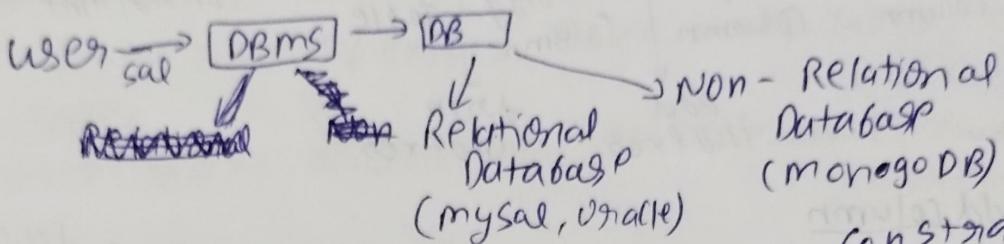


Sal Notes

DB = collection of Data

DBms = software that manage DB



Constraint

- ① Primary Key
- ② NOT NULL
- ③ Unique
- ④ Foreign key
- ⑤ DEFAULT
- ⑥ CHECK
- ⑦ INDEX

| | |
|-------------------|---------------|
| CHAR | → CHAR(50) |
| VARCHAR | → VARCHAR(50) |
| INT | → INT |
| TINYINT | → TINYINT |
| BIT | → BIT(2) |
| FLOAT | → FLOAT |
| DOUBLE | → DOUBLE |
| DATE (YYYY-mm-DD) | → DATE |
| TIME (HH:MM:SS) | → TIME |
| YEAR (YYYY) | → YEAR |

2) Types of SQL commands

DQL (Data Query language) → Select

DDL (Data Definition language) → Create, drop, alter, rename, Truncate

DML (Data manipulation language) → insert, update, delete

DCL (Data control language) → grant, revoke

TCL (Transaction control language) → commit, Rollback, start, transaction, savePoint

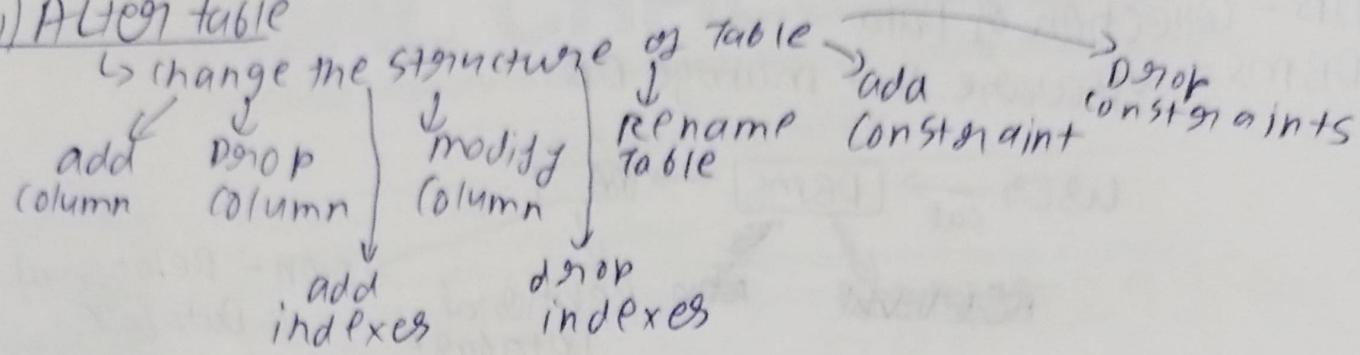
3) DDL

(i) CREATE Table

CREATE table employee table name (

id INT primary key,
name varchar(50), NOT NULL,
salary decimal(10,2));

(ii) A UG91 table



→ Add column

dd column
alter table tablename add column-name datatype
→ alter table employee add email varchar(100) NOT NULL;

→ Drop column

drop column -> file name drop column-name

→ alter table employee drop email;

→ modify column

modify column alter table table-name alter column column-name Desc.
alter column salary decimal(12,2);

→ alter table employee alter column salary DECIMAL(12,2);

→ Renaming a column Table

Renaming a column table
alter table Old-table-name
→ alter table employee rename to emp;

→ alter table employee rename to emp;

→ adding constraints

adding constraints
ALTER TABLE table-name ADD CONSTRAINT constraint-name
constraint-definition

~~constraint~~ ~~constraint~~ ~~constraint~~

→ drop constraint

~~ALTER TABLE table-name~~ drop constraint constraint-name

→ adding indexes

`alter table table-name add index index-name (column-name)`

→ Dropping Indexes

~~alter table table-name drop index index-name;~~

(iii) Drop Table

Drop Table Table-name

→ Drop Table Table-name IF EXISTS Table-name;
→ Drop Table If exists Table1-name, Table2-name;

(iv) Create Index

Create index column-index-name ON table-name(column-name);

(v) Drop Index

Drop index column-index-name

(vi) Truncate (Delete data inside a table)

Truncate table table-name

Create table example with
multiple constraints

Create table table-name (

a INT Primary Key,

b INT NOT NULL

c INT Check (c >= 16),

d INT default 13,

e INT Unique,

Foreign Key (e) references table2(a);

| |
|--------|
| table2 |
| |

Data Query Language

(i) Select select * from table-name

→ select * from table-name where "a" group by "b"

→ select * from table-name where "a" group by "b";

Having "c" order by "d";

→ (a) where (=, >, <, >=, <=, <>)

→ (b) AND, OR, NOT → used by where

→ (c) Distinct → remove Duplicate from row

select Distinct column-name from Table-name.

(d) Like
↳ select * from table-name where name like '%/0/';

examples

- ↳ start with a = 'a%'
- ↳ end with a = '%a'
- ↳ or at any position = '%0%'
- ↳ or at 2nd Position = '_%'
- ↳ start with a, at least = 'a-%'
- ↳ have 2 character
↳ start with a end = 'a%0%'
with 0

(e) IN
select * from Table-name where a IN (1, 2, 3);

(f) Between
select * from table-name where a Between (1 AND 3);

(g) IS NULL
select * from table-name where a IS NULL;

(h) AS
select f-name AS "First Name" from table-name;

(I) ORDER By
↳ return result based on (column) → sorted.

↳ select * from table-name order by column2 ASC; [ESC]

↳ multiple columns
select * from table-name order by column2, column2 ASC;

↳ sorting By Expression
select * from table-name order by price * 1.1 AS "Discount"

select * from table-name ~~order by~~, price * 1.1 AS "Discount"
from table-name order by discount;

↳ sorting NULL values

↳ consider NULL (as smallest and as largest) in
ASC and DESC order

↳ we can control it using [NULL FIRST, NULL LAST]

Select / Column-name from table-name Order By Column-name

NULL Last;

↳ Sorting By Position

Instead of specifying column names, you can sort by column positions in the order by clause.

↳ Select Product-name, Price from Products Order By ?

desc, asc;

(3) Group By

↳ group rows based on one or more columns

Select * from table group by column1, column2;

↳ Aggregation Function

↳ count, sum, AVG, max, min

↳ often used with group by clause

↳ Select count(column2) from group by column 2;

→ Having clause

→ Having clause often used with group by to filter groups based on aggregation function result

→ similar to where clause by for groups

→ Select Avg(salary), department from emp group by department, Having Avg(salary)>100;

→ order by

→ Select Avg(salary), department from emp group by department, Having Avg(salary)>100 Order by department desc;

⑤ Data Manipulation Language

NOTE QUESTION
Join using commas

(i) Insert

Insert into table-name (column₁, ...) value (v₁, ..., v_B);

SELECT *
FROM

tableA,
tableB.

(ii) Update → update existing record

Update employee set salary = 100 where ...;

where A.Id = B.Id;

(iii) Delete → remove record from Table

Delete from employee where last-name = ' ';

Explicit
JOIN

⑥ Joins → Inner, Outer, Cross, Self

Implicit
JOIN

↳ combine rows from 2 or more tables
↳ used to retrieve data from multiple tables

(i) Inner Join

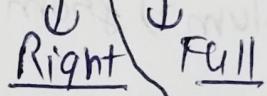


Select column
from tableA
inner join
tableB
on tableA.col-name
= tableB.col-name

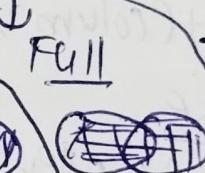
(ii) Outer Join



Select *
from tableA
left join
tableB
on



Select *
from tableB
right join
tableA on
tableA.col-name =
tableB.col-name;



(iii) Cross Join

Join

Select * from tableA
left join tableB on
tableA.col-name = tableB.col-name
Union

Select * from
tableA right join
tableB on tableA.col-name =
tableB.col-name

Left exclusive Join



Select * from tableA
left join tableB on
tableA.column-name =
tableB.column-name
where B.column-name IS NULL;

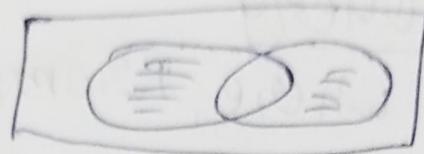
Right exclusive Join



Select * from tableA right join
tableB on tableA.column-name =
tableB.column-name where
A.column-name IS NULL;

FULL EXCLUSIVE JOIN

AIG → Left Exclusive Join



Union
Right Exclusive Join

SPJ JOIN

Select a. name as manager-name, b.name

Select column(s) from table as a join table as b on
a.col-name = b.col-name;

⑦ Set Operations → Union, intersect, Except, Union ALL

- ↳ every select should have same no. of columns
 - ↳ columns must have similar data types
 - ↳ order of column, select should be same.
- Only Then we used it.

(i) Union

↳ return combination of 2 select query.

↳ remove duplicate

Select * from ~~from~~ customer

Union

Select * from supplier

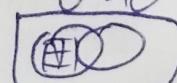
| customer-id | customer-name |
|-------------|---------------|
| 1 | a |
| 2 | b |

| supplier-id | supplier-name |
|-------------|---------------|
| 3 | c |
| 4 | d |

| customer-id | customer-name |
|-------------|---------------|
| 1 | a |
| 2 | b |

(ii) Union ALL → It allow the duplicate

(iii) Except minus → query1 - query2 + remove duplicate



(iv) Intersect →



+ remove duplicate

⑧ SQL Sub Query

↳ Inside where, from, select

(i) where

select (column(s)) from table-name where column-name
= ~~(query)~~ (subquery);

(ii) From

select columns(s) from ~~table-name~~ (subquery) as temp;

(iii) select

select (subquery) from table;

| Non null | Null |
|----------|------|
| 5 | 5 |

| Non null | Null |
|----------|------|
| 5 | 5 |

| Non null | Null |
|----------|------|
| 5 | 5 |