:- Data Base Management System :-

DataBase – Collection of related Data

- **Structured** (RDBMS)
  - eg: → IRCTC
  - → University
- **Unstructured**
  - → webpages

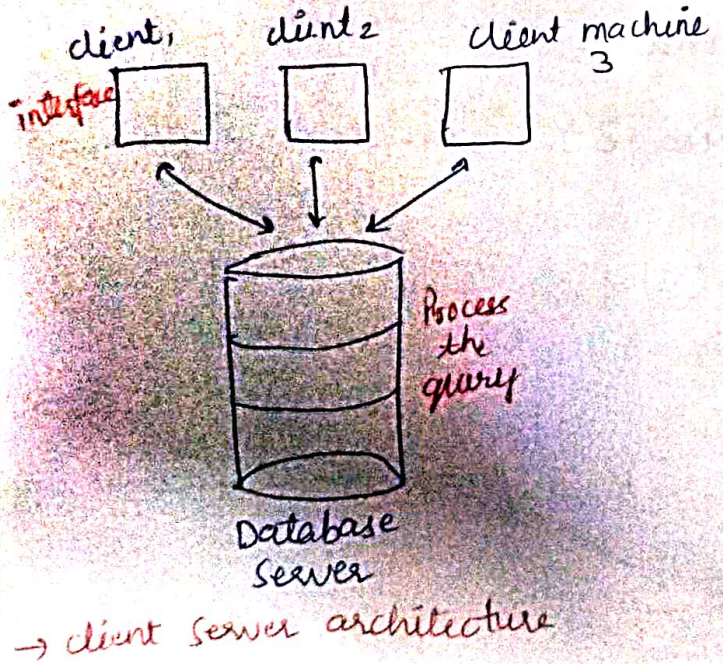
DBMS

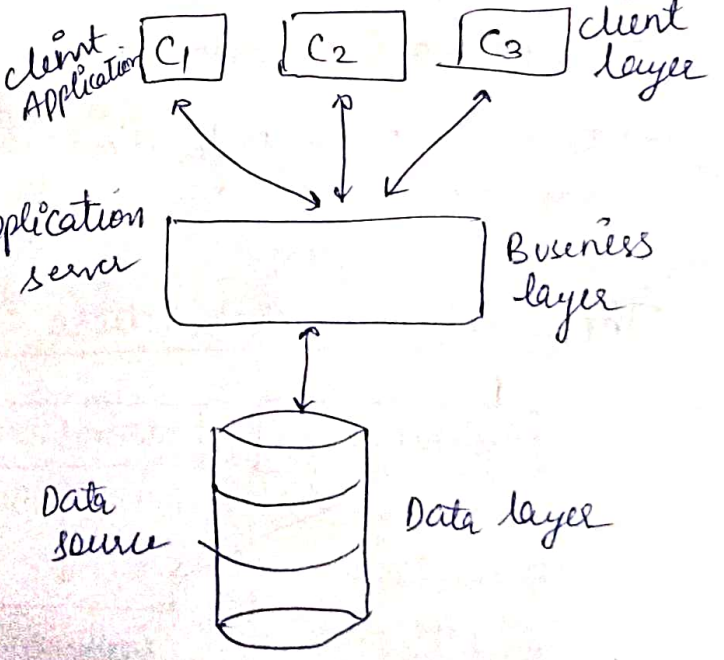
- → SQL server
- → Oracle 9i, 11, 12c etc
- → MySQL
- → DB_2

File System Vs DBMS

- → searching is fast
- → efficient memory utilization
- → Concurrency (protocols exist)
- → Security (Role based & Access control)
- → Reduce Redundancy

2 Tier and 3 Tier Architecture





→ client server architecture

eg: Bank
Railways [offline]

→ fails when no. of users are more

DataBase:- It is a collection of related data and storage area where we store it for retrieving information.

2 Tier

→ Scalability problem (No. of users are large)

→ Good Maintanance

→ Low security (clients directly interact with database)

3 Tier

→ query will be processed at business layer

→ Less load at database server

→ More secure

→ Hard to maintain

eg. web applications

schema :-

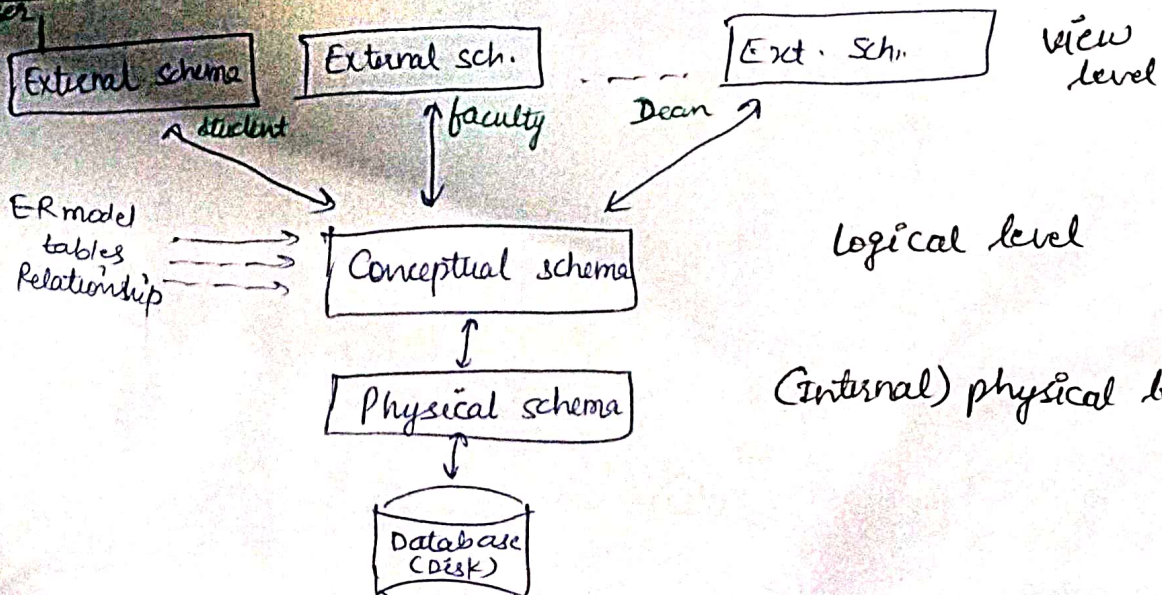
logical representation of database

↳ Basically the frontend

→ In RDBMS, table is a schema

→ In SQL, Data Definition language is used to implement the scheme

Three schema Architecture :-

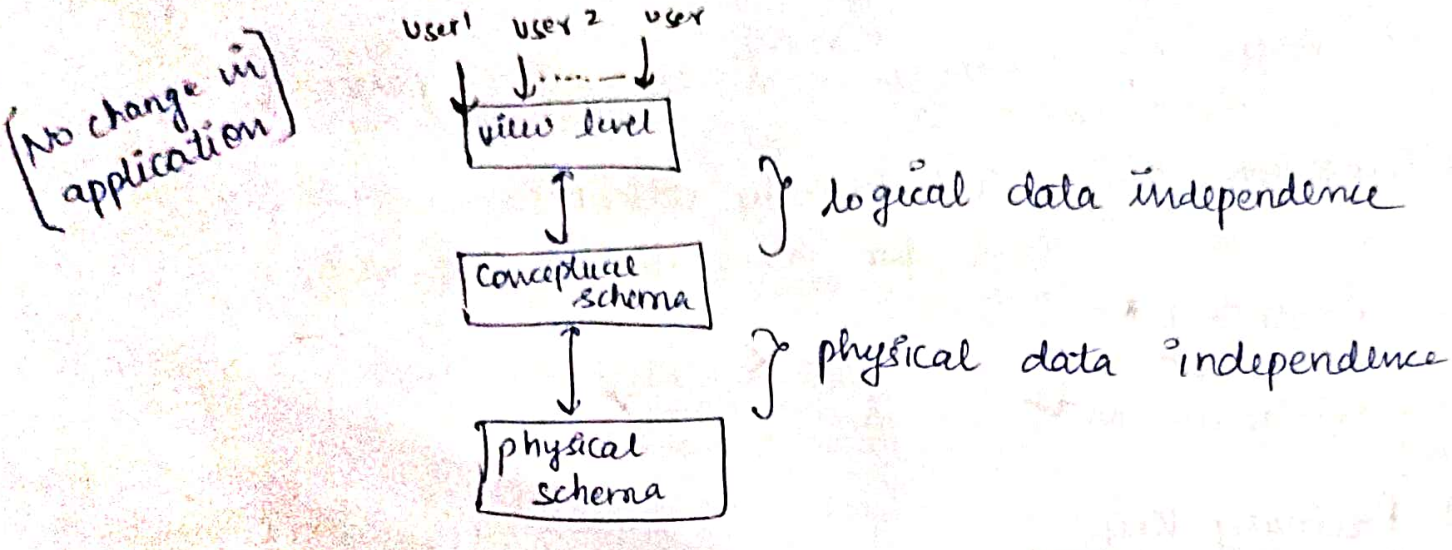


- front end (View)
- Data Base designer (logical)
- DataBase Administrator (physical)

(what data is actually stored) ←

* In Hard disk, data is stored in files

Data Independence / Abstraction :-



* The change at Conceptual level (adding or removing columns etc) will not affect the view level. This is called logical data independence.

* The change at physical level (eg. shifting data from hard disk 1 to a hard disk 2) will not affect the conceptual schema. This is called physical data independence.

eg: 1) storage structure
2) Data structure change
3) Index

Key

Attribute → Uniquely identification.

eg: Student table

Aadhar Card, Roll no, Registration no., Licence no, voter id, email ID, phone no.

Collection of all the Keys is called Candidate Key Set

Primary Key is choosen from the set of Candidate Key.

→ Candidate Keys include the attributes which can never have same values for more than one entry.

eg: Adhar Card ✓
PAN ✓
Registration no ✓

Age ✗
father Name ✗

Primary Key

{Unique + NOT Null}

eg: Registration Number

→ Only one primary Key

Foreign Key :-

It is an attribute or set of attribute that references to primary key of same table or another table (relation)

→ maintains referential Integrity

eg:

PK ✓ Base/Referenced table

Roll No	Name	Address
1	A	Delhi
2	B	Mumbai
3	C	Hapur
1		
1		

Referencing table — FK

Course Id	Course Name	Roll No.
C_1	DBMS	1
C_2	Networks	2

→ create table Course (Course_id varchar (10), Course_name varchar(20), Rollno int references Student (Rollno));

→ Alter table Course ADD constraint fk foreign key (Rollno) references Student (Rollno);

→ One table can have more than one foreign key.

Referenced table	Referencing table
1) Insert - No violation	1) Insert - May cause violation
2) Delete - May cause violation (Solution → On delete Cascade) → (On delete set Null) → (On delete no action)	2) Delete - will not cause any violation
3) Updation - May Cause Violation	3) Updation - May cause violation

Super Key

→ Super key is a combination of all possible attributes which can uniquely identify two tuples in a table

→ Super key set of any candidate key is super key.

Q:- R (A_1 A_2 A_3 A_4 ---- A_n) then how many super keys are possible? If A_1 is candidate key =? 2^{n-1}

If A_1, A_2 are candidate keys =? $2^{n-1} + 2^{n-1} - 2^{n-2}$

A_1 A_2 A_3
2 2 2

A_1 (———) = 2^{n-1}
fix $n-1$

Entity Relationship Model

Entity → Any object which has physical existence

Student (Rollno, age, address) Entity type or Schema

Types of Attributes:-

1) Single Vs Multivalued Attribute

↓ eg. Registration no.

↓ eg. Phone no.

2) Simple Vs Composite Attribute

↓ eg. Age

↓ (Composed of more than one value)

eg. student Name [First Name Middle Name last Nam

3) Stored Vs Derived Attribute

↓ (Derived from another attribute)

eg. age (Can be derived from DOB)

4) Key Vs Non Key Attribute

↓ used to uniquely identify each and every row.

5) Required Vs Optional Attributes

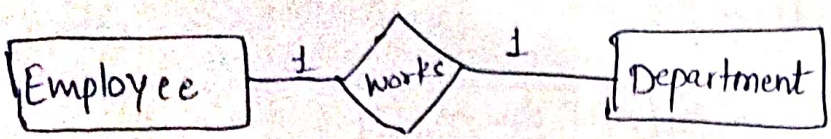
(Not part of ER model)

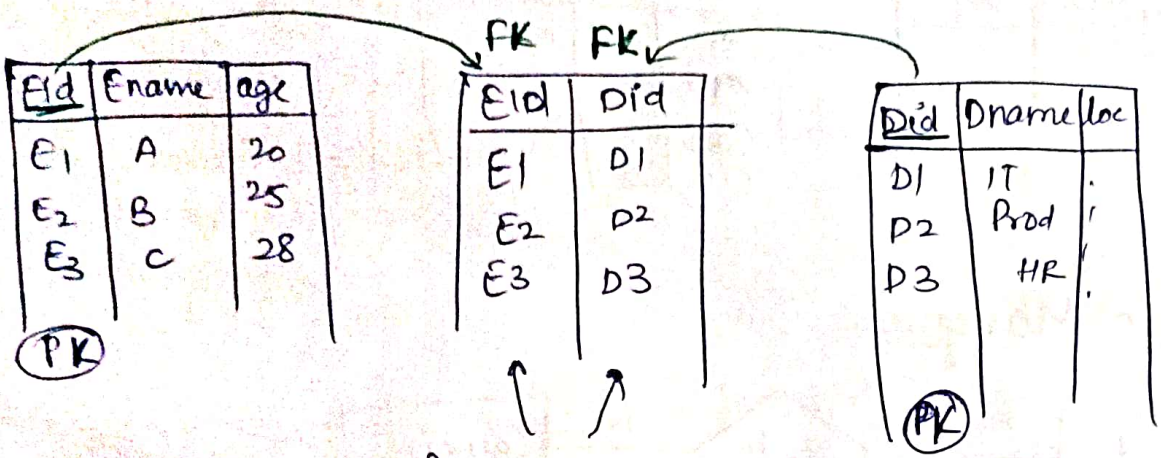
6) Complex Attribute

(Composite + Multivalued)

Degree of Relationship (Cardinality)

One to One :-



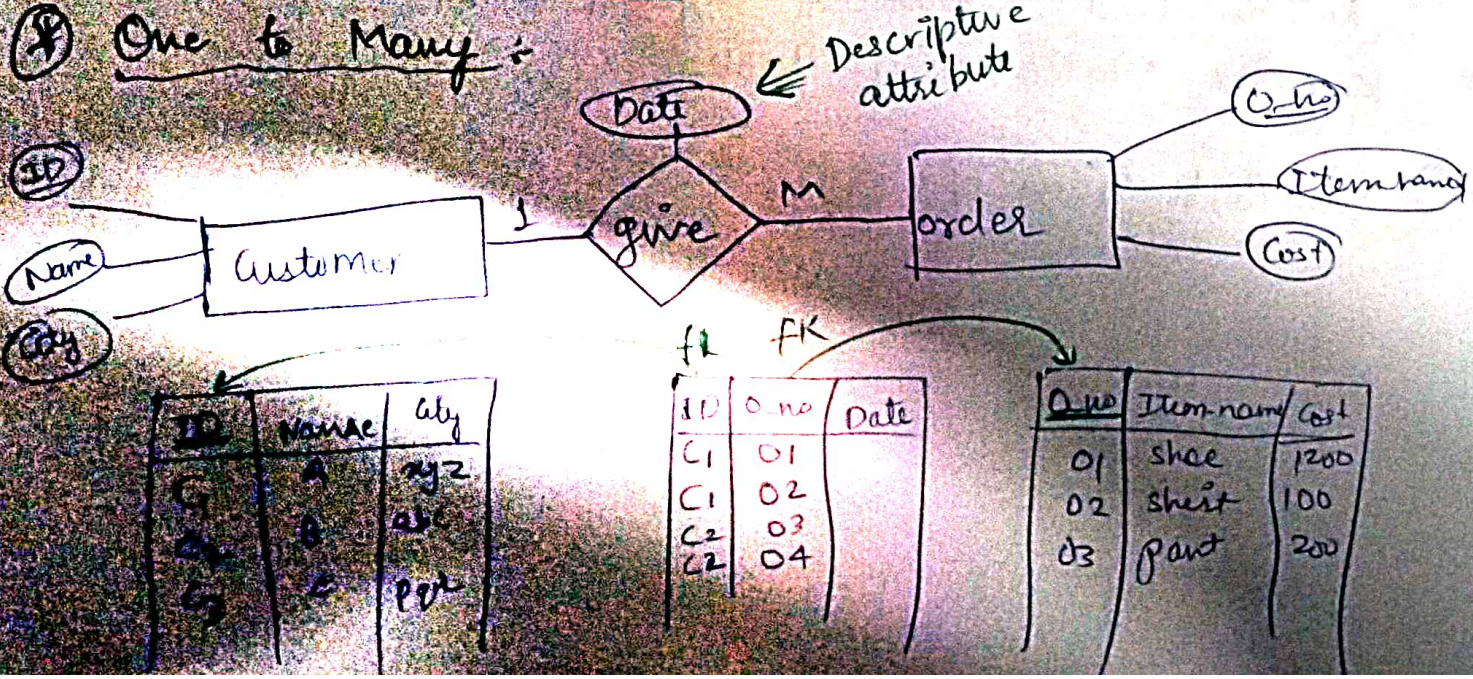


These tables can be reduced

Eid	Ename	age	Did
E1	A	20	D1
E2	B	25	D2
E3	C	28	D3

Did	Dname	loc
D1	IT	
D2	Prod	
D3	HR	

One to Many :-



→ primary key will be O_no in this example

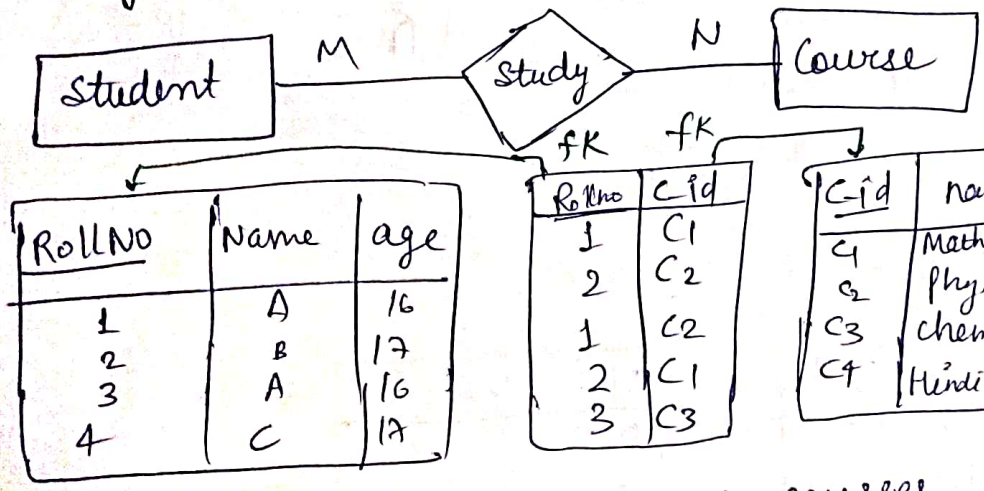
* The primary key to the side of many ~~is~~ acts as the primary key for relationship table.

* Tables can be reduced :-

ID	Name	city

O_No	ID	item name	Cost
O1	C1		
O2	C1		
O3	C2		
O4	C2		

* Many to Many.



RollNO	Name	age
1	A	16
2	B	17
3	A	16
4	C	17

Rollno	C_id
1	C1
2	C2
1	C2
2	C1
3	C3

C_id	name	Credit
C1	Maths	4
C2	Phy.	4
C3	chem.	4
C4	Hindi	4

→ 1 student can study many courses
1 course can be studied by many students

* (Rollno, c_id) Combined together acts as primary key for relationship table.
Composite key

* No Reduction of tables

Normalisation :

It is a technique to remove or Reduce redundancy from a table.

SID	Sname	Age
1	RAM	20
2	Varun	25
1	RAM	20

Row level duplicacy

→ Can be handled with primary key

SID	Sname	Cid	Cname	Fid	Fname	Salary
1	RAM	C1	DBMS	F1	John	30000
2	Ravi	C2 &	JAVA	F2	Bob	40000
3	Nitin	C1	DBMS	F1	John	30000
4	Amit	C1	DBMS	F1	John	30000

Column level duplicacy

1) Insert Anamoly
2) Deletion Anamoly
3) Updation anamoly

First Normal form :

Table should not contain any multivalued Attribute

Rollno	Name	Course
1	Sai	C/C++
2	Harsh	Java
3	Onkar	C/DBMS

→ X

②

Rollno	Name	Course1	Course2
1	Sai	C	C++
2	Harsh	Java	NULL
3	Onkar	C	DBMS

Course

Null

drawback

①

Rollno	Name	Course
1	Sai	C
1	Sai	C++
2	Harsh	Java
3	Onkar	C
3	Onkar	DBMS

③

Roll no	Name
1	Sai
2	Harsh
3	Onkar

Roll no	Course
1	C
1	C++
2	Java
3	C
3	DBMS