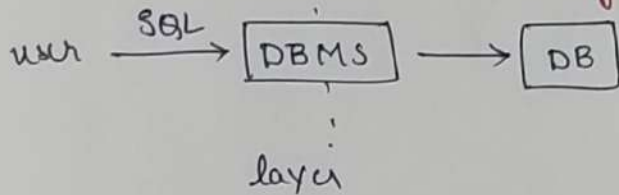


# SQL

① Database: Collection of data in a format that can be easily Accessed.

→ A SQL application — used to manage our DB =, DataBase Management System



RDBMS  
Relational

Data — tables

MySQL

ORACLE

PostgreSQL

Non-relational

Not stored in Tables

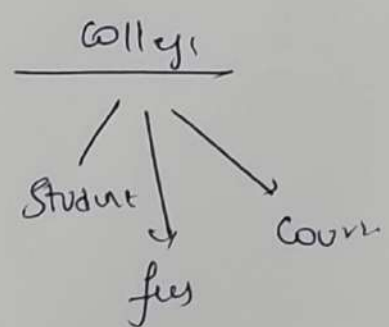
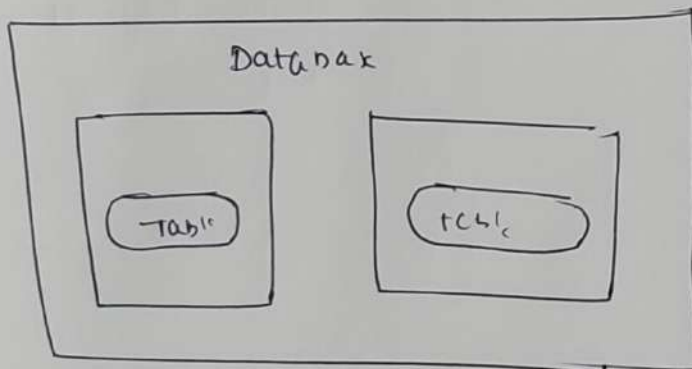
(NoSQL)

MongoDB

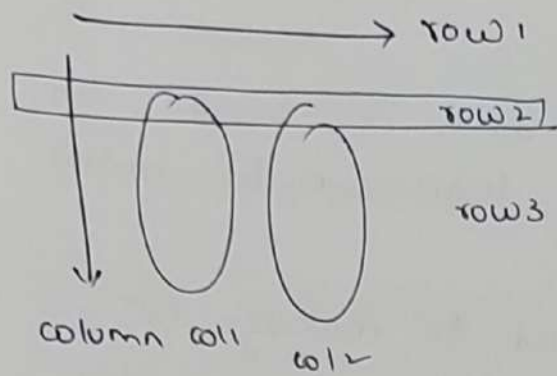
SQL → Structured Query Language.  
used to → interact with RDBS

CRUD      SEQUEL → Structured English Query Language. } IBM

Database struct



Inter Related data



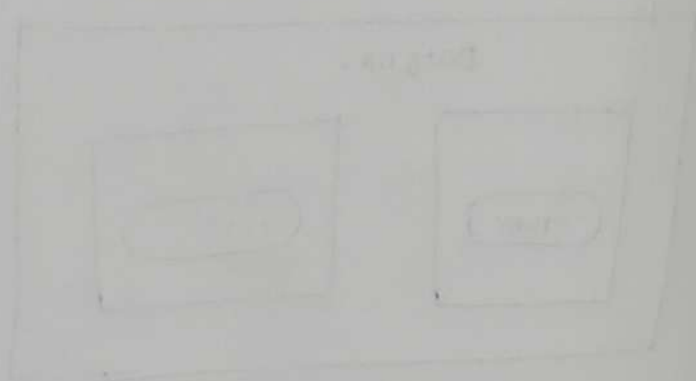
Columns  $\rightarrow$  Schema/  
design

row  $\rightarrow$  Individual  
Data

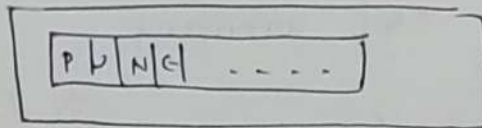
Create database \_\_\_\_\_;

Drop database \_\_\_\_\_;

29:43

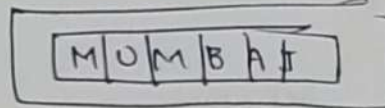


col inefficient  
char (50)



col varchar (50)

(0-255) string



BLOB - BLOB (1000) (0-65535)

INT (-2147483648 to 2147483647) INT

TINYINT -128 to 127

BIGINT

BIT(2) — 2 bit value  
00  
01  
10  
11

FLOAT 23 digit

DOUBLE 24-53 digit

BOOLEAN => TINYINT 20000 implement

Date yyyy-MM-DD

YEAR

### Signed & Unsigned

numeric

-ve & +ve  
age  
salary

128 127  
TINYINT (-128 to 127)

TINYINT UNSIGNED (0 to 255)

## Types of SQL commands

### ① Data Def Lan

Create  
Alter  
Rename  
Truncate  
drop

### ② Data Query lan

Select

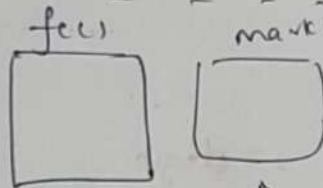
### ③ Data Manipulation lan

Insert  
Update  
Delete

### ⑤ Transaction Control Language

start transaction  
Commit  
Rollback

### ④ Data Control Language



Grant & Revoke  
Permissions

create database db-name

IF NOT EXISTS

Drop Database IF exists

Primary key  $\begin{cases} \text{Unique} \\ \text{Not NULL} \end{cases}$

insert into student

values (1, "Neer"),

(2, "Ara"),

(3, "kathy");

## key

Primary key:

Column /  
Set of columns

Uniquely identifies  
each row

1 PK  $\neq$  NULL

Foreign key

col /  
Set of columns

→ references to the  
PK of another  
table



Multiple Fk

Duplicates Allowed

Null also Allowed

## Constraint

col name    data-type    constraints

NOT NULL

UNIQUE

Unique  
pk

← Primary key

foreign key (    ) references customer(id)

Default

Salary INT Default 25000

def val

CHECK

id	Salary	
	25000	
	28000	
	25000	



constraint ——— CHECK (age >= 18 and  
city = 'Delhi')

⑥ age int check (age >= 18)

### Select

select col1, col2 from table

select distinct city from student;

Where Clause → condition

&

Arithmetic

+

-

\*

/

%

Compari.

=

!=

>

>=

<

<=

Logical op

AND

OR

NOT

IN

BETWEEN

ALL

LIKE

ANY

range

where city IN ('Delhi',  
'Mumbai')

[80, 90]

★ Inclusive

NOT in (

ଅବସ୍ଥା ଦେଖିବା

Limit

Select \* from stud

ORDER BY

order by city ASC  
DESC

Aggregate functions

count()  
max()  
avg()  
sum()  
min()

Calculation on  
set of values &  
return 1 value

1 single value

Group By  $\Rightarrow$   
same values

select city, count(name)

select

Aggregate function

from student  
group by city;

Having

↓

Group

↓

rows

Having

where  
↓  
rows

Having  
↓  
group

Select }  
from }  
where }

cond

Group By

Having

Order by AS

cond

Table related

Update existing row

SET SQL\_SAFE\_  
UPDATES = 0

update table-name

set col = val, col = val

where grade = "A"

Delete rows

Foreign Key: