PAPER IV :

DATABASE MANAGEMENT SYSTEMS

Unit 1 :

1. Define database and DBMS. Explain different advantages of DBMS.

⇒ Database :- A database is a collection of data that is related in some way.

DBMS :- A DBMS is a collection of programs that enables the user to create and ~~run~~ a maintain a database.

Advantages of DBMS :-

The database management system has a no. of advantages as compared to traditional computer file-based processing approach. The ~~BP~~ DBA must keep in mind these benefits or capabilities during databases and monitoring the DBMS. The main advantages of DBMS are :-

* Controlling Data Redundancy :-

In non-database systems each application program has its own private files. In this case, the duplicate copies of the same data is created in many places. In ~~DMS~~ DBMS, all data of an organization is integrated into a single database file. The data is recorded in only one place in the database and it is not duplicated.

* Sharing of data :-

In DBMS, data can be shared by authorized users of the organization. The database administrator manages the data and gives rights to users to access the data. Many users can be authorized to access the same piece of information simultaneously. The remote users can also share same data. Similarly, the

data of same database can be shared between
application programs.

* Data consistency :-

By controlling the data redundancy, the data
is obtained. If a data item appears only once
update to its value has to be performed only
and the updated value is immediately available
all users. If the DBMS has controlled redundan
the database system enforces consistency.

* Integration of data :-

In DBMS, data in database is stored in tables. A
single database contains multiple tables and relations
can be created between tables. This makes easy to
retrieve and update data.

* Integration constraints :-

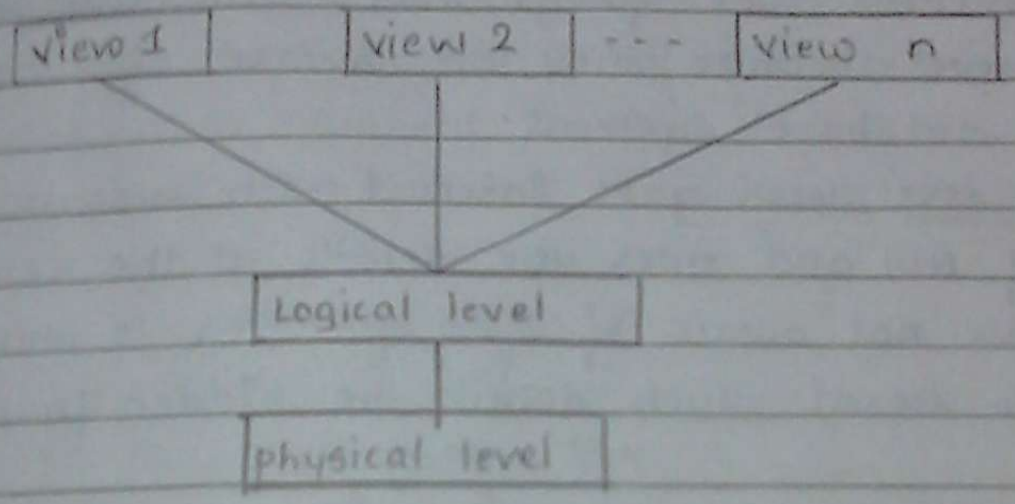
Integrity constraints or consistency rules can be app
to database so that the correct data can be entered
database. The constraints may be applied to data
item within a single record or may be applied to
relationships between records

* Backup and recovery procedures :-

In a computer file based system, the user creates
backup of data regularly to protect the valuable data
from damage due to failures to the computer sys
or application program. It is very time consuming
method; if amount of data is large most of the
provide the backup and recovery subsystems that
automatically create the backup of data and
restore data if required.

2. What do you mean by abstraction? Explain different levels of abstraction with example.

⇒ Database systems are made of complex data structures. To ease the user interaction with database, the developers hide internal irrelevant details from users. This process of hiding irrelevant details from user is called data abstraction.



Three levels of data abstraction:

We have three levels of abstraction:

Physical level :- This is the lowest level of data abstraction. It describes how data is actually stored in database. You can get the complex data structure details at this level.

Logical level :- This is the middle level of 3 level data abstraction architecture. It describes what data is stored in a database.

View level :- Highest level of data abstraction. This level describes the user interaction with database system.

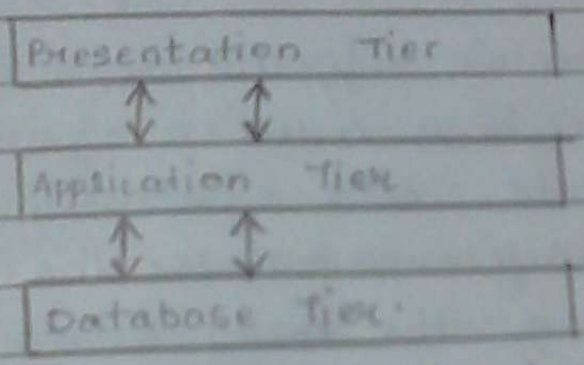
Example :- Let's say we are storing customer info
In a customer table.
At physical level, these records can be described a
blocks of storage in memory. This details are of
hidden from the programmers.
At the logical level, these records can be described
fields and attributes along with their datatype
Their relationship among each other can be logic
implemented. The programmers generally work a
this level because they are aware of such thing
about database systems.
At view level, user just interact with system with th
help of GUI and enter the details at the screen.
They are not aware of how the data is stored and
data is stored. Such details are hidden from them

3. Explain architecture of DBMS with proper diagram.
⇒ The design of a DBMS depends on its architecture
It can be centralized or decentralized or hierarchic
The architecture of a DBMS can be seen as either
single tier or multi-tier. An n-tier architecture div
the whole system into related but independend n-m
which can be independently modified, altered, changed
replaced.
In 1-tier architecture, the DBMS is the only entity
where the user directly sits on the DBMS and use
Any changes done here will directly be done on t
DBMS itself. It does not provide handy tools for
end users. Database designers and programmers
normally prefer to use single tier architecture.

If the architecture of DBMS is 2-tier, then it must have an application through which the DBMS can be accessed. Programmers use 2-tier architecture where they access the DBMS by means of an application. Here, the application tier is entirely independent of the database in terms of operation, design, and programming.

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.



Database (Data) Tier :- At this tier, the database resides along with its query processing languages. We also have the relations that define the data and their constraints at this level.

Application (middle) Tier :- At this tier reside the application servers and the programs that access the database. For a user, this application tier presents an abstracted view of the database. End-users are unaware of any existence of the database beyond the application. At the other end, the database tier is not aware of any other user beyond the application tier. Hence, the application layer sits in the middle

and acts as a mediator between the end user and the database.

User (Presentation) Tier :- End users operate on this tier and they know nothing about any existence of the database beyond this layer. At this layer, multiple views of the database can be provided by the application. All views are generated by applications that reside in the application tier.

multiple tier database architecture is highly modifiable as almost all its components are independent and can be changed independently.

4. List and explain different types of data models.

⇒ A data model is a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints. The two types of data models are - the entity relationship model and the relational model.

Data model structure and constraints :-

* Constructs are used to define the database structure.
* Constructs typically include elements as well as groups of elements and relationships among such groups.
* Constraints specify some restrictions on valid data; these constraints must be enforced at all times.

Data model operations :-

These operations are used for specifying database retrievals and updates by referring to the consts

of the data model.
operations on the data model may include basic model operations and user-defined operations.
The hierarchial is the oldest DBMS data model, and the object-oriented being the newest DBMS data model.

1. The entity relationship (ER) model :
It is a high-level data model. It is based on a perception of a real world that consists of a collection of basic objects, called entities, and of relationships among these objects.

2. The relational model :-
It is a lower level model. It uses a collection of tables to represent both data and the relationship among those data. Its conceptual simplicity has led to its widespread adoption; today a vast majority of database products are based on the relational model. Designers often formulate database schema design by first modeling data at a high-level, using the E-R model and then translating it to the relational model.

3. Hierarchial Data model :-
In the hierarchial data model, information is organized as a collection of inverted tree of records. The inverted trees may be of arbitrary depth. The record at the root of a tree has zero or more child records; the child records, in turn, serve as parent records for their immediate descendants.
This parent-child relationship recursively continues down the tree. The records consists of fields where each field may contain simple data values

or a pointer to a record. The pointer graph is not allowed to contain cycles. Some combinations of fields may form the key for a record relative to its parent.

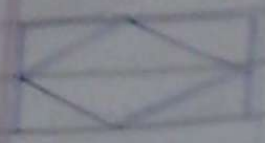
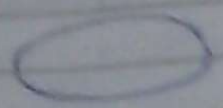
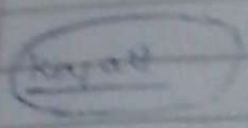
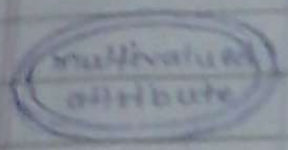
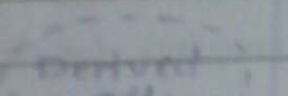
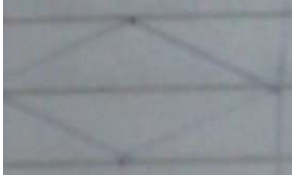
4. Network Data model:

In the network datamodel, information is organized as a collection of graphs of record that are related with pointers. Network data models represent data in a symmetric manner, unlike the hierarchical data model. A network data model is more flexible than a hierarchial data model and still permits efficient navigation.

5. List and explain different symbols used to draw an E-R diagram.

=>

Symbol	Shape Name	Symbol Description:
Entity	Entity	An entity is represented by a rectangle which contains the entity's name.
Weak Entity	Weak entity	An entity that cannot be uniquely identified by its attributes alone. The existence of a weak entity is dependent upon one another entity called the owner entity. The weak entity's identifier is a combination of the identifier of the owner identity and the partial key of the weak entity

	Associative entity	An entity used for a many-to-many relationship. All relationships for the associative entity should be many-to-many.
	Attribute	Each attribute is represented by an oval containing attribute's name.
	Key attribute	An attribute that uniquely identifies a particular entity. The name of a key attribute is underlined.
	multivalued attribute	An attribute that can have many values. Multivalued attribute is depicted by a oval.
	Derived attribute	An attribute whose value is calculated from other attributes. The derived attribute may or may not be physically present in the database.
	Strong relationship	A relationship where entity is existence independent of other entities and PK of child doesn't contain PK of component of parent entity. A strong relationship is represented by a single rhombus.

6. Explain relational model in DBMS.

⇒ Relational data model is the primary data model which is widely used around the world for data storage and processing. This model is simple and it has all the properties and capabilities required to process data with storage efficiency.

Concepts

Tables :- In relational data model, relations are saved in the format of tables. This format stores the relation among entities. 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.

Relational 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, 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 uniquely.

Attribute domain :- Every attribute has some pre-defined value scope, known as attribute domain.

7. List and explain different types of users in DBMS.

⇒ Database users are the one who really use and take the benefits of database. There will be different types of users depending on their need and way of accessing the database.

1. Application programmers :- They are the developers who interact with the database by means of DML queries. These DML queries are written in the application programs like C, C++, JAVA, Pascal, etc. These queries are converted into object code to communicate with the database. For example, writing a C program to generate the report of employees who are working in particular department will involve a query to fetch the data from database. It will include a embedded SQL query in the C program.

2. Sophisticated users :- They are database developers, who write SQL queries to select/insert/delete/update data. They do not use any application or programs to request the database. They directly interact with the database by means of query language like SQL. These users will be scientists, engineers, analysts who thoroughly study SQL and DM DBMS to apply the concepts in their requirements.

3. Specialized users :- These are also sophisticated users, but they write special database application programs. They are the developers who develop the complex programs to the requirement.

4. Stand-alone users :- These users will have stand-alone database for their personal use. These kinds of database will have readymade database packages which will have menus and graphical interfaces.

5. Native users :- These are the users who use the existing application to interact with the database. For example, online library system, ticket booking systems, ATM's etc. which has existing application and users use them to interact with the database to fulfil their requests.

8. List and explain different types of relationships in E-R model.

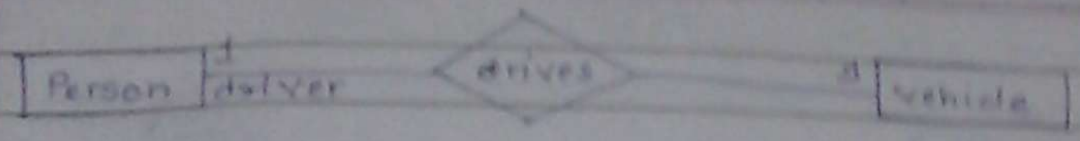
⇒ A relationship type defines a relationship set among entities of certain entity types. A relationship type is illustrated in an ERD using a diamond symbol.

Types of Relationships :-

i) One to one relationship :

One to one relationships have 1 specified for both cardinalities and do not seem to arise very often. To illustrate a one-to-one, we require very specific business rules.

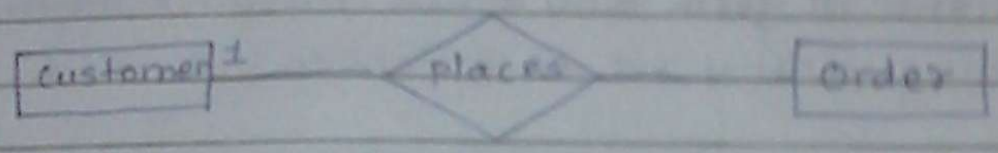
Suppose we have people and vehicles. Assume that we are only concerned with the current driver of a vehicle and that we are only concerned with the current vehicle that a driver is operating. Then we have a one-to-one relationship between vehicle and person.



ii) one-to-many relationships :-

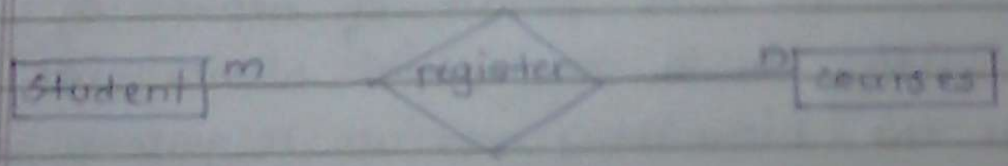
this type of relationship has 1 and n specified for cardinalities and is very common in database designs. Suppose we have customers and orders and the business rules :

- an order is related to one customer and
- a customer can have any number of orders.



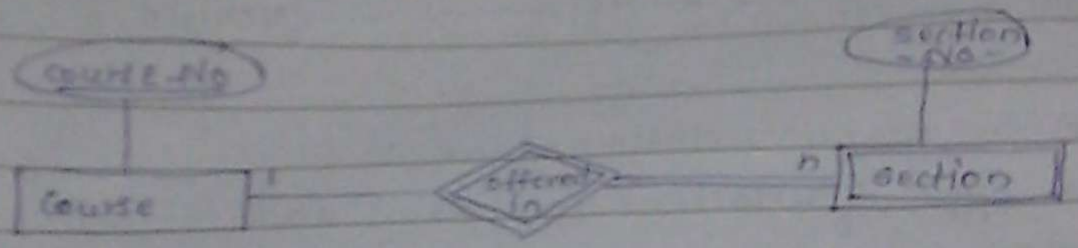
iii) many-to-many relationships :- many-to-many relationships have "many" specified for both cardinalities and are also very common.

Suppose, we are interested to courses and students and the fact that students register for courses. Any student may take several courses. A course may be taken by several students.

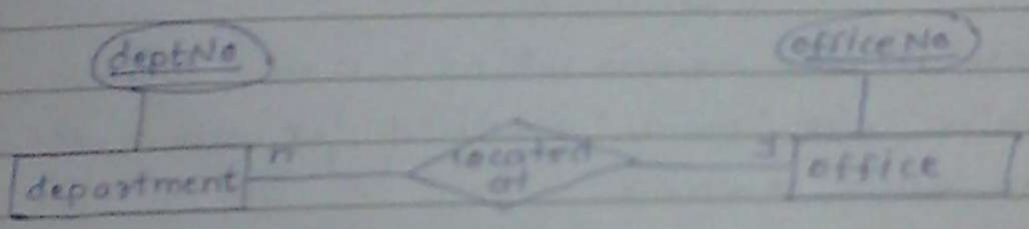


iv) Identifying relationships :-

An Identifying relationship is a relationship between a strong and a weak entity types where the key of the strong entity type is required to uniquely identify instances of the weak entity type. The weak entity type will have a partial key attribute that, in conjunction with the key of the strong entity type, uniquely identifies a weak entity instances.



v) Non-Identifying Relationships : A non-identifying relationship is a relationship between strong entity types. Each entity type has a key specified; each entity has an attribute that uniquely identifies it and distinguishes it from any other instance in the corresponding entity set or extension.



vi) Recursive relationships :

9) Write a note on generalization and specialization.

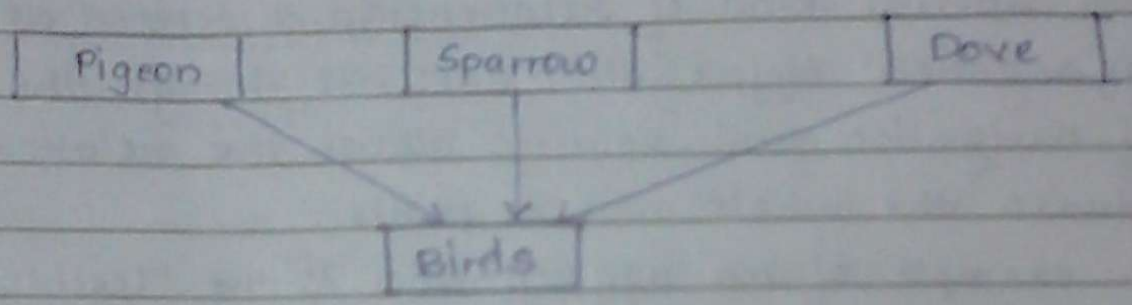
→ The ER model has the power of expressing database entities in a conceptual hierarchial manner. As the hierarchy goes up, it generalizes the view of entities, and as we go deep in the hierarchy, it gives us the detail of every entity included.

Going up in this structure is called generalization, where entities are clubbed together to represent a more generalized view. For example, a particular student named mina can be generalized along with all the students. The entity shall be a

student, and further, the student is a person. The reverse is called specialization, where a person is a student and that student is mine.

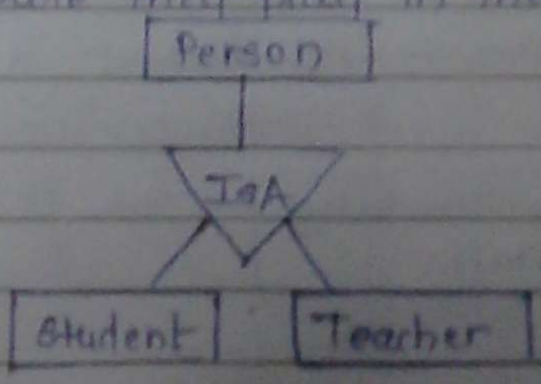
Generalization :

As mentioned above, the process of generalizing entities where the generalized entities contain the properties of all the generalized entities is called generalization. In generalization, a no. of entities are brought together into one generalized entity based on their similar characteristics. For eg., pigeon, house sparrow, crow and dove can all be generalized as birds.



Specialization :

It is the opposite of generalization. In specialization, a group of entities is divided into sub-groups based on their characteristics. Take a group 'Person' for example. A person has name, date of birth, gender, etc. These properties are common in all persons, human beings. But in a company, persons can be identified as employee, employer, customer, or, vendor, based on what role they play in the company.

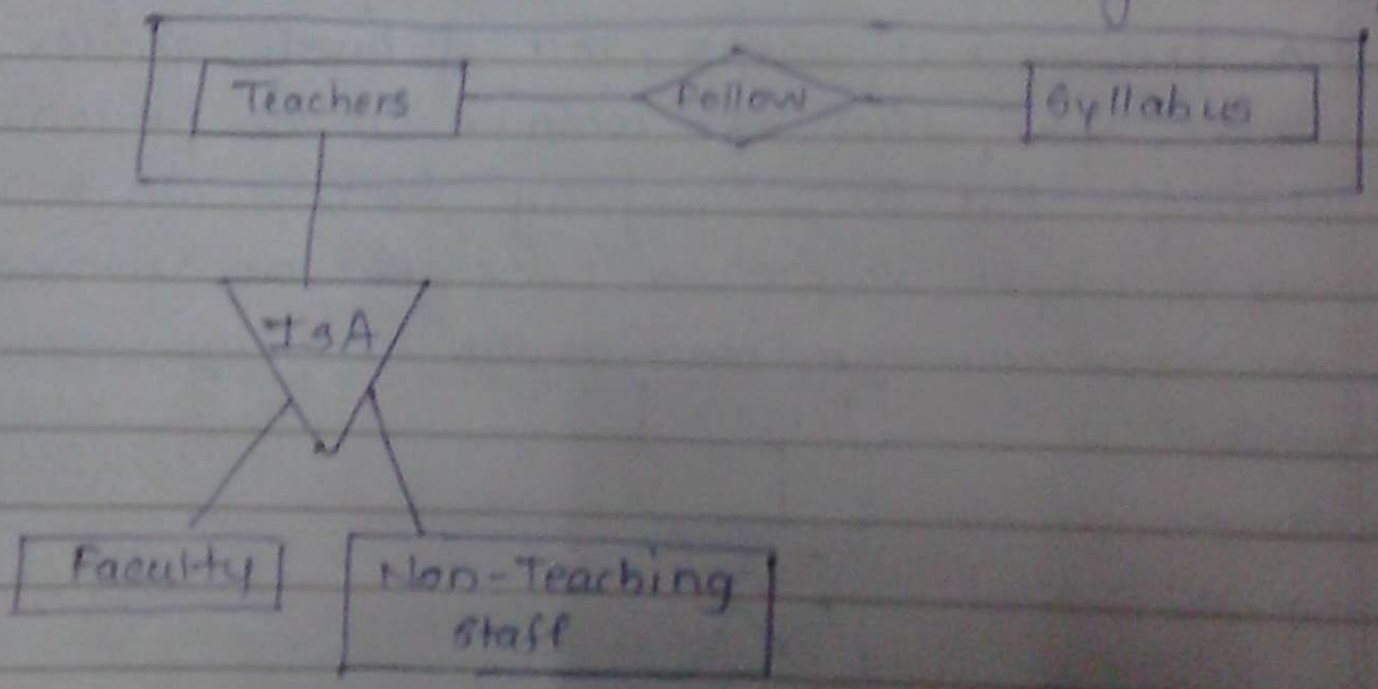


Similarly, In a school database, persons can be specialized as teacher, student, or a staff, based on what roles they play In a school as entities.

11) What do you mean by aggregation? Explain with proper example.

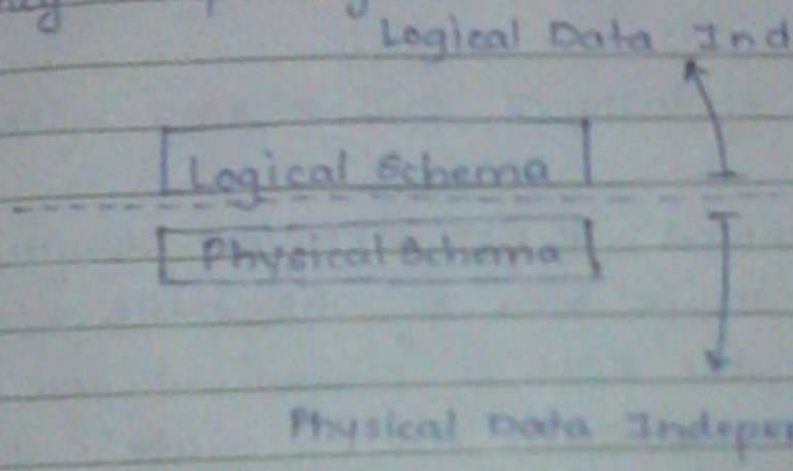
⇒ A relationship represents a connection between two entity types that are conceptually at the same level. Sometimes you may want to model a 'has-a', 'is-a' or 'is-part-of' relationship, in which one entity represents a larger entity (the 'whole') that will consist of smaller entities (the 'parts'). This special kind of relationship is termed as aggregation. Aggregation does not change the meaning of navigation and routing across the relationship between the whole and its parts.

An example of an aggregation is the 'Teacher' entity following the 'Syllabus' entity acts as a single entity in the relationship. In simple words, aggregation is the process where the relationship between two entities is treated as a single entity.



14) What do you mean by data Independence?

→ A database system normally contains a lots of data in addition to user's data. For example, it stores data about data, known as metadata, to locate and retrieve data easily. It is rather difficult to modify or update a set of metadata once it is stored in the database. But as a DBMS expands, it needs to change over time to satisfy the requirements of the users. If the entire data is dependent, it would become a tedious and highly complex job.



metadata itself allows a layered architecture, so that when we change data at one layer, it does not affect the data at another level. This data is independent but mapped to each other.

Logical Data Independence :-

Logical data is data about database, i.e., it stores information about how data is managed inside. For eg, a table stored in the database and all its constraints applied on that relation.

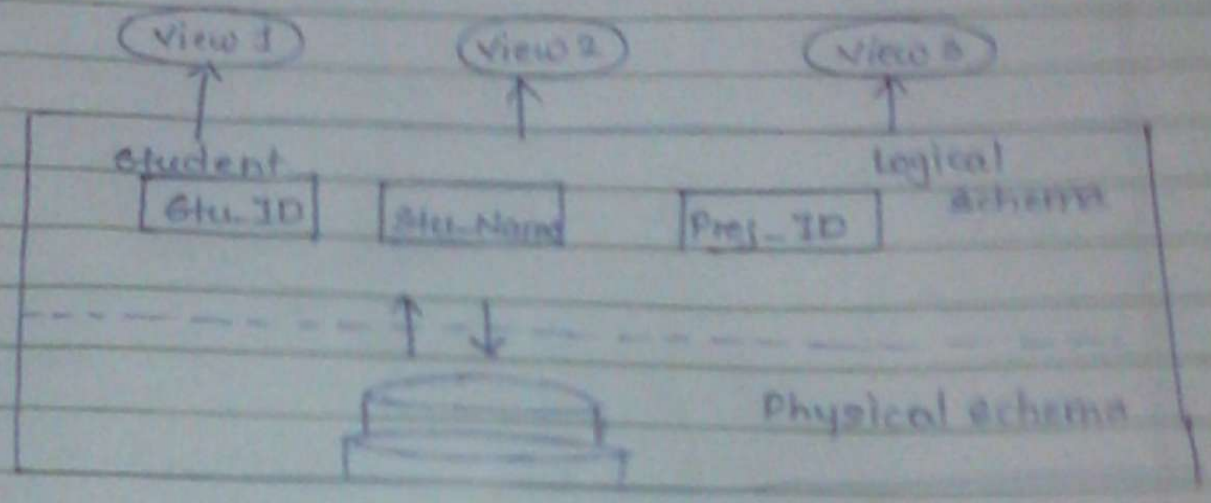
Logical Data Independence is a kind of mechanism, which liberalizes itself from actual data stored on the disk. If we do some changes on table format, it should not change the data residing on the disk.

Physical Data Independence :
All the schemas are logical, and the actual data is stored in bit format on the disk. Physical data Independence is the power to change the physical data without impacting the schema or logical data.
For example, in case we want to change or upgrade the storage system itself - suppose we want to replace hard disks with SSD - it should not have any impact on the logical data or schemas.

18. Explain concept of instance and schema with example.
* A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.
A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. Its the database designer who design the schemas to help programmers understand the database and make it useful.

A database schema can be divided broadly into two categories :
* physical Database schema : this schema pertains to the actual storage of data and its form of storage like files, indices, etc, It defines how the data will be stored in the secondary storage.

* Logical Database Schema : This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views and Integrity constraints.



Database Instance : It is important that we distinguish these two terms individually. Database Schema is the skeleton of database. It is designed when the database doesn't exist at all. Once the database is operational, it is very difficult to make any changes to it. A database schema does not contain any data or information.

A database Instance is a state of operational database with data at any given time. It contains a snapshot of the database. Database Instances tend to change with time. A DBMS ensures that its every instance is in a valid state, by diligently following all the validations, constraints and conditions that the database designers have imposed.