CHENNAI INSTITUTE OF TECHNOLOGY
(Autonomous)
(Affiliated to Anna University, Approved by AICTE, Accredited by NAAC & NBA)
Sarathy Nagar, Kundrathur, Chennai – 600069, India.

UNIT V - LECTURE NOTES

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CCS341 DATA WAREHOUSING

III CSE/VI SEMESTER

UNIT V SYSTEM & PROCESS MANAGERS

Data Warehousing System Managers: System Configuration Manager- System Scheduling Manager - System Event Manager - System Database Manager - System Backup Recovery Manager - Data Warehousing Process Managers: Load Manager – Warehouse Manager- Query Manager – Tuning – Testing

1.Data Warehousing System Managers:

System management is mandatory for the successful implementation of a data warehouse. The most important system managers are −

- System configuration manager
- System scheduling manager
- System event manager
- System database manager
- System backup recovery manager

1.1 System Configuration Manager

- The system configuration manager is responsible for the management of the setup and configuration of data warehouse.
- The structure of configuration manager varies from one operating system to another.
- In Unix structure of configuration, the manager varies from vendor to vendor.
- Configuration managers have single user interface.
- The interface of configuration manager allows us to control all aspects of the system. The most important configuration tool is the I/O manager.

1.2 System Scheduling Manager

System Scheduling Manager is responsible for the successful implementation of the data warehouse. Its purpose is to schedule ad hoc queries. Every operating system has its own scheduler with some form of batch control mechanism. The list of features a system scheduling manager must have is as follows −

- Work across cluster or MPP boundaries
- Deal with international time differences
- Handle job failure
- Handle multiple queries
- Support job priorities
- Restart or re-queue the failed jobs
- Notify the user or a process when job is completed
- Maintain the job schedules across system outages
- Re-queue jobs to other queues
- Support the stopping and starting of queues
- Log Queued jobs
- Deal with inter-queue processing

Note − The above list can be used as evaluation parameters for the evaluation of a good scheduler.

Some important jobs that a scheduler must be able to handle are as follows −

- Daily and ad hoc query scheduling
- Execution of regular report requirements
- Data load
- Data processing
- Index creation
- Backup
- Aggregation creation
- Data transformation

Note − If the data warehouse is running on a cluster or MPP architecture, then the system scheduling manager must be capable of running across the architecture.

1.3 System Event Manager

The event manager is a kind of a software. The event manager manages the events that are defined on the data warehouse system. We cannot manage the data warehouse manually because the structure of data warehouse is very complex. Therefore we need a tool that automatically handles all the events without any intervention of the user.

Note − The Event manager monitors the events occurrences and deals with them. The event manager also tracks the myriad of things that can go wrong on this complex data warehouse system.

Events

Events are the actions that are generated by the user or the system itself. It may be noted that the event is a measurable, observable, occurrence of a defined action.

Given below is a list of common events that are required to be tracked.

- Hardware failure
- Running out of space on certain key disks
- A process dying
- A process returning an error
- CPU usage exceeding an 805 threshold
- Internal contention on database serialization points
- Buffer cache hit ratios exceeding or failure below threshold
- A table reaching to maximum of its size
- Excessive memory swapping
- A table failing to extend due to lack of space
- Disk exhibiting I/O bottlenecks
- Usage of temporary or sort area reaching a certain thresholds
- Any other database shared memory usage

The most important thing about events is that they should be capable of executing on their own. Event packages define the procedures for the predefined events. The code associated with each event is known as event handler. This code is executed whenever an event occurs.

1.4 System and Database Manager

System and database manager may be two separate pieces of software, but they do the same job. The objective of these tools is to automate certain processes and to simplify the execution of others. The criteria for choosing a system and the database manager are as follows −

- increase user's quota.
- assign and de-assign roles to the users
- assign and de-assign the profiles to the users
- perform database space management
- monitor and report on space usage
- tidy up fragmented and unused space
- add and expand the space
- add and remove users
- manage user password
- manage summary or temporary tables
- assign or deassign temporary space to and from the user
- reclaim the space form old or out-of-date temporary tables
- manage error and trace logs
- to browse log and trace files
- redirect error or trace information
- switch on and off error and trace logging
- perform system space management
- monitor and report on space usage
- clean up old and unused file directories
- add or expand space.

1.5 System Backup Recovery Manager

The backup and recovery tool makes it easy for operations and management staff to back-up the data. Note that the system backup manager must be integrated with the schedule manager software being used. The important features that are required for the management of backups are as follows −

- Scheduling
- Backup data tracking
- Database awareness

Backups are taken only to protect against data loss. Following are the important points to remember −

- The backup software will keep some form of database of where and when the piece of data was backed up.
- The backup recovery manager must have a good front-end to that database.
- The backup recovery software should be database aware.

- Being aware of the database, the software then can be addressed in database terms, and will not perform backups that would not be viable.

2.Data Warehousing Process Managers:

Process managers are responsible for maintaining the flow of data both into and out of the data warehouse. There are three different types of process managers −

- Load manager
- Warehouse manager
- Query manager

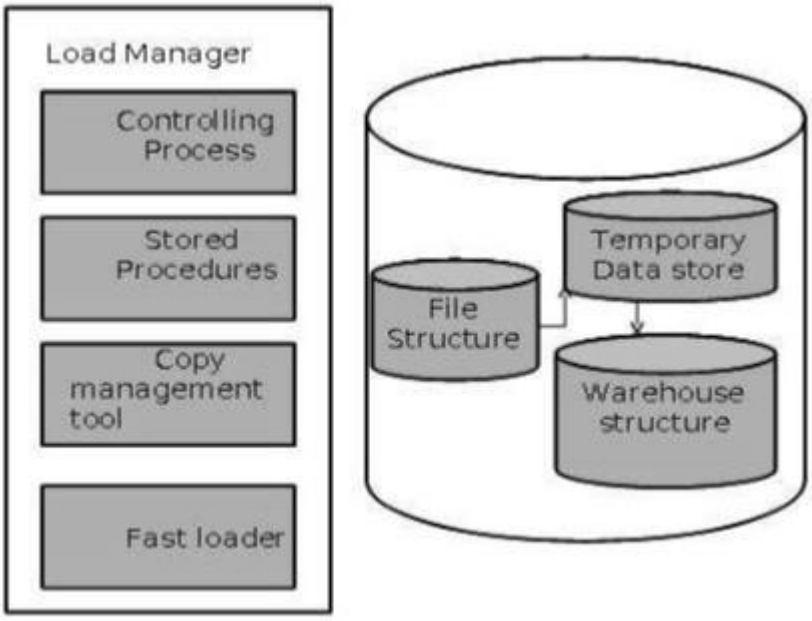
2.1 Data Warehouse Load Manager

Load manager performs the operations required to extract and load the data into the database. The size and complexity of a load manager varies between specific solutions from one data warehouse to another.

2.1.1Load Manager Architecture

The load manager does performs the following functions −

- Extract data from the source system.
- Fast load the extracted data into temporary data store.
- Perform simple transformations into structure similar to the one in the data warehouse.



2.1.2.Extract Data from Source

The data is extracted from the operational databases or the external information providers. Gateways are the application programs that are used to extract data. It is supported by underlying DBMS and allows the client program to generate SQL to be executed at a server.

Open Database Connection (ODBC) and Java Database Connection (JDBC) are examples of gateway.

2.1.3 Fast Load

- In order to minimize the total load window, the data needs to be loaded into the warehouse in the fastest possible time.
- Transformations affect the speed of data processing.
- It is more effective to load the data into a relational database prior to applying transformations and checks.
- Gateway technology is not suitable, since they are inefficient when large data volumes are involved.

2.1.4 Simple Transformations

While loading, it may be required to perform simple transformations. After completing simple transformations, we can do complex checks. Suppose we are loading the EPOS sales transaction, we need to perform the following checks −

- Strip out all the columns that are not required within the warehouse.
- Convert all the values to required data types.

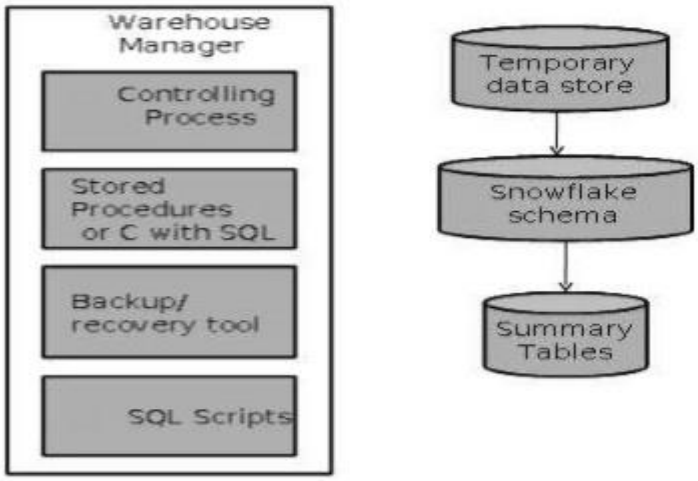
2.2 Warehouse Manager:

The warehouse manager is responsible for the warehouse management process. It consists of a third-party system software, C programs, and shell scripts. The size and complexity of a warehouse manager varies between specific solutions.

2.2.1 Warehouse Manager Architecture

A warehouse manager includes the following −

- The controlling process
- Stored procedures or C with SQL
- Backup/Recovery tool
- SQL scripts



2.2.2 Functions of Warehouse Manager

A warehouse manager performs the following functions −

- Analyzes the data to perform consistency and referential integrity checks.
- Creates indexes, business views, partition views against the base data.
- Generates new aggregations and updates the existing aggregations.
- Generates normalizations.
- Transforms and merges the source data of the temporary store into the published data warehouse.
- Backs up the data in the data warehouse.
- Archives the data that has reached the end of its captured life.

Note − A warehouse Manager analyzes query profiles to determine whether the index and aggregations are appropriate.

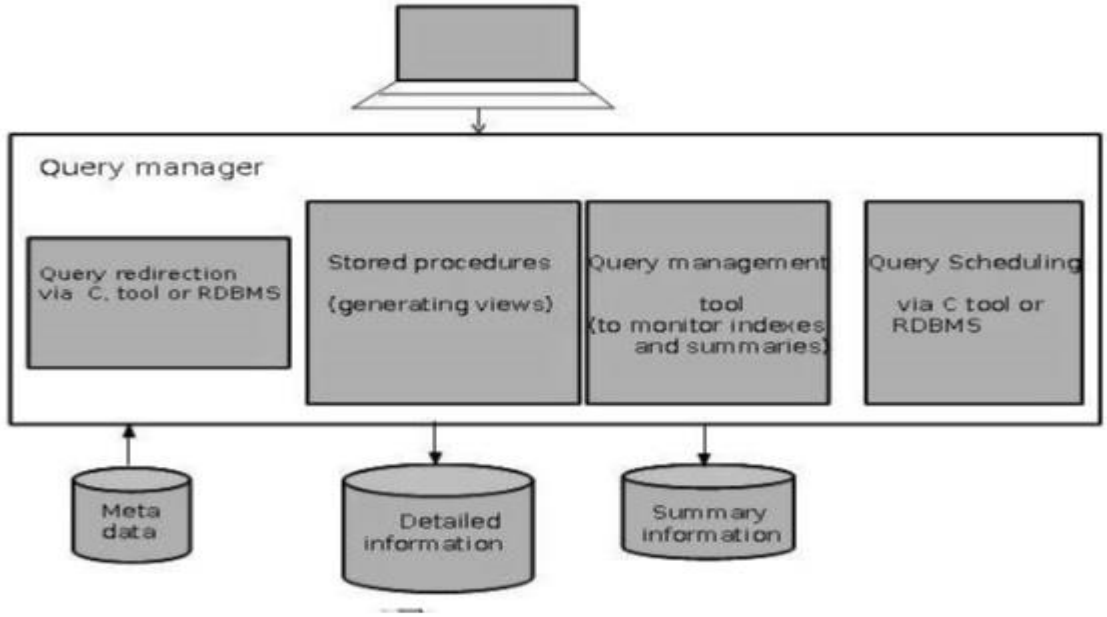
2.3. Query Manager:

The query manager is responsible for directing the queries to suitable tables. By directing the queries to appropriate tables, it speeds up the query request and response process. In addition, the query manager is responsible for scheduling the execution of the queries posted by the user.

2.3.1 Query Manager Architecture

A query manager includes the following components −

- Query redirection via C tool or RDBMS
- Stored procedures
- Query management tool
- Query scheduling via C tool or RDBMS
- Query scheduling via third-party software



2.3.2 Functions of Query Manager

- It presents the data to the user in a form they understand.
- It schedules the execution of the queries posted by the end-user.
- It stores query profiles to allow the warehouse manager to determine which indexes and aggregations are appropriate.

3.Data Warehouse Tuning:

A data warehouse keeps evolving and it is unpredictable what query the user is going to post in the future. Therefore it becomes more difficult to tune a data warehouse system. In this chapter, we will discuss how to tune the different aspects of a data warehouse such as performance, data load, queries, etc.

3.1 Difficulties in Data Warehouse Tuning

Tuning a data warehouse is a difficult procedure due to following reasons −

- Data warehouse is dynamic; it never remains constant.
- It is very difficult to predict what query the user is going to post in the future.
- Business requirements change with time.
- Users and their profiles keep changing.
- The user can switch from one group to another.
- The data load on the warehouse also changes with time.

Note − It is very important to have a complete knowledge of data warehouse.

3.2 Performance Assessment

Here is a list of objective measures of performance −

- Average query response time
- Scan rates
- Time used per day query
- Memory usage per process
- I/O throughput rates

Following are the points to remember.

- It is necessary to specify the measures in service level agreement (SLA).
- It is of no use trying to tune response time, if they are already better than those required.
- It is essential to have realistic expectations while making performance assessment.
- It is also essential that the users have feasible expectations.
- To hide the complexity of the system from the user, aggregations and views should be used.
- It is also possible that the user can write a query you had not tuned for.

3.3 Data Load Tuning

Data load is a critical part of overnight processing. Nothing else can run until data load is complete. This is the entry point into the system.

Note − If there is a delay in transferring the data, or in arrival of data then the entire system is affected badly. Therefore it is very important to tune the data load first.

There are various approaches of tuning data load that are discussed below −

- The very common approach is to insert data using the **SQL Layer**. In this approach, normal checks and constraints need to be performed. When the data is inserted into the table, the code will run to check for enough space to insert the data. If sufficient space is not available, then more space may have to be allocated to these tables. These checks take time to perform and are costly to CPU.
- The second approach is to bypass all these checks and constraints and place the data directly into the preformatted blocks. These blocks are later written to the database. It is faster than the first approach, but it can work only with whole blocks of data. This can lead to some space wastage.
- The third approach is that while loading the data into the table that already contains the table, we can maintain indexes.
- The fourth approach says that to load the data in tables that already contain data, **drop the indexes & recreate them** when the data load is complete. The choice between the third and the fourth approach depends on how much data is already loaded and how many indexes need to be rebuilt.

3.4 Integrity Checks

Integrity checking highly affects the performance of the load. Following are the points to remember −

- Integrity checks need to be limited because they require heavy processing power.
- Integrity checks should be applied on the source system to avoid performance degrade of data load.

3.5 Tuning Queries

We have two kinds of queries in data warehouse −

- Fixed queries
- Ad hoc queries

3.5.1Fixed Queries

Fixed queries are well defined. Following are the examples of fixed queries −

- regular reports
- Canned queries
- Common aggregations

Tuning the fixed queries in a data warehouse is same as in a relational database system. The only difference is that the amount of data to be queried may be different. It is good to store the most successful execution plan while testing fixed queries. Storing these executing plan

will allow us to spot changing data size and data skew, as it will cause the execution plan to change.

Note − We cannot do more on fact table but while dealing with dimension tables or the aggregations, the usual collection of SQL tweaking, storage mechanism, and access methods can be used to tune these queries.

3.5.2 Ad hoc Queries

To understand ad hoc queries, it is important to know the ad hoc users of the data warehouse. For each user or group of users, you need to know the following −

- The number of users in the group
- Whether they use ad hoc queries at regular intervals of time
- Whether they use ad hoc queries frequently
- Whether they use ad hoc queries occasionally at unknown intervals.
- The maximum size of query they tend to run
- The average size of query they tend to run
- Whether they require drill-down access to the base data
- The elapsed login time per day
- The peak time of daily usage
- The number of queries they run per peak hour

Points to Note

- It is important to track the user's profiles and identify the queries that are run on a regular basis.
- It is also important that the tuning performed does not affect the performance.
- Identify similar and ad hoc queries that are frequently run.
- If these queries are identified, then the database will change and new indexes can be added for those queries.
- If these queries are identified, then new aggregations can be created specifically for those queries that would result in their efficient execution.

4. Data Warehousing - Testing

Testing is very important for data warehouse systems to make them work correctly and efficiently. There are three basic levels of testing performed on a data warehouse −

- Unit testing
- Integration testing
- System testing

4.1 Unit Testing

- In unit testing, each component is separately tested.
- Each module, i.e., procedure, program, SQL Script, Unix shell is tested.
- This test is performed by the developer.

4.2 Integration Testing

- In integration testing, the various modules of the application are brought together and then tested against the number of inputs.
- It is performed to test whether the various components do well after integration.

4.3 System Testing

- In system testing, the whole data warehouse application is tested together.
- The purpose of system testing is to check whether the entire system works correctly together or not.
- System testing is performed by the testing team.
- Since the size of the whole data warehouse is very large, it is usually possible to perform minimal system testing before the test plan can be enacted.

4.4 Test Schedule

First of all, the test schedule is created in the process of developing the test plan. In this schedule, we predict the estimated time required for the testing of the entire data warehouse system.

There are different methodologies available to create a test schedule, but none of them are perfect because the data warehouse is very complex and large. Also the data warehouse system is evolving in nature. One may face the following issues while creating a test schedule −

- A simple problem may have a large size of query that can take a day or more to complete, i.e., the query does not complete in a desired time scale.
- There may be hardware failures such as losing a disk or human errors such as accidentally deleting a table or overwriting a large table.

Note − Due to the above-mentioned difficulties, it is recommended to always double the amount of time you would normally allow for testing.

4.5 Testing Backup Recovery

Testing the backup recovery strategy is extremely important. Here is the list of scenarios for which this testing is needed −

- Media failure
- Loss or damage of table space or data file
- Loss or damage of redo log file
- Loss or damage of control file
- Instance failure
- Loss or damage of archive file
- Loss or damage of table
- Failure during data failure

4.6 Testing Operational Environment

There are a number of aspects that need to be tested. These aspects are listed below.

- **Security** − A separate security document is required for security testing. This document contains a list of disallowed operations and devising tests for each.
- **Scheduler** − Scheduling software is required to control the daily operations of a data warehouse. It needs to be tested during system testing. The scheduling software requires an interface with the data warehouse, which will need the scheduler to control overnight processing and the management of aggregations.
- **Disk Configuration.** − Disk configuration also needs to be tested to identify I/O bottlenecks. The test should be performed with multiple times with different settings.
- **Management Tools.** − It is required to test all the management tools during system testing. Here is the list of tools that need to be tested.
    - Event manager
    - System manager
    - Database manager
    - Configuration manager
    - Backup recovery manager

4.7 Testing the Database

The database is tested in the following three ways −

- **Testing the database manager and monitoring tools** − To test the database manager and the monitoring tools, they should be used in the creation, running, and management of test database.
- **Testing database features** − Here is the list of features that we have to test −
    - Querying in parallel
    - Create index in parallel
    - Data load in parallel
- **Testing database performance** − Query execution plays a very important role in data warehouse performance measures. There are sets of fixed queries that need to be run regularly and they should be tested. To test ad hoc queries, one should go through the user requirement document and understand the business completely. Take time to test the most awkward queries that the business is likely to ask against different index and aggregation strategies.

4.8 Testing the Application

- All the managers should be integrated correctly and work in order to ensure that the end-to-end load, index, aggregate and queries work as per the expectations.
- Each function of each manager should work correctly
- It is also necessary to test the application over a period of time.
- Week end and month-end tasks should also be tested.

Logistic of the Test

The aim of system test is to test all of the following areas −

- Scheduling software
- Day-to-day operational procedures
- Backup recovery strategy
- Management and scheduling tools
- Overnight processing
- Query performance

Note − The most important point is to test the scalability. Failure to do so will leave us a system design that does not work when the system grows.