Relational Database Management Systems – Oct/Nov 2013

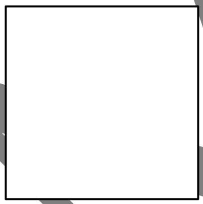
I. Section-A: 5 X 4 =20 Marks

1. Database Development

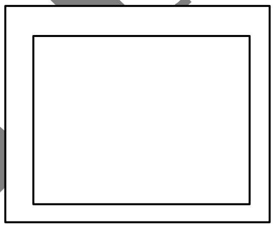
DDLC (Database Development Life Cycle):
It is a process for designing, implementing and maintaining a database system.
It consists of six stages:
1. Preliminary design
2. Feasibility design
3. Requirements definition
4. Conceptual design
5. Implementation
6. Database evaluation and maintenance.

2. Weak & Strong Entity

Strong Entity
A strong relationship also known, as identifying relationship, which is not dependent upon others.



Weak Entity
It is otherwise called as non-identifying relationship, which is dependent upon others.



3. Developing Client Application.

Out of Syllabus

4. DBA Goals.

A database must be protected from accidents, such as input or programming errors, from malicious use of the database, and from hardware of software failures that corrupt data. Protection from accidents that cause data inaccuracies is part of the goal of maintaining data integrity. These accidents include failures during transaction processing. Logical errors that violate the assumption that transactions preserve database consistency constraints, and anomalies due to concurrent access to the database (concurrent processing).

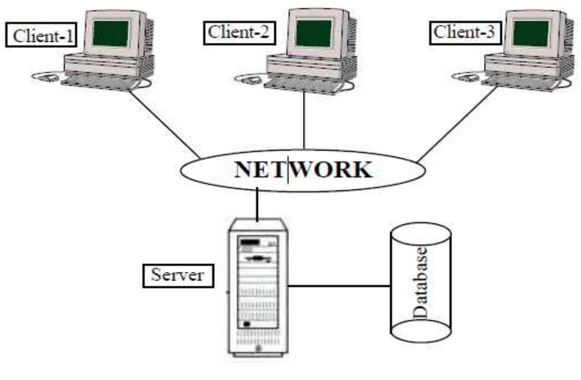
Protecting the database from unauthorized or malicious use is termed data security. Although the dividing line between data integrity and data security is not precise, a working definition is as follows:

- Integrity is concerned with making certain that operations performed by users are correct and maintain database consistency.
- Security is concerned with limiting users to performing only those operations that are allowed.
- Having Knowledge of SQL.
- Having Knowledge of Operating System
- Database Initialization: Database initialization is an important part of application deployment. Typically a DBA applies a set of SQL scripts to initialize a database or perform an upgrade.
- Able to create database: The Oracle database administrator must be able to create database objects in its tablespace (either unlimited or with a space quota) with privileges to create session, table, procedure, and views.
- Perform Backup : Backup and recovery procedures are tested regularly to assure their effectiveness in restoring the database after any disruption of service/ a disaster plan has been drawn up and is tested periodically to make sure it works.

The possibility of hardware or software failure requires that database recovery procedures be implemented as well. That is, means must be provided to restore databases that have been corrupted by system malfunctions to a consistent state.

5. Client-Server System.

Client/server architecture may be either two-tier or three-tier. In a two-tier architecture, the server performs database functions and the clients perform the presentation (user interface) functions. Either the server or the clients may perform business functions. The term fat client refers to an arrangement where the clients perform the business functions. If the business functions reside on the server, each client is called a thin client. In a three-tier architecture, the clients perform the presentation functions, a database server performs the database functions, and separate computers, called application servers, perform the business functions and act as interface between clients and database server.



6. Database Recovery.

Information stored on computer media is subject to loss or corruption caused by a wide range of events, it is important to provide means for resorting correct data to the database. Restoring the database to precisely the same state that existed at the time of system failure is not always possible, but database recovery procedures can restore the database to the state that existed shortly before the failure and identify the status of transaction processing at the time of the failure. With this capability, unprocessed transactions can be processed against the restored database to bring it back to a fully current status.

Sources of Failure:

A useful classification of failure types includes the following:

1. System errors: the system has entered an undesirable state, such as deadlock, which prevents the program from continuing with normal processing. This type of failure may or may not result in corruption of data files.
2. Hardware failures: Two of the most common types of hardware failure and loss of transmission capability over a transmission link. In the former case, the cause usually results from the disk read/write head coming in physical contact with the disk surface.
3. Logical errors: Bad data or missing data are common conditions that may preclude a program's continuing with normal execution.

Recovery Procedures: -

To maintain data integrity, a transaction must be in one of the two following states:

1. Aborted: A transaction may not always complete its process successfully. To be sure the incomplete transaction will not affect the consistent state of the database, such transactions must be aborted, and restoring the database to the state it was in before the transaction in question began execution. Such restoration is achieved by rollback.
2. Committed: A transaction that successfully completes its processing is said to be committed. A committed transaction always leaves the database in a new consistent state.

The LOG is a history of all the changes made to the database as well as the status of
each transaction. LOG information is stored on a mythical "stable storage" that survives all failures.
A recovery strategy can be pursued by one of two approaches
1) Logging with deferred updates: In this technique, all data is stored in database. after committed statement, then use redo ( ) operation. To perform commit statement again.

2) logging with Immediate database Modification: In this technique, all data is updated on database before commit statement is performed. When crash occurs before commit statement, then Undo operation is performed.

UNDO (Ti): which restores the value of all data items updated by transaction (Ti) to the old values.
REDO (Ti): which sets the values of all data items updated by transaction (Ti) to the new values.

These two operations are important in order to guarantee correct behavior even if a failure occurs during the recovery process.

7. DBMS Selection.

An effective database system will allow growth and modification in the database without comprising the integrity of its data. The data dictionary/directory (DD/D) aids the accomplishment of this objective by allowing the definitions of data to be maintained separately from the data itself. This allows changes to be made to the data definitions with no effect on the stored data. For example, the subschema used by a particular program could be modified without in any way affecting the stored data. Other benefits provided by the DD/D include these:
Physical storage structures can be changed without affecting the programs that use the data. Passwords and other security measures can be stored in the DD/D to facilitate control over data access. Centralized data definition enables easy reporting on the status of the database: Why is responsible for the various data items To yield these benefits, the DD/D usually includes the following features:

- A language for defining entries in the DD/D.
- A manipulation language for adding, deleting, and modifying entries in the DD/D
- Methods for validating entries in the DD/D

Means for producing reports concerning the data contained in the DD/D.
Data Security and Integrity:
1. Access Controls: -

Access control is an important factor because they are a means of preventing unauthorized access to data. In the data-sharing database environment, good access controls are essential.

2. Concurrency controls: -

Concurrency controls are a means of manipulating data integrity in the multi-user environment. Suppose user A and user B both access a given record at (essentially) the same time in order to process a transaction against the record. The DBMS must limit access by one of the users until the others transaction has been completed. Without this type of facility, the accuracy and consistency of the database can rapidly erode.

3. View Controls: -

It provides an automated means of limiting what a user is allowed to access from a given relation. This is a powerful feature that is commonly provided by relational DBMS. The ease of creating views and the capability of the view facility can be a useful distinguishing factor among DBMSs. The DBMS purchaser may also be interested in whether views can be updated and what limitations may apply.

4. Encryption:

It facilitates may be important to institutions whose databases contain very sensitive data. Encryption can also be important for the maintenance of a secure password directory.

5. Backup and Recovery controls:

Effective Backup and recovery controls are absolutely essential to efficient operation of the database system. The ease of use of backup and recovery controls, and their completeness, and their reliability should be major factors in the DBMS selection decision.

8. Authentication view DBMS Implementation

Authorization and views : A view is a means of providing a user with a personalized model of the database. It is also a useful way of limiting a user's access to various positions of the database: Data a user does not need to be are simply hidden from view. This simplifies system usage while promoting security. Executing selects, projections, and joins on existing relations can represent views. The user might also be restricted from seeing any part of the existing relation or from executing joins on certain relations.

Types of Views: Different types of access authorization may be allowed for a particular view, such as the following:

- Read authorization: allows reading, but not modification of data.
- Insert authorization: allows insertion of new data, but no modification of existing data.
- Update authorization: allows modification of data, but not deletion.
- Delete authorization: allows deletion of data.

Views and security in SQL:
CREATE VIEW viewname (list of attributes desired, if different from base table)
As query.

I. Section-B: 5 X 10 =50 Marks

9(a). Explain Database Development.

DDLC (Database Development Life Cycle):
It is a process for designing, implementing and maintaining a database system.
It consists of six stages:
7. Preliminary design
8. Feasibility design
9. Requirements definition
10. Conceptual design
11. Implementation
12. Database evaluation and maintenance.

Preliminary Planning: It is a specific database system takes place during the strategic database planning project. After the database implementation project begins, the general information model produced during database planning is reviewed and enhanced if needed. During this process, the firm collects information to answer the following questions:

- How many application programs are in use, and what functions do they perform?
- What files are associated with each of these applications?
- What new applications and files are under development?

This information can be used to establish relationships between current applications and to identify uses of application information. It also helps to identify future system requirements and to assess the economic benefits of a database system.

Feasibility Study: A feasibility study involves preparing report on the following issues:

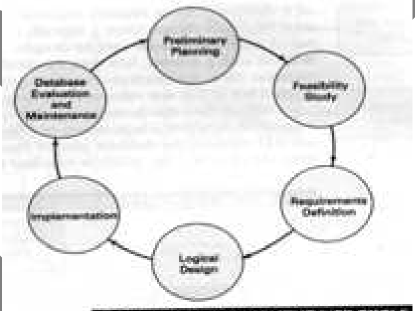
1. <u>Technological feasibility:</u> Is suitable technology available to support database development?
2. <u>Operational feasibility:</u> Does the company have personnel, budget and internal expertise to make a database system successful?
3. <u>Economic feasibility:</u> Can benefits be identified? Will the desired system be cost-beneficial? Can costs and benefits be measured?

Requirements Definition: It involves defining the scope of the database identifying management and functional area information requirements and establishing hardware/software requirements. Information requirements are determined from questionnaire responses, interviews with managers and clerical users and reports and forms currently being used.

Conceptual Design: The conceptual design stage creates the conceptual schema for the database. Specifications are developed to the point where implementation can begin. During this stage, detailed models of user view are created and integrated into a conceptual data model recording all corporate data elements to be maintained in the database.

Implementation: During database implementation, a DBMS is selected and acquired. Then the detailed conceptual model is converted to the implementation model of the DBMS, the data dictionary built, the database populate, application programs developed and users trained.

Database Evaluation & Maintenance: Evaluation involves interviewing users to determine if any data needs are unmet. Changes are made as needed. Over time the system is maintained via the introduction of enhancements and addition of new programs and data elements as business needs change and expand.



A DATABASE DEVELOPMENT LIFE CYCLE

9(b). Database Systems.

CASE tools:

Automated tools used to design databases and database application programs.

Repository:

A centralized store house of all data definitions, data relationships, screen, report formats and other components. A repository contains an extended set of metadata for managing databases as well as other components of Knowledge Base Systems.

Database management system (DBMS):

The software used to define, create, maintain, and provide controlled access to the database and the repository.

Database:

An organized collection of logically related data designed to meet the information needs of multiple users in an organization. It is important to distinguish between database and repository. The repository contains definitions of data whereas the database contains occurrences of data.

Application programs:

Computer programs that are used to create maintain the database and provide information to users.

User interface:

Languages, menus, and other facilities by which users interact with the various system components.

Data administrators:

The set of persons who are responsible for the overall information resources of an organization. Data administrators use CASE tools to improve the productivity of database planning and design.

System developers:

The set of persons such as systems analysts and programmers who design new application programs. System developers often use CASE tools for system requirements, analysis and program design.

End users:

Persons who add, delete, and modify data in the database and who request information from it. All user interactions within the database must be routed through the DBMS.

Data Sharing:
The most beneficial thing about database system when compared with a traditional file system is data sharing. It allows an enterprise to manage its database for different applications.

10(a). Relational Calculus.

Out of Syllabus

10(b). Explain the process of converting Conceptual model to Relational Model.

Out of Syllabus

11(a). Explain about DDL Commands.

1. CREATE 2. ALTER 3. DROP 4. TRUNCATE 5.RENAME

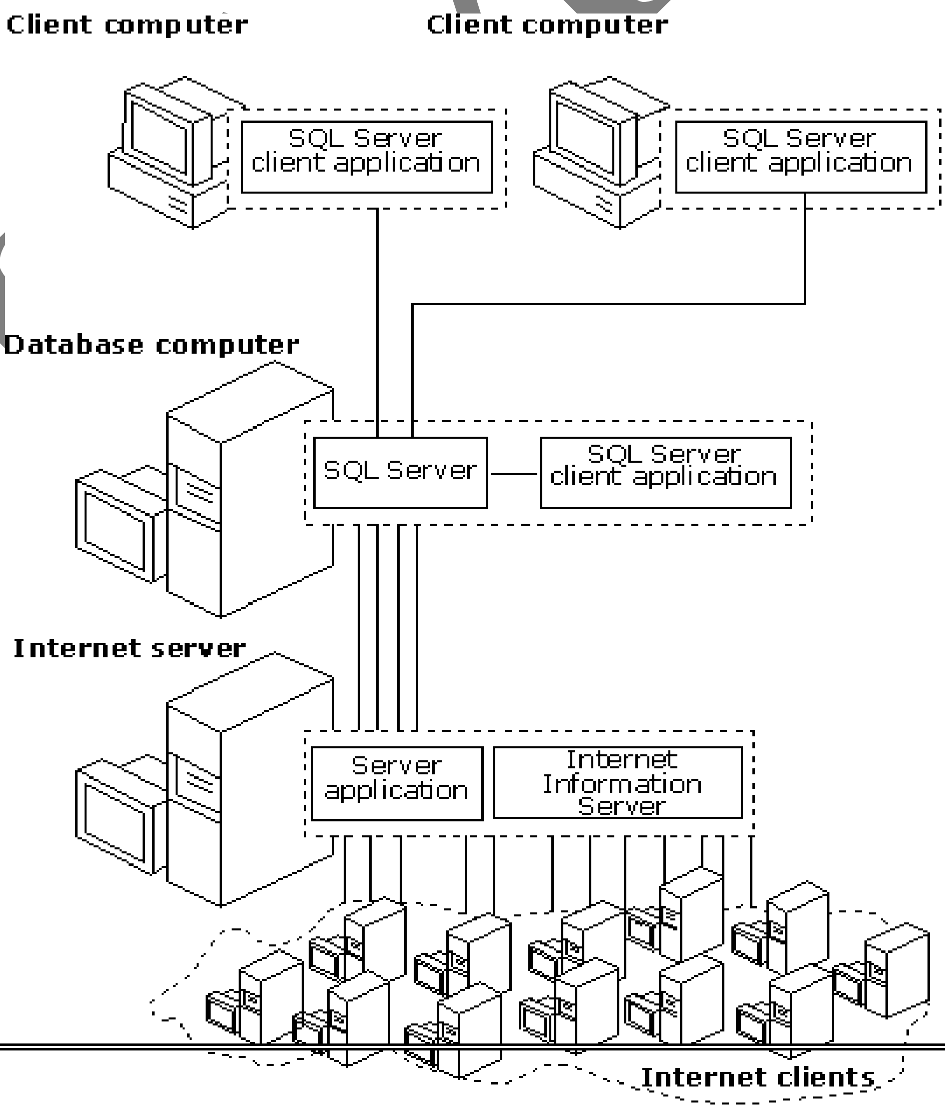
11(b). Explain Client-Server Database Systems.

Client/server systems are constructed so that the database can reside on a central computer, known as a server, and be shared among several users. Users access the server through a client or server application:

In a two-tier client/server system, users run an application on their local computer, known as a client, that connects over a network to the server running SQL Server. The client application runs both business logic and the code to display output to the user, and is also known as a thick client.

In a multi tier client/server system, the client application logic is run in two locations:

The thin client is run on the user's local computer and is focused on displaying results to the user.


Client computer
Client computer
SQL Server client application
SQL Server client application
Database computer
SQL Server
SQL Server client application
Internet server
Server application
Internet Information Server
Internet clients

The business logic is located in server applications running on a server. Thin clients request functions from the server application, which is itself a multithreaded application capable of working with many concurrent users. The server application is the one that opens connections to the database server and can be running on the same server as the database, or it can connect across the network to a separate server operating as a database server.

This is a typical scenario for an Internet application. For example, a server application can run on a Microsoft Internet Information Services (IIS) and service thousands of thin clients running on the Internet or an intranet. The server application uses a pool of connections to communicate with a copy of SQL Server. SQL Server can be installed on the same computer as IIS, or it can be installed on a separate server in the network.

12(a). Define File Organization. Explain the types of File organization.

There are three basic ways of physically organizing files on storage devices.

- Sequential organization
- Indexed-Sequential organization
- Direct organization.

The terms organization and access are often used loosely if not interchangeably. The reason is that the way in which data are stored is closely intertwined with the method access.

1. Sequential File Organization: Sequential file organizational means that records are stored adjacent to one another according to a key such as employee number, account number, and so forth. A conventional implementation arranges the records in ascending order of key values. This is efficient method of organizing records when an application. Such as a payroll program, will be updating a significant number of the stored records.

If a sequential file is maintained on magnetic tape, its records can only be accessed in a sequential manner. That is, if access to the tenth record in sequence is desired generally the preceding nine records must be read. Direct access of a particular record is impossible. Consequently magnetic tapes are not well suited for database operations and are usually relegated log files and recording archival information.

2. Indexed- Sequential File Organization : - When files are sequentially organized on a disk pack, however, direct access of records is possible. Indexed-sequential file organization provides facilities for accessing records both sequentially and directly. Records are stored in the usual physical sequence by primary key. In addition an index of record locations is stored on the disk. This allows records to be accessed sequentially for applications requiring the updating of large numbers of records, as well as providing the ability to access records directly in response to user queries.

3. Direct File Organization: The third type of file organization is called direct or hashed.
It is of two types:

- Static Hash Function
- Dynamic Hash Function

(a). Static Hash Function: The use of hashing is a method of record addressing that eliminates the need for maintaining and searching indexes. The basic idea is that of trading the time and effort associated with storing, maintaining, and searching an index for the time required for the central processing unit (CPU) to execute a hashing algorithm, which generates the record address. The hashing algorithm is a procedure for calculating a record address from some field in the record, usually the key.

(b). Dynamic hash functions: The static hash function is fairly simple. As the database grows, however, the static hash function loses its appeal. One strategy for dealing with this problem is to allocate estimated space for future requirements at the outset; but this wastes storage space. Another scheme is to allocate additional storage and reorganize the file as it grows.

A better approach is provided by the dynamic hash function. This hashing splits and combines blocks as the database grows or shrinks. This ensures efficient space utilization. Moreover, since reorganization involves only one block at a time, the associated overhead is minimal.

Dynamic hashing uses a hash function "h" that has the useful characteristics of randomness and uniformity. It also (typically) uses a 32-bit binary string in order to create and identify block indexes.

12(b). Database Security.

Database integrity problems can be challenging, but they are generally easier to cope with than malicious access to the database, which includes the following:

1. Theft of information
2. Unauthorized modification of data
3. Unauthorized destruction of data

Thus, database security methods focus on preventing unauthorized users from accessing the database. Because DBMS features that make the database easy to access and manipulate also open doors to intruders, most DBMS include security features that allow only authorized persons or processing that can be accompanied once access is made.

Authentication: - Database access usually requires user authentication and authorization. For user authentication, the first level of security establishes that the person seeking system the user knows, such as log-on number and password, (2) something the user possesses, such as plastic ID card, or (3) a physical representation of the user, such as fingerprint or voiceprint.

Authorization and views: - A view is a means of providing a user with a personalized model of the database. It is also a useful way of limiting a user's access to various positions of the database: Data a user does not need to se are simply hidden from view. This simplifies system usage while promoting security. Executing selects, projections, and joins on existing relations can represent views. The user might also be restricted from seeing any part of the existing relation or from executing joins on certain relations.

Types of Views: - Different types of access authorization may be allowed for a particular view, such as the following:

1. Read authorization: allows reading, but not modification of data.
2. Insert authorization: allows insertion of new data, but no modification of existing data.
3. Update authorization: allows modification of data, but not deletion.

4. Delete authorization: allows deletion of data.

Views and security in SQL: -
CREATE VIEW viewname As (select statement)

Encryption: - The various authentication and authorization measures that are standard for protection access to database may not be adequate for highly sensitive data. In such instances, it may be desirable to encrypt the data. Encrypted data cannot be read by an intruder unless that party knows the method of encryption. Considerable research has been devoted to developing encryption methods.

13(a). Explain Distributed Query Processing.

A database that is distributed among a network of geographically separated locations. A distributed database is not entirely stored in one central location but is distributed among a network of locations that are geographically separated and connected by communication links. Each location has its own database and it also able to access data maintained at other locations.

The reasons for the development and use of distributed database systems are several and include the following:

Need of Distributed Database: -

1. Often organizations have branches or divisions in different locations. For a given location, L, there may be a set of data that is used frequently perhaps exclusively, at L. In addition, L may sometimes need data that are used more frequently at another location, L.
2. Allowing each site to store and maintain its own database allows immediate and efficient access to data that are used most frequently. Such data may be used at others site as well, but usually with less frequency. Similarly, data stored at other locations can be accessed as required.
3. Distributed database can upgrade reliability. If one site's computer fails, or if a communication link goes down, the rest of the network can possibly continue functioning. Moreover when data are replicated at two or more sites, required data may still be available from a site, which is still operate.
4. Allowing local control over the data used most frequently at a site can improve user satisfaction with the database system. That is to say, local database can more nearly reflect an organization's administrative structure and thereby better service its manager's needs.

Distributed Query Processing:

Some database systems support relational databases whose parts are physically separated. Different relations might reside at different sites, multiple copies of a single relation can be distributed among several sites, or one relation might be partitioned into subrelations and these subrelations distributed. In order to evaluate a query posed at a given site, it may be necessary to transfer data between various sites. The key consideration here is that the time required to process such a query will largely be comprised of the time spent transmitting data between sites rather than the time spent on retrieval from secondary storage or computation.

Semijoin: - Suppose the relations R and S shown in Figure. Is stored at sites 1 and 2, respectively. If we wish to respond to a query at site 1 which requires the computation:

JOIN (R, S),

We could transmit all of S from site 2 to site 1 and compute the join at site 1. This would involve the transmission of all 24 values of S.

Site 1		*Site 2*		
R		S		
A1	*A2*	A2	A3	A4
1	3	3	13	16
1	4	3	14	16
1	6	7	13	17
2	3	10	14	16
2	6	10	15	17
3	7	11	15	16
3	8	11	15	16
3	9	12	15	16

Another way would be to compute
T = R [A2]

At site 1; then send T (6 values) to site 2, and compute

U = JOIN (T, S);

And finally send U (9 values) to site 1. We can then compute the desired

JOIN (R, S),

As
JOIN (R, U).

These steps and their results are shown in Figure 12.16. Note that with this approach we have only transmitted 15 values to complete the query.
This example provides a basis for defining a semijoin. The semijoin of R with S is

SEMIJOIN (R, S) = <a projection of those attributes of R that intersect those of S>,

Which is simply that portion of R that joins with S. Therefore, JOIN (R, S) = JOIN (R, (SEMIJOIN (R, S), S)).

If R and S are at different sites, computing join (R, S) as shown previously saves transmitting data whenever R and S not join completely.

13(b). Explain about DBMS Implementation issues

Implementation planning and administration is as important to database systems as it is -to the effective implementation of any new technology. In this section we outline some of the important considerations associated with DBMS implementation.

DBA:The responsibility for database administration is usually assigned to an individual called a database administrator (DBA). The DBA is charged with ensuring that the database system operates effectively and efficiently. In order to do this, the DBA's daily activities are concerned with the following tasks:
1. Serving end-user requirements.
2. Ensuring database security and integrity.
3. Establishing backup-and-recovery procedures.

Database Testing: The loading of the database has been accomplished without violating the data integrity. The applications interface correctly with the database. The performance of the system satisfies the requirements for which the DBMS was acquired. An objective of testing, which is sometimes overlooked, is finding out where the database system does not function as expected.

Preparing Users for Change: Since user acceptance is essential to a successful database system implementation, preparing the users for the changes that will affect them is essential. How is this best accomplished?Conceptually, the answer is simple: Involve the users. in the development of the new systems, train them thoroughly, and include user-acceptance tests as part of the implementation effort. Carrying out these activities may be a bit more complex.

Loading the Database
Often the data to be stored in the database already exist on some computer-based medium such as magnetic tape. In the best cases, all the required data exist, and database loading may simply involve restructuring the existing data. That is to say, a program can be written that reads the old files and creates the structure needed for the new ones.

Database Maintenance
Once the DBMS is installed and in operation, maintenance activities need to be organized and performed to ensure that effective service and operations are provided.
- Managing Resources
- Backup and Recovery
- Managing Changes to the Database System