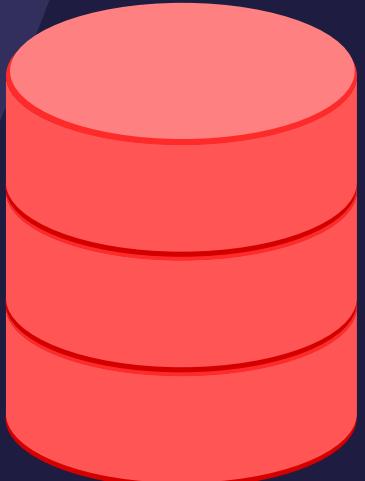
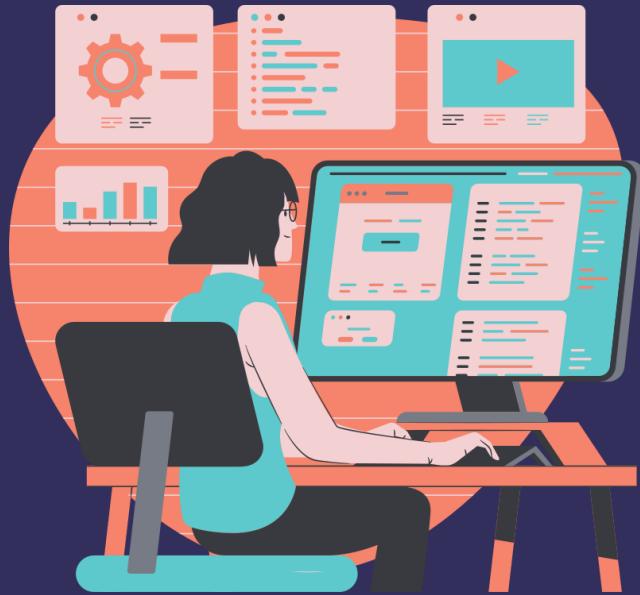


# SQL QUERIES



# SQL

DOMS

Page No.

Date

/

Date

/

[ Structured Query Language ]  
( Case insensitive )

There are four sub-languages

- DDL (Data Definition Language)  
[ create , DROP, ALTER, --- etc ]
- DQL (Data Query Language)  
[ SELECT ]
- DML (Data Manipulation Language)  
[ UPDATE , INSERT , DELETE ]
- Data Control Language  
[ GRANT , REVOKE ]

Eg:- CREATE TABLE student (  
student\_id INT PRIMARY KEY,  
f-name VARCHAR(20) NOT NULL,  
l-name VARCHAR(20)  
);

INSERT INTO student VALUES(1, "xyz", "abc");

INSERT INTO student (student-id , f-name)  
VALUES ( 3, "cde");

→ To add a column

ALTER TABLE ADD address VARCHAR(50)

→ To drop a column

ALTER TABLE DROP address

→ To Retrieve data

SELECT \* FROM student;

SELECT f-name, l-name FROM student

SELECT DISTINCT (f-name) FROM student

SELECT \* FROM student where student-id > 2

\* Data is inserted like this '24-jul-1998'  
DD-MON-YEAR

\* Dynamic Insert (using ampersand)

Values are entered by the user at the execution time.

## Copying DDL

→ all elements from one table to another

~~SELECT~~ INSERT INTO sample2 SELECT \* from sample1;

→ specific column copy

INSERT INTO sample2 sample2(name) SELECT name  
FROM sample1;

→ specific row copy

INSERT INTO sample2 (SELECT \* FROM sample1  
WHERE age = 21);

## SELECT DDL Command

→ SELECT column1, column2...  
FROM table-name;

① SELECT \* from emp  
WHERE dept = 20;

→ Performing some operation arithmetic

SELECT sal\*12 as annualsal from emp;

## INSERT ALL

INSERT ALL

- INTO mytable(column1, column2, column-n)  
VALUES (expr1, expr2, expr-n)
- INTO mytable(column1, column2, column-n)  
VALUES (expr1, expr2, expr-n) ~~;~~  
select \* from dual;  
↳ This is required

## UPDATE

DML

→ UPDATE table-name

SET column1 = val1, column2 = value2, ...  
WHERE condition;

→ To update a record with data from another table

UPDATE emp

SET job = (select dname from dept  
where rownum=1);

## BETWEEN

- Can be used with SELECT, INSERT, UPDATE or DELETE
- ~~Exe~~ It includes the range

Eg:- SELECT \* from emp WHERE SAL  
BETWEEN 2000 AND 3000;  
(write in asc. order or else there is error)

```
SELECT * FROM emp  
WHERE hiredate BETWEEN  
TO_DATE ('1981/04/02', 'yyyy/mm/dd') AND  
TO_DATE ('2001/04/02', 'yyyy/mm/dd');
```

```
SELECT * from emp WHERE & sal  
NOT BETWEEN 2000 AND 3000;
```

## DELETE

```
DELETE FROM emp  
WHERE condition;
```

- DELETE FROM emp WHERE name = 'SMITH';
- DELETE FROM emp; /All delete

## ALTER (ADD, DROP)

→ add a new column

ALTER TABLE table\_name ADD (column-name datatype);

- adding a single column
- adding multiple column
- adding constraints

ALTER TABLE emp ADD (ADDRESS VARCHAR2(15));

ALTER TABLE emp ADD (address VARCHAR2(15),  
father-name VARCHAR2(15),  
mother-name VARCHAR2(15));

ALTER TABLE emp ADD PRIMARY KEY(emp\_no);

ALTER TABLE emp ADD (gender CHAR(1)  
DEFAULT 'M');

→ Drop a column using ALTER drop

- drop single/multiple/constraints of column

ALTER TABLE emp DROP (column\_name);

ALTER TABLE emp DROP CONSTRAINT  
emp\_id;

## ALTER (RENAME, MODIFY)

- Use ALTER-MODIFY to change the datatype of existing column or the size of datatype  
• modify single/multiple columns

ALTER TABLE emp MODIFY address  
VARCHAR(75);

ALTER TABLE emp MODIFY  
gender VARCHAR(2), commission INT;

- ALTER-RENAME to change the name of column

ALTER TABLE emp RENAME COLUMN  
address TO location;

## DESC

- Describe datatypes of table

DESC emp;

## LIKE

→ used to search specified pattern in the data and retrieve the record when there is a pattern match as required

### GENERAL PATTERNS

#### Pattern

#### Meaning

'r%'      Matches strings which start with 'r'

'%or'      Matches strings which end with 'r'

'r%t'      , , , start with 'r' & end with 't'

'%tri%'      ' contain the substring 'tri' in them at any position .

'\_tri%'      contain the substring 'tri' in them at 2<sup>nd</sup> position .

' -r%'      contain 'r' at second pos

'r--%'      contain 'r' with at least 2 more character

→ SELECT ename from emp WHERE ename LIKE 'M%';

→ SELECT ename, hiredate from emp WHERE hiredate LIKE '% DEC %';

## CREATE

CREATE TABLE employee  
empno number(4,0),  
cname varchar2(10),  
hiredate date);

precision in digits  
scale  
precision in decimal

→ create a table having same structure as another table and copying the data.

CREATE TABLE emp1 AS (SELECT \* FROM emp);

→ CREATE a table having specific column from another table

CREATE TABLE emp2 AS (SELECT empno, ename,  
sal FROM emp);

→ create a table having specific rows from another table.

CREATE TABLE emp3 AS (SELECT \* FROM emp  
WHERE deptno = 10);

→ To copy ONLY structure & no data

CREATE TABLE emp4 AS (SELECT \* FROM emp  
WHERE 1=2);

## AND/OR

- `SELECT column1, column2 FROM table-name  
WHERE condition_1 AND condition_2;`
- `SELECT column1, column2 FROM table-name  
WHERE condition_1 OR condition_2;`
- `SELECT * from EMP WHERE  
(SAL > 500 OR JOB='MANAGER') AND  
(DEPTNO = 10 OR DEPTNO = 30);`

## DISTINCT

`SELECT DISTINCT column1, column2  
FROM table-name;`  
 • on a single column or multiple [your choice]

If you do it for multiple columns, if only one attribute is unique it will be considered

Manager 2400 ✓ Both  
 Manager 3000 ✓

Manager 3000 ✓  
 Manager 3000 X

`SELECT DISTINCT job, DISTINCT sal FROM  
emp;`

You do not need to write distinct multiple times

TIN

BONDS Page No.  
Date / /

SELECT \* FROM emp WHERE ename IN ('BLAKE',  
'SCOTT', 'FORD');

SELECT \* FROM emp WHERE ename sal  $\in$   
IN (6000, 3000, 2000);

DELETE FROM emp WHERE ename  
IN ('James', 'Adams');

\* Interview Question  
Display 2<sup>nd</sup> highest salary

SELECT max(sal) FROM emp  
WHERE sal not IN (SELECT max(sal)  
FROM emp);

# AGGREGATE Functions and GROUP BY

## Aggregate Functions

- SUM
- MAX
- MIN
- COUNT
- AVG

`SELECT MAX(marks) FROM student;`

→ NULL is ignored by all functions

→ Find max marks in every subject

`SELECT Subject, MAX(marks) FROM student GROUP BY Subject.`

→ Group by clause has 3 Rules

① either have all or part of clause in the select like in above example subject is displayed.

② or have a aggregate function in select

③ In grp by where is not used instead having is used.

**In**

Eg:-  
 SELECT name, sum(marks) AS msum  
 FROM student  
 GROUP BY name  
 HAVING sum(MARKS) > 150;

→ To select employees joined in a particular year.  
 according to

SELECT TO-CHAR(hiredate, 'YYYY') "YEAR", COUNT(\*)  
 FROM emp  
 GROUP BY TO-CHAR(hiredate, 'YYYY');

Res:

YEAR	COUNT(*)
1981	10
1987	2
1982	1
1980	1

ORDER BY

SELECT ename, sal FROM emp ORDER BY  
 sal DESC;

→ Multiple orders

SELECT \* from emp ORDER BY deptno, sal;

Result:

Deptno	sal
10	1000
10	2000
10	3000
20	1000
20	2000

[DROP] → completely delete a table.

Drop TABLE emp;

You can FlashBack it  
but to delete permanently  
Drop TABLE emp PURGE;

[RENAME] can also be used without ALTER

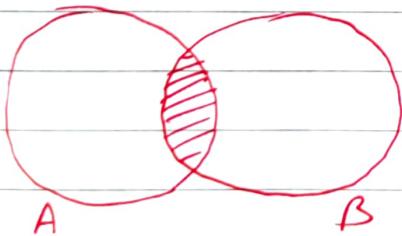
RENAME old-table-name to new-table-name;

# JOINS IN SQL

- Cross Product
- Inner Join
  - ↳ Equi Join
  - ↳ Natural Join
- Outer Join
  - Left Outer
  - Right
  - FULL
- Self Join

## INNER JOIN

```
SELECT Employee.Name, Employee.Dept-ID, Department.
Dept-ID, Department.Dept-name FROM Employee
INNER JOIN Department ON Employee.Dept-ID
= Department.Dept-ID;
```



### Equi Join

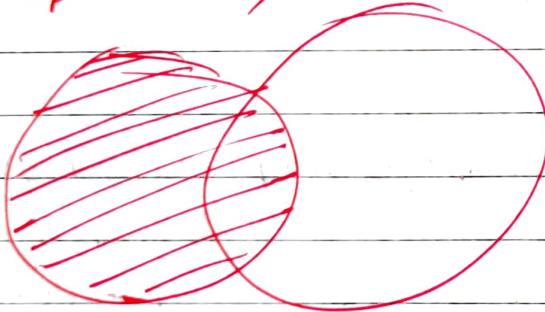
```
SELECT Employee.Name, Employee.Dept-ID, Department.Dept-
ID, Department.Dept-name FROM Employee
INNER JOIN Department USING Dept-id;
(Not allowed in SQL server)
```

Natural Join (when one or more column name is same)

SELECT Employee-Name, Employee.Dept-ID, Department-Dept-Name  
FROM employee NATURAL JOIN Department;

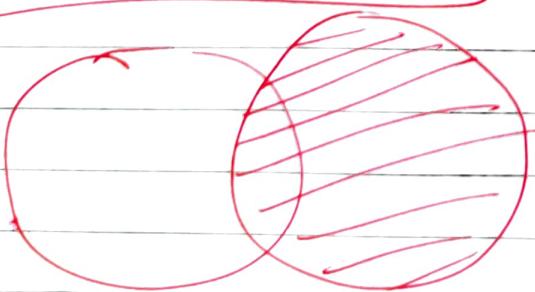
### LEFT Outer JOIN

SELECT Employee-Name, Employee.Dept-ID, Department.Dept-ID, Department.Dept-Name  
FROM employee LEFT OUTER JOIN Department  
ON Employee.Dept-ID = Department.Dept-ID;



Values that do not follow constraint are kept NULL

### RIGHT Outer JOIN



SELECT 'table-name.column-name';  
FROM table-name RIGHT OUTER JOIN  
Department ON Employee.DeptID = Employee.Dept-id;

## FULL OUTER JOIN

(MySQL does not allow this)  
so there we use UNION



SELECT t1.col1, t1.col2, t2.col1, t2.col2  
OUTER JOIN Table-name ON t1.c1 = t2.c1;  
FROM Table-name;

## SELF JOIN