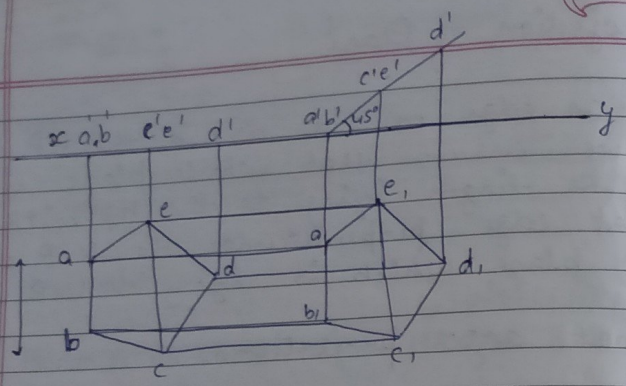
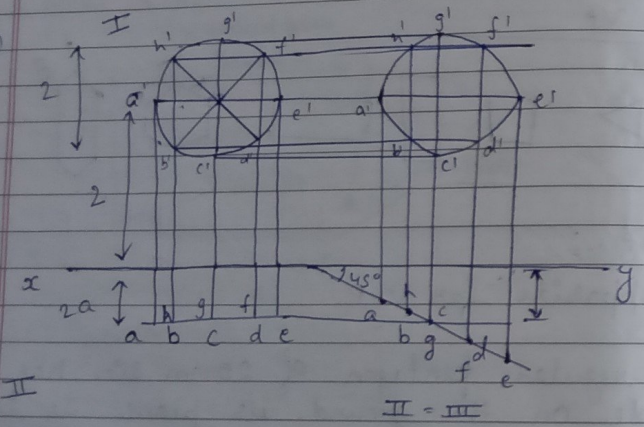




24/02/22

Data - Raw fact that can be recorded in any form like image, audio, speech.

Database System
- → Relational DBMS
- Database
  - Structured Data → Relations (Rows & columns)
  - Unstructured Data
- Database Management System (DBMS)
  - → Oracle
  - → SQL
  - → MySQL
  - → DB2

Database - It is a collection of interrelated data which is used to retrieve, insert, delete the data efficiently. It is also used to organize the data in the form of table, report etc

For example - university database organizes the data about students, faculty, admin, staff etc.

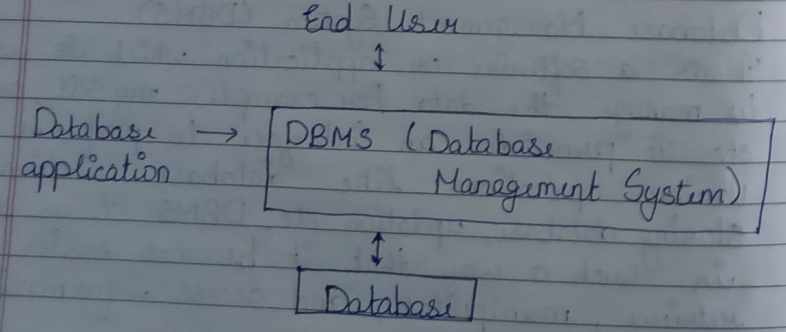
Database Management System (DBMS)
It is a software or application which is used to manage the data. For example - mySQL, oracle, etc. It provides an interface to perform various operations like database creation, storing database, updation etc. DBMS stores data in such a way that it becomes easier to retrieve, manipulate, and access information.

DBMS Application -

1. Banking - all transactions.
2. Airlines - reservation.
3. Universities - registration, grades
4. Sale
5. Human resource - employe, record, salary tax

File System

- It allows access to single files or table.
- Data is directly stored in set of files.
- It contains file that has no relation with other files.
- File system consist of different files which are grouped in two directories. It contains flat files.
- It is a technique of arranging the files in storage medium like hard disk, dvd, pendrive, etc



Difference between filesystem and DBMS

Point	Filesystem	DBMS
Structure	Filesystem is a software that manage and organize the files in a storage medium within a computer.	It is a software for managing database.
Data redundancy	Ridundancy can be present.	No redundant data
Backup & recovery	Doesn't provide any backup and recovery if it is lost.	Provides backup & recovery of data even if it is lost.
Cost	less expensive	Higher cost than file System.
Security	It offers less security.	Offers high security.
Query processing	No efficient query processing.	You can easily query data in a database using sql language.

25/02/22

Rows → records
Column → Attributes

1. Data redundancy
Repetition of data or duplication of data is called as data redundancy.
Ex: customer address present in saving account file and customer address is also present in current account file even when same customer have saving and current account address is present at 2 places

2. Data Inconsistency
Data redundancy leads to greater problem it may lead to inconsistant data
example: costamer request to change the address for his account in the bank but address is updated in saving account not in current account so there is no match.

3. Data concurrency
permanant access to data means more than 1 user is accessing the same data at the same time difficulties will arise when changes made by one user get lost because of change made by other user DBMS provides locking system to stop search anomalies (difficulties)

* Advantages of DBMS

① Better data security
② Faster data access
③ Better decision making
④ Increased and user productivity
⑤ It is simple
⑥ Better data transfering
⑦ Minimized data inconsistancy

DBMS Architecture

→ 2 tire
→ 3 tire
DBMS design dipends upon its architecture. The basic client server architecture is used to deal basic client server architecture is used to deal with large number of PC, web servers and other components that are connected with networks. DBMS architecture dipends upon how user are connected to database to get their request done

Types of architecture

→ 2 tire (client / server)
→ 3 tire

→ 2 tire
The 2 tire architecture is similar to a basic client server model
The application at the client end directly communicates with the database at the server side

API like ODBC, JDBC are used for this interaction
The server side is responsible for providing query processing and transaction managements on the client side the user interface and application programs are run.
The application on the client-side establish a connection with the server side in order to communicate with the DBMS.

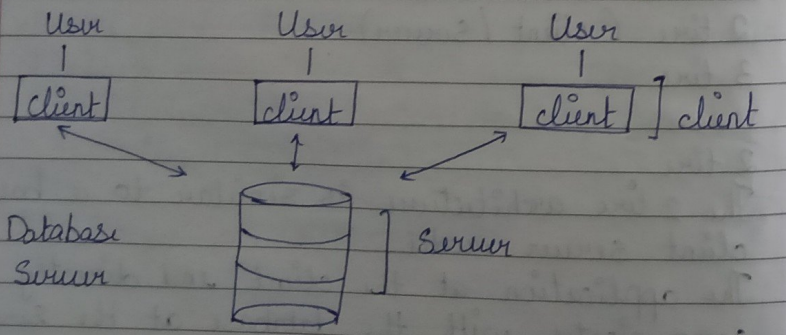
Advantage:

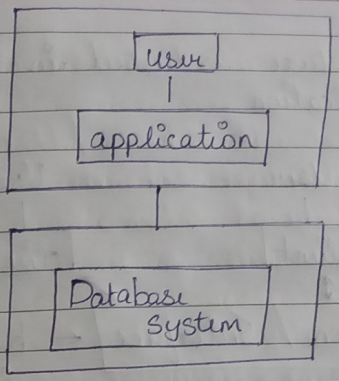
1. Maintenance
2. Compatible

Disadvantage:

1. Security
2. Scalability (no of user)

This model gives poor performance when there are large number of users.





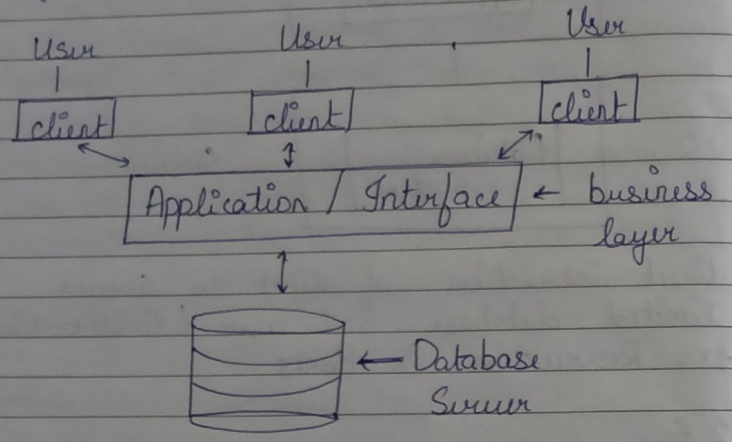
Short Points
→ direct interaction of client to server
→ limited database and user (limited time slot)
ex: Reservation in Banks

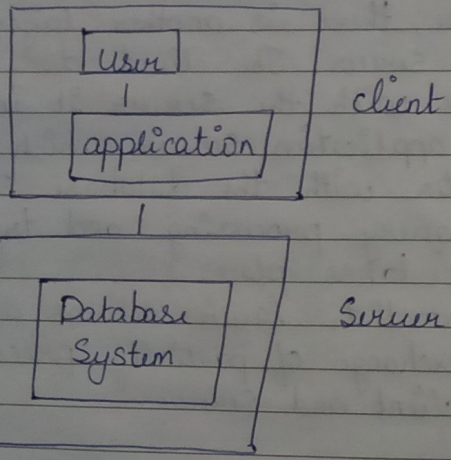
→ 3 tire
In this type, there is another layer between client and server. The client does not directly communicate with the server it interacts with an application server which further communicates with the database system and then the query processing and transaction management takes place.
This intermediate layer act as a medium for the exchange of partially processed data between client and server.

Advantage:

- Enhanced scalability
- Data integrity
- Security

The 3 tier architecture is used in the case of large web application





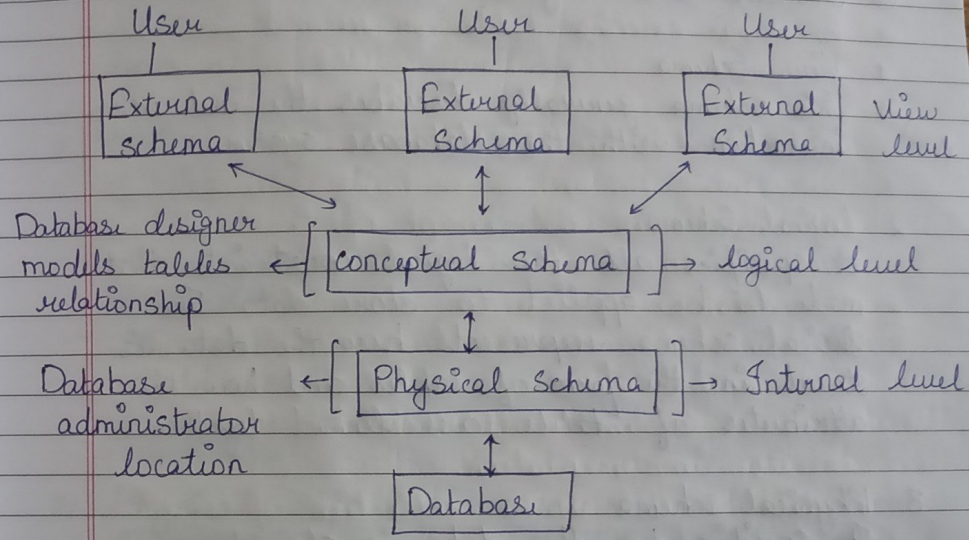
Short points
- → high maintainance (more man power)
- → high security than 2 tier
- → example: online reservations

03/03/22

DBMS Schema

- → logical structure
- → Blueprint

3 schema Architecture



A database schema is the logical representation of database which shows how the data is stored logically in the entire database. It contains list of attributes and instructions that inform the database engine that show the data is organized and how the elements are related to each other.

The schema does not physically contain the data itself it gives information about the shape of data and how it can be related to other table or models.

The database schema is divided into 3 types

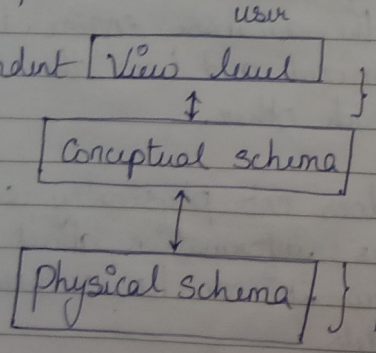
① View level / External schema / User schema
② Logical level / conceptual schema
③ Internal level / logical schema

1 View Schema
This schema generally describes the end user interaction with the database system.

2 Logical Schema
It specifies all the logical constraints that made to be applied to store data. The logical schema represents how the data is stored in the form of tables and how the attributes of table are linked together.

3 Physical schema
It specifies how the data is stored physically on a storage system or disk storage in the b form of files and indices

Data Independence



→ user independent of data
→ 24 x 7
→ location

} logical data independence

} physical data independence

Data independence can be explained using 3 schema architecture.
Data independence refer characteristic of being able to modify the schema at one level of the data base system without altering the schema at the next higher level. There are 2 types of data independence

1. Logical data independence - It refers to characteristics of being able to change the conceptual schema without having to change view level or external schema.
If we do any changes in the conceptual view of the data then the user view of the data would not be affected.
It occures at interface level

2. Physical data independence - It can be define as the capacity to change the physical schema without changing the conceptual schema.
If any changes in the storage size of the database system server then the conceptual structure of the database will not be affected
example of changes under physical data independence using a new storage device like hard-drive magnetic tapes, switching to different data structure
Modefing indexes
change location of database from C drive to B drive.

Example of changes under logical data independence.

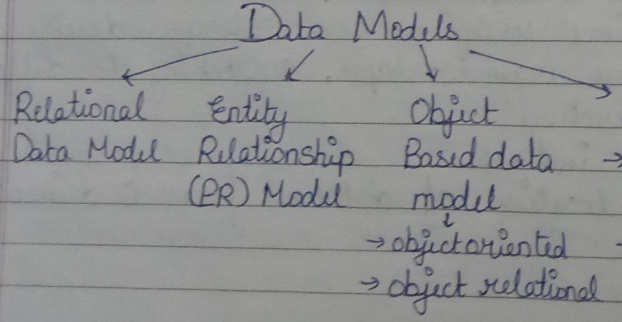
→ Add / modify / delete an attribute
→ Merging two records into one
→ Breaking existing table or record into two or more

04/03/22

Data Models in DBMS

Data model is the concept of tools that are developed to summarize the description of the database
It defines logical structure of database
"Data model gives us an idea that how the final system will look like after its complete implementation.
A data model is a collection of conceptual tools for describing

1. Data
2. Data relationship
3. Data semantics
4. consistency constraints.

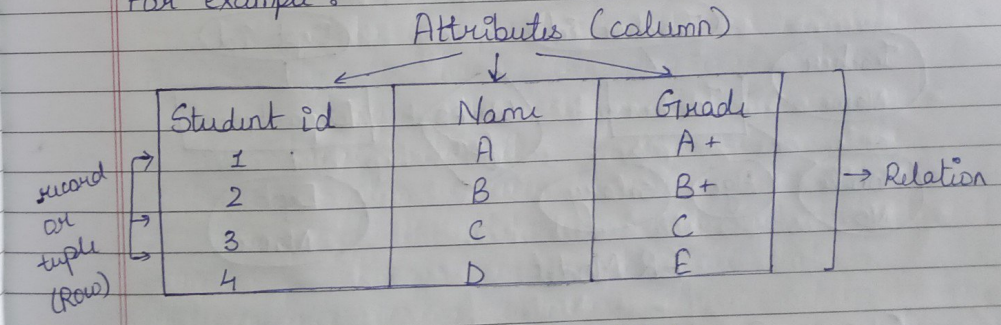


Relational Data Model

Most widely used model by commercial data processing applications.
It uses collection of tables for representing data and the relationship among those data.

- Data is stored in tables called relation.
- Each table is a group of column and rows where column equals to attributes and row equals to records / tupple.
- The value of the attribute should be from the same domain.
- A tupple defines a collection of attributes, value so, each row in a relation contains unique value.

For example :



Entity Relationship Model (ER)

- ER Model is a high level data model diagram
- It describes the structure of database with the help of diagram which is known as entity relationship diagram.

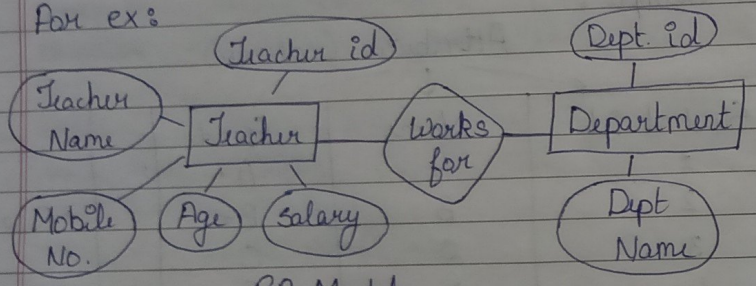
- It is based on real world entities and relationship among them.
- It has following three components :

→ Entity : It is a real world thing or object. It can be person, place, or even a concept.
ex: teachers, students, course, building etc.

→ Attribute : An entity contains a real world property called attributes (characteristic).

Property : Teacher, salary, Age etc.

→ Relationship : It tells how two attributes are related
For ex:



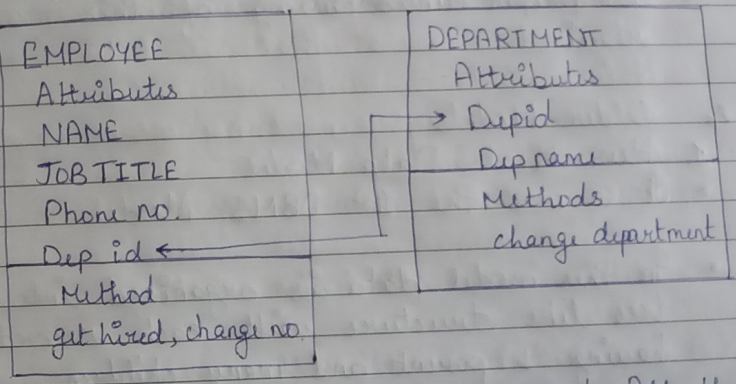
ER Model

Object Based Data Model

1. Object Oriented data model

An extention of ER model with notions of function, encapsulation and object identity is known as object oriented data model. In this model both data and relationship are present in a single structure known as object.

Two or more objects are connected through links, we use this link to relate one object to other object.



Here two objects employee and department. All the data and relationship of each object are contained as a single unit. The attributes like name, jobtitle & the methods which will be performed by that object are stored as a single object. The 2 objects are connected through a common attributes i.e. dep ID and the communication b/w these two will be done with the help of common dep ID.

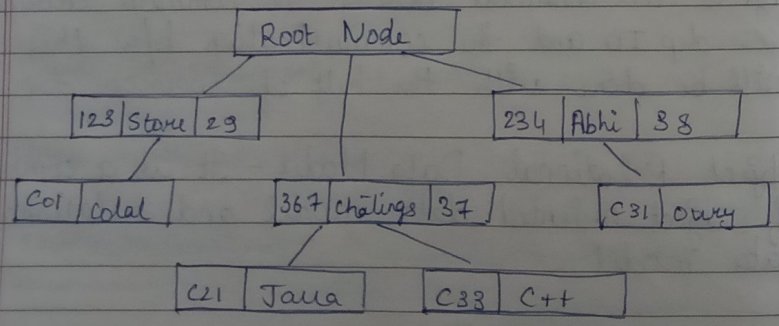
2. Object Relational Data Model - It is a combination of object oriented data model and relational data model.

- This model was build to fill the gap b/w object oriented model & Relational model
- It has many advanced features like complex data types
- The problem with this model is that it can get complex and difficult to handle so proper understanding of this model is required.

2. Older Models

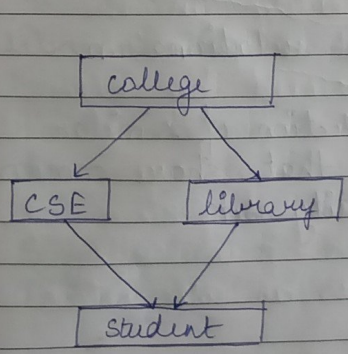
1 Hierachical Data Model

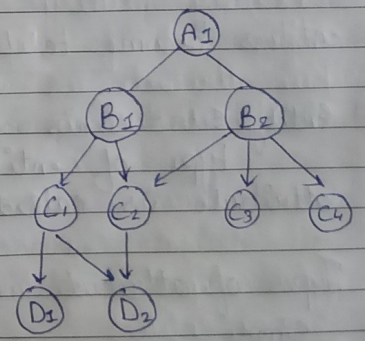
- It ~~co~~ was the 1st DBMS Model
- In this model data is organised into a tree like structure with each record having 1 parent and many children
- The main drawback is that it can have only one-to many relationship b/w nodes
- It is rarely used now.



2. Network Model

- This model is an extension of the hierarchical model.
- It was most popular model before relational model.
- It is same as hierarchical model except it has graph like structure rather than tree based structure and are allowed to have more than one parent node.
- It supports many to many data relationship
- This was the most widely used database model before relational data based model was introduced.



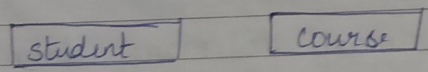


09/03/22

ER Model

1. It was introduced in 1976
2. ER Model defines the logical view of database
3. It is used for designing database
4. It works around real world entities and the relationship among them.
5. A database schema in the ER Model can be represented by ER diagrams.

* Entities → An entity is a thing or object in the real world that is different from other objects.
→ Entities are represented by means of rectangles
ex:



→ entities have attributes that give them their identity
ex: student have roll no, name and address

→ entities become table in relational model

* Entity Set → It is a collection of similar type of entities that share common attribute is called entity set

* Types of attribute

1. Simple attribute
2. Composite attribute
3. Simple & composite attribute
4. Single value & multi value attribute
5. Stored attribute
6. Derived attribute
7. Key attribute

1. Simple attributes are atomic values which can not be divided further
For example :→ Student's mobile no is an atomic value of 10 digit.
→ DOB. is an atomic value of date, month, year.

2. Composite attribute are made of more than 1 simple attribute.
It is divided into tree like structure
For example :→ A student's complete name may have first name, last name and middle name
→ Address may have city, state, country & pincode

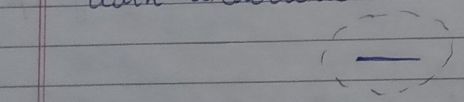
3. Single value attribute contain single value
For example : Adhar card no, roll no.

4. Multi vale → attribute which have multiple values.
ex: → Student mobile no., address, email

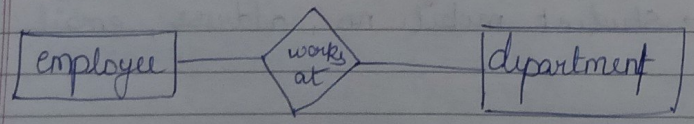
⑤ Store attribute - These attributes are physically stored in the data base. Mostly are attributes are stored in database except few one
Ex - phone no, DOB, name

⑥ Derived attribute - Derived attribute are the attribute that do not exist physical database but their values are derived from other attributes present in the database. Derived attributes are represent by dash ellipse
Ex: age can be derived from DOB, average salary in a department should not be same in the database it can be derived.

⑦ Key attribute - It is a attribute which uniquely identify each entity in the empty set if represent a primary key. Key attribute is represented by ellipse with underlines line



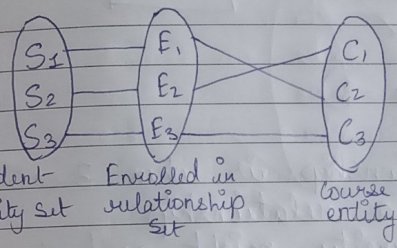
Relationship A relationship is an association among entities
Ex:



Here works at is a relationship.

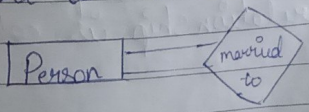
Relationship set: A set of relationship of similar type is called relationship set

The following relationship set enrolls (E_1, E_2, E_3) dipits.



Degree of Relationship
The no. of different entity set participating in a relationship set is called Degree of Relationship
1) Unary 2) Binary 3) Ternary
4) N-Ary

① Unary Relationship (degree = 1)
A unary relationship is only one entity participate in a relationship
Ex:

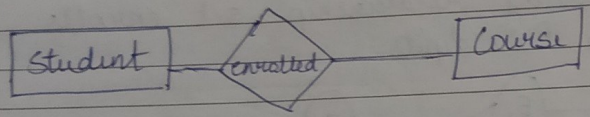


one person is married to only one person

② Binary relationship (degree = 2)

A binary relationship is when 2 entities participate in a relationship & this is the most common relationship degree

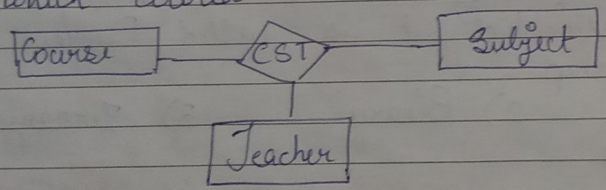
Ex: Student is enrolled in course



③ Ternary (degree = 3)

A ternary relationship is when 3 entities participated in a relationship

Ex: The university might need to record which teacher teaches which sub in which course



④ N-Ary (degree = n)

When there are n entities set participated in a relation the relationship is called N-Ary relationship

Mapping Cardinalities

→ One-to-one (1-1)

→ One-to-many (1-M/N)

→ Many-to-many (M-M) or (N/N)

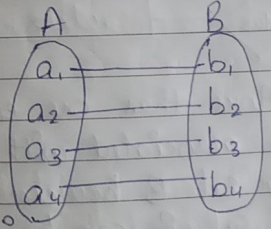
→ Many-to-one (M-1)

Cardinatry defines the number of entity of an entity set participate in a relationship set

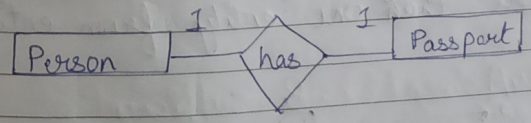
Most useful in discribing binary relationship.

* Cardinantry can be characterised in 4 types

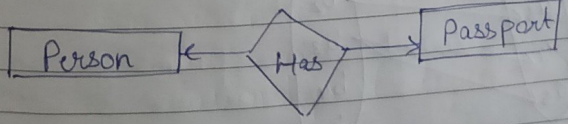
① One-to-one relationship → one entity from entity set A can be associated with atmost one entity of entity set B and visa versa



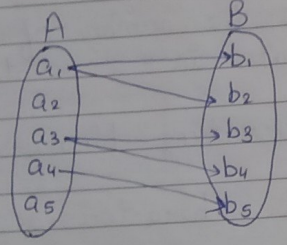
Representation

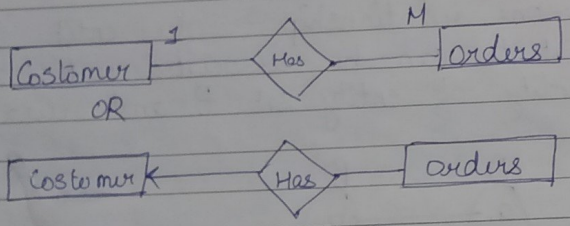


OR

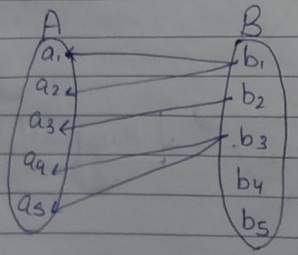


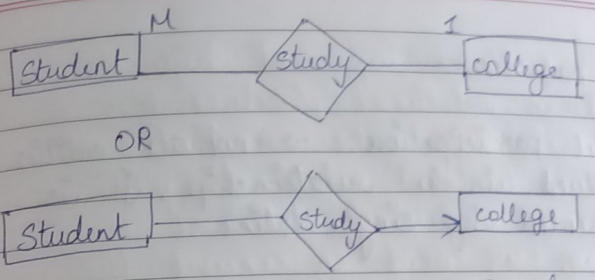
② One-to-many → One entity set from entity set A can be associated with more then one entities of entity set B however an entity from entity set B can be associated with almost one entity.



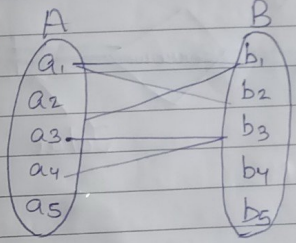


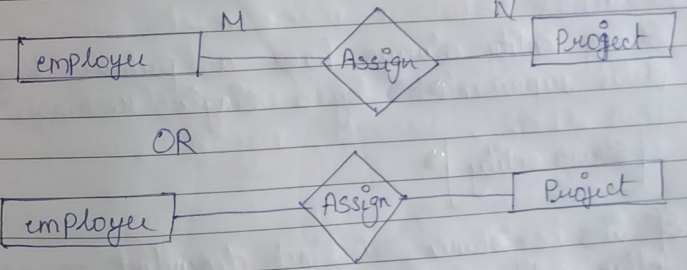
③ Many-to-one → More than one entity from entity set A can be associated with almost one entity of entity set B however an entity from entity set B can be associated with more than one entity from entity set A





④ Many-to-many → One entity from A can be associated with more than one entity from B and visa versa.





★ Participation constraints

Total participation → each entity is involved in the relationship total participation is represented by double lines

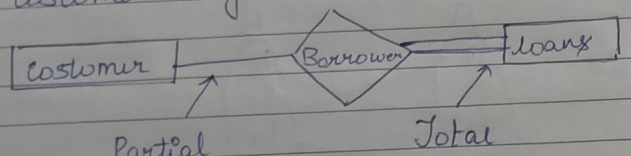
Ex: participation of loan in borrow one is total 1

every loan must have a costomer to it

by borrower

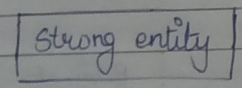
Partial participation → not all entities are involved in the relationship. It is represented by single line

Ex: participation of costumer in borrower is partial
A costomer may have no loans.



Entity Type

→ Strong entity always have a primary key. It is represented by a single rectangle

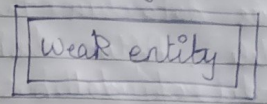


If its existance is not dipendent on any other entity. It is indipendent of other entity.

→ Weak entity doesn't have sufficient attributes to form a primary key

Weak entity doesn't have primary key.

A weak entity is dipendent on a strong entity. It is represented by double rectangle



UNIT ⇒ 2

Relational Model

Relational Model represent how data is stored in relational database. Relational database stores data in the form of relations (table). After designing the conceptual model of database using ER diagram we need to convert the conceptual model into relational model which can be implemented using any RDBMS languages

Ex: RDBMS languages (oracle, sql, mysql server)

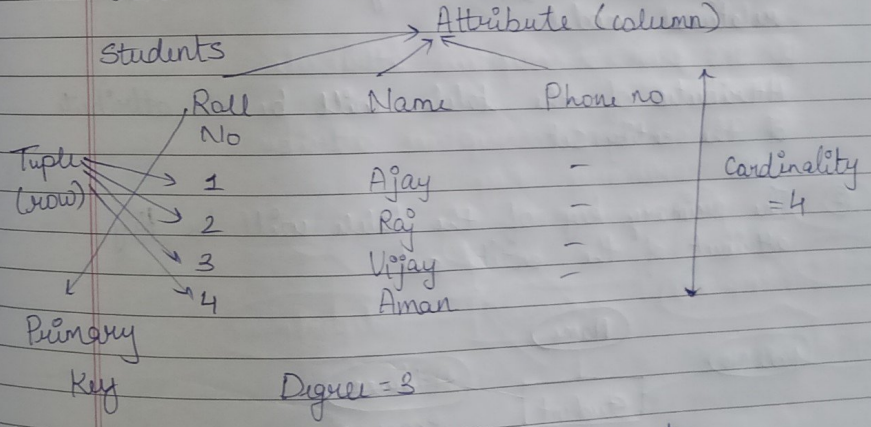
Ques What is RDBMS?

Relational Database management system
It is a basis for sql and all modern database systems like ms sql server, ibm dB2, oracle, mysql, microsoft excel

It is based on relational model as introduced by EF codd

Relational Model Concept

* Relation → It is a table with columns & rows.
* Attribute → It is a name column of a relation
* Domain → It is the set of valid values for one or ~~two~~ ~~o~~ more attributes
* Tuple → It is a row of a relation
* Degree → Total no of columns or attributes in the relations
* Cardinality → Total no of rows present in a table
* Relational Schema → It represents the name of relation with its attribute



Properties of Relational Operator

Each relation has unique name
Each tuple is unique (no duplicate row)

→ Entries in any column have the same domain
→ Each attribute or column have a unique name
→ Attributes are required to be atomic (has exactly one value)

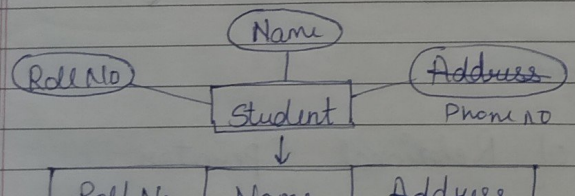
→ Order of columns and row is irrelavent

Rules For Conversion
* ER diagram → Relational Tables

Rule 1 Strong entity set with only simple attributes
① A strong entity set with only simple attribute will require only one table in relational model

Attributes of the table will be the attribute of the entity set

② The primary key of the table will be key attribut of the entity set

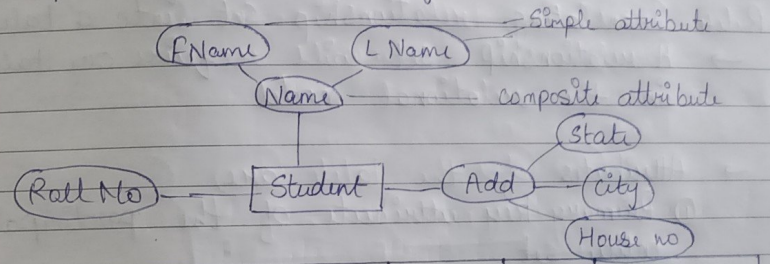


Roll No	Name	Address Phone no

Schema : Student (Roll No, Name, Add)
Phone no

Rule 2 Strong entity set with composite attribute
A strong entity set with any number of composite attribute will require only one table in relational model

While conversion simple attribute of the composite attribute are taken into account and not the composite attribute itself

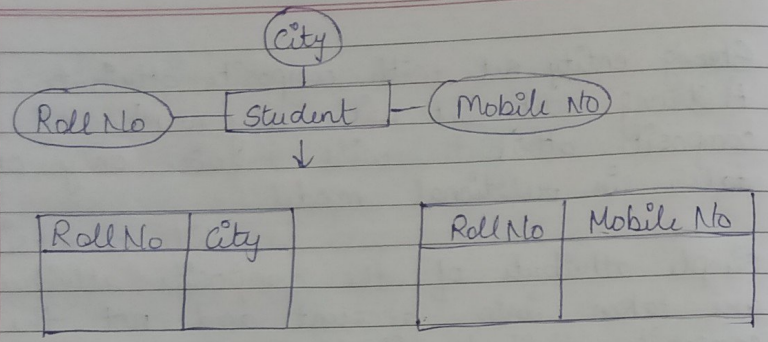


Roll No.	FName	LName	State	City	House No

Rule 3 Strong entity set with multivalue attributes
→ A strong entity set with any number of multivalued attributes will require 2 tables in relational model

→ One table will contain all the simple attributes with primary key

→ Other table will contain the primary key and all the multivalued attributes



Roll No	City

Roll No	Mobile No

Rule 4 Translating relationship set into a table

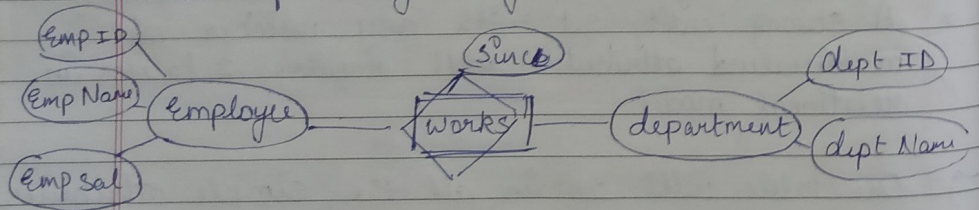
→ A relationship set will require one table in the relational model

→ Attributes of the table are

① Primary key attribute of the participating entity set.

② It's own descriptive attribute if any.

③ Set of non descriptive attributes will be the primary key



Employee

Em ID	E Name	E sal

Work

E Id	Since	d Id

department

dept ID	dept name	dept

For given ER diagram 3 tables will be required
1 table for entity set employee,
1 table for entity set department,
1 table for relationship set works in

23/03/22 For Binary relationships with

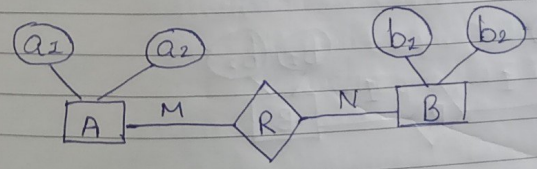
Case 1 Binary relationship with cardinality ratio (M:N)

Case 2 Binary relationship with cardinality ratio one-to many

Case 3 Binary relationship with cardinality ratio many to one

Case 4 Binary relationship with cardinality ratio one to one

1 Many to many



3 tables for A, B and for relationship

A (a_1 , a_2)

B (b_1 , b_2)

R (a_1 , b_1) (Primary key attributes are included in Relationship)

A

a_1	a_2

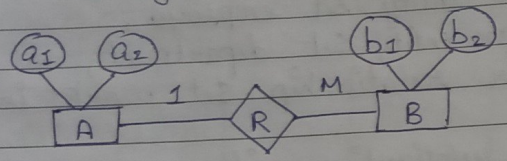
B

b_1	b_2

R

a_1	b_1

2 one to many

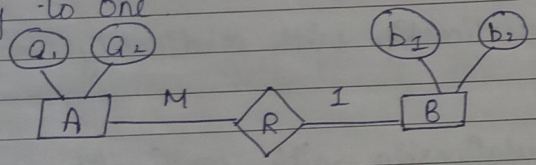


Tables = 2 A, BR

A (a_1, a_2), BR (a_1, b_1, b_2)

Relationship tables gets merged with the table at the side of many

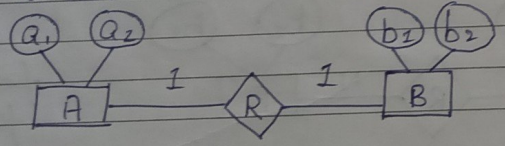
3. Many to one



AR (a_1, a_2, b_1) , B (b_1, b_2)

attributes

4. One to one



In one to one R gets merged with either A or B.

A (a_1, a_2) AR (a_1, a_2, b_1)

or

BR (a_1, b_1, b_2) B (b_1, b_2)

KEYS

Emp ID	Name	Adhar no	email	dept NO.
1	Rahul			1
2	NehaRaj			2
3	Neha ~~Sushil~~			~~3~~
4	Varun			~~5~~

~~Key is a attribute or uniquely identified attribute of a tuple~~

★ Keys in DBMS

A Key is an attribute or set of attributes that uniquely identifies any record (tuple) from the table

→ 1 column or combination of columns can be made as key.

★ Key is used to uniquely identify any record or row of data from the table

★ It is also used to establish and identify relationships b/w tables

Types of Keys

① Super Key

② Candidate Key

③ Primary Key

④ Alternate Key

⑤ Foreign Key

⑥ Composite Key

1 Super key → Super key is a combination of all possible attributes that can uniquely identify the rows or tuple in the given relation.
→ Super key is a super set of candidate key
→ A table can have many super keys.
→ A super key may have additional attribute that are not needed for unique identity.

• Max 5 columns can be made for {uniquely identifying} the rows
↳ no repitation

These are unique for each emp

Ex: EmpID, Adhar No, email ID
for combination {empId Adhar No}, {Adhar No, email}
{email ID, empID}

2 Candidate key → A candidate key is an attribute or a set of attribute which can uniquely identify a tuple or row. A candidate key is a minimal super key or a super key with no duplicate attributes it is called as minimal super key but we select a candidate key from a set of super key such that the selected candidate key is the minimum attribute required to uniquely identify the table.
Candidate key are defined as distinct set of attributes from which primary key can be selected.
Candidate key are not allowed to have null values.

3 Primary key → It is one of the candidate key selected by database designer - to uniquely identify the tuple in the relation.

The value of primary key can never be null. The value of primary key must always be unique. The value of primary key can never be changed means no updation is possible.
A relation allow to have only one primary key

4 Alternate key → Out of all candidate key only one get selected as primary key remaining are known as alternate key

In employee table empID - primary key rest of the attributes Adharno, email are considered as alternate key

6. Composite key → Set of attributes the key that has more than 1 attribute is known as composite key also known as compound key.

24/08/22

Imp 5. Foreign key → It is used to link 2 tables together. It is an attribute or set of attributes in one table that refers to the primary key of another table or in same table.

The purpose of foreign key is to maintain referential integrity of data.

Foreign key can take only those values which are present in the primary key of the refrenced relation

Foreign key may have a name other than that of the primary key.

Foreign key can take null values, their is no restriction on foreign key to be unique

Refrenced relation also called base table / master table / primary table and refrencing table also called Foreign table

★ It ~~can be~~ is an attribute or collection of attribute which refrence or indicate primary key of another table or its own table

Ex: Employee table (Refrencing ~~column~~ Relation) foreign key

EmpID	Name	No	email	depId
1	Rahul	-	-	1
2	Neha	-	-	2
3	Neha	-	-	2
4	Varun	-	-	3

Primary key

DepID	Dept Name
1	Sales
2	Marketing

Department table (refrenced relation)

★ Integrity constraints → condition

Integrity constraints are used to ensure accuracy and consistency of the data in a relational database. These are set of rules that the database is not permitted to violate. Constraints may apply to each attribute or they may apply to relationships b/w table. Integrity constraints ensure that changes (update, dilution, insertion) made to database by authorized user do not result in a loss of data consistency.

Ex: A blood group must be A or B or AB or O only

★ Thir are 4 type of integrity constraints

① Domain constraint
② Key constraint
③ Entity integrity constraint
④ Referential integrity constraint

① Domain constraints → It defines the domain or the valid set of values for an attribute

The datatype of domain includes string, char, int, time, day, currency etc.

The value of the attribute must be available in the corresponding domain.

3. Entity Integrity Constraints
It states that primary key value can't be null. This is because the primary key value is used to identify individual rows in relation and if the primary key has a null value then we can't identify those rows.
A table can contain a null value other than primary key field

2. Key Constraint → An entity set can have multiple keys or candidate keys but out of which one key will be the primary key. Key constraints specify that in any relation all the values of primary key must be unique.
The value of primary key not be null.

4. Referential Integrity Constraints
It specifies b/w 2 tables. Referential integrity constraints is inforced when a foreign key references the primary key of a table. In Referential integrity constraints if a foreign key in table 1 refers to primary key of table 2 then either every value of the foreign key in table 1 must be present in primary key value of table 2 or must be null.

Rules:
① You can't delete a record from a primary table if matching record exist in related table.
② You can't change a primary key value in the primary table if that record has related records.
③ You can't insert a value in the foreign key field of the related table that doesn't exist in the primary key of the primary table however you can enter a null value in the foreign key specifying that the records are unrelated.

25/03/22
* Query language is the language in which user request information from the database.
Ex: SQL

- There are 2 types of query language
① Procedural language
② Non procedural language

1. Procedural query language → In this language user instruct the system to perform a series of operations to produce the desired result.
User tells what data to be retrieved from database and how to retrieve it

2. Non procedural language (declarative)

Selection Operator (σ)

Selection operator is unary operator that perform selection operation.

It selects tuple or rows that satisfy given condition from a relation. It is denoted by sigma (σ)

Notation $\sigma_{p}(r)$ r = relation name / table name

where p = logical connectives and relational operator ($=, \neq, >, <, >=, <=$)

The where clause of SQL command corresponds to relational select operator.

Query 1 Select tuple from student table whose age is greater than 17

$\sigma_{age > 17}(Student)$

Std table	ID	age
A	1	17
B	2	18
C	3	16

Query 2 Select student whose roll no is 2

$\sigma_{roll\ no=2}(Student)$

Query 3 Select student whose name is D

$\sigma_{"name"="D"}(Student)$

Query 4 Select student whose age is greater than 17 and who lives in delhi.

$\sigma_{age>17 \wedge lives="delhi"}(Student)$

* Projection → Projection operator is a unary operator in relational algebra. It projects or display the particular columns or attributes. It delets column that are not in the projection list denoted by π

$\pi_{A_1, A_2, \ldots A_n}(r)$

or

$\pi_{attribute\ is}(relation\ name)$

* Duplicate rows are automatically eleminated from resulted.

- The SQL Select command corresponds to relational project operator

Select $A_1, A_2 \ldots A_n$ from relation r (table name)

① display the columns, roll no, name, from relation student

$\pi_{roll\ no,\ name}(Student)$

② display age of student in student table.

$\pi_{age}(Student)$

③ display the roll no, name and student whose age is greater than 17

$\Pi_{name, Rollno}(\sigma_{age > 17}(Student))$