DataBase Management System

Question Bank

Module 1:- Introduction Database Concepts

1. Explain Characteristics of databases.

1. Relation-based tables − DBMS allows entities and among them to form tables. A user can understand the architecture of a database just by looking at the table names.

2. Less redundancy − DBMS follows the rules of normalization, which splits a relation when any of its attributes is having redundancy in values.

3. Consistency − Consistency is a state where everyrelation in a database remains consistent. A DBMS can provide greater consistency as compared to earlier forms of data storing applications.

4. Query Language − DBMS is equipped with query language, whichmakes it more efficient to retrieve and manipulate data. A user can apply as many and as different filtering options as required to retrieve a set of data.

5. ACID Properties − DBMS follows the concepts of Atomicity, Consistency, Isolation, and Durability .These concepts are applied on transactions, which manipulate data in a database. ACID properties help the database stay healthy in multi-transactional environments and in case of failure.

6. Multiuser and Concurrent Access − DBMS supports multi-user environment and allows them to access and manipulate data in parallel.

7. Security − DBMS offers methods to impose constraints while entering data into the database and retrieving the same at a later stage. DBMS offers many different levels of security features, which enables multiple users to have different views with different features. For example, a user in the Sales department cannot see the data that belongs to the Purchase department. Additionally, it can also be managed how much data of the Sales department should be displayed to the user.

2. Terminology and definitions related to databases . Example for any concepts.

3.Discuss drawbacks of file system.

1. Limited data integrity: File systems lack built-in mechanisms to ensure data consistency and prevent corruption, increasing the risk of errors and inconsistencies.

2. Limited query capabilities: File systems do not offer advanced querying functionalities like databases, making it challenging to efficiently retrieve specific data or perform complex searches.

3. Data redundancy: File systems often result in redundant storage of data across multiple files, leading to inefficiencies in storage usage and difficulties in maintaining data consistency.

4. Concurrency issues: File systems may encounter conflicts when multiple users or processes attempt to access or modify the same files simultaneously, risking data corruption or loss.

5. Limited scalability: As the volume of data increases, file systems may struggle to handle the growing workload efficiently, leading to performance degradation and administrative challenges.

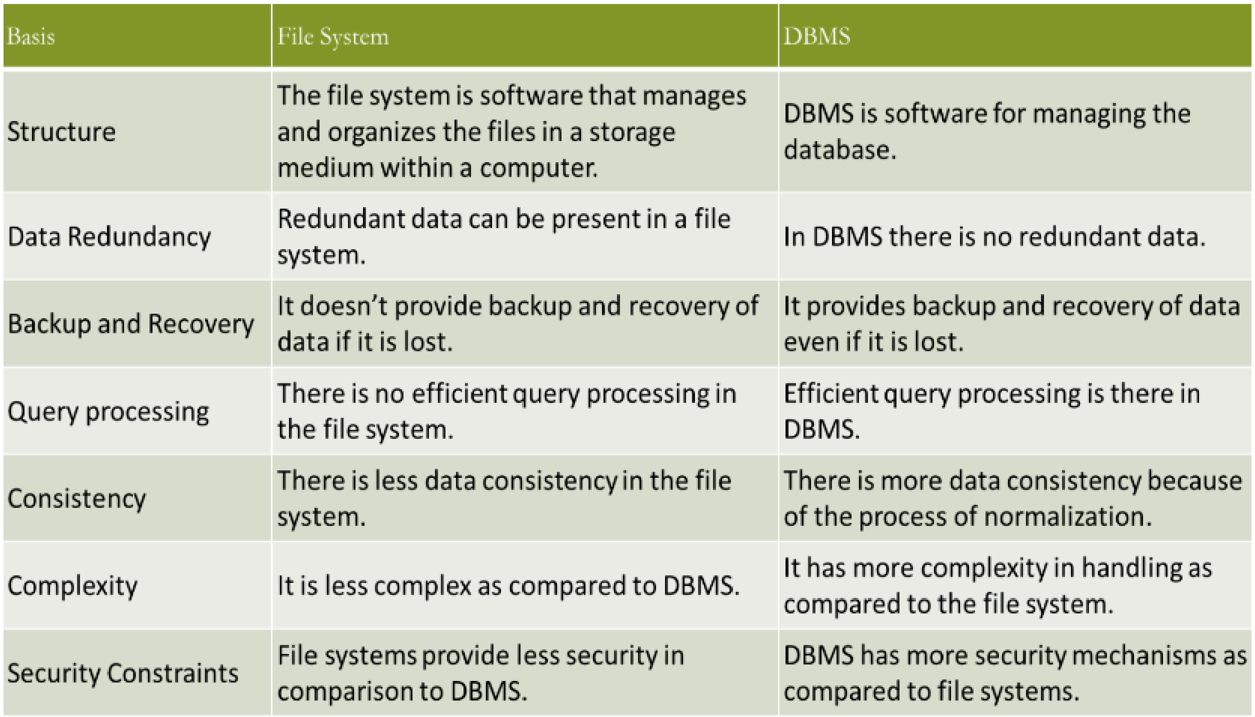
6. Limited security: File systems typically offer basic security features, but they may lack advanced security mechanisms like encryption and comprehensive access controls, leaving data vulnerable to unauthorized access or breaches.

7. Difficulty in data management: Managing data in file systems can be cumbersome, especially when dealing with complex relationships or dependencies between files, leading to challenges in maintaining data coherence and consistency.

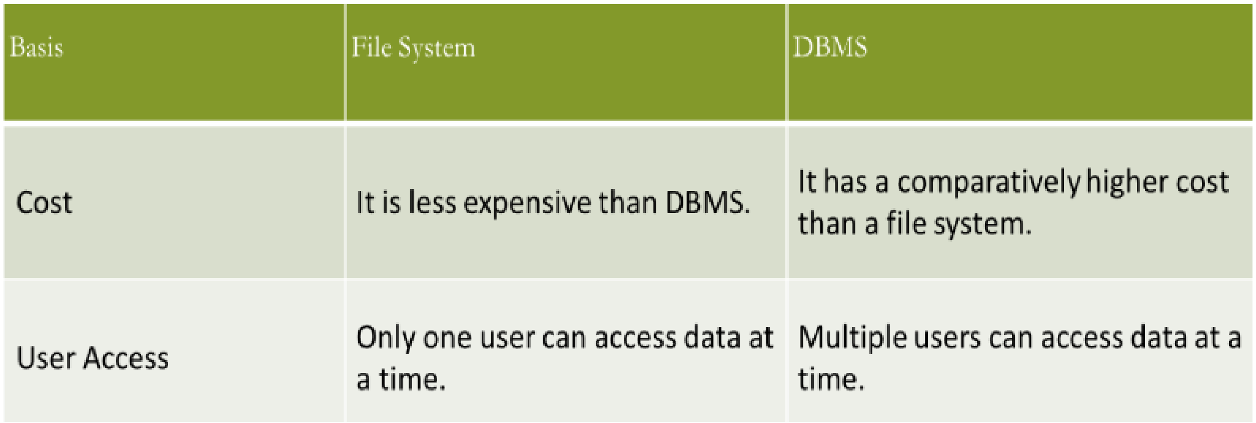
8. Limited data manipulation features: File systems lack advanced data manipulation features such as transactions and stored procedures, limiting automation and enforcing business logic at the database level.

9. Backup and recovery challenges: File systems may lack robust backup and recovery mechanisms, making it challenging to ensure data availability and recoverability in the event of system failures or disasters.

4. Advantages of DBMS over file system



Basis	File System	DBMS
Structure	The file system is software that manages and organizes the files in a storage medium within a computer.	DBMS is software for managing the database.
Data Redundancy	Redundant data can be present in a file system.	In DBMS there is no redundant data.
Backup and Recovery	It doesn’t provide backup and recovery of data if it is lost.	It provides backup and recovery of data even if it is lost.
Query processing	There is no efficient query processing in the file system.	Efficient query processing is there in DBMS.
Consistency	There is less data consistency in the file system.	There is more data consistency because of the process of normalization.
Complexity	It is less complex as compared to DBMS.	It has more complexity in handling as compared to the file system.
Security Constraints	File systems provide less security in comparison to DBMS.	DBMS has more security mechanisms as compared to file systems.



Basis	File System	DBMS
Cost	It is less expensive than DBMS.	It has a comparatively higher cost than a file system.
User Access	Only one user can access data at a time.	Multiple users can access data at a time.

5. How database offers solution to the problem of file system ? OR Difference between DBMS and File system.

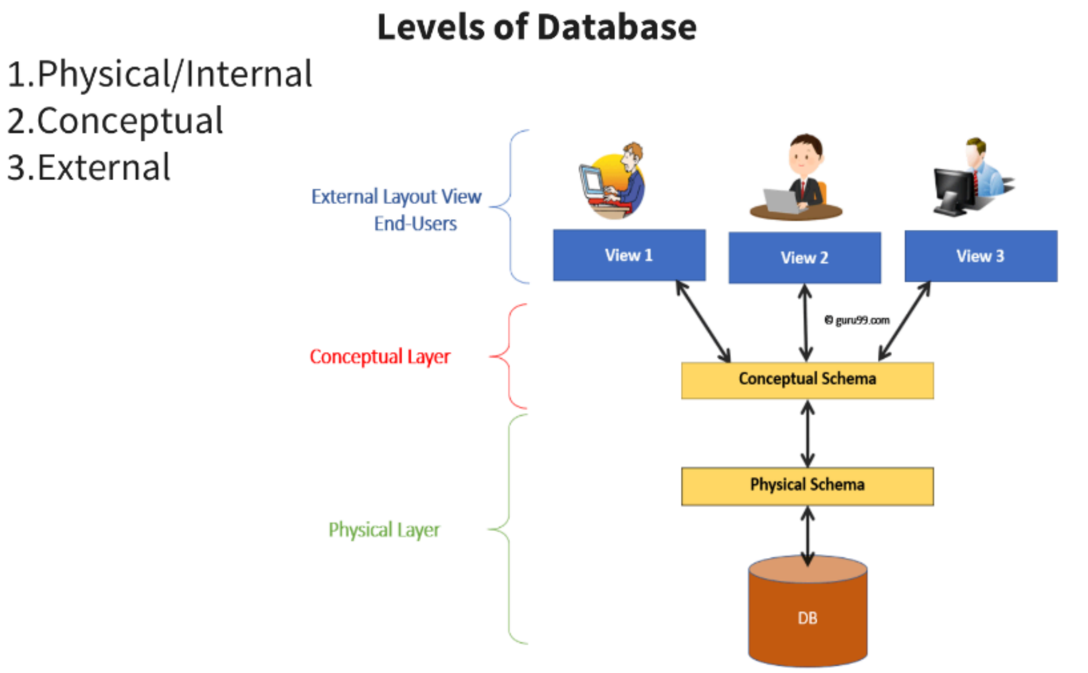
6 .Discuss 3-schema architecture.

Or

.Discuss 3-layers of DBMS system.

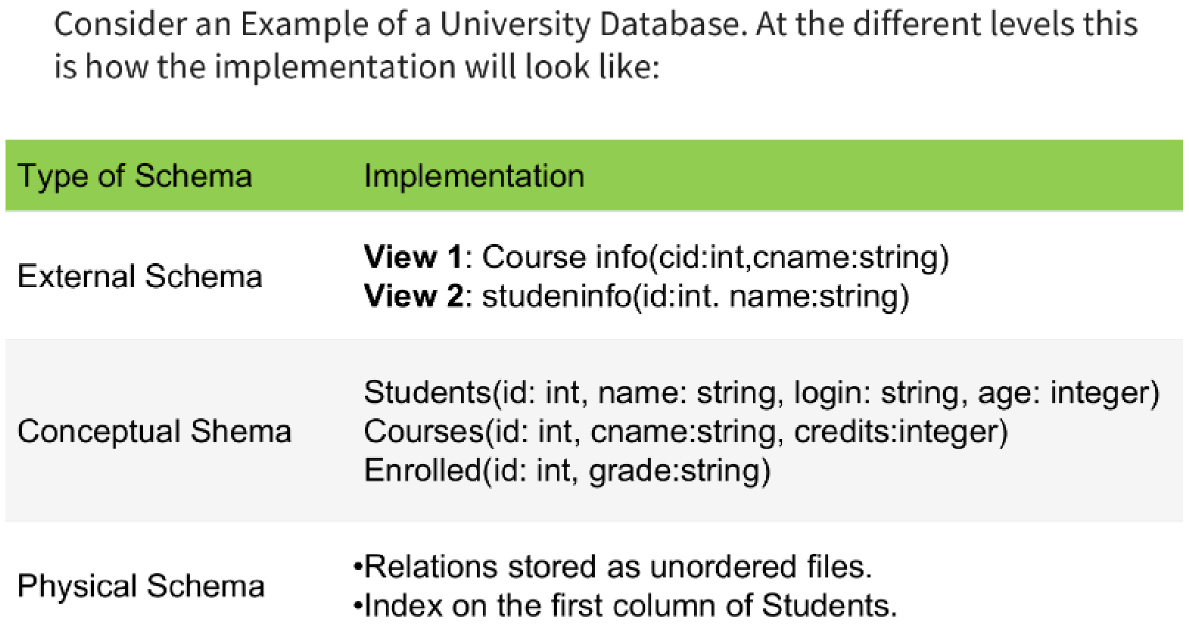
Or

Discuss overall structure of DBMS in detail with appropriate diagram. [5/10 marks]



Levels of Database

1.Physical/Internal
2.Conceptual
3.External



Consider an Example of a University Database. At the different levels this is how the implementation will look like:

Type of Schema	Implementation
External Schema	**View 1**: Course info(cid:int,cname:string) **View 2**: studeninfo(id:int. name:string)

Conceptual Shema	Students(id: int, name: string, login: string, age: integer)
	Courses(id: int, cname:string, credits:integer)
	Enrolled(id: int, grade:string)

Physical Schema	•Relations stored as unordered files. •Index on the first column of Students.

7. Classify the Database users.

1. Database administrators

Responsible for overseeing the overall management, design, security, and performance of the database system.

2. Naïve users

Casual users with limited technical knowledge of the database system.

3. Application programmers

Developers creating and maintaining software applications that interact with the database.

4. Sophisticated users

Users with a good understanding of the database system and its structure.

5. Specialized users

Users with expertise in specific domains or fields, requiring specialized knowledge for effective database use.

8. Explain Data abstraction (same for 3 level schema architecture)

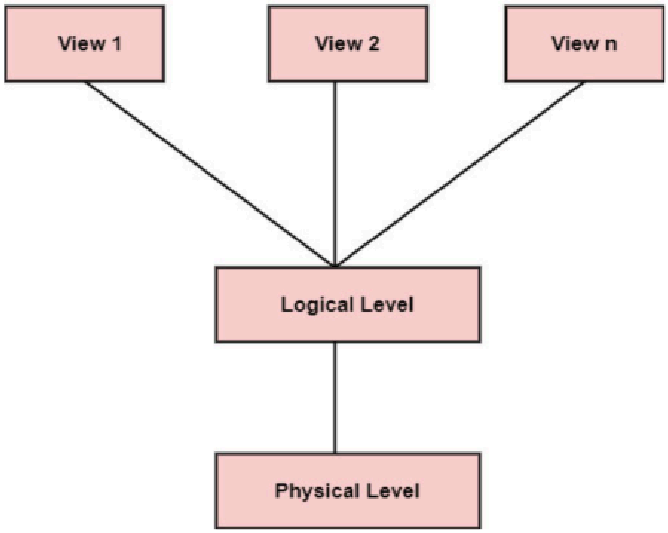
Data Abstraction is a process of hiding unwanted or irrelevant details from the end user.

The database systems consist of complicated data structures and relations. For users to access the data easily, these complications are kept hidden, and only the relevant part of the database is made accessible to the users through data abstraction.

Mainly there are three levels of abstraction for

DBMS, which are as follows −

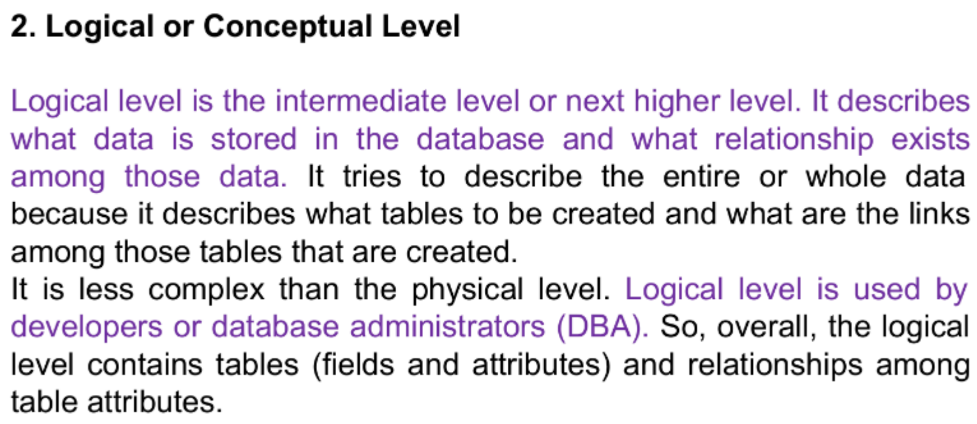
- Physical or Internal Level
- Logical or Conceptual Level
- View or External Level


Three Levels of Data Abstraction

1.Physical or Internal Level

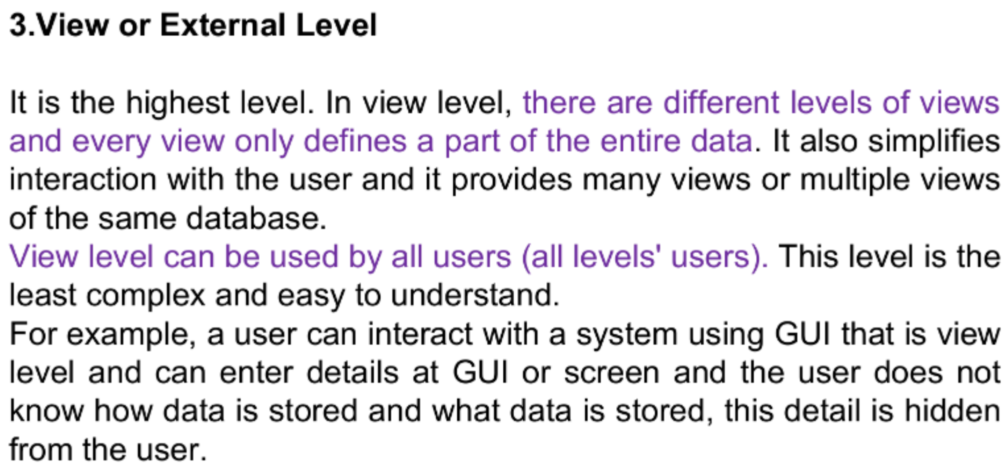
It is the lowest level of abstraction for DBMS which defines how the data is actually stored, it defines data-structures to store data and access methods used by the database. Actually, it is decided by developers or database application programmers how to store the data in the database.
So, overall, the entire database is described in this level that is physical or internal level. It is a very complex level to understand.

For example, customer's information is stored in tables and data is stored in the form of blocks of storage such as bytes, gigabytes etc.



2. Logical or Conceptual Level

Logical level is the intermediate level or next higher level. It describes what data is stored in the database and what relationship exists among those data. It tries to describe the entire or whole data because it describes what tables to be created and what are the links among those tables that are created.
It is less complex than the physical level. Logical level is used by developers or database administrators (DBA). So, overall, the logical level contains tables (fields and attributes) and relationships among table attributes.



3.View or External Level

It is the highest level. In view level, there are different levels of views and every view only defines a part of the entire data. It also simplifies interaction with the user and it provides many views or multiple views of the same database.
View level can be used by all users (all levels' users). This level is the least complex and easy to understand.
For example, a user can interact with a system using GUI that is view level and can enter details at GUI or screen and the user does not know how data is stored and what data is stored, this detail is hidden from the user.

9. Explain data Independence.

Data Independence is defined as a property of DBMS that helps you to change the Database schema at one level of a database system without requiring to change the schema at the next higher level.

Types of Data Independence

In DBMS there are two types of data independence

- Physical data independence
- Logical data independence.

Physical Data Independence

Physical data independence refers to the ability to change the data’s physical

structure without affecting the conceptual level.Physical changes include using a new storage device or moving the database’s location, changing the data structure, or altering indexes to speed up data retrieval.

Example: The patient database could be moved from drive C to drive D, but the conceptual schema and external views remain unchanged because of physical data independence.

Logical data independence, on the other hand, allows users to change the conceptual schema without changing the external views. For example, the hospital billing department could add a column to the database table for each patient's insurance policy number. Having logical data independence means that the view of the food services department and other users are not changed even though modifications were made to the conceptual level.

10. Explain types of DBA.

A Database Administrator (DBA) is individual or person responsible for controlling, maintenance, coordinating, and operation of database management system.

Administrative DBA – Their job is to maintain server and keep it functional. They

are concerned with data backups, security, trouble shooting, replication, migration etc.

Data Warehouse DBA – Accountable for merging data from various sources into data

warehouse. They also design warehouse, with cleaning and scrubs data prior to loading.

Development DBA – They build and develop queries that meets firm or organization needs.

Application DBA –They particularly manages all requirements of application components that interact with database and accomplish activities such as application installation and coordinating,

application upgrades, database cloning, data load process management, etc.

Architect – They are held responsible for designing schemas like building tables. They work to build structure that meets organization needs. The design is further used by developers and development DBAs to design and implement real application.

11. Write different roles and responsibilities of DBA.

Decides hardware – They decides economical hardware, based upon cost, performance and

efficiency of hardware, and best suits organization. It is hardware which is interface between end users and database.

Manages data integrity and security –Data integrity need to be checked and managed accurately as it protects and restricts data from unauthorized use. DBA eyes on relationship within data to maintain data integrity.

Database design – DBA is held responsible and accountable for database design ,integrity

and security control.

Database implementation – DBA implements DBMS and checks database loading at

time of its implementation.

Query processing performance – DBA enhances query processing

Module 2: Entity–Relationship Data Model

1. **Explain cardinality and participation with examples**
   Cardinality

   Cardinality defines the number of attributes in one entity set, which can be associated with the number of attributes of another set via a relationship set.

   it refers to the relationship one table can have with the other table.

   Defines the numerical attributes of the relationship between two entities or entity sets.

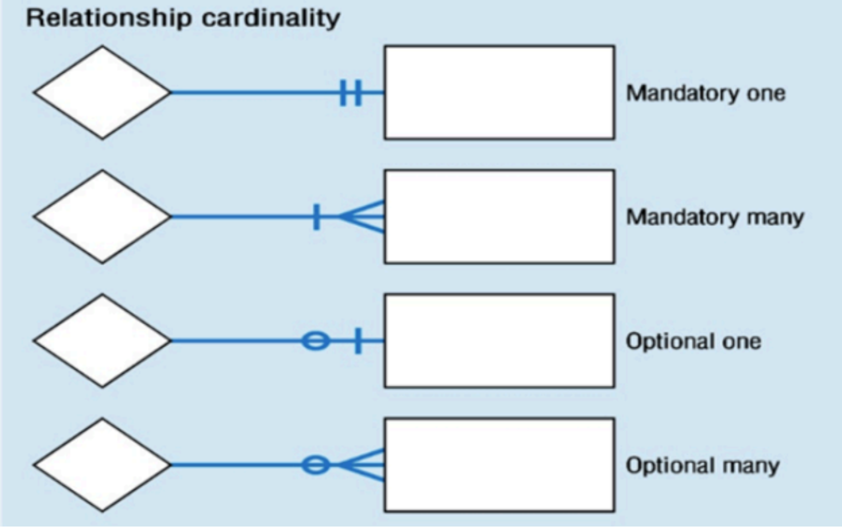
   Different types of cardinal relationships are:

   One-to-One Relationships

   One-to-Many Relationships

   May to One Relationships

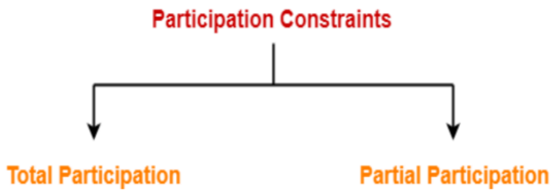
   Many-to-Many Relationships



Participation

Participation constraints deal with the participation of entities from an entity set in a relationship set

- **Types of Participation Constraints-**


Participation Constraints

Total Participation

Partial Participation

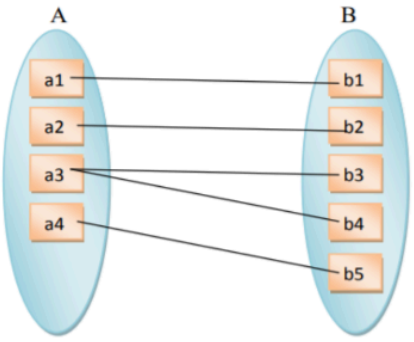
1. Total Participation-

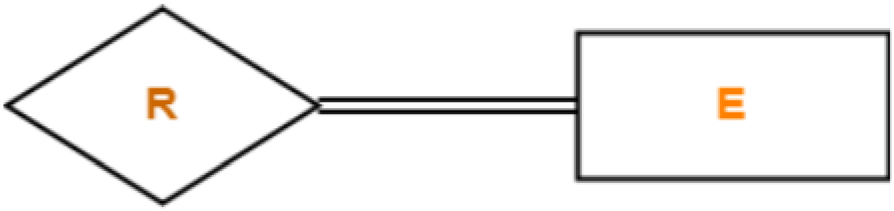
- The Participation of an entity set E in a relationship set R is said to be total if every entity in E participates in at least one relationship in R.

The participation of entity set A in the relationship set is **total** because every entity of A participates in the relationship set.

and

The participation of entity set B in the relationship set is also **total** because every entity of B also participates in the relationship set.





Total Participation

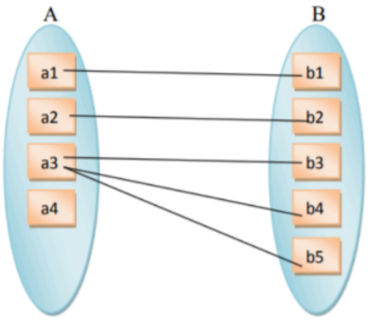
2. Partial Participation-

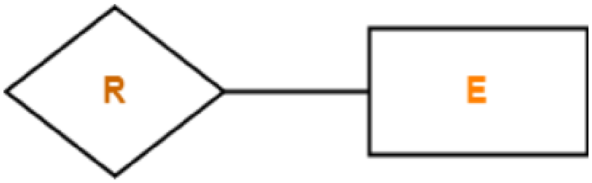
The participation of an entity set E in relationship set R is said to be partial if only some entities in E participate in relationships in R

The participation of entity set A in the relationship set is **partial** because only some entities of A participate in the relationship set.

while

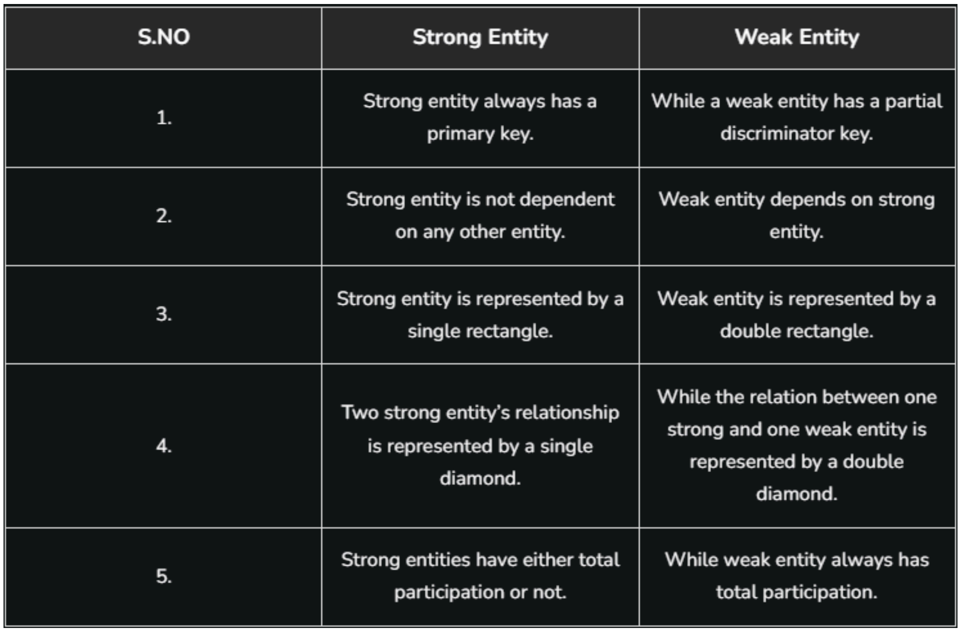
The participation of entity set B in the relationship set is **total** because every entity of B participates in the relationship set.





Partial Participation

2. **Draw E-R Diagram for given Problem statement**
3. **Draw and Explain EER with suitable example**
4. **Differentiate weak entity set and strong entity set supported with example**

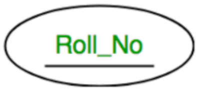


S.NO	Strong Entity	Weak Entity
1.	Strong entity always has a primary key.	While a weak entity has a partial discriminator key.
2.	Strong entity is not dependent on any other entity.	Weak entity depends on strong entity.
3.	Strong entity is represented by a single rectangle.	Weak entity is represented by a double rectangle.
4.	Two strong entity's relationship is represented by a single diamond.	While the relation between one strong and one weak entity is represented by a double diamond.
5.	Strong entities have either total participation or not.	While weak entity always has total participation.

5. Discuss different types of attributes with example and notations

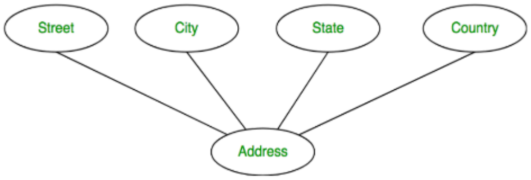
A. Key Attribute

The attribute which **uniquely identifies each entity** in the entity set is called key attribute.
For example, Roll_No or student ID will be unique for each student.
In ER diagram, key attribute is represented by an oval with underlying lines.



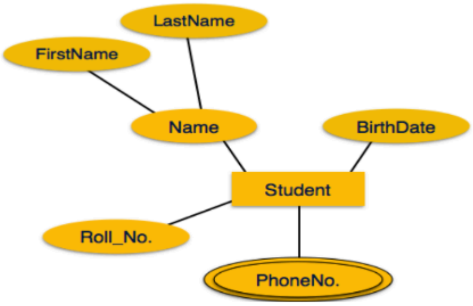
b. Composite Attribute

An attribute that composed of many other attributes is known as a composite attribute. The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.
For example, Address attribute of student Entity type consists of Street, City, State, and Country.



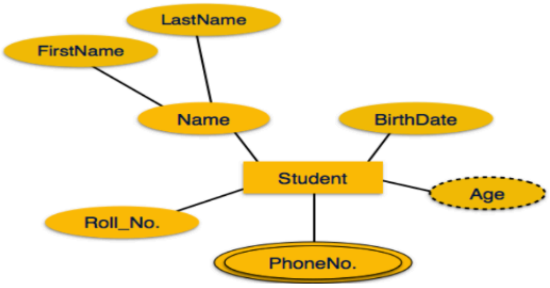
c. Multivalued Attribute

An attribute can have more than one value. These attributes are known as a multivalued attribute. The double oval is used to represent multivalued attribute.
For example, a student can have more than one phone number.

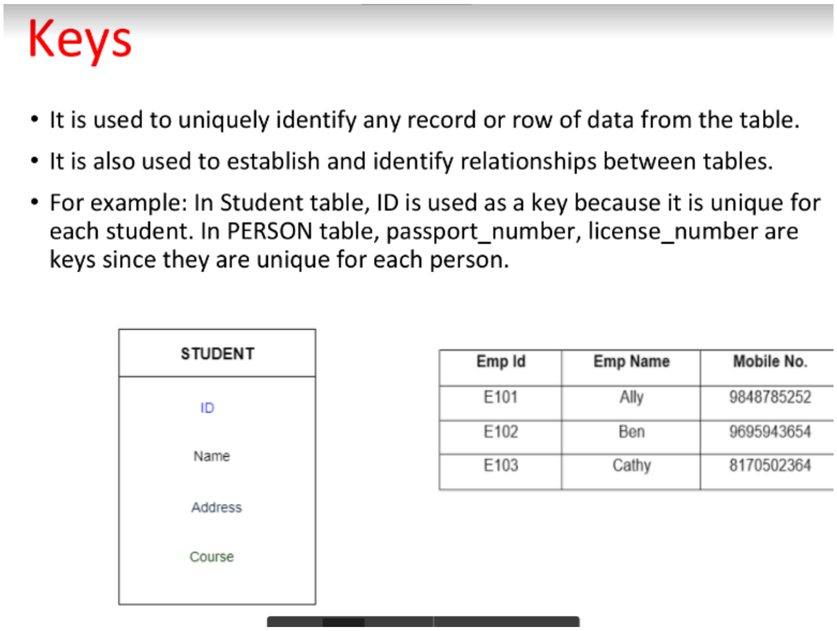


d. Derived Attribute

An attribute that can be derived from other attribute is known as a derived attribute. It can be represented by a dashed ellipse.
For example, A person's age changes over time and can be derived from another attribute like Date of birth.



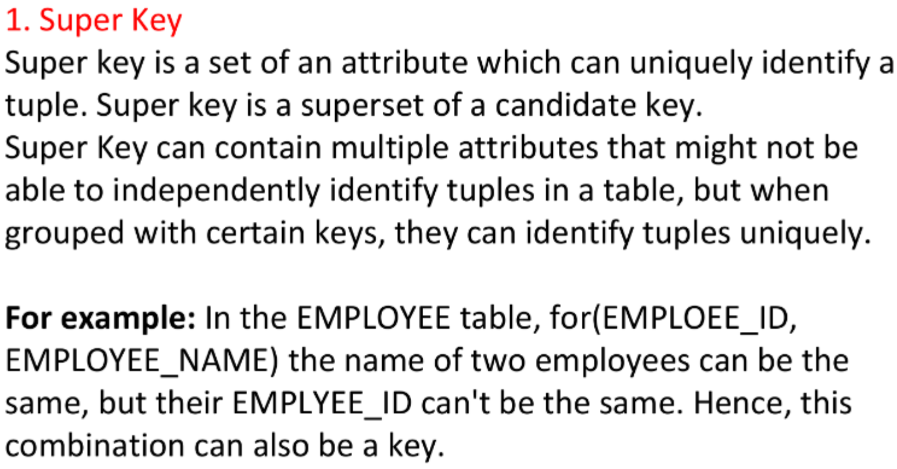
6. What is Concept of keys in database? Explain different types of keys with example.



Keys

- It is used to uniquely identify any record or row of data from the table.
- It is also used to establish and identify relationships between tables.
- For example: In Student table, ID is used as a key because it is unique for each student. In PERSON table, passport_number, license_number are keys since they are unique for each person.

Emp Id	Emp Name	Mobile No.
E101	Ally	9848785252
E102	Ben	9695943654
E103	Cathy	8170502364

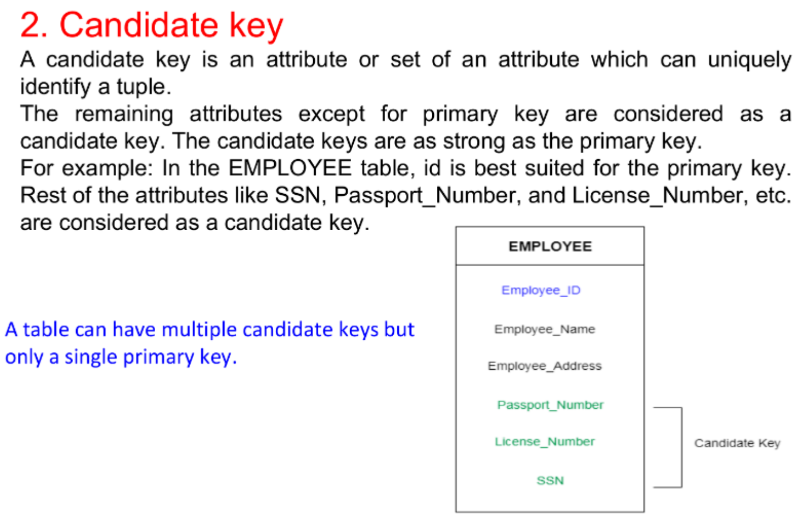


1. Super Key

Super key is a set of an attribute which can uniquely identify a tuple. Super key is a superset of a candidate key.

Super Key can contain multiple attributes that might not be able to independently identify tuples in a table, but when grouped with certain keys, they can identify tuples uniquely.

For example: In the EMPLOYEE table, for(EMPLOEE_ID, EMPLOYEE_NAME) the name of two employees can be the same, but their EMPLYEE_ID can't be the same. Hence, this combination can also be a key.



2. Candidate key

A candidate key is an attribute or set of an attribute which can uniquely identify a tuple.

The remaining attributes except for primary key are considered as a candidate key. The candidate keys are as strong as the primary key.

For example: In the EMPLOYEE table, id is best suited for the primary key. Rest of the attributes like SSN, Passport_Number, and License_Number, etc. are considered as a candidate key.

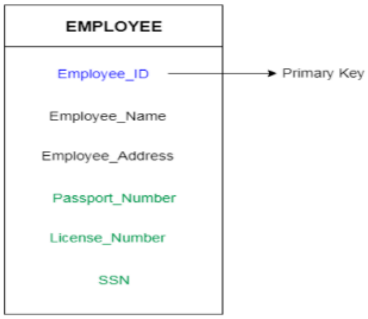
A table can have multiple candidate keys but only a single primary key.

3. Primary key

It is the first key which is used to identify one and only one instance of an entity uniquely.

A table cannot have more than one primary key.

In the EMPLOYEE table, ID can be primary key since it is unique for each employee. In the EMPLOYEE table, we can even select License_Number and Passport_Number as primary key since they are also unique.



4. Foreign key

- Foreign keys are the column of the table which is used to point to the primary key of another table.
- A foreign key is the one that is used to link two tables together via the primary key. It means the columns of one table points to the primary key attribute of the other table.

7. Explain all the relevant terminology with respect to ER and EER diagram

1. **ER Diagram (Entity-Relationship Diagram):** It's a visual representation of entities and their relationships within a database. ER diagrams are commonly used in database design to model the logical structure of a database.

2. **Entity:** An entity is a real-world object or concept that has attributes describing it. In a database context, entities are represented as tables.

3. **Attribute:** An attribute is a characteristic or property of an entity. In a database, attributes are represented as columns in a table.

4. **Relationship:** A relationship is an association between entities. It describes how entities are related to each other.

5. **Cardinality:** Cardinality specifies the number of occurrences of one entity that are related to the number of occurrences of another entity in a relationship. It's typically expressed as one-to-one (1:1), one-to-many (1:M), or many-to-many (M:N).

6. **Key Attribute:** A key attribute uniquely identifies each entity within an entity set. For example, in a student entity, the student ID might be a key attribute.

7. **Primary Key:** A primary key is a key attribute (or a combination of attributes) that uniquely identifies each record in a table. It ensures the uniqueness of each row in a table.

8. **Foreign Key:** A foreign key is an attribute (or a set of attributes) in one table that refers to the primary key in another table. It establishes a link between two tables.

9. **Weak Entity:** A weak entity is an entity that cannot be uniquely identified by its attributes alone. It depends on another entity, known as the owner entity, for its identity.

10. **Generalization:** Generalization is the process of defining a more general entity type from a set of more specialized entity types. It's used in hierarchical modeling to represent inheritance relationships.

11. **Specialization:** Specialization is the process of defining a set of more specialized entity types from a more general entity type. It's used in hierarchical modeling to represent inheritance relationships.

12. **Subtype:** A subtype is a subset of a supertype in a specialization hierarchy. Subtypes inherit attributes and relationships from their supertype.

13. **Supertype:** A supertype is a more general entity type from which one or more subtypes inherit attributes and relationships.

14. **Aggregation:** Aggregation is the process of treating a group of related entities and relationships as a single entity. It's used to model complex relationships between entities.

15. **EER Diagram (Enhanced Entity-Relationship Diagram):** EER diagram extends the ER model with additional concepts such as specialization, generalization, aggregation, etc., to provide a more expressive way to represent complex database designs.

8. Discuss Specialization, Generalization and aggregation with suitable example?

Specialization

The process of defining subclasses of an entity type is called specialization.

• This entity type is called “superclass” of the specialization.

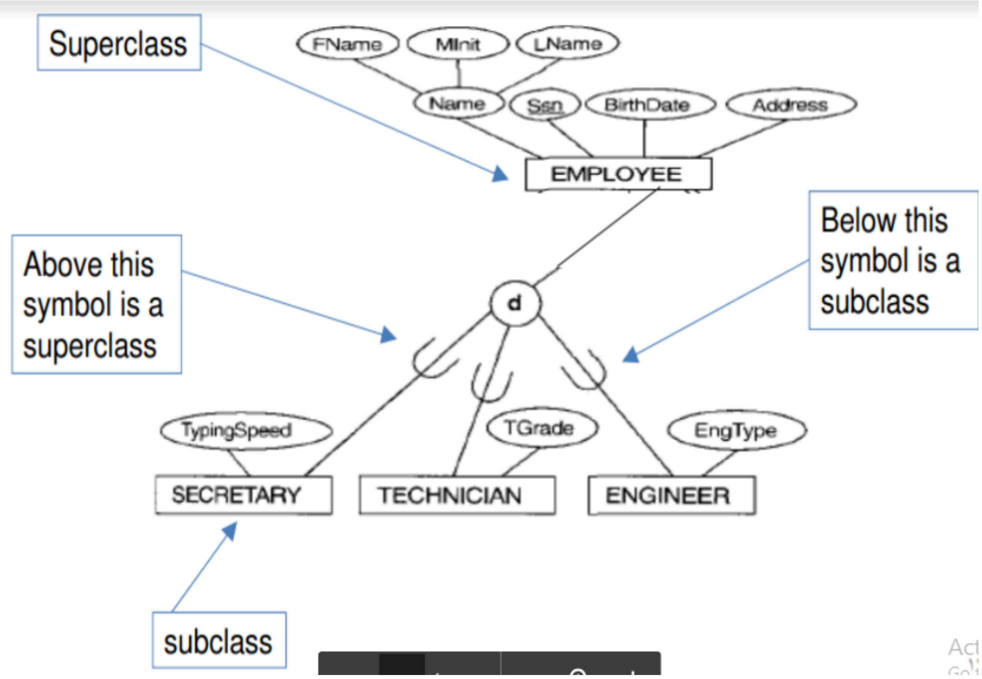
• Specialization distinguishes between subclasses based on a certain method:-

Example: Salaried Employee and Hourly Employee are grouped together because they are classified based on paying method.

The Employee entity type has specializations:

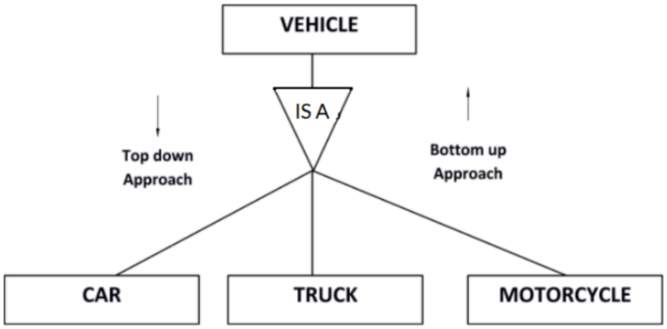
1) {SalariedEmployee,HourlyEmployee}
   - Classified based on paying method.

2) {Secretary, Technician, Engineer}
   - Classified based on job type.



Generalization is a Bottom up process i.e. consider we have 3 sub entities Car, Truck and Motorcycle. Now these three entities can be generalized into one super class named as Vehicle

Specialization is a top down approach in which one entity is broken down into low level entity.



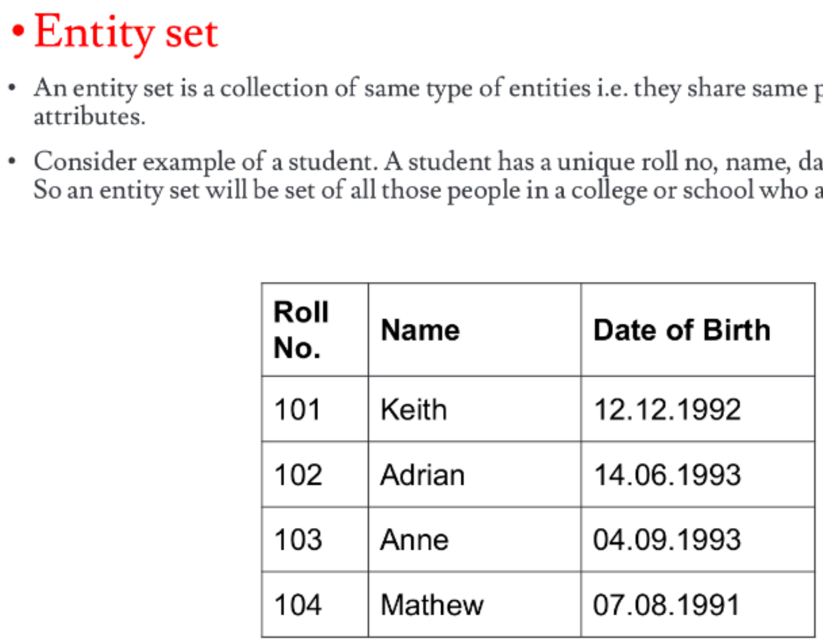
Generalization

- Generalization is the reverse process of specialization.
- In generalization, you generalize a set of entity types into one superclass entity type. Therefore, the generalized entity types are considered subclasses.
- If you find a set of classes with many common attributes, they can be considered subclasses and generalized to a common entity (superclass).

Aggregation

- An ER diagram is not capable of representing 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.
- Aggregation is an abstraction through which we can represent relationships as higher level entity sets.

9. Explain Entity set



• Entity set

- An entity set is a collection of same type of entities i.e. they share same properties or attributes.
- Consider example of a student. A student has a unique roll no, name, date of birth etc. So an entity set will be set of all those people in a college or school who are students.

Roll No.	Name	Date of Birth
101	Keith	12.12.1992
102	Adrian	14.06.1993
103	Anne	04.09.1993
104	Mathew	07.08.1991

Module 3: Relational Model and relational Algebra

1. Discuss concept of keys in detail.

2. Apply and show the use of all the relevant terminology of Relational Model with example.

Concepts

Tables − In relational data model, relations are saved in the format of Tables.

A table has rows and columns, where rows represents records and columns represent the attributes.

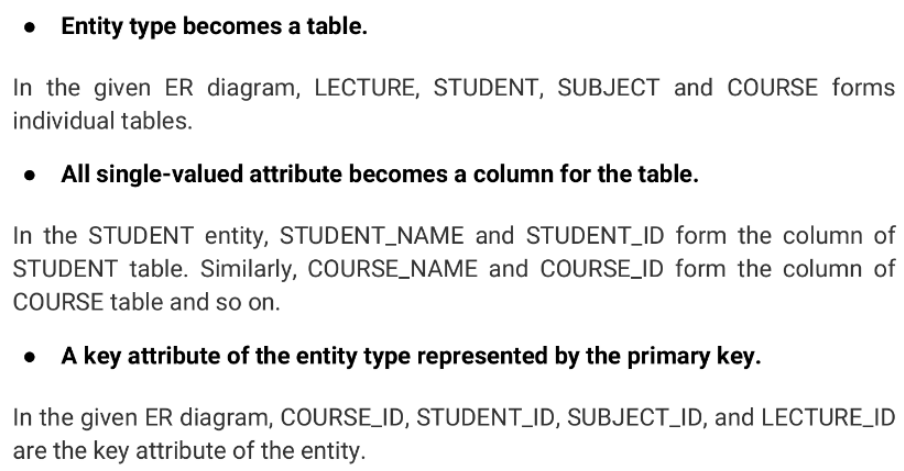
Tuple − A single row of a table, which contains a single record for that relation is called a tuple.

Relation instance − A finite set of tuples in the relational database system represents relation instance. Relation instances do not have duplicate tuples.

Relation schema − A relation schema describes the relation name (table name), attributes, and their names.

Relation key − Each row has one or more attributes, known as relation key, which can identify the row in the relation (table) uniquely.

3. Explain Mapping the ER Model to the Relational Model with example.



- **Entity type becomes a table.**

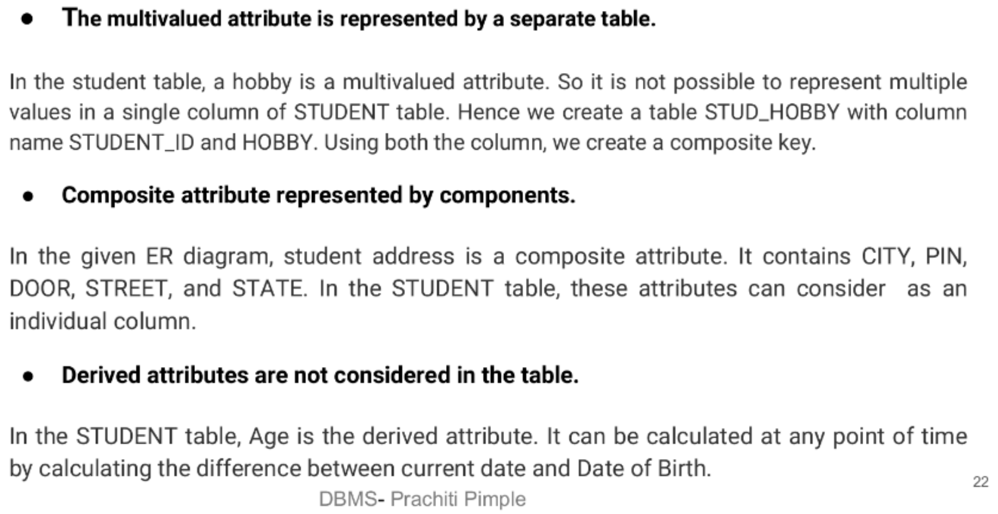
In the given ER diagram, LECTURE, STUDENT, SUBJECT and COURSE forms individual tables.

- **All single-valued attribute becomes a column for the table.**

In the STUDENT entity, STUDENT_NAME and STUDENT_ID form the column of STUDENT table. Similarly, COURSE_NAME and COURSE_ID form the column of COURSE table and so on.

- **A key attribute of the entity type represented by the primary key.**

In the given ER diagram, COURSE_ID, STUDENT_ID, SUBJECT_ID, and LECTURE_ID are the key attribute of the entity.



- **The multivalued attribute is represented by a separate table.**

In the student table, a hobby is a multivalued attribute. So it is not possible to represent multiple values in a single column of STUDENT table. Hence we create a table STUD_HOBBY with column name STUDENT_ID and HOBBY. Using both the column, we create a composite key.

- **Composite attribute represented by components.**

In the given ER diagram, student address is a composite attribute. It contains CITY, PIN, DOOR, STREET, and STATE. In the STUDENT table, these attributes can consider as an individual column.

- **Derived attributes are not considered in the table.**

In the STUDENT table, Age is the derived attribute. It can be calculated at any point of time by calculating the difference between current date and Date of Birth.

4. Explain Mapping the EER Model to the Relational Model with example.

same as above