ECAP200 :-

DBMS

CH-1 :-

All the data from the multiple DB is get stored in particular data storage that is called DB - Centro repository

Data is generated from every transcation in the org.
Users can fetch data from that particular Data storage for the purpose of decision making, monitoring the organisation & can also input the data

* Database :- A collection of related data stored in a std. format, designed to be shared by multiple users.
A collection of interrelated data items that can be processed by one or more appl^n programs.

→ Feeding into DB, data cannot be directly taken from heterogenous sources & feeding into that particular DB. That DB need to be Pre - processed.
That process is called as ETL :- Extraction, Transform & Lodding

Extracting the data from multiple sources & put it into 1 particular Central repository. T
Transforming the external & internal data into a particular format
Process the data & put it into central repository.

Dataprocessing The process of converting the facts is into meaningful info.

* MetaData :- Data that describes the properties or characteristics of other data.

⇒ Purpose of DB :-

- A DB is typically designed so that it is easy to store & access info.
- A good DB is crucial to any company or org.
- This is because the DB stores all the imp. details about the company such as employee records, transactional records, salary details.
  In an org. 3 types of people - operational level → managarial level
  ↓
  technical level
- DBMS are essential for business because they offer an efficient way of handling large amounts & multiple types of data.
- The ability to access data efficiently allows companies to make informed decisions quicker

Why DB?

- Redudancy can be reduced → duplicacy of data
- Inconsistency can be avoided
- The data can be shared
- Std. can be enforced
- Security restrictions can be applied.
- Integrity can be maintained.
- Provision of data independence
  if there is updation in one level it will not affect the other level

In DB there are various constraints, restrictions, limitation that can be applied on content that once the content is returned. If it is not written return required it will not be return again

Inconsistency :- if data is entered at one particular DB at one particular place At other place the same data will appear if there is no updation done on that.

⇒ Files Vs DB:

File:- A collection of records or docs. dealing with one org., person, area or subject (Manual (paper) files. e.g:- Comp. files.

DB:- A collection of similar records with relationships b/w the records. e.g:- bibliographic, statistical, business data, images etc.

With the advent of DB systems, the file processing approach is no longer used. Problems with the file processing systems :-

- Catalog
- Program - data independence
- Views

⇒ Benefits of DBMS:

- Minimal data redundancy
- Consitency of data
- Integration of data
- Sharing of data
- Ease of Appl^n development
- Uniform security, privacy & integrity controls
- Data accessiblity & responsiveness
- Reduced program maintenance.

⇒ DB Appl^n:

- Mahindra & Mahindra org. collecting data from DTO & feeding into DB for purpose of decision making.
- Disney uses Magic bands to collect data for purpose operational importance

DTO :- District Transport Office.

RFID Chip:- Radio frequency Indentification Device

- Banking :- For customers info. accounts & loans, & banking transactions.
- Airlines :- For reservations & schedule info.
- Universities :- • Credit card transactions :- • Telecommun^n :
- Finance.

- Finance :- For storing info. about holdings, sales & purchase of financial instruments such as stocks & bonds.
- Sales:
  - Manufacturing :-
  - Human resources :- (pay checks, payroll taxes)

DB :- A safekeeping of logically related & similar data.
An organized collection of related info. so that it can easily be accessed, managed & updated.

DBMS :- Set of programs to access interrelated data. It contains info. about a particular enterprise. Computerized record keeping system.

DB Users :- ~~Naive Users; App~~
- Neive Users :- New users
- Appl^n Programs :- Those who are expert in DB. Want to ~~store~~ use DB for creating an appl^n.
- Specialized Users :- Analyze the data → provide the hidden info. to their managers which was previously unknown & potentially useful for the org.

⇒ DB Administrator (DBA) :-

- Schema definition
- Storage str. & access-method definition
- Schema & physical-org. modification

- Granting of authorization ~~of~~ for data access
- Routine maintenance
- DBA manages all level of DBMS model

⇒ 3-Level Architecture :-

This framework is used for describing the str. of specific DB system.
In this architecture the DB schemas can be defined at 3 levels ~~explained in~~ Physical Schema → Logical Schema → View level.

- Physical Schema also known as internal schema near to DB
- Logical Schema : All the commands & logics are written related to SQL
- View Level :- The end user will see that particular level.

also known as external level. models a user-oriented description of part of the DB.

→ Deals with modeling of the whole DB
→ Show how data are stored inside system. #

→ Highest level of data abstraction (No knowledge of DBMS S/W & H/W or physical storage).
Describes the part of the DB that a particular user is interested in & hides the rest of the DB from user.

- There is only one conceptual view for single DB. It hides the details of physical storage str. & concentrates on describing entities, data types, relationships, user op. & constraints

- Internal View :- It keeps the info. about the actual representation of entire DB. The actual storage of data on the disk in the form of records or blocks

Tells what data is stored in DB & how

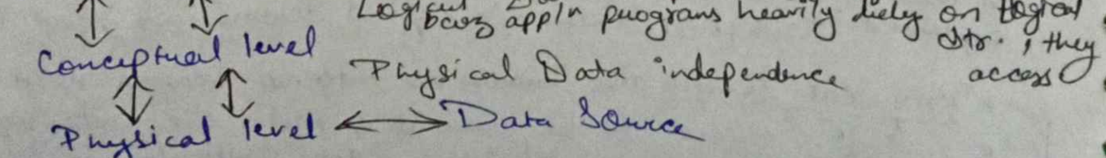
Atleast the following aspects are considered at this level :-
Storage allocation, Access Paths etc.

⇒ Objectives of 3-level architecture :-

- Each user should be able to access the same data but have a diff. customize data view of the data.
- User should not have to deal directly with physical DB storage detail.
- The DBA should be able to change the DB storage str. without affecting the user views.
- The internal str. of the DB should be unaffected by changes to the physical aspects of storage.
- The DBA should be able to change the conceptual str. of the DB without affecting all users.

⇒ Mapping :- The key for providing data independence.

Data independence is the capacity to change the schema at one level without having to change the schema at the next higher level.

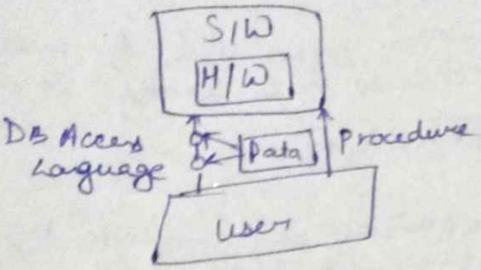

External level
Conceptual level
Physical level
Logical Data independence (Difficult)
bcoz appl^n programs heavily dely on logical str., they access
Physical Data independence
Data Source

Logical data independence :- ~~provides~~ Ability to modify conceptual schema without changing :- External views ; appl^n programs

Physical data independence :- Ability to modify internal schema without changing :- Conceptual schema ; Appl^n programs ; Changes to physical schema may be necessary to improve performance of retrieval or update

Components of DBMS :-

- H/W
- S/W
- Data
- Procedures
- DB Access Languages



<u>H/W</u> :- Comp., HD, I/O Channels for data, & any other physical component involved before any data is successfully stored into the memory.

<u>S/W</u> :- The program which controls everything. The DBMS S/W is more like a wrapper around the physical DB, which provides us with an easy to use interface to store, access & update data.

<u>Data</u> :- That resource, for which DBMS was designed. The motive to create DBMS was to store & utilize data.

<u>Procedures</u> :- General instr. to use a DBMS. This includes procedure to setup & install a DBMS, To login & logout of DBMS s/w, to manage DB, to take backups, generating reports etc

<u>DB Access Language</u> :- a simple language designed to write commands to access, insert, update & delete data stored in any DB.

⇒ Purpose of Data Model :-

To represent data & to make the data understandable.

⇒ Categories of Data Models :-

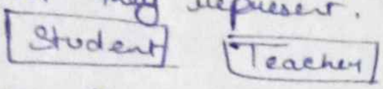
- <u>Object based data models</u> :- Used to describe data at the conceptual & external levels, the physical data model is used to describe data at internal level. Use concepts such as <u>entities</u>, <u>attributes</u> & <u>relationships</u>.

i) <u>Entity - Relationship</u> :- Has emerged as one of the main techniques for modelling DB design & forms the basis for the DB design methodology

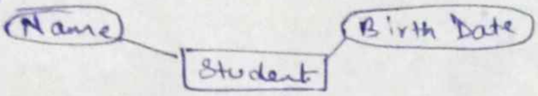
ii) <u>Object oriented</u> :- extends the definition of entity to include, not only the attribute that describes the state of the obj.

but also the action that are associated with the obj. that is its behaviour.

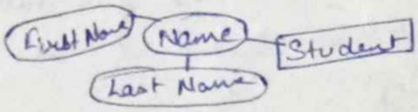
Entity:- are represented by means of □. □ are named with the entity set they represent.



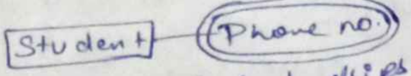
Attributes:- the properties of entities. represented by means of ellipses. Every ellipse represents one attribute & is directly connected to its entity □



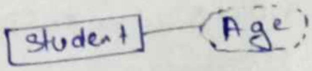
Composite attributes:- If the attributes are composite they are further divided in a tree like str.
Meta data



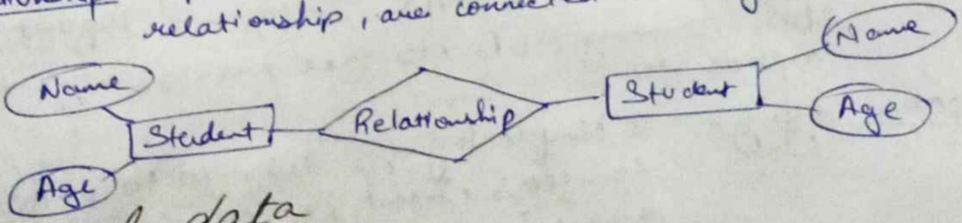
Multivalued attributes:- are depicted by double ellipse



Derived attributes:- depicted by dashed ellipse



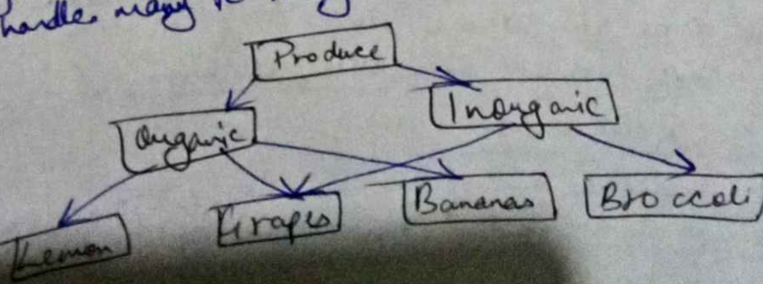
Relationship:- represented by ◇ All the entities participating in a relationship, are connected to it by a line.



• Physical data ~~Record based~~ Model

Hierarchical DB Model:- one of the oldest DB models. It is like a str. of tree with the records forming the nodes & fields forming the branches of tree.

Network DB Model:- replaces the hierarchical tree with a graph thus allowing more general connections among the nodes. It has the ability to handle many to many (N:N) relations



Relational Model : Stores data in the form of tables.

The set of relations & set of domains that defines the way data can be represented (data str.)

Integrity rules that define the procedure to protect the data (data integrity)

The opr. that can be performed on data (data manipulation)

Domain :- Set of all possible value that we can enter in that column is

Advantages :-

- Simple :- This model is simpler as compared to the network & hierarchical model.
- Scalable :- This model can be easily scaled as we can add as many rows & columns we want.
- Structural Independence :- We can make changes in DB str. without changing the way to access the data.

Which Model to Use?

A model that best suits an org. depends on :-

- The org.'s pri. goals & requirements.
- The vol. of daily transcations that will be done.
- The estimated no. of enquires that will be made by the org.

~~Data Independence~~

Str. ~~level~~ data Independence :-

Record based Model :- Relational Model

Database Constraints :- Restrictions apply on DB

- Pri. Key Constraints :- also called a pri. keyword, is a key in a ~~relati~~ RDB that is unique for each record.

  It is a unique identifier, ~~such as a d~~

  VIN :- Vechile Identification No.

  e.g :- CREATE TABLE emp (Id Varchar (90) Primary key, Name Varchar (90));

  Does not accept duplicate or null value. There is always one pri. key in a table.

- Unique ~~Key~~ Constraint :- Use for unique values. There can be many unique constraints in a table. It accepts NULL values.

  eg :- CREATE TABLE emp (Id varchar (90) Primary key, Atar Phone no. int Unique);

- Foreign Key Constraints :- Attribute of one table which is pri. key of another table. It is a key used to link 2 tables together. also called referencing key. It is a column or a combination of columns whose Values match a pri. Key in a different table.

  This key is used for security. It prevents invalid data from being inserted into foreign key column, because it has to be one of the values contained in the table it points to.

- Check Constraint :

  e.g :- CREATE TABLE emp (Id varchar (90), age int ~~k~~ check (age > 18));

  Can be applied to any attribute.

  Used to limit the value range that can be placed in a column.

  If it defined on a table, it can limit the values in certain columns based on values in other columns in the row.