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

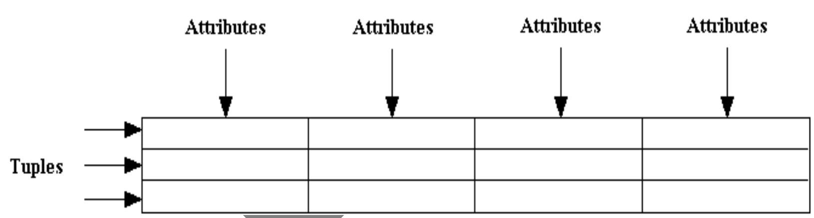
2. Data Redundancy.

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.

3. Relation

The Relation

The *Relation* is the basic element in a relational data model.



A relation is subject to the following rules:

- Relation (file, table) is a two-dimensional table.
- Attribute (i.e. field or data item) is a column in the table.
- Each column in the table has a unique name within that table.
- Each column is homogeneous. Thus the entries in any column are all of the same type (e.g. age, name, employee-number, etc).
- A Tuple (i.e. record) is a row in the table.

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

4. Normalization

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)

5. Schema

Schema: The logical design of the database is called Schema. The concept of relation schema corresponds to the programming language notion of type definition.

Create Schema Sch_name

Authorization Auth_name

Domain definition

Table definition

View definition

6. Encryption.

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.

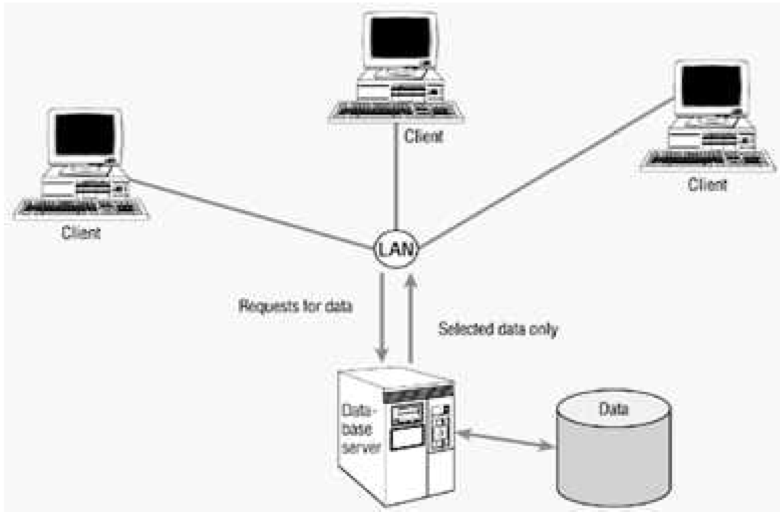
7. Two-Tier Architecture.

- Client is responsible for
  - I/O processing logic
  - Some business rules logic
- Server performs all data storage and access processing.
  - ➔ DBMS is only on server

Advantages of Two-Tier Approach:

- Clients do not have to be as powerful
- Greatly reduces data traffic on the network
- Improved data integrity since it is all processed centrally

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



8. Database 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.

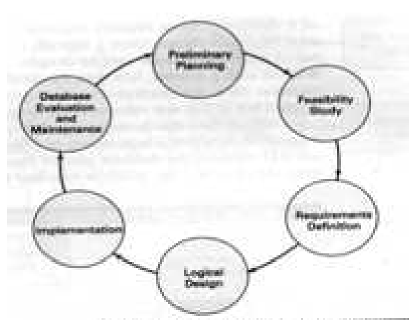
I. Section-B: 5 X 10 =50 Marks

9(a). Explain steps in DDLC.

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.



A DATABASE DEVELOPMENT LIFE CYCLE

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:

1. How many application programs are in use, and what functions do they perform?
2. What files are associated with each of these applications?
3. 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. Technological feasibility: Is suitable technology available to support database development?
2. Operational feasibility: Does the company have personnel, budget and internal expertise to make a database system successful?
3. Economic feasibility: 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.

9(b). Write about Components of DBMS.

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

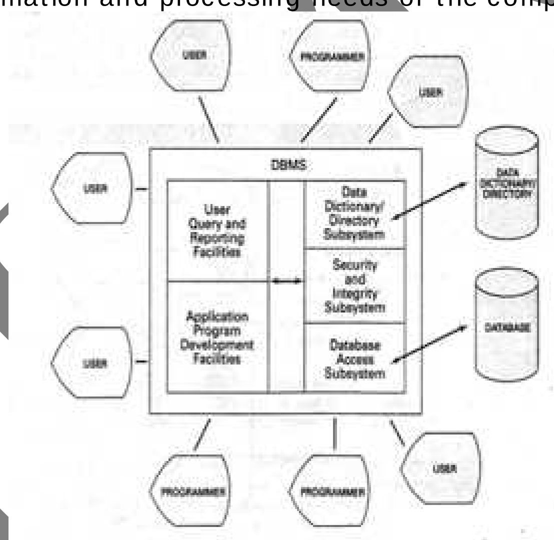
(i). Hardware: The hardware is the set of physical devices on which a database resides. It consists of one or more computers, disk drives, CRT terminals, printers, tape drivers, connecting cables, etc.

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.

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.

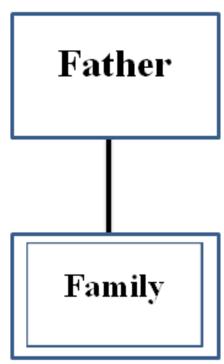
(v). Database: Database is a collection of inter-related data items that can be processed by one or more application systems.

10(a). 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.

10(b). Explain 1NF, 2NF and 3NF.

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'.

11(a). Explain nested queries in SQL.

Sub query or Inner query or Nested query is a query in a query.

A sub query is usually added in the WHERE Clause of the SQL statement. Most of the time, a sub query is used when the user is familiar with the method of posting the query but not sure with the value. When one know how to search for a value using a SELECT statement, but do not know the exact value.

Sub queries are an alternate way of returning data from multiple tables.

Sub queries can be used with the following SQL statements along with the comparison operators like =, <, >, >=, <= etc.

Sub queries are an alternate way of returning data from multiple tables.

A sub query is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved. Sub queries can be used with the SELECT, INSERT, UPDATE, and DELETE statements along with the operators like =, <, >, >=, <=, IN, BETWEEN etc.

There are a few rules that sub queries must follow:

- Sub queries must be enclosed within parentheses.
- A sub query can have only one column in the SELECT clause, unless multiple columns are in the main query for the sub query to compare its selected columns.
- An ORDER BY cannot be used in a sub query, although the main query can use an ORDER BY. The GROUP BY can be used to perform the same function as the ORDER BY in a sub query.
- Sub queries that return more than one row can only be used with multiple value operators, such as the IN operator.
- The SELECT list cannot include any references to values that evaluate to a BLOB, ARRAY, CLOB, or NCLOB.
- A sub query cannot be immediately enclosed in a set function.
- The BETWEEN operator cannot be used with a sub query; however, the BETWEEN can be used within the sub query.

Usually, a sub query should return only one record, but sometimes it can also return multiple records when used with operators like IN, NOT IN with the where clause. The query would be like,

<u>Sub queries with the SELECT Statement:</u>

Sub queries are most frequently used with the SELECT statement. The basic syntax is as follows:

```
SELECT COLUMN_NAME [, COLUMN_NAME ] FROM   TABLE1 [, TABLE2 ]
WHERE  COLUMN_NAME OPERATOR
          (SELECT COLUMN_NAME [, COLUMN_NAME ] FROM TABLE1 [,
          TABLE2 ] [WHERE])
```

<u>Example without SUB Query</u>

```
SELECT FIRST_NAME, LAST_NAME, SUBJECT
FROM STUDENT_DETAILS
WHERE GAMES NOT IN ('CRICKET', 'FOOTBALL');
```

The output would be similar to:

first_name	last_name	subject
Shekar	Gowda	Badminton
Priya	Chandra	Chess

Consider the student_details table used in previous example. The query is executed if the names of the students are known to the user who are studying science subject, you can get their id's by using this query below,

```
SELECT ID, FIRST_NAME FROM STUDENT_DETAILS
WHERE FIRST_NAME IN ('RAHUL', 'STEPHEN');
```

In case if the names if the students are not known, then to get their id's the following sub query is used.

```
SELECT ID, FIRST_NAME FROM STUDENT_DETAILS WHERE FIRST_NAME IN
(SELECT FIRST_NAME FROM STUDENT_DETAILS
WHERE SUBJECT= 'SCIENCE');
```

OUTPUT:

ID	FIRST_NAME
100	RAHUL
102	STEPHEN

In the above SQL statement, first the inner query is processed first and then the outer query is processed.

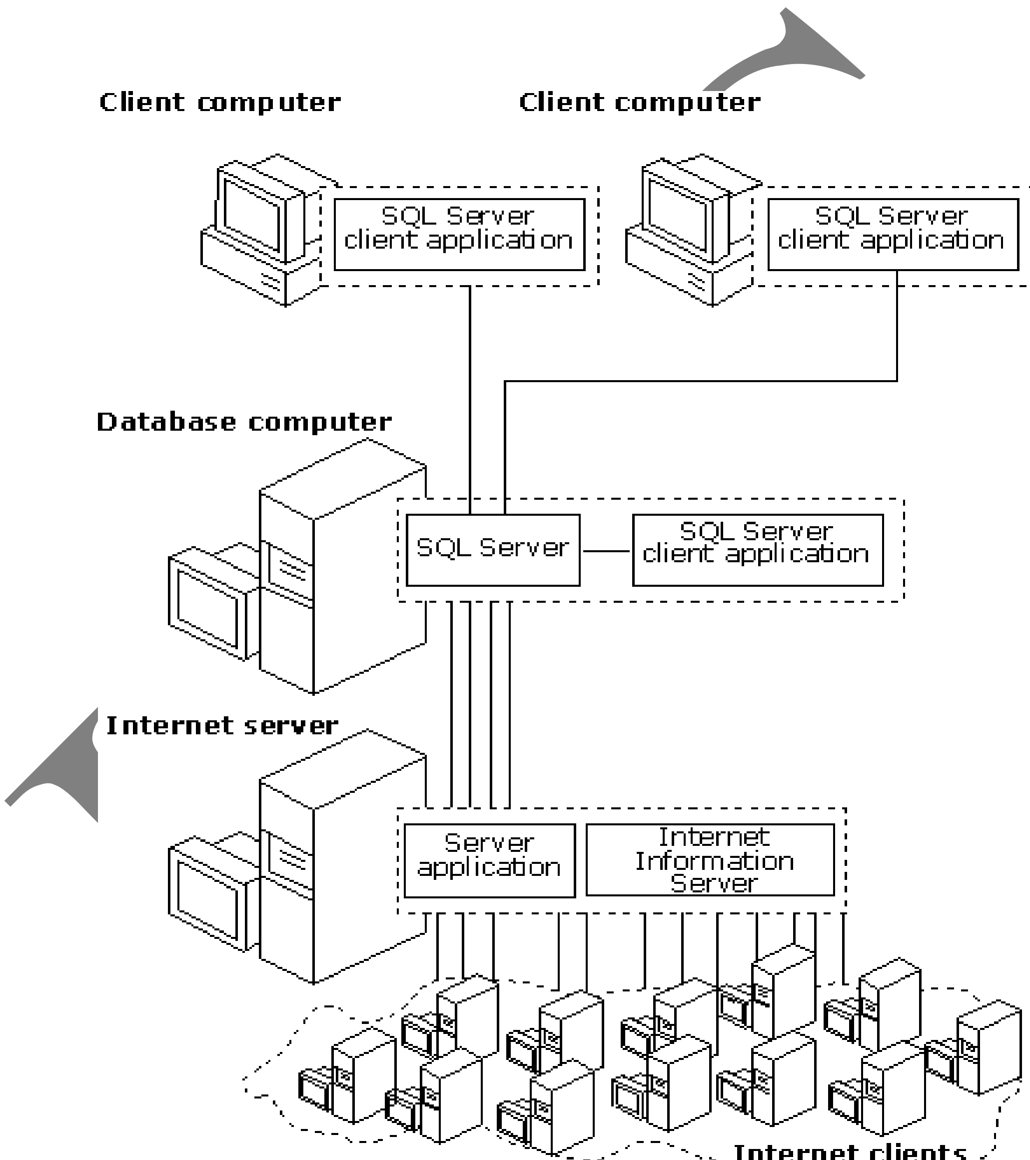
11(b). Explain Client Server Database System.

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.



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). Goals of DBA.

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.

12(b). Write about Codd Rules in Database.

In 1985, Dr.E.F.Codd published a list of 12 rules to define a relational database system. The reason Dr.Codd published the list was his concern that many vendors were marketing products as relational even though those products did not meet minimum relational standards.

Information: All information in a relational database must be logically represented column values in rows within tables.

Guaranteed Access: Every value in a table is guaranteed to be accessible through a combination of table name, primary key and column name.

Systematic Treatment of Nulls: Nulls must be represented and treated in a systematic way, independent of data type.

Dynamic On-Line Catalog Based on the Relational Model: The metadata must be stored and managed as ordinary data, that is, in the table.

Comprehensive Data Sublanguage: The relational database may support many languages. However, it must support one well defined, declarative language with support for data definition, view definition, data manipulation (interactive and by program), integrity constraints, authorization and transaction management (begin, commit and rollback).

View Updating: Any view that is theoretically updatable must be updatable through the system.

High-Level Insert, Update and Delete: The database must support set-level inserts, updates and deletes.

Physical Data Independence: Application programs and ad hoc facilities are logically unaffected when physical access methods or storage structures are changed.

Logical Data Independence: Application programs and ad hoc facilities are logically unaffected when changes are made to the table structures that preserve the original table values (changing order of column or inserting columns).

Integrity Independence: All relational integrity constraints must be definable in the relational language and stored in the system catalog, not at the application level.

Distribution Independence: The end users and application programs are unaware and unaffected by the data location (distributed or local database).

Non Subversion: It the system supports low-level access to the data, there must not be a way to bypass the integrity rules of the database.

Rule Zero: All preceding rules are based on the notion that in order for a database to be considered relational, it must use its relational facilities exclusively to manage the database.

13(a). Explain about 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

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

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 functions and capabilities of DBMS.

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.