Relational Database Management Systems – Mar/Apr 2014

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

1. Data Model

A conceptual method of structuring data is called Data Model.
The development of systems based on following data models. They are
1. Entity-Relationship Model
2. Object Oriented Model
3. Relational Model
4. Hierarchical Model
5. Network Model

2. E-R Diagram

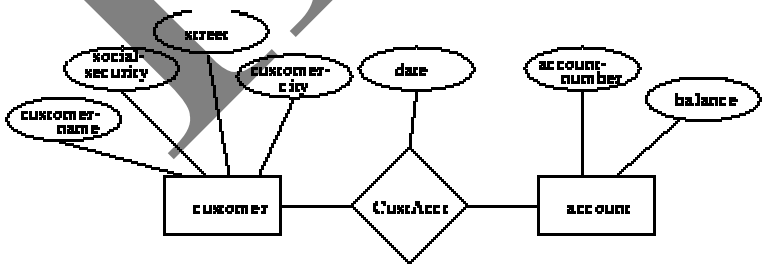
E-R Diagram: An entity-relationship diagram (ERD) is a data modeling technique that creates a graphical representation of the entities, and the relationships between entities, within an information system. Any ER diagram has an equivalent relational table, and any relational table has an equivalent ER diagram.

Entity: The entity is a person, object, place or event for which data is collected. It is equivalent to a database table. An entity can be defined by means of its properties, called attributes. For example, the CUSTOMER entity may have attributes for such things as name, address and telephone number.

Relationship: The relationship is the interaction between the entities.

E-R diagram components are:

1. Rectangles representing entity sets.
2. Ellipses representing attributes.
3. Diamonds representing relationship sets.
4. Lines linking attributes to entity sets and entity sets to relationship sets.



3. Structural Query Language.

SQL stands for “Structured Query Language” and can be pronounced as “SQL” or “sequel – (Structured English Query Language)”. It is a query language used for accessing and modifying information in the database. IBM first developed SQL in 1970s. Also it is an ANSI/ISO standard. It has become a Standard Universal

Language used by most of the relational database management systems (RDBMS). Some of the RDBMS systems are: Oracle, Microsoft SQL server, Sybase etc. Most of these have provided their own implementation thus enhancing it's feature and making it a powerful tool. Few of the SQL commands used in SQL programming are SELECT Statement, UPDATE Statement, INSERT INTO Statement, DELETE Statement, WHERE Clause, ORDER BY Clause, GROUP BY Clause, ORDER Clause, Joins, Views, GROUP Functions, Indexes etc.
In a simple manner, SQL is a non-procedural, English-like language that processes data in groups of records rather than one record at a time. Few functions of SQL are:

- Store data
- Modify data
- Retrieve data
- Modify data
- Delete data
- Create tables and other database objects

4. File Organization.

File Organization refers to the organisation of the data of a file into records, blocks, and access structures, this includes the way records and blocks are placed on the storage medium and interlinked.

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

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

5. Distributed Databases

A distributed database is integrated database which is built on top of a computer network rather than on a single computer. The data which constitute the database are stored at the different sites of the computer network, and the application programs which are run by the computers access data at different sites. Databases may involve different database management systems, running on different architectures that distribute the execution of transactions.

A Distributed Database Management System (DDBMS) is defined as the software that handles the management of the DDB (Distributed Database) and makes the operation of such a system appear to the user as a centralized database.

6. Data Redundancy.

Since files and application programs are created by different programmers over a long period of time, the files are likely to be having different formats and the programs may be written in several programming languages. Moreover, the same piece of information may be duplicated in several places. This redundancy leads to higher storage and access cost. In addition, it may lead to data inconsistency, i.e. the various copies of same data may no longer agree.

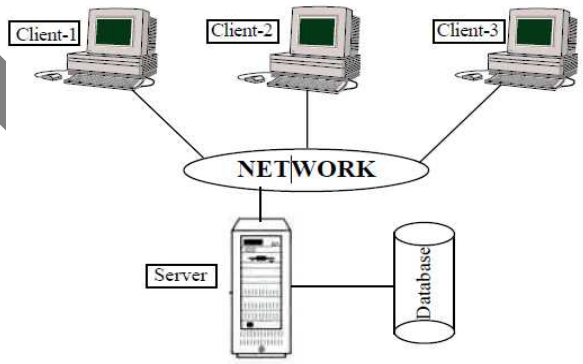
7. DBA.

Database Administrator is a person with the responsibility of controlling and protecting the data. The DBA should coordinate the design of the database, guide the development and implementation of data security procedures, protect the integrity of data values and make sure system performance is satisfactory.

In a small organization, one person carries out all these responsibilities. Often, these functions are assigned to a group of people. This is most likely in a large organization where DBA responsibilities are divided among several people managed by a chief administrator.

8. 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.



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

9(a). Components of DBMS and Relational Model.

A complete DBMS in an organization consists of the following components.

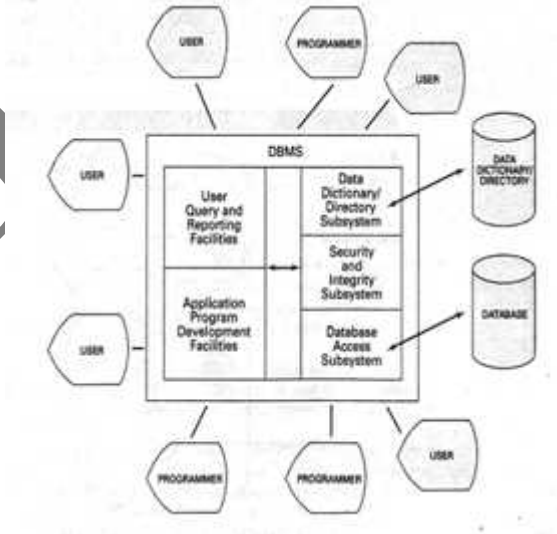
(i). Hardware: The hardware is the set of physical devices on which a database resides. It consists of one or more computers, disk drives, CRT terminals, printers, tape drivers, connecting cables, etc.

The computers used for processing the data in the database may be mainframe, mini computers or personal computers. Mainframe and mini computers have traditionally been used on a stand-alone basis to support multiple users accessing a common database. Personal computers are often used with stand-alone databases controlled and accessed by a single user.

Disk drivers are the main storage mechanism for databases. Desktop computers, CRT terminals and printers are used for entering and retrieving information from the database. The success of the database system has been heavily dependent on advances in hardware technology. A very large amount of main memory and disk storage is required to maintain and control the huge quantity of data stored in a database.

(ii). Software: A database system includes two types of software:
a. General Purpose database management software usually called the database management system (DBMS).
b. Application software that uses DBMS facilities to manipulate the database to achieve a specific business functions.

Application software is generally written by programmers to solve a specific company problem. It may be written in languages like COBOL or C or it may be written in a language supplied by DBMS like SQL. Application software uses the facilities of the DBMS to access and manipulate data in the database providing reports or documents needed for the information and processing needs of the company.



The DBMS is system software similar to an operating system. It provides a number of services to end users and programmers.

Prashanth Kumar K
(Head-Dept of Computers, IIMC)

DBMS typically provides most of the following services.

1. A central data definition and data control facility known as a data dictionary/directory).
2. Data security and integrity mechanisms.
3. Concurrent data access for multiple users.
4. User-oriented data query, manipulation and reporting capabilities.
5. Programmer-oriented application system development capabilities.

(iii). Data: No database system can exist without data. Data can be collected and entered into the Database according to the defined structure.

(iv). People: Two different types of people concerned with the database.

They are:

1. Users: Executives, Managers, Staff, Clerical personnel.
2. Practitioners: Database Administrators, Programmers.

(v). Database: Database is a collection of inter-related data items that can be processed by one or more application systems.

Relational Model:

The relational model uses a collection of tables to represent both data and relationships Among those data, each table has multiple columns and each column has a unique name. Here relation refers to a two dimensional table containing rows and columns of data.

CUSTOMER TABLE

Cust-no	Cust-name	Cust-address
11500	ABC	Hyderabad
11501	DEF	Mumbai
11502	IJK	Chennai
11503	XYZ	Bangalore
11504	PQR	Pune

9(b). Advantages of Database approaches.

Program Data Independence:

If a database approach is used, data is stored in a central location called repository. The process of database allows an enterprise's data to change the database without modifying the application programs which are able to process this data. This allows the separation of database from the application programs.

Minimal Data Redundancy:

Data Redundancy exists when the same data are stored unnecessarily at different places. The design with the database approach is that separate data files are integrated into a single structure. The database approach does not eliminate redundancy completely, but it provides the facilities to the designer to carefully control the amount of redundancy.

Improved Data Consistency: If the amount of data redundancy is controlled, it will reduce data inconsistency also. It is also highly recommended to maintain the same version of the data at all the locations. For example, when a customer address is stored at only one location, if the customer changes the address, it will be automatically reflected in all the applications related to that particular customer.

Improved Data Sharing:
A database is designed as a sharable component. The DBMS helps in creating an environment in which end users have better access to more data and better manages data. Such access makes it possible for end users to respond quickly to change in their environment. Users are allowed to utilize the services of the database by authentication and authorization.

Enforcements of Standards:
To facilitate the services of database management, every database administrator designs establishing procedures and enforcement of standards. Procedures are the instructions and rules that govern the design and use of the database system. Procedures play an important role in an enterprise because they enforce the standards by which business is generated within the organization and with customers. Procedures are also used to ensure that there is an organized way to monitor and audit both the data and the information that is generated through the use of the data.

Improved Quality:
The database approach provides an optimum number of tools and processes to improve data quality. Every data designer can specify a rule called integrity constraint which users cannot violate. The availability of data combined with the tools that transform data into usable information, empowers end users to make quick, informed decisions that can make the difference between the success and failure in the global economy.

10(a). What is Normalization? Explain different types of Normal Forms.

Normalization is the process of converting a relation into standard form.

It is of different types:

1st Normal Form: A table is in first normal form if all the key attributes have been defined and it contains no repeating groups

2nd Normal Form: A table is in second normal form (2NF) if and only if it is in 1NF and every non key attribute is fully functionally dependent on the whole of the primary key (i.e. there are no partial dependencies).

3rd Normal Form: A table is in third normal form (3NF) if and only if it is in 2NF and every non key attribute is non transitively dependent on the primary key (i.e. there are no transitive dependencies)

1. Anomalies can occur when a relation contains one or more transitive dependencies.
2. A relation is in 3NF when it is in 2NF and has no transitive dependencies.
3. A relation is in 3NF when 'All non-key attributes are dependent on the key, the whole key and nothing but the key'.

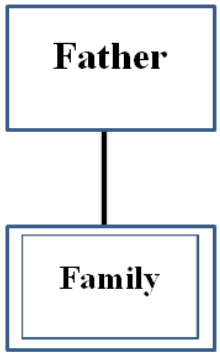
Boyce-Codd Normal Form: A table is in Boyce-Codd normal form (BCNF) if and only if it is in 3NF and every determinant is a candidate key.

10(b). Explain strong entity type, weak entity type in E-R model.

An entity can be a person, place, event, or object that is relevant to a given system. For example, a school system may include students, teachers, major courses, subjects, fees, and other items. Entities are represented in ER diagrams by a rectangle and named using singular nouns.

Strong Entity: Strong entity is one whose existence does not depend on other entity or entities. For instance, a staff, a student, etc. can be said as strong entity. In a parent/child relationship, a parent is considered as a strong entity and the child is a weak entity.

Weak Entity: In a relational database, a weak entity is an entity that cannot be uniquely identified by its attributes alone. Weak entity is on whose existence depends on other entity or entities. For instance, student takes a particular course than that course cannot be offered. That course entity depends on the student entity. The weak entity is shown in double rectangle box.



In the above diagram, Family is a weak entity because it is dependant to another entity.

11(a). Define Data Manipulation. Explain commands used in data manipulation.

DML is abbreviation of Data Manipulation Language. The DML commands help the user to access the table and insert the content into the table of modify the data or delete the data available in the table. It is used to retrieve, store, modify, delete, insert and update data in database. These commands are limited to the data part of the table does not involve with the structure of the table. Some of the commands which are used to enter the data and perform manipulation or retrieve the selected content from the table are

1. Insert 2. Select 3. Update 4. Delete.

11(b). Explain Aggregate Functions or Group functions with examples.

Aggregate Function	*Returns...*
MIN(X)	the smallest value in column x
MAX(X)	the largest value in column x
AVG(X)	the average value in column x
SUM(X)	The total of the values in column x
COUNT(X)	the number of values in column x
COUNT(*)	the number of records in the table being searched

12(a). Functions of DBA.

Database Administrator is a person with the responsibility of controlling and protecting the data. The DBA should coordinate the design of the database, guide the development and implementation of data security procedures, protect the integrity of data values and make sure system performance is satisfactory.

In a small organization, one person carries out all these responsibilities. Often, these functions are assigned to a group of people. This is most likely in a large organization where DBA responsibilities are divided among several people managed by a chief administrator.

Functions of DBA:

1. Schema definition. The DBA creates the original database schema by executing a set of data definition statements in the DDL.

2. Storage structure and access-method definition: writing a set of definitions translated by the data storage and definition language compiler

3. Schema and physical-organization modification. The DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization, or to alter the physical organization to improve performance.

4. Granting of authorization for data access. By granting different types of authorization, the database administrator can regulate which parts of the database various users can access. The authorization information is kept in a special system structure that the database system consults whenever someone attempts to access the data in the system.

5. Integrity constraint specification: generating integrity constraints. These are consulted by the database manager module whenever updates occur.

12(b). Database Recovery and Security.

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.

Prashanth Kumar K
(Head-Dept of Computers, IIMC)

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

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). DBMS functions and capabilities.

The Data Dictionary/Directory: - 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 somehow 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.

Query, Data Manipulation, and Reporting Capabilities: -

The DBMS's ability to support reporting requirements, along with users' query and data manipulation needs, is the cornerstone of today's management information

systems. A sound DBMS is going to provide the capability to generate structured reports in a variety of formats. In addition, the DBMS will provide a query language that is powerful, yet easy to learn and use. The language should be able to support both planned and unplanned query requirements with short response times.

Support of Specialized Programming Requirements: -
Developing specialized programs to interface with the DBMS requires facilities for supporting program development and program testing. A worthy DBMS will provide a host language for expressing standard procedural program structures or will provide an interface capability for quick prototyping of applications.

Physical Data Organization Options: -
The firm acquiring a DBMS may not wish to involve itself in the details of physical data organization. Instead, it may gauge the efficiency of a DBMS's physical organization by running sample applications.

For those who are interested, however, exploring the physical organization features may be of value. For example, it is known that the inverted list is most efficient in supporting multikey retrieval, whereas the chain list is superior for file updating since there is no need for updating a separate file. Information on other architectural features may be elicited in the process of considering the DBMS's capability to support the types of applications common to the firm.

13(b). 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