1. Pen paper system
2. Fill system

Father of DBMS
Edger F Codd.

SQL commands

DDL

Data definition language
1. create
2. alter
3. Drop
4. Desc
5. use

DML

Data manipulation language
1. Insert
2. Select
3. update
4. Delete

DCL

Data control languages
1. GRANT
2. REVOKE

Table name :- client_master

Client No	Name	City	Pincode	State	Baldue
C0001	Ivan	Mumbai	680503	Kerala	150000
C0002	Ashwini	Chennai	780001	TamilNadu	0
C0003	Toshi	Manglore	56001	Karnataka	5000
C004	Deepak	Chennai	78005	Tamil Nadu	0
C005	Sharma	Mumbai	40054	Maharastra	20000

Queries : DDL

* Create table client_master (Client No varchar(10), Name char(8), City char(20), Pincode integer(10), State Char(30), Baldue integer(10))

→ Alter to change structure of Data base

Alter add Alter drop

Alter modify

Alter close

Alter table client_master add (gender char (5));

Alter table client_master drop gender;

Alter table client_master modify (Name varchar (20));

3. Drop

used for ~~ext~~ delete entire structure of table

Table drop and database drop

syntax:

1. drop table client_master;
2. drop database database name;

4. Desc

or describe

To view the structure of table

Syntax: Desc client_master;

5 Use

used for using the database

Syntax: use <table name>;

(DML) Data Manipulation Language

1. Insert: Inserting records in a table

insert into client_master values ('10001', 'ivan', 'Mumbai', 400054, 'Maharastra', 15000);

Syntax: insert into <tablename> values (<row values>);

2. Select:

Retrieving data from table

Select * from client_master; for view whole table

Select Name, city from client master; Specific columns.

2. update query: To update ~~client~~ the data inside the table

update client_master set Name = 'Ram' where client No = '10001';

we use where clause for specify the condition.

Syntax: update <tablename> set <column name> = ' ' where <condition>;

4. Delete command:

for deleter data from table

delete from client_master; for delete whole table

delete from client_master where clientNo = '10002';

(Row naming client No. '10002' will be deleted)

for multiple condition use and or or

&&

delete from client_master where city = 'Mumbai';

like clause: To check the pattern gives:

_ (underscore): To use in select command to display exact no. of characters

% : used to display character afterwards can be any string.

Data control Language (DCL)

1. GRANT for permitt grant / To geant privilege.

2. REVOKE

To take back the given privilege.

Q: From client_master Table given abelow:

Table name :- client_master

clientNo	Name	city	pincode	State	BalDue.
C0001	Ivan	Mumbai	400054	Maharastra	15000
C0002	Ashwini	chennai	780001	Tamil Nadu	0
C0003	Joshi	Mangalore	560001	Karnataka	5000
C0004	Deepak	chennai	780001	Tamilnadu	0
C0005	Sharma	Mumbai	400054	Maharastra	2000

Q.1 ~~Find the~~

change the city of client no C0005 to Bangalore

Sol: update client_master set city = 'Bangalore' where client no = 'C0005'

Q.2

List the names of all client having alphabate 'a' as the second letter in their names.

like for checking matching

Sol Select Name from client_master where Name like '_a%';

Q3. List the name of all clients having exactly char - for print 5 character in their name. med %

Sol Select Name from client_master where Name like '_____';

Q.3 List all the names of all client have alphabet 'a' in second last

Sol Select Name from client_master where name like '%a_';

Q.4 List the name of client who stay in a city whose first letter is m.

Sol Select name city from client_master where city like 'm%';

Q.5

List all the client where Baldue is greater than 10000

Sol Select Baldue* from client_master where Baldue >1000';

Q.6

Count the total number of clients.

Sol. Select count (name) or Select count(client_No) from client masters.

Select count (*) from client_master.

Note: If we use count(*) in the table we will get 5 no. of rows but if there is a 'NULL' value in the Name column and we count (Name) there will be 4 as count.

count

inorder to count the number of row

count (columnname) avoids null value

count (*) counts all value inclusive of duplicates and null.

distinct

Select count (distinct column: name/city) from client master;
gives output as 3.

groupby
having clause

Introduction:

collection of queue interrelated data raw facts and figures.

Meanigful data is called information.

Data examples: girl

infamation eg: color of pen is blue.

Database: Collection queue interrelated data that can be stored.

It is a storage area queue related data.

DBMS: Software - set of queue programs

DBMS software is a set of programs used to store data and retrieve data from the database

Types of DBMS

Relational Database Management system software

RDBMS: There is a relation in DBMS called tables which contains rows and columns

eg: Oracle, MySQL or Mongodb

→ rows are called tuple / record.

→ columns are called field.

Application of DBMS:

Universities, Social Media, School, Reservation system, Medical fields - to store patient data, doctor's data, pharmacy etc.

Characteristics of DBMS :

1. Reduced Redundancy
2. Data Consistency :
   Change in data should be reflected everywhere in a database.
3. Support multiple user and concurrent uses
   we can access same time.
4. It uses query language.
5. It provides security
6. It supports teansaction
   (no partial teansaction takes place)
7. It uses real world entity.
8. It support multiple data types
9. It support ACID propesty
   - A - Atomicity - It teansaction complete 0% or 100%
   - C - Consistency - change in data should be reflected everywhere
   - I - Isolation - Multiple teansaction can occur independently
   - D - Durability - without the enteeference of some othee teasaction
     - A transaction that has been 100% computed should be reflected in database
10. Ease of access
11. Stores any kind of steuctured data
12. Reduced storage space.

1. STRUCTURED DATA :

It concerns all data which can be stored in database SQL in a table with rows and columns.
eg: Relational Data

2. Semi structured Data

Is information that does not reside in a relational database but that has some organisational properties that make it easier to analyse.
eg: XML Data

3. Unstructured Data :

It is a data which is not organised in a predefined manner.
eg: Word, pdf, text

Data Models :

- Primary Keys :
  It is the column that contain values that uniquely identity each row in a table.
- Constraint :
  It is used to specify rules for data in a table
- Unique Constraint :
  It ensures that all values in a column are different.
- Check Constraint :
  Used to check condition
- NOT NULL constraint :
  There should be not NULL constraint in a colum

- Default constraint :
  to set default value

→ Create table client_master (clientNo varchar(20) primary key, Name char(20));

→ Check :

eg: Create table client_master (clientNo varchar(20) primary key, Name char(20), salary integer check (sal > 500 and sal < 10000);

→ Unique Constraint :

eg: Create table client_master (clientNo varchar(20) primary key, Name char(20), salary integer unique (salary));

→ Not Null :

eg: create table <tablename> (clientNo varchar(20) primary key, Name char(20), salary integer not null);

→ Default Constraint :

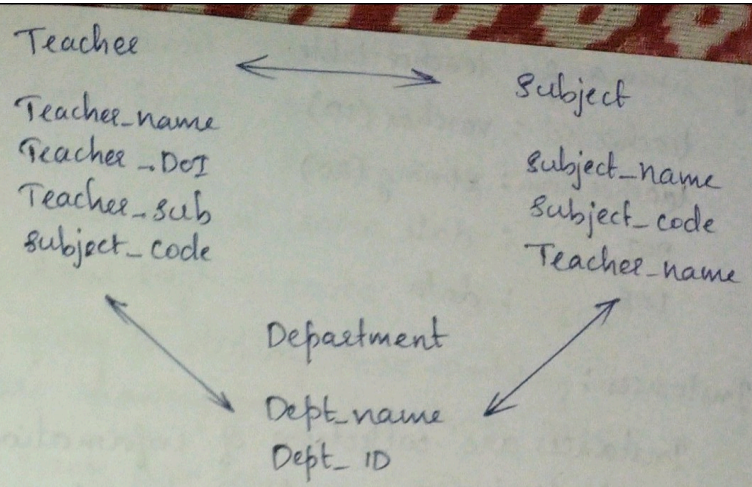
eg: Create table <tablename> (clientNo varchar(20) primary key, Name char(20), salary integer default (1000));

* Data Models : It is a tool for describing data, data relationship and consistency constraints. It consist of one or more schema.

* Schema : Schema specifies what field will be present and what would be their types.

eg: An employee table will have employee_id with a string of there are 3 types of schema-

eg:



3-Level architechture :

a. Physical schema :

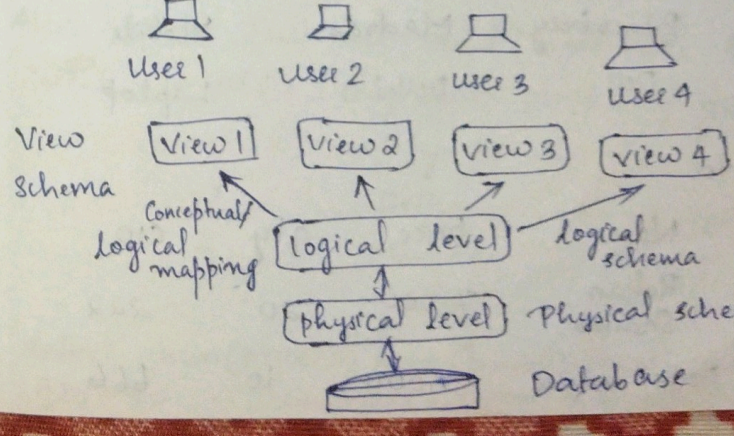
It defines how the data is stored in secondary storage device

b. Logical schema :

It defines all the logical constraints that need to be applied on the data stored.

c. View schema :

It defines how and any end user will interact in the database system.



eg: Schema for teacher table

teacher_id : varchar (10)
teacher_name : string (20)
DOJ : date
DOB : data

Instances:

Instances are collection of information stored at a particular moment. Instances can be changed by operations like: addition, updation, deletion of data.

eg:

schema is overall description of database.

Q: company:

CID	Name	City	product Name
111	Sony	Delhi	TV
222	Nokia	Mumbai	Mobile
333	Onida	Delhi	TV
444	Sony	Mumbai	Mobile
555	Blackberry	Madras	Mobile
666	Dell	Delhi	Laptop

Customer

Custid	Name	price	qty	CID
101	Rohan Sharma	70,000	20	222
102	Deepak kumar	50000	10	666
103	Mohan kumar	30000	5	111
104	Sahil Bansal	35000	3	333
105	Neha Soni	25000	7	444
106	Sonal Aggrawal	20000	5	333
107	Arnav Singh	30000	15	666

a. Write a query to display those company name which are having price less than 30000.

Sol. Select Name from company, customer where company.CID = customer.CID and price < 30000;

Q: To display the name of the company in reverse alphabatical order

Sol. Select Name from company order by Name desc;

Q: To increase the price by 1000 for those customers whose name starts with capital S.

Sol update customer set price = price + 1000 where Name like 'S%';

Q: To add one more column total price with decimal (10,2) to the customer

Sol Alter table customer add (Total price decimal (10,2));

Sol select count(x) city from company group by city.

count (x)	city
3	Delhi
2	Mumbai
1	Madras

Sol Select min(price), max(price) from customer where qty > 70 Min(price) = 50000 & Max(price) = 70000

Sol: select average of Qty from customer where Name like '%r%';

avg(qty)

Note: Order by name is written then it will be sorted according to ascending order.

Q: Vehicle Table:

Vcode	VehicleType	per km
V01	Volvo bus	150
V02	AC Delux	125
V03	Ordinary	80
V05	SUV	30
V04	Car	18

Travel Table:

CNo	CName	TravelDate	km	Vcode	NOP
101	K.Niwal	2015-12-13	200	V01	32
103	Fredick sym	2016-03-21	120	V03	45
105	Hitesh Jain	2016-04-23	450	V02	40
102	Ravi Anish	2016-01-13	80	V02	40
107	John Malina	2015-02-10	65	V04	2
104	Sahanubuti	2016-01-28	90	V05	4
106	Rakesh Jaya	2016-04-06	100	V01	25

Qa. To display CNO, (Name, Travel) data from the table in descending order of CNO.

Sol: select CNO, CName, Travel from Travel order by CNo desc;

Qb. To display the CName of all customers from table Travel who are travelling by vehicle with code V01 or V02.

Sol: Select CName from Travel where Vcode = 'V01' or Vcode = 'V02'; or

Select CName from Travel where Vcode in (V01, V02)

Qc. To display CNO and CName of those customers from the table travel who travelled between '2015-12-31' and '2015-05-01'.

Sol: Select CNo, CName from Travel where TravelDate >= '2015-05-01' and TravelData <= '2015-12-31';

OR

Select CNo, CName from Travel where TravelData between ('2015-05-01', 2015-12-31);

Qd. To display all the details from table: Travel for the customers who have travel distance more than 120km in ascending order of NOP.

Sol: select * from Travel where km > 120 order by NOP;

Qe. select count(*), Vcode from travel group by Vcode having count(*) > 1; find the action of the query.

Sol

Count(*)	Vcode
2	V01
2	V02

Qf. Select distinct vcode from teavel; find the output

Sol

distinct Vcode

V01
V02
V03
V04
V05

Qg. Select a.vcode, cname, velicle type from teavel a, velicle b where a.vcode = b.vcode and km < 90; find the output.

Sol

Vcode	Cname	Velicle Type
V02	Ravi Anisn	AC Delux Bus
V04	John Malina	Car

Qh Select cName, km * per km from velicle b, teavel a from a.vcode = b.vcode and a.vcode = 'V05'; find output.

Sol

cName	km per km
Sahanu butti	2700

Q: Applicants

ID	NAME	COURSE	PH NO.	JOIN YR	GEN
1.	Ram	BCA	4210716	2000	M
2.	Rita	MBA	215611	2002	F
3.	Sunil	MCA	321157	2001	M
4.	Kumar	BCA	512771	2000	M

Course

NAME	DURATION	FEE	ID
MBA	2yr	40,000	1
MCA	3yr	70,000	2
BCA	3yr	20,000	3

Qa: To display name, fee, gender, Joinyear of the applicants who have joined before 2002

Sol. Select Name, Course, Fee, Gen, Join yr from Applicants from Applicants. course = courses where join yr < 2002;

Qb: To display names of applicants who are paying fee more than 30,000.

Sol. Select Name from Applicants, courses where applicant course = course. cours and fee > 30,000;

Qc. To display names of all applicants in ascending order of their join year.

Sol. Select Name from Applicants order by join yr;

Qd: To display the year and total no. of applicants joined in each year from table applicants

Sol. Select year, count(*) from Applicants group by year;

Qe: To display course and number of applicants registered in the course from applicants table.

Sof select course count(*) from Applicant group by course;

Q: Given the output of the SQL table

ITEMS

Code	Iname	Qty	price	Company	TCode
1001	Digital pad 12b	120	11000	Xenita	T01
1006	Ledscreen 40	70	38000	Santora	T02
1004	Car GPS sys	50	21000	Geoknow	T01
1003	Digital Camera 12X	160	8000	digildeck	T02
1005	Pendmic 32GB	600	1200	Starchome	T03

Tradees.

Tcode	Tname	City
T01	Electronic Sales	Mumbai
T02	Busy store shop	Delhi
T03	Disp Home	Chennai

Qa: To display the details of all items in ascending order of itemname.

Sof select * from items order by Iname;

Qb: To display Itemname and price of all those items whose price is in the range of 10000 and 22000 (both values inclusive);

Sof select Itemname, price from Item where price >= 10,000 and price <= 22000;

Qc: To display number of items which are traded by each tradee.

Sof select Tname, count(*) from Tradees, Items where tradees.Tcode = Items.Tcode group by Items.Tcode.

Qd: To display price, Itemname and Qty of those items which have Qty > 150.

Sof select * price, Itemname, Qty from items where Qty > 50;

Qe: To display names of those tradees who are either from Delhi / Mumbai

Sof. Select TName from Tradees where city in (Delhi, Mumbai);

Qf: To select max (price), min (price), from Itemes

Sof: Select distinct Tcode from Items.

Q: Given the output of the SQL Table

Supplies

Supcode	Sname	City
S01	Getall inc	Kolkata
S03	Easy market corp	Delhi
S02	Digibusy group	Chennai

Table Name : Products

PID	PName	Qty	price	Company	Supcode
101	Digital Camera 14x	120	12000	Renfix	S01
102	Digital Pad 11i	100	22000	DigiPop	S02
104	Pendrive 16GB	500	1100	Storeking	S01
106	Led Screen 32	70	28000	Dispixpots	S02
105	Car GPS System	60	12000	Morson	S03

Qa: To display the details of all the products in ascending order of product names.

Sol: Select * from products order by pname;

Qb: To display Product Name and price of all those products whose price is in the range of 10000 and 15000 (both values inclusive)

Sol: Select PName, Price from Products where Price >= 10000 and price <=15000;

Qc: To display number of products which are supplied by each supplier.

Sol:

```
S01  2
S02  2
S03  1
```

Select supcode, count(*) from products group by Supcode;

Qd. To display price, product name and Qty of those products which have Qty > 100;

Sol. Select price, Pname, Qty from products where Qty > 100;

Qe. To display the names of those supplies who are either from Delhi or from Chennai.

Sol: Select SName from supplies where Qty in (Delhi, Chennai);

Qf. To display the name of the companies and name of products in descending order of company name.

Sol. Select company, Pname from Products order by company desc;

Qg:

Sol: select distinct supcode from products

Qh.

Sol. Select max(charges) from price.

Four types of integrity Constraints:

1. Domain Constraints: If we input values of wrong datatype domain constraints generated.
2. Referential Integrity Constraints:
3. Assertion
4. Authorization: to different users:

- Read authorization
- Insert authorization
- Update authorization
- Delete authorization.

DDL - Interpreter } High level
DML - Compiler } ↓ Machine level

- Data Dictionary: A storage location where output of DDL is stored.

These stored data termed as / contains Metadata.

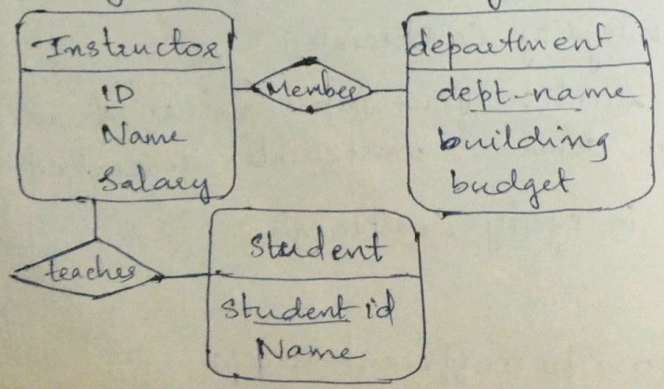
- Metadata - Data about data.

Data Model.

- Relational Data Model
- ER Diagram / Entity-Relationship data model.
- Object Based data Model
- Semi-structured data Model

eg: XML extensible markup language

- Entity Relationship Diagram:



Entity: Real time objects

Primary key must be underlined.

Data Storage:

- Database storage manger - communication between different components
  - ⊙ Authorization and integrity manager
  - ⊙ Transaction Manager - No partial transaction of System failure.
  - ⊙ File Manager - Disk storage
  - ⊙ Buffer Manager (Swapping in and swapping out)

Data structure:

1. Data file → file to store database
2. Data dictionary → contains metadata
3. Indices → for fast access of data like index of textbook.

(Hashing is an alternative to indices)

Query Processor:

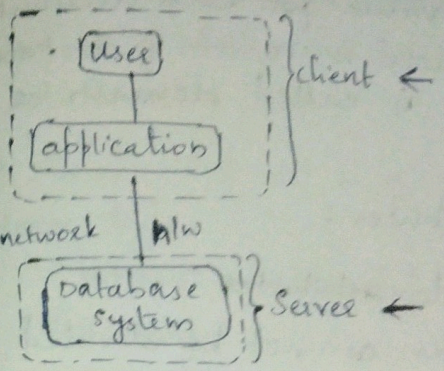
- → DDL Interpretor
- → DML Compiler
- → Query Evaluation Engine: Executes the converted low level language (output)

Interpretor - line by line converters high level language to low level language.

Compiler - As a whole at once converts to low level languages.

System Architecture

2-tier Architechture

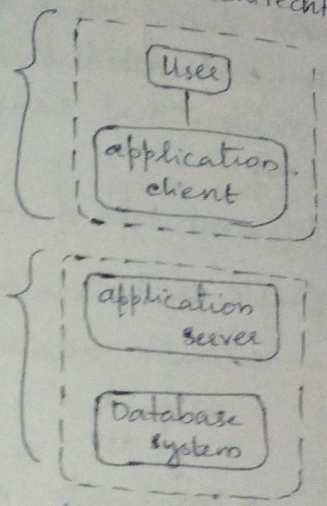


- direct communication
- JDBC, ODBC.

JDBC - Java Database Connectivity

ODBC - Object Database Connectivity.

3-tier Architechture



- No direct communication
- Used in web processes
- More appropriate for large application

Database Administeration : Has central control over system

Functions :

1. Schema definition : Using DDL Commands
2. Storage structure and access method definition
3. Schema and physical organisation modification
4. Granting of authorization for data access
5. Routine maintainence
   - Periodically backup free
   - Ensuring memory disk space for normal operations
   - Monitoring jobs running on the database

- An entity is represented by a set of attributes
- Each entity has a value for each attributes
- For each attribute there is a set of a permitted values. That range is called domain or value set.

Different type of attributes :

1. Simple and Composite attributes :
   * Simple attributes cannot be divided into subparts
     eg: ID
   * Composite attributes can be divided into subparts
     eg: Name (can be divided into first name, middle name, last name)
2. Single valued and Multivalued attributes :
   * Single valued attributes have only one value
     eg: ID
   * Multivalued attributes have more than one value
     eg: Phone no.
3. Derived attribute :
   - Attribute derived from existing attribute of table.
   - The value of this type of attribute can be derived from the value of other related attribute
     eg: DOB derived from age (almost)

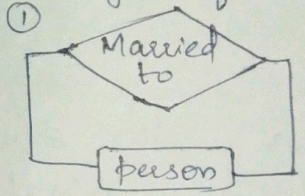
Mapping Coardinabities :

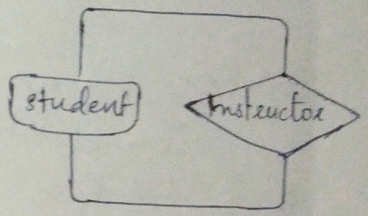
It express the number of entities to which another entity can be associated via a relationship set.

A relationship set is a set of relationship of some type
→ Unary relationship set,
→ Binary,
→ Ternary,
→ n-ary relationship set

1. Unary relationship : Unary relationship set exists when both entity types are the same. Degree of relationship 1.

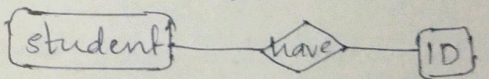
eg: ①



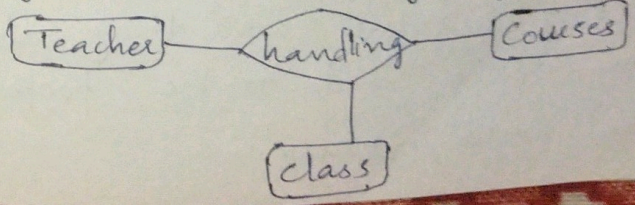


2. Binary relationship : A binary relationship exists when there are two types of entities and degree of relationship is 2.

eg:



3. Ternary relationship : Degree of relationship is 3.



Degree means how many entities connected.

Mapping Cardinalities:

1. One-One : An entity in A is associated with atmost one entity in B and Vice versa
2. One-Many relation : An entity A is associated with any number of entity in 'B' ~~and vice~~ OR An entity 'B' is associated with atmost 1 entity in 'A'
3. Many to one relation : An entity A is associated with atmost one entity in B OR An entity in B is associated with atleast one entity in 'A'.

Database Schema:

The database schema is the structure of a database described in a formal language supported by the database Management systems (DBMS).

Data Model:

A data model is an abstract model that organizes elements of data and standardize how they relate to one another and to the properties of real world entities.

Type of Data Models:

- Relational DM
- ER / Entity Relationship DM
- Object based DM
- Semistructured DM

DDL	DML	DCL	TCL
Data Definition language	Data Manipulation language	Data Control language	Transaction Control Language
→ Create	→ Insert	→ Grant	
→ Alter	→ Select	→ Revoke	
→ Drop	→ Update		
→ Desc	→ Delete		
→ Use	COMPILER		

INTERPRETOR

Integrity Constraints:

It is a mechanism for limiting the possible states of the database.

Types of IC:

- Domain Constraint : A limit on the values that a certain variable can take
- Referential IC : A part of an association between two entity types. (same column name while joining tables)

- Assertion Constraints : An assertion is a statement in SQL that ensures a certain condition will always exist in the database.
- Authorization Constraints : The process where the database manager gets information about the authentificated user.

Different types of keys in DBMS:

1) Primary key - A column/attribute which uniquily identify a tuple in Relation.
2) Alternate key - All Candidate key which are not primary key.
3) Candidate key - An attribute which can be primary key.
4) Foreign key - A table's primary key is another tables foreign key.
   The column of the table used to point to primary key of another table.

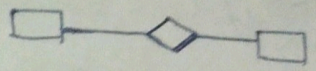
Any of the Candidate key has to be present in the other table as foreign key.

Primary key + Alternate key = Candidate key.

Strong Entity and Weak entity

Strong Entity -

* Not dependent on any other entity.
* Atleast one primary key
* The relation between these entity are single lined.



Weak Entity -

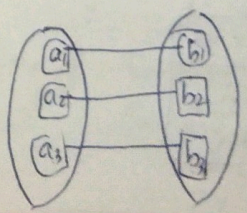
* Dependent on other entities
* No primary key
* Relation between these entity are double

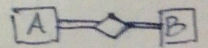
Participation set :

Specifies number of instances of an entity that can participate in a relationship set.

① Total Participation
② Partial Participation

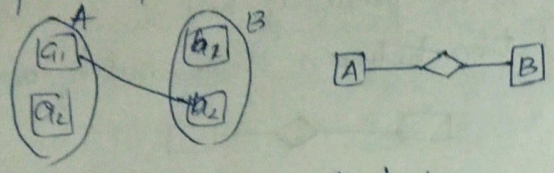
① Total participation : Participation of an entity set 'E' in a relationship set 'R' is said to be total if every entity in 'E' participates atleast one relationship in R.





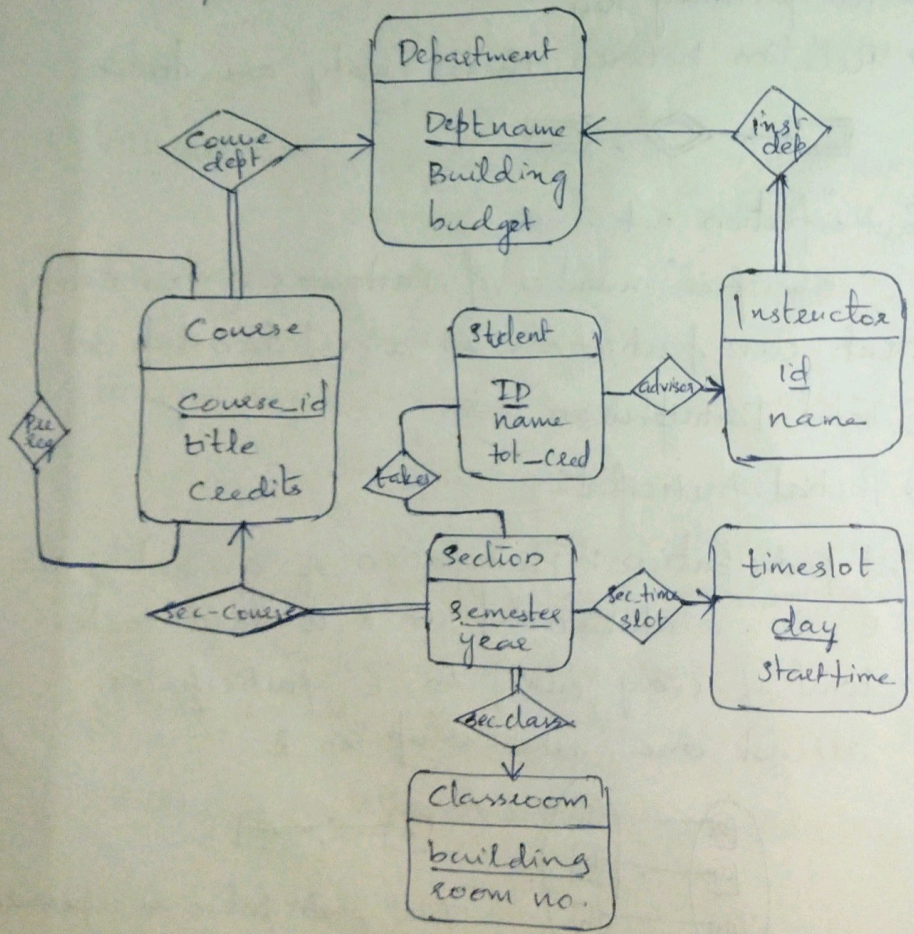
Total participation is represented by double line.

(2) Partial participation : participation of a entity set E in relation R' is said to be partial if and only if some entities in E participates in relationship with R



Participation represented by single line.

Example : ER Diagram of University Department



Synthesizing ER Diagram to Schema

step 1 : ER Diagram → entity → table

step 2 : attributes → columns

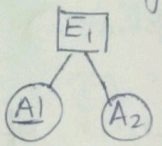
step 3 : key attributes → primary key

step 4 : Derived attributes → ignore

step 5 : Multivalued attributes → Create table from multivalued attribute

step 6 : Composite attributes → the root is not included in the table only the sub-braches are included as columns in the table.

(1) ER Diagram of strong entity :



step 1 : ER diagram → Entity → table

step 2 : attributes → columns

step 3 : key attributes → primary key

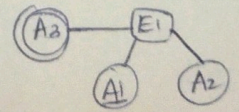
Schema representation :

$E_1 (\underline{A_1}, A_2)$

table representation

E_1	A_1	A_2

(2) Multivalued Entity :

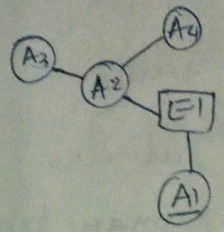


Schema representation

$E_1 (\underline{A_1}, A_2)$

$E_2 (\underline{A_4}, A_1, A_3)$

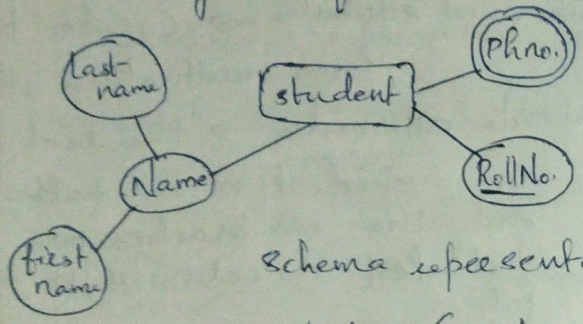
③ ER Diagram of Composite Attribute



Schema Representation

E1 (A1 , A3 , A4)

④ ER Diagram of Student (example) :



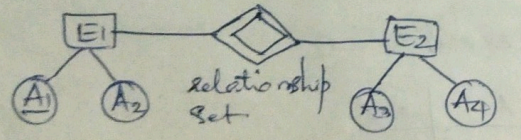
Schema representation :

Student (RollNo. , firstname , lastname)

phone (phId , RollNo. , phno.)

Note ER diagram is said to be blue print of relational model.

⑤ ER Diagram for Weak entity



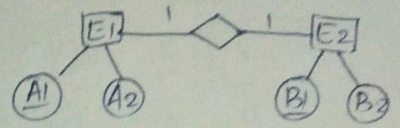
Schema representation

E1 (A1 , A2)

E2 (A1 , A3 , A4)

⑥ Mapping Cardinality ER diagram :

(i) One - One mapping :



Schema representation

E1 (A1 , A2)

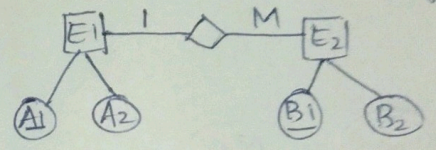
E2 (B1 , B2 , A1)

OR

E1 (A1 , A2 , B1)

E2 (B1 , B2)

(ii) One - Many Mapping : 1:M

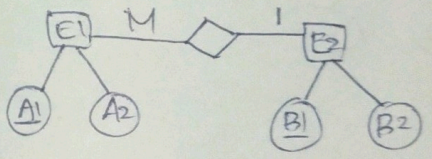


Schema representation

E1 (A1 , A2)

E2 (B1 , B2 , A1)

(iii) Many - One Mapping : M:1

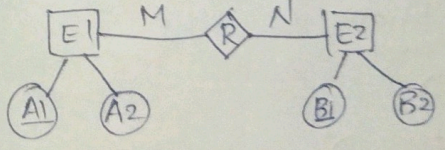


Schema representation

E2 (B1 , B2)

E1 (A1 , A2 , B1)

(iv) Many - Many mapping : M : N



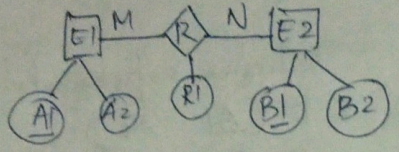
Schema Representation

E1 (A1 , A2)

E2 (B1 , B2)

R (A1 , B1)

⑦ Relationship entity:



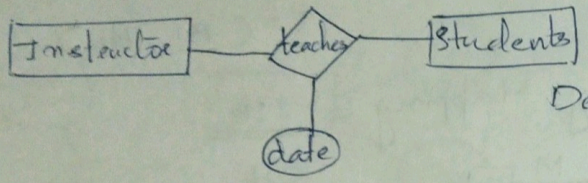
Schema representation

E1 (A1 , A2)

E2 (B1 , B2)

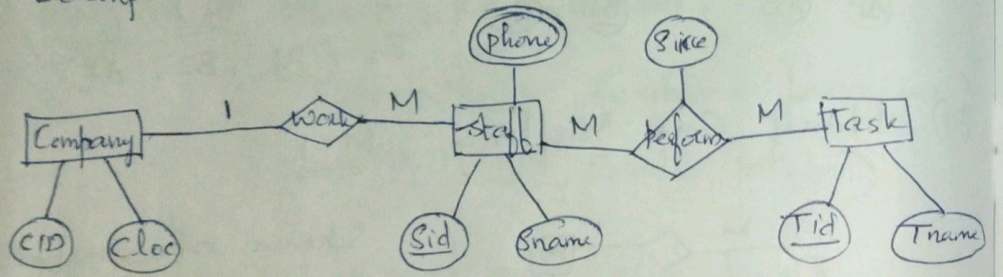
R (A1 , B1 , R1)

eg:



Descriptive attribute

Example:



Schema representation

company (CID, Cloc)

staff (Sid, Sname, CID)

phoneNo (phid , Sid , phone)

Task (Tid , Tname)

perform (sid , Tid , Since)

Relational Algebra

- Procedure Query Language
- Operators in relational Algebra
  1) Unary Operators
  2) Binary Operators
  3) From set theory

① Unary Operators
   a) Select (σ)
   b) Project (π)
   c) Rename (ρ)

② Binary Operator
   a) Join (⋈)
   b) Cartesian product (×)

③ Set theory (∪ , ∩ , −)

a) Select Operator (σ):

It is used to find tuples in a relation, which satisfy the given condition

syntax: σ condition (relation name)

eg: To retrieve entire details from student table:

σ (student)

To display ID, name of student:

$\sigma_{id,\ name}$ (student)