9.0 FEMIS Oracle Database

Oracle Release v8.1.6, a commercial database management system (DBMS), manages the relational database in FEMIS. The distributed processing features of Oracle are utilized to produce a multi-server distributed data architecture. Data replication is widely used to provide a local copy of most shared tables. This replication is important because it allows an EOC to operate autonomously in case the links to other EOCs are not operational. Also, performance is enhanced because the local tables are located on the local database.

The FEMIS relational database is made up of approximately 190 tables. The FEMIS logical data model describes graphically what information is present and how the data objects are interrelated. The model represents a large collection of general-purpose tables, evacuation data, GIS tables, and dispersion tables. Additional information about the data model is available in the ***Data Management Guide for FEMIS Version 1.5***.

Based on design efforts and testing results, each relational database table is local to an EOC or shared with the other EOCs. Data in the local tables can be accessed only by users logged in to that EOC. The data in shared tables is available to several EOCs. Details of data placement are made transparent to the FEMIS users, so the FEMIS database appears to be a single, unified collection of tables. This physical design of the Oracle database is provided as a part of database implementation and should be applicable to all CSEPP sites. More details about the DBMS are provided in the ***Data Management Guide for FEMIS Version 1.5***.

For information on the recommended backup strategy and performing Oracle database backups, see Section 13.0, Backup Strategy for FEMIS.

9.1 Data Description

When creating the first database for a new site or when making major database modifications, it is necessary to create the database structure from scripts and load basic data so the FEMIS application can operate. For most situations, the new database will be created in a development facility and then packaged so it can be delivered to the operational site. Section 3.0, Building the Initial Information, in the ***Data Management Guide for FEMIS Version 1.5***, describes how a new database is installed at the site.

For cases where the FEMIS software is updated to a new release, the existing site database can be updated, if necessary, to support new capabilities. In this case, one or more scripts are developed to make the data structure and/or data content modifications. Instead of recreating the database, the scripts are run to make it compatible with the new FEMIS version of software.

9.2 Replication

Oracle provides several ways to share data between EOC servers in a distributed, multiserver environment. When the site environment is not tightly controlled by one group, it makes sense to operate in a mode where operations can proceed in each server independent of what the other servers are doing. To make this happen, data sharing has to be asynchronous so that data changes in one server are not dependent on making similar changes in the other servers in the same transaction.

Shared data record changes are propagated to other servers using event driven, push replication built from Oracle's Replication Management application program interface (API). This method is currently used by FEMIS since it is asynchronous, flexible, and uses much less processing and network resources than the previous replication scheme. The database where the change occurs creates a deferred remote procedure call `RPC` that is placed in the `Deferred Queue`. This queue is pushed to remote servers thus causing the `RPC` to execute and pass the data change parameters. Then the remote server executes a request for the updates. In FEMIS versions before 1.4.5, remote servers polled at a 45-second rate looking for data changes. Due to constant polling, all parts of the system had to be available day and night. The new push replication does not do any work until a data change occurs. This reduces the polling overhead at the remote sites and the request traffic on the network.

When the database is installed at a site, either a configuration with all EOCs on a single server or a configuration of several servers is chosen. Single server configurations are used in development and test centers, but all of the CSEPP sites use multi-servers. In the former case, there is no replication since the data is shared by Oracle views. If the multiple server option is used, then scripts delivered with the database are run to create the data sharing objects (see Section 2.9, Creating or Updating the FEMIS Database, in the ***Installation Guide for FEMIS Version 1.5***).

Once the distributed objects are created, replication can be initiated by running the scripts provided. Before doing this, establish that the other servers at the site are in a ready state to be able to participate in data sharing. If a local site is going to be down for several hours or more, replication can be stopped at the other servers by running the stop scripts.

9.3 Database Maintenance

FEMIS has a monitoring tool, called AutoRecovery, that continually checks the status of the EOC's critical hardware and software components. When failures are detected or thresholds are exceeded, warning messages are sent to the System and Database Administrators. In certain cases, this tool attempts to remedy problems directly. In other cases, the System and Database Administrators must take manual actions to remedy the problems or take measures to correct situations that caused threshold warnings.

AutoRecovery monitors the portions of the database that are most likely to have problems. In most cases, it tries to warn the Database Administrator before the problem causes a serious failure; this is done by thresholds and looking for symptoms of problems, such as network interruptions. In cases where the problem exists and can be resolved, an immediate fix is attempted.

The local database and the database listener are checked each cycle. If the listener is down, a restart is immediately attempted. A database failure is a serious condition that must be analyzed before a restart is attempted since the restart may result in bigger problems. If the database is not functioning, the Database Administrator should look in Oracle's alert log to determine the cause. If the condition is no longer present or has been fixed, the database can be restarted from a command line sequence as follows:

```
> su - oracle      <If not already logged in as the oracle user>
> <pwd>
>svrmgrl
>connect internal
>startup
```

Section 2.0, FEMIS Monitoring Tools, describes the operation of this AutoRecovery tool and other tools that are available to troubleshoot and repair the database. In the ***Installation Guide for FEMIS Version 1.5***, Section 2.12 Installing the FEMIS AutoRecovery System, discusses how to install these tools and configure AutoRecovery to support the site.

9.4 How AutoRecovery Works with the Database

AutoRecovery monitors the database tablespaces and warns when the thresholds are exceeded. When these warnings are present for an hour or longer, the Database Administrator should take action to prevent the tablespace from reaching the full (or 100% used) condition that will cause a serious database failure. The common causes of tablespace increase are that more data has been added intentionally or some old data, which is not essential, exists in the database. The Database Administrator should check to see if old data is present and if so, remove it. This will cause the tablespace warnings to cease and have the added benefit of increasing system performance by reducing table sizes. The two most common old data types are meteorological data that has not been archived and extra, nonessential exercises.

If the system has recently added new records to the database intentionally, then one or more tablespace sizes should be increased to give a margin for additional growth. If this is necessary, find the name(s) of the data files from the AutoRecovery log and enter the following commands logged in as the UNIX `oracle` user:

```
>svrmgrl
SVRMGR> connect internal
SVRMGR> alter database datafile '<full path of new file>' resize xxM;
SVRMGR> exit
```

A real example to increase the size of the `FMAIN` datafile to add 100 MB data is

```
SVRMGR> alter database datafile  '/files2/app/oracle/oradata/fi1/fmain01.dbf'
resize 100M;
```

AutoRecovery monitors remote servers and then sends warnings if problems are seen. If these problems persist beyond a threshold count, a `Disable Node` command is sent to the database to stop pushing changes to the bad server and also to stop any update processing from the bad node. This will normally prevent the local database from suffering problems. When AutoRecovery can communicate with the disabled node reliably, an `Enable Node` command is sent to the local database to reestablish replication.

Database replication is dependent on all components at the network functioning properly including communications, servers, and database. When some failure occurs, replication may not be able to copy database changes. Oracle has built in error recovery that will keep trying up to 16 times, but if all tries are unsuccessful, Oracle will stop and declare that replication is broken. AutoRecovery monitors local replication processing and will attempt to fix errors when they are detected.

There are also sets of fix scripts that can be used to manually correct replication problems. Your Database Administrator should look over these scripts and become familiar with their use. Under normal conditions, AutoRecovery will fix all replication problems.