Sem-II

BARFI Page No. Date

DS

1 Dec 2022 DBMS

CRUDE - Create Read update Delete Execute

Defn - Database

* Intro -
- The Database management system is required as software system that allows users to define, create, maintain and control access to the database

- DBMS makes it possible for end users to create, read, update, delete data in database.

Advantages - DBMS

① Reducing data Redundancy (Repitition).
meaning - The file base DBMS contained multiple files that work stored in many different location in a system.
→ because of this there must work sometimes multiple copies of same file which lead to data Redundancy.
→ This is prevented in Database as there is a signal database and any change in it is reflected immediately.
→ Because of this there is no change of encounter duplicate data.

② sharing of data
→ In a database, the users of the database can share the data among themself.
→ There are various levels of authorization to access the data and consequently the data can only be shared based on the correct authorization protocols to be followed

motherboard
magnetic disk
access
handel - changes
BARFI Page No. Date
No, changes

③ Data integrity -
→ - It means that the data is accurate and consistent in the database.
→ It is very important as there multiple databases in a DBMS and it is visible to multiple users.
→ So, it is necessary to ensured that data is correct in all the databases and for all the users.

2 Dec 2022

④ Data security -
- It is a vital (important) concept in a database.
- only authorized user's should be allows to access the database and there identity should be authenticated using the username and passwd.

⑤ privacy -

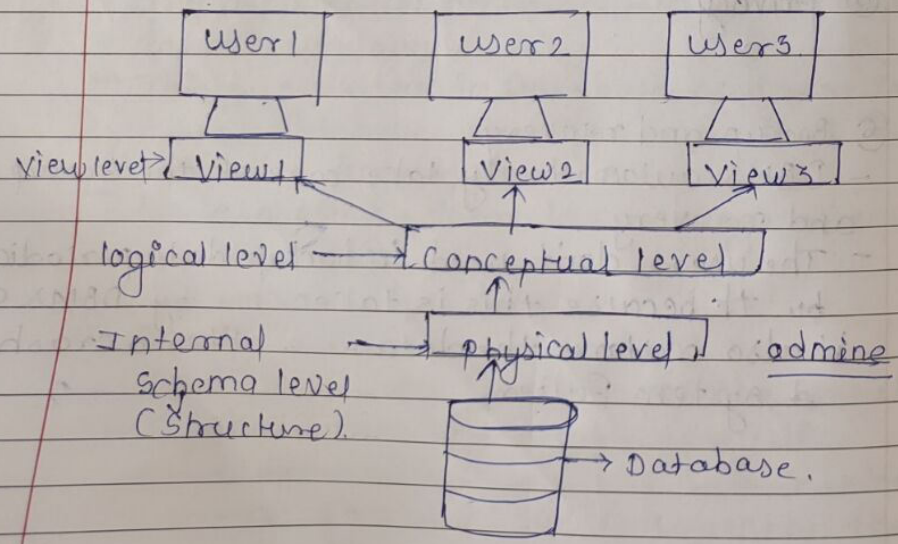
⑥ Backup and recovery
- DBMS automatically take care of the backup and recovery.
- The users don't need to backup data periodically because this is taken care by DBMS and it also restores the database after a crash or system failier.

hiding of data

* Abtraction in DBMS -

- It is refers to the processes of hiding in irrelavant details from the user.

For ex:-
- If we want to access any mail from our gmail there we don't know where the data is physically stored. that is the data is present in India or USA.
- We are not confined about this things and hence location of data and data models are irrelvant to us, and in data abtraction we do this only.
- There are 3 levels of data Abtractions,
  1) View level
  2) conceptual
  3) physical level



13 Dec

Relational DBMS

1) View level :-
- This level tell's the application how the data should be shown to the user.

eg:-
- If we have a login id and passwd in a university system then as a student, we can view our marks, fee structure etc.
- but the faculty of university have different view. we will have options like edit marks of student enter attendence etc.
- So both the student and faculty have a diffrn view, and by view so the security and privacy of the system is maintained.

2) Conceptual level (logical).
create updation
- It tell's how the data is acutaly stored and structured

eg:-
- If We use the relational model for storing the data we have to store the data of the student. using columns in a the student table naming it as stu_name, age, roll_no, etc. these
- We have to define all this at level by creating the database hence the blueprinter is created at conceptual level.

8 Dec 2022

3) physical level -
- The physical level tell's us that where the data is actualy stored
- The DBA decide that which data should be kept at which particular disk drive, how the data have to be fragmented, where it has be stored etc.

Data Base admin

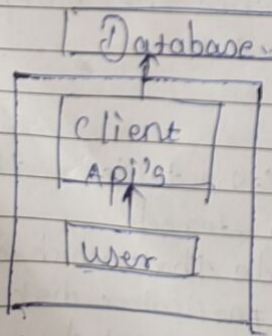
Api
- Application interface

- They also decide if data has to be centralised or distributed hence it depends on a DBA how he or she manages the database at the physical level.

* DBMS Architecture { 1 Tier, 2 Tier, 3 Tier }

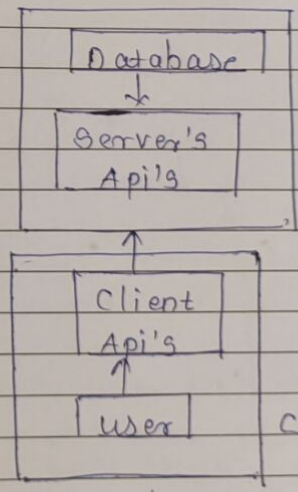
- It depends on how users are connected to the database to get their request done.
- There are 3 types of DBMS Architecture
  ① 1 tier.
  - The database is directly available through the user.
  - It means that user can be directly handle the DBMS and used it.
  - It is use for development of the local application, where programers can directly communicate with the database for quick response.

② 2 tier
Diagram -



- It is same as the basic client sever Architecture.
- Here the client application & directly communicate with the data at the server side. For this interaction Api's like SQL, oracal, ODBC, JDBC
- The server side is responsible to provide the functionalities like queries processing & transaction managment.

③ 3 tier -
Diagram -



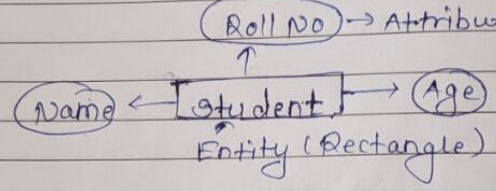
- It contains another layer between the client and server
- In this architecture, client can not be directly communicate with server.
- end user has not idea about the existence of the database beyond the application server
- The 3 tier architeture is used in case of large web application

chapter-2

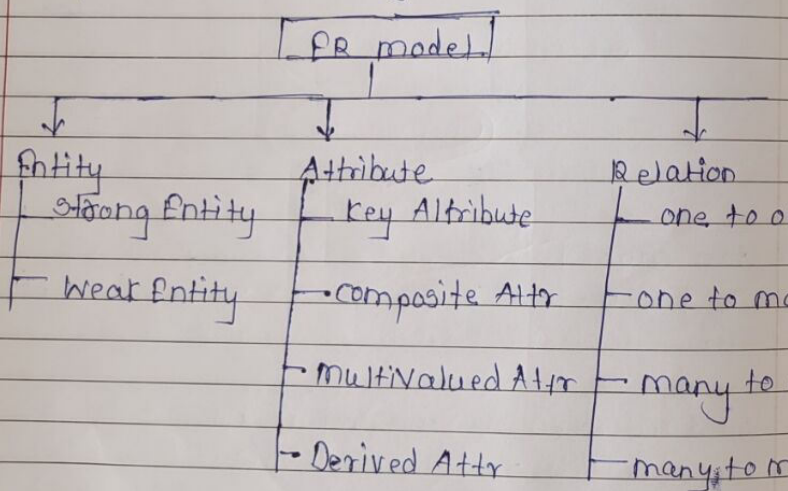
3 Dec2022 Entity-Relationship Model (ER Model)

- It is a high level data model
- This model is use to define the data elements, & relationship for a specified system.
- In ER modeling, the database structure is portrayed as a diagram called an ER diagram

fg :-



* Components of ER diagram.



* Entity -

- It is an object, class, person or place It can be represented in rectangles.
- There are 2 types of Entity

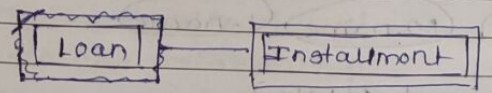
① Weak Entity

depend on another entity

eg - has money and trust

- An entity is depends on the another entity is called as weak entity. It doesn't contain any key Attributes on it own.
- It is represented using ▭ be double rectangle.

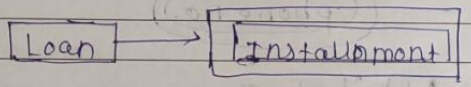
eg :-



② Strong entity.

- An entity is doesn't depends on the another entity is called as strong entity.
- It contain key Attributes on it own.
- It is represented using ▭ sigmas rectangle.

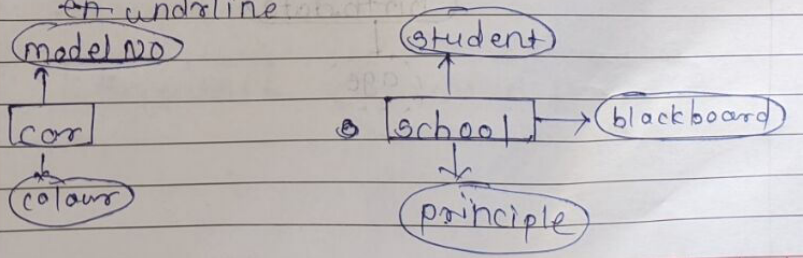
eg :-



3 Dec2022

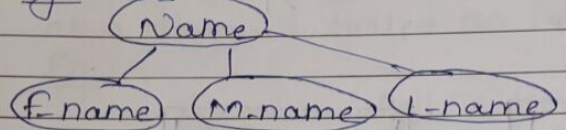
* Key Attributes -

- The Attribute is use to discribe the property of an entity
- elipse is used to represent the Attribute
- It is use to represent the main characteristic of an entity. it represents the primary key. It is represented by an elipse with the text underline

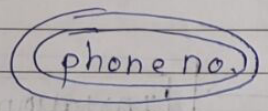


name ⇒ n

*2) Composite Attribute -
- The Attribute that composite of many other attributes is known as a composite Attribute.
eg:-

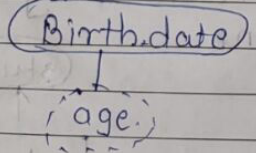


3) multivalued Attribute -
- An attribute can have more than 1 value are known as multivalued Attribute. it is represented using double oval.
eg:-
• A student can have more than 1 phone no.



1 person many mobile No

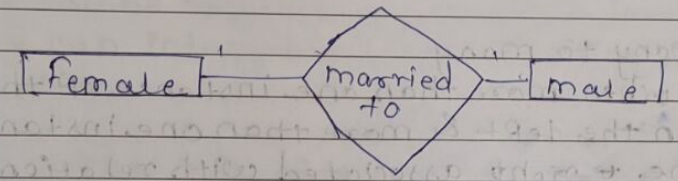
4) Derived Attribute -
- An attribute can be deived from other attribute it known as the derived attribute it is is represented as the dash elipse
eg:-
A person's age changes over time and can be derived from other attribute such as birthdate.



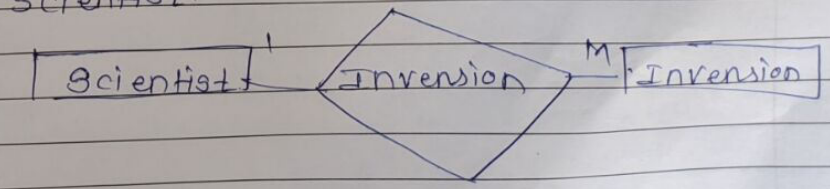
* Relationship -
- It is used to discribe the relation between entities, it is represented using ◇ diamonds
* Types *.

1) one to one - number
- When one instance of an entity is associated with the relationship there it is known as one to one relationship.
eg:-
A female can marry to one male, and male can marry to one female.

1 country
1 capital

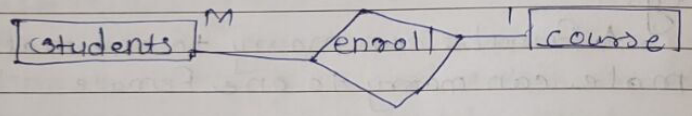


2) one to many
- When only one instance of the entity on the left ~~it's~~, & more than 1 instance of an entity on the right associated with the relationship then this is known as one to many
eg:-
Scientist is one and inventions are many but the invention is done by only specific scientist.



3) many to one - ~~more than~~ only
- When the only one instance of the entity on the right & more than instance of entity on the left is called as many to one relationship. associated with relation.

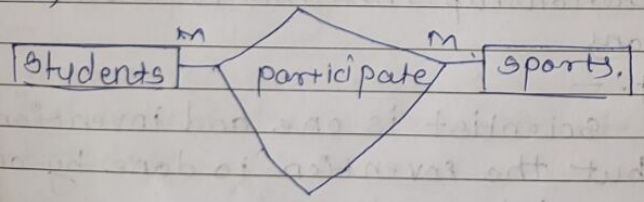
eg:-
student and ~~a~~ rolles for only one course but course can have many student.



4) many to many
- when more than one instance of the entity on the left & more than one instance on the ~~t~~ right associated with relation is called as many to many relationship.

eg:-
1) students ~~are~~ was participate in more than one sports.

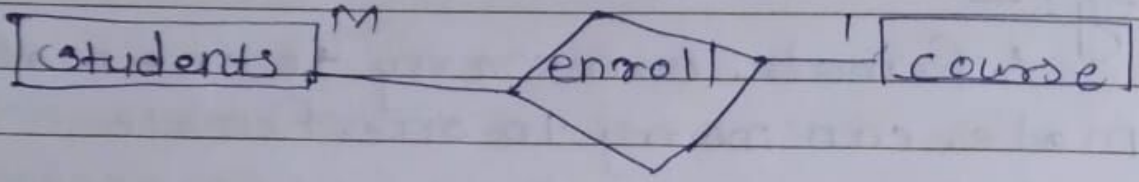
2)



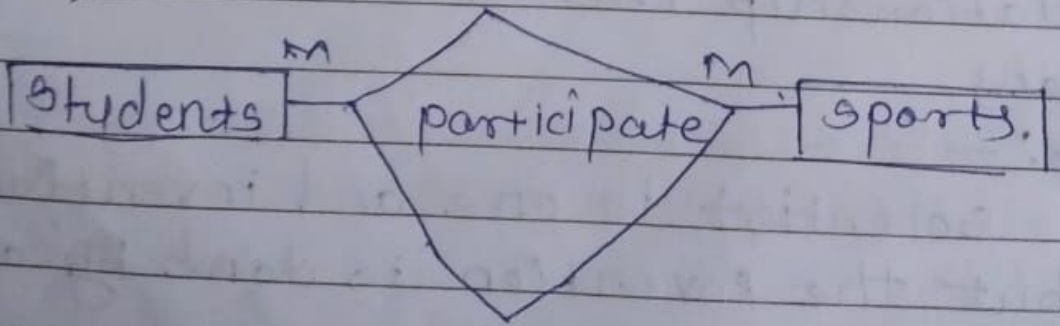
Diff entity vs Attribute

Entity	Attribute
1) An entity is a distinguishable real world object that exist	- It describes the elementary feature of an entity.
2) A row in a database table structure is an entity (Name, Roll, id,)	- ~~It~~, A column header of a database table is an attribute.
3) It can be tangible or can intangible entity.	- It is involed in forming a key.
4) Eg:- the info. of a signle student in a student table represents an entity	eg - name, roll No, etc; that forms the complete info. of ~~the~~ a student represent attribute.

3) many to one -
- When the ~~more than~~ only one instance of the entity on the right & more than instance of entity on the left associated with relation is called as many to one relationship.
eg:-
student and ~~de~~Rolles for only one course but course can have many student.



4) many to many
- when more than one instance of the entity on the left & more than one instance on the right associated with relation is called as many to many relationship.
eg:-
1) students ~~are~~ was participate in more than one sports.
2)



BARFI | Page No. | Date

Diff entity vs Attribute

Entity	Attribute.
1) An entity is a distinguishable real world object that exist	- It discribes the elementary feature of an entity.
2) A row in a database table structure is an entity	- It, A column header of a database table is an attribute.
3) It can be tangible or can intangible entity.	- It is involed in forming a key.
4) Eg:- the info. of a signle student in a student table represents an entity	eg - name, roll No, etc; that forms the complete info. of a student represents attribute.