 records where the third position of the mnemonic is Y. QLP has a MASK clause that allows you to specify only portions of a data item as the selection criteria. This tells the computer you want to list all WCR–360 records where the third position of the work center mnemonic is the letter D. Use an asterisk (*) for the positions you want to ignore in the data item and a dash (-) for the position you want to see. If you are only interested in the second position of a four-character data field, make your selection using either of the following examples: ***-**** or ***-**. As you can see, asterisks after the last significant field are optional. You are not limited to a single dash. If your retrieval dictates, multiple dashes are allowed. For example, **-*-*** or **-*-**, specifies only the first and third positions of a data identifier and is represented as ‘AD’ for the first and third positions if you want all data items with a first character of ‘A’ and a third character of ‘D’. In our example, a LIST command using MASK might look like the following:

```
LIST BY COLUMN WCR-WK-CTR-MNEM-ID ;
WHERE WCR-WK-CTR-MNEM-ID MASK **-* = ‘D’
```

In this example, WK-CTR refers to “work center,” and MNEM refers to “mnemonic.”

Ease-of-use commands

We use ease-of-use commands to make quick corrections; however, they are restricted to conversational QLP.

Repeat

REPEAT executes the previous action clause with a new WHERE clause, thus establishing different data selection criteria for the previous processing command. Suppose the following command contains an invalid data identifier in the WHERE clause:

```
LIST MPR-NAME-PERS, MPR-UNIT-ID WHERE MPR-GRD = ‘006’
```

In this example, the user wanted to select grade 007. To correct the error, the user could repeat the command with a new WHERE clause by entering the following:

```
REPEAT WHERE MPR-GRD = ‘007’.
```

Where same

WHERE SAME is used to indicate that the data selection criteria specified on the previous command are to be applied to the current command. It allows the selection criteria for the current command to be the same as that of the immediately preceding command. (SAME cannot be used with a VIA clause.). If, for example, you unintentionally omitted a required item identifier from your LIST command, you could request the required data item without reentering the entire LIST statement. Suppose you entered the following command and realized you forgot to request the assigned unit:

```
LIST MPR-NAME-PERS WHERE MPR-GRD = ‘007’
```

You can easily attain the required information by entering the following:

```
LIST MPR-UNIT-ID WHERE SAME
```

Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

235. Syntax elements

1. Define a character string.

2. What are the types of words used with QLP?

3. What is alphanumeric literal?

4. What is a user-defined term?

5. What user-defined term identifies the occurrence of a data item?

6. How would you write a picture clause for a numeric, 7-position data identifier?

7. As used with QLP, what do verbs mean?

8. Identify is the relational operator in the following statement: LIST WHERE MPR-GRD = ‘007’

9. What are Boolean operators?

10. What is a conditional qualifying expression?

236. Syntax skeleton

1. How can a user determine the construction requirement of a QLP statement?

2. Match each notational symbol in column B with its related definition in column A. Items in column B may be used once, more than once, or not at all.

Column A		*Column B*
____ (1)	When underlined, they are mandatory.	a. { }.
____ (2)	Must be replaced with an actual name.	b. [ ].
____ (3)	Used when a command exceeds one line of text.	c. || ||.
____ (4)	The preceding construct may be repeated indefinitely as long as a space separates each repetition.	d. ....
____ (5)	When more than one statement is enclosed, each statement may be used once.	e. ,....
____ (6)	Indicates that statement is mandatory.	f. Uppercase words.
____ (7)	Indicates that the preceding construct may be repeated any number of times, but each repetition must be separated by a comma.	g. Lowercase italic words.
____ (8)	Must be used where indicated by the skeleton.	h. A space.
____ (9)	When more than one statement is enclosed, the user may choose only one of the statements.	i. ;.
____ (10)	Indicates the order in which numeric operand expressions are to be evaluated.	j. ().

237. Initialization and termination

1. What command initializes QLP?

2. What command stops the echoing?

3. What command terminates a QLP session?

238. Using inquiry commands

1. What information is obtained by using the COUNT command?

2. What information is obtained by using the LIST command?

3. What conversational inquiry command makes your output more readable and presentable?

4. What SORT command sorts numeric values from highest to lowest?

5. How would you write a MASK statement indicating that the fourth position of a five-digit literal is to be selected if it equals T?

6. What command allows you to execute a previous action clause with a new where clause?

7. What command allows the selection criteria for the current command to be the same as that of the preceding command?

3–2. Query Language Processor Macro and Procedure

It's easy to develop QLP programs with the use of QLP macros and procedure. These are mini-programs designed to do small and large retrievals. Learning how to use these two QLP entities could make your job easier, especially if you routinely extract information from the DMS database. Database managers and maintenance analysts alike benefit a lot when they utilize these powerful short retrieval programs.

239. Macro

Within this lesson, we will discuss the purpose of a QLP macro, as well as its definition. We will also consider the parts that make it up and how it is used. A QLP macro is a savefile entity that allows you to reference a unique name that you defined with the macro anytime you want. You build a macro for short QLP statements or expressions when you have a need to use it repeatedly. It may contain data items or commands to execute what you want to obtain. A macro is simple and easy to use because it does not require a command to invoke the macro. However, you need to remember the macro name you defined in order to invoke it.

Definition

The process of building a macro is called the macro definition. A macro cannot be used until it's defined. During macro definition, all source input is stored in the temporary savefile. After successful completion of the macro definition, an executable version of the source is also created in the temporary savefile. If there are errors in the macro definition, an executable version is not created. The macro definition begins with the MACRO clause and terminates with the END-MACRO clause.

SYNTAX:
MACRO macro-name–1
macro-statement–1
END-MACRO

Macro name

This can be any legal element name. Be careful when you use a name because the macro name can override other QLP keywords, procedures, or data identifiers (i.e., MACRO MY-QLP).

Macro statement

The macro statement is in the next line after the macro name. It can be a single QLP command or series of commands.

End-macro

The END-MACRO causes the element to be compiled and saved into the temporary savefile.

Macro clause

The macro clause tells the QLP that you plan to define a macro and give an entity name to it. The name must not duplicate any other macro name that already exists in the temporary savefile. The MACRO clause below specifies a macro name of CHIEFS. In this example, we defined a macro name of 'CHIEFS' and created a statement with a conversational QLP command LIST to list the names of all chief master sergeants in the DMS database (MPR–251 records). After the macro clause, you must enter the actual macro definition.

```
MACRO CHIEFS
LIST MPR-NAME-PERS WHERE MPR-GRD = '009'
END-MACRO
```

End-macro clause

END-MACRO tells the QLP you are finished with the macro definition. After the initial macro definition, QLP attempts to compile your macro. If there are no errors, the executable version of the macro is automatically stored in the temporary savefile. In the event of errors, go into the EDIT mode and correct all errors. After exiting the EDIT mode, you must manually issue a compile command for your macro. (We discuss detailed instructions for using the EDIT mode in another lesson.) Since both the source and executable versions of the macro are stored in the temporary savefile, it's a good idea to save the macro into either the user or permanent savefile before using it.

Macro usage

Next time you want a list of all the chief master sergeants again, when you enter QLP you invoke the macro by entering 'CHIEFS.' Doing so executes the statement again and provides you the list you specified. Once the macro executes, it displays the names of the chief master sergeants in the DMS database. In other words, once a macro is defined and saved, you can invoke it by using its name; for example: CHIEFS.:

240. Procedure

This lesson covers QLP procedure definitions.

Query Language Processor procedure

A procedure allows a QLP user to group and define a sequence of commands that can be referenced and executed later.

Definition

The process of building a procedure is called the procedure definition. While a procedure is being built, all input source is stored in the temporary savefile. Upon compilation, the executable version of the source is also stored in the temporary savefile. Procedures are invoked or referenced using the CALL command. The procedure definition consists of all lines between the PROCEDURE and END PROCEDURE statements. Both statements are mandatory when building a procedure.

Statement

The procedure statement tells QLP that the user is about to define a procedure and gives an entity name to the procedure. The procedure name can be from 1 to 12 alphanumeric characters long. It

must not duplicate any other procedure name that already exists in the temporary savefile. After the procedure statement, the user enters the actual procedure definition. For example:

```
LIST BY COLUMN MPR-NAME-PERS, MPR-GRD, MPR-PAFSC ;
WHERE MPR-GRD WRG ‘007’ ‘009’ AND MPR-UNIT-ID = ‘A’
LIST USING FORMAT TOP–3 ;
WHERE MPR-GRD WRG ‘007’ ‘009’ ;
AND MPR-UNIT-ID = ‘A’
```

End procedure

The END PROCEDURE statement tells QLP that the user is finished with the procedure definition. As in format definition, QLP attempts to translate an executable entity out of the input source after the initial procedure definition. Any syntax errors require correction within the EDIT mode of QLP. With the preceding procedure, two separate list inquiries were grouped and can be executed with one line input.

Call manning

CALL executes a previously defined procedure. Remember to save your newly-defined procedure into either the permanent or user savefile.

241. Output

You can now obtain a listing of desired records and know ahead of time how many you’ll get. However, this listing is not beneficial if you cannot look at them. If you are listing only a few records, you can let them scroll across the screen and review them without printing. However, if you are requesting more than a screen of records (about 23 lines), the first records will probably scroll off the top of your screen before you can see them. You can get a printed copy of your listing in two ways.

Side-by printer

If you have a side-by printer (desktop) attached to your terminal and are printing a small number of lines (100 or less), send the output to your printer as it scrolls across your screen. Do this by entering the command @@PRNT. This command activates your terminal to echo everything that comes to the screen to your side-by printer. This can be slow, depending on the type of printer you have. To deactivate the printer echo, enter @@NOPR. The following is a sample output to the side-by printer:

```
@@PRNT
LIST MPR-NAME-PERS, MPR-GRD ;
WHERE MPR-UNIT-ID = ‘A’ AND MPR-GRD WRG ‘007’ ‘009’
@@NOPR
```

Enterprise output manager

Use the Enterprise Output Manager to print copies of lengthy inquiries. QLP has an OUTPUT command that allows you to direct output to the Enterprise Output Manager. Prior to entering the LIST command, enter the following command:

```
OUTPUT TO DEVICE DP0069
```

This tells the computer that any output you get from your commands should be sent to the Enterprise Output Manager printer with device ID DP0069. Then enter a LIST command to retrieve your data. In this case, the only response to return to the screen will be a message telling how many records were selected.

Print on the enterprise output manager printer

At this point, the output will not print on the Enterprise Output Manager printer until you enter the following command:

```
OUTPUT TO ORIGINATOR
```

This command tells the computer to print your output through the Enterprise Output Manager. There are two things to remember about sending data to the Enterprise Output Manager. First, be sure to enter the correct device ID. Second, if you do not have access to the printer, check with the operator of the printer to make sure it is available before printing.

The following is a sample output to the Enterprise Output Manager:

```
OUTPUT TO DEVICE DP0069
LIST MPR-NAME-PERS, MPR-GRD ;
WHERE MPR-UNIT-ID = 'A' AND MPR-GRD WRG '007' '009'
OUTPUT TO ORIGINATOR
```

Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

239. Macro

1. What is the purpose of a macro?

2. What is the purpose of the MACRO clause?

3. What is the purpose of the END-MACRO clause?

240. Procedure

1. How are procedures invoked?

2. What are the first and last statements of a QLP procedure?

241. Output

1. What command tells the terminal to echo an inquiry that comes to the screen to your side-by printer?

2. What statement directs an inquiry to print at the Enterprise Output Manager printer?

3–3. Query Language Processor Report Writer

Report writer allows the user to tie QLP commands together to create a program that can gather, evaluate, manipulate, and display data from a DMS or RDMS database. This feature of QLP is used to produce reports from a DMS database. Report Writer allows you to define the layout, organization, and content of an output report.

242. Definition

The report definition consists of the clauses between and including the REPORT clause and END REPORT. When the system encounters the END REPORT clause, QLP analyzes the report description and lists it on the terminal along with error diagnostics. QLP also saves the compiled report definition into the temporary savefile. Certain components are used in the construction of a report definition and are discussed in this lesson.

Report

The REPORT clause must be the first clause in a report definition and is mandatory. It allows the user to name the report. The name must be unique. An example is:

```
REPORT MYFIRSTQLP
```

The REPORT clause puts the user into INPUT mode.

Page

The PAGE clause enables the user to specify the length and width of a page in terms of the number of lines and columns and describes the dimensions of the field of print of the page. The output device limits the field of print. The following is an example:

```
PAGE IS 60 LINES, 120 COLUMNS
```

When PAGE is not specified, the system defaults to 80 lines and 132 characters per page.

Margin

The MARGINS clause defines the number of nonprintable lines and columns on a page. It also specifies the location of the print field defined by the PAGE clause. A typical MARGINS clause would look like this:

```
MARGINS ARE 2 TOP 2 BOTTOM 4 LEFT 4 RIGHT
```

When margins are not specified, the system defaults to four lines top and bottom, zero margins left and right. When using PAGE and MARGINS, the total number of columns between them should not exceed 132 characters.

Page-counter

The PAGE-COUNTER clause specifies the beginning page number of a report and is optional. The following is an example:

```
PAGE-COUNTER = 1
```

This clause causes the system to begin numbering the pages with "2." If no PAGE-COUNTER clause is specified, the default is "1." When processing two related retrievals, you can specify a beginning page number for the second retrieval to allow continuous page numbering between reports.

Sort

The SORT clause specifies the order in which the report is to be presented. Up to 40 data identifiers can be sorted. You can specify the order, ASCENDING or DESCENDING. When no order is specified, the sort defaults to ascending order. The data identifiers are sorted in the order they are presented. The order can be changed as desired. In the SORT clause, all of the data identifiers that fall after ASCENDING or DESCENDING are sorted in that order until the order is changed. An example of the sort clause is as follows:

```
SORTED MPR-GRD, MPR-NAME-PERS
```

Select

SELECT allows the user to specify selection criteria within a report using either a conditional clause or a procedure. When using a conditional qualifying expression, you can specify the path (or WHERE) on which the data lies in the subschema by using the VIA PATH phrase. VIA PATH expedites processing by specifying the navigational route from one record set to another. This is an optional clause. The following is an example of a select clause:

```
SELECT WHERE MPR-UNIT-ID = 'B' AND MPR-PAFSC = '2R051'
```

Default

The DEFAULT clause defines the attributes (type data and length of field) of a derived name that was not explicitly declared. This clause is optional. When a data identifier has a large predefined picture, you may establish a DEFAULT picture clause. If, for example, your output will only use five of the 10 print positions, you may specify a default picture. Doing so allows the computer to print more data on a single line rather than leaving leading spaces or using a second line. An example of a DEFAULT statement is as follows:

```
DEFAULT PIC IS 9(05)
```

Declare

The DECLARE clause defines the explicit attributes of a derived name. It includes the data size in digits, a scale factor, and a signed/unsigned indication. The DECLARE clause has an accompanying PICTURE clause. Remember, use the DECLARE clause to specify the picture of a user-defined term. This clause is optional. The following is an example of a DECLARE clause:

```
DECLARE WORKHOURS PIC 9(04)
```

Controls

The CONTROLS clause specifies the items that are to control processing within a report and establishes a hierarchy for processing. A change in the value of a control or the satisfaction of a Boolean expression involving a control typically signals that other reports processing, such as

heading or footing presentations, is to take place. The order of the data identifiers and Boolean expressions in the CONTROLS clause determine the hierarchy. The first item in this list establishes the highest level of control. Each time a data identifier changes value from one path to the next and each time a Boolean expression is true, control breaks occur. When control breaks occur, all specified headings and footings for that data item are printed. An example is as follows:

```
CONTROLS MPR-GRD, MPR-NAME-PERS, MPR-UNIT-ID
```

This clause takes effect only if a heading is specified for that data identifier used (HEADING FOR data-identifier clause). The items specified in the CONTROLS clause must be previously defined in the SORT clause and the order should match. This clause is optional.

Message

The MESSAGE clause enables you to make a cover page for your report. You specify the line number and starting column for each statement on the message page. The message page is presented prior to any report group presentation. The following is an example of this optional clause:

```
MESSAGE LINE 35 COL 15 'USE THIS AS A SAMPLE QLP'
```

Line

The LINE clause not only lets you specify a line and column number on the present page, but also a starting line and column on the next page.

```
LINE 1 NEXT PAGE COL 10
```

Other options of the LINE clause let you indicate spacing. The following is an example:

```
LINE PLUS 1 COL PLUS 5
```

In the example, the line plus lets you specify spacing between lines. The "1" indicates single spacing. If you want double spacing, use "2". COLUMN PLUS 5 indicates that the next field will start "5" spaces to the right of the previous field.

Skip

The SKIP clause controls the vertical spacing in the report. A SKIP clause allows you to skip to the next page. Examples include the following:

```
SKIP TO NEXT PAGE
SKIP TO LINE 10
SKIP TO NEXT PAGE LINE 10
```

The SKIP clause may appear anywhere within a report definition where an action clause is permitted.

Heading/footing

Heading and footing clauses create headings and footings for different parts of a report. You can use a HEADING FOR REPORT, HEADING FOR PAGE, FOOTING FOR REPORT, and FOOTING FOR PAGE only once per report. Only *one* footing and one heading is allowed for each data identifier or Boolean expression listed in the CONTROLS clause. The associated action clause may contain a COMPUTE, RESET, TERMINATE, REPORT, SUPPRESS, PRINT, LINE, or SKIP clause.

Report heading

The HEADING FOR REPORT clause can create a title page, preface, introduction, or acknowledgment for your report. The following in an example:

```
HEADING FOR REPORT LINE 1 COL 1 'Work Center Capability Data—MARK FOR PA.'
```

The HEADING FOR REPORT clause prints as specified on the first page of the report only.

Page heading

The HEADING FOR PAGE clause lets you print a heading on each page of the report. For example:

```
HEADING FOR PAGE LINE 3 COL 10 'RANK' COL 30 'NAME' COL 50 'AFSC' COL 70 'SSAN' COL
90 'MAN-NUMBER.'
```

Data heading

A HEADING FOR data identifier or Boolean expression clause prints a heading each time a control break occurs on that data identifier or Boolean expression. The following is an example:

```
HEADING FOR OAR-MDS LINE PLUS 2 COL 50 'SORTIES FLOWN' COL 80 'HOURS FLOWN'
```

Report footing

When you want something printed at the bottom of the page of the report, use a FOOTING clause. The following is an example:

```
FOOTING FOR REPORT LINE 50 COL 10 'MY FIRST QLP RETRIEVAL'
```

To print a footing at the bottom of each page of the report, input the clause as follows:

```
FOOTING FOR PAGE LINE 60 COL 1 'PAGE' COL 6 PAGE-COUNTER
```

This prints the word PAGE at the bottom of each page in column 1, followed by the page number. If a FOOTING is desired each time a control break occurs, use a FOOTING clause. For example:

```
FOOTING FOR OAR-MDS LINE PLUS 2 COL 10 'HOURS FLOWN = ' COL 40 RCOUNT OF OAR-
CURR-HRS
```

This prints "HOURS FLOWN =" followed by the value of RCOUNT. RCOUNT only gives a tally of occurrences, not a cumulative total.

Detail

The DETAIL clause describes that part of the report that is the lowest hierarchical logical output of the report. It sets up the body of the report. The associated action clause may contain a COMPUTE, RESET, TERMINATE, REPORT, SUPPRESS, PRINT, LINE, or SKIP clause. For example:

```
DETAIL LINE PLUS 1 COL 10 MPR-GRD COL 30 MPR-NAME-PERS COL 50 MPR-DAFSC COL
70 MPR-SOC-SCTY COL 90 MPR-MAN-NUMBER
```

When constructing a DETAIL clause, use a LINE PLUS 1 clause. In the example above, LINE PLUS 1 prints the data specified in single space. If LINE PLUS is not specified, the system prints only one line per page. As a result, this is a mandatory clause.

End report

END REPORT must be the last clause in the report definition. When END REPORT is input, the system completes your report definition and lists it back on your terminal with applicable error messages.

Compile

COMPILE causes a source description of an entity to be processed producing an executable entity of the same name in the temporary savefile. It takes the definition you put in English and translates it into a language the computer understands. The English version is referred to as the source, and the computer language version is referred to as the executable.

Building

When building a report definition, use the clauses in the order specified in the syntax. The computer processes the clauses in the order they are inputted. The clauses are read from left to right. When the specified order is not followed, a logic error can result. End report is a mandatory clause.

243. Generating

The GENERATE command executes a previously defined report once you have created and stored it in a QLP savefile. It also lets you specify selection criteria, indicate output destination, select line density, and specify the number of copies. There are several options with this command as described in the following paragraphs.

Detail report

If you request a DETAIL report, the report is generated as is.

Summary report

If you prefer a summary report, the entire report is printed, with the LINE and SKIP clauses being disregarded.

No preference

If you do not state a preference, the system assumes you want a DETAIL report. You may indicate a savefile from which the report is generated. If a savefile is not generated, the computer goes to the temporary file, then to the user savefile, and finally to the permanent savefile, until it finds the proper entry. Operand expressions may be included. The TO clause lets you specify where you want the report to print.

Originator

If you choose the ORIGINATOR, the report returns to your terminal.

File

When you select FILE, QLP attempts to assign the file name specified. If the file name cannot be assigned, the file is catalogued and the report output is sent to this file.

Device

If you choose DEVICE, the report is sent to the Enterprise Output Manager printer indicated by the device name. The TO clause allows you to specify where you want your report printed; however, if you don’t use a TO clause in the GENERATE command, the destination of the report output is the ORIGINATOR as long as the GENERATE command was not within a request. If it was within a request, the report is sent to the system configured printer.

Forms

The FORMS clause may be specified to send a message to the operator before the report output is sent to its designation. A one-character response is required from the operator for the report to be printed. This clause is used most often to tell the operator what special print form to mount.

Density

The DENSITY clause allows you to specify the number of lines per inch (six or eight) to be printed.

Copies

A COPIES clause specifies the number of copies. If COPIES is not specified, the system defaults to one copy.

Where clause

If you know the beginning record of a linear path that includes all of the data identifiers referenced in the GENERATE addition to the SELECT clause, you can use a WHERE clause to specify additional selection criteria. When you use the WHERE clause, the clause is linked to the SELECT clause in the report by a Boolean AND.

Via path

You may also specify a path name by using a VIA PATH clause. The following is an example of the GENERATE command:

```
GENERATE MYFIRSTQLP FROM TEMPORARY TO DEVICE NTR09P
```

244. Editing

By now you may have run a QLP savefile entity. What if your savefile entity contained errors or you need to make modifications? This lesson assists you in making the necessary corrections or modifications before reinputing your data. This lesson provides you with the knowledge of how to store and delete programs and covers commands that are used to store and delete savefile entities.

Correct errors

Occasionally, you may need to correct errors in a report definition or modify an existing report. To do this, enter the EDIT mode. To edit the report MYFIRSTQLP stored in the user savefile, input the following:

```
EDIT REPORT MYFIRSTQLP FROM USER.
```

Edit

When you edit a report, it is copied into the temporary savefile if it was stored in the user or permanent savefile. You can edit reports, macros, etc., *only* in the temporary savefile. Once you enter the EDIT mode, there are a variety of commands you may use.

Line

Each line has a line number that is always a positive integer. A line number must be specified when a new line is entered. The line number is not a part of the line, but is a method of identification. The smallest allowable line number is 100 and the largest is 99999.

List

This command, if used alone, lists the entire report definition. If you want a particular line, follow the word LIST with a space and the line number of that line. In that case, state the command as LIST 100 (100 is the line number). If you use the word GO, the command is GO 100. Both of these statements accomplish the same thing. Examples of the LIST command are as follows:

```
LIST
LIST 200
LIST 200,300 to list a range of lines from 200 - 300 inclusive
```

New line insertion

Use this command when you wish to add new lines of text between two existing lines. An example is as follows:

```
100 REPORT MYFIRSTQLP 300 MARGINS 2 TOP 2 BOTTOM 2 LEFT 2 RIGHT
```

You may want to add a PAGE clause between the REPORT (100) and MARGINS (300). The line number must be between 100 and 300. The input could look like this: 200 PAGE 60 LINES 120 COLUMNS. This can be input from any position on the screen as long as the line is preceded by a start of entry (SOE) character.

Line deletion

To delete a line, simply type in the line number and transmit. You can also use this command to delete multiple lines. For example:

```
DEL 100,500 Deletes lines 100 - 500 inclusive.
```

Change existing lines

To change an existing line of text, call up that line number using the GO or LIST command.

Start of entry

Insert a SOE to the immediate left of the line number, make the necessary changes, move the cursor to the end of the line, and press the XMIT key.

Replacing strings—CHANGE

This option allows for easy correction of typographical errors when in EDIT mode. To locate occurrences of a given string and replace them with another given string, use the CHANGE command to correct errors in a single line, as in the following example:

```
LIST 200
200 WHERE UNIT-ID = ‘B’
CHANGE /B/A/
200 WHERE UNIT-ID = ‘A’
```

Error

The erroneous literal or word is easily corrected. In addition to the previous examples, you can expand the replacement string as shown in the following examples:

```
CHA /881001/88110/ALL (changes all lines containing 881001 to 88110)
CHA /881001/88110/100,500 (changes literals 881001 between lines 100 and 500 inclusive)
```

Leave-edit

To leave the EDIT mode and simultaneously place the updated entity in the temporary savefile, use the LEAVE-EDIT command. You can abbreviate this command to LEA. When this command is transmitted, you leave the EDIT mode. Only the source version changes. You must now do a COMPILE to create the executable version of the report.

Omit

If you want to discard the changes made while you were in the EDIT mode and leave edit session, enter OMIT and press the XMIT key. You can abbreviate this command to OMI.

245. Savefile

Once you create your QLP savefile entity and make the necessary changes, where do you store it for future use? In this lesson, you’ll learn about the various savefiles available to you for storing your savefile entities along with the commands you use with them. We begin with a discussion about savefiles.

Savefiles

Savefiles provide the user with the necessary library function to handle the savefile entities. It allows the user to save both symbolic and executable entities such as formats, procedures, and reports. There are three separate savefiles, including temporary, user, and permanent.

Temporary

This file is assigned automatically to each user when the QLP is activated. It is erased when the user enters a different INVOKE command, an @ command, or the EXIT command.

User

A user can establish a unique savefile through the use of the USE command. For example:

```
USE RAMOS FOR USER
```

If a user savefile is not specified, you are not allowed to specify USE command in the savefile commands. It allows specification of a username file to be used as a savefile. If more than one is specified, only the most recently specified savefile is to be used.

Permanent

This file is automatically assigned to QLP when the INVOKE command is processed.

Savefile commands

We discuss the required savefile commands in the following paragraphs.

Print

This command gives a list of the elements residing in a particular savefile, including their usage and time and date of creation. The PRINT command also lists all or specific entities. You may specify whether you want a source or executable listing. You may also specify the savefile from which you want a listing. If you do not specify a particular savefile, the system assumes you want the temporary savefile. The following is an example:

```
PRINT ALL REPORTS FROM PERM
```

Copy

COPY command allows the user to copy a savefile entity from a QLP savefile or external file into a QLP savefile. All macros, procedures, and so on, may be copied at once or only specified ones may be copied. The user has the option to copy either a source or executable listing. If neither is specified, both will be copied. You, the user, must specify from what savefile or file name you wish to copy. However, you do not have to specify into what savefile you would like to copy. If no savefile is specified, the computer assumes it is the temporary savefile. If a savefile entity already exists in the destination savefile with the same name as the one to be copied, the copy will not be done for that entity. Instead, you will receive a response stating “that entity already exits, please purge.” The following is an example of the COPY command:

```
COPY MYFIRSTQLP FROM TEMP INTO USER
```

It is often wise to copy an entity from the temporary savefile into either a user or permanent savefile after you make the needed changes and before you compile it. This is handy if there are many errors; when compiling, if there are still any errors remaining, you may be “kicked out” of QLP and your temporary savefile may be abolished.

Save

The SAVE command copies a savefile entity from the temporary savefile into either a user or a permanent savefile and then purges it from the temporary savefile. You may save all entities or just

save specific ones from a temporary to a user or permanent savefile. If you do not specify which savefile, the computer looks for a user savefile if a USE command was previously input. If no user savefile exists, the entity is saved into the permanent savefile. The following is an example:

```
SAVE REPORT MYFIRSTQLP INTO USER
```

Purge

This command lets you erase entities from savefiles. You can eliminate all of the entities at once or specific ones from a savefile. You can eliminate either the source or the executable version; however, if you do not specify, both are eliminated. If you do not specify which savefile you wish purged, the computer assumes it to be the temporary savefile and purges it. An example use of this command would be:

```
PURGE MYFIRSTQLP FROM PERMANENT
```

Compile

As previously mentioned, COMPILE causes a source description of an entity to be processed producing an executable entity of the same name in the temporary savefile. It takes the definition you put in English and translates it into a language the computer understands. The English version is referred to as the source, and the computer language version is referred to as the executable. When constructing your compile command, you have the option of specifying what type of entity (macro, procedure, report, etc.) is meant. If you do not specify which is meant, the computer determines the type by the first line of the source description of the savefile entity name. You can also specify the savefile from which the entity is to be compiled. If you don't, the computer will look for it. The computer looks to the temporary savefile first, and then proceeds on to the user and permanent savefiles until the entity is found. The first savefile in which the entity is found is the one from which it is compiled. You can use the COMPILE command at any time, except when in the EDIT mode. You must use the COMPILE command after you edit an entity if changes are made. If you change an entity but do not compile the resulting file, it reverts to its previous composition when executed. Use COMPILE prior to the execution of a modified program. If an executable element already exists in the temporary savefile, that executable element is automatically purged. An example use of this command would be as follows:

```
COMPILE MYFIRSTQLP FROM TEMP
```

Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

242. Definition

1. What clause must appear first in a report definition?

2. What does the PAGE clause do?

3. What are the default margins?

4. What clause specifies the beginning page number of a report?

5. What is the purpose of the SORT clause?

6. What statement contains the WHERE clause?

7. What is the purpose of the DEFAULT clause?

8. What is the purpose of the DECLARE clause?

9. What clause establishes a hierarchy for processing?

10. What clause enables the user to develop a cover page?

11. How would you write a statement requesting a double-spaced report?

12. How many headings and footings are allowed for each data identifier or Boolean expression listed in the CONTROLS clause?

13. What statement sets up the body of a report?

14. What clause must be the final clause in a report definition?

243. Generating

1. If you do not specify DETAIL or SUMMARY, what is the system response?

2. What clause allows you to specify where you want your report printed?

3. What does the DENSITY clause control?

4. In addition to the SELECT clause, what clause may you use to specify additional selection criteria?

244. Editing

1. What does the EDIT feature do?

2. What is the function of the editing LIST command?

3. How are line numbers deleted?

4. What two commands do you use to call up line numbers?

5. How do you change an existing line of text?

6. What is the command to exit the edit mode?

245. Savefile

1. What savefile is erased when the user executes the exit command?

2. What savefile may be established by the user?

3. What savefile is automatically assigned?

4. What does the PRINT command do?

5. What does the COPY command do?

6. What does the SAVE command do?

7. What does the PURGE statement do?

8. What does the COMPILE statement do?

3–4. Structured Query Language

QLP can retrieve data from the RDMS database using SQL. However, it can do so with very limited commands. SQL was designed to build and access relational database tables. Most of the QLP commands you learned in the previous lessons only work in a DMS database. For example, QLP conversational commands work only for the DMS database. To query or retrieve data from the RDMS database, you must use SQL commands. Nevertheless, QLP is capable of accessing the RDMS database by inserting some SQL commands (called “imbedding”). This section teaches basic retrieval methods using a single SQL command and how to imbed an SQL command in a QLP Report. You will also learn some basics of using SQL within the IPF environment.

246. Structured Query Language basics

An RDMS provides all necessary functionality to create storage structures (tables), insert data, modify it, delete it, and to perform queries against the data. It uses the logical relationships created by the primary/foreign key relationships laid out in your tables. You use SQL exclusively for performing queries in the 7R1 CDB RDMS. If you need to delete, insert, modify, or create tables in the RDMS portion of the 7R1, contact the enterprise database manager at Gunter. This l covers how to use the SELECT statement to perform queries from a single table of the RDMS. The SELECT statement in SQL is similar to the LIST command in QLP.

Select

All SQL queries begin with the SELECT statement. You can specify which columns you want in a list by using commas to separate the column names or you can select all columns from a table by using an ‘*’. You complete the SELECT statement with a FROM clause. The FROM clause specifies the table or group of tables from which the system is to get the columns. In the FROM clause of the

SELECT command you can list a single table, multiple tables, or even a sub-query. We only limit our queries from a single table.

The complete *syntax* for the SELECT statement is as follows:

$$\text{SELECT}\left[\begin{matrix}\text{ALL}\\ \text{DISTINCT}\end{matrix}\right]\left\{\begin{matrix}select-list\\ *\end{matrix}\right\}$$

FROM table-specification-list

[WHERE *Boolean-expression*]

[GROUP BY *column-specification-list*]

[HAVING *Boolean-expression*]

NOTE: The clauses are indented to distinguish them from the statement. The syntax is not position-sensitive or case-sensitive. You may write it from the first column on your screen or indent it any way you want. Even the commands and clauses need not be in uppercase. We use uppercase characters for the command and clauses to recognize immediately their usage in the statement. We present specifications and expressions in lowercase.

Now let's describe the clauses within the SELECT syntax in the following table:

Clause	Description
ALL	Selects all rows identified by the FROM, WHERE, GROUP BY, and HAVING clauses. ALL is assumed if neither ALL or DISTINCT is used.
DISTINCT	Omits records that contain duplicate data in the selected fields. Each value listed in the SELECT statement must be unique to be included in the results.
select-list	Normally lists the column names. May also be a numeric, character, or datetime value expression.
*	Selects all columns of the tables listed in the FROM clause.
WHERE *Boolean-expression*	Restricts the rows (or records) selected.
GROUP BY *column-specification-list*	Describes how to divide rows into separate groups. Constructs one row for each group specified, usually containing similar data from columns selected.
HAVING *Boolean-expression*	Further restricts the search conditions on groups formed by the GROUP BY clause.

Boolean expressions

As in QLP, SQL uses the same Boolean expressions. A Boolean expression is one or more relational expressions separated by a Boolean operator. These operators are comparison operators.

The comparison operators listed in following table are used in SQL:

Operator	Definition
=	Equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Operator	Definition
<>	Not equal

The following are special comparison operators in SQL:

Special operator	Description
LIKE	Allows you to use wild card characters in your condition criteria. Example: WHERE jdd_ddr.wuc_lcn LIKE '23%' *Note: The '%' and '_' are the SQL standard wild card characters and work with SQL statements performed on the mainframe.*
BETWEEN	Similar to the WRG operator discussed earlier in this unit, in that it looks for all values between the criteria input. Example: WHERE jdd_ddr.wuc_lcn BETWEEN '23000' AND '23ZZZ'
IN	Allows you to specify a list of possible conditions for the comparison. Example: WHERE jdd_ddr.wuc_lcn IN ('23000', '16ABD', '13AAA')

Simple Structured Query Language query

Following the preceding syntax, we can write it the way presented in the following table for a basic query. The statements in the table are in the format we use for the purpose of learning simple SQL queries using the SELECT statement.

Query	Statement
Syntax	SELECT *column_name1*, *column_name2*,… FROM table_name or table_name1, table_name2,…
Example 1	SELECT organization_number, organization_kind, organization _type, detachment_number, start_date_zdt, geographic_location_indicator FROM org_code
Example 2	SELECT * FROM action_taken_code

Clauses

Now that you know the basic SELECT statement, we are going to show you how to add conditions to it. This allows you to only receive data back from the table that meets the criteria for which you are looking.

Where clause

The WHERE clause is used to set conditions and follows the FROM clause. The following syntax will help demonstrate how it is set up, along with a couple examples.

Syntax
SELECT column_name1, column_name2 FROM table_name WHERE column_name1 = expression

Example 1
SELECT * FROM action_taken_code WHERE action_take_cd = ‘a’
Example 2
SELECT organization_number, organization_kind, organization _type, detachment_number, start_date_zdt, geographic_location_indicator FROM org_code WHERE organization_number = ‘0001’

Group by clause

This clause specifies grouping of rows in the result table. For the first column specified in the GROUP BY clause, SQL arranges rows of the result table into groups whose rows all have the same values for the specified column. If a second GROUP BY column is specified, SQL groups rows in each main group by values of the second column. SQL groups rows for values in additional GROUP BY columns in a similar fashion. Let’s look at the following syntax along with an example.

Syntax
SELECT <*column_name1*>, COUNT(<*column_name2*>),… FROM *table_name1* WHERE *column_name1* LIKE *expression1* GROUP BY *column_name1*
Example
SELECT aue_mission_design_series, COUNT(*) as nbr_end_items FROM jdd_end_item_emds WHERE aue_mission_design_series LIKE ‘ F%’ GROUP BY aue_mission_design_series

Having clause

The HAVING clause allows conditions to be set on the groups returned by the GROUP BY clause. This allows you to select only certain groups. The HAVING clause works just like a WHERE clause but is processed after the WHERE and GROUP BY clauses have done their work. If you use the HAVING clause without the GROUP BY clause, the implicit group against which the search condition is evaluated is all the rows returned by the WHERE clause. A condition of the HAVING clause can compare one aggregate function value with another aggregate function value or a constant. Let’s consider the syntax and an example.

Syntax
SELECT *column_name1*, COUNT(*column_name2*),… FROM *table_name1* WHERE *column_name1* LIKE *expression1* GROUP BY *column_name1*

HAVING <aggregate function> = <aggregate function> | <constant> <= >= < >
Example
SELECT aue_mission_design_series, COUNT(*) as nbr_end_items FROM jdd_end_item_emds WHERE aue_mission_design_series LIKE ‘ F%’ GROUP BY aue_mission_design_series HAVING COUNT(*) > 50

Functions

These are a type of SQL expression that return a value based on an argument they are supplied. There are several functions available through the CDB; however, we only cover a few of them here. You can find a complete list with syntax in Unisys publication 88079330–016 Data Access SQL Reference Manual. There are two types of functions used in SQL: aggregate and row.

Aggregate functions

Aggregate functions are used to summarize the data returned by a SELECT statement. The following paragraphs present some of the more commonly used aggregate functions.

NOTE: Anytime you use aggregate functions in addition to one or more column names in the Select statement, you must use the Group By clause. All column names not part of the aggregate function must be contained in the Group By clause.

Count

The aggregate function COUNT computes either the number of rows in a group of rows or the number of non-null values in a group of values. The keyword DISTINCT specifies that the duplicate values are to be eliminated before computing the count. When DISTINCT is used and the SELECT clause contains more than one field, the combination of values from all fields must be unique for a given record to be included in the results. If the argument to COUNT function is “*”, the function computes the count of the number of rows in a group. If the argument to COUNT function is not “*”, null values are eliminated before the number of rows is computed. The argument column_ref or expression can be of any type. The result of the function is of INTEGER type. The result is never null. Let’s examine the syntax and a few examples.

Syntax
COUNT ({ [ALL] *expression* } | { DISTINCT *column_ref* } | *)
Example 1
SELECT COUNT (*) FROM org_code
Example 2
SELECT COUNT (DISTINCT organization_number) from org_code
In a SELECT statement, COUNT can be used in the following example: **Example 3**
SELECT aue_mission_design_series,

```
COUNT(*) as nbr_end_items
FROM jdd_end_item_emds
WHERE aue_mission_design_series LIKE ‘ F%’
GROUP BY aue_mission_design_series
HAVING COUNT(*) > 50
```

Avg

The aggregate function AVG computes the average of a collection of values. The keyword DISTINCT specifies that the duplicate values are to be eliminated before computing the average. Null values are eliminated before the average value is computed. If all the values are null, the result is null. The argument to the function must be of type SMALLINT, INTEGER, NUMERIC, REAL, or FLOAT. The result is of type NUMERIC. Let’s consider the syntax and an example.

Syntax
AVG ({ [ALL] *expression* } | { DISTINCT *column_ref* })
Example
SELECT mission_design_series, AVG (start_dt_tm_uts – stop_dt_tm_uts) FROM status_and_inventory WHERE mission_design_series LIKE ‘ F015%’

Sum

The aggregate function SUM returns the sum of the values in a group. The keyword DISTINCT specifies that the duplicate values are to be eliminated before computing the sum. The argument column_ref or expression can be of any type. The result of the function is of the same data type as that of the argument except that the result is of type INTEGER when the argument is of type SMALLINT or TINYINT. The result can have a null value. Consider the following syntax and example.

Syntax
SUM ({ [ALL] *expression* } | { DISTINCT *column_ref* })
Example
SELECT mission_design_series, SUM (start_dt_tm_uts – stop_dt_tm_uts) FROM status_and_inventory WHERE mission_design_series LIKE ‘ F015%’

Row functions

Row functions perform the indicated argument on each row returned by the SELECT statement.

Length

The scalar function LENGTH returns the string length of the value of the given character expression.

Syntax
LENGTH (char_expression)
Example
SELECT LENGTH(aue_mission_design_series) FROM jdd_end_item_emds

Timestampdiff

The scalar function TIMESTAMPDIFF returns an integer representing the number of intervals by which date_time_exp2 is greater than date_time_exp1. Let's consider the syntax and an example.

Syntax
TIMESTAMPDIFF(interval, date_time_exp1, date_time_exp2) Intervals used:
SQL_TSI_FRAC_SECOND | SQL_TSI_SECOND | SQL_TSI_MINUTE | SQL_TSI_HOUR | SQL_TSI_DAY | SQL_TSI_WEEK | SQL_TSI_MONTH | SQL_TSI_QUARTER | SQL_TSI_YEAR

Example
SELECT mission_design_series, serial_number, elc, equip_id, sortie_date_udt, sortie_number, sortie_mod, TIMESTAMPDIFF(SQL_TSI_HOUR, actual_stop_dt_tm_uts, actual_start_dt_tm_uts) FROM utilization WHERE mission_design_series LIKE ' F015%'

Sorting query output

The output for your SQL query defaults in the order the records are placed in the table. The data values in each column display in an unsorted fashion, because they are displayed from the first to the last record (row). To arrange your information (sort) you must use the ORDER BY clause. The ORDER BY clause arranges the values accordingly, whether ascending or descending. The default is ascending. The columns you want to sort must be specified in the select list that preceded the ORDER BY in the SELECT statement. You cannot sort a column of data values if it is not listed on the preceding SELECT statement. The following shows the basic syntax, along with an example.

Syntax
ORDER BY column-specification [ascending | descending]

Example
This query produces a list of all aircraft assigned to ELC 5361 sorted by mission design series first, then by serial number, arranged in ascending order. SELECT mission_design_series, serial_number FROM status_and_inventory WHERE elc = '5361' GROUP BY mission_design_series, serial_number ORDER BY mission_design_series, serial_number

Similarity to Query Language Processor

To summarize, the SQL SELECT statement is similar to the LIST command in conversational QLP. There is also a SELECT command in QLP. To differentiate between the two, we commonly refer to the SELECT statement in SQL as 'SQL SELECT', even though we don't write it as such. Both these queries produce a list of information based on the conditions you set. Remember, to query the DMS, use QLP; to query the RDMS, use SQL.

247. Using Structured Query Language for Query Language Processor reports

The Unisys QLP provides support for interaction with relational databases from within QLP Reports. You cannot use any other form of QLP (e.g., conversational commands) to access the RDMS tables besides QLP reports. However, this is not a problem as the support from within reports works very well for creating flat files using the "For Transfer Option." This is by far the easiest method to obtain data from the mainframe. Once this data is downloaded to your PC, it is possible to manipulate it in several ways using commercial off the shelf database software.

Using QLP to access RDMS tables

You might ask, "Why use QLP for the RDMS if we have SQL?" The ELC DBMs are given limited privileges as far as accessing the RDMS tables. We are generally going to be assigned querying privileges when we go to the RDMS tables. Only the enterprise DBM may use more privileges in SQL to access the RDMS tables. Using QLP reports allows us to store or save the SQL query for future and regular use. This simplifies our file management of our QLP user files.

Entering the RDMS environment

You use the INVOKE statement to establish connection to the RDMS environment. To direct QLP to connect to the RDMS tables, you use a slightly different and simpler form of the INVOKE statement than what you use to access the DMS areas. This is important. If you don't enter the correct invoke statement, your report will not run. The following is what the statement looks like:

INVOKE RDMS APPL01 FS

Using QLP reports for RDMS tables

You select data for use in QLP Reports against the RDMS tables with the SQL SELECT statement. There are a few differences between the two programs such as QLP does not support the "*" to indicate all columns from a table are to be displayed; instead, each column must be listed separately. Also, QLP does not use or need an escape character to use the "%" wildcard character. QLP does not need the "&" to indicate a SELECT statement extends over multiple lines.

Creating RDMS QLP reports

Using QLP to access the RDMS tables is very similar to writing a QLP to access the DMS. You still have the four mandatory entries for each report: REPORT, SELECT, DETAIL, and END REPORT. The primary difference is in the SELECT clause of the QLP Report command. Instead of performing

a SELECT PROCEDURE or a SELECT WHERE, you substitute your SQL SELECT statement. You don't need to enter "SQL SELECT"; you only enter "SELECT." When you invoke the RDMS environment in your INVOKE statement, the QLP automatically recognizes the SELECT statement in the report definition to apply to the relational tables of the RDMS. Other than that, all the other parts of the QLP report are the same.

Preparing your QLP report

There are basically two ways to prepare your QLP report depending on your proficiency in QLP and SQL and your familiarity with the DMS and the RDMS. These two approaches are defined in the following paragraphs.

Preparing the SQL separately, then integrating it into the QLP report

This means you write down the SQL query first to determine what information you need (the columns) and where the information is to come from (the tables). You narrow down the information with the conditions needed with the appropriate clauses. After you prepare your SQL query, then you create your QLP report definition, inserting your SQL query at the line where you make your selection criteria.

Preparing the QLP report definition with the SQL query at the same time

Preparing the QLP report definition with the SQL query at the same time. When you reach the selection criteria of you report, you fill in the appropriate SQL queries using the SELECT statement and all the necessary clauses. The following are examples of basic QLP reports (without the other clauses that affect the appearance of the output) that access the RDMS.

Example 1

This QLP displays the count of aircraft status records in the table for the specified ELC (5361) and org_number (0007) by mission design series (type of aircraft) and equipment ID. We call our report "Status Count." Note that the LINE clause in QLP arranges the information by columns from the tables to be displayed in a horizontal fashion.

```
@QLP
INVOKE RDMS APPL01 FS
REPORT Status Count
SELECT COUNT(*) AS cnt, mission_design_series, serial_number, equip_id
    FROM status_and_inventory
    WHERE elc = '5361' AND org_number = '0007'
    AND org_kind = 'BHV' AND org_type = 'WG'
    GROUP BY mission_design_series, serial_number, equip_id
    ORDER BY mission_design_series, serial_number, equip_id
DETAIL
LINE cnt, mission_design_series, serial_number, equip_id
END REPORT
GENERATE Status Count
EXIT
```

Example 2

This QLP displays an overall count of non-aircraft SRDs in the table. The name of the report is 'Non-aircraft SRD.'

```
@QLP
INVOKE RDMS APPL01 FS
REPORT Non-aircraft SRD
SELECT count(*) as tot_acft
        FROM Standard_Rpt_Dsgntr
        WHERE srd_key not like ‘A%’
DETAIL
LINE ‘There are ‘, tot_acft, ‘ Aircraft SRDs in the table.’
END REPORT
GENERATE Non-aircraft SRD
EXIT
```

QLP session

When you are in DEMAND mode, a QLP session with your report might look like the example given. It is a good practice to purge your savefile (the previous compiled report) in order for QLP to process your current report. If you keep different versions of similar reports, especially customized reports (e.g. an SRD report for different organizations), then it is a good idea to give them different report names. Notice that a run-ID (rprqlp) and a temporary user savefile (rp0124) were established.

For example: This QLP displays several columns from the SRD table for Aircraft SRDs (as seen on your terminal in DEMAND mode). The report name is ‘Aircraft SRD.’

```
@run rprqlp
@cycle,c rp0124-temp.,0
@asg,cp rp0124-temp.,f///10000
@QLP
INVOKE rdms appl01 fs
PURGE Aircraft SRD
REPORT Aircraft SRD
SELECT srd_key, eqp_designator, jdd_rpt, tcto_rpt,
micap_rpt, type_equip from Standard_Rpt_Dsgntr
WHERE srd_key like ‘A%’
DETAIL
LINE srd_key, eqp_designator, jdd_rpt, tcto_rpt,
micap_rpt, type_equip
END REPORT
GENERATE Aircraft SRD to file rp0124-temp
EXIT
@free rp0124-temp.
@fin
```

For transfer option

The FOR TRANSFER option is an excellent tool to use when writing QLP reports that access the RDMS. Without this option, the output lines from a QLP report are limited to 132 characters. However, using this option allows you to exceed this limitation. The only drawback is that when you use the FOR TRANSFER option, you can only generate the report to a file. You cannot generate it to a printer or to the screen. You may use the QLP temporary savefile or a catalogued file in your master file directory. To use the option, simply enter FOR TRANSFER after the report name in the REPORT line of your QLP.

```
@QLP
INVOKE RDMS APPL01 FS
REPORT Status Count FOR TRANSFER
SELECT COUNT(*) AS cnt, mission_design_series, serial_number, equip_id
        FROM status_and_inventory
        WHERE elc = '5361' AND org_number = '0007'
        AND org_kind = 'BHV' AND org_type = 'WG'
        GROUP BY mission_design_series, serial_number, equip_id
        ORDER BY mission_design_series, serial_number, equip_id
DETAIL
LINE cnt, mission_design_series, serial_number, equip_id
END REPORT
GENERATE Status Count to file RP0123
EXIT
```

Simple to use

From the examples given, we can see that it is very simple to insert an SQL query in a QLP report. Using the different clauses for a report definition, you can customize the output for printing. The 'for transfer' option enables you to import the data to a commercial application to improve the viewing of the data.

248. Using Structured Query Language within interactive processing facility

The Unisys IPF supports interaction with relational databases. IPF provides a very good avenue to perform quick, simple queries against an RDMS table using a SELECT statement. You can use IPF to create, edit, and test SQL SELECT statements before copying the statement directly into a QLP report. It can make creating queries against relational tables much simpler and quicker.

Establishing a RDMS session in IPF

Establishing an IPF session to gain access to the IMDS relational database tables is fairly simple. After entering the IPF environment, use the following entries to set relevant environmental variables and process SQL statements:

```
2.1.1 $DATAMANAGER := RDMS
2.1.2 $SQLSCREEN := TRUE
2.1.3 SQL BEGIN THREAD ABC123 FOR APPL01 READ
2.1.4 SQL USE DEFAULT QUALIFIERS
```

Using SQL statements in IPF

Enter SQL SELECT statements in IPF the same way we've previously discussed, with a few minor changes. SELECT statements in IPF must be preceded with 'SQL'. If your SELECT statement is long and must continue on the next line, you must use the standard IPF continuation character '&'. Unlike Microsoft Access, the most common relational database, IPF uses standard wildcard characters. The Access equivalent of a '*' character is '%'. However, since this is a special character in IPF, it must be escaped (separated) by a '\' character. The same with the character '?', who's Access equivalent is '_'. When viewing multiple lines of output, it is best done in full screen mode. The following is an example of a simple SQL SELECT retrieval written in IPF:

1. SQL SELECT * from Standard_Rpt_Dsgntr & WHERE SRD_KEY LIKE 'A_A';
2. SQL SELECT COUNT(*) from Standard_Rpt_Dsgntr & WHERE SRD_KEY LIKE 'G\%';
3. SQL SELECT * from Standard_Rpt_Dsgntr & WHERE SRD_KEY in ('AAW','ATF');

The first SELECT statement will select all records where the SRD begins and ends in A from the Standard_Rpt_Dsgntr table. The second SELECT statement will count all records where the SRD begins with G from the same table. The third SELECT statement will select all records where the SRD equals AAW or ATF.

Ending the IPF RDMS session

To disconnect from the RDMS–2200 database and end the IPF SQL session, you simply enter the following command:

SQL END THREAD

If you forget to enter the command, you will receive an error message that can be safely ignored.

Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

246. Structured Query Language basics

1. The SQL SELECT statement is similar to which QLP command?

2. Describe what this SQL query will do: SELECT * FROM Org_Code.

3. What clause completes the SELECT statement?

4. What is the purpose of an aggregate function?

5. What clause sorts your SQL queries?

247. Using Structured Query Language for Query Language Processor reports

1. When you use the SQL SELECT statement within a QLP report, how do you enter the columns you want to display?

2. When does QLP recognize or know the SELECT statement in the QLP report refers to the RDMS tables?

3. What is the drawback to using the FOR TRANSFER clause option?

4. Where do you enter the FOR TRANSFER clause when you want to use this option in your QLP report?

248. Using Structured Query Language within interactive processing facility

1. List the four entries used to begin processing SQL statements in IPF.

2. What must precede SQL SELECT statements in IPF?

Answers to Self-Test Questions

235

1. A set of continuous characters.
2. Reserved and key words.
3. A character string enclosed by single quotes.
4. A sequence of one or more words, literals, or combination of both that must be supplied by the user to satisfy the syntactical requirements of a QLP statement.
5. Data identifier.
6. PIC IS 9(07).
7. Keywords that specify the action to be performed for a given clause.
8. Equal (=).
9. Key words that link relational operators in a logical sequence.
10. An expression that specifies the data conditions that must be met in order to retrieve or update data.

236

1. By referencing the syntax skeleton.
2. (1) f.
   (2) g.
   (3) i
   (4) d.
   (5) c.

(6) a.
(7) e.
(8) h.
(9) b.
(10) j.

237

1. @QLP.
2. MODE IS NO ECHO.
3. EXIT.

238

1. A tally of data item occurrences.
2. The contents of specific records, ranging from a specific data item to an entire record.
3. LIST BY COLUMN.
4. SORT DESC.
5. MASK ***-* = ‘T’ or MASK ***- = ‘T’.
6. REPEAT.
7. WHERE SAME.

239

1. It allows you to reference a unique name that you defined with the macro anytime you want.
2. To tell the QLP that you plan to define a macro and give an entity name to the macro.
3. To tell the QLP you are finished with your macro definition.

240

1. With the CALL command.
2. PROCEDURE and END PROCEDURE.

241

1. @@PRNT.
2. OUTPUT TO DEVICE DP0069.

242

1. REPORT.
2. It enables the user to specify the length and width of a page in terms of lines and columns.
3. Four lines top and bottom; zero left and right margins.
4. PAGE-COUNTER.
5. To specify the order in which a report is to be presented.
6. SELECT.
7. Defines the attributes (type data and length of field) of a derived name that was not explicitly declared.
8. To define the explicit attributes of a derived name.
9. CONTROLS.
10. MESSAGE.
11. LINE PLUS 2.
12. Only one per data identifier.
13. DETAIL clause.
14. END REPORT.

243

1. It assumes you want a DETAIL report.

2. The TO clause.
3. The number of lines per inch to be printed.
4. A WHERE clause.

244

1. Correct errors or modify an existing report.
2. To list the entire report or a particular line.
3. By entering the line number and transmitting.
4. GO and LIST.
5. Call up the line number, make the necessary changes, and transmit. (The line number must be preceded by an SOE character.).
6. LEAVE-EDIT or LEA.

245

1. Temporary.
2. User.
3. Permanent.
4. It lists the elements residing in a particular savefile, including the usage and time and date of creation.
5. It allows the user to copy a savefile entity from a QLP savefile or an external file into QLP savefile.
6. It copies a savefile entity from the temporary savefile into either a user or permanent savefile and then purges it from the temporary savefile.
7. It allows the user to erase an entity from a savefile.
8. It translates an English definition into a language the computer understands.

246

1. LIST.
2. Display all columns from the Org_Code table.
3. FROM.
4. It is used to summarize the data returned by a SELECT statement.
5. ORDER BY.

247

1. Each column must be listed separately.
2. It does so automatically when you invoke the RDMS environment in your INVOKE statement.
3. You can only generate the report to a file.
4. After the report name in the REPORT line of your QLP.

248

1. Use the following entries to set relevant environmental variables and process SQL statements:
   1) $DATAMANAGER := RDMS
   2) $SQLSCREEN := TRUE
   3) SQL BEGIN THREAD ABC123 FOR APPL01 READ
   4) SQL USE DEFAULT QUALIFIERS
2. “SQL”.

Unit Review Exercises

Note to Student: Consider all choices carefully, select the *best* answer to each question, and *circle* the corresponding letter. When you have completed all unit review exercises, transfer your answers to the Field Scoring Answer Sheet.

Do not return your answer sheet to AFCDA.

74. (235) Which is an example of an integer literal?
    a. 2.
    b. 2.5.
    c. +2.0.
    d. ‘A’.

75. (235) Which is an example of a Boolean operator?
    a. AND.
    b. PLUS.
    c. EQUAL.
    d. WHERE.

76. (235) The Query Language Processor (QLP) statement ESH-ENG-FLY-HRS = ‘01000’ represents
    a. a Boolean expression.
    b. an operand expression.
    c. a relational expression.
    d. a conditional qualifying expression.

77. (236) Which words in a Query Language Processor (QLP) syntax skeleton are mandatory if they are underlined?
    a. Proper.
    b. Generic.
    c. Uppercase.
    d. Lowercase.

78. (236) What character do you use at the end of a Query Language Processor (QLP) statement when you want to continue the command to the next line?
    a. Comma (,).
    b. Ellipsis (...).
    c. Asterisk (*).
    d. Semicolon (;).

79. (237) The Query Language Processor (QLP) statement “INVOKE FS-QLP–8821 DMS*CAMSDBG–7R1” means you want QLP to
    a. exit the data management system (DMS) database of Enterprise Location Code (ELC) 8821.
    b. access 8,821 schemas of the DMS database.
    c. exit 8,821 records of the DMS database.
    d. access the DMS database of ELC 8821.

80. (237) When more than one INVOKE statement is entered in one session, the second or subsequent INVOKE statement
    a. accesses all databases mentioned in all the INVOKE statements.
    b. automatically overrides the previous INVOKE statement.
    c. is ignored by Query Language Processor (QLP).
    d. accesses two databases at a time.

81. (237) Which Query Language Processor (QLP) sign-off procedure purges all reports that were *not* saved by use of the SAVE command?
    a. @@TERM.
    b. $$CLOSE.
    c. @FIN.
    d. EXIT.

82. (238) Which Query Language Processor (QLP) statement displays the contents of the maintenance personnel records (MPR)–251 accessed?
    a. LIST MPR–251.
    b. SORT MPR–251.
    c. MASK MPR–251.
    d. COUNT MPR–251.

83. (238) Which Query Language Processor (QLP) clause allows you to specify only portions of a data item as the selection criteria?
    a. LIST.
    b. SORT.
    c. MASK.
    d. LIST IN COLUMN.

84. (238) What Query Language Processor (QLP) command executes the previous action clause with a new WHERE clause?
    a. SAME.
    b. REPEAT.
    c. WHERE SAME.
    d. REPEAT SAME.

85. (239) You invoke a Query Language Processor (QLP) MACRO that you have previously defined and saved by
    a. using its name.
    b. using the INVOKE command.
    c. creating a MACRO definition again.
    d. entering MACRO followed by the name.

86. (240) The Query Language Processor (QLP) procedure definition consists of the mandatory statements PROCEDURE and
    a. TERMINATE PROCEDURE.
    b. LEAVE PROCEDURE.
    c. EXIT PROCEDURE.
    d. END PROCEDURE.

87. (241) When printing a Query Language Processor (QLP) output, what command do you use to activate a side-by printer?
    a. PRNT.
    b. PRINT.
    c. @PRNT.
    d. @@PRNT.

88. (241) The entry required to print a Query Language Processor (QLP) output to an enterprise output manager printer with a device ID of DP0069 is the command OUTPUT TO
    a. PRINT DP0069.
    b. DEVICE DP0069.
    c. SIDE-BY DP0069.
    d. ORIGINATOR DP0069.

89. (242) The Query Language Processor (QLP) clause that defines the explicit attributes of a derived name is
    a. DEFAULT PICTURE.
    b. CONTROLS.
    c. DECLARE.
    d. REPORT.

90. (242) Which Query Language Processor (QLP) clause sets up the body of a report?
    a. CONTROLS.
    b. DEFAULT.
    c. DETAIL.
    d. PAGE.

91. (243) The Query Language Processor (QLP) command that allows the user to specify an output destination is
    a. LINE.
    b. PAGE.
    c. DENSITY.
    d. GENERATE.

92. (244) Which is the proper format for listing line numbers 1100 through 2250 of a Query Language Processor (QLP) retrieval?
    a. LIST 1100,2250.
    b. LIST 1100 2250.
    c. LIST 1100*2250.
    d. LIST 1100–2250.

93. (245) What savefile is automatically assigned to Query Language Processor (QLP) when the INVOKE command is processed?
    a. User.
    b. Temporary.
    c. Executable.
    d. Permanent.

94. (245) Which Query Language Processor (QLP) savefile command is used to list all or specific entities?
    a. SAVE.
    b. COPY.
    c. PRINT.
    d. PURGE.

95. (245) In which mode can the Query Language Processor (QLP) COMPILE command *not* be used?
    a. LIST.
    b. EDIT.
    c. SAVE.
    d. DEFINE.

96. (246) In Structured Query Language (SQL), what do you normally enter in the select list of a SELECT statement?
    a. Row names.
    b. Table names.
    c. Column names.
    d. Boolean expressions.

97. (246) In Structured Query Language (SQL), what special operator allows you to use wild card characters in your condition criteria?
    a. In.
    b. Like.
    c. Where.
    d. Between.

98. (247) If used on a Query Language Processor (QLP) report, on what line do you enter the clause FOR TRANSFER?
    a. DETAIL.
    b. INVOKE.
    c. SELECT.
    d. REPORT.

99. (248) When entering SELECT statements in Interactive Processing Facility (IPF), what *must* precede the SELECT statement?
    a. An Asterisk.
    b. WHERE.
    c. SQL.
    d. SRD.

100. (248) What is the standard Interactive Processing Facility (IPF) continuation character?
    a. ‘&’.
    b. ‘*’.
    c. ‘%’.
    d. ‘>‘.

Student Notes

Glossary

Terms

Batch processing—The accumulation of assets to achieve economic process quantity inducted into maintenance.

Central database—A Unisys platform designed and developed to consolidate 106 separate Core Automated Maintenance System (CAMS) databases and data. The central data base (CDB) can be accessed by maintenance users worldwide for enterprise and AEF visibility.

Enterprise—A coordinated group of business organizations functioning together to achieve a common goal. Enterprise-wide within the Air Force means Air Force-wide.

Graphical user interface—A program interface that takes advantage of the computer's graphics capabilities to make the program easier to use by providing reference data to the user as "pick lists" (called dropdowns, pop-ups, or dynamic retrievals).

Infrastructure—The underlying foundation or basic framework of a system.

Integration—Identifies functions and data commonalities among systems and eliminates redundancy by aggregating these aspects into a reduced number of modernized data systems in a shared environment.

Legacy System—Existing information systems, such as CAMS, Core Automated Maintenance System For Mobility (CAMS-FM/G081), Reliability and Maintainability Information System (REMIS), Comprehensive Engine Management System (CEMS), and Reliability, Availability, and Maintainability of Pods (RAMPOD) that predate the GCSS-AF and DII COE initiatives. These systems must be maintained and supported until a modernized system is fielded.

Migration—The process used to move legacy applications to the integrated logistics domain.

Stand-alone—The ability to store, manipulate, retrieve, and process data/information with minimal power over a specified period of time.

Total asset visibility—Ability to access all data assets in an enterprise.

Abbreviations and Acronyms

1NF	first normal form
2NF	second normal form
3NF	third normal form
ACC	Air Combat Command
ADS	Automated Data System
AEF	Air & Space Expeditionary Force
AF	Air Force
AFB	Air Force Base
AFCSM	Air Force Computer System Manual
AFI	Air Force instruction
AFLCMC	Air Force Life Cycle Management Center

AFMAN	Air Force manual
AFMC	Air Force Materiel Command
AFP	781 Automated Forms Discrepancies Print
AFR	Air Force Reserve
AGE	aerospace ground equipment
AHE	automated history
AIS	Automated Information System
AKH	a standard reporting designator representing an F–16C fighter
ALC	Air Logistics Center
ANG	Air National Guard
ARC	automated records check for equipment-IDs
ASC	ascending
ASCII	American Standard Code for Information Interchange
AT	action taken
BES	Business Enterprise Systems
BHM	a standard reporting designator representing a satellite radio
BNCC	base network control center
C-E	communications-electronics
CAMS	Core Automated Maintenance System
CALC	calculation
CDB	central database
CDC	career development course
CEMS	Comprehensive Engine Management System
CHG	change
COBOL	common business oriented language
DB	databse
DBA	database administrator
DBE	database editor
DBK	database key
DBLOOK	database look
DBM	database manager
DBP	database pointer
DDL	deviation detail listing or data definition language
DDN	Defense Data Network
DECC	Defense Enterprise Computing Center
DESC	descending from highest to lowest

DFS	REMIS inventory file output
DIREP	difficulty report
DISA	Defense Information Systems Agency
DMB	debriefing MOA builder
DMR	data management routine
DMS	Data Management System
DMU	data management utility
DOD	Department of Defense
DOM	documented maintenance list
DRC	debriefing sortie recap
DSD	data system designator
ECL	Executive Control Language
ELC	enterprise location code
EMOC	Enhanced Maintenance Operations Center
ERP	equipment record print
FAS	field assistance service
FS	follows the schema name
FSA	functional system analyzer
FSP	functional system program
FTP	file transfer protocol
FTR	code 3 fix time report
FUD	file update
GCSS-AF	Global Combat Support System –Air Force
GENRUN	generated runstream
GRD	pay-grade
GUI	graphical user interface
HICS	Enterprise Services
HOWMAL	how malfunctioned
HUM	heads-up message
ICI	interactive communication interface
ID	identification/identifier
IFM	REMIS file manager
IGDM	interrupt guard mode
IMDS	Integrated Maintenance Data System
IMIS	Integrated Maintenance Information System
INDX	index

IPF	interactive processing facility
IQU	interactive query utility
ISO	international syntactical operator
ISP	inspection
ITSRD	Information Technology Systems Requirements Document
JCN	job control numbers
JDD	job data documentation
JSM	JDD send MOA
JST	job standards
LAN	local area network
LRUD	line replaceable unit data record
MAJCOM	major command
MDS	mission design series
Mgr	manager
MICAP	mission capable
MIS	Maintenance Information System
MOA	memorandum of agreement
MPR	maintenance personel record
mx	maintenance
NAF	Numbered Air Force
NATO	North Atlantic Treaty Organization
NFS	network file system
NOSC	network operating security centers
NSN	national stock number
Num	number
OI	operating instructions
OKC	Oklahoma City
OPR	office of primary responsibility
OTR	operating time reinitialization
PAFSC	primary Air Force specialty code
PC	personal computer
PIC	picture
PNSN	part number/serial number
PRA	planning requirements
PRD	pilot reported discrepancies report
PSUPRB	pseudo processor for batch

PTI	parts tracked inquiry
QLP	Query Language Processor
QMH	maintenance history report
RAMOS	name of the user savefile
RAMPOD	Reliability, Availability, and Maintainability for Pod
RB	rollback
RBAC	role based access clearance
RCS	report control symbol
RDMS	Relational Data Management System
REI	REMIS error inquiry
REM	REMIS error correction menu
REMIS	Reliability and Maintainability Information System
RFM	REMIS error files maintenance
RFO	REMIS file output
RFS	REMIS file separator
RIN	REMIS incoming DDN
RIR	REMIS inventory reconciliation
RIS	IMDS/REMIS reconciliation
RJETS	Remote Job Entry Terminal System
SAE	shop equipment operational inquiry
SAN	system advisory notice
SBLC	standard base-level computer
SBSS	Standard Base-Level Supply System
SBU	sensitive but unclassified
SHD	significant historical data
SHM	significant historical maintenance
SIMAN	site management
SOE	start of entry
SQL	Structured Query Language
SRD	standard reporting designator
STL	serial number detail listing format #1, input for equipment-IDs
TAV	total asset visibility
TBE	transfer between ELC
TCTO	time compliance technical order
TIP	transaction interface package
TIS	timer interrupt switch

TPR	table validation problem reporting
TRE	equipment transfer-format 1
TRIC	transaction identification code
UDS	Universal Data System
UDSMON	Universal Data System monitor
URWG	User Requirements Working Group
UVR	unit variable record
WAH	monthly man-hour summary
WCR	work center record
WebTS	web transaction server
WHC	work-hour update
WRG	within range
WUC	work unit code

Student Notes

AFSC 2R051
2R051 02 1704
Edit Code 07