Jai Shree Ram.

Name : Anurag Singhal

Standard : ..... Division : ..... Roll : .....

Subject : DBMS

INDEX

(Live SQL).

E.F. Codd → Father of DBMS

1. Database Management System (DBMS) :→

→ Basic Introduc° → 2 tier, 3 tier, 3 schema, (3 level of abstraction).
sup.
↓
(comes in Data Independence)

→ Various Data models :→
Network, Hierarchical,
(Relational, ER), object oriented.

(DBMS) or (RDBMS). — same.
↓
(Relational).

→ ER MODEL :→
conceptual. (Entity - Relationship)

Attributes & its types,
Relationship —" — sup.

sup.
→ Basics of keys :→
- primary key & its characteristic.
Candidate key —" —
Super key —" —

&
Foreign key —" —

→ Normalization →
closure Method → to find candidate key
functional dependencies
1st NF (normal form), 2NF,
3NF,
BCNF

→ Transaction Control & Concurrency →
ACID properties.
R-W problem (Read-write).
W-R "
W-W "
Conflict Serializability.
Recoverability.

Concurrency → locks.
2-PL, timeStamp. (2 phase lock).

→ SQL & Relational algebra.

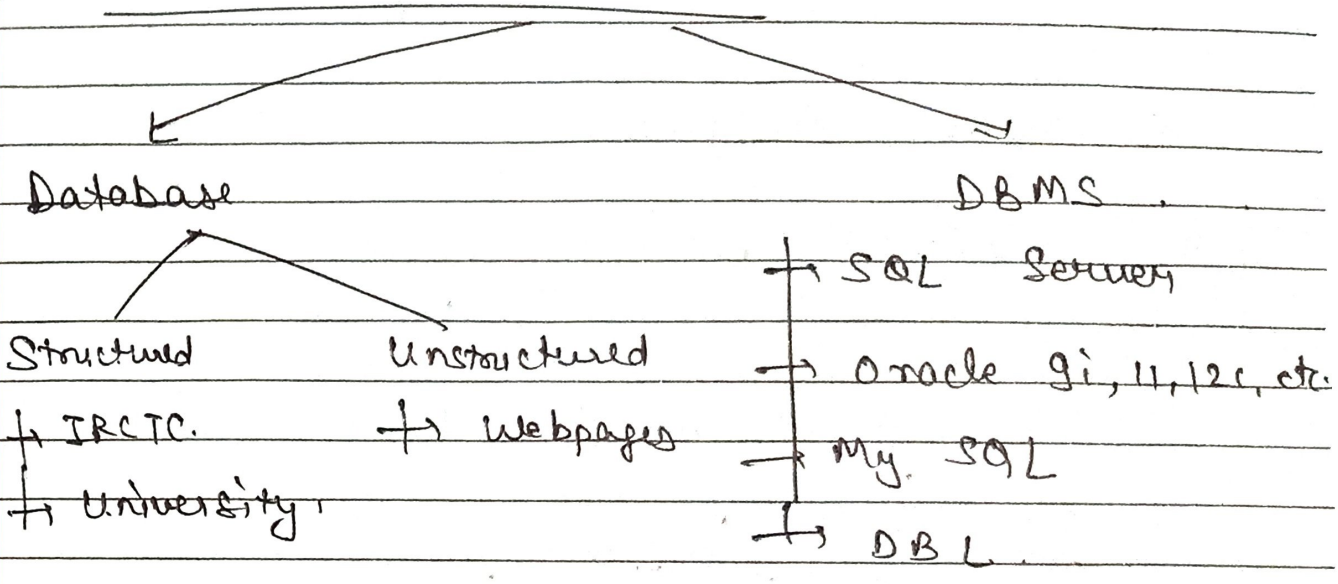
SQL → Structured Query language.
(It is a programming lang.).

DDL command. (Data-Definition).
DML (Data-Manipulation).
DCL
→ Constraint.
→ Aggregate func.
→ Joins
→ Nested Query.
→ In, Not in, any, all.

→ Indexing: → (Single level Indexing).
→ primary, cluster & Secondary Indexing.

→ B tree, B+ tree. (in Multi level).

(2) Database System :→

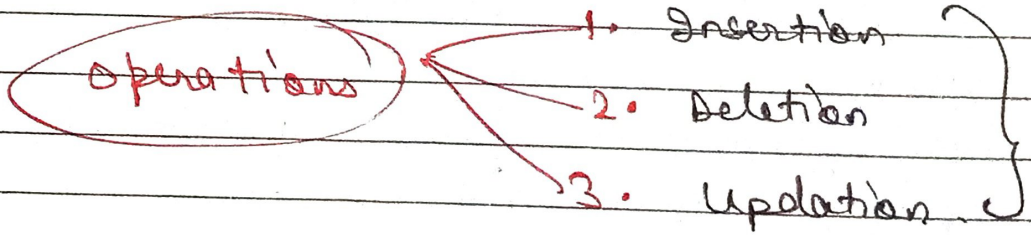


=) Database :→ Collec" of Related Data.

Ex :- Indian Railways & Indian passport have their different data.

=) Structured :→
RDBMS → Rela'al Database Management Sys"

=) DBMS :→ collection of operations



It provide easyness to perform opera's.

=) Diff. companies have made diff. DBMs.

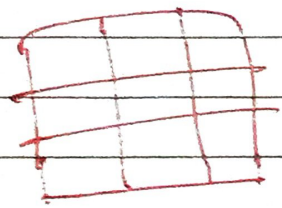
Ex :-
Microsoft Co. ⇒ SQL Server.
(Sequal Server).

⇒ Oracle ⇒. 9i, 11, 12 c, etc.
My SQL.

⇒ IBM ⇒. DBL.

(1) Structured Data! → RDBMS
↳ Relation.

Mean,
We stored in the table form.


→ (Table).

Now, Table is technically called as Rela".

→ Rela" is most usable form.

Now
→ Store Rela"s & access Rela"s is done by the Management System i.e. RDBMS.

⇒ When it is used for Relations, it is called as RDBMS.

Hence,
→ We perform Insert, Delete & updation opera"s on Rela"s.

(2) Unstructured Data! → There is not predefined Structure Ex! →
A webpage is a collec" of photos, videos, Chats, etc. i.e.
There is not a particular format in these.

⇒ Hence, RDBMS works only on the structured Data.

Note:→ 90% of data on this Earth is unstructured.

⇒ ∴ There is more technologies on unstructured data.

Ex:→ Bigdata, Hadoop, etc.

---

3. File System Vs DBMS :→

⇒ File System is used before DBMS.

⇒ User always manages its data in file form.

→ OS has inbuilt File System.

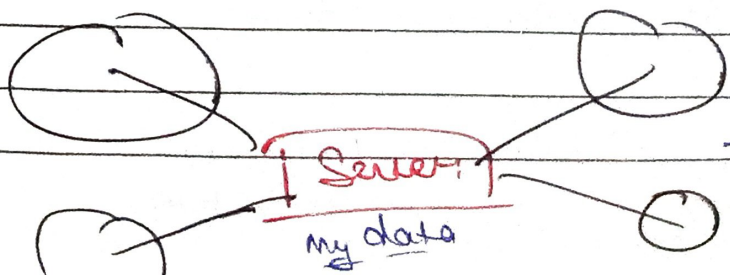
Ex:→ CIFS, NFS File Systems.

→ File System:→ We stores are data in file form. & then into our drives.

→ **Why we use DBMS?**

Bcz, we are using the client-server architecture, means.

→ Our data is at a centralised loca^ & all over the world users can access that.



→ Hence, we can't use file system here. Now, DBMS comes into picture.

1.) If we have to search only for 1 KB, then in file system, complete file comes to us (approx of 25 GB). i.e., More memory usage. Now, DBMS :→ gives us only of 1 KB data from the Server.

(Searching is fast & Memory utilizn is Efficient).

2.) In file system, we require attributes to search data. i.e, Attributes (name, locn, permission, etc.)

Meta Data :→ Data about Data. (Data of a particular file).

In DBMS, no locn, it is totally independent. We don't require any attributes here. (Easeness provide).

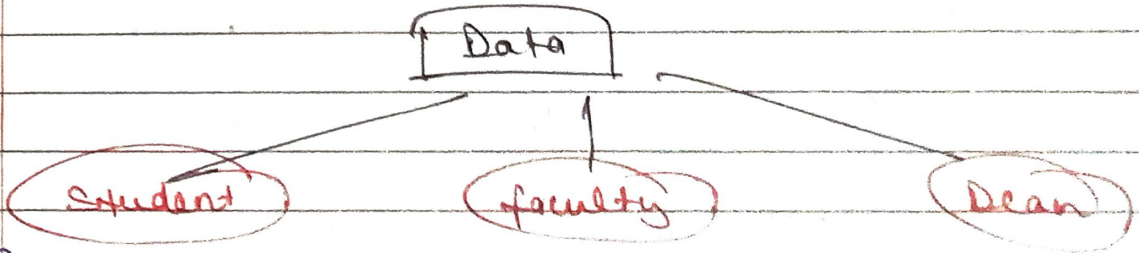
3.) Concurrency :→ means Concurrent Access → Multiple persons can access the data at the same time.

Ex:- IRCTC (Indian Railway System)

File system may show inconsistency when multiple users want access of a file at the same time.
DBMS, have proper protocol for concurrency.

RR
RW
WR
WW
(Read - write)

4.) Security :→ Role based security



(Role based Access Control.)

If we are users, we only get user data.
" - faculty, " - faculty
(O.S.)
File system, has no security for this.
No level-by-level. No Hierarchical.
DBMS has this role-based security.

5.) Data Redundancy :→ (Duplicacy).

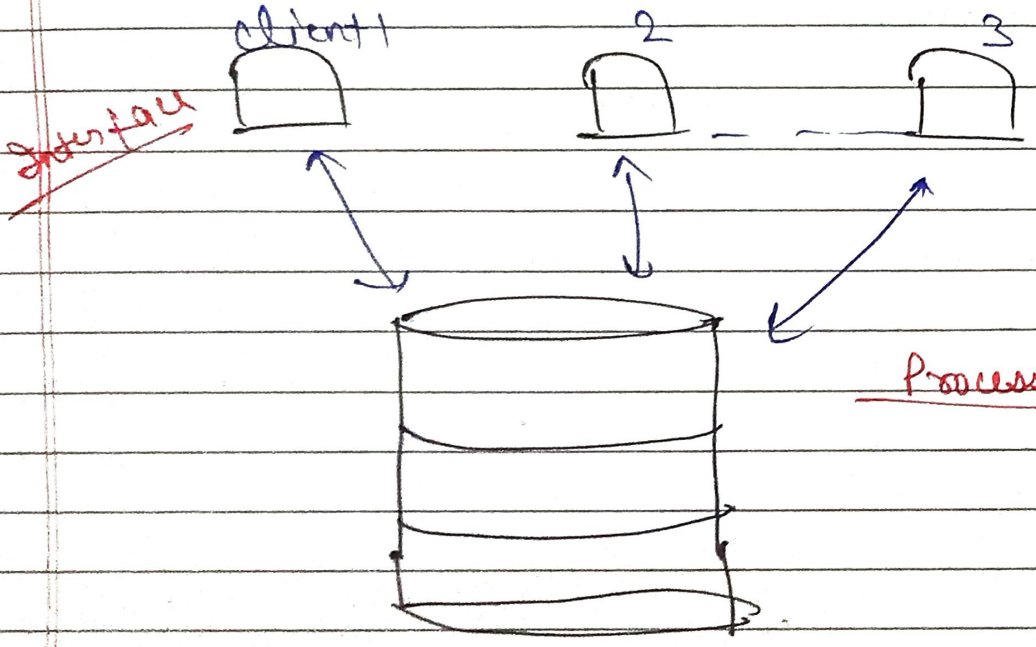
In file system, we can save same content by diff file names.
In DBMS, has many constraints to ensure unique data. Stops redundancy of data.

→ DBMS is at back end of every client server & web apps.

4. 2 tier & 3 tier architecture in DBMS :-

(i) 2 tier :- means 2 layers.

1. client (machine) layer.
2. Database Server. (Data layer).



→ 2 tier also called as client-Server Architecture.

Ex:- i) Indian Railway :- If we reserve ticket by going to railway Stn at ticket window.

ii) Bank :- When physically we draw or post some money.

(client → Request → Database Server → finds info)

→ Hence, limited clients & limited Database to which we access. i.e. why maintenance is very Easy.

But, the users are not limited. They are in big nos. Then, this 2 tier system fails. We call it Scalability.

Security :- → not gud, bcz clients are directly interact with the database.

Advantage :- Maintenance is Easy.

(#) 3-tier :- → (Now, mostly uses).

1.) client layer
2.) Business layer
3.) Data layer.

→ In 2-tier, Query is process in Data-base Server.

But,

→ here, Business layer supports interface. Our query is process in Business layer.

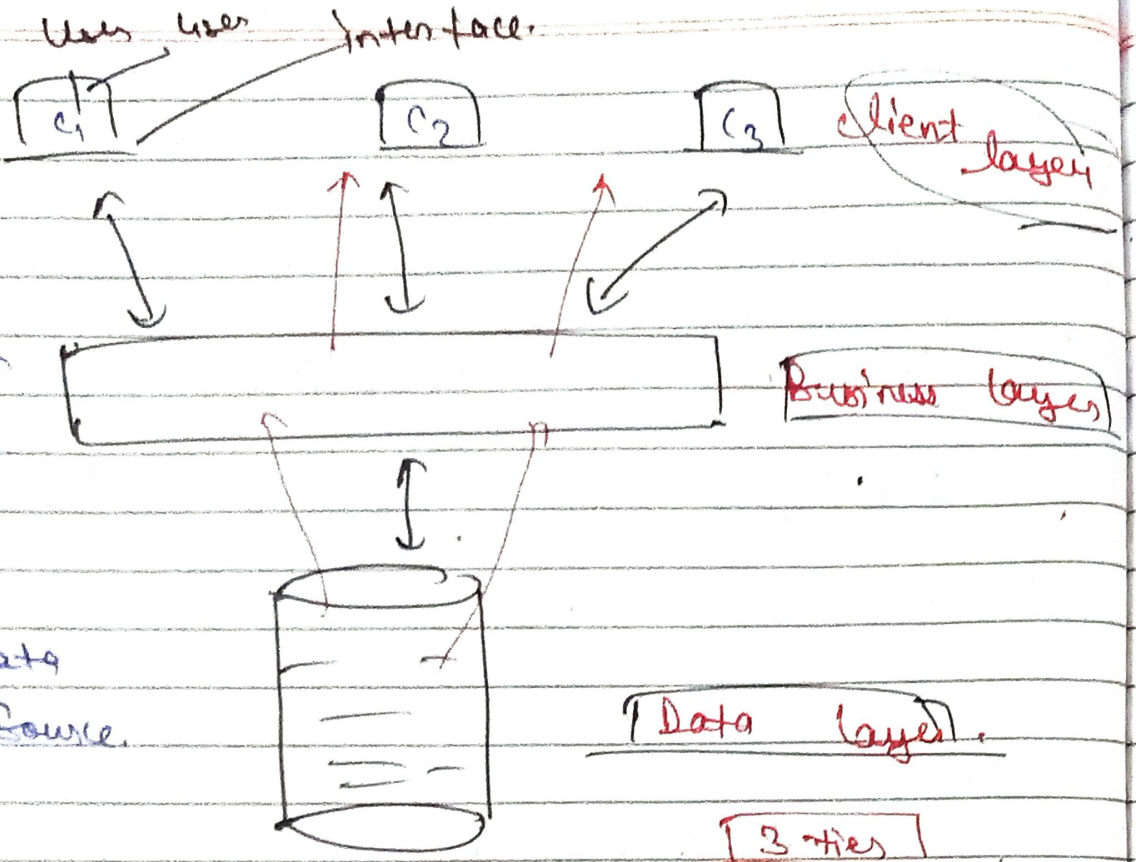
Hence,

We don't give load to Database Server here.

Hence, Business layer acts as Intermediate.

Q:- IRCTC & banking app & Gmail app

Advantage :- 1) Scalability.
2) Security. (no direct interact of data & user)



Here, Maintence is not Easy bcz, it is Complex.

Ex: Web Applica's (APPS) are kind of 3-tier Architecture.

If we go physically to any bank or Railway Sta", then it is 2-tier Architecture.

Schema → Logical Representa" of a database.

⇒ Ex: In RDBMS, data is stored in the form of Tables. (Rela's).

DBMS access & manages the data in this Schema (table) form. It is not actually stored in this table form in drives.

Ex:(i) Schema of a Student :- (Entity) → E-R

Roll. No.	name	addresses

↓ Relationships.

(ii) Course :- (Schema, Entity).

C. ID	name	Dura^n

↓ (Relation ship)

↳ logical Representation (structure).

→ But, we implement it by SQL. (Sequel).
(Integers, characters - - -).
Data Defi^n language (DDL) → to implement Schema. (Or to design)

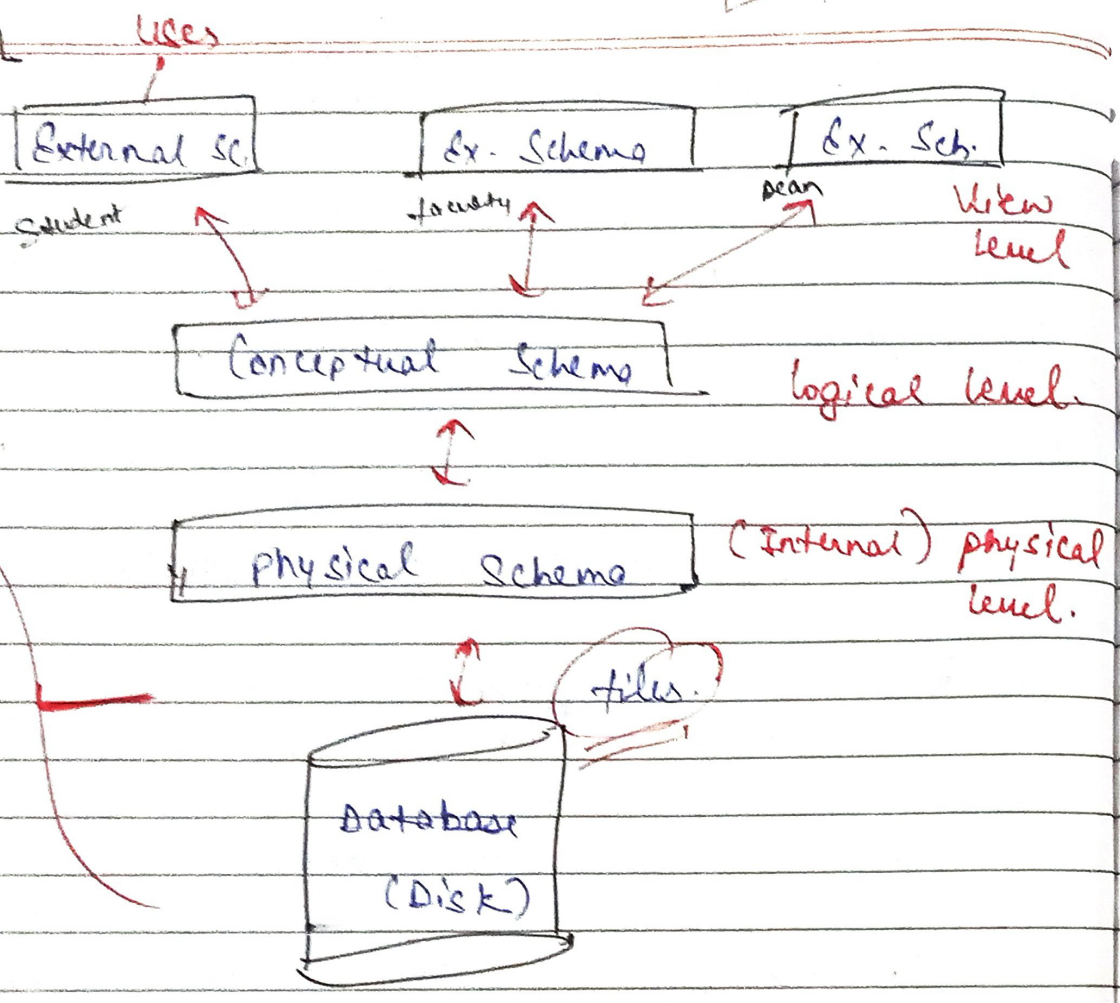
Schema is simply a structure (table form).

---

6. (Three Level of Abstraction) (or) (Three Schema Arch^n.)

→ (We hide the loca^n of where the data is stored, from the user)

Ex:- Our Mails, we don't know physically (exact loca^n) where our mails are stored. Ex- Delhi, UK, US, - - - etc.



External Sch: (View Level) :→ View that will be given to the User.

{ Students have their View.
Faculty have —"— . }.

Ex:→ After Login, which View comes to us is External Schema.

Conceptual Sch: E-R Model

Ex: Student (Roll No, age, addr, ----)

⇒ Info^n of all tables that we use, & their Relo^ship. A type of Blueprint is this.

physical Schema :→ where the data is actually physically present. The loca^ of data.

→ A Normal Data Base Designer is working at level of Conceptual Schema.

→ front End or Interface Developers are at level of Ext. Schema.

→ Database Administrater (who has all control of data) is at physical Schema.

⇒ Centralised — Data at one place.
Multiple — Data all over the World.

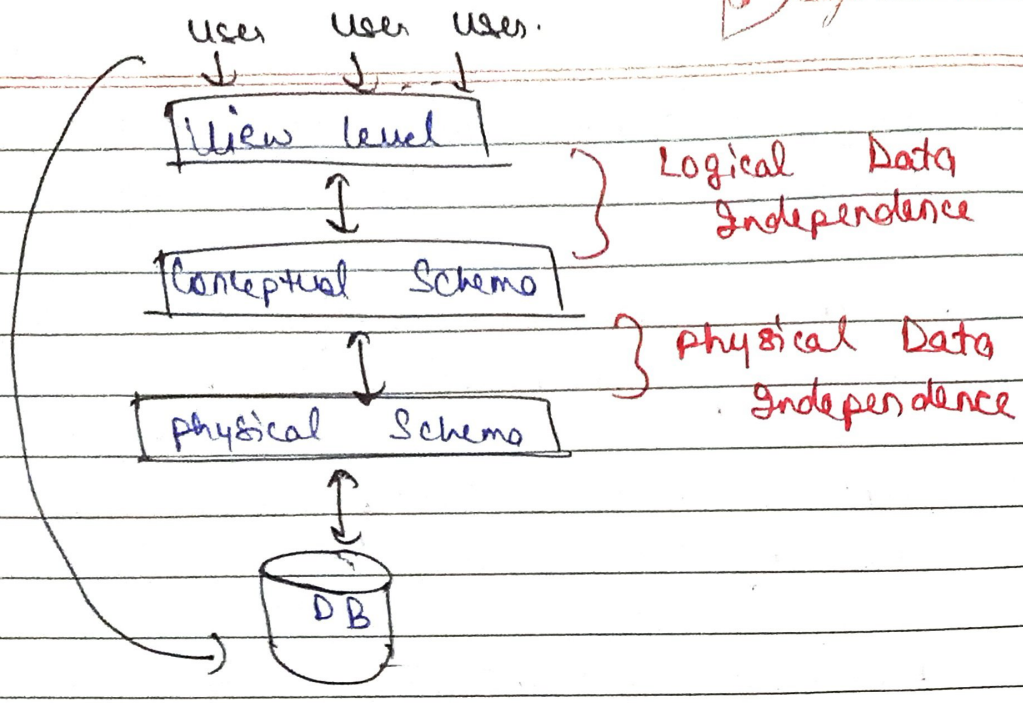
Note:) When we see the data as a user, then we see it in Table form. But, Actually in hard disk, data is stored as files and we apply the layer of DBMS on it.

There are 3 layers b/w the user & the data

7. Data INDEPENDENCE :→

→ Make user independence of data. Hide its loca^ & etc.

→ Conceptual Schema :→ which table we use, how many tables are there, how many attributes, relationship b/w them.



(#) Logical Data Independence! →

Let

→ Student Table. → add by user 1.

Name	Age	Mob. No.

It will not affect the Applica^n program.

→ We don't need to write the App^n pgm again. What user 1 changes want, we give him. But, It will not affect the actual logical structure.

mean, User 2 still can see only col 1 & col 2. Mob No. Colu^m is only ∀ user 1.

→ We use this concept, by Views. (Virtual Table).

→ In actual Table, may be we have 50 col's but user can only see 5-7 col's. So, user think there is only 5-7 col's. But, there are many.

→ Hence, View level don't change by users changes in Conceptual Schema.

Ex:- UMS (University Management System), Shopping website → They can add or delete any col's.

Hence,
It is Logical Data Independence.

physical Data Independence :→ Any change in physical ~~Data Independence~~ Schema won't affect our Conceptual Schema.

Ex:- If we take data from Hard Disk 1 to 2, then data not changes. Tables & Structures remains the same.

Any change in Back End, won't affect the user. It is Data Independence.

---

8) Candidate key & Primary key :→

⇒ key :→ It is one of the attribute in the table.

Use of key: To uniquely identify any 2 tuples in the table.

Roll.No.	S.name	City	Age
1	Reddy	Shamli	20
2	Anurag	Kanpur	21
3	Reddy	Shamli	20

1.) Same 1 student repeats or 2 diff. students with same attri.

2.) 2

4 To identify this, we must have a key.

Ex:) Student Table

1. Aadhar Card
2. Roll No.
3. Reg. No.
4. License No.
5. Voter Id
6. Phone No.
7. Email - Id.

These all attributes can uniquely identify any 2 rows.

The set of all 1-7 values. If we make a set of all of them. Then we call it Candidate key.

{Aadhar Card, Roll No, Reg No ——— Email-Id}.

Now, from above Candidate key set, we choose the one most appropriate & call it Primary key & rest all keys known as Alternative keys.

---

9. PRIMARY KEY :→

(Every lock has its unique key.)

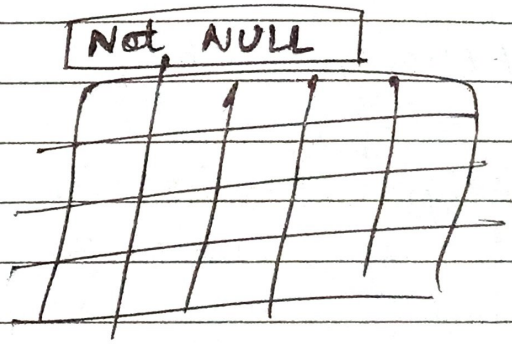
i.e.
We uniquely identify 2 things.

Ex:) Any 2 student can have same name, age, D.O.B. but, some attributes like phone No., Aadhar Card —— are always unique

⇒ Candidate keys :- These are Unique.

→ phone no
→ aadhar card
→ Pan
→ Reg no.
→ Roll no.

} Candidate Keys.



→ phone no, aadhar card vagera को NULL नहीं रख सकते। भरना ही पड़ेगा।

Case 1: We fill wrong aadhar card no.
Case 2: We don't take admn without aadhar card no.

In student case, most appropriate key is
Reg no.
Roll no.

→ PRimary key = { unique + NOT NULL }

→ We don't give primary key to them, they give to us.

Ex :- university give us Reg. No.
passport office give us passport No.
License office give us License No.

- We have only 1 primary key in a Database. Software don't allow us this.

(why need 2 or more, when we only need 1)

---

10 **Foreign key in DBMS** :→

(F.k.)

- **Foreign key** :→ It is an attribute or set of attributes that references to primary key of same table or another table (relation).

→ It Maintains referential Integrity.

Ex:- Student →

P.K (primary key).	Roll no	name	Add.
	1	A	Delhi
	2	B	Shamli
	3	A	Mumbai

Course →

(F.k.)

C. ID	C. name	Roll. No.
C_1	DBMS	1
C_2	networks	2

→ Roll. No is same b/w the student & course. It shows Rela^ship b/w them.

→ In Table 2, Roll No. col^n value ^ references from Roll no. attribute (primary key) in Table 1. take

Q! Can I write Roll. No. '10' in Foreign key (F.k.)?

→ No, Sir, It is not in Table 1 (Roll. No) Until now.

(with f.k.)

⇒ Table 2 is called Referencing Table.

⇒ Table 1 is called Referenced Table. (with p.k.). or Base Table.

* Create table Course
(
Course_id varchar (10),
Course_name varchar (20),
Rollno int references
Student (Roll no).
);

⇒ Code, Ex.

↳ Now, how to write a query after the table is created.

⇒.
Alter table Course
ADD constraint f.k.
foreign key (Roll no)
references Student (Roll no);

Note: (1.) We don't have to keep the name 'Roll. No.' same of both the attributes necessarily, we can also take diff. names.
(2.) In a table, there can be more than 1 foreign key.

(11.) Foreign key :- (Referential Integrity) :-

⇒ Integrity means same value for the database