SQL

→ Commands :-

① Data Defination Lang (DDL) :- (cadtr)

- → 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

* Contraints :- To insert data there r some rules

→ Create table :-

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

→ Alter statement :-

① alter table sre add (int age (2));

② modify (age(3));

③ drop colum (age);

④ rename column roll to id;

→ alter and update diff :-

alter	update
① DDL	① DML
② add, modify, drop, rename amdr	② cant chg structure only can chg data
	③ set, where

→ Del	Drop	Truncate
① ~~DDL~~ DML	DDL	DDL
② when v want to del table data (eg row)	② struct del	- same like del - but del all rows in one go.
③ slow		- fast
④ rollback✓ but bef commit		- × R.B nt possi

→ ~~Unique~~ constraints :-

① unique :- duplicate val shldn't exist is same col

② not null :- cant b 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 dis val if v dont insert

→ highest

① select max(sal) frm emp;

② select name frm emp where sal = (select ~~*~~ max (sal) frm emp);

→ second highest

nt written den will display all dan max sal

① select max (sal) frm emp where ~~col~~ sal <> (select max (sal) frm emp);

② select emp name fraom emp where sal = (①   );

→ dept name and no. of emp working in it

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

↓

if v want to use directly v can only write wht there aft grp by any other or col wid agg funct

→ dept name where no. of emp less dan 2

select dept frm 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 comparision if v hv more dan one val v dont use equal to
use IN

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

select name frm emp where $\underbrace{\text{eid}}$ in (select district eid from project)

↓

common in both table

→ exist nt exists

emp who is working on at least one proj   skip

→ 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 corelated:- top down approach bahar wala pehla row uthake andar wale saare row se compare karenge

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

→ Nth highest salary :-

①

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

'_A%.' → aft dis any no. char

↓

sirf 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 v maonage database in diff form,
v can store data base in form of graph, table
and doc, file

② RDBMS :- storing data in relational or table form
eg: mqsql, SQLserver, oracle.

→ Diff group by and order by :-

① group by :- HR eg. se to group and den 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

↓ non-relational database

→ varchar and varchar2
↳ ANSI std ↳ oracle std

→ Triggers :-

special type stored procedure dat 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 :- Transs dat run concurrently appear to be serialized

④ Durability :- trans successfully comp den 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~~ aft end sum of money shld be same.

eg. A → B A = 200 B = 300 Bef = A+B = 500

R(A) = 100
A = A − 100
W(A) = 100
R(B) = 300 } local mems den shift to database After = A+B = 500
B = B + 100
W(B) = 400 commit