Relational Database Management Systems – Nov 2011

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

1. Write short notes on the following:
   (a) Data Redundancy and Inconsistency
   (b) Database

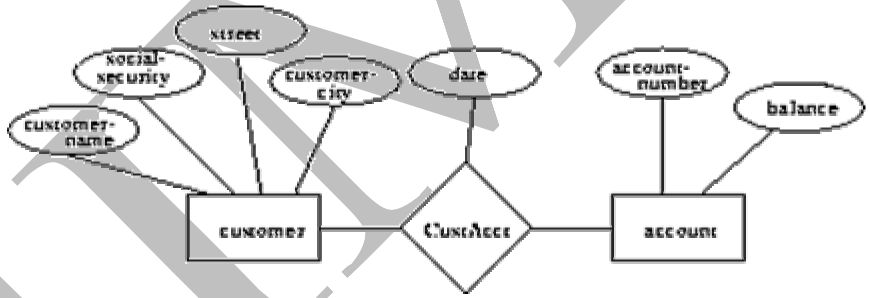
(i). Data redundancy and Inconsistency: Since files and application programs are created by different programmers over a long period of time, the files are likely to be have different formats and the programs may be written in several programming languages. Moreover, the same piece of information may 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.

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

2. Define object, relation with example.

Object: Object represents things that are important to users in the portion of reality we want to model. A set of things of the same kind are called as object sets.

Relation: It is a representation between the objects.



3. Explain the difference between table and view.

<u>Table</u>: 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))
```

<u>View:</u> A definition of a restricted portion of the database. Views are useful for maintaining confidentiality and restricts access to selected parts of the database and for simplifying frequently used
query types.

The formats of create view command is

Create view V As select statement
As = query specification
V= view name

Create view stud-view as
(Select stud-no, stud-name
From student
Where percentage<=35)

A view is a database object that represents one or more database tables. It doesn't occupy any table Space.

4. Explain about 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. Resorting 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 resorting 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.

5. Write about Secondary Key Access.

A key is used to retrieve the value uniquely. Technically this is called as Primary Key. There is also a key called Secondary Key which identifies a set of records having the same value for a column. Secondary key plays an important role in supporting the information requirements of DBMS users.
Ex: To get hall-ticket, one can give their Name and Date of Birth.

6. Explain about client and server architecture.

The term Client/Server (CS) involves multiple computers connected in a network is a concept of CS systems is that one or more of these computers may function as a provider of services to the remaining computers, which function as Clients that process applications. Client and Server have more formally been defined as follows:

Client: - A computer or workstation attached to a network that is used to access network resources.

Server: - A computer that furnishes clients with services such as database, connection to a network, or large disk drives. Server can be mainframes, minicomputers, large workstations or LAN devices. More than one server can be involved in providing services to clients.

7. Write about 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.

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.

8. Explain the advantages of Distributed Database System.

Advantages: -

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 operated.
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.

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

9 (a). Explain the advantages of DBMS over file oriented System.

The typical file-oriented system is supported by a conventional operating system. Permanent records are stored in various files and a number of different application programs are written to extract records from and add records to the appropriate files.

The following are the disadvantages of File-Oriented System:

(i). Data redundancy and Inconsistency: Since files and application programs are created by different programmers over a long period of time, the files are likely to be have 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.

(ii). Difficulty in accessing data: the conventional file processing environments do not allow needed data to be retrieved in a convenient and efficient manner. Better data retrieval system must be developed for general use.

(iii). Data isolation: Since data is scattered in various files, and files may be in different formats, it is difficult to write new application programs to retrieve the appropriate data.

(iv). Concurrent access anomalies: In order to improve the overall performance of the system and obtain a faster response time, many systems allow multiple users to update the data simultaneously. In such an environment, interaction of concurrent updates may result in inconsistent data.

(v). security problems: Not every user of the database system should be able to access all the data. For example, in banking system, payroll personnel need only that part of the database that has information about various bank employees. They do not need access to information about customer accounts. It is difficult to enforce such security constraints.

(vi). Integrity problems: The data values stored in the database must satisfy certain types of consistency constraints. For example, the balance of a bank account may never fall below a prescribed amount. These constraints are enforced in the system by adding appropriate code in the various application programs. When new constraints are added, it is difficult to change the programs to enforce them. The problem is compounded when constraints involve several data items for different files.

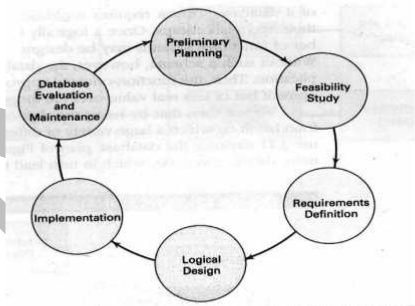
(vii). Atomicity problem: A computer system like any other mechanical or electrical device is subject to failure. In many applications, it is crucial to ensure that once a failure has occurred and has been detected, the data are restored to the consistent state existed prior to the failure.

9(b). List and describe each of the six steps in database development life cycle.

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.

10(a). (i).Explain the different Integrity Constraints.
(ii). Write short note on Functional Dependency.

(i) Integrity Constraints: The different integrity constraints are:

1. Domain Constraint
2. Entity Constraint
3. Referential Integrity

(ii). Functional Dependency: A functional dependency can be described as follows:

- An attribute is functionally dependent if its value is determined by another attribute.
- That is, if we know the value of one data items, then we can find the value of another.
- Functional dependencies are expressed as X→Y, where X is the determinant and Y is the functionally dependent attribute.

10 (b). Explain how to transform a Conceptual model to Relational Model.

11(a). Explain the following commands in SQL with examples.

(a) Create (b) Insert (c) Alter (d)Update (e)Delete

(a) Create: This command is meant for creation of a table
General syntax:

```
Create table tablename
(
 Col1 Datatype(size),
 Col2 Datatype(size),
  ...
 Coln Datatype(size)
);
```

```
Eg. Create table emp
      (
       Empno number(5),
       Ename varchar(20),
       Sal number(7,2),
       Dob date
      );
```

(b) Insert: To get data into a database, we need to use the ‘insert’ statement. The general syntax is:

```
INSERT INTO <table-name> (<column1>,<column2>,<column3>,...)
VALUES (<column-value1>,<column-value2>,<column-value3>);
```

```
Eg.  Insert into emp(empno,ename,sal) values(200,’abc’,10000);
```

(c) Create: This command is meant for alter the structure of a table
General syntax:

```
Alter table tablename
Add columnname Datatype(size);
```

```
Eg. Alter table emp
      Add commission number(6,2);
```

(d)Update: The update statement is used for changing records.
The general syntax is:

```
UPDATE <table-name>
SET <column> = <value>
[where condition]
```

```
E.g. Update emp
       set sal=sal+1000;
```

The above statement increases the salary of all employees by Rs.1000.

```
e.g. Update emp
       set sal=sal+1500
```

```
      where job='clerk';
```

The above statement increases the salary of clerical employees by Rs.1500.

11(b). Define a view with an example. Explain the different operators on a view.

View: A definition of a restricted portion of the database. Views are useful for maintaining confidentiality and restricts access to selected parts of the database and for simplifying frequently used
query types.

The formats of create view command is

```
Create view V As select statement
        As = query specification
         V= view name
```

```
Create view stud-view as
(Select stud-no, stud-name
 From student
Where percentage<=35)
```

A view is a database object that represents one or more database tables. It doesn't occupy any table Space.

Views are mechanism of data independence. They provide automatic security of hidden data. A view is window to a specific use of the database.

Usage of views:

- Views are used to secure attributes that contain private information.
- Views can be used for creating short cuts for complex queries involving multiple tables.
- Views can be created with a Check option that prevents the updating of rows and attributes that are not part of the view but are part of complete database

Creating a View

For creating a view following command is used:

Syntax: CREATE or REPLACE VIEW <view-name> AS <query>;

```
Example:    CREATE VIEW emp_view AS
                    SELECT e_code, e_name FROM employee;
```

Please note that while defining a view an ORDER BY clause cannot be used.

Displaying the view

For displaying all the attributes of the View give the following command:

```
SELECT * FROM emp_view;
```

For displaying the attributes of the view use the following command:

```
DESC emp_view;
```

Deleting a View

For deleting a View please use the following command:

```
DROP VIEW emp_view;
```

Updating through a View

The data through the views cannot be updated if it include JOIN, SET operations, GROUP BY, DISTINCT, AGGREGATE function; or have composite attributes; or does not contain the field of the base table that contains NOT NULL including primary key; or violates WITH CHECK OPTION clause.

12(a). Explain the disk performance factors.

There are four factors that directly affect the speed with which data are transferred to and from disk storage: access motion time, head activation time, head activation, rotational, delay and data transfer rate.

1) Access motion Time: - The time required to move the read/write heads of the disk drive over the desired cylinder. Sometimes termed seek time, is the time required to move the read/write heads from their current position to a new cylinder address. Obviously, a move to an adjacent cylinder will not take the same amount of time as a move across the entire disk surface (inner most track to outer most track, or vice versa). As a compromise in calculation, the average access motion time may be used-roughly the time required to move across one-half of the cylinders, although more sophisticated methods may be used. A standard assumption is that the likelihood of access for every record is the same giving a uniform probability distribution. The average for a uniform distribution is halfway between the extreme values. For access motion time, the extreme would be (1) Stay positioned over the current cylinder, or (2) Move from the inner most cylinders to the outer most or (vice versa). Given the uniform distribution assumption, the average will be the time to move across one-half of the cylinders. 12-20 million seconds are typical average access motion time, varying with the make and model of the disk drive.

2) Head Activation Time: - The time required to activate a read /write head. It is the time required to electronically activate the head i.e., over the surface the data transfer is to take place relatively other performance factors, this time generally regarded as being negligible. Consequently, head activation time is seldom used in performance calculations.

3) Rotational Delay: - The time required for the disk to rotate the sought-for record under the read/write head. Rotational delay, or latency, is the third timing factor. It denotes the amount of time required for the desired block to rotate to the head, so that the data transfer may commence. Rotational delay depends upon two factors: How fast the disk is rotating and the location of the block being sought in relationship to the read/write head at time of its activation physically, this time reach from zero to the time required to complete one complete revolution of the disk(R). Performance computations usually assume an average rotational delay of R/2.

4) Data Transfer Rate: - The rate at which data can be read from the disk from the main, memory, or equivalently, the rate at which data are written from main memory to disk. Data Transfer Rate refers to the amount of time required to transfer data from the disk to primary memory. It is a function of rotational speed and the density recorded data. Data Transfer Time is usually expressed in thousands of bytes per second.

5) Data Transfer Time: - The expected time (T) to access a disk address and transfer a block of data is estimated as

$T = A+R/2+L/2$

Where A is the Access motion time, R is the Rotational delay, L is the length of the block in bytes, and D is the Data Transfer Rate.

12(b). Explain the functions of DBA.

DBA functions may generally fall into the areas of communicating with database users; planning, designing, and implementing database systems; and establishing standards and procedures. The planning, designing, and implementing of database systems.

Communicating with Users: - Database systems often have three components: a central, widely used database containing much of the firm's data; several functional database (e.g., for accounting) used by a more limited set of programs; and perhaps a few dedicated database, used for a single application (e.g., a bill-of-materials database). The important organizational issue here is that the general impact of implementing a database system is the centralization of a significant portion of the firm's data.

Centralizing data through a database system tends to eliminate local ownership of data and to reduce redundancy. Ownership and control are transferred to the central data dictionary, which maintains a record of the ownership and use of each data element. Such a shifting of control over data may generate resistance from some users. This resistance can be mitigated by actively educating users as to the advantages of learning database technology: how it can make them more effective and efficient at their jobs. The DBA, in cooperation with top management, should provide this education

Establishing standards and procedures: - organizations having few standards and procedures may encounter difficulty in converting to the database environment, since the record shows that the integrated data management facilitated by database systems requires good, comprehensive standards and procedures. An organization that is beginning to implement a database system may find it useful to examine the standards in use at other organizations that are already using database systems.

1. Analysis and routing of trouble reports: - A formal trouble-reporting system was established in order to report all errors to the DBA. Trouble reports are analyzed to determine the likely cause of each reported problem. The reports are then routed to the appropriate manager or user group for disposition. Each trouble report contains a complete log and descriptive information. Each report requires a formal response to the report's initiator specifying how the problem has been resolved.

2. Monitoring of hardware and software: The status of all hardware and software is regularly monitored, and reports of failures and consequent action are made to appropriate mangers and user groups. Periodic analysis of hardware and software requirements is made, forming the basis for decisions on replacement and upgrading, including needs for additional database storage media.

3. Testing: Performance acceptance testing is conducted to evaluate all new procedures, software, and hardware. Structural and consistency checks of the database are conducted on a regular basis.

4. Security: security classifications are implemented that identify which user groups are authorized to access specific data elements sin the database and what actions may be performed thereon. Computer operations area frequently monitored to assure that these access controls are functioning in the intended way.

5. Backup and recovery: 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.

6. Performance evaluation: Priorities have assigned to activities that compete for database resources, such as processing transactions, generating reports, and processing queries, system performance is monitored by collecting statistics on transaction volume, response time, error rates, and hardware utilization. Input is elicited from system users to monitor their satisfaction with the system's performance. Database size and growth is also tracked. File expansion programs are run and database reorganizations are performed as necessary. Activity logs and abnormal termination logs are reviewed and summaries prepared for management evaluation.

7. Integrity checking: Schedules have been developed for testing the integrity of the date stored in the database.

13(a). Briefly explain important considerations associated with DBMS implementation issues.

13(b). Explain the general model of Distributed Database Design.

Distributed Database Design: 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 design of a distributed database system can be a complex task. Careful consideration must be given to the objectives and strategies to be served by the design and parallel decisions must be made as to how the data are to be distributed among the various network sites.

Components of DDBMS:

1. Computer Work-stations.
2. Network hardware and software components that reside in each work-station.
3. Communication media that carry the data from one work station to another.
4. The Transaction Processor (TP) which is the software component found in each computer that requests data.
5. The Data Processor (DP) which is the software component residing on each computer that stores and retrieves the data located at the site.
6. Protocols to transport data commands between DP and TP.

Objectives that are common to most implementation of distributed database systems are:

1) Location Transparency: - Location transparency enables a user to access data without knowing, or being concerned with, the site at which the data reside. The location of the data is hidden from the user.

2) Replication Transparency: - Replication transparency means that when more than one copy of the data exists, one copy must be chosen when retrieving data, and all copies must be updated when changes are made. Choosing one copy of the data for retrieval and always ensuring that all copies of the data are updated can be a burden on users. A DBMS should handle all such requirements, thereby freeing the user to concentrate on information needs.

3) Configuration Independence: - Configuration independence enables the organization to add or replace hardware without changing the existing software components of the DBMS. Configuration independence results in a system that is expandable when its current hardware is saturated.

4) Non-homogeneous DBMSs: - It is sometimes desirable to integrate database maintained by different DBMS on different computers. Often the DBMS are supplied by different vendors and may support different data models. One approach to integrating this database is to provide a single user interface that can be used to access the data maintained by the non-homogenous DBMS. The different data models supported by the non-homogenous DDBMS are hidden from the user by this single, system wide interface.

5) Data Replication: - Data replication occurs if the system maintains several identical copies of a relation, R, with each copy being stored at a different site. Typically replication is introduced to increase the availability of the system: When a copy is unviable due to site failure(s), it should be possible to access another copy.