Database Management System(RCS-501)

UNIT - 1

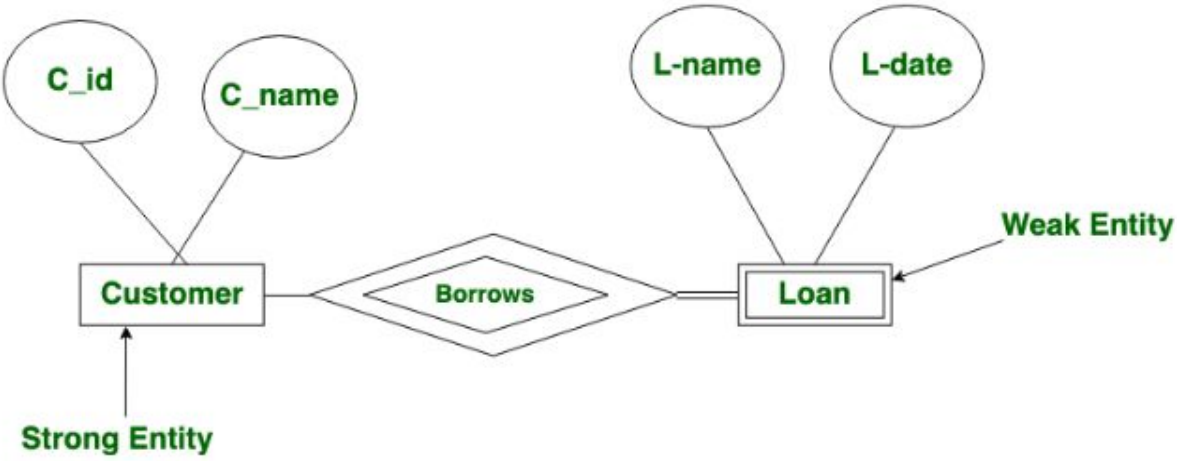
Section-A

1. **Define weak entity set.**

Weak Entity Set

The entity sets which do not have sufficient attributes to form a primary key are known as **weak entity sets** and the entity sets which have a primary key are known as strong entity sets.

An entity type should have a key attribute which uniquely identifies each entity in the entity set, but there exists some entity type for which key attribute can't be defined. These are called Weak Entity types.



2. **Define participation constraints.**

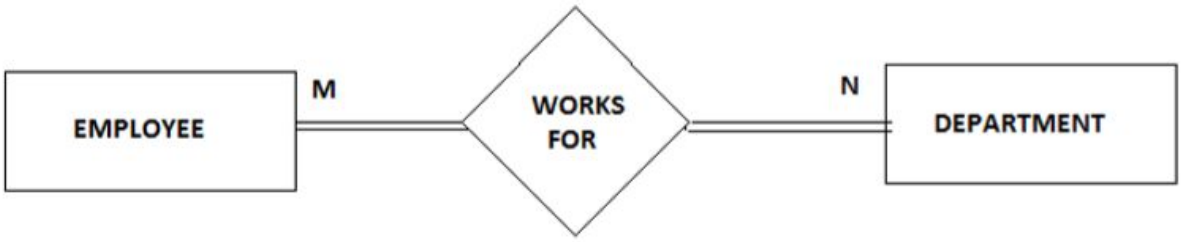
In a Relationship, Participation constraint specifies the existence of an entity when it is related to another entity in a relationship type. It is also called minimum cardinality constraint.

This constraint specifies the number of instances of an entity that can participate in a relationship type.

There are two types of Participation constraint −

Total Participation

Each entity in the entity set is involved in at least one relationship in a relationship set i.e. the number of relationships in every entity is greater than 0.



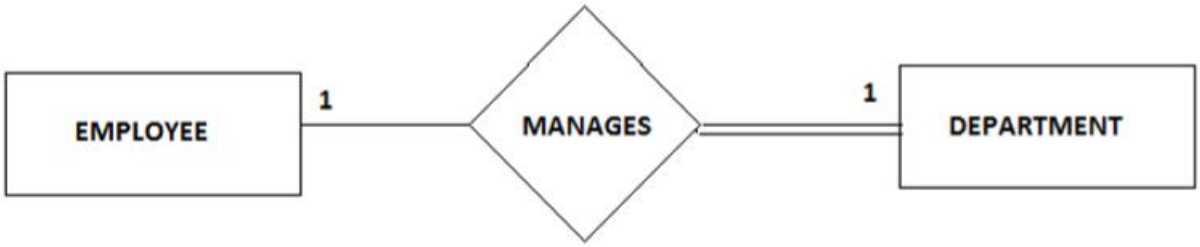
Consider two entities Employee and Department related via Works_For relationship. Now, every Employee works in at least one department therefore an Employee entity exists if it has at least one Works_For relationship with a Department entity. Thus the participation of an Employee in Works_For is a total relationship.

Total Participation is represented by a double line in the ER diagram.

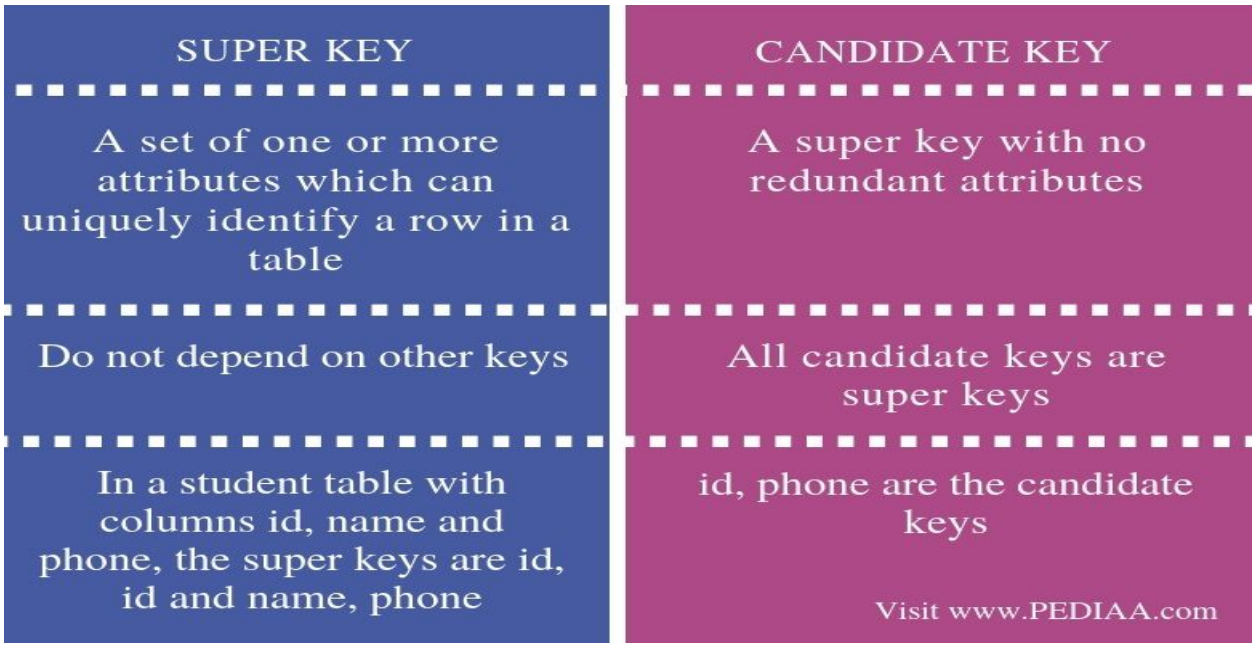
Partial Participation

Each entity in the entity set may or may not occur in at least one relationship in a relationship set.

For example: Consider two entities Employee and Department and they are related to each other via Managed relationship. An Employee must manage a Department, he or she could be the head of the department. But not every Employee in the company manages the department. So, participation of employees in the Managed relationship type is partial i.e. only a particular set of Employees will manage the Department but not all.



3. **Differentiate between candidate key and super key**



Sr. No.	Key	Super Key	Candidate key
1	Definition	Super Key is used to identify all the records in a relation.	Candidate key is a subset of Super Key.
2	Use	All super keys can't be candidate keys.	All candidate keys are super keys.
3	Selection	Super keys are combined together to create a candidate key.	Candidate keys are combined together to create a primary key.
4	Count Wise	Super keys are more than Candidate keys.	Candidate keys are less than Super Keys.

4. **Let E1 and E2 be two entities in an E/R diagram with simple single-valued attributes. R1 and R2 are two relationships between E1 and E2, where R1 is one-to-many and R2 is many-to-many. R1 and R2 do not have any attributes of their own. What is the minimum number of tables required to represent this situation in the relational model?**

(A) 2 **(B) 3** **(C) 4** **(D) 5**

Answer: (B)

Explanation: The answer is B, i.e minimum 3 tables.

Strong entities E1 and E2 are represented as separate tables.

In addition to that many-to-many relationships(R2) must be converted as separate table by having primary keys of E1 and E2 as foreign keys.

One-to-many relationship (R1) must be transferred to 'many' side table(i.e. E2) by having primary key of one side(E1) as foreign key(this way we need not to make a separate table for R1).

Let relation schema be E1(a1,a2) and E2(b1,b2).

5. **Given the basic ER and relational models, which of the following is INCORRECT?**

(A) An attribute of an entity can have more than one value

(B) An attribute of an entity can be composite

(C) In a row of a relational table, an attribute can have more than one value

(D) In a row of a relational table, an attribute can have exactly one value or a NULL value

Answer: (C)

Explanation: The term 'entity' belongs to ER model and the term 'relational table' belongs to relational model.

A and B both are true. ER model supports both multivalued and composite attributes See this for more details.

(C) is false and (D) is true. In Relation model, an entry in relational table can can have exactly one value or a NULL.

6. **Define DDL and DML terms.**

DDL is Data Definition Language which is used to define data structures. For example: create table, alter table are instructions in SQL.

DML is Data Manipulation Language which is used to manipulate data itself. For example: insert, update, delete are instructions in SQL.

DDL	DML
It stands for Data Definition Language.	It stands for Data Manipulation Language.
It is used to create database schema and can be used to define some constraints as well.	It is used to add, retrieve or update the data.
It basically defines the column (Attributes) of the table.	It add or update the row of the table. These rows are called as tuple.
It doesn't have any further classification.	It is further classified into Procedural and Non-Procedural DML.
Basic command present in DDL are CREATE, DROP, RENAME, ALTER etc.	BASIC command present in DML are UPDATE, INSERT, MERGE etc.
DDL does not use WHERE clause in its statement.	While DML uses WHERE clause in its statement.

7. **Which of the following gives a logical structure of the database graphically?**
   a) Entity-relationship diagram
   b) Entity diagram
   c) Database diagram
   d) Architectural representation

   Answer: a
   Explanation: E-R diagrams are simple and clear—qualities that may well account in large part for the widespread use of the E-R model.

8. **Consider a directed line(->) from the relationship set advisor to both entity sets instructor and student. This indicates _____ cardinality**

   a) One to many

   b) One to one

c) Many to many

d) Many to one

Answer: b
Explanation: This indicates that an instructor may advise at most one student, and a student may have at most one advisor.

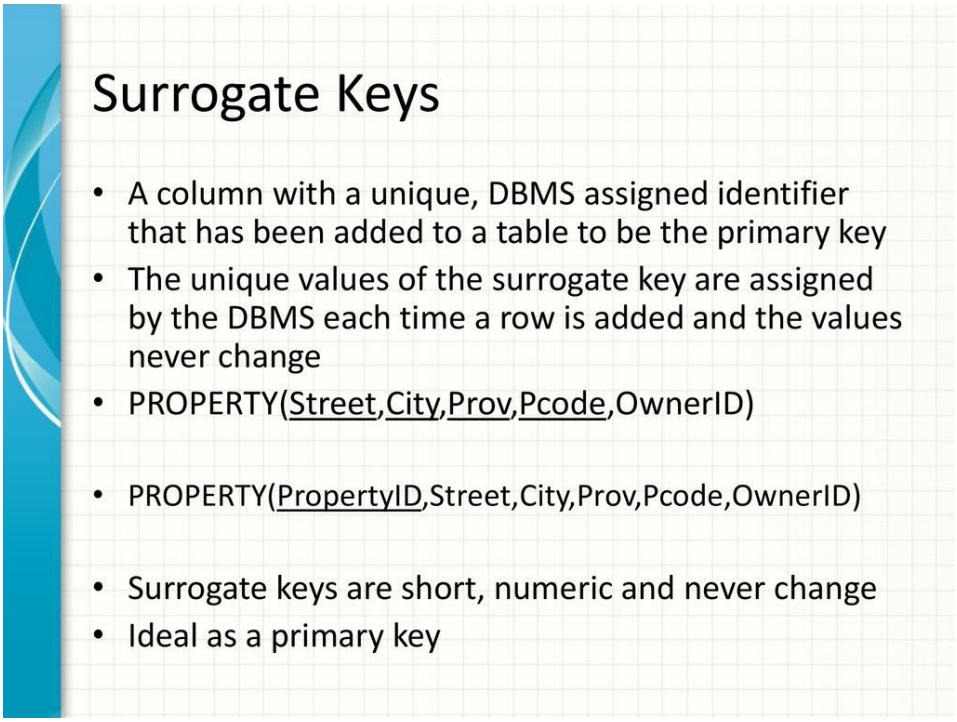
9. **Data model:** a collection of concepts that can be used to describe the structure of a database to achieve the data abstraction.

Categories of Data Models:-

- High Level or Conceptual Data Model
- Representational (or implementation) data models
- Low Level or Physical Data Model

10. Surrogate key

A surrogate key is a special key which has no meaning or purpose other than to uniquely identify each record. Surrogate keys are often a combination of letters and numbers. A new column is created to be the primary key.



(For more details about keys in database -> click here)

11.Write Symbol for following

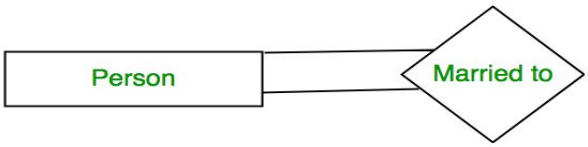
(a)Relationship (b) Derived Attribute (c) Identifying Relationship (d) Total Participation

a) diamond
b) Dotted - oval
c) Double diamond (What is identifying relationship? -> Answer)
d) Double line

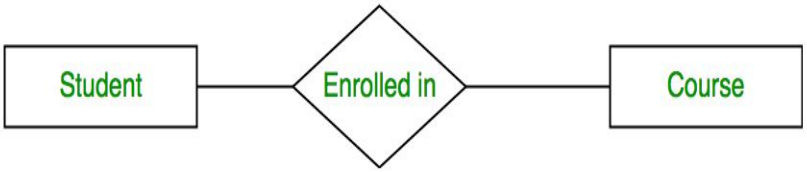
12. Degree of relationship.

The number of different entities **participating in a relationship** is called the degree of a relationship set.

1. **Unary Relationship –**
   When there is **only ONE entity set participating in a relation**, the relationship is called a unary relationship. For example, one person is married to only one person.



2. **Binary Relationship –**
   When there are **TWO entities set participating in a relation**, the relationship is called a binary relationship.For example, a Student is enrolled in a Course.

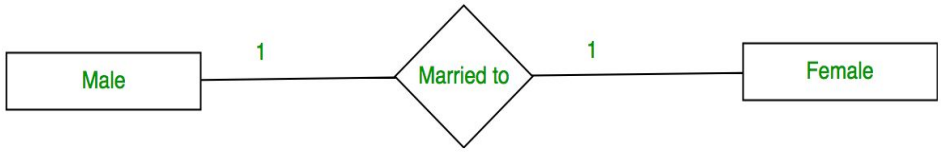


3. **n-ary Relationship –**
   When there are n entities set participating in a relation, the relationship is called as n-ary relationship

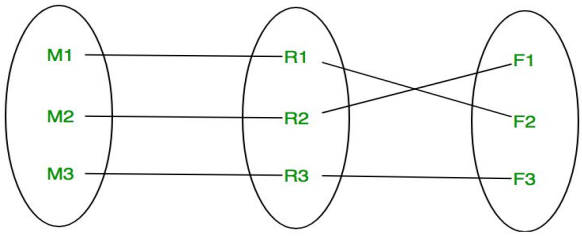
13. **Cardinality**

The **number of times an entity of an entity set participates in a relationship** set is known as cardinality. Cardinality can be of different types:

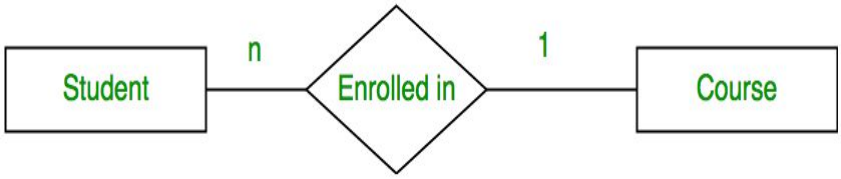
1. **One to one –** When each entity in each entity set can take part **only once in the relationship**, the cardinality is one to one. Let us assume that a male can marry to one female and a female can marry to one male. So the relationship will be one to one.



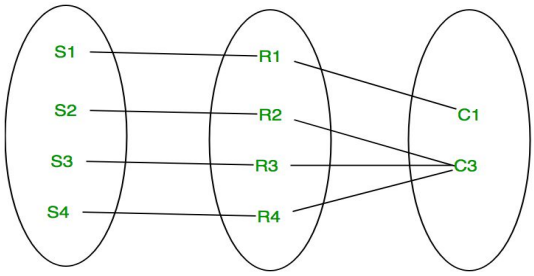
Using Sets, it can be represented as:



2. **Many to one –** When entities in one entity set **can take part only once in the relationship set and entities in another entity set can take part more than once in the relationship set,** cardinality is many to one. Let us assume that a student can take only one course but one course can be taken by many students. So the cardinality will be n to 1. It means that for one course there can be n students but for one student, there will be only one course.



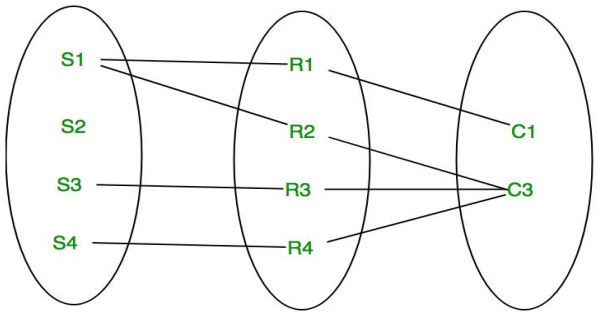
Using Sets, it can be represented as:



In this case, each student is taking only 1 course but 1 course has been taken by many students.

3. **Many to many –** When entities in all entity sets can **take part more than once in the relationship** cardinality is many to many. Let us assume that a student can take more than one course and one course can be taken by many students. So the relationship will be many to many.

Using sets, it can be represented as:



In this example, student S1 is enrolled in C1 and C3 and Course C3 is enrolled by S1, S3 and S4. So it is many to many relationships

14. What is an Entity

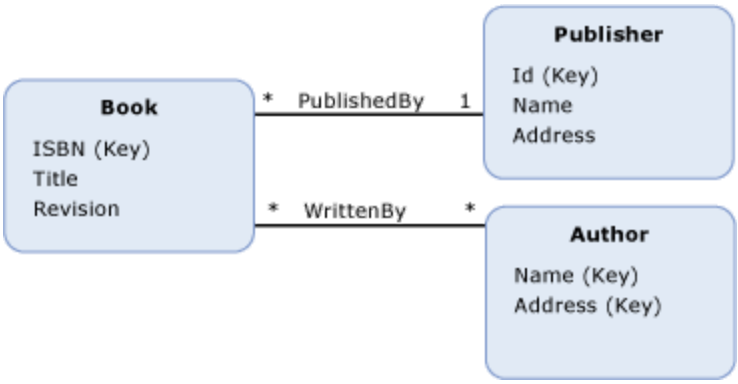
An **entity** can be a real-world object, either animate or inanimate, that can be easily identifiable. For example, in a school database, students, teachers, classes, and courses offered can be considered as **entities**. An **entity** set is a collection of similar types of **entities**.

15. What is an Entity Type

The *entity type* is the fundamental building block for describing the structure of data with the Entity Data Model (EDM). In a conceptual model, an entity type represents the structure of top-level concepts, such as customers or orders. An entity type is a template for entity type instances.
An Entity is an object of Entity Type and the set of all entities is called an entity set. e.g.; E1 is an entity having Entity Type Student and the set of all students is called Entity Set.

Example

The diagram below shows a conceptual model with three entity types: Book, Publisher, and Author:

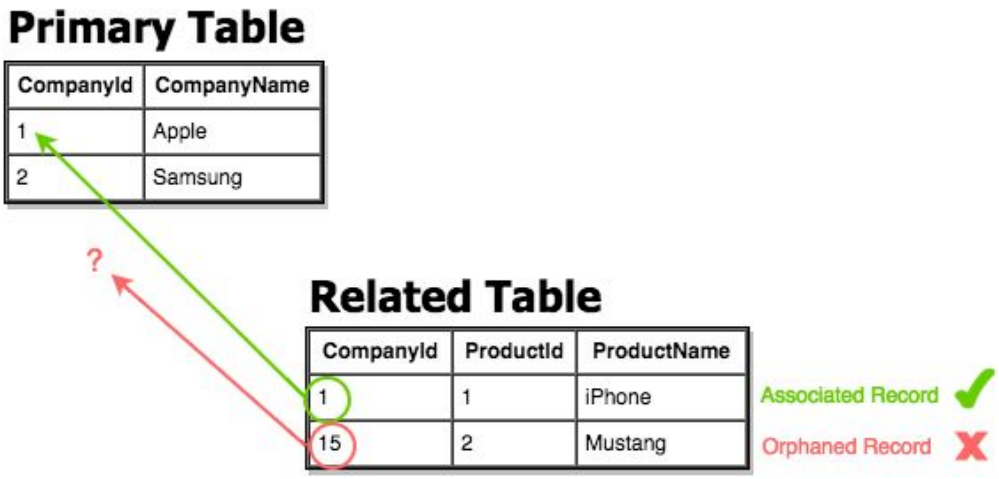


16. Referential integrity

Referential integrity refers to the accuracy and consistency of data within a relationship. In relationships, data is linked between two or more tables. This is achieved by having the foreign key (in the associated table) reference a primary key value (in the primary – or parent – table). Because of this, we need to ensure that data on both sides of the relationship remain intact.

So, referential integrity requires that, **whenever a foreign key value is used it must reference a valid, existing primary key in the parent table.**

For example, if we delete record number 15 in a primary table, we need to be sure that there's no foreign key in any related table with the value of 15. We should only be able to delete a primary key if there are no associated records. Otherwise, we would end up with an orphaned record.



(For more detail about referential integrity -> click here)

17. Differentiate between Logical Data Independence and Physical Data Independence

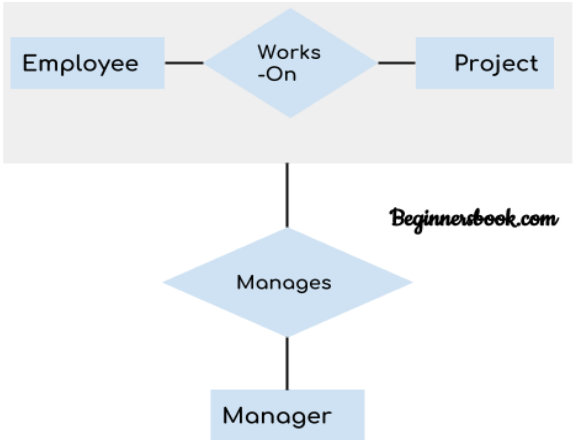
- **Logical data independence:** is the capacity to change the conceptual schema without having to change external schemas or application programs.
- **Physical data independence:** is the capacity to change the internal schema without having to change the conceptual schema. Hence, the external schemas need not be changed as well.

Logical Data Independence	Physical Data Independence
It is concerned with the structure of the data or changing the data definition.	It is concerned with storage of the data.
It is very difficult as the retrieving of data are heavily dependent on logical structure of data.	It is easy to retrieve.
Application program need not be changed if new fields are added or deleted from the database.	Physical database is concerned with the change of the storage device.
It is concerned with the conceptual schema.	It is concerned with the internal schema.

18. Aggregation

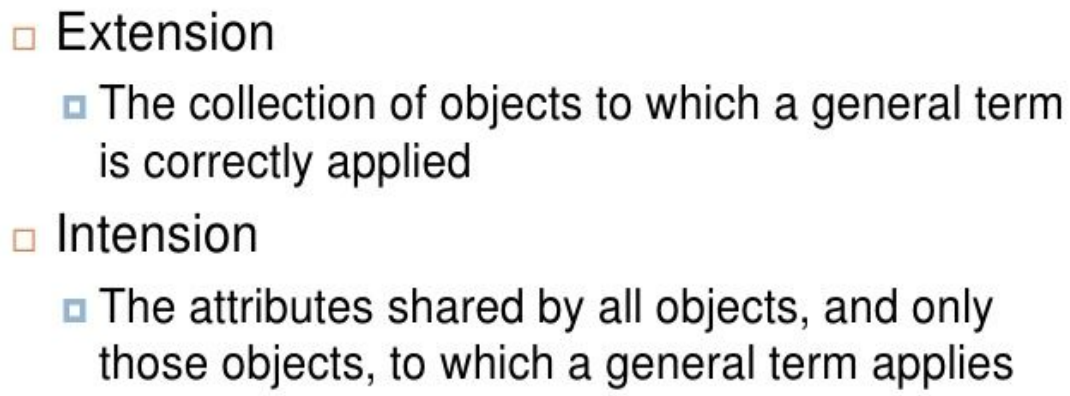
Aggregation is a process in which a single entity alone is not able to make sense in a relationship so the relationship of two entities acts as one entity. I know it sounds confusing but don't worry the example we will take, will clear all the doubts.

Aggregation Example



19. Differentiate between Intension and Extension?

- **Schema:** The description of a database is called the database schema, which is specified during database design and is not expected to change frequently
- **Database State:** The data in the database at a particular moment in time is called a database state or snapshot. It is also called the current set of occurrences or instances in the database
- The schema is sometimes called the **intension**, and a database state is called an **extension** of the schema.



20. What is a Query Evaluation Engine ?

ANS: Query evaluation engine is an important part of SQL(structured query language) because all the query evaluated in SQL with the help of query evaluation engine, it executes low-level instruction generated by the compiler and provides specific output.

==========

Section-B

1. **Discuss the main characteristics of the database approach and how it differs from traditional file systems.**

Main Characteristics of the database approach :

- Self-describing nature of a database system.

a fundamental characteristic of the database approach is that the database system contains not only the database structure and constrains. this definition is stored in the system catalog. A DBMS catalog stores the description of a particular database (e.g. data structures, types, and constraints) The description is called meta-data this allows the DBMS software to work with different database application.

- Insulation between programs and data.

In traditional file processing, the structures of data files is embedded in the access programs, so many change to the structure of a file may require changing all programs that access this file. By constraint, DBMS access programs do not require

such changes in most cases. the structure of data files is stored in the DBMS catalog separately from the access programs. we call this property program-data independence

program-data independence: allows changing data structures and storage organization without having to change the DBMS access programs.

- Data abstraction.

The characteristic that allows program-data independence and program-operation independence is called data abstraction. A data model is a type of data abstraction that is used to hide storage details and present the users with a conceptual view of the database.

- Support of multiple views of the data.

Each user may see a different view of the database, which describes only the data of interest to that user. A view may be a subset of the database or it may contain virtual data that is derived from the database files but is not explicitly stored.

- Sharing of data and multi-user transaction processing.

Allowing a set of concurrent users to retrieve from and to update the database. concurrency control within the DBMS guarantees that each transaction is correctly executed or aborted. Recovery subsystem ensures each completed transaction has its effect permanently recorded in the database. OLTP (online Transaction Processing) is a major part of database applications. this allows hundreds of concurrent transactions to execute per second.

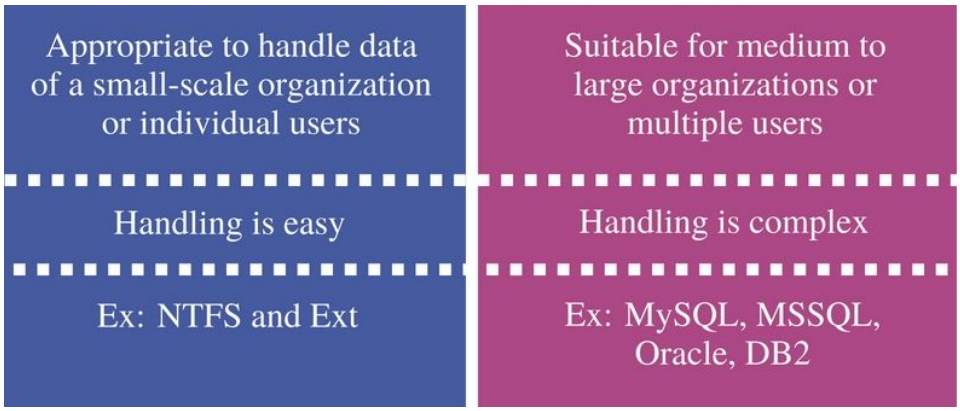
Database approach Differs from traditional file system:

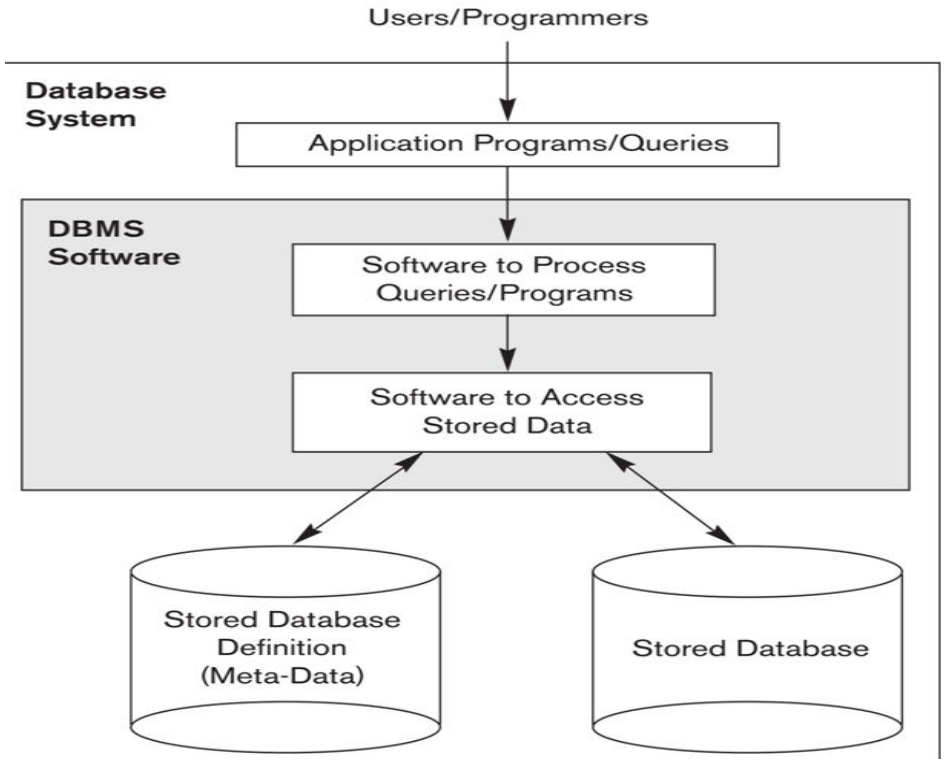
FILE SYSTEM

VERSUS

DBMS

FILE SYSTEM	DBMS
Software that manages the data files in a computer system	Software to create and manage databases
Helps to store a collection of raw data files into the hard disk	Helps to easily store, retrieve and manipulate data in a database
Tasks such as storing, retrieving and searching are done manually, so it is difficult to manage data	Operations such as updating, searching, selecting data is easier since it allows using SQL querying
Has data inconsistency	Provides higher data consistency using normalization
There is more redundant data	There is low data redundancy
Provides more security to data	Comparatively less data security
Backup and recovery process is not efficient because it is not possible to recover the lost data	Has a sophisticated backup and recovery
Appropriate to handle data of a small-scale organization or individual users	Suitable for medium to large organizations or multiple users
Handling is easy	Handling is complex
Ex: NTFS and Ext	Ex: MySQL, MSSQL, Oracle, DB2





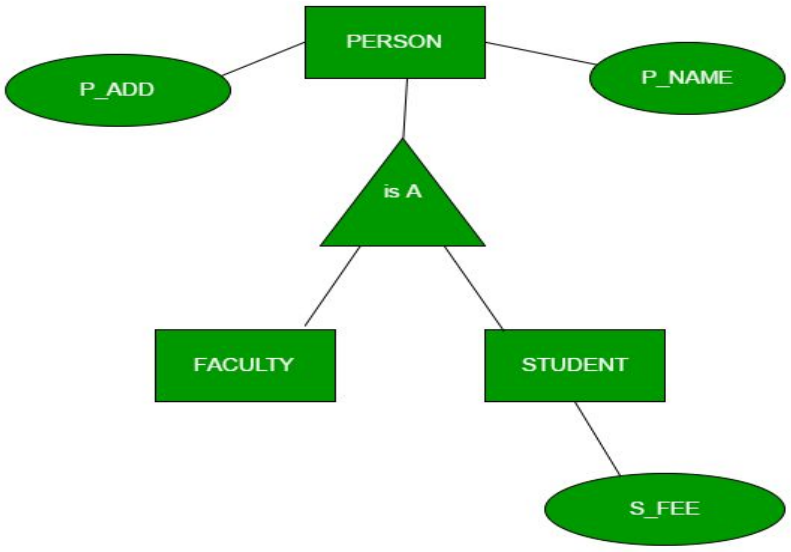
-

2. **Explain specialization and generalization with examples.**

Generalization –

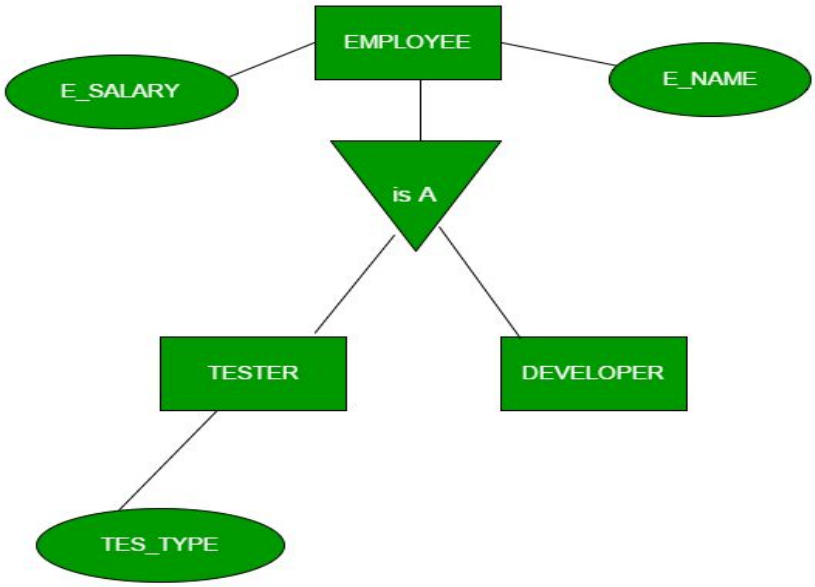
- Generalization is the process of extracting common properties from a set of entities and creating a generalized entity from it.
- It is a bottom-up approach in which two or more entities can be generalized to a higher level entity if they have some attributes in common.
- For Example, STUDENT and FACULTY can be generalized to a higher level entity called PERSON as shown in Figure 1. In this case, common attributes like P_NAME,

P_ADD become part of higher entities (PERSON) and specialized attributes like S_FEE become part of specialized entities (STUDENTS).



Specialization –

- In specialization, an entity is divided into sub-entities based on their characteristics.
- It is a top-down approach where a higher level entity is specialized into two or more lower level entities.
- For Example, EMPLOYEE entities in an Employee management system can be specialized into DEVELOPER, TESTER etc. as shown in Figure 2. In this case, common attributes like E_NAME, E_SAL etc. become part of higher entities (EMPLOYEE) and specialized attributes like TES_TYPE become part of specialized entities (TESTER).



3. What are the responsibilities of the DBA and the database designers?

The **database designer** is responsible for defining the detailed **database** design, including tables, indexes, views, constraints, triggers, stored procedures, and other **database**-specific constructs needed to store, retrieve, and delete persistent objects.

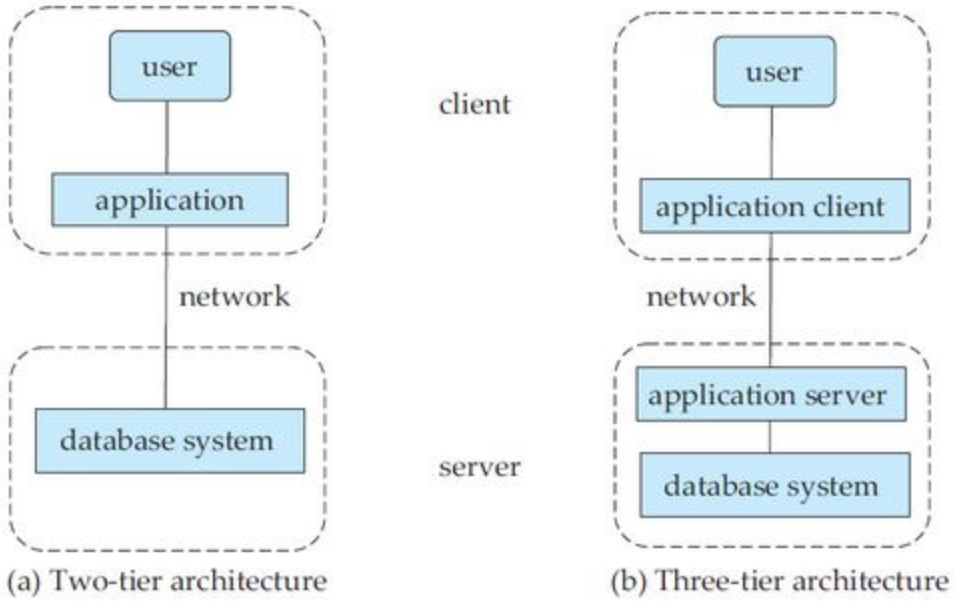
Responsibilities of DBA:

DBA stands for Data Base Administrator. The purpose of a database administrator is highly technical, who is responsible for managing the database used in the organization.

- The database administrator has the responsibility to build the physical design of the database.
- The database administrator deals with the technical responsibilities like,
  - o Defence enforcement
  - o Performance of the database
  - o Provide access to the database
  - o Acquire resources such hardware and software components
  - o Backup of the data from the database
  - o Recovery of the lost data from the database
  - o Monitoring and Coordinating the use of database
  - o Monitoring response time and security breaches.

4. **What is the difference between the two-tier and three-tier client/server architectures?**

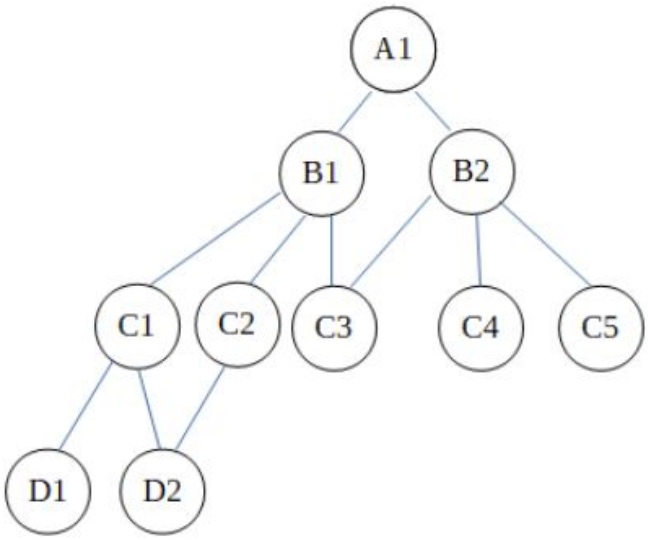
S.NO	TWO-TIER DATABASE ARCHITECTURE	THREE-TIER DATABASE ARCHITECTURE
1	It is a Client-Server Architecture.	It is a Web-based application.
2	In two-tier, the application logic is either buried inside the user interface on the client or within the database on the server (or both).	In three-tier, the application logic or process resides in the middle-tier, it is separated from the data and the user interface.
3	Two-tier architecture consists of two layers : Client Tier and Database (Data Tier).	Three-tier architecture consists of three layers : Client Layer, Business Layer and Data Layer.
4	It is easy to build and maintain.	It is complex to build and maintain.
5	Two-tier architecture runs slower.	Three-tier architecture runs faster.
6	It is less secured as client can communicate with database directly.	It is secured as client is not allowed to communicate with database directly.
7	It results in performance loss whenever the users increase rapidly.	It results in performance loss whenever the system is run on Internet but gives more performance than two-tier architecture.
8	Example – Contact Management System created using MS-Access or Railway Reservation System, etc.	Example – Designing registration form which contains text box, label, button or a large website on the Internet, etc.



5. Explain the Network data model ?

The network database model was created to solve the shortcomings of the hierarchical database model. In this type of model, a child can be linked to multiple parents, a feature that was not supported by the hierarchical data model. The parent nodes are known as owners and the child nodes are called members.

The network data model can be represented as −



Advantages of Network Model

The network model can support many to many relationships as seen in the diagram. D2 and C3 each have multiple masters. The masters for D2 are C1 and C2 while for C3 are B1 and B2. In this way, the network data model can handle many to many relationships where the hierarchical data model didn't.

Disadvantages of Network Model

There are some disadvantages in the network model even though it is an improvement over the hierarchical model. These are −

- The network model is much more complicated than the Hierarchical model. As such, it is difficult to handle and maintain.
- Although the Network model is more flexible than the Hierarchical model, it still has flexibility problems. Not all relations can be handled by assigning them in the form of owners and members.
- The structure of the Network Model is quite complicated and so the programmer has to understand it well in order to implement or modify it.

6. Explain all the types of users and administrator.?

Different types of Database Users

Database users are categorized based up on their interaction with the database.

These are seven types of database users in DBMS.

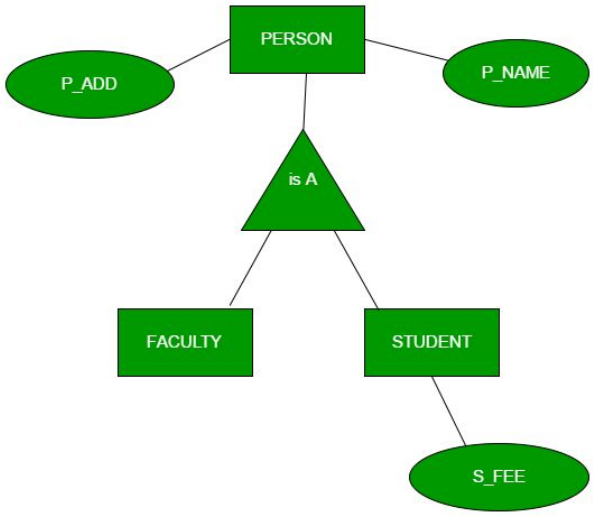
- **Database Administrator (DBA) :**
  Database Administrator (DBA) is a person/team who defines the schema and also controls the 3 levels of the database.
  The DBA will then create a new account id and password for the user if he/she needs to access the database.
  DBA is also responsible for providing security to the database and he allows only the authorized users to access/modify the database.
  - DBA also monitors the recovery and backup and provides technical support.
  - The DBA has a DBA account in the DBMS which is called a system or superuser account.
  - DBA repairs damage caused due to hardware and/or software failures.
- **Naive / Parametric End Users :**
  Parametric End Users are the unsophisticated who don't have any DBMS knowledge but they frequently use the database applications in their daily life to get the desired results. For examples, Railway's ticket booking users are naive users. Clerks in any bank are naive users because they don't have any DBMS knowledge but they still use the database and perform their given task.
- **System Analyst :**
  System Analyst is a user who analyzes the requirements of parametric end users. They check whether all the requirements of end users are satisfied.

- **Sophisticated Users :**
  Sophisticated users can be engineers, scientists, business analysts, who are familiar with the database. They can develop their own database applications according to their requirement. They don't write the program code but they interact with the database by writing SQL queries directly through the query processor.
- **Database Designers :**
  Database Designers are the users who design the structure of the database which includes tables, indexes, views, constraints, triggers, stored procedures. He/she controls what data must be stored and how the data items are related.
- **Application Program :**
  Application Program are the back end programmers who write the code for the application programs.They are the computer professionals. These programs could be written in Programming languages such as Visual Basic, Developer, C, FORTRAN, COBOL etc.
- **Casual Users / Temporary Users :**
  Casual Users are the users who occasionally use/access the database but each time when they access the database they require the new information, for example, Middle or higher level manager.

7. Explain the generalization and aggregation with example. ?

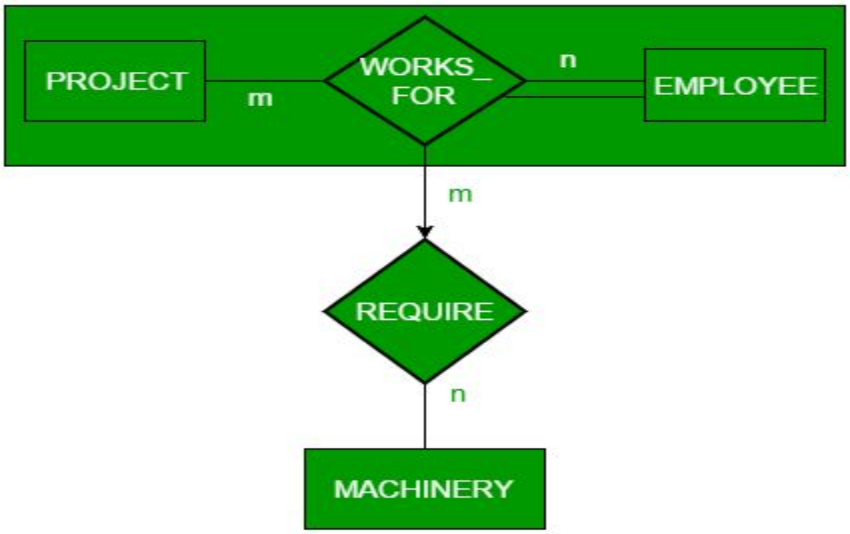
Generalization –

- Generalization is the process of extracting common properties from a set of entities and creating a generalized entity from it.
- It is a bottom-up approach in which two or more entities can be generalized to a higher level entity if they have some attributes in common.
- For Example, STUDENT and FACULTY can be generalized to a higher level entity called PERSON as shown in Figure 1. In this case, common attributes like P_NAME, P_ADD become part of higher entities (PERSON) and specialized attributes like S_FEE become part of specialized entities (STUDENT).



Aggregation –

- An ER diagram is not capable of representing the relationship between an entity and a relationship which may be required in some scenarios.
- In those cases, a relationship with its corresponding entities is aggregated into a higher level entity.
- For Example, Employees working for a project may require some machinery. So, REQUIRE relationship is needed between relationship WORKS_FOR and entity MACHINERY. Using aggregation, WORKS_FOR relationship with its entities EMPLOYEE and PROJECT is aggregated into a single entity and relationship REQUIRE is created between the aggregated entity and MACHINERY.



Explain various types of database languages ?

Data Definition Language

The language is used to create databases, tables, alter them, etc. With this, you can also rename the database, or drop them. It specifies the database schema.

The DDL statements include −

- **CREATE**: Create new database, table, etc.
- **ALTER**: Alter existing database, table, etc.
- **DROP**: Drop the database
- **RENAME**: Set a new name for the table.

Data Manipulation Language

The language used to manipulate the database like inserting data, updating table, retrieving record from a table, etc. is known as Data Manipulation Language −

- **SELECT**: Retrieve data from the database

- **INSERT**: Insert data
- **UPDATE**: Update data
- **DELETE**: Delete all records

Data Control Language

Grant privilege to a user using the GRANT statement. In the same way, revoke the privilege using the REVOKE statement. Both of these statements come under the Data Control Language (DCL). −

- **GRANT**: Give privilege to access the database.
- **REVOKE:** Take back the privilege to access the database.

Transaction Control Language

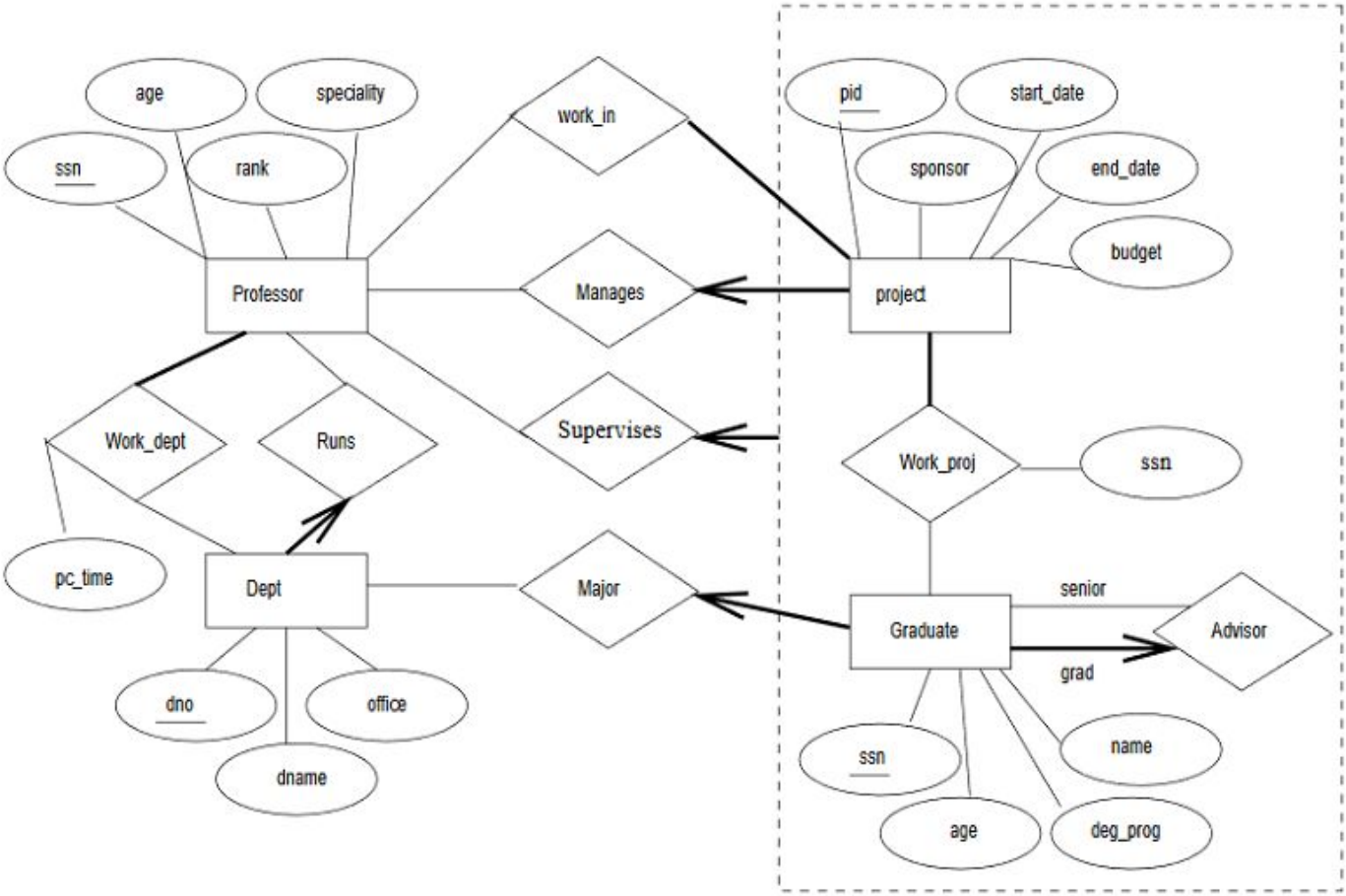
Manage transactions in the Database using the Transaction Control Language −

- **COMMIT**: Save the work.
- **SAVEPOINT**: Set a point in transaction to rollback later
- **ROLLBACK**: Restores since last commit

==========

Section -C

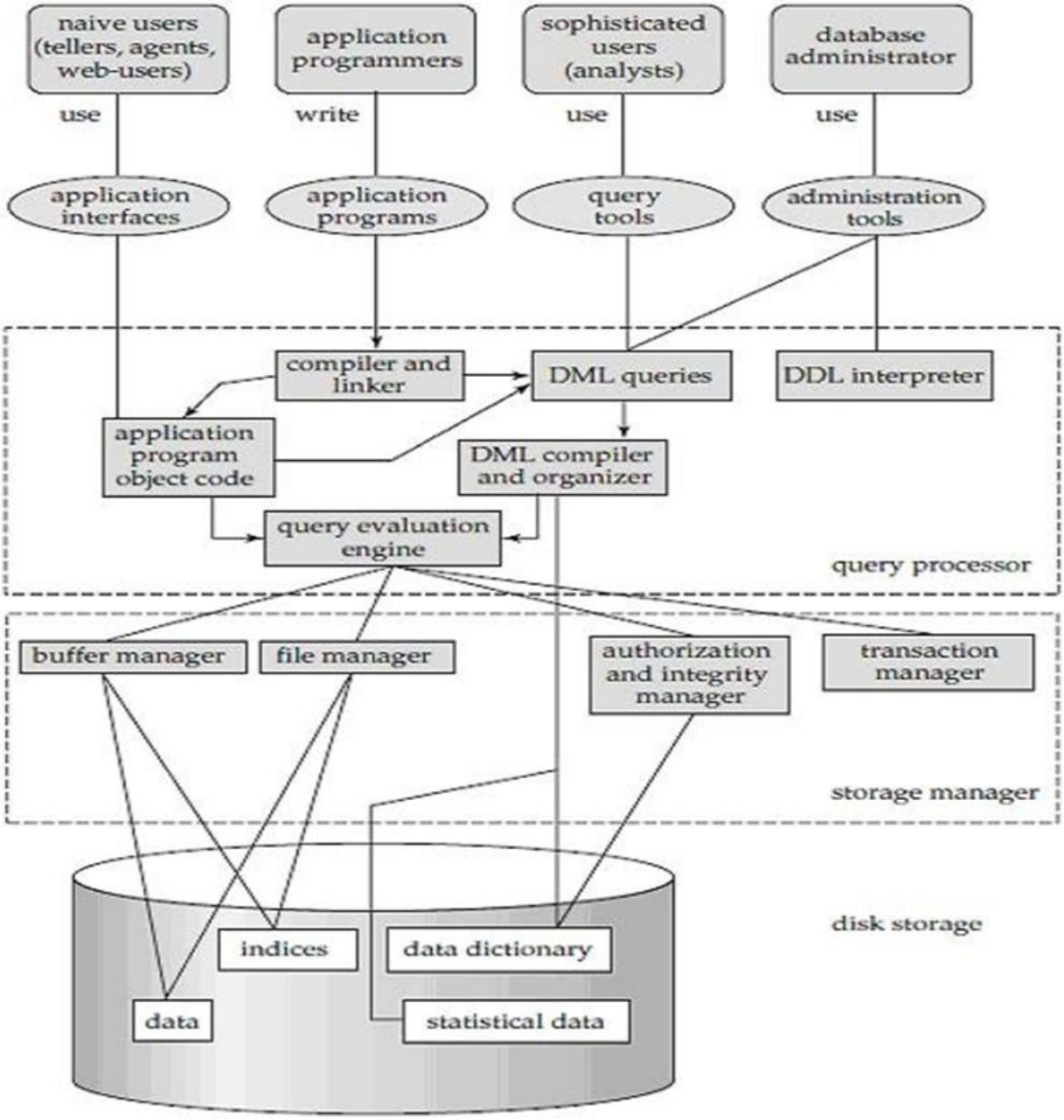
1. **Consider the following information about a university database: Professors have an SSN, a name, an age, a rank, and a research specialty. Projects have a project number, a sponsor name (e.g., NSF), a starting date, an ending date, and a budget. Graduate students have an SSN, a name, an age, and a degree program (e.g., M.S. or Ph.D.). Each project is managed by one professor (known as the project's principal investigator). Each project is worked on by one or more professors (known as the project's co-investigators) Professors can manage and/or work on multiple projects. Each project is worked on by one or more graduate students (known as the project's research assistants). When graduate students work on a project, a professor must supervise their work on the project. Graduate students can work on multiple projects, in which case they will have a (potentially different) supervisor for each one. Departments have a department number, a department name, and a main office. Departments have a professor (known as the chairman) who runs the department. Professors work in one or more departments, and for each department that they work in, a time percentage is associated with their job. Graduate students have one major department in which they are working on their degree. Each graduate student has another, more senior graduate student (known as a student advisor) who advises him or her on what courses to take.Design and draw an ER diagram that captures the information about the university. Use only the basic ER model here; that is, entities, relationships, and attributes. Be sure to indicate any key and participation constraints.**


age
speciality
ssn
rank
work_in
pid
start_date
sponsor
end_date
budget
Professor
Manages
project
Work_dept
Runs
Supervises
Work_proj
ssn
pc_time
Dept
Major
senior
Graduate
Advisor
grad
dno
office
dname
ssn
name
age
deg_prog

2. **Draw an E-R diagram for a hospital with a set of patients and a set of medical doctors; with each patient a log of the various conducted tests is also associated.**

3. Explain database system architecture. ?

Database architecture uses programming languages to design a particular type of software for businesses or organizations. Database architecture focuses on the design, development, implementation and maintenance of computer programs that store and organize information for businesses, agencies and institutions. A database architect develops and implements software to meet the needs of users.



The design of a DBMS depends on its architecture. It can be centralized or decentralized or hierarchical. The architecture of a DBMS can be seen as either single tier or multi-tier. The tiers are classified as follows :

1. **1-tier architecture**
2. **2-tier architecture**
3. **3-tier architecture**
4. **n-tier architecture**

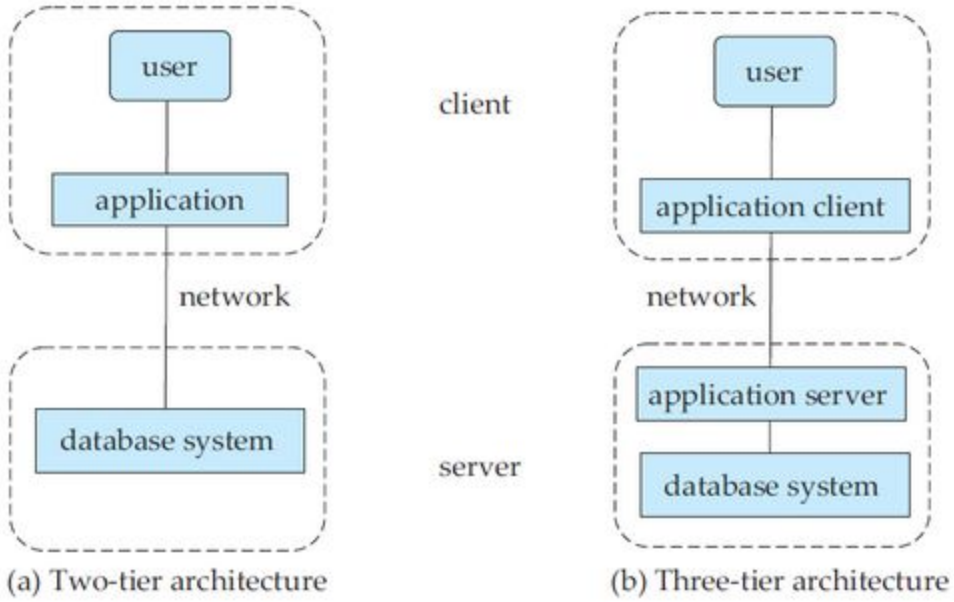
2-tier architecture:

The two-tier is based on Client Server architecture. The two-tier architecture is like client server application. The direct communication takes place between client and server. There is no intermediate between client and server.

3-tier architecture:

A 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. It is the most widely used architecture to design a DBMS.

This architecture has different usages with different applications. It can be used in web applications and distributed applications. The strength in particular is when using this architecture over distributed systems.



(a) Two-tier architecture (b) Three-tier architecture

4. A University registrar's office maintains data about the following entities: - Courses including course_number, title, credits, syllabus and prerequisites. Course Offerings including course_number, year, semester, section_number, instructor, timing and classroom. Students including student_id, name, and program. Instructor including identification_number, name, department and title. Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled. Construct an E-R diagram for the registrar's office.