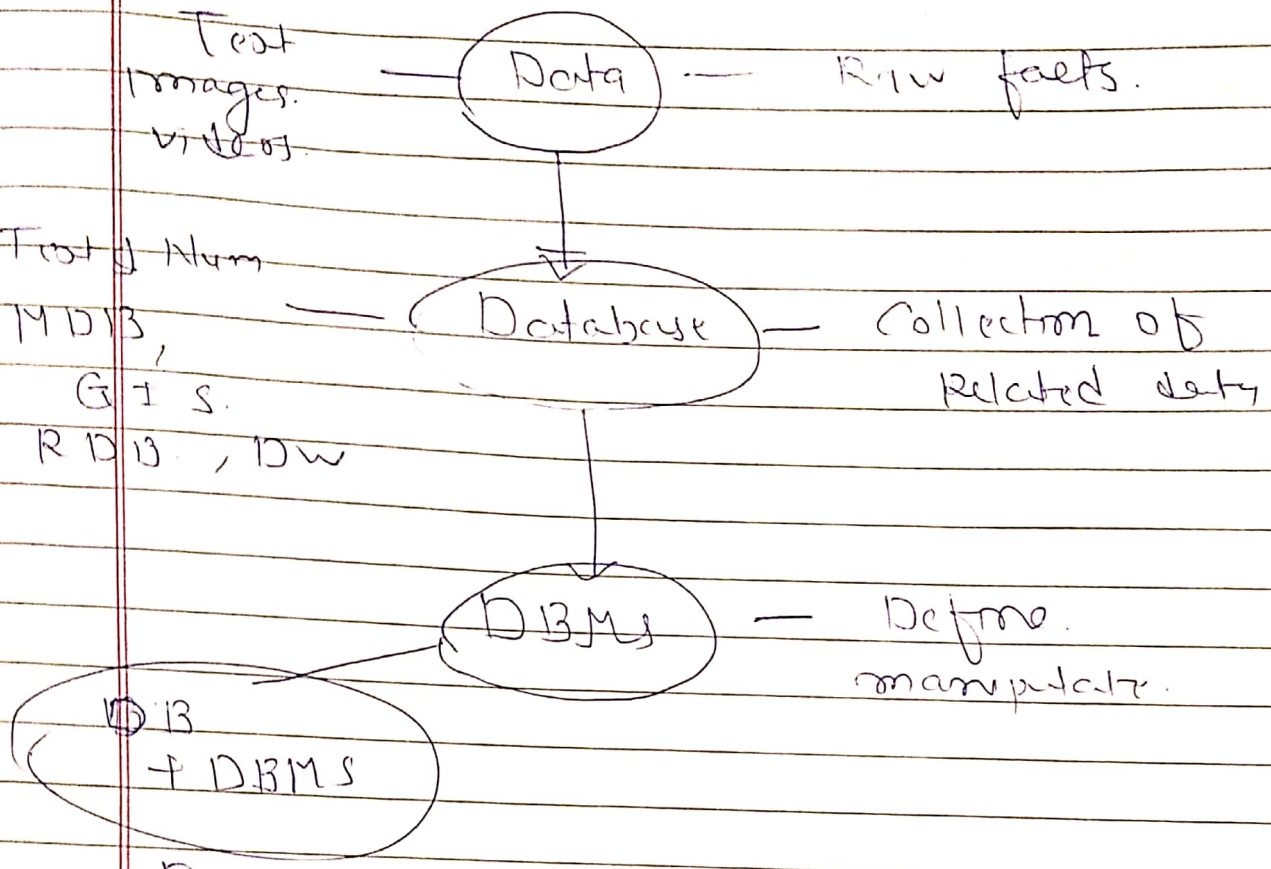
DBMS



Applications

Banking.
Airline.
Universities,
Credit Card.
Telecommunication
frames.
Sales.
Online Retailing.
Manufacturing.
Human Resources

(*) View of Data

(1) Higher level or Conceptual level - ER Diagram

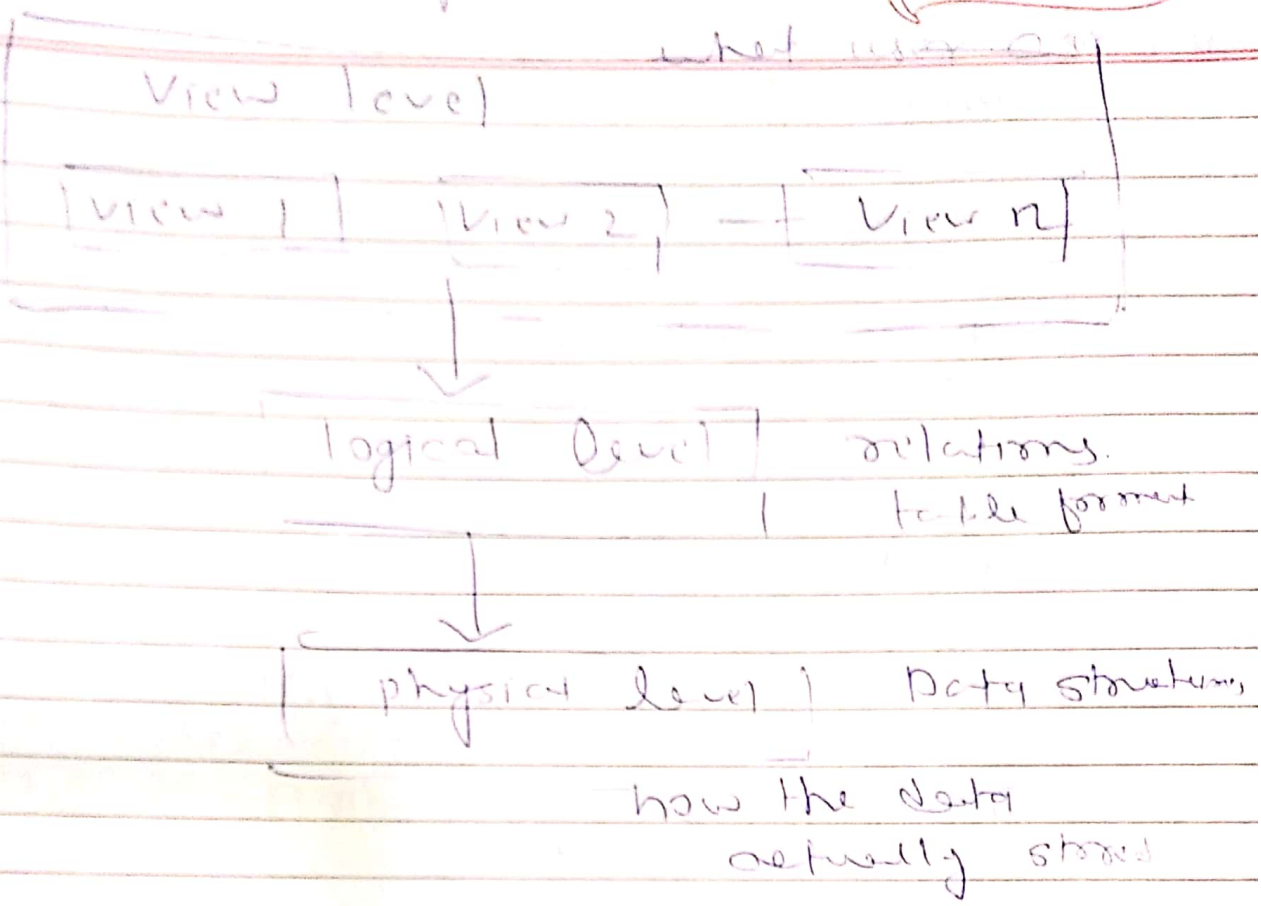
(2) Representational or implementation logical level.

⇒ What data are stored in database
⇒ Relations among those data
⇒ Decide what info to keep

(3) low level - Physical level

How the data are actually stored

Complex low level Data structures



(x) Data models

↳ structure of database

Collection of tables to represent both data & relationships.
multiple columns & each column is unique.

Record based models.

(?) ER model

Entity - Relationship

 – Double Ellipse

Represent multivalued Attribute

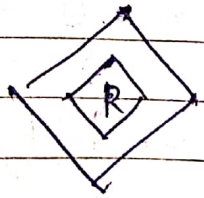
Dashed Ellipse : 

└ Derived Attribute

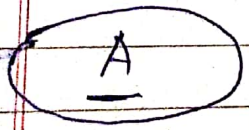
 – represent weak entity set.

(*) Weak Entity Set :- An entity which do not have sufficient attributes to form a primary key is known as weak entity set.
Denoted by double Rectangles as above

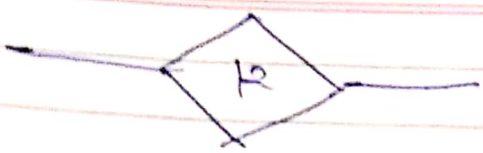
(*) An entity set has a primary key is termed as strong entity set.



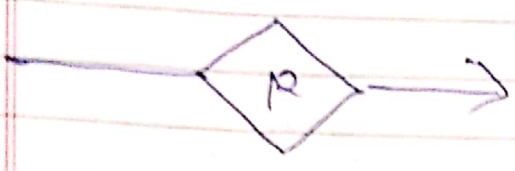
– Identifying relationship for weak entity set.



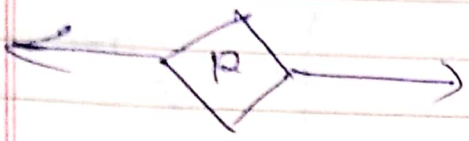
– Primary key



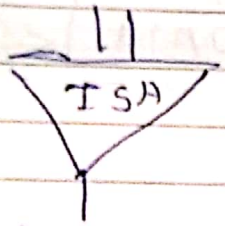
many to many relationship.



many to one relationship



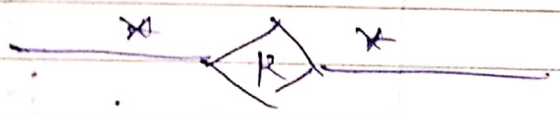
One to one relationship.

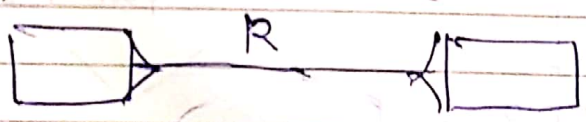


total generalization

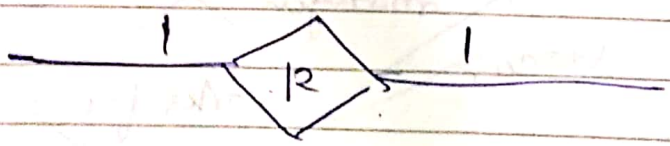
→ Alternative Notation.

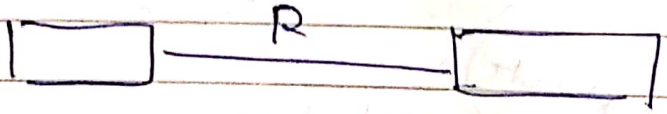
many to many



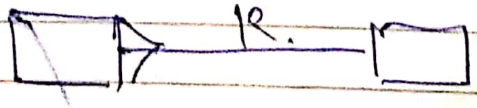


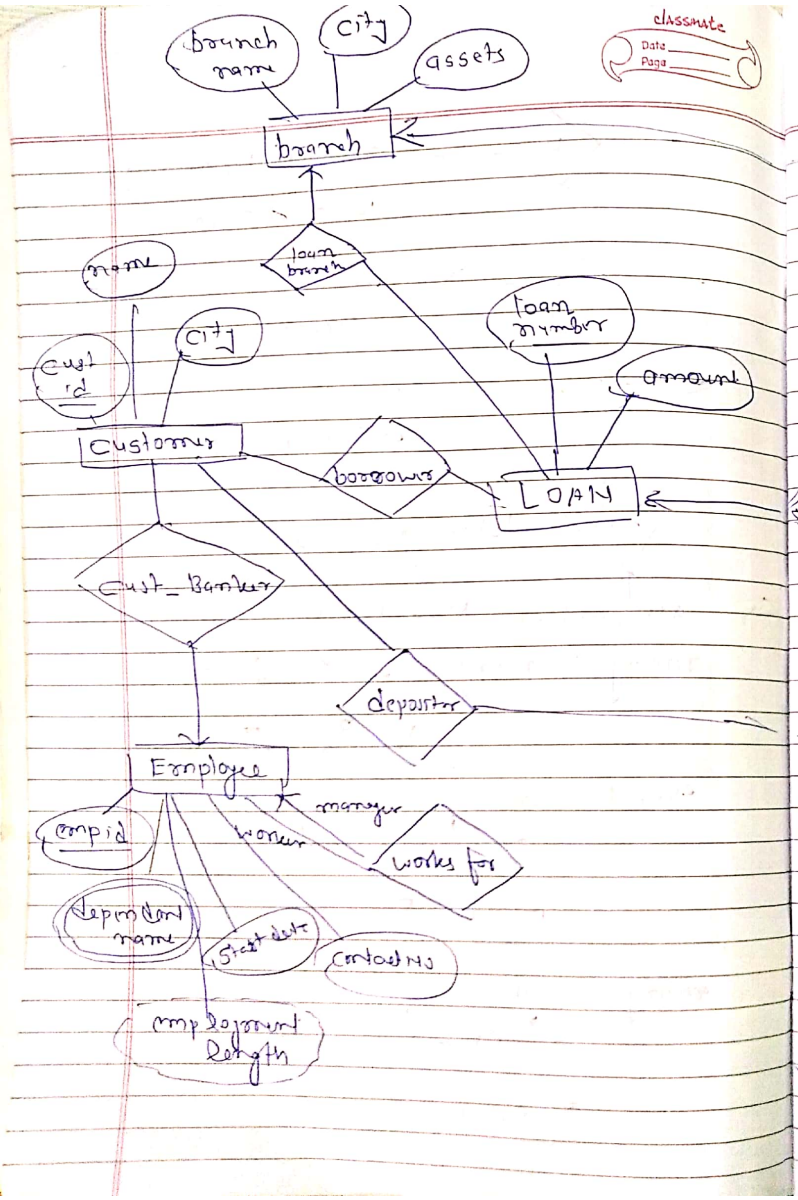
One to One

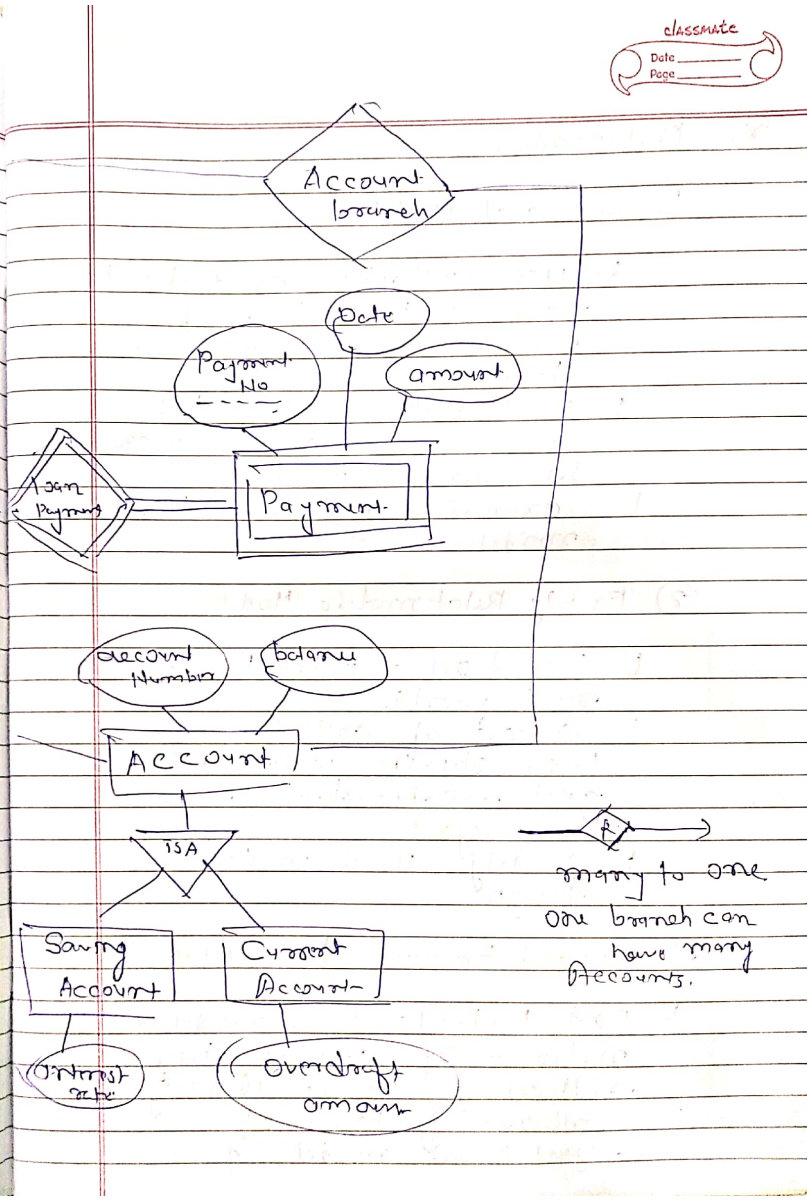




many to one







(X) Datamodels

(1) Relational Model

- Uses collection of tables to represent both data & relationships among those data
- Each table has multiple columns & each column has unique name
- example of record based model.

(2) Entity Relationship Model.

- Based on a perception of real world.
- Consist of collection of basic objects called entities and relationships among those objects.
- Widely used in database design.

(3) Object based data model.

- Extends E-R model with notions of encapsulation methods & object identity
- Combines features of object based model &

relational model in object based relational model.

(4) Semistructured data model.
- permits the specification of data where individual data items of the same type may have different sets of attributes
- XML (Extensible Markup language) is widely used to represent semistructured data.

(*) Datamodel Languages.

(1) DML - Data Manipulation Language.

→ DML is a language that enables user to access or manipulate data as organized by the appropriate data model.

e.g.
- INSERT - Insert data into the table
- DELETE - Delete data from table
- UPDATE - Update existing data within a table

→ Retrieval of information stored into the database is done by DML.

Types of DML :-

① Procedural DMLs requires user to specify what data are needed and how to get those data.

② Declarative DMLs.
- also referred as non procedural DMLs.
- require a user to specify what data are needed without specifying how to get those data.

(*) Query : is a statement requesting the retrieval of information.

The portion of DML that involves information retrieval is known as query language.

DaL

SELECT Statement

classmate
Date
Page

(2) DDL - Data Definition Language

Database schema by a set of definitions expressed by special language called a DDL.

- used to specify additional properties of data.
- We can specify storage structure & access methods used by database system by a set of statements in DDL.

Examples:

- CREATE - to create objects in the database.
- ALTER - Alter the structure of database.
- DROP - Delete object from DB.
- TRUNCATE - Remove all the records from a table (object) including memory allocated.
- COMMENT - Add comment to data dictionary

③ DCL - Data Control Language
control access to data stored in DB

⇒ COMMIT - Save the work done

⇒ SAVE POINT - Identify the point in a transaction to which you can later roll back

⇒ ROLLBACK : Restore database to original since COMMIT.

⇒ SET TRANSACTION
↳ Change the transaction options like what rollback segment to use.

⇒ Grant & Revoke

Grants and take back permissions to or from the users of DB.

④ DQL : Data Query Language

SELECT :- Retrieve data from the Database.

(*) Purpose Database systems.
OR.
Disadvantages of file management System.

(1) Data redundancy & Inconsistency.

→ Since different programmers create the files & the application programs over a long period the various files likely to have different structures and programs may be written in different programming languages

→ Same information may be duplicated in several places.

→ Duplication of information in various places causes data redundancy this redundancy leads to higher storage & access cost. This may leads to data inconsistency.

→ Data inconsistency is, the various copy of same data may no longer agree

E.g. change in address of customer may be reflected in saving account record but not else where in the system

② Difficulty in accessing data

→ Conventional file system do not allow needed data to be retrieved in convinient & efficient manner.

→ more responsive data retrieval systems are required.

③ Data isolation

→ As data are scattered in various files, & files may be different formats writing new application programs to retrive the appropriate data is difficult.

④ Integrity problems.

→ Database must satisfy some consistency constraints.
→ Ex. e.g. balance in certain type of account must not fall below specified limit
→ Developers enforce this constraints to the system.

(5) Atomicity Problems

atomic means it must happen in its entirely or not at all

It is difficult to ensure atomicity in conventional file system.

(6) Concurrent access anomalies.

In multiuser environment, interaction of concurrent updates is possible and may result in inconsistent data.

(7) Security Problems

Not every user of database system should be able to access all the data Since application programs are added to the file processing system in an adhoc manner, enforcing a security constraint is difficult.

(4) Integrity Constraints..

(1) Domain Constraints.

→ A domain of possible values must be associated with every attribute
e.g. int, char, date datatypes

→ Declaring an attribute to be of particular domain acts as a constraint on the value that it can take.

→ They are tested easily by the system whenever a new data item is entered into database.

(2) Referencial Integrity :-

There are cases where we wish to ensure that a value that appears in certain relation for a given set of attributes also appears for a certain set of attributes in other relation

Database modification can cause violation of referential integrity

→ Whenever the referencial integrity is violated, the normal procedure is to reject the action that

caused the violation.

③ Assertions

→ An assertion is any condition that the database must always satisfy

→ Domain constraints & referential integrity constraints are special form of assertions.

④ Authorization

→ We may want to differentiate among the user as far as the type of access they are permitted on several data values in the database.

→ These differentiations are expressed in terms of authorization.

such.

read authorization

Write "

Insert "

update "

delete. "

Constraints

An E-R enterprise schema may define certain constraints to which the content of a database must conform

↳ Mapping Cardinalities.
↳ key constraints.
↳ participation constraints.

Mapping Cardinalities or Cardinality ratio

⇒ express the number of entities to which another entity can be associated via a relationship set.

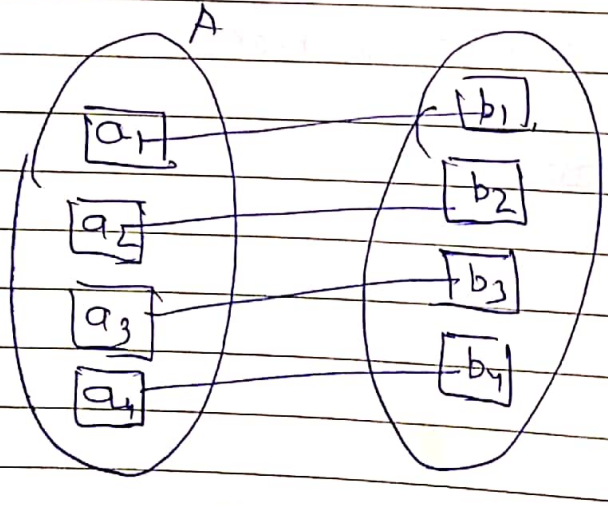


fig. One to one cardinality

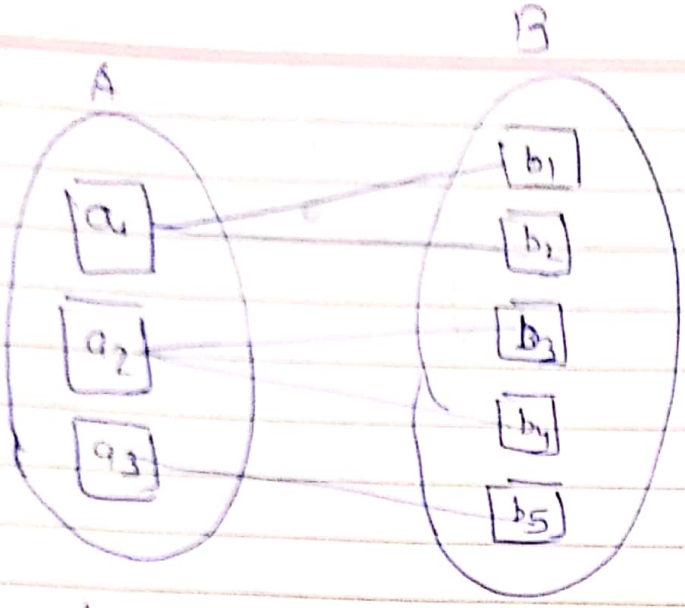


fig. One to many cardinality.

→ for a binary relationship set R. between entity sets A & B, the mapping cardinality must be must be one of following.

① One to One.
An entity A is associated with at most one entity in B. and an entity B is associated with atmost one entity in A.
country — capital. city → pincode.

② One to many.
An entity A is associated with many number of entities in B however B can be associated with only one entity in A.
book → pages.

③ Many to One.
A is associated with atmost one entity in B. however