Data: (1Mark-2012)

Data is a representation of a Facts, figures, statistics etc. having no particular meaning. Data can be in the form of numbers, characters, symbols, or even pictures.

Data Base: (1Mark-2014)

A Database is a collection of information (related data) that is organized so that it can easily be accessed, managed, and updated.

DATABASE APPROACH:

Implicit Properties of Database Approach: (2Marks- 2015)

- A database represents some aspect of the real world, sometimes called the mini world
- A database is a logically collection of data with some inherent meaning
- A database is designed, built with data for a specific purpose. It has an intended group of users and some applications in which these users are interested

Characteristics of database approach:(6M-2015)

1. Self-describing nature of a database system
2. Insulation between programs and data
3. Data Abstraction
4. Support of multiple views of the data
5. Sharing of data and multiple-user transaction processing

DATABASE MANAGEMNENT SYSTEM (DBM): (1M-2013)(2M-2015)(2M-2016)

- **A database management system (DBMS)** is a collection of programs that enables user to create and maintain a database
- **DBMS** is a software package with computer programs that control the creation, maintenance, and the use of a database.
- A DBMS allows different user application programs to concurrently access the same database

Functions of DBMS:(3M-2011) (3M-2013) (6M-2016)

Data Definition:

The DBMS provides functions to define the structure of the data in the application

Data Manipulation:

Once the data structure is defined, data needs to be inserted, modified or deleted

Data security & Integrity:

The DBMS contains functions which handle the security and integrity of data in the application

Data recovery & concurrency:

Recovery of data after a system failure and concurrent access of records by multiple users are also handled by the DBMS

Data Dictionary Maintenance:

Maintaining the data Dictionary which contains the data definition of the application is also one of the functions of a DBMS

Performance:

Optimizing the performance of the queries is one of the important functions of DBMS

Capabilities/Advantage of DBMS: (5M- 2012)

- ✓ Reduced data Repetition
- ✓ Restricting Unauthorized Access
- ✓ Providing Persistent Storage of Program Objects
- ✓ Providing Storage Structure for Efficient Query processing
- ✓ Providing Backup and Recovery
- ✓ Multiple User Interfaces
- ✓ Representing Complex Relationships Among Data
- ✓ Enforcing Constraints

Disadvantages of DBMS: (5M-2012)

- **x** Cost of Hardware and Software
- **x** Cost of Data Conversion
- **x** Cost of Staff Training
- **x** Appointing Technical Staff
- **x** Database Damage

Applications of DBMS: (5M-2014)

- Effective processing complex data and data with set of the references for expression of the relations between them.
- Building of internet-shops and distributed information systems.
- Building of the virtual company office and virtual kiosks.
- Storage and reproduction of graphic images, video and audio.
- Creation of WEB-sites, allotted to unlimited opportunities.

DBMS users (different User behind DBMS): (7M-2013)

Actors on the Scene:

1. Database Administrator (DBA)
2. Database Designers
3. End Users
4. Systems Analysts and Application programmers (Software Engineers)

Workers behind the scene:

5. DBMS system designers and implementers
6. Tool developers
7. Operators and maintenance personal

DATABASE ADMINISTRATOR (DBA): (3M-2012)

Database administration is responsible for physical database design, security and database performance

Roles and Responsibilities of DBA:(4M-2011)(5M-2014)(2M-2015)

- Installation, configuration and upgrading of Oracle server software and related products
- Evaluate oracle features and oracle related products
- Establish and maintain sound backup and recovery policies and procedures
- Take care of the Database design and implementation
- Implement and maintain database security
- Perform database tuning and performance monitoring
- Perform application tuning and performance monitoring
- Setup and maintain documentation and standards
- Work as part of a team and provide 7x24 support when required
- Interface with Oracle Corporation for technical support.

Data Model: (1M-2011)

Data Model is Collection of concept for describing the structure of database and relationships between data, and constraints on the data in an organization

Categories of data models:

1. **High level/conceptual data models** – Provide concept close to the way users perceive the data.
   - **Entity:** represents a real world object or concept
   - **Attribute:** represents property of interest the describes an entity such as name or salary
   - **Relationships:** among two or more entities, represent an association among two or more entities
2. **Low level/ physical data models** – provide concepts that describe the details of how data is stored in the computer.
3. **Representational/ implementation data models** – provide concepts that may be understood by the end user but far removed from the way data is organized.

DIFFERENT TYPES OF DATA MODELS: (1M-2013)

1. Hierarchical data model
2. Network data model
3. Relational data model
4. Object-Oriented data model
5. Object – relational data model

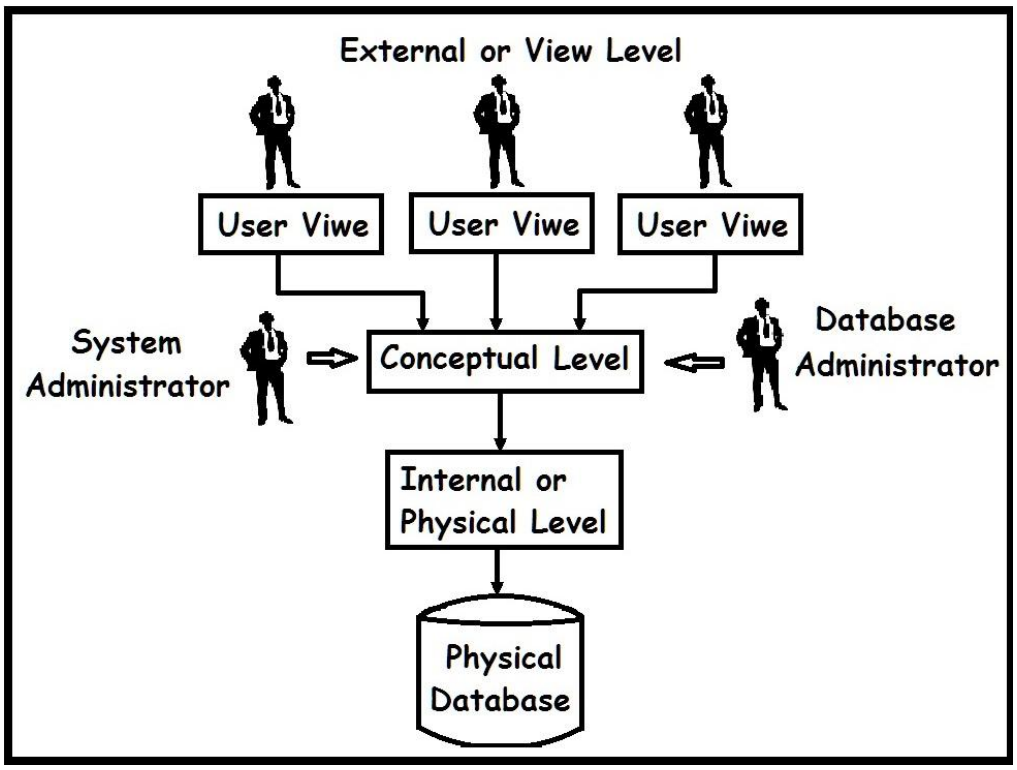
Network Data Model: (1 M -2014)

The network Model replaces the hierarchical tree with graph thus allowing more general connection with the nodes.

Database schema (Meta-data): (1M- 2012) (1M-2013) (1M-2014) (2M-2016)

Differentiate between the description of the database and the database itself. The description of the database is called database schema (or) metadata.

Logical DBMS Architecture(Three Schema/Level DBMS Architecture)(3M-2012)(7M-2013)



1. The External (or) View Level:

External level is the one which is closest to the end users each users are given different views according to the user's requirement.

2. The conceptual Level:

The conceptual level represents

- All entities, their attributes, and their relationships;
- The constraints on the data;
- Semantic information about the data;
- Security and integrity information.

3. The internal (or) Physical Level:

This level describes how the data the way the data are physically stored on the hardware.

The internal level is concerned with such things as:

- Storage space allocation for data and indexes;
- Record descriptions for storage (with stored size for data items);
- Record placement
- Data compression and data encryption techniques.

♥ **DATA INDEPENDENCE:** (3M-2011)(1M-2012)(3M – 2013)(4M-2015)(4M-2016)

Definition: Data independence:

The ability to modify a schema in one level without affecting schema in the next higher level is called "Data independence"

There are two kinds of data independence:

1. **Physical Data independence:**
   - The ability to change the internal schema without having to change the conceptual schema
   - Physical file reorganization to improve performance (such as creating access structures)
   - The physical data independence allows changes in the physical storage devices or organization of the files to be made without requiring changes in the conceptual view (or) any of the external views)

1. **Logical data independence:**
   - The ability to change the conceptual schema without having to change the external schemas or application programs.
   - If the conceptual schema undergoes a logical reorganization, application programs that reference the external schema constructs must work as before.
   - Logical data independence indicates that the conceptual schema can be changed without affecting the existing external schemas.

DATABASE LANGUAGES: (3M-2014)

DDL (Data Definition Language): (2M – 2013)

DDL is used by the DBA (Database Administrators) and by database designer to define both schemas. The DBMS will have a DDL compiler, whose function is to process DDL statements in order to identify descriptions of the schema construct and to store the schema descriptions in the DBMS catalogue.

DML (Data Manipulation Language): (3M – 2013) (1M-2014)

DML stands for Data Manipulation Language. DML is a language that enable users to access or manipulates date, as organized by the appropriate data model.

DCL (Data Control Language):

DCL is used to control access to data stored in a database. DCL statement s executed based on the type of action to be granted (or) revoked. Grant to allow specified users to perform specified tasks and revoke to cancel previously granted permission.

TCL (Transection Control Language):

Statements which are used to manage the changes made by DML statements are called Transaction Control Language. Used to manage transactions in a database. It allows statements to be grouped together into logical transactions.

■■■■■■■(Chapter3- Data Modeling Using Entity- Relationship Model)■■■■■■■

ENTITY:(1 M-2012) (1M-2014)

An entity is any object in the system that we want to model and store information about individual object is called entities.

ATTRIBUTES: (1M – 2014)

An attribute is a property of an entity. For Ex: a person has an age, a car has a colour.

TYPES OF ATTRIBUTES: (3M-2011)(3M-2013)(2M-2016)

1. **Simple Attribute:**
   Simple attribute that consist of a single value. A simple attribute cannot be subdivided. **For example** the attributes age, sex etc.
2. **Composite Attribute:**
   A composite attribute is an attribute that can be further subdivided **For example** the attribute ADDRESS can be subdivided into street, city, state and zip code.
3. **Single-valued Attribute:**
   A single values attribute can have only a single value. **For example** a person can have only one 'date of birth', 'age' etc. A single valued attributes can have only single value. But it can be simple or composite attribute.
4. **Multi-value Attribute:**
   Multivalued Attributes can have multiple values. **For example** a person may have multiple phone numbers
5. **Stored Attribute:**
   Stored Attribute that supplies a value to the related attribute. **For example** 'Date of birth' of a person is a stored attribute.
6. **Derived Attribute:**
   The value for the derived attribute is derived from the stored attribute. **For example** 'Date of birth' of a stored attribute.
7. **Complex Attribute:**
   Complex Attribute that is both composite and multi valued.
8. **Null Value Attribute:**
   A particular entity may not have an applicable value for an attribute. **For example**, the apartment_number

Types of Entity:

Strong Entity:

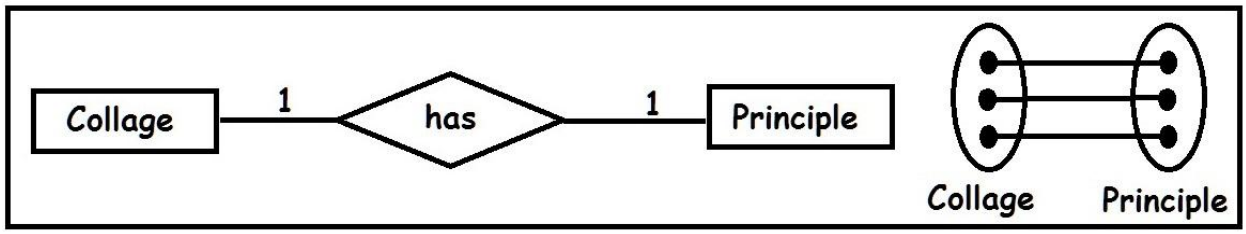
- Entity types that have primary keys are called strong entity type.

Weak Entity:(1M-2012) (4M-2014)

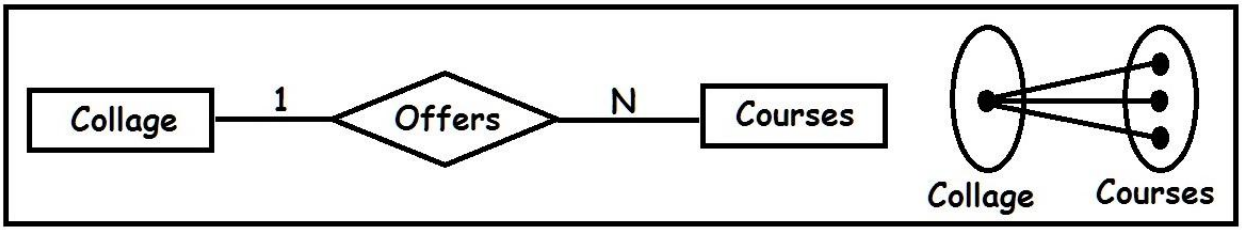
- Entity types that do not have key attribute of their own are called weak entity type.

Relationship Between Entity sets: (3M-2013) (3M-2015)

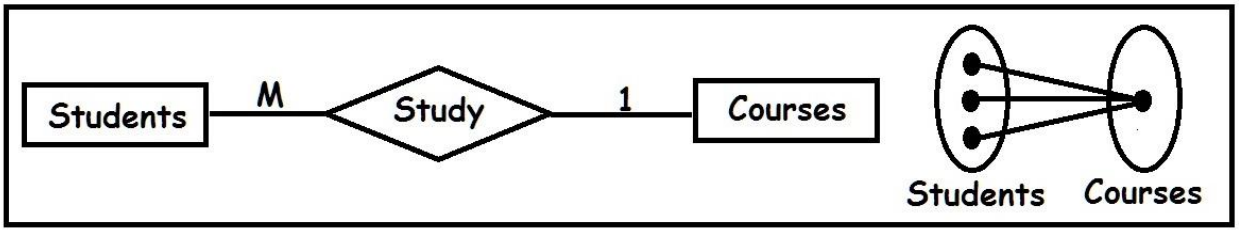
1. **One-to-One Relationship (1:1):** **For ex: (1:1)** college can have only one principle



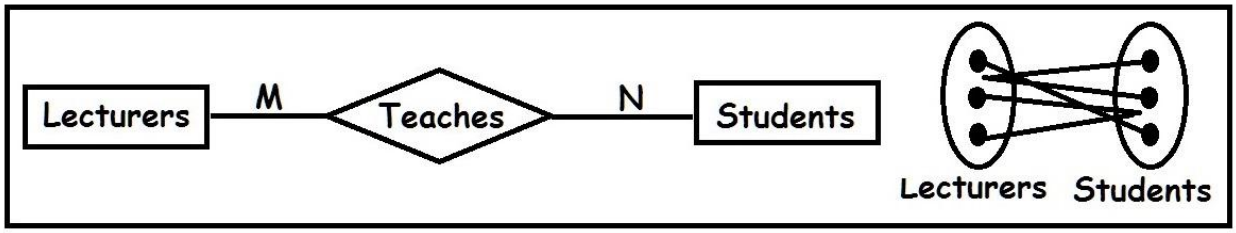
2. **One-to-Many Relationship (1:N):** **For ex: (1:N)** Collage offers many courses



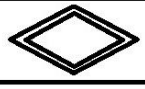
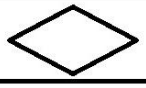
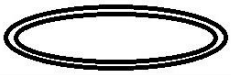
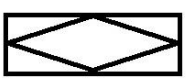
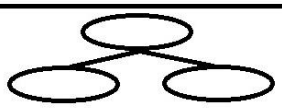
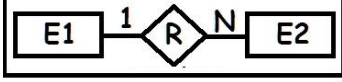
3. **Many-to-One Relationship (M:1):** **For ex: (M-1)** STUDENTS STUDY COURSE



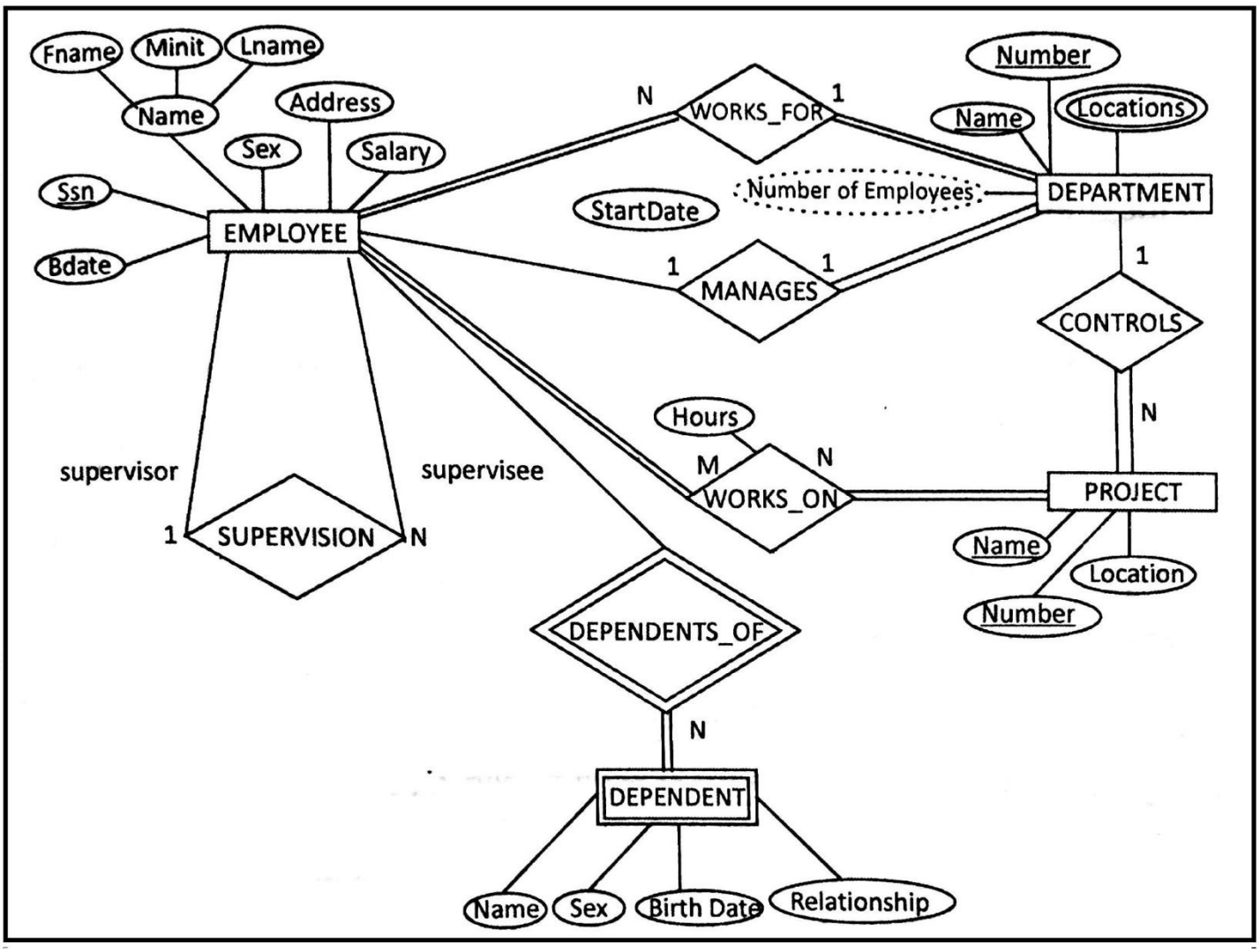
4. **Many-to-Many Relationship (M:N):** **For ex: (M:N)** students enrolled in a classes



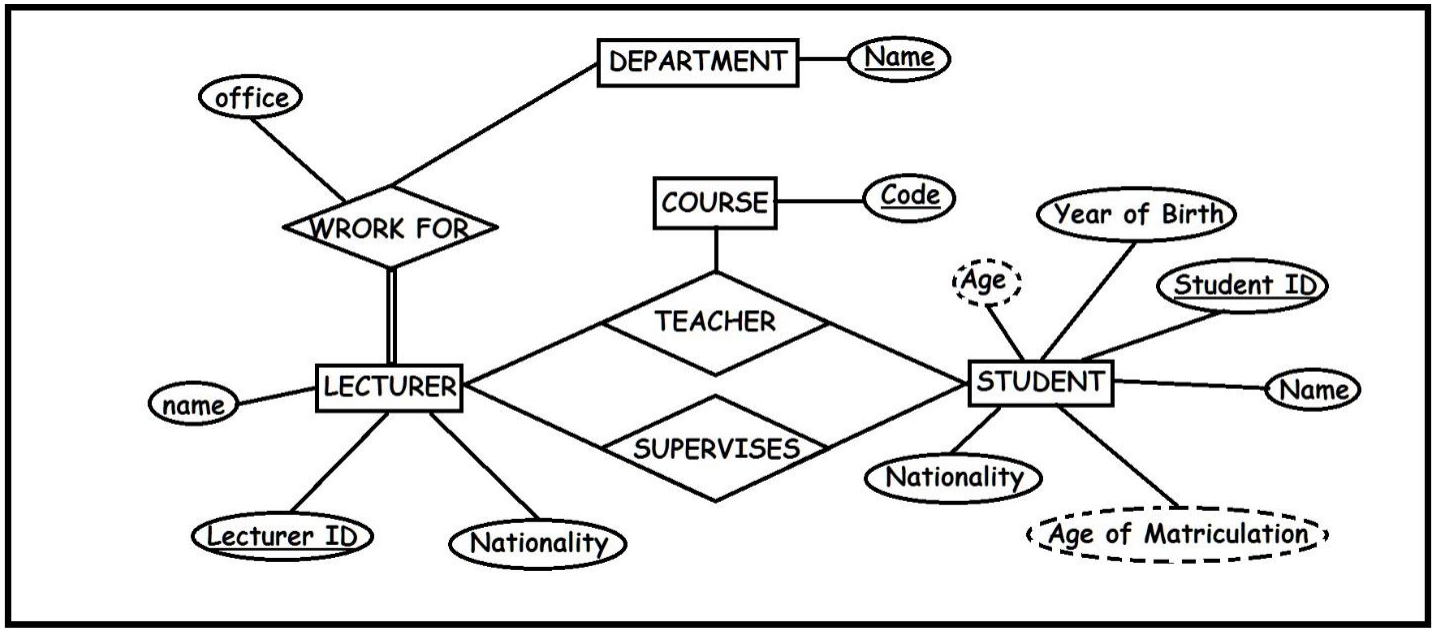
ER DIAGRAM NOTATION: (7M-2012) (5M-2014) (5M-2016)

Symbol	Meaning	Symbol	Meaning
	WEAK ENTITY		ENTITY
	IDENTIFYING RELATIONSHIP		RELATIONSHIP
	KEY ATTRIBUTE		ATTRIBUTE
	MULTIVALUED ATTRIBUTE		DERIVED ATTRIBUTE
	ASSOCIATIVE ENTITY		COMPOSITE ATTRIBUTE
	Cardinality ratio 1:N for E1:E2 in R		Total Participation of E2 in R

ER DIAGRAM for Company: (5M-2015)



ER DIAGRAM for university: (10m-2012) (5M-2014)



ABSTRATION: (1M-2011) (2M-2016)

Data Abstraction: (1M-2013)

Hides details of data storage that are not needed by most database users and applications

The two main aspects of abstraction are:(2M-2016)

1. **Specialization:**(1M-2011)
   - The process of designating sub-groupings within an entity set is called specialization
2. **Generalization**: (1M-2011)
   - Generalization proceeds from the recognition that a number of entity sets share some common feature are described by the same attributes and participate in the same relationship sets.

■■■■■■■■(Chapter4- Record Storage and Primary file Organization)■■■■■■■■

STORAGE DEVICES: (3M-2011)(4M-2015)(2M-2016)

- The Hardware that writes data to (or) reads data from a storage medium is called storage device

Computer storage media form a storage hierarchy that includes two main categories

1. Primary Storage:

This category includes storage media that can be operated on directly by the computer Central processing unit (CPU) such as the computer main memory. **Primary storage is volatile and temporary**. Primary storage usually provides fast access to data but limited storage capacity.

2. Secondary storage:

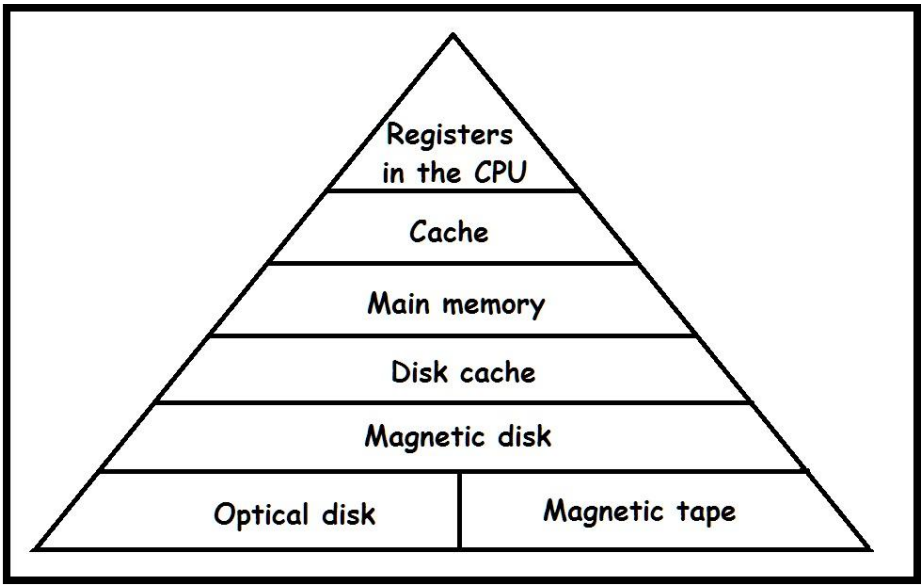
Data in secondary storage cannot be processed directly by the CPU, first copied into primary storage.

Example: Magnetic disks and Optical Disks. These devices usually have a larger capacity, cost less and provide slower access to data than primary storage devices.

Organization of Disk:(1M-2013) (3M-2014)

- Disk contains concentric **tracks**. Tracks are divided into sectors.
- A **sector** is the smallest addressable unit in a disk.
- A **Block** is a physical data record separated on the medium from other blocks by inter-block gaps.
- A **Cylinder** is the set of matched tracks at a given radius of a disk pack.

MEMORY HIERARCHIES: (7M-2012)(7M-2014)



Registers in the CPU:

A special high-speed storage area within the CPU, All data must be represented in a register before it can be processed

Cache Memory:

Cache memory is random access memory (RAM) that a computer microprocessor can access more quickly it can access regular RAM. **There are Two types of RAM are:** Static RAM **&** Dynamic RAM

Main memory:

Main memory is the storage that is directly available to the CPU of a computer and is made up RAM and ROM. (**R**andom **A**ccess **M**emory & **R**ead **O**nly **M**emory)

Dick Cache:

- A disk cache is a mechanism for improving the time it takes to read from or write to a hard disk
- The disk cache holds data that has recently been read and, in some cases, adjacent data areas that are likely to be accessed next.

Magnetic Dick:

A memory device such as a floppy disk, a hard disk (or) a removable cartridge that is covered with a magnetic coating on which digital information is stored in the form of microscopically

Optical Dick:

Optical disks record data by burning microscopic holes in the surface of the disk with a laser.

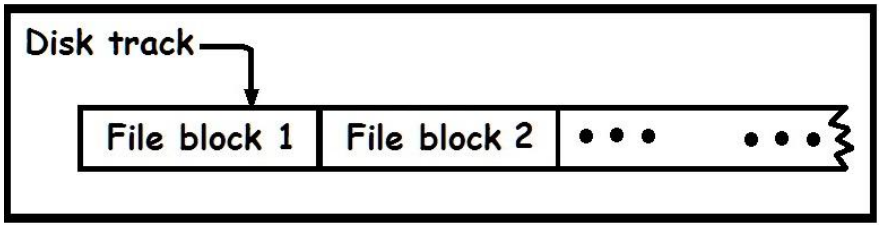
Magnetic Tape:

Magnetic tapes are used for archiving and backup storage of data. Magnetic tape is an external storage device that can be used for making copies of audio, video and data.

ALLOCATING FILE BLOCKS ON DISK: (6M-2015)(6M-2016)

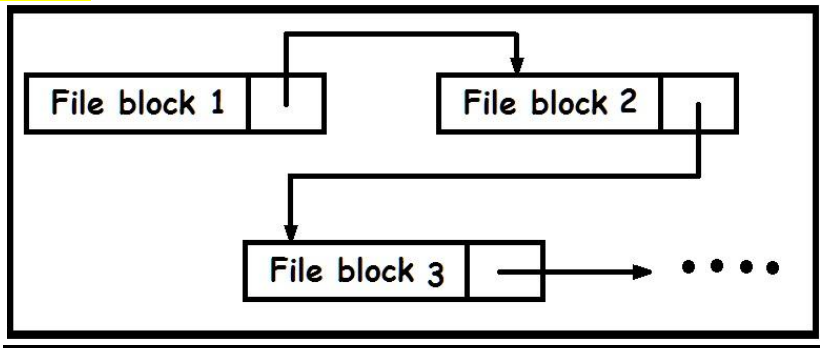
There are various methods for allocating the blocks of a file on disk are

1. Contiguous Allocating:



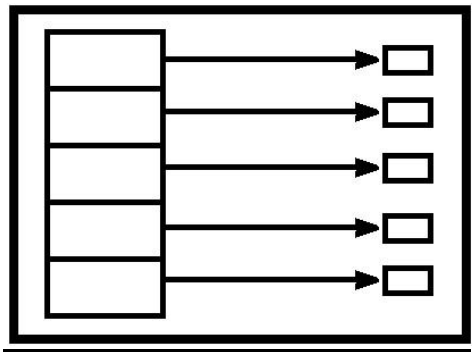
- Requires each file to occupy a set of contiguous blocks on the disk
- Disk addresses defile a linear ordering on the disk
- This makes reading the whole file very fast using double buffering, but it makes expanding the file difficult.

2. Linked Allocation:



- Solves all the problems of contiguous allocation
- Each data block contains the block address of the next block in the file
- The directory contains a pointer to the first and last Blocks of a file
- A file can grow as long as free blocks are available

3. Indexed Allocation:



- Solves the problems of contiguous allocation
- Supports direct access by bringing all the pointers together into the index block
- Each file has its own index block, which in an array of disk- block addresses
- In this one or more index blocks contain pointers to the actual file blocks.

FILES OF UNORDERES RECORDS (HEAP FILES): (1m-2011) (2M-2015)

Records are placed in the file in the order in which they are inserted, so new records are inserted at the end of the file are called heap (or) Pile file.

HASHING TECHNIQUES: (7M-2013) (7M-2011)

- Hashing techniques is the efficient method in the searching to the exact data item in a very short time.
- Hashing is used to index and items in a database because it is faster to find the item using the shorter hashed key than to find it using the original value.

Hashing (1M-2014)

Hashing is the process in which we place the each data item at the index of the memory location for the purpose of ease of usability

Advantage of Hashing

- ✓ Hash tables are efficient and speed, when the number of entries is large (thousand or more)

Disadvantage of Hashing

- x Hashing table are not effective when the number of entries is very small

There are two types of hashing:

1) INTERNAL HASHING:

- ➢ Hashing for an internal file
- ➢ Hash table as an array of records

2) EXTERNAL HASHING:

- ➢ External hashing makes use of buckets, each of which can hold multiple records.

RAID TECHNOLOGY (1M-2012) (1M-2016)

- **RAID** – **S**tands for **R**edundant **A**rrays of **I**nexpensive **D**isks (or) Redundant Arrays of independent Disks.
- RAIDs are used to provide increased reliability increased performance or both.

■■■■■■■■■■■■■■■■■■■■■■■■(Chapter 5)■■■■■■■■■■■■■■■■■■■■■■■■

■■■(Functional Dependencies And Normalization for Relational Databases)■■■

Key Constraints:(3M-2012)

A key Constraint is a statement that a certain minimal subset of the fields of a relation is a unique identifier for a tuple.

Super Key: (2M-2014)

An Attribute that is used to identify the records uniquely is known as Super key. A table can have many Super Keys. Example: Dictator ID, Name (Dilatory), MovieID,Titel(Movie table).

Candidate Key:(1M-2012)

Attribute that identifies the record uniquely but none of its proper subset can identify the records uniquely. Example: DitatorID, Name and address (Dictator table).

♥ **Primary Key:** (1M-2011)(1M-2013)(1M-2014)(2M-2016)

A unique identification of each row in a table is known as Primary key. A Primary Key can consist of one (or) more attribute of a table. Example: DirectorID (Director table), MovieID.(Movie table).

Foreign Key:(2M-2014)

A foreign Key is an attribute in one table that point to the primary key of another table. The purpose of the foreign key is to ensure referential integrity of the data. Example DitectorID(Movies table) reference Directors table.

♥ **FUCNTIONAL DEPENDENCIES:**(4M-2011) (1M-2013) (2M-2016) (4M-2015)

- A **functional dependency (FD)** is a constraint between two sets of attributes in a relation from database.
- The determination of functional dependencies is an important part of designing databases in the relational model and in database normalization and de-normalization.
- The functional dependencies, along with the attribute domains are selected so as to generate constraints that would exclude as much data inappropriate to the user domain from system as possible.
- A functional dependency is denoted by X Y between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuple that can from a relation state r or R.

♥ NORMALIZATION: (1M-2012)(3M-2013)(7M-2016)

Nominalization is a process that is carried out to minimize the redundancies that are present in data in relational database. This process will mainly divide large tables in to smaller tables with fewer redundancies. These smaller tables will be related to each other through well defined relationship. In a well normalized database, any alteration or modification in data will requires modifying only a single table.

Normalization is a set of rules and techniques concerned with

- Identifying relationships among attributes.
- Combining attributes to form relations.
- Combining relations to form a database.

NORMAL FORMS

1. First Normal Form (1NF)
2. Second Normal Form (2NF)
3. Third Normal Form(3NF)
4. BCNF
5. Fourth Normal Form (4NF)
6. Fifth Normal Form (5NF)

Properties of Normalization

1. No data value should be duplicated in different rows unnecessarily
2. A value must be specified (and required) for every attribute in a row.
3. Each relation should be self-contained.
4. When a row is added to a relation, other relations in database should not be affected.

Advantage of Normalization:

- ✓ Reduces data redundancy (same data stored many times in same/ different tables.
- ✓ All the update anomalies and does not have any loss of data
- ✓ All data are stored efficiently since there is no redundancy.

CONSTRAINTS ON RELATIONS: (7M-2011)

A property assigned to a table column that prevents certain types of invalid data values from being placed in the column.

CATEGORIES OF CONSTRAINTS:

1. Inherent model based constraints.
2. Schema Based Constraints.
3. Application Based Constraints.
4. Data Dependencies.

INHERENT MODEL BASED CONSTRAINTS:

Constraints that are inherent in the data model are called as inherent model based constraints.

SCHEMA BASED CONSTRAINTS: (5M-2016)

Constraints that can be directly expressed in the schema of the data model, typically by specifying them in the DDL (Data Definition Language), are called as **Schema Based Constraints**.

APPLICATION BASED CONSTRAINTS:

Constraints that cannot be directly expressed in the schema of the data model, and hence must be expressed and enforced by the application programs.

DATA DEPENDENCIES CONSTRAINTS:

Which include function dependence and multivalued dependencies. They are used mainly for testing the "goodness" of the design of a relational database and are utilized in a process of normalization.

DOMAIN CONSTRAINTS: (5M-2015)

Domain constraints in the schema specify an important condition that the user want each instance of the relation to satisfy. Domain integrity helps to ensure the length and type of data, the mull value, the allowable value and default value

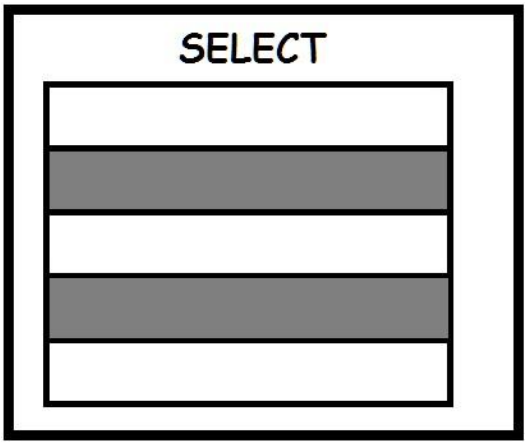
RELATIONAL ALGEBRA:(1M-2014)

The relational algebra is a procedural query language. It consists of a set of operations that take one or more relations as input and produce a new relation as their result.

UNRY RELATIONAL OPERATIONS SELECT & PROJECT: (7M-2014)(5M-2015)(5M-2016)

The SELECT Operation:

The SELECT operation is used to select a subset of the turtles from a relation that satisfy a selection condition. One can consider the SELECT operation to be a filter that keeps only those tuples that satisfy a qualifying condition.

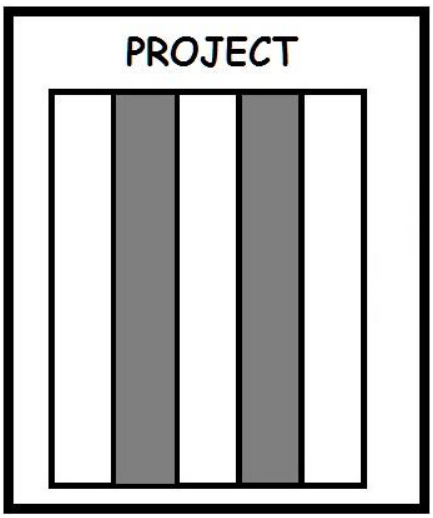


Syntax: σ < section condition > (R)

Where the symbol σ (sigma) is used to denote the SELECT operator and the selection condition is Boolean expression

PROJECT Operation:

The SELECT operation selects some of the rows from the table while discarding other rows. The PROJECT operation, on the other hand, selects certain columns from the table and discards the other columns.



Syntax : Π <attribyte ilst > (R)

Where the symbol Π (pi) is used to represent the PROJECT operation and <attribyte ilst > is the desire list of attribute from the attribute of relation R.

RDBMS: (2M-2016)

- A **R**elational **D**ata**B**ase **M**anagement **S**ystem (**RDBMS**) is a database management system (DBMS) that is based on the relational model introduced by E. F. Codd.
- RDBMS may be a DBMS in which data is stored in the form of tables and the relationship among the data is also stored in the form of tables

SQL:(1M-2011) (1M-2014)

- **SQL** stands for **S**tructured **Q**uery **L**anguage.
- SQL is used to access and manipulate databases.
- SQL is an ANSI (American National Standard Institute) Standard.

TYPES OF SQL STATEMENTS: (2M-2016)

DDL STATEMENTS: (4M-2016)

DDL statements are used to build and modify the structure of database tables and other objects in the database. When DDL statements are expected, it takes effect immediately. The DDL part of SQL permits database tables to be created or deleted

The most important DDL statements in SQL are: (1M-2013) (7M-2014)

- CREATE TABLE – creates a new table,
- AFTER TABLE – modifies a table,
- DROP TABLE – deletes a table,
- TURNCATE – deletes data in a table

THE CREATE STATEMENT:

Create statement is used to create the tables

Syntax	Example
CREATE TABLE table_name	CREATE TABLE suppliers

THE ALTER STATEMENT:(3M-2013)

THE **ALTER TABLE** statement is used to rename an existing table. It can also be used to add new columns and drop a column from an existing table.

Syntax	Example
ALTER TABLE table_name	ALTER TABLE **SUPPLIERS** RENAME to **DISTRIBUTORS**;

THE DROP TABLE STATEMENT:

Purpose: Removes table from database.

Syntax	Example
DROP TABLE [user.] table_name	DROP TABLE **DISTRIBUTORS;**

THE TRUNCATE TABLE STATEMENT:

The TRUNCATE statement deletes all the records from the table

Syntax	Example
TRUNCATE TABLE <Table_Name>	TRUNCATE TABLE **DISTRIBUTORS**

♥ **DML STATEMENT:**(3M-2011)((3M-2013) (5M-2015) (3M-2016)

Data Manipulation Language (DML) statements are used to work with the data in tables. These statements are used to append, change or remove the data in a table. The DML statements access and manipulate data in existing schema objects. The most important DML statements in SQL are: SELECT INSERT, UPDATE< and DELETE.

THE SELECT STATEMENT:

- The SELECT statement is a used to retrieve the data from the tables.

Syntax	Example
SELECT * From Table_Name;	SELECT * FROM **STUDENT**

THE INSERT STATEMENT:

The INSERT statement is used to insert a record into a table. We can insert one record or multiple records using insert statement. It Add new row(s) to the databased table.

Syntax	INSERT INTO table [(column[,column]..)]
Example	INSERT INTO SUPPLIERS VALUES (1000,'TOSHIBA','961 99');;

THE UPDATE STATEMENT:

The **UPDATE** statement is used to update exiting records in a table. The UPDATE statement is used to update a single record or multiple records in a table. The update statement is used to change values that are already in a table.

Syntax	UPDATE <table-name>
Example	UPDATE **SUPPLIERS** SET **CONTACT_NUMBER=9845012345**

THE DELETE STATEMENT:

The DELETE statement is used to delete rows in a table it is possible to delete all rows in a table without deleting the table.

Syntax	DELETE FROM <table name> WHERE <condition>;
Example	DELETE FROM **SUPPLIERS** WHERE **SUPPLIER_NAME**=' DELL';

ORDER BY CLAUSE:

The ORDER BY clause can be used to sort the rows. If we use the ORDER BY clause, it must be the last clause of the SQL statement.

Syntax	[ORDER BY {column,expr,numeric_position}[ASC|DESC]]
Example	SELECT * FROM **BOOK** ORDER BY **PUBLISHER** ASC;

♥ **GROUP BY CLAUSE:** (2M-2015)(5M-2012)(7M-2013)

We can divide the table of information in smaller groups using the GROUP BY clayse. We can then use the group function to return summery information for each group.

Syntax	[GROUP BY expression]
Example	SELECT PUBLISHER, PRICE FROM **BOOK** ORDER BY **PUBLISHER**;

JOINTS:(3M-2011)

1. Joins used to combine information (columns) from Tow or more tables
2. The table being joined must have one common column that have the same data type and data width in the tables.

♥ **Different Types of Joins:** (7M-2013) (2M-2015) (6M-2016)

1) Cross Join:

Each and every column of the first table will combine with each and every column to the second table

2) Equi-Join:

A JOIN based on an exact match between two columns is called an Equi-Join. The comparison operator in the join condition is = (equals).

Syntax	Example
SELECT **[table .]** Column , **[table .] Column…** FROM **table , table2…** WHERE **[table .]** Column = **[table2 .]** Column	SELECT **E.EMP_ID,E.EMP_NAME,D.DEPT_NAME** FROM **EMPLOYEE E,DEPERTMENT D** WHERE **E.DERT_NO=D.DERT_NO;**

3) Non Equi – Join:

This JOIN is similar to Equi-join except that comparison operator in the join condition is not = (equals). We can use any operator apart from '=' operator in joining condition.

1. Not-equal (<>),
2. Less than (<),
3. Greater than (>),
4. Less than or equal to (<=),
5. Greater than or equal to (>=),

4) Outer Join:

- The standard SQL JOIN operation combines information from two tables by forming pairs of related rows from the two tables.
- The outer join symbol (+) cannot be on both the sides

Types of OUTER join

1. LEFT OUTER JOIN
2. RIGHT OUTER JOIN
3. FULL OUTER JOIN

Syntax	Example
SELETE **[table .]** Column , **[table .]** Column, . . . FROM table **1** , table **2** . . . . WHERE **[table 1 .]** Column **(+)** = **[table 2.]** Column	SELETE E.EMP_ID , E.EMP_NAME, D .DEPT_NO FRPM EMPLOYEE E, DEPERTMENT D WHERE E.DEPT_NO (+) = D.DEPT_NO;

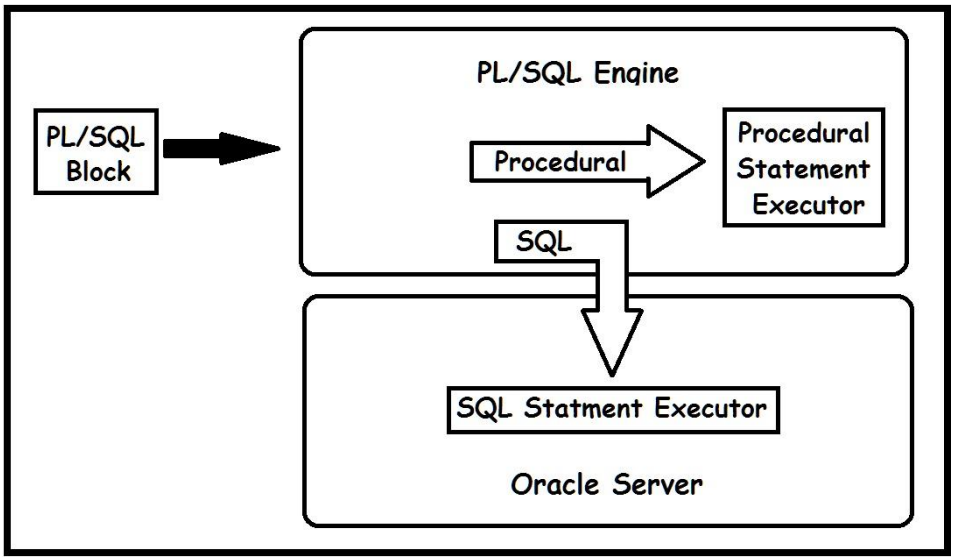
5) Self join:

The self join is used to join a table to itself, as if the table were tow tables

Syntax
SELECT **[alias1 .]** Column, **[alias2 .]** **Column**... FROM table **aliasl**, table **alias2** WHERE **[alias1 .]** Column = **[alias2 .]** Column

PL/SQL: (1M-2011) (3M-2012)

- **PL/SQL** is Oracle's **Procedural Language** extension to SQL **(Structure Query Language)**
- PL/SQL Allows you to mix SQL statements with procedural statements like IF statement and looping statement



ADVANTAGES OF PL/SQL: (3M-2011) (2M-2015)

- ✓ **Block Structures:** PL SQL consists of blocks of code, which can be nested within each other. Each block forms a unit of a task or a logical module. PL/SQL Blocks can be stored in the database and reused.
- ✓ **Procedural Language Capabilities**: PL SQL consists of procedural language constructs such as conditional statements (if else statements) and loop statements (FOR loops).
- ✓ **Batter Performance**: PL SQL engine processes multiple SQL statements simultaneously as a single block.
- ✓ **Error Handling**: PL/SQL handles errors or exceptions effectively during the execution of a PL?SQL program. Once an exception is caught, specific actions can be taken depending upon the exception or it can be displayed to the user with a message.

EXCEPTIONS HANDLING: (2M-2016)

Types of Exceptions (2M-2016)

1. **System defined Exceptions**
   - ➢ Named system exceptions
   - ➢ Unnamed system exceptions
2. **User-defined Exceptions**
   - ➢ Named programmer-defined exceptions.
   - ➢ Unnamed programmer-defined exceptions.

♥ DATABASE TRIGGERS: (3M-2011)(7M-2013)(3M-2014)(5M-2016)

A database triggers is stored PL/SQL program unit associated with a specific database table (or) view. The code in the trigger defines the action the database needs to perform whenever some database manipulation (INSERT,UPDATE,DELETE) takes place.

A database trigger has three parts:

- **A triggering event:** A triggering event can be an insert, update (or) delete statement (or) an instance shutdown (or) start up etc.
- **A trigger constraint: (Optional)**: A trigger constraint specifies a Boolean executed that must be true for the trigger to fire
- **Trigger action:** the trigger action is a procedure that contains the code to be executed when the trigger fries.

Trigger concepts:

Triggers are PL/SQL programs that automatically fire when:

- A DML statement such as INSERT,UPDATE (or) DELETE statement occurs on a table.
- A system statement, such as SHUTDOWN,START UO (or) SERVERERROR occurs.
- A User event, such as LOGOFF (or) LOGON occurs
- A DDL statement, such as create, drop (or) alter occurs.

Trigger Restrictions:

- Trigger cannot be fired for a SELECT statement.
- Trigger can never be executed explicitly like procedures (or) functions
- Triggers can call database procedures, functions and packages

Types of triggers: (1M-2011)

1. **Row triggers and statement triggers**
   A Row trigger fires once for each row affected. It users FOR EACH ROW clause.
2. **Before and After Triggers**
   - **BEFORE** triggers the trigger action here is run before the trigger statement.
   - **AFTER** triggers the trigger action here is run after the trigger statement.

♥ CURSORS: (1M-2012)(1M-2014) (5M-2015)

- Cursor represents a structure in memory and is different from cursor variable. When you declare a cursor, it gets a pointer variable
- When the cursor is opened, the memory is allocated for cursor is created.
- When the cursor is closed the memory allocated for the cursor is released.

Types of Cursors

There are two types of cursors implicit cursor and explicit cursors.

1. An IMPLICIT cursor is automatically declared by Oracle every time an SQL statement is executed.
2. An EXPLICIT cursor is declared by the programmer within the PL/SQL code block.

■■■■■■■■■■■■■■■■■■■■■■■■■■■(Chapter 9)■■■■■■■■■■■■■■■■■■■■■■■■■■■■

■■■(Transaction Processing Concepts & Concurrency Control Techniques) ■■■

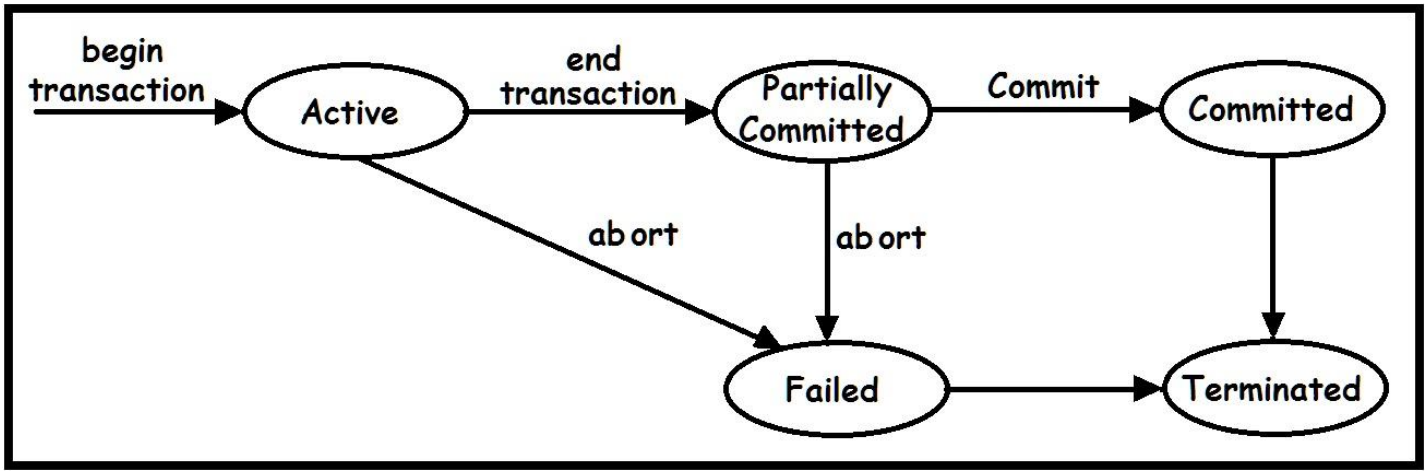
♥ TRANSACTION: (3M-2012)(1M-2014)(5M-2015)(2M-2016)

A transaction is a unit of a program execution that accesses and possibly modifies various data objects (tuples, relations).

Transaction properties

- **Atomicity:** A transaction is an atomic unit of processing and it either has to be performed in it's entirely (or) not at all.
- **Consistency:** A successful execution of a transaction must take a consistent database
- **Isolation:** A transaction must not make its modifications visible to other transactions until it is committed.
- **Durability:** Once a transaction has committed its changes, these changes must never get lost.

TRANSACRION STATES: (3m-2013)(6m-2016)



- **Active state** ; the transaction stays in this state while it is executing
- **Partially committed State**, after the final statement has been executed
- **Failed State**, After the discovery that normal executive can no longer proceed
- **Committed State**, after successful completion
- **Terminated** if has either **committed (or) aborted.**

LOCKING TECHNIQUES FOR CONCURRENCY CONTROL: (1M-2013)

A **lock** is a variable associated with a data item that describes the status of the item with respect to possible operations that can be applied to it.

TYPES OF LOCKS: (1M-2012) (3M-2014) (5M-2015)

1. Binary Lock

A binary lock can have two states (or) values **locked and unlocked** (or1 and 0, for simplicity). A distinct lock is associated with each data item X. If the value of the lock on X is 1, item X cannot be accessed by database operation that requests the item.

2. Shared / exclusive (or) read/write locks

In binary locking system, almost one transaction can hold a lock on a given item. We should allow several transactions to access the same item X, if they all access X for reading purposes only.

There are three locking operations:

1. Read _ lock (X)
2. Write_lock(X)
3. Unlock(X)

There are three possible state read locked, write locked (or) unlocked.

1. A **read locked** item is also called shared-locked because outer transaction are allowed to read the item.
2. **Write locked** item is called exclusive-locked because a single transaction exclusively holds the lock on the item.

Timestamp: (2M:2015) (4M:2016)

- The timestamp values are assigned in the order in which the transactions are submitted to the system.
- The timestamp can be generated by using a **counter** that is inctemented each time its value is assigned to a transaction
- It is important that the timestamp concurrency protocols are deadlock-free because there is no lock.

The Timestamp Ordering Algorithm (TO)

- It orders the transactions according to their timestamps.
- It guarantees serializability of schedules.
- In this algorithm, the schedule is equivalent to the particular serial order corresponding to the order of the transaction timestamps.