

Content → confidence → communication

SQL

→ Commands:-

① Data Definition Lang (DDL):- (adder)

- create
- alter
- drop
- truncate
- Rename

deals with schema, deals with creation of structure
eg. courier box

② Data Manipulation Lang (DML):- (sudi)

- Select
- Insert
- Update
- Del

eg. save student info

③ Data Control Lang (DCL)

- Grant
- Revoke

④ Transaction Control Lang:- (TCL)

- Commit
- Rollback
- Savepoint

* Constraints:- To insert data there are some rules

→ Create table :-

① create table snc (int roll (10), varchar name (20));

→ Alter statement :-

① alter table snc add (int age (2));

② modify (age (3));

③ drop column (age);

④ rename column roll to id;

→ alter and update diff :-

alter

① DDL

② add, modify, drop, rename
order

update

① DML

② can't chg structure only
can chg data

③ set, where

| → Del | Drop | Truncate |
|---|--------------|---|
| ① DDL DML | DDL | DDL |
| ② when u want to del table data (eg now) | ③ struct del | - same like del - but del all rows in one go. |
| ③ slow | | - fast |
| ④ rollback ✓ but b/c commit | | - X R.B not possi |

- ⑧ candidate key :- nominee of P.K
 ⑨ super :- set of attri, including P.K

→ Unique constraints :-

- ① unique :- duplicate val shldn't exist in same col
- ② not null :- cant be empty
- ③ primary :- 1+2
- ④ check :- adding constraints eg age > 18
- ⑤ foreign :- reference, refers to P.K in another table. maintain referential integrity.
- ⑥ default :- by default it takes default val if u don't insert
- ⑦ composite :- P.K jo 2 attr se milke banata hai
 or
 two or more attri which helps to uniquely identify a table

→ highest

- ① select max(sal) from emp;
- ② select name from emp where sal = (select ^{max(sal)} * from emp);

→ second highest

- not written den will display all den max sal
- ① select max(sal) from emp where ~~sal~~ < (select max(sal) from emp);

- ② select emp name from emp where sal = (1);

→ dept name and no. of emp working in it

Select dept, count(*) from emp group by dept;



if u want to use directly u can only write what there of group by
 one attr or col and one funst

→ dept name where no. of emp less than 2

select dept from emp group by dept having count(*) < 2;

→ highest sal dept wise n name of emp taking dat sal

select emp name from emp where sal IN (select max(sal) from emp group by dept);

→ for comparison if v hv more than one val v dont use equal to use IN

→ find emp name working on a project. (P.K, F.K)
In / nt is

select ~~emp~~ name from emp where eid in (select distinct eid from project)

↓
common in both table

→ exist nt exists

emp who is working on at least one proj skip

from worker mod (worker-id, 2) = 0;

→ max, min, count, sum, avg

→ Diff nested, correlated, joins

① nested:- Bottom up, select * from emp where eid in (select eid from dept);
Outer + Inner

② correlated:- top down approach bahar wala pehla row uthake andar wala same row se compare karringe

③ join:- ~~cross~~ cross product + cond

→ Nth highest salary:-

① select * from worker order by desc limit (1, 1); n=5
top 5 = (5) (1, 2) n=5, 6

(N-1, 1)

→ ① display last name of emp 'A' as second char

'A%' → aft dis any no. char
↓
srf one char bef a

eg: - - - where last name like 'A%';

② command to remove rel from SQL database
drop table <tablename>

③

→ Char functions :-

① lower = select lower (name) emp ;

② upper = upper (name);

③ initcap = initcap (name);

→ Diff DBMS and RDBMS :-

① DBMS :- software, hw & manage database in diff form,
✓ can store data base in form of graph, table
and doc, file

② RDBMS :- storing data in relational or table form
eg. mysql, sqlserver, oracle,

→ Diff group by and order by :-

① group by :- HR eg. ~~se~~ to group and then apply aggregate func.

② order by :- sort data in asc or desc

→ Types of Join :-

① cross, inner, left, right, full

→ SQL vs MySQL :-

↳ relational database
query prog lang, manages DBMS

↳ ~~non~~ relational database, uses SQL
management sys

→ varchar and varchar2

↳ ANSI std

↳ oracle std

→ Triggers:-

special type stored procedure that runs automatically when event occurs.

eg: v made 3/4 table, I want to enter one table n baaki mein bhi automatically enter ho.

→ ACID properties

① atomicity:- either all d chgs r performed or none of them

② consistency:- data is in consistent state when transaction starts and ends

③ Isolation:- Trans that run concurrently appear to be serialised

④ Durability:- trans successfully comp then it shld be permanent

1) either all or none = if transaction fail before commit v will roll back eg. paytm transaction

2) Bef trans start and ~~and~~ off end sum of money shld be same.

eg: $A \rightarrow B$ $A = 200$ $B = 300$

$R(A) = 100$

$A = A - 100$

$W(A) = 100$

$R(B) = 300$

$B = B + 100$

$W(B) = 400$ commit

} local mem then shift to database

$$\begin{aligned} \text{Bef} &= A + B \\ &= 500 \end{aligned}$$

$$\begin{aligned} \text{After} &= A + B \\ &= 500 \end{aligned}$$

3) parallel schd to serial convert \rightarrow consistent

4) d chgs v do in database shld be permanent

\rightarrow Join:- Select

① Natural Join:- dono table ke beech common attri ko eq rakhna

eg.

| eno. | ename | add |
|------|-------|-------|
| 1 | Ram | delhi |
| 2 | Varun | chd |
| 3 | Ravi | chd |
| 4 | Amrit | delhi |

| dept no. | name | eno. |
|----------|--------|------|
| d1 | HR | 1 |
| d2 | IT | 2 |
| d3 | market | 4 |

g. find ename who is working in a dept

$4 \times 3 = 12$ rows

Select ename from emp, dept where emp.eno. = dept.eno; (yeh karra niche wala)

\downarrow
select ename from emp natural join dept

② Self Join:- in which table is join wid itself

eg.

| Sid | Cid | Since |
|-----|-----|-------|
| S1 | C1 | 2016 |
| S2 | C2 | 2017 |
| S1 | C2 | 2017 |

g. find student who enrolled in at least 2 courses

select T1.sid from study as T1, study as T2
where T1.sid = T2.sid and T1.cid \neq T2.cid

select, from, where, group by, having, order by

→ **Equi Join** :- $=$ can put equal b/w any of 2 attri

eg.

| Empno | ename | addr |
|-------|---------|-------|
| 1 | Ram | delhi |
| 2 | Varun | del |
| 3 | Ravi | del |
| 4 | Aravind | delhi |

| Deptno | loc | Empno |
|--------|-------|-------|
| D1 | delhi | 1 |
| D2 | Pune | 2 |
| D3 | Patna | 4 |

Q. find the emp name who work in a dept having loc same as their addr?

Select ename from emp, dept where emp.empno = dept.empno and emp.addr = dept.loc;

→ **Left outer join** :-

gives matching rows or data rows which are in left table but not in right table



eg.

| Empno | ename | deptno |
|-------|-------|--------|
| E1 | varun | D1 |
| E2 | arav | D2 |
| E3 | ravi | D1 |
| E4 | richa | - |

| deptno | dname | loc |
|--------|---------|-------|
| D1 | IT | delhi |
| D2 | HR | hyd |
| D3 | Finance | Pune |

Select empno, ename, dname, loc from emp left outer join dept
on (emp.deptno = dept.deptno),

E1, E2, E3, E4 all will come

→ Right outer join :-

select ename, empno, dname, loc from emp right outer join dept on
(emp.deptno = dept.deptno);

→ Normalization :-

technique to remove duplicacy and reduce redundancy

• Anomaly :-

- 1) Insertⁿ - if v insert a new course other blank
- 2) Deletion - if v del a row eg. Java which is unique
- 3) updatation - if v update for data v need to upd all fi

• 1NF = atomic, single value attribute

• 2NF = non prime key shdn't depend on partially key on
candidate key.

• 3NF = no non key attri is transitively dependent on key

key attr :- parts of C.K

non key :- not part of C.K

→ Three layer archi dbms :-

external level - describe part of database in which the user is intd.

conceptual level - data is represented in the form of database table

Internal level /
physical - describes how the data is being store
in secondary storage.