Relational Database Management Systems – Mar/Apr 2015

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

1. Database

Database: Database is a collection of inter-related data which contains the information of an enterprise. It is obtained by collecting the data from all sources of the organization. Database can be shared, integrated and managed with other computer applications.

2. Strong Entity Type.

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.

3. SQL

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 its 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. 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.

5. Database Integrity

A condition or integrity that is applied to a particular set of data is commonly termed **Integrity Control** or **Constraint.**

In relational model terminology, integrity controls may apply to

(1) Individual attributes

(2) the relationship between two different attributes (perhaps in different relations)

(3) the relationship between tuples of one or more tables. Ideally, the enforcement of integrity constraints would be carried out by the DBMS currently as each new data item is entered.

6. Database Security

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.

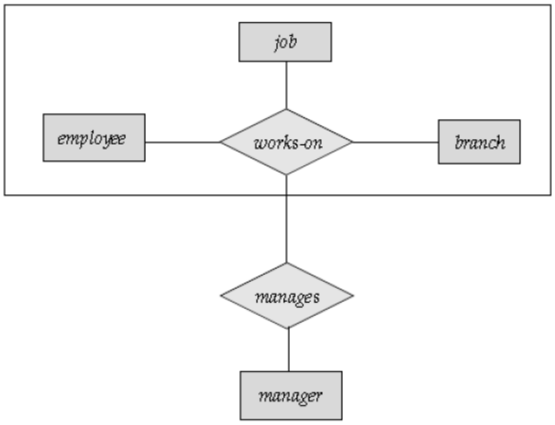
7. Relational Model

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.

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

8. Aggregation

Aggregation is an abstraction through which relationships are treated as higher level entities. For an example, the relationship set works-on (relating the entity sets employee, branch, and job) as a higher-level entity set called works-on. Such an entity set is treated in the same manner as is any other entity set.



II. Section-B: **5 X 10 =50 Marks**

9(a). Write about advantages of Database Approach.

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:

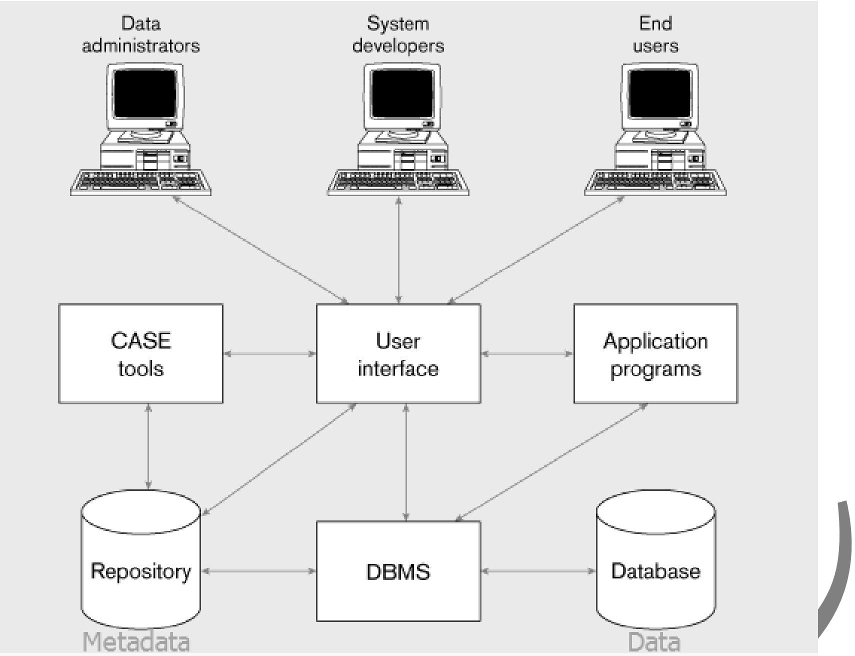
The database approach provides an optimum number of tools and processes to improve data quality. Every data designer can specify a rule called integrity constraint which users cannot violate. The availability of data combined with the tools that transform data into usable information, empowers end users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

9(b). What are the Components of Database Environment.

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.

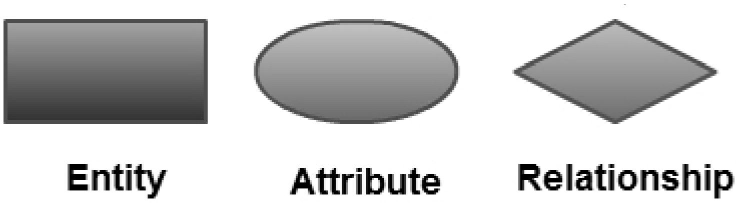
End users: Persons who add, delete, and modify data in the database and who request information from it. All user interactions within the database must be routed through the DBMS.

10(a). Explain E-R diagram with an example.

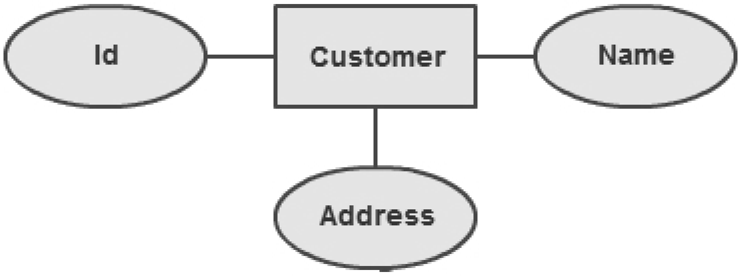
The E-R (Entity-Relationship) model views the real world as a set of basic objects and **relationships** among these objects. This represents the overall logical structure of the database.

The basic elements of E-R Model are Entity, Attribute and Relationship.

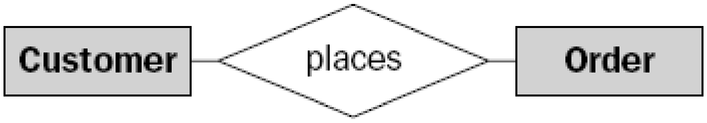
E-R Diagram: E-R Model is pictorial represented as E-R Diagram. The notations of this diagram is as follows:



An Entity (Customer) contains Attributes (Id, Name, Address) is represented as follows:



Two Entities (Customer) & (Order) are related as follows:



The relationship between the two entities is given the name “place”. It gives the meaning of the Customer places an Order.

An Entity is a real-world object which is distinguishable from other objects. An entity may be defined as a thing which is recognized as being capable of an independent existence and which can be uniquely identified. Preferably, every entity must have a minimal set of uniquely identifying attributes, which is called the entity's primary key.

General examples for entities are:

- A physical object like computer or bike.
- A human-being
- An organization or a department of an organization
- Country or a place e.g. Hyderabad
- Software like Oracle, MS-Access etc.

Entity Set: A collection of similar entities is called Entity Set or Entity Type.

Some examples of Entity Set are:

- All employees of an organization, EMP.
- All students of a college or university, STUDENT.
- All departments of a university, DEPT.

Attribute: An Attribute is a property or characteristic of an entity. In other words, An Entity is described using Attributes in database. All entities in an entity set have the same set of attributes.

Some examples of Attributes:

- EMP contains emp-no, emp-name, sal
- STUDENT contains student-no, student-name, course
- DEPT contains dept-no, dept-name.

Relationship:

A Relationship is an association of two or more entities. A meaningful relationship is called Relationship Set or Relationship Type. Relationships can be classified by their degree.

Some examples of relationship set are:

- Working is the relationship between EMP and DEPT.
- Borrows is the relationship between CUSTOMER and LOAN.
- Performs is the relationship between ARTIST and SONG.

10(b). Define Normalization. Explain 1NF, 2NF

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

Un-Normalized Form (UNF): If a table contains non-atomic values at each row, it is said to be in UNF. An atomic value is something that cannot be further decomposed. A non-atomic value, as the name suggests, can be further decomposed and simplified.

EmpNo	**EmpName**	**Month**	**Sales**	**Bankno**	**BankName**
100	ABC	Jan	1000	B01	SBI
		Feb	1200		
		Mar	850		
101	DEF	Jan	2200	B02	HDFC
		Feb	2500		
102	XYZ	Jan	1700	B01	SBI

In the above table, there are multiple occurrences of rows under each key EmpNo. Although considered to be the primary key, EmpNo cannot give us the unique identification facility for any single row. Further, each primary key points to a variable length record(3 for Empno.100 , 2 for Empno.101 and 1 for Empno.102).

First-Normal Form (1NF): A relation is said to be in 1NF if it contains no non-atomic values and each row can provide a unique combination of values. The above table in UNF can be processed to create the following table in 1NF.

EmpNo	EmpName	Month	Sales	Bankno	BankName
100	ABC	Jan	1000	B01	SBI
100	ABC	Feb	1200	B01	SBI
100	ABC	Mar	850	B01	SBI
101	DEF	Jan	2200	B02	HDFC
101	DEF	Feb	2500	B02	HDFC
102	XYZ	Jan	1700	B01	SBI

Each row contains unique combination of values. Unlike in UNF, this relation contains only atomic values, i.e. the rows cannot be further decomposed, so the relation is now in 1NF.

Second Normal Form (2NF): A relation is said to be in 2NF if it is already in 1NF and each and every attribute fully depends on the primary key of the relation. Speaking inversely, if a table has some attributes which is not dependant on the primary key of that table and then it is not in 2NF.

EmpNo	EmpName	Month	Sales	Bankno
100	ABC	Jan	1000	B01
100	ABC	Feb	1200	B01
100	ABC	Mar	850	B01
101	DEF	Jan	2200	B02
101	DEF	Feb	2500	B02
102	XYZ	Jan	1700	B01

Bankno	BankName
B01	SBI
B02	HDFC

11(a). Explain operators present in SQL.

An operator manipulates individual data items and returns a result. The data items are called *operands* or *arguments*. Operators are represented by special characters or by keywords. For example, the multiplication operator is represented by an asterisk (*) and the operator that tests for nulls is represented by the keywords IS NULL. There are two general classes of operators: unary and binary. SQL also supports with set operators.

UNARY OPERATORS:

A unary operator uses only one operand. A unary operator typically appears with its operand in the following format:

Syntax:- operator operand

BINARY OPERATOR

A binary operator uses two operands. A binary operator appears with its operands in the following format:

Syntax:- operand1 operator operand2

CHARACTER OPERATORS

Character operators are used in expressions to manipulate character strings.

Syntax:- Character Operators

Operator : ||

Description : Concatenates character strings

Example: SELECT 'The Name of the employee is: ' || ENAME FROM EMP;

SQL ARITHMETIC *Operators:*

Operator	Description	Example
+ (unary)	Makes operand positive	SELECT +3 FROM DUAL;
- (unary)	Negates operand	SELECT -4 FROM DUAL;
/	Division (numbers and dates)	SELECT SAL / 10 FROM EMP;
*	Multiplication	SELECT SAL * 5 FROM EMP;
+	Addition (numbers and dates)	SELECT SAL + 200 FROM EMP;
-	Subtraction (numbers and dates)	SELECT SAL - 100 FROM EMP;

SQL COMPARISON OPERATORS

Operator	Description	Example
=	Equality test.	SELECT ENAME "Employee" FROM EMP WHERE SAL = 1500;
!=, ^=, <>	Inequality test.	SELECT ENAME FROM EMP WHERE SAL ^= 5000;
>	Greater than test.	SELECT ENAME "Employee", JOB "Title" FROM EMP WHERE SAL > 3000;
<	Less than test.	SELECT * FROM PRICE WHERE MINPRICE < 30;
>=	Greater than or equal to test.	SELECT * FROM PRICE WHERE MINPRICE >= 20;
<=	Less than or equal to test.	SELECT ENAME FROM EMP WHERE SAL <= 1500;
IN	"Equivalent to any member of" test. Equivalent to "= ANY".	SELECT * FROM EMP WHERE ENAME IN ('SMITH', 'WARD');
ANY/ SOME	Compares a value to each value in a list or returned by a query. Must be preceded by =, !=, >, <, <=, or >=.	SELECT * FROM DEPT WHERE LOC = SOME ('NEW YORK','DALLAS');

Operator	Description	Example
	Evaluates to FALSE if the query returns no rows.	
NOT IN	Equivalent to "!= ANY". Evaluates to FALSE if any member of the set is NULL.	SELECT * FROM DEPT WHERE LOC NOT IN ('NEW YORK', 'DALLAS');
ALL	Compares a value with every value in a list or returned by a query. Must be preceded by =, !=, >, <, <=, or >=. Evaluates to TRUE if the query returns no rows.	SELECT * FROM emp WHERE sal >= ALL (1400, 3000);
[NOT] BETWEEN *x* and *y*	[Not] greater than or equal to *x* and less than or equal to *y*.	SELECT ENAME, JOB FROM EMP WHERE SAL BETWEEN 3000 AND 5000;
EXISTS	TRUE if a sub-query returns at least one row.	SELECT * FROM EMP WHERE EXISTS (SELECT ENAME FROM EMP WHERE MGR IS NULL);
x [NOT] LIKE *y* [ESCAPE *z*]	TRUE if *x* does [not] match the pattern *y*. Within *y*, the character "%" matches any string of zero or more characters except null. The character "_" matches any single character. Any character following ESCAPE is interpretted litteraly, useful when *y* contains a percent (%) or underscore (_).	SELECT * FROM EMP WHERE ENAME LIKE '%E%';
IS [NOT] NULL	Tests for nulls. This is the only operator that should be used to test for nulls.	SELECT * FROM EMP WHERE COMM IS NOT NULL AND SAL > 1500;

SQL LOGICAL OPERATORS

NOT	Returns TRUE if the following condition is FALSE. Returns FALSE if it is TRUE. If it is UNKNOWN, it remains UNKNOWN.	SELECT * FROM EMP WHERE NOT (job IS NULL)
		SELECT * FROM EMP WHERE NOT (sal BETWEEN 1000 AND 2000)

AND	Returns TRUE if both component conditions are TRUE. Returns FALSE if either is FALSE; otherwise returns UNKNOWN.	SELECT * FROM EMP WHERE job='CLERK' AND deptno=10
OR	Returns TRUE if either component condition is TRUE. Returns FALSE if both are FALSE. Otherwise, returns UNKNOWN.	SELECT * FROM emp WHERE job='CLERK' OR deptno=10

SET OPERATORS:-

Set operators combine sets of rows returned by queries, instead of individual data items. All set operators have equal precedence.

Operator	Description	Example
UNION	Returns all distinct rows selected by either query.	SELECT * FROM (SELECT ENAME FROM EMP WHERE JOB = 'CLERK' UNION SELECT ENAME FROM EMP WHERE JOB = 'ANALYST');
UNION ALL	Returns all rows selected by either query, including all duplicates.	SELECT * FROM (SELECT SAL FROM EMP WHERE JOB = 'CLERK' UNION SELECT SAL FROM EMP WHERE JOB = 'ANALYST');
INTERSECT and INTERSECT ALL	Returns all distinct rows selected by both queries.	SELECT * FROM orders_list1 INTERSECT SELECT * FROM orders_list2
MINUS	Returns all distinct rows selected by the first query but not the second.	SELECT * FROM (SELECT SAL FROM EMP WHERE JOB = 'PRESIDENT' MINUS SELECT SAL FROM EMP WHERE JOB = 'MANAGER');

11(b). Explain Table Definition with example in SQL.

Table: Table are defined in three steps.

1. The name of the table is given.
2. Each column is defined, possibly including column constraints.
3. Table constraints are defined.

```
Create table Tablename(column-name1, data type(number of characters))
                      (column-name2, data type(number of characters))
                       ……………………………………………………...
                      (column-name, data type(number of characters))
```

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). Explain 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.

13(a). Explain need of Distributed database.

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.

13(b). Explain Implementation issues of DBMS

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