

# SQL

Page No.: 1

Date:

youva

09/04/21

## ① SQL (Structured Query language):

SQL is Structured query language which is used to "interact with Database" is called SQL.

### User's of SQL:-

There are three users of SQL.

- ① Database Developer.
- ② Database tester.
- ③ DBA (Database administrator)

### ① Database Developer:-

Developer will use SQL to "create database" and "Create table's" in Database.

### ② Database tester:-

Test Engineer will use SQL to perform "Back-end validation".

### ③ DBA:- Back-end validation:-

whatever the data which is entered in front-end as successfully stored in back-end (database) or not is called back-end validation.

### ④ DBA:-

DBA will use SQL to maintain successful environment in database.

## Contact

<input type="text"/>
+ New Contact
[Edit Contact]
[Delete Contact]
[Block Contact]
[Favourite Contact]

(Front-end)  
New Contact

Name	Vishnu
Mob	9876543210
Email ID	Vishnu...@gmail.com
Address	longanapet
Nickname	Vishnu
<input type="button" value="Save"/>	<input type="button" value="Cancel"/>

"Back-end validation" → SQL → Query

## Data base (Back-end)

New contact					
Name	Mobile	E-mail ID	Address	Nickname	
Vishnu	9876543210	Vishnu...@gmail.com	longanapet	Vishnu	-ganti

01/7/21

Importance of SQL:

\* Creating Database

\* Creating tables in Database

\* Creating Views

\* Inserting new data into database.

\* Deleting unwanted data from database.

\* updating / modifying old data to new data

\* fetching / retrieving / accessing data from database

- \* Creating stored procedure
- \* Creating index, triggers, cursors

advance SQL (PL/SQL)  
(procedural language extension in SQL)

### History of SQL:

- \* SQL was developed by "Raymond Boyce" in a year 1970.
- \* "Raymond Boyce" is known as father of SQL
- \* "E.F. Codd (Edgar Frank Codd)" is a "Co-developer of SQL."
- \* SQL was developed in "IBM Company" but current owner of SQL is "Oracle company."
- \* In earlier day's SQL was also known as "SEQUEL" (simple English Query executable language). Once "ANSI" (American National Standard Institute) standard was acquired SEQUEL and they renamed SEQUEL into SQL in a year 1979.

### Basic terminology:

#### Data base:-

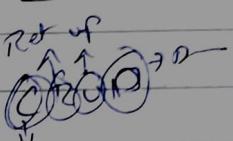
Data base is a centralized place which is used to store some data and manage data is called as database.

(Centralized place means it is a common place where everyone can access or use).

\* Generally database is a collection of information (or) data.

\* In database the data will be stored in a table format.

Each & every database should support the following interaction to user :-

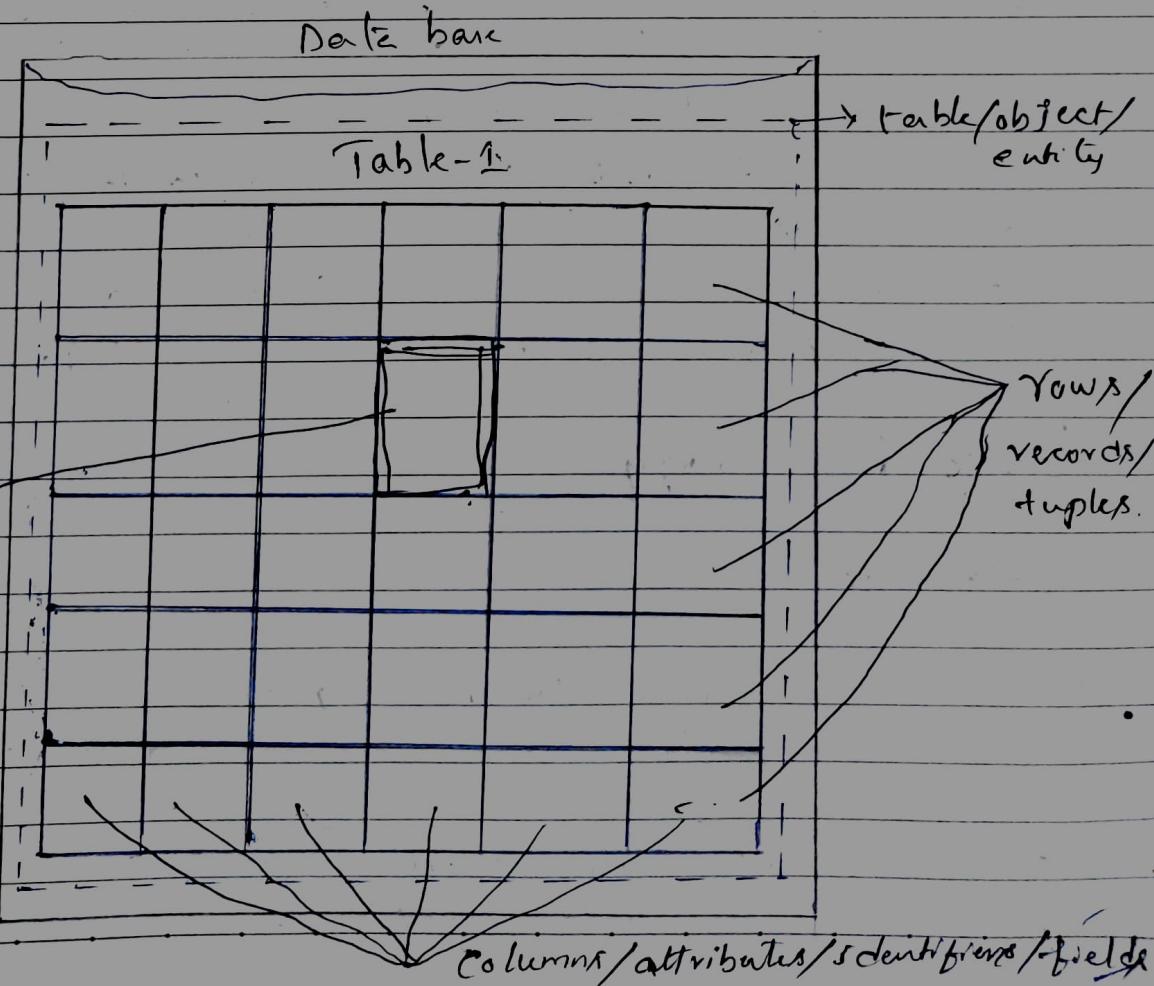


\* Inserting new data into database.

\* Deleting unwanted data from database.

\* updating / modifying old data to new data.

\* fetching / retrieving / accessing data from database.



Note:-

\* Rows is also known as "records" & tuples.

\* Table is also known as "object or entity".

\* Column's are also known as "attributes, identifiers, fields".

\* Cell means / Interaction of rows & columns

(Interaction of records & attributes)

(Interaction of tuples & identifiers)

(Interaction of tuples & fields)

12/7/21

Basic Commands of SQL:-

1) To use SQL Search for SQLplus (click on window button  
 ↓  
 Type sqlplus).

2) we can login 2 users.

a) scott      b) hr

a) scott user:-

User name :- scott ↵

password :- Oracle1234 ↵

Connected to.

b) hr user:-

User name :- hr ↵

password :- Oracle1234 ↵

Connected to.

3) cl-scr user:-

\* This Command is used ~~display~~ clear the screen.  
to

4) Show user:-

This Command is used display logged in as which user.

example:-

SQL > show user.

USER is "SCOTT".

SQL > show user.

USER is "HR".

5) Conn (connect) :-

This Command is used to switch from one user to another user.

example:-

SQL > Conn

Enter user-name :- hr ↵

Enter password :- Oracle1234 ↵

Connected.

6) Select \* from tab; :- (tab is the shortcut of table)

This query is used to displayed number of tables in respective user.

a) table in "Scott" user:-

① BONUS

- ② DEPT
- ③ EMP
- ④ SALGRADE

b) Tables in "hr" user:-

- ① REGIONS
- ② COUNTRIES
- ③ LOCATIONS

Note:- about "scott & hr" tables:-

\* To display the scott table.



Connect with scott



Type [select \* from tab; ] ↵

\* Same as to display the hr table.



Connect with hr



Type [select \* from tab; ] ↵

7) DESC (describe):-

This command is used to describe the table structure (or) describe number of columns in a table.

Syntax:-

DESC table\_name.

in app

SQL > desc emp ↴ (after entering it display's like)

Name	Null ?	Type
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		NUMBER(4)
HIRE DATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)

↓  
In system again it shows

SQL > desc dept ↴ (after entering it display's like)

Name	NULL ?	TYPE
DEPTNO	NOT NULL	NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(15)

↓

In system again it shows

SQL >

Note:-

\* while entering desc emp ↴ & desc dept ↴ it should be connect with "Scott".

8) Set lines 1000 pages 1000

Set lines 100 pages 100

Note: This command is used to set screen resolution.

\* Steps to install oracle 11g RDBMS database:-

① Download database 1 of 2 & database 2 of 2.

Database 2 of 2 :-

① Extract/unzip database 2 of 2.

② Rename database to database 2.

⑤ Click on database 2



Click on Stage



Click on Components



Select all 4 folders (ctrl+a)



Copy all folders (ctrl+c).

Database 1 of 2 :-

③ Extract/unzip database 1 of 2.

④ rename database to database 1.

⑥ Click on database 1



Click on Stage



Click on Components



Paste all folders (ctrl+v).

⑦ Click on database 1



Double click on setup.exe



Click on Yes



↓  
please wait (manually)

↓  
Click on yes

↓  
unselect check box (I wish to receive etc..)

↓  
Click on next

↓  
Click on yes

↓  
Click on next

↓  
Click on next

↓  
Enter administrative and Conform password  
password (oracle1234/Tiger 1234)

Note:- "O & T" are capitals in passwords

↓  
click on next

↓  
click on finish

wait until progress bar become 100% 2 times.

Note:

after completion of installation click on win-down key

↓  
Search Sqlplus.

Installation Done

31/7/21

[cloud]

## SQL statements / SQL language's :-

- \* SQL statements / language's is used to interact with database.
- \* There are 5 sub-languages present in SQL. They are.
  - ① DQL (Data Query language) → Select.
  - ② DDL (Data Definition language) → Create, alter, drop, truncate, rename.
  - ③ DML (Data manipulation language) → Insert, update delete.
  - ④ DCL (Data control language) → grant, revoke
  - ⑤ TCL (Transaction control language) → Commit, rollback, Savepoint.

### DQL (Data Query language) :-

fetching / retrieving / accessing data from the database is known as DQL (Data Query language).

- \* In DQL we have one statement that is "Select" statement.

#### Select :-

fetching/retrieving/accessing data from specific table in database is called "select."

- \* Select statement are classified into three types
- ① projection
- ② selection
- ③ joins.

① P



## ① Projection :-

fetching / retrieving / accessing data from specific table by selecting column's (or) specific / required no. of column's in database is known as projection. Drawback :-

- \* In projection we can restrict column's but can't restrict rows.

### Note :-

Irrespective column's all rows will select.

### Syntax :-

Select \* / [distinct] C.n's / expression alias\_name → ②

from table\_name; → ①

### Note :-

/ → OR

C.n → Column (or) Column name

[ ] → optional

C.n's → column names

\* → meta character

Stmt = clause (same)

### join

### projection

fetching / retrieving / accessing data from specific column in the table (or) all column's in the table in the database is known as projection.

## data base

Table 1

	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$
$r_1$					
$r_2$					
$r_3$					
$r_4$					
$r_5$					

 $C_1, C_2, C_3, C_4, C_5$ 

+

(If we select all column's it can)

 $r_1, r_2, r_3, r_4, r_5$ 

(Select all rows)

 $C_1, C_3, C_5$ 

+

(If we select a specified column

 $r_1, r_2, r_3, r_4, r_5$ 

respectively it select all rows)

 $C_3$ 

+

(If we select a single column  
respectively it select all  
rows) $r_1, r_2, r_3, r_4, r_5$

### characteristics of from clause:-

- \* In from clause we can specify table name as an argument.
- \* In from clause we can specify more than one table name as an argument.
- \* from clause will select whole table for query execution.

### characteristics of select clause:-

- \* In select clause we can specify \* (meta character), distinct, c.n's, expression, alias name as an argument.
- \* select clause will execute "row by row" / "record by record".
- \* Select clause is the responsible to prepare "output" / "output/result" in a table format.

Student		
Sname	branch	per
Dinga	CS	95
Dingi	IS	75
Penga	ME	35
Pengi	CIVIL	.45
Manga	EEE	55

WQD student name, percentage of all students?  $\rightarrow$  display

Select Sname, per  $\rightarrow$  ②

from Student;  $\rightarrow$  ①

O/p :- Sname per

Dinga 95

Dingi 75

Penga 35

rule 1 :- what to display  $\rightarrow$  select clause

rule 2 :- from where to display  $\rightarrow$  from clause

14/7/12

Page No.: 15

Date:

youva

① WQD (write a Query to display) student name, branch of all the students   
 display

Select Sname, branch  
from student;

② WQD branch, percentage of all the students.  
↳ display

Select branch, per  
from student;

③ WQD student name of all the students.  
↳ display

Select Sname  
from student;

④ WQD student details of all the students.  
↳ display

Select Sname, branch, per  
from student;  
(or)

Select \*  
from student;

Note:-

\* (meta character) :- when ever we want to display all columns of the table then we should use \* (meta character)

⑤ WQD employee name, designation, salary of all employees.  
↳ display

Select ename, job, sal  
from emp;

⑥ W<sub>QD</sub> employee name, Date of joining, employee number of all the employees  
Select ename, hiredate, empno  
from emp;

⑦ W<sub>QD</sub> employee number, manager number, department number of all the employees  
Select empno, mgr, deptno  
from emp;

⑧ W<sub>QD</sub> employee name; commission, salary of all the employees  
Select ename, comm, sal  
from emp;

⑨ W<sub>QD</sub> employee details of all the employees  
Select \*  
from emp;

⑩ W<sub>QD</sub> employee name, manager number, designation, date of joining of all the employees  
Select ename, mgr, job, hire date  
from emp;

⑪ W<sub>QD</sub> department details of all the employees department  
Select \*  
from dept;

⑫ W<sub>QD</sub> department name, location of all the department  
Select Dname, loc  
from dept;

- (13) WQD department number, location of all the department.  
 display Select Dept-no, loc  
 from dept;
- (14) WQD salary, incentives, manager number of all employees.  
 display Select Sal, Comm, mgr  
 from emp;
- (15) WQD department name of all the departments  
 display Select Dname  
 from dept;
- (16) WQD region name of all the regions.  
 display Select region-name  
 from regions;
- (17) WQD region-details.  
 display Select \*  
 from regions;
- (18) WQD Country-details.  
 display Select \*  
 from Countries;
- (19) WQD country-name, country-ID of all the countries.  
 display Select country-name, country-id  
 from Countries;
- (20) WQD country-ID, region-ID of all the countries  
 display Select country-ID, region-ID  
 from Countries;

(21) WQD location\_details.

Select \* → display  
from locations;

(22) WQD location\_id, street\_address, city of all locations.  
display Select location\_id, street\_address, city  
from locations;

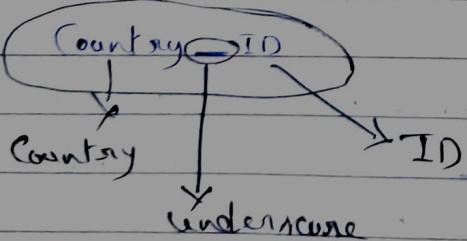
(23) WQD location\_id, postal\_code, country\_id of all the locations.  
display Select location\_id, postal\_code, country\_id  
from locations;

(24) WQD state\_province, city of all the locations.  
display Select state\_province, city  
from locations;

(25) WQD Country\_name, region\_id of all Countries.  
display Select Country\_name, region\_id  
from Countries;

Note:-

→ ; → end of the query



Alias name :-

- \* Alias\_name is alternative name (or) temporary name given to column present in result/output is called as alias\_name.
- \* whenever we want to change column present in result/output then we should go alias\_name.
- \* whenever we are writing alias\_name it is a temporary changes not permanent changes.

Syntax:-

```
Select column_name [as] alias_name
      from table_name;
```

Example:-

SQL > Select Sal  
2. from emp;

Output	Q/P	Sal
		800
		1600
		:

(or)

SQL > Select sal as salary  
2. from emp;

Q/P	SALARY
	800
	1600
	:

(or)

SQL > select sal salary  
2. from emp;

O/P

SALARY
800
1600
(or)

Note:-

we can write two Query's at a single time

SQL > Select sal as salary, job as designation  
2. from emp;

O/P

SALARY	DESIGNATION
800	CLERK
1600	SALESMAN
(or)	

SQL > Select sal salary, job designation  
2. from emp;

O/P

SALARY	DESIGNATION
800	CLERK
1600	SALESMAN

Note:-

[as] is an optional, we should "as" to convert the sal to salary in output column name or else sal. salary.

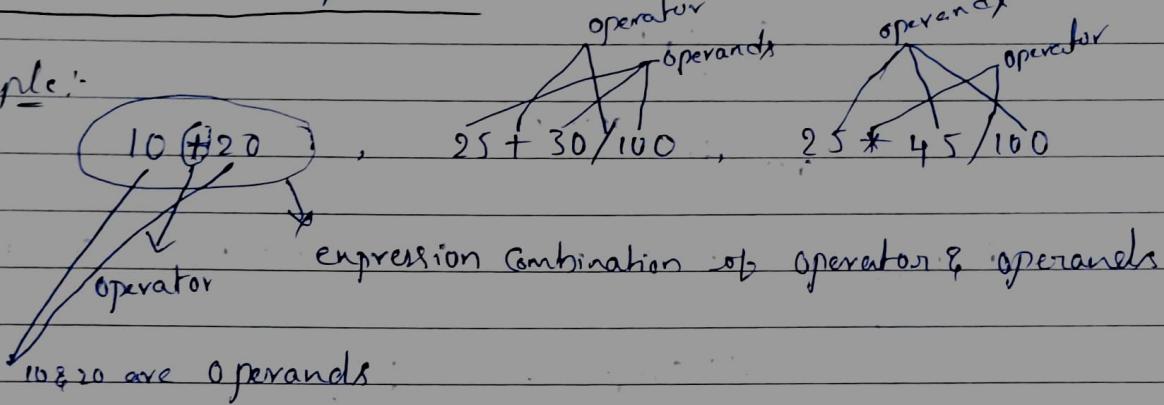
By using

15/7/21

Expression :-

An expression is a statement if will generate result (or) output is called an expression.

\* Generally an expression is a combination of "operator and operands".

example:-Operator's:-

(1) Arithmetic operator ----> +, -, \*, /

(2) Relational operator -----> <, >, <=, >=, =, !=

(3) Logical operator -----> AND, OR, NOT

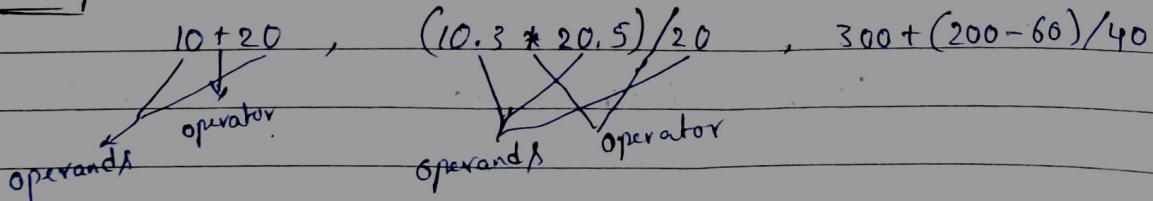
Operands:-

\* An operand is a direct value.

\* operands are classified into two types.

(1) Direct value / immediate value

(2) variables / identifiers

(1) Direct value / immediate value :-example:-

## ② Variables / identifiers :-

## Variablen

example :-

```
int a=10, b=20, c
```

$$c = a + b$$

Identifiers :- In SQL we will use identifiers/c.n as an operand  
example:-

Salt comm , Salt + 100

operator      operands

operator      operands

## Literals :-

- \* literals is a direct value / immediate value given to any of SQL statements is called as literals.
  - \* literals are classified into three types
    - (1) Number literal
    - (2) String/character literal
    - (3) Date literal.

① number literal:-

- \* It is a direct value/immediate value which is of type number and given to any of sql statement is called as number literals.

### example:

420, 840, 9000, 00, 10.12

## ② String / character literal :-

- \* It is a direct value / immediate value which is of type string / character and given to "any SQL statements" is called as string / character literals.
- \* (Alphabets, numbers, special character) should be enclosed with single quote (' ') is known as string / character.

example:-

'Dinga' 'Dingi@123' 'mottu@123'  
      '@#%123'.

## ③ Date literals :-

- \* It is a direct value / immediate value which is of type date format and given to "any SQL statement" is known as date literals.
- \* Always date literal should be enclosed within single quote (' ').

Oracle date format:-

DD - MON - YY      (or)      DD - MON - YYYY

example:-

15 - AUG - 47      (or)      15 - AUG - 1947

## ④ WQD

(1) WAP employee name, employee number, salary, new salary with 1000rs increment (+) of all employees.

Query: Select ename, empno, sal,  $sal + 1000$  as new-sal  
from emp;

(cor)

$\downarrow$  expression  $\downarrow$  operator  $\downarrow$  operator  $\downarrow$   
 $\downarrow$  optional alias-name

Select ename, empno, sal, sal+1000 new-sal  
from emp;

Note:

Here (as) is optional

(2) WAP employee name, designation, salary, new salary with 1000rs decrement (-) of all employees.

Query

Select ename, job, sal,  $sal - 1000$  as new-sal  
from emp;

$\downarrow$  expr (cor) optional alias-name  
 $\downarrow$  new-sal

(3) WAP employee name, department number, date of joining, salary, annual salary of all the employees.

Query

Select ename, deptno, hiredate, sal, sal \* 12 as Ann-sal  
from emp;

$\downarrow$  (cor) [ ]  
sal \* 12 Ann-Sal

Note:

Annual salary = one year salary  
it can be written as sal \* 12

Sal \* 12

$\downarrow$  operator  $\searrow$  operands.

(4) WQD employee name, manager number, salary, commission, new salary with 30% increment. from employee table.

display  $\text{Sal} + (\text{Sal} * \frac{30}{100})$

$$\text{Sal} + (\text{Sal} * 0.3) \rightarrow ①$$

(or)

$$\text{Sal} (1 + (1 * 0.3)) \quad 13\%$$

$$\text{Sal} (1 + 0.3)$$

$$\text{Sal} * 1.3 \rightarrow ②$$

Query

Select ename, mgr, sal, comm,  $\text{Sal} + (\text{Sal} * 0.3)$  as new\_sal

(or)

$$\text{Sal} * 1.3 \text{ as new\_sal}$$

from emp;

(or)

(5) WQD employee name, employee number, salary, new salary with 30% decrement of all the employees.

display

$$\text{Sal} - (\text{Sal} * 0.3) \rightarrow ①$$

$$\text{Sal} (1 - (1 * 0.3))$$

$$\text{Sal} (1 - 0.3)$$

$$\text{Sal} (0.7)$$

$$\text{Sal} * 0.7 \rightarrow ②$$

Query

Select ename, empno, sal,  $\text{Sal} - (\text{Sal} * 0.3)$  as new\_sal

(or)

$$\text{Sal} - (\text{Sal} * 0.3) \text{ new\_sal}$$

from emp;

Select ename, empno, sal,  $\text{Sal} - (\text{Sal} * 0.3)$ ,  $\text{Sal} * 0.7$  as new\_sal

(or)

$$\text{Sal} * 0.7 \text{ new\_sal}$$

from emp;

~~16/8/21~~ ⑥ WAPD employee name, designation, commission, new commission with 20% increment.  $\rightarrow$  display

## Query

Select ename, job, comm, comm\*1.2 as new\_sal  
(or)  
from emp;  
: : Comm\*1.2 new\_sal

$$\text{Comm} + (\text{Comm} * 0.2) \rightarrow ①$$

$$\text{Comm} \left( 1 + (1 * 0.2) \right)$$

Comm (1+0.2)

Comm \* 1.2 → ②

F) WQD employee name, employee number, manager number, commission, new commission with 10% decrement.

display Comm  $\rightarrow$  (Comm \* 0..1)  $\longrightarrow$  ①

$$\text{Comm} \quad (1 - (1 * 0.1))$$

Comm ( $1 - 0.1$ )

$\text{Comm}(0.9)$

Comm \* 0.9 --> ②

Over

Select ename, empno, mgr, comm, comm\*0.9 new-comm  
(or)

from emp;

(Comm \* 0.9 as new-Say)

(8) WAD employee name, employee number, commission, new commission with 500\$ increment.

Query

Select ename, empno, comm, Comm+500 new-~~sal~~<sup>comm</sup>  
from emp;

- ⑦ WQD employee name, department number, commission, new commission with 200rs decrement.

*Query*

→ display

Select ename, deptno, comm, Comm - 200 as new-comm  
from emp;

- ⑧ WQD employee name, employee number, salary, half-term salary.

*Query*

→ display

→ optional

Select ename, empno, sal, sal \* 6 (as) new-sal (Half-term cor)

Sal \* 6      Half-term  
new-sal

from emp;

- ⑨ WQD employee name, date of joining; salary, annual salary, new annual salary with 1000rs increment.

*Query*

→ display

Select ename, hiredate, sal, (sal \* 12) + 1000 new-sal  
from emp;

- ⑩ WQD employee name, department number, salary, new annual salary with 2000rs decrement.

*Query*

→ display

Select ename, deptno, sal, (Sal \* 12) - 2000 as new-sal  
from emp;

- ⑪ WQD employee name, commission, annual commission of all the employees.

*Query*

→ display

Select ename, comm, Comm \* 12 as new-comm  
from emp;

(14) W/QD employee name, manager number, salary, by giving 500Rs increment salary. find out annual salary.

Query  $\rightarrow$  display

Select ename, mgr, Sal, (Sal+500)\*12 as new\_ann\_sal  
from emp;

(15) W/QD employee name, employee number, Commission, by giving 500Rs increment Commission. find out annual Commission, by giving 200Rs decrement then find out half-term Commission.  $\rightarrow$  display

Query Select ename, empno, comm, (Comm+500)\*12 as ann\_com, (Comm-200)\*6 as half\_com

Select ename, empno, comm, Comm+500 as new\_com, (Comm\*12)-200 as ann\_sal, Comm\*6 half\_com

Select ename, empno, comm, Comm+500 as new\_com, (Comm\*12)-200 as ann\_com, Comm\*6 as half\_com  
from emp;

(16) W/QD employee name, date of joining, annual salary, new annual salary with 30% increment.

Query

Select ename, hiredate, Sal\*12 as new\_ann\_sal,  
 $(Sal*12) + ((Sal*12)*0.3)$  as new\_annual  
from emp;

$$(Sal*12) + (Sal*12*0.3) \rightarrow ①$$

$$(Sal*12) + ((Sal*12)*0.3) \rightarrow ① \quad Sal [ (1*12) + (1*0.3) ]$$

$$Sal*12 [ 1 + (1*0.3) ] \quad Sal [ 12 + 0.3 ]$$

$$Sal*12 [ 1 + 0.3 ]$$

$$Sal * 12.3 \rightarrow ②$$

$$(Sal*12) * 1.3 \rightarrow ②$$

$$(Sal*12) * 1.3$$

(17) W00 employee name, employee number, salary, by giving 20% increment salary, then find out half-term salary, by giving 10% decrement salary find out quarterly salary.

Query

Select ename, empno, sal,  $sal + (sal * 0.2) * 6$  as half-term-sal,

$(sal * 6) - (sal * 0.1)$  as half-sal,  $sal * 3$  as qud-sal,  
from emp;  $(sal - (sal * 0.1)) * 3$  as qtr-sal.

$$Sal + (Sal * 0.2) \rightarrow ①$$

$$Sal(1 + (1 * 0.2))$$

$$Sal(1 + 0.2)$$

$$Sal * 1.2 \rightarrow ②$$

$$(Sal * 6) - (Sal * 0.1) \rightarrow ①$$

$$Sal[(1 * 6) - (1 * 0.1)]$$

$$Sal[6 - 0.1]$$

$$Sal * 5.9 \rightarrow ②$$

(or)

Select ename, empno, sal,  $Sal * 1.2$  as near-sal,  $Sal * 5.9$  as  
 $Sal * 5.9$  as half-sal,  $Sal * 3$  as qud-sal.

from emp;

Select ename, empno, sal,  $Sal + (Sal * 0.2) * 6$  as half-term-sal  
 $Sal - (Sal * 0.1) * 3$  as qtr-sal

from emp;

## Order by clause:-

\* Order by clause is used to sort the output/result in ascending order/ descending order

Note:-

ascending order - low to high (A,B,C...Z)

descending order - high to low. (Z,Y,X...A)

\* Always order by clause should be last statement in query.

\* By default order by clause will sort the output in ascending order, [writing Asc keyword is optional]

\* If in case we want to sort the output in descending order, then writing DESC is mandatory.

Shortcut :-

(optional) → Ascending order - ASC

(mandatory) → Descending order - DESC

Syntax:-

Select \* / [distinct] c.n's / emp alias-name  
---> (2)

from table-name ---> (1)

order by c.n / emp / alias-name [ASC] / DESC;  
---> (3)

① INQD employee details and display the output in ascending order based on salary.

Query

Select \*

from emp

order by Sal asc;

QF

(or)

Select \*  
from emp

order by sal;

display

- ② WLD employee name, employee number, salary and display the output in descending order based on salary.

Query

Select ename, empno, sal

from emp

order by sal desc;

- ③ WLD employee name, salary, annual sal, and display the output in descending order based on annual salary.

Query Select ename, sal, sal \* 12 as ann-sal

from emp

order by ann-sal desc;

(or)

Select ename, Sal, Sal \* 12 as ann-sal

from emp

order by ann-sal desc;

Note:-

whenever we writing alias name in order by clause then alias name must and should be present in select clause. otherwise it throw's error.

## Distinct clause:-

- \* Distinct clause is used to eliminate the duplicate values present in result (or) output.
- \* Distinct clause will be execute after select clause.

Syntax:

→ ①      → ③  
 Select distinct c.n / expr  
 → ②      from table-name ; → ④

Syntax:-

Select count(distinct <sup>c.n / expr</sup>)  
 from table-name ;

Example:-

emp	Sal	from emp
	3000	→ The overall table will select
	4000	by writing a query as
	3000	→ from table-name ;
	5000	→ So → select dis.
	4000	O/P:-      Sal

→ So → select dis.

O/P:-      Sal

3000

4000

5000

Note:-

(no duplicate)

WAP unique salary of all the employees.

① WAP unique designation of all the employees.

query

Select job  
 Select distinct job  
 from emp;

(2) W/OD unique manager number of all the employees.

Query

Select distinct mgr  
from emp;

(3) W/OD unique annual salary of all the employees.

Query

Select distinct sal \* 12 <sup>optional</sup> (as) Ann\_sal  
(or)

sal \* 12 Ann\_sal

from emp;

(or)

Select distinct sal \* 12  
from emp;

(4) W/OD unique department number of all the employees.

Query

Select distinct deptno  
from emp;

X/X

(5) W/OD unique salary, unique designation of all the employees.

Query

Select distinct sal, distinct job  
from emp;

→ Select distinct sal, job  
from emp; ✓

O/P:- missing expression

\* According to above query we cannot write multiple distinct clause but we can write multiple column's in single distinct clause.

Rule:-

Any clause/statement can be written only one time in a query.

## multiple C.N's in distinct clause:-

whenever we are specifying multiple C.N's in single distinct clause, distinct clause will not eliminate duplicate values in individual C.N.

Instead of that distinct clause will combine multiple C.N values into a single value then it will eliminate duplicate values.

	emp			②	③
ename	sal	job		Select distinct sal, job	
Aleash	3000	TE		from emp;	→ ①
Amith	4000	DEV	OF		
Miller	5000	HR		sal	job
Smith	3000	TE		3000	TE
Blalce	4000	DL		4000	DEV
Allen	3000	TL		5000	HR
				4000	DL
				3000	TL

Ans

Ques) find unique salary, unique designation of all the employees.

query

Select distinct sal, job

from emp;

Ques) Ques) (a) find unique department number, unique manager number of all the employees.

Ques

Select distinct deptno, mgr  
from emp;

④ WAD, unique annual sal, unique designation, unique department number of all the employees.

query

Select distinct sal\*12, job, deptno  
from emp;

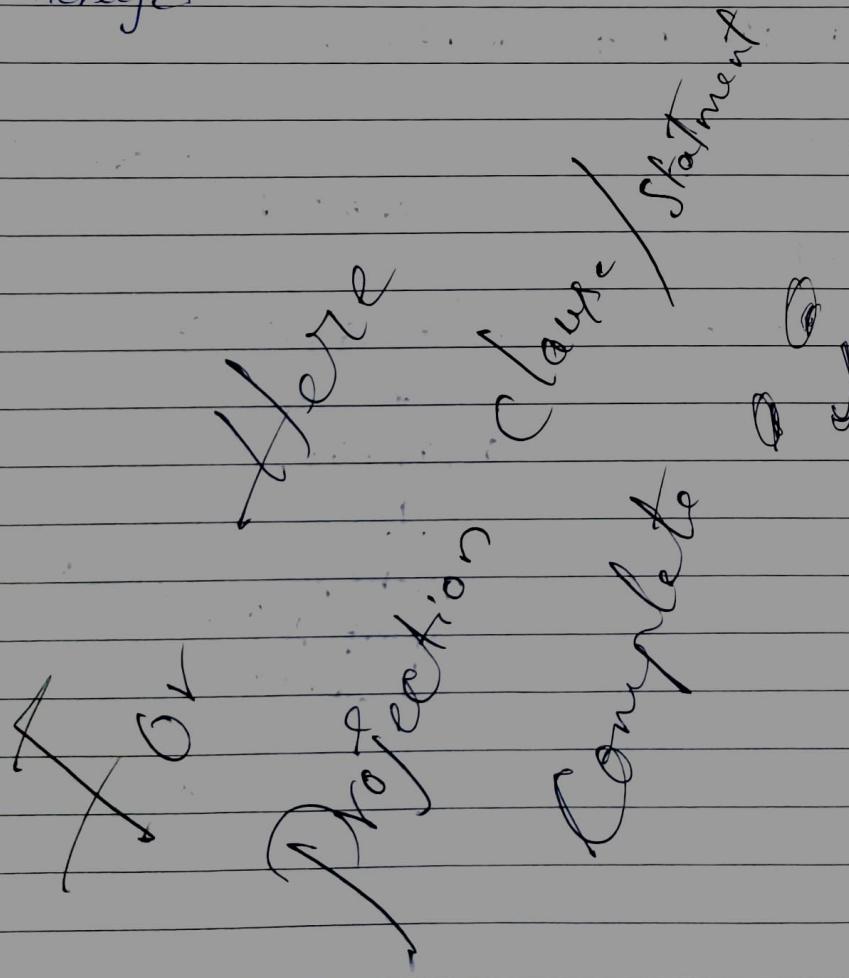
(or)

Select distinct sal\*12 as Ann-sal, job, deptno  
from emp;

Note:-

We cannot write any column name before distinct clause (or) statement.

If you write like then the SQL throws an error message.



## ② Selection:-

- \* fetching/retrieving/accessing data from specific table by selecting all col's (or) specific/required no. of column's and also specific/required no. of row's in database is called as Selection.
- \* In selection we can restrict both row's & column's.

## Syntax of Selection:-

Select \* / [distinct] col's/expr alias\_name ----> ③  
 from table-name -----> ①  
 where < cond's > ; -----> ②

### Note:-

Cond's is stands for Conditions.

data base

table 1				
	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>
r <sub>1</sub>				
r <sub>2</sub>				
r <sub>3</sub>				
r <sub>4</sub>				
r <sub>5</sub>				

↑  
 Selecting all columns  
 C<sub>1</sub>, C<sub>2</sub>, C<sub>3</sub>, C<sub>4</sub>, C<sub>5</sub>  
 +  
 r<sub>1</sub>, r<sub>3</sub>, r<sub>5</sub>  
 ↓  
 Selecting  
 specifying/required no. of  
 column's.

Note:-

According to above syntax there are three clauses are present that is select, from, where.

- First      from clause will execute
- Second    where clause will execute.
- Third     select clause will execute.

Characteristics of where clause:-

- \* where clause is used to filter the records.
- \* where clause will select the record and reject the record through conditions.
- \* In where clause we will specify condition.

Syntax of Condition:-

where LHS-value relational-operator RHS-value;

$\downarrow$                            $\downarrow$                            $\downarrow$   
 Con/expr       $<$ ,  $>$ ,  $<=$ ,  $>=$ ,  $=$ ,  $\neq$       Con/expr/literals.

- \* where clause will execute "row by row / record by record"
- \* If specific condition is TRUE where clause will retain the record, else if the specific condition is FALSE where clause will reject the record.
- \* If specific condition is TRUE or FALSE where clause will not stop the execution, it will continue the execution for all the records present on table.
- \* where clause will not execute multiple records at one shot but where clause will execute multiple conditions for a single record at one shot.

19/1/19

Ex:-

Student			$\text{Per} > 60$
Sname	branch	per	
Dinga	CS	95	$95 > 60 \dots T$
Dangi	IS	55	$55 > 60 \dots F$
Penga	ME	65	$65 > 60 \dots T$
Dangi	EEE	70	$70 > 60 \dots T$
Manga	Civil	45	$45 > 60 \dots F$
Mangi	EC	35	$35 > 60 \dots F$

Woo student details only the percentage should be more than 60.

↳ condition.

query

Select \* ---> ③

from student\_table ---> ①

where  $\text{Per} > 60$ ; ---> ②

Condition

LHS\_value

on

relational\_operator

RHS\_value

(number literal)

output display's as:-

Sname	branch	per
Dinga	CS	95
Penga	ME	65
Dangi	EEE	70

$y_1$  :- what to display ---> select clause

$y_2$  :- where to display ---> from clause

$y_3$  :- is there any condition ---> where clause

(1) WLD student name, branch only the percentage should be less than 70.  $\rightarrow$  display  
 $\downarrow$  condition

Select Sname, branch  
from student ~~table~~  
where per < 70 ;

(2) WLD student name, percentage only the Percentage should be exactly 60.  $\rightarrow$  display  
 $\downarrow$  condition

Select Sname, per  
from student ~~table~~  
where per = 60 ;

(3) WLD employee details only the employee salary should be more than 2000/-  $\rightarrow$  display  
 $\downarrow$  condition

Select \*  $\overset{\text{emp}}{\text{from student table}}$   
where sal > 2000 ;

(4) WLD employee name, designation, salary those are getting salary less than 3000/-  $\rightarrow$  display  
 $\downarrow$  condition

Select ename, job, sal  
from emp ~~table~~  
where sal < 3000 ;

(5) WLD employee name, employee number, salary, annual salary only the salary should be exactly 1250  $\rightarrow$  display  
Select ename, empno, sal, sal \* 12  
from emp  
where sal = 1250 ;

⑥ WQD employee details only the manager number should be more than 7,600 and display the output in descending order based on salary.

Select \*

from emp

where mgr > 7,600

order by sal DESC;

⑦ WQD employee name, employee number, designation, date of joining only the employee number should be 7839.

Select ename, empno, job, hire date  
from emp  
where empno = 7839;

⑧ WQD employee details only the employee should be working as salesman.

Condition

Select \*

from emp

where job = 'salesman'.

O/P:-

No rows selected.

Rules:-

\* SQL language (Table name, Cn, Stmt/ clause, operator) is not a Case sensitive only the values present in a table is a Case sensitive.

\* whenever we are writing a query then we should write query by looking the table structure not by looking the values present in the table.

Not Case Sensitive:- It means we write statement in upper case or lower case not mandatory.

~~20/7/21~~

## Case manipulation function:-

\* There are three types of Case manipulation function

- ① `upper()`
- ② `lower()`
- ③ `initcap()`

### ① `upper()` :-

\* This function will Convert the given string into upper case.

\* We can pass ('string'/c.n) as an argument to upper function.

#### Syntax:-

`upper ('string'/c.n)`

#### Example:- from SQL

SQL > `select upper ('akash')`

2. `from dual;`

UPPER

AKASH

SQL > `select upper ('AKASH')`

2. `from dual;`

UPPER

AKASH

SQL > `select upper ('Akash')`

2. `from dual;`

UPPER

AKASH

SQL > Select upper(ename)  
 2. from emp;

UPPER(ENAME)

ALLEN

WARD

JONES

;

;

## (2) Lower :-

- \* This function will convert the given string in lower case.
- \* we can pass 'string/character/c.n' as an argument to lower function

Syntax:-

lower ('string' / c.n)

SQL > Select lower ('AKASH')  
 2. from dual;

LOWER

akash

SQL > Select lower ('akash')  
 2 from dual;

LOWER

akash

SQL > Select lower ('Akash')  
 2. from dual;

LOWER

akash

SQL > Select lower (ename)  
 2 from emp;

LOWER(ENAME)

Smith

allen

### (3) initCap :-

- \* The function will Convert the given string/character into initCap case (initial letter case)
- \* We can pass 'string/character'/c.n as an argument to initCap function.

Syntax:-

initCap ('string'/c.n)

Example

SQL > Select initCap ('akash')  
2 from dual;

INITC

Akash

SQL > Select initCap ('AKASH')  
2 from dual;

INITC

Akash

SQL > Select initCap ('Akash')  
2 from dual;

INITC

AKash

SQL > Select initCap (ename)  
2 from emp;

INITC

Smith

Allen

Ward

Note:- for above three Case manipulation functions.

we can write a single c.n when we are writing in select - clause it won't accept \* or two c.n names. likewise we can also write single ('string') when we are writing in Select clause.

Note:- Any 'string' we can write in select clause. It means any name shorte who are not present in table 'n'- Vishnu

own

Note:-

Case manipulation function is only used for to specify about string/character literal.

Page No.: 214p

youva

Date:

Applications of Case manipulation function / when to go case manipulation function :-

\* when ever we want to write "string/character literal" in where clause then it is better to go case manipulation function.

Note:-

what type of Case manipulation function writing in the LHS side same type of Value <sup>or</sup> should be present in RHS side.

(i) When employee details only the employee should be working as salesman to display.

query      Select \* <sup>for</sup> condition

from emp

where upper(job) = 'SALESMAN';

(or)

Select \*

from emp

where lower(job) = 'salesman';

(or)

also Select \*

from emp

where initcap(job) = 'Salesman';

Note:-

Either we write upper(job) = 'SALESMAN', lower(job) = 'salesman', initcap(job) = 'Salesman'. But the output will display's the same from the query. because.

because SQL lang/stat are not Case sensitive but SQL values in output are Case sensitive.

display

② WQD employee name, designation, department number only the employees should be working as clerk.

\* condition

Select ename, job, deptno

from emp

where lower(job) = 'clerk';

③ WQD employee details only the employee should be working as analyst and display the output in ascending order based on salary.

analy

Select \*

from emp

where initCap(job) = 'Analyst'

(or)

where upper(job) = 'ANALYST'

(or)

where lower(job) = 'analyst'.

order by Sal;

(or)

order by Sal Asc;

④ WQD employee name, employee number, date of joining, if an employee smith.

Select ename, empno, hiredate

from emp

where lower(ename) = 'smith';

⑤ WQD employee details of an employee allen.

display

condition

Select \*

from emp

where lower(ename) = 'allen';

- ⑥ W<sup>O</sup>D employee details only the employee should be hired after a year 81.
- display  
condition

Select \*

from emp

where hiredate >= '01-JAN-82';

(or)

Select \*

from emp

where hiredate > '31-DEC-81';

Note 1:-

Always month should be write in upper case.

Note 2:-

after >= 1st day of month

before <= last day of month.

- ⑦ W<sup>O</sup>D employee name, employee number, date of joining only the employees should be hired after February 81.
- display  
condition

Select ename, empno, hiredate.

from emp

where hiredate >= '01-MAR-81';

(or)

where hiredate > '28-FEB-81';

- ⑧ W<sup>O</sup>D employee details only the employees should be hired before a year 87

Select \*

from emp

where hiredate <= '31-DEC-87';

where hiredate < '01-JAN-87';

(9) W<sub>Q</sub>D employee name, department number, annual salary, date of joining only the employee's should be hired before November 85.

Select ename, deptno, Sal\*12, hiredate  
from emp

where hiredate <='31-~~OCT~~<sup>OCT</sup>-85';

(or)

where hiredate <'01-~~DEC~~<sup>Nov</sup>-85';

(10) W<sub>Q</sub>D employee name, salary, annual salary, only the annual salary should be more than 10,000

Select ename, sal, sal\*12 as ann\_sal  
from emp

where (sal\*12) > 10000 ;

Note:-

\* we cannot write alias-name in where clause because where-clause will execute before the select clause.

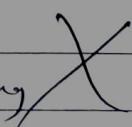
\* we should write alias-name in only order by clause because it will execute after the select clause (or). it will execute after the result/output..

Note:-

Alias-name should be ~~not~~ write in only order by clause because it is the "last statement in a query."

where ann\_sal > 10000 ;

is wrong



## Assignment 3

Page No.: 48

Date:

youva

① WQD (write a query to display) name of the employee.

Select ename

from emp;

② WQD ename and salary of all employee.

Select ename, sal  
from emp;

③ WQD department name and its locations for all the employee department.

Select dname, loc  
from dept;

④ WQD name, salary, commission and date of joining of every employee.

Select ename, sal, Comm, hire date  
from emp;

⑤ WQD employee name, his designation and his manager's employee number.

Select ename, job, mgr  
from emp;

⑥ WQD employee name and his department number.

Select ename, deptno  
from emp;

⑦ WQD all the details of each & every employee of the Company.

Select \*  
from emp;

⑧ List all the details of department present in a company.

Select dept \*  
from dept;

⑨ List employee name, his employee number, his salary and also his annual salary for every employee.

Select ename, empno, sal,  $sal * 12$  as ann-sal  
from emp;

⑩ List employee name, his employee number and salary with the hike of 30% to every employee.

Select ename, empno, sal,  $sal + (sal * 0.3)$  as new-sal.  
from emp;

*we can write*

$$Sal + (Sal * 0.3) \rightarrow ①$$

$$Sal (1 + 0.3)$$

$$Sal (1 + 0.3)$$

$$Sal * 1.3 \rightarrow ②$$

(or)

Select ename, empno, sal,  ~~$sal * 1.3$~~  as new-sal  
from emp;

⑪ List name of the employee with annual salary with a monthly bonus of 200 and provide suitable alias-name.

Select ename,  $(Sal * 12) + 200$  as Mon-Annlal  
from emp;

(12) W<sub>Q</sub>D name, salary along with annual salary of employee with yearly bonus of 1000.

Select ename, sal, (sal\*12)+1000 as ann\_sal  
from emp;

(13) W<sub>Q</sub>D to select all the salary of employee if the salary is greater than 1500.

Select sal

from emp

where sal > 1500

(14) W<sub>Q</sub>D employee name, his hire date, his salary and annual salary only if his annual salary greater than 10,000.

Select ename, hiredate, sal, sal\*12 as Ann\_sal

from emp

where sal\*12 > 10000

(15) W<sub>Q</sub>D all the details of employee if the department number is 20.

Select \*

from emp

where deptno = 20

(16) W<sub>Q</sub>D all the details of the employees if designation is manager.

Select \*

from emp

where job = 'MANAGER';

(or)

where lower(job) = 'manager';

(or)

where upper(job) = 'MANAGER';

(or) where initcap(job) = 'Manager';

(17) WQD all the details of employee only if they were hired after the year 1995.

Select \*

from emp

where hiredate >= '01-JAN-1996' ;

(18) WQD to all the details of employee only if his designation is clerk.

Select \*

from emp

where job = 'CLERK' ;

(19) WQD department name, department number for all the department which is located in BOSTON.

Select dname, deptno

from dept

where loc = 'BOSTON' ;

~~operator's :-~~

operator's :-

\* when ever we want to write multiple condition's in where clause

(or) where statement then it is better to go operator's.

\* when ever we writing multiple conditions in where clause, where clause/stmt will execute multiple cond's for single record at one shot.

Operator's are classified into 5 types :-

(1) Arithmetic operator ----> +, -, \*, /

(2) Concatenation Operator ----> ||

(3) Relational operator ----> <, >, != (or) <>

(4) Special operator ----> IN, BETWEEN, IS, LIKE

(5) Logical operator ----> AND, OR, NOT

own note:  
\* Arithmetic & Concatenation operators is specified in "Select clause"  
\* relational, special & logical operators is specified in "where clause"

Page No. 52  
Date

Youth

## (2) Concatenation operator (II) :-

\* This operator is used to combine/merge/concatenate multiple string into a single string then we should go concatenation operator.

Syntax:-

Example: 'string1'/c.n || 'string2'/c.n || 'string3'/c.n || 'string4'/c.n

SQL > select 'don' || 'key'  
2. from dual;

O/P  
'DON'  
-----  
donkey

SQL > select 'don' || 'key' || 'Abhishek'  
2. from dual;

O/P  
'DON'||'KEY'|| Abhishek  
-----  
donkey Abhishek

SQL > select 'don' || 'key' || 'Abhishek'  
2. from dual;

O/P  
'DON'||'KEY'|| Abhishek  
-----  
donkey abhishek;

Note:-

If you give the space after pipeline and give the string in single quote. then it displays in output with some space word to word

ex:- Select 'don' || 'key' || 'Abhishek'  
from dual;

O/P  
'DON'||'KEY'||  
donkey abhishek

If in case we cannot give a space after starting a single quote and give a space after pipe line and start with string/character in single quote. then it displays in output without space.

Ex-  
1

SQL > Select 'Don''ll'key''ll'Abhishek'  
? from dual;

O/P 'DON'T HEAR ME KEY!!!

donkeyabhishek ;

1 imp point:-

point:- If we give space the output displays as 'don't like'll' donicey abhishek

If we not give space the output displays as Don'll'key'll  
donkeyabhishek;

SQl > select 'Don''ll' key'11 ename  
2 from emp;

✓P

'DON'T'LL KEY'LL EN

donkey SMITH

donkey ALLEN

donkey WARD

⑦ WAD Hello  $x$ , your salary is  $y$  and you working as  $z$

$\downarrow$                      $\downarrow$                      $\downarrow$   
ename                    sal                    job

## Query

```
select 'Hello '||ename||' your salary is '||sal||' and you working  
as'||job  
from emp;
```

Q/P 'HELLO'||ENAME||'YOUR SALARY IS'||SAL|| AND YOU'RE WORKING AS'||JOB

hello SMITH your salary is 800 and you're working as CLERK

hello ALLEN your salary is 1600 and you're working as SALESMAN

② WQD Department (x) is located in (y) write the output

↓                      ↓  
dept dname            loc

Query

select 'Department'||dname||'located in'||loc  
from emp;

Q/P

'DEPARTMENT'||DNAME||'LOCATED IN'||LOC

department ACCOUNTING located in NEW YORK

department RESEARCH located in DALLAS

③ WQD hello x your salary is y and you working as z  
and the salary should be exactly 800.

Query

Select 'Hello'||ename||' your salary is'||sal|| and your  
working as'||job.

from emp

where sal=800;

## ⑤ Logical operators :-

There are three types of logical operators.

① AND

② OR

③ NOT

### ① AND :-

when the given Conditions are mandatory to execute then we should go AND operator.

Symbol :-



Truth Table

Cond1	Cond2	O/P
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE

### ⑥ Case 1 :-

- \* If Cond1 is TRUE the O/P of AND operator is depends upon Cond2.
- \* If the Cond1 & Cond2 is TRUE then O/P is TRUE else if the Cond1 is TRUE & Cond2 is FALSE then O/P is FALSE.

### Case 2:-

If Cond1 is FALSE in AND operator then whatever it might be the Cond2 the O/P is FALSE

Syntax :-

where Cond1 and Cond2 and Cond3 .....;

## ② OR Operator :-

\* when either one of the Condition is mandatory to execute then we should go "OR" operators.

Symbol:-



TRUTH TABLE

Cond1	Cond2	O/P
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

Case 1:-

If Cond1 is TRUE whatever it might be Cond2 the output of OR operator is TRUE.

Case 2:-

\* If Cond1 is FALSE the output of 'OR' operator is depends on Cond2.

\* If Cond2 is TRUE then the O/P is TRUE else if the Cond2 is FALSE then O/P is FALSE.

Syntax:-

where Cond1 or Cond2 or Cond3 ----;

22/7/21

display

Page No.: 57

Date:

youva

- ① WAPD employee details only the employee should be working as clerk (and) the salary should be less than 3000

→ cond  
↓  
operator

→ cond2

ename	Job	Sal	
Amith	Salesman	5000	x
Smith	Clerk	2500	✓
Aakash	Analyst	3000	x
Allen	Clerk	4000	x
Blalce	Salesman	1000	x
Miller	Clerk	2000	✓

①  $\Rightarrow$  salesman = clerk Select \*  $\rightarrow$  ①

from emp  $\rightarrow$  ①

where lower(Job) = 'clerk' and sal < 3000;  
 $\rightarrow$  ②

Salesman = clerk  $\rightarrow$  F and  $5000 < 3000 \rightarrow$  F  $\rightarrow$  F

Clerk = clerk  $\rightarrow$  T and  $2500 < 3000 \rightarrow$  T  $\rightarrow$  T

Analyst = clerk  $\rightarrow$  F and  $3000 < 3000 \rightarrow$  F  $\rightarrow$  F

Clerk = Clerk  $\rightarrow$  T and  $4000 < 3000 \rightarrow$  F  $\rightarrow$  F

Salesman = Clerk  $\rightarrow$  F and  $1000 < 3000 \rightarrow$  F  $\rightarrow$  F

Miller = Clerk  $\rightarrow$  T and  $2000 < 3000 \rightarrow$  T  $\rightarrow$  T  
Clerk

%/p	ename	Job	sal
	Smith	Clerk	2500
	Miller	Clerk	2000

WAPD employee details Same as above question but the salary is greater than 3000.

Select \*

from emp

where lower(Job) = 'clerk' and sal > 3000;

%/p :- no rows selected Allen clerk 4000

display

② WQN employee details only the employee's should be working for a department number 20 and their manager number should be 7698.

↓ cond①

Select \*

from emp

where deptno = 20 and mgr = 7698;

③ WQN employee name, designation, date of joining only employee number should be less than 7900 and they should be working for a department number 30.

Select \* ename, job, hiredate

from emp

where empno &lt; 7900 and deptno = 30;

④ WQN employee name, employee number, salary, Annual salary only the Annual salary should be 9600 and the employee name should be Smith.

↓ cond 2

AND

Select ename, empno, sal, sal\*12 as Ann\_Sal

from emp

where sal\*12 = 9600 and uper(ename) = 'SMITH';

⑤ WQN employee details only the employee should be hired after a year 80. and they should be getting exactly 1500.

Select \*

from emp

where hiredate &gt; '01-JAN-1981' and comm = 1500;

(or)

where hiredate &gt; '31-DEC-1980' and comm = 1500;

- ⑥ WQD employee name, employee number, manager number, department number only employee should be working ~~as~~ in salesman and they should be hired before November 87.

Select ename, empno, mgr, deptno

from emp

where <sup>lower job</sup> (ename) = 'salesman' and hiredate <= '31-OCT-87';  
(or)

where lower(job) = 'Salesman' and hiredate < '01-Nov-87';

~~OR query~~

- ⑦ WQD employee details only the employee should be working in department number 10, 20.

Select \*

from emp

where deptno = 10 or deptno = 20;

Note:-

\* when C.N is same but the values are different  
then it is better to go "OR" operator.

\* All relational operators will accept single value on LHS side and single value in RHS side.

Example

where deptno = 10, 20

- ⑧ WQD employee name, designation, Commission only the employee should be <sup>display</sup> working as salesman, Analyst.

↳ condition

Select ename, job, comm

from emp

where lower(job) = 'salesman' or lower(job) = 'Analyst';

display

- ⑨ WQD employee details only the employee should be reporting to a manager number 7698, 7839, 7788;

Select \*

from emp

where mgr = 7698 or mgr = 7839 or mgr = 7788 ;

- ⑩ WQD employee name, employee number, salary, new sal with 1000 increment only the employee should be earning salary 1500, 3000

Condition

Sal ename, empno, sal, Sal + 1000 as new\_sal

from emp

where sal = 1500 or sal = 3000 ;

- ⑪ WQD employee details only for a employee blake, miller, turner.

Select \*

from emp

where lower(ename) = 'blake' or lower(ename) = 'miller'  
or lower(ename) = 'turner' ;

- ⑫ WQD employee details only the employee number should be 7900, 7698

Select \*

from emp

where lower(empno) = 7900 or empno = 7698 ;

~~AND, OR combine queries~~

Page No.:

61

Date:

you're

- (13) W/QD employee details only employee should be working from a department <sup>display</sup> number 20, 30 and their commission should be less than 2000

Select \*  
from emp

where (deptno=20 or deptno=30) and (comm<2000);

Note: when two operators

P keep parenthesis for e

when two operators in a single query then we should use parenthesis to separate the condition

1 Cond = 1 ( )

- (14) W/QD employee details, only employee should be working as salesman, clerk, manager and their manager number should be 7698.

Select \*

Select \*

from emp

where (lower(job)='salesman' or lower(job)='clerk' or lower(job)=manager) and (mgr=7698);

- (15) W/QD employee name, manager number, date of joining only the employee salary should be 5000 and the employee number should be 7839, 7698, 7900.

Select ename, mgr, hire date

from emp

where (sal=5000) and (empno=7839 or empno=7698 or empno=7900);

display

- (16) WQD employee name, employee number, annual salary only the annual salary should be 60000, 36000 (and) they should be working for a department number 10, 20, 30
- ↑  
cond1  
↓ cond2

Select ename, empno, Sal \* 12 as Ann-Sal

from emp

where (Sal \* 12 = 60000 or Sal \* 12 = 36000) and (deptno = 10 or deptno = 20 or deptno = 30);

### Assignment - 4

display

- (1) WQD employee name, designation, hiring date and also annual salary those who are working as analyst, manager and they should be earning by exactly 1250, 300, 800
- ↑  
cond1  
↓ cond2  
operator  
1250, 300, 800

Select ename, job, hiredate, Sal \* 12

from emp

where (lower(job) = 'analyst' or lower(job) = 'manager') and  
(Sal = 1250, 300, 800);  
(Sal = 1250 or Sal = 300, or Sal = 800);

- (2) WQD employee details those who are hired on 31-dec-1991, 01-may-1981 and they should be earning commission more than 200 and their employee name should be foord, James.
- ↑  
cond1  
↓ cond2  
↓ cond3

Select \*

from emp

where (hiredate = '31-DEC-1991' or hiredate = '01-MAY-1981') and (comm > 200) and (lower(ename) = 'foord' or lower(ename) = 'james')

display

- (3) WAP employee name, salary, annual salary only those who are earning annual salary should be more than 500. and they should be earning Commission <sup>↳ cond<sup>1</sup></sup> 300,500,1400 and they should be reporting to a <sup>↳ cond<sup>2</sup></sup> manager number <sup>↳ cond<sup>3</sup></sup> 7698.

Select ename, sal, sal\*12 Ann-sal  
from emp

where (sal\*12 > 500) and (comm = 300,500,1400) and  
(mgr = 7698)

where (sal\*12 > 500) and (comm = 300 or comm = 500 or comm = 1400)  
and (mgr = 7698);

- (4) WAP employee details only those who are hired before 01-MAY-1981 and they should be working as salesman, analyst and display %p in desc order <sup>↳ cond<sup>1</sup></sup> based on <sup>↳ cond<sup>2</sup></sup> employee number.

<sup>↳ cond<sup>3</sup></sup>

Select \*;

from emp;

where (hiredate < '01-MAY-1981') and (lower(job) = 'salesman' or  
lower(job) = 'analyst') <sup>and</sup>

order by empno desc;

<sup>display</sup>  
<sup>details</sup>

- (5) WAP employee details only those who are hired after 20-feb-1981 and they should be earning salary less than <sup>↳ cond<sup>1</sup></sup> 3000 and they commission should be more than <sup>↳ cond<sup>2</sup></sup> 1000.

<sup>↳ cond<sup>3</sup></sup>

Select \*;

from emp;

where (hiredate >= '20-FEB-1981') and (sal < 3000) and  
(comm > 1000)

display

⑥ WAP employee name, salary and also new salary with 100rs increment and new commission with 500rs increment only the employees those who are reporting to a manager number 7839, 7698 and there employee name should be allen, jones, turner and they should be working to a department number 30 and display the output in descending order based on new-salary.

Select ename, sal, sal+100 new\_sal, comm+500 new\_comm  
from emp

where (mgr=7839 or mgr=7698) and (lower(ename) = 'allen' or  
lower(ename) = 'jones' or lower(ename) = 'turner') and (deptno = 30)

order by sal+100 desc;  
(or)

order by new\_sal desc;

⑦ WAP employee details only those who are earning salary less than 3000 and they should be hired before 22-feb-1981 and there department number should be 10,20,30 and display the output in descending order based on salary.

Select \*

from emp

where (sal < 3000) and (hiredate <= '22-FEB-1981') and  
(deptno = 10 or deptno = 20 or deptno = 30)

order by sal desc;

## SQL statements / language's:-

We have 5 different languages / statements.

### ① Data definition language (DDL) :-

① DDL

#### ⓐ Create :-

It is used to create a table or an object.

##### Syntax:-

Create Table table\_name (

column\_name1 datatype not null/ null,

column\_name2 data type not null/ null,

;

Constraint constraint\_name unique (column\_name),

Constraint constraint\_name primary key (column-name),

Constraint constraint\_name check (condn),

Constraint constraint\_name foreign key (column-name) Refer  
-ences parent table (column-name)

);

Any name  
(user-defined  
constraint)

#### ⓑ Alter :-

Alter statement is used to modify the table structure

##### 1. To add a column:-

Alter table table-name

Add column-name datatype constraint ;

##### 2. To drop a column:-

Alter table table-name

Drop column column-name ;

##### 3. To rename a column:-

Alter table table-name

Rename column oldcolumn-name to newcolumn-name;

##### 4. Drop :-

### ③ Drop :-

Drop is used to delete the table permanently from a database (db).

`Drop table table-name ;`

From oracle 10g onwards we have a concept of recycle bin, we can get back the table from recycle bin by using flashback statement.

`Flashback table table-name to before drop ;`

We can also delete the table from recycle bin by using purge statement.

`Purge table table-name ;`

### ④ Truncate :-

Truncate statement is used to delete all the records from the table permanently.

`Truncate table table-name ;`

### ⑤ Rename :-

Rename statement is used to change the table-name permanently.

`Rename old-table-name to new-table-name ;`

## ② Data manipulation language (DML) :-

### ① Insert :-

Insert statement is used to add a record in a table.

`Insert into table-name values (v1, v2, v3...);`

`Insert into table-name (c1, c2...) values (v1, v2);`

(b) update:-

update statement is used to modify the records present in a table. update table-name.

Set  $\rightarrow$  Set c.n<sub>1</sub> = val<sub>1</sub>, c.n<sub>2</sub> = val<sub>2</sub>...  
where <condn>;

(c) Delete:-

Delete statement is used to delete the records present in a table. delete from table-name.  
where <condn>;

(3) Transaction Control language:-(a) Roll back:-

roll back statement is used to undo the DML changes.

Rollback;

(b) Commit:-

Commit statement is used to save the DML changes.

Commit;

(c) Savepoint:-

Savepoint statement is used to mark the position in db(database) which helps the rollback statement to rollback to the particular position.

Savepoint savepoint-name;

Rollback to savepoint-name;

## (4) Data Control Language (DCL) :-

### (a) Grant :-

It is used to give permission to the user.

Grant SQL statements on table-name to username;

### (b) Revoke :-

It is used to get back the permission from the user.

Revoke SQL statements on table-name from username;

~~28/7/21~~  
Continuation of 'OR' and 'AND':

display

(17) WAP employee details only for a employee \*Smith, allen  
and the Commission should be more than 100.

↳ cond<sub>1</sub>

↳ cond<sub>2</sub>

Select \*

from emp

where (lower(ename)='Smith' or lower(ename)='allen') and  
(comm>100);

(18) WAP employee name, employee number, new salary with 1000rs increment only for a employee number 7698, 7839, 7900 and the new salary should be less than 5000.

Select ename, empno, sal+1000 as new\_sal  
from emp

where (empno=7698 or empno=7839 or empno=7900) and  
(sal+1000 < 5000);

display

- (17) WQD employee details only the salary should be in a range of 1000 to 5000.

cond

Select \*

from emp

where ( $sal \geq 1000$  and  $sal \leq 5000$ )Note:- for rangewe can write  $\geq$  and  $\leq$ 

display

- (20) WQD employee details only the employee number should be in a range of 7000 to 7900.

cond.

Select \*

from emp

where ( $empno \geq 7000$  and  $empno \leq 7900$ );Note:-

- (21) WQD employee name, salary, Annual salary: only the annual salary should be in a range of 5000 to 20,000 and the manager number should be 7698, 7839.

Select ename, sal, sal\*12 as ann\_sal

from emp

where ( $(sal * 12) \geq 5000$  and  $(sal * 12) \leq 20000$ ) and ( $mgr = 7698$  or  $mgr = 7839$ );

- (22) WQD employee name, manager number, Commission only the manager number should be in a range of 7600 to 7900 and display the output in descending order based on manager number

Select ename, mgr, comm

from emp

where ( $mgr \geq 7600$  and  $mgr \leq 7900$ ) and  
order by mgr desc;

(23) WQD employee details only the employee should be hired in a year 81

Select \*

from emp

where ( $\text{hiredate} \geq '01-\text{JAN}-1981'$  and  
 $\text{hiredate} \leq '31-\text{DEC}-1981'$ );

Note: year 81 means

$\geq '01-\text{Jan}-81'$  and  $\leq '31-\text{dec}-81'$

(24) WQD employee details, only the employee should be hired in feb 81.

Select \*

from emp

where ( $\text{hiredate} \geq '01-\text{FEB}-1981'$  and  $\text{hiredate} \leq '28-\text{FEB}-81'$ );

Note:

Feb - 81 means

$\geq '01-\text{Feb}-81'$  to  $'28-\text{Feb}-81'$ .

(25) WQD employee name, date of joining only the employee should be hired in a year 81 to 87.

Select ename, hire date

from emp

where ( $\text{hiredate} \geq '01-\text{JAN}-81'$  and  $\text{hiredate} \leq '31-\text{DEC}-87'$ );

(26) WQD employee details only the employees should be hired in feb 81 to november 82.

Select \*

from emp

where ( $\text{hiredate} \geq '01-\text{FEB}-81'$  and  $\text{hiredate} \leq '30-\text{NOV}-82'$ );  
 (Or)

where  $\text{hiredate} \geq '01-\text{FEB}-81'$  and  $\text{hiredate} \leq '30-\text{Nov}-82'$ ;

## Special operators :-

① IN

- \* IN is multivalued operator.
  - \* IN operator will accept single value in LHS side and multiple values in RHS side.
  - \* IN operator will perform equalTo (=) comparaison.
  - \* IN operator will compare single LHS value
  - \* In IN operator single LHS value will compare with all the values present in RHS side.
  - \* IN operator will execute TRUE if any one condition/Comparaison is TRUE.
  - \* IN operator will Execute FALSE if all conditions/Comparaison is FALSE
  - \* IN operator is working similar to OR operator.

## Syntax :-

where LHS-value IN ( RHS-value );

4

( value1, value2, value3, value4, ... )

### Example 1 :-

10 in (10, 20, 30)

$$\left. \begin{array}{l} 10 = 10 \rightarrow T \\ 10 = 20 \rightarrow F \\ 10 = 30 \rightarrow F \end{array} \right\} \rightarrow T$$

### Example 2 :-

10 in (20, 30, 40)

$$\left. \begin{array}{l} 10 = 20 \rightarrow F \\ 10 = 30 \rightarrow F \\ 10 = 40 \rightarrow F \end{array} \right\} \rightarrow F$$

## ② BETWEEN :-

- \* when the input is in a form of range then it is better to go BETWEEN operator.
  - \* Always lower limit value written first and then write upper limit value. like

Ex :-

Write 20 and 30, 50 and 60, 800 and 900.

- \* IN BETWEEN operator in front of lower-limit value and upper limit value like

Ex:-

BETWEEN 20 and 30, BETWEEN 50 and 60.

- \* BETWEEN operator will work like inclusive range.  
(it means including lower-limit and upper-limit value)
  - \* BETWEEN will <sup>execute</sup> return TRUE when all the conditions/  
Comparation is TRUE.
  - \* BETWEEN will return (execute) FALSE when any one of the  
Condition/Comparation is FALSE.
  - \* BETWEEN is working similar to AND operator.

## Syntax:-

### Example 1:

90 between .80 and 200

$$q_0 \geq 80 \rightarrow T \quad \text{and} \quad q_0 \leq 200 \rightarrow T \rightarrow T$$

example 2 :-

90 between 100 and 200

$q_0 \geq 100 \rightarrow F$  and  $q_0 \leq 200 \rightarrow T \rightarrow F$

- ① WQD employee / details only the employee should be working for a department number 10, 20, 30.
- display  
↳ condition

```
Select *  
from emp  
where deptno in (10, 20, 30);
```

Note:-

when the column name is same but the values are different then it is better to go IN operator.

- ② WQD employee name, employee number, manager number only the employee should be reporting to a manager number 7698, 7839 and display the output in descending order based on employee number.

```
Select ename, empno, mgr  
from emp  
where mgr in (7698, 7839)  
order by empno desc;
```

- ③ WQD employee / details only the employee should be working as salesman, clerk and the salary should be less than 5000.

```
so sal * select *  
from emp  
where lower(job) in ('salesman', 'clerk') and  
sal < 5000;
```

- ④ WQD employee / details only the employee should be earning salary 1250, 300 and they should be working for a department 30.

↳ condition      Select \* from emp

```
where (sal in (1250, 300)) and (deptno = 30);
```

(or)

where (sal in (1250, 300)) and (deptno in (70));

Note:-

- \* when single value in LHS-side and single value in RHS-side then it is better to go 'OR' operator.
- \* when single value in LHS-side but we should compare with the multiple values in RHS-side then it is better to go 'IN' operator.

⑤ ~~W/OD employee details only the salary should be in a range of 1000 and 5000.~~  
 ↓ cond

Select \* from emp  
 where sal in (1000, 5000);

where sal between 1000 and 5000;

(or)

where (sal between 1000 and 5000);

⑥ ~~W/OD employee name, salary, annual salary only the annual salary should be in a range of 10000 to 50000.~~  
 ↓ cond

Select ename, sal, sal\*12 as Ann-Sal  
 from emp

where sal\*12 between 10000 and 50000;

(or)

where (sal\*12) between 10000 and 50000;

(or)

where ((sal\*12) between 10000 and 50000);

Note:-

We cannot place the parenthesis after 'Between' to the values literals.

We can place the parenthesis after 'IN' is mandatory to the literals.

display

- (7) WAP employee name, employee number, department number only for a employee smith, allen, blake and the employee number should be in a range of 7000 to 7900.  
 ↴ cond②

Select ename, empno, deptno  
 from emp

where (lower(ename) in ('smith', 'allen', 'blake')) and  
 (empno between 7000 and 7900);

- (8) WAP employee details only for a employee number 7499, 7654 and the Commission should be in a range of 1000 to 2000.  
 ↴ cond②

Select \*

from emp

where (empno in (7499, 7654)) and

(Comm between 100 and 2000);

- (9) WAP employee details only employee should hired in a year 81.  
 ↴ condition

Select \*

from emp

where (hiredate between '01-JAN-1981' and '31-DEC-1981');

- (10) WAP employee details only employee should be hired in a month feb-81 and they should be working as analyst, clerk.  
 ↴ cond①  
 ↴ cond②

Select \*

from emp

where (hiredate between '01-FEB-81' and '28-FEB-81') and  
 (lower(job) in ('analyst', 'clerk')).;

# Assignment - 5

Page No.:	76
Date:	youva

- ① WQD employee name, designation, hiring date and also annual salary those who are working as analyst, manager and they should be earning salary exactly 1250, 300, 800

Select ename, job, hire date,  $\text{Sal} * 12$  as ann-sal  
from emp

where  $(\text{lower}(\text{job}) \text{ in } ('analyst', 'manager'))$  and  
 $(\text{Sal} \text{ in } (1250, 300, 800))$ ;

- ② WQD employee details those who are hired on 31-dec-1991, 01-may-1981 and they should be earning Commission more than 200 and their employee name should be ford, james

Select \*

from emp

where hiredate between '31-DEC-1991'

where  $(\text{hiredate} \text{ in } ('31-DEC-1991', '01-MAY-1981'))$

and ( $\text{Comm} > 200$ ) and  $(\text{lower}(\text{ename}) \text{ in } ('ford', 'james'))$ ;

- ③ WQD employee name, salary, annual salary only those who are earning annual salary should be more than 500 and they should be earning Commission 300, 500, 1400 and they should be reporting to a manager number 7698.

Select ename, sal,  $\text{Sal} * 12$  as ann-sal

from emp

where  $(\text{Sal} * 12 > 500)$  and  $(\text{Comm} \text{ in } (300, 500, 1400))$  and

$(\text{mgr} = 7698)$  ;

(cor)

$(\text{mgr} \text{ in } (7698))$  ;

Possibility to write  $(\text{mgr} \text{ in } (7698))$  but not in standard way ..

display

- (4) WAD employee details only those who are hired / before 01-may-1981 and they should be working as salesman, analyst and display the output in descending order based on employee number.

↳ cond(3)

Select \*

from emp

where (hiredate <= '01-MAY-1981') and (lower(job) in ('salesman', 'analyst'))

Order by empno desc ;

display

- (5) WAD employee details only those who are hired / after 20-feb-1981 and they should be earning salary less than 3000 and their Commission should be more than 100.

↳ cond(5)

Select \*

from emp

where (hiredate >= '20-FEB-1981') and (sal < 3000) and (comm > 100);

display

- (6) WAD employee name, salary and also new salary with 100rs increment and new Commission with 500rs increment only the employees those who are reporting to a manager number 7839, 7698 and their employee name should be allen, jones, turner and they should be working to a department number 30 display the output in descending order based on new salary.

↳ cond(6).

Select ename, sal from emp

where (sal+100 as new-sal) and (comm+500 new-comm) and (mgr in -(7839, 7698)) and (lower(ename) in ('allen', 'jones', 'turner')) and (deptno = 30)

Order by sal+100 desc ;

Select ename, sal, sal+100 as new\_sal, Comm+500 as new\_Comm  
from emp

where (mgr in (7839, 7698)) and (lower(ename) in ('allen', 'jones',  
'turner')) and (deptno = 30)

order by sal+100 desc;

display

⑦ WAP employee details only those who are earning salary less than 3000 and they should be hired before 22-feb-1981 and their department number should be 10,20,30 and display the output in descending order based on salary.

Select \*

from emp

hiredate < '22-FEB-1981'  
(or)

where (sal < 3000) and (hiredate <= '21-FEB-1981') and

(deptno in (10, 20, 30))

order by sal desc;

⑧ WAP employee details those who are earning salary in a range of 1000 to 5000 and those should be working as manager, salesman, analyst.

\* In Between?

Select \*

from emp

where (sal between 1000 and 5000) and

(lower(job) in ('salesman', 'manager', 'analyst'))

In range only.

(or)

Select \*

from emp

where (sal >= 1000 and sal <= 5000) and

(lower(job) = 'salesman' or lower(job) = 'manager' or lower(job) = 'Analyst');

display

⑨ WAP employee details those who are hired in year 81.  
 ↳ condition

Select \*

from emp

where hiredate between '01-JAN-1981' and '31-DEC-1981';  
 (or)

Select \*

from emp

where hiredate >= '01-JAN-1981' and hiredate <= '31-DEC-1981';

⑩ WAP employee details those who are hired on feb-81.  
 ↳ display ↳ condition

Select \*

from emp

where hiredate between '01-FEB-81' and '28-FEB-81';  
 (or)

Select \*

from emp

where hiredate >= '01-FEB-81' and hiredate <= '28-FEB-81';

21/5/21

Page No.:

80

Date:

youva

### ③ LIKE operator :-

LIKE operator is used to perform "pattern matching / wild-card search".

'pattern matching / wild-card search' :-

It is a combination of "ordinary characters and special characters".

#### ① ordinary characters :-

ordinary characters are the characters which does not have any special significant. (there is no meaning).

#### ② special character :-

Special character are the characters which have some special significant (there is a meaning).

\* There are two special characters present in SQL.

① percentile (%)

② underscore (-)

In another words percentile (%) character will accept all characters at any number of times.

It is a special character it will accept all characters at any number of time.

All characters

↓  
[A-Z]

[a-zA-Z]

[0-9]

[!-#]

(Except % and -)

any number of time

↓

min → 0 char occurrence.

max → ∞ occurrence

Normal words: All values which are present in keyboard in a laptop  
out p: except (%) and (-) they are all known as ordinary characters.

In another word the underscore character will accept all characters at only one time

Page No.:

81

Date:

youva

### (b) underscore (-) :-

It is a special character it will accept "all characters at only one time".

all characters



[A - Z]

[a - z]

[0 - 9]

[\\$ - #]

(except % and \_)

At only one time



min → 1 occurrence

max → 1 occurrence.

Note: every, one occurrence will be denoted as one underscore (-).

Note:-

If they particularly mentioned some value in a query. depends upon values. like if they mention as 4th letter then (- - - S)

Ex:- b

4th letter should be 'S' / 3rd letter 'n' / 2nd letter 'O  
(' - - - S') / (' - - n') / (' - O').

We should terminate with single quote (' ').

Syntax :-

where LHS value LIKE 'pattern/wild-card Search';

↓

c.n/enpr

↓

Combination of 'ordinary characters and special characters'.

① Write pattern ename should begins with 'S' character.

' S % '

Here

S = ordinary character

% = special character

Ex:-

↑ ordinary char  
Gamanth

Sam

Suneel

→ special character.

(2) write pattern ename should ends with 's' char.

'%s'  
Ans -  
Suhar, Vishwas, ~~Srinivas~~ <sup>special char (%)</sup> ordinary char (s)

(3) write pattern ename should consists 's' char.

'%s%' <sup>anywhere</sup>

(4) write pattern ename should begins with two consecutive 's' chara  
-cter. (consecutive means one after another)

'ss%'

(5) write pattern ename should end with two consecutive 's' char.

'%ss'

(6) write pattern ename should consists of two consecutive 's' char.

'%ss%'

(7) write pattern ename should consists of two 's' char.

'%s%s%'

(8) write pattern ename should consists of 'man' a substring.

'% man %'

(9) write pattern ename beginning 2nd char should be 's'.

' - s %'

(10) write pattern ename beginning 4th char should be 's'.

' --- s %'

(11) write pattern ename last 2nd char should be 'n'.

'% n -'

(12) write pattern ename last 3rd char should be 'S'.

'%S\_\_'

(13) write a pattern ename should consists of exactly 5th char.

'\_\_\_\_\_'

(14) write a pattern employee should be earning & digital sal.

'\_\_\_\_\_'

(15) write a pattern employee should be hired in a year 81.

'%81'

(16) write a pattern employee should hired in 'FEB-81'.

'% FEB-81'

(17) write a pattern employee should hired in 'FEB'.

(we can't go BETWEEN)

'%FEB%'

Q UERY'S

display

(1) W&D employee details only the employee name begins with 'a' character.

↳ Condition

Select \*

from emp

where lower(ename) like 'a%' ;

display

(2) W&D employee details only employee name should consist of 'a' character.

↳ Condition

Select \*

from emp

where lower(ename) like '%a%' ;

28/9/21

Page No.: 84  
Date: 20/09/21

display

- ① W<sub>QD</sub> employee name, employee number, designation, department number only the employee should be work for a department number 10,20,30 and the designation should be ending with character 'S'.

↳ cond(2)

Select ename, empno, job, deptno  
from emp

where deptno in (10,20,30) and (lower(job) like '%S');

display

- ② W<sub>QD</sub> employee details only the employee name should consist of 2 consecutive 'L' character and the salary should be in a range ↳ cond(1) 1000 to 5000.  
↳ cond(2)

Select \*

-from emp

where (lower(ename) like 'LL%') and

(sal between 1000 and 5000);

display

- ③ W<sub>QD</sub> employee details only the designation should ends with 'man' as a sub string and their manager number should be ↳ cond(1)  
7698, 7839. ↳ cond(2)

Select \*

from emp

where (lower(job) like '% man') and (mgr in (7698, 7839));

display

- ④ W<sub>QD</sub> employee details only the employee number should be in a range ↳ cond(1) of 7000 to 7900 and the designation should consists of 2 'a' char. ↳ cond(2)

Select \*

-from emp

where (empno between 7000 and 7900) and

(lower(job) like '%a%a%');

display

- ⑤ when employee details only the employee should be hired after a year 80, and their employee name beginning 2nd character <sup>(second)</sup> should consist of 'a' as a character.

Select \*

from emp

where (hiredate &gt;= '01-JAN-80') and (lower(ename) like '%a%')

Note:-

- \* If incase we write as(' -a') it means after 'a' character there should ~~man~~ be no character present.
- \* If we write as(' -a%') it means after 'a' character there is a chance to present any character or not. because special character (%) is percentile. Consist minimum occurrence is '0' and maximum occurrence is '∞'.

- ⑥ when employee name, employee number, annual salary only the employee name should be exactly in '5' character. and the annual salary should be <sup>(second)</sup> in a range of 10000 to 50000.

Select ename, empno, Sal \* 12 as ann\_sal

from emp

where (ename like '\_\_\_\_') and ((sal\*12) between 10000 and 50000);

display

- ⑦ when employee details only employee should be earning '4' digit salary and they should be working clerk, analyst.

Select \*

from emp

where (sal like '\_\_\_\_') and

(lower(job) in ('clerk', 'analyst'));

using this page  
on my own.

Page No.:

86

Date:

youva

Note:-

- \* AND is used for to separate the two Conditions.
- \* OR is used for to separate the single Conditions into two conditions.

(Hint point :- 'OR' operator will be useful for only the C.n.  
(c.n-column name) is single and the Values are different. @

(values - values are nothing but "literal's").

\* String/character or Date literal should be definitely enclosed with a single quote. otherwise it throws an error.

① Number literal

② String / character literal

③ Date literal.

Forgot about this  
Page  
No  
15 Complete

display

- ⑧ WAP employee details only the manager number last 2nd digit should be '9' and they should be earning salary less than 5000.  
 ↳ cond(1)  
 ↳ cond(2)

Select \*

from emp

where (mgr like '%9') and (sal &lt; 5000);

- ⑨ WAP employee name, date of joining, new salary with 1000rs increment only the employee should be hired in a year 81. and the sal should be less than comm.  
 ↳ cond(1)  
 ↳ cond(2)

Select ename, hiredate, sal+1000 as new sal

from emp

where (hiredate like '%81') and (sal &lt; comm);

- ⑩ WAP employee details only the employee should be hired in a year 'Feb-81' and they should be working for a department number 30.  
 ↳ cond(1)  
 ↳ cond(2)

Select \*

from emp

where (hiredate like '%FEB-81') and (deptno = 30);

display

- ⑪ WAP employee details only the employee name should consist of 'e' as a char. and they should be hired in a year 80 to 87.  
 ↳ cond(1)  
 ↳ cond(2)

Select \*

from emp

where (lower(ename) like '%e%') and

(hiredate between '01-JAN-80' and '31-DEC-87');

ownNote: -  
when ever they ask pattern matching with single employee name with multiple values then it should be better to place 'OR' operator in between them (n; empno like '%8%' or empno like '%9%')

Page No.:

88

you've

Note:-

\* If they are specifying exactly "1 year/month" then it is better to go LIKE operator.

(Ex:- exactly means)

(year 81, year98, FEB, FEB-81 etc....)

\* If they are specifying in range (not exactly year or month) then it is better to go BETWEEN operator.

(Ex:- Not exactly means)

(year 80 to 85, Feb-80 to Nov-80, 01-FEB-81 to 28-FEB-81  
(likewise etc...)).

(12) WAP employee details only designation should consist of 'es' as a sub string and they should be hired on <sup>display</sup> a month <sup>cond(1)</sup> december. <sup>cond(2)</sup>

Select \*

from emp

where (lower(job) like '%es%') and (hiredate like '%DEC%');  
<sup>display</sup>

(13) WAP employee details only employee should be hired on Feb-81 to Nov-85 and the manager number should begin with a digit '7'. <sup>cond(1)</sup> <sup>cond(2)</sup>

Select \*

from emp

where (hiredate between '01-FEB-81' and '30-Nov-85')  
and (mgr like '7%');

Note:-

If they are asking "from and to" in a query then it should be "BETWEEN operator".

display

- (4) ~~W<sup>Q</sup>O D employee details only employee should be earning '5' digit annual salary and they should be earning Commission in a range of 0 to 2000.~~

↳ cond ②

Select \*

from emp

where (salary like '\_\_\_\_') and (comm between 0 and 2000);

display

- (5) ~~W<sup>Q</sup>O D employee name/ only the employee name should begin's with 'a' or 's' character.~~

Select ename

from emp

where (lower(ename) like 'a%' or lower(ename) like 's%');

Note:-

One like operator will accept single value in LHS-Side and Single-pattern in RHS Side.

- (6) ~~W<sup>Q</sup>O D employee name/ only employee name should begin's with VOWELS. (a,e,i,o,u)~~

↳ condition-

Select \*

from emp

where lower(ename) like 'a%' or  
lower(ename) like 'e%' or  
lower(ename) like 'i%' or  
lower(ename) like 'o%' or  
lower(ename) like 'u%' ;

- (7) ~~W<sup>Q</sup>O D employee details only employee should be hired in a month December, February.~~

Select \*

from emp

where (hiredate like '%DEC%' or hiredate like '%FEB%');

# display Assignment - 6

Page No.: 90

Date:

youva

- ① WAP regions details only the region name should consist of 'a' character.

→ cond

Select \* ~~region~~

from regions

where lower(region\_name) like '%a%' ;

- ② WAP country name, region id for the Country id 'IN', 'us' and Country name should consists of two 'L'.

Select country\_name, region\_id

from countries

where (lower(country\_id) like 'IN' or lower(country\_id) like 'us')

and (lower(country\_name) like '%L%L%') ;

(or)

Select country\_name, region\_id

from countries

where (lower(country\_id) in ('IN', 'us')), and

(lower(country\_name) like '%L%L%') ;

- ③ WAP Country details only the Country id should consist of 'n' character and the region should be in a range of 1 to 4.

→ cond

Select \*

from countries

where (lower(country\_id) like '%n%') and

(region\_id between 1 and 4) ;

display

- (4) WAD Country details only Country id should be exactly two character and the Country name should be Japan, India, Italy.

Cond(1)

Cond(2)

Select \*

from Countries

where (Country-id like '--') and

(lower(Country-name) in ('Japan', 'India', 'Italy')) ;

display

- (5) WAD location details only the postal code should be exactly 4 digit and the location id should be in a range of 1800 to 2600.

Cond(1)

Select \*

from locations

where (Postal-code like '----') and

(location\_id between 1800 and 2600);

display

- (6) WAD street address only for the city 'Sydney', 'Bombay', 'Rome' and street address should be consist of 'rd' as consecutive 'rd' as mistake & sub string.

Select street\_address

from locations

where (lower(city) in ('sydney', 'Bombay', 'Rome')) and

(lower(street\_address) like '%rd%');

display

Mistake (location\_id)

- (7) WAD location details for Country id beginning second character should be 'N' and street address beginning with number '2' and postal code be more than 1000.

Select \* from locations

where (lower(location\_id) like '-N%') and (street\_address like '2%')

and (postal\_code &gt; 1000);

- ⑧ When city name, Country-id, Postal Code except city 'Tokyo' and Postal code in a range of 1000 and 5000, Street address should be consist of two consecutive 'a'. display

Select city, Country-id, Postal-code,  
from locations.

where, (Postal-Code between 1000 and 5000) and  
lower(city) != 'Tokyo'

display (lower(Street-address) like '%.aa%');

- ⑨ When locations details for the location id 1600, 1500, 1700 and Country id should be 'us' and Postal Code should be exactly 5 digit. cond① cond② cond③

Select \*  
from locations

where (location-id in (1600, 1500, 1700)) and (lower(country-id) = 'us')  
and (Postal-Code like '-----');

- ⑩ When location details only state province should consist of 'maha' as substring, City name should consist of begining third character 'm' and Postal Code should be in a range of 3,00,000 to 600000. cond③

Select \*  
from locations

where (lower(state-province) like '%maha%') and

(lower(city) like '--m%') and

(Postal-Code between 300000 and 600000);

Doubt Explain about Postal Codes.

IS operator:-

'IS' operator is used to compare 'NULL' Values.

Note:-

WON employee details only the employees those who are not earning any Commission.

Select \*

from emp

where comm=NULL ;

%:- No rows selected

\* According to above example we get a wrong output because

\* According to oracle RDBMS one NULL is not equal to another NULL ( $\text{NULL} \neq \text{NULL}$ ), in order to compare one NULL with another NULL we should go IS operator.

Characteristics of NULL:-

\* NULL is 'reserved keyword' in oracle RDBMS.

( $\because$  reserved = predefined, in built, built-in, default).

\* NULL is not representing for either it is '0' or 'space' because they are not null values.

\* Null means it is representing as "NOTHING" (No data).

\* NULL will not consume any memory allocation.

\* If we perform any Arithmetic operators with NULL the result will be NULL.

Example:-

Select 10+null  
from dual;

qp

10+null

-----

(Nothing will present)

### Syntax:-

where LHS value is NULL ;  
 ↓              ↓

Con/expr      No data (nothing)

\* IS operator will return TRUE if LHS value is NULL.

\* IS operator will return FALSE if LHS value is not NULL.

### TRUE in LHS value :-

If when we are presenting a NULL value in the column name depending on the

\* If when we are writing a column name in LHS-side if there is not data (nothing) will specified in that column either it may be one record or many records then we said it is a "TRUE".

### FALSE in LHS value :-

\* It means there is no empty records present in a specified column then it may be said as 'FALSE' value.

Earning → Something is present or some value is present  
 it may be written as FALSE → NOT NULL

Not earning → No value/nothing is present/empty → NULL

① WAP employee details only the employee those who are not earning any commission.

emp

ename comm

Allen 100

Smith 0

Blake 0

Turner

Miller 300

Here 100 is null → FALSE

NULL is null → TRUE

0 is null → FALSE

NULL is null → TRUE

300 is null → FALSE

O/P :→

ename comm

Smith

Turner.

Query

Select \* ...> (3)  
 from emp ...> (1)  
 where comm is null ; ....> (2)

(2) WQD employee details only the employee's those who are not reporting to any up the manager number.

Select \*  
 from emp  
 where mgr is null ;

Note:-

\* when the column name is same but values are different then it is better to go IN operator.

\* when the input is in a form of range then it is better to go BETWEEN operator.

\* when the input is in a pattern match/wild card search then it is better to use LIKE operator.

\* whenever we want to compare the NULL values with Null column name then it is better to use IS operator.

\* like operator will accept single value in LHS-side and single value in 'pattern/wild card' search in RHS-side.

Own Note:- writing a query in three ways.

SQL > select \* from emp

2 where deptno in(10,20); (IN)  $\Rightarrow$  Single LHS in multiple RHS

(OR)

2. where deptno=10 or deptno=20;  $\rightarrow$  Single LHS relational cond = Single RHS relational cond = single RHS

(OR)

3. where deptno != 30;  $\rightarrow$  where not deptno=30; {Not operator}

3. where deptno <> 30;

Not operator is from logical operator:-

\* whenever we want to "exclude the record" the we should go for 'NOT' operator.  
(Exclude  $\rightarrow$  except, ignoring, apart from, rejecting).

Symbol:-



Truth table

cond      o/p

TRUE      FALSE

FALSE      TRUE

my syntax

Syntax:-

where not condition;  $\rightarrow$  Struct syntax

Not  
where  $\star$  LHS-value relational operator RHS-value;  
|  
c.n/expres       $<$ ,  $>$ ,  $=$ ,  $!=$  (or)  $<$ , c.n/expres / literals  
 $<=$ ,  $>=$

\* NOT operator will return TRUE if Condition is FALSE.

\* NOT operator will return FALSE if Condition is TRUE.

① Display employee details except an employee working in a department number '30'.

Select  $\star$

from emp

where not deptno = 30 ;

(or)

where deptno != 30 ;

(or)

where deptno < > 30 ;

Note:-

when ever we are give a not symbols  $!=$  (or)  $<$   $>$ . then it will accept Single LHS-value and Single RHS-value. (we want to write multiple RHS then if is better to go (where not condition)).

(2) W<sup>Q</sup>D employee details except those are working as 'Salesman'.

↓ display                      ↓ condition

Select \*

from emp

where lower(job) = 'salesman';

(or)

where lower(job) != 'salesman';

(or)

where lower(job) < > 'salesman';

(3) W<sup>Q</sup>D employee details except an employee miller.

↓ display                      ↓ condition

Select \*

from emp

where lower(ename) != 'miller';

(4) W<sup>Q</sup>D employee details except any employee those are earning salary 3000.

↓ display                      ↓ condition

Select \*

from emp

where sal != 3000;

(5) W<sup>Q</sup>D employee details those who are not hired on '17-DEC-80'.

↓ display                      ↓ condition

Select \*

from emp

where hiredate != '17-DEC-80';

Note:-

If they ask a Question deptno 10, 20 we write → where deptno in (10, 20);

Instead of above Question they as except/exclude deptno 30.

Both outputs are same.

→ where not deptno=30;

30/7/21

## Combination of Special operators and NOT operator :-

### (1) NOT IN :-

- \* when LHS-value should not equal to any RHS-value then we should go "NOT IN" operator.

Syntax:-

where LHS-value NOT IN (Value1, Value2, Value3....) ;

↓	↓	↓
C.n/expr	2	1

### (2) NOT BETWEEN :-

- \* when LHS-value is not ~~BETWEEN~~-in/within/between a range of lower-limit and upper-limit then we should go "NOT BETWEEN" operator.

Syntax:-

where LHS-value NOT BETWEEN lower-limit and upper-limit ;

--↓--	↓	↓
C.n/expr	2	1

### (3) NOT LIKE :-

- \* when LHS-value is not matching RHS-side pattern then we should go "NOT LIKE" operator

Syntax:-

where LHS-value NOT LIKE 'pattern/wild card search' ;

↓	↓	↓
C.n/expr	2	1

### (4) IS NOT :-

- \* when LHS-value is not null value then we should go "IS NOT" operator.

Syntax:-

where LHS-value IS NOT NULL ;

↓	↓	↓
C.n/expr	1	2.

Note:-

In SQL (Structured Query language) we should write the conditions, statements / language's in a "simple english" manner.

- ① ~~WAP employee details only the employee should not be working for a department number 10, 20.~~

Select \*

from emp

where deptno not in (10, 20);

(or)

else we can write as where not deptno in (10, 20);

Note:-

Both the conditions are true in where clause. but mainly we can prefer for where deptno not in (10, 20) because it is a perfect & proper simple english.

- ② ~~WAP employee details except employee those are working as 'clerk', 'analyst'.~~

Select \*

from emp

where lower(job) not in ('clerk', 'analyst');

- ③ ~~WAP employee name, employee number, manager number. except a manager number 7698, 7900 and employee name should consist of 'a' character.~~

Select ename, empno, mgr  
from emp

where mgr not in (7698, 7900) and (lower(ename) like '%a%')

display  
cond①

(4) Who employee details except an employee allen, blake, turner and manager number should be in a range of 7600 to 7900.

↓ cond①

Select \*

from emp

where (lower(name) not in ('allen', 'blake', 'turner')) and  
(mgr between 7600 and 7900);  
display

(5) Who employee details only the salary should not be in a range of 1000 and 5000.

Select \*

from emp

where sal not between 1000 and 5000;

(6) Who employee details only employee number should not be in a range of 7000 and 7500 and they should be working for a department number 20,30

Select \*

from emp

where (empno not between 7000 and 7500) and  
(deptno in (20,30));  
display

(7) Who employee details only employee should not be hired in a year 81 to 87.

Select \*

from emp

where (hiredate not between '01-JAN-81' and  
'31-DEC-87');  
display

display.

- ⑧ WAP employee name, designation, date of joining only employee should not be hired in Feb-81 to Nov-82.
- ↳ cond 1

Select ename, designation, job, hire date  
from emp

where hiredate not between '01-FEB-81' and '30-NOV-82';  
display

- ⑨ WAP employee details only employee name should not consist of 'a' character.
- ↳ cond 2

Select \*

from emp

where lower(ename) not like '%a%' ;

- ⑩ WAP employee details only the designation should not consist of man as substring and display the output in descending order based on salary.
- ↳ cond 3

Select \*

from emp

where (lower(job) not like '%man%')

order by sal desc;

- ⑪ WAP employee details only employee name should begin with consonants.
- ↳ cond 4

opposite to AND

Not AND output

T and F and F and F and F → T

Select \*

from emp

where lower(ename) not like 'a%' and  
lower(ename) not like 'e%' and  
lower(ename) not like 'i%' and  
lower(ename) not like 'o%' and  
lower(ename) not like 'u%' ;

↳ F

display.

- (12) WAN employee details except an employee those who are hired in a month Feb, nov.

Select \*

from emp

where hiredate not like '%FEB%' and hiredate not like '%Nov%';  
(or)

where hiredate not like '---FEB---' and

hiredate not like '---Nov---';

Note:-

when they are asking about two like operators it should be better to keep AND in between of two like operators.

Ex:- LHS-value like RHS-value AND LHS-value like RHS-value

because in like operator single value in LHS-side and single-value in RHS side. (pattern/wild-card search)

Imp Note:-

when we are specifying "Not with the Combination of special operators".

Then if the Column name is same but the values are different then we should use AND operator

Ex:- let it be in a simple English language.

Jeewan and yaswin (Q) Jeewan, yaswin.

- (Q) class teacher said Jeewan, yaswin don't come to class?  
it means

Jeewan And yaswin not come to class. (✓)

It doesn't mean

Jeewan OR yaswin not come to class. (X)

∴ Here OR means any one may be come to class.

(13) WQD employee details only the employees those are earning some commission.

Select \*

from emp

where Comm is not null;

Note:-

Null can be only specified with nothing (no data)

display

(14) WQD employee details employee should be reporting to some manager number.

Select \*

from emp

where mgr is not null;

display

(15) WQD employee details only the employee should be earning some salary.

Select \*

from emp

where sal is not null;

## Functions:-

functions is a block of statement / set of statements it will get execute whenever ever it is called / invoke is called as function.

example:-

Void add( )

{

int a=10, b=20, c ;  
c = a+b ;

printf ("%d", c) ;

}

→ function name.

Set  
from entry

→ block / set of statement

\* functions are classified into 2 types:-

- (1) user defined function
- (2) pre-defined / built-in / in-built function

### (1) user defined function:-

\* The functions which are defined / developed by user, based on customer requirement is called as user defined function.

\* We can change / modify user defined function when the customer requirement is changed.

\* SQL will not support user defined function but C, C++, Java, Python, C#, PL/SQL etc... will support user defined functions.

### (2) pre-defined / built-in / in-built function:-

\* The functions which are automatically present in software (SW) whenever the software (SW) has been installed is called pre-defined function.

- \* We Cannot change/modify the pre-defined function.
- \* In SQL we will always use pre-defined/built-in/in-built function.

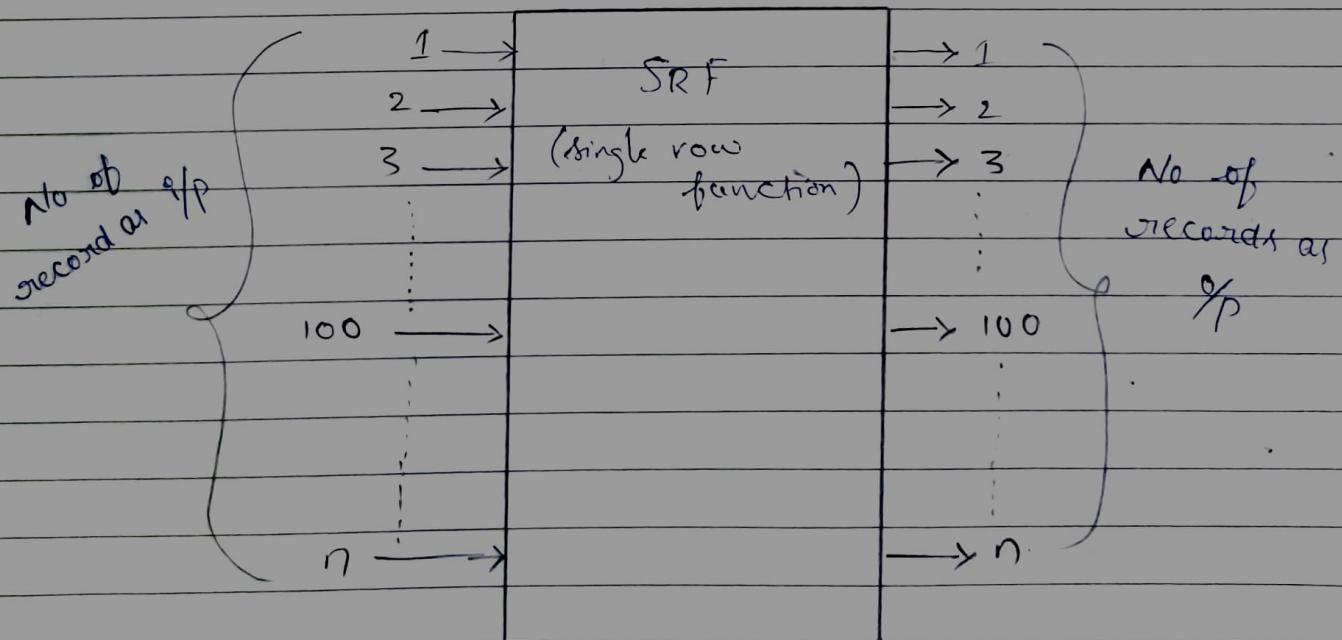
Pre-defined function are classified into 2 types.

- (1) single row function (SRF)
- (2) multi row function (MRF) / Aggregate / Group function.

### (1) Single - row function (SRF) :-

- \* SRF will accept 'n' records as input (I/P) and it will provide 'n' records as output (O/P).
- \* In SRF for every individual input has separate output - is present.
- \* In SRF number of outputs is directly proportional to number of inputs.

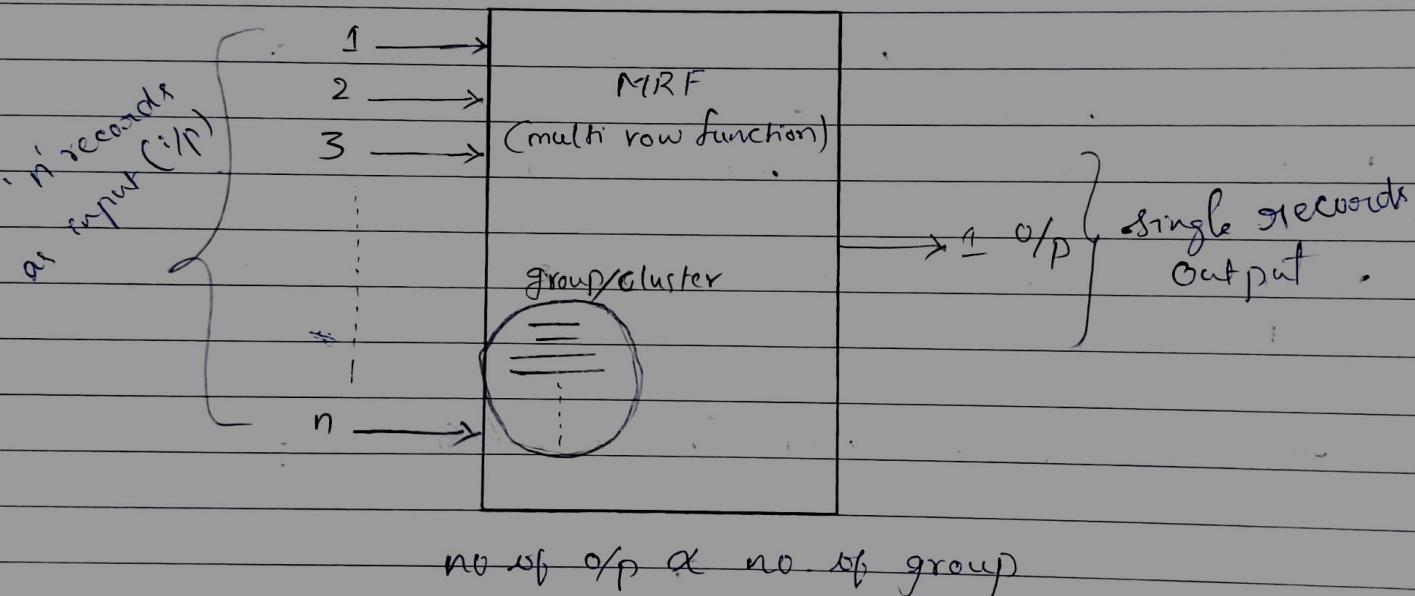
### (2) Multi - Row function (MRF) / Aggregate / Group function :-



no. of O/P & no of I/P

## ② Multi-row function (MRF) / Aggregate / Group function :-

- \* multi-row function (MRF) will accept 'n' records as input and it will generate create group/cluster and it will provide single record as output (%p).
- \* By default MRF will create single group and it will provide single record as output.
- \* In MRF no. of outputs is directly proportional to no. of group/cluster.



\* Multi-row function are classified into 5 types.

- (1) Count ( )
- (2) Min ( )
- (3) Max ( )
- (4) Sum ( )
- (5) Average ( )

### ① Count () :-

- \* This function is used to count no. of records in a table.
- \* We can pass \*, distinct, c.n., expr as an argument to count function.

Syntax:-

Count (\* / distinct / column / expr)

practically

SQL &gt; Select Count (\*)

2. from emp;

O/P

COUNT(\*)

14

SQL &gt; Select Count (sal)

2. from emp;

O/P

COUNT(SAL)

14

SQL &gt; Select Count (distinct sal)

2. from emp;

O/P

COUNT(DISTINCT SAL)

12

SQL &gt; Select Count (Sal \* 12)

2. from emp;

O/P

COUNT(SAL \* 12)

14

Note:-

\* when we are using distinct in count column name should be single

\* without distinct also we can specify single column name in count

Ex:-

① SQL > Select Count (sal) }  
2 from emp;② SQL > Select Count (sal, job) }  
2 from emp;

like wise

SQL > Select Count (distinct sal) }  
from emp;SQL > Select Count (distinct sal, job) }  
from emp;

## ② min ( ) :-

\* This function is used to obtain minimum value present in Cn/expr

\* We can pass Cn/expr as an argument to min function.

Syntax:- min (Cn/expr)

Practically

SQL > Select min (comm)

2. from emp;

O/P MIN (COMM)

0

SQL > Select min (sal)

2. from emp;

O/P MIN (SAL)

800

SQL > Select min (sal \* 12)

2 from emp;

O/P MIN (SAL \* 12)

9,600

As per ASCII → (American Standard Code for Information Interchange)

As per ASCII when ever we are passing (writing) a column name/ expression as an argument to find out the max and min values from 'string/character' names from Cn/expr. then it will generate the output/result minimum (or) max column name base on using the min and max functions.

SQL > Select min (ename)

2 from emp;

O/P MIN (ENAME)

ADAMS

(3) Max():

- \* This function is used to obtain maximum value present in column/expr.
- \* we can pass column/expr as an argument to max function.

Syntax:-  $\text{max}(\text{c.n}/\text{expr})$

Practically

SQL > Select max(sal)

2. from emp;

O/P MAX(SAL)

5000

SQL > Select max(sal\*12)

2. from emp;

O/P

MAX(SAL\*12)

-----  
60000

SQL > Select max(ename)

2. from emp;

O/P

MAX(ENAME)

-----  
KIRAN

SQL >

(4) Sum():

- \* This function is used to obtain summation of all the values present in column/expr.
- \* we can pass column/expr as an argument to sum function.

Syntax:-

$\text{sum}(\text{c.n}/\text{expr})$

Practically

SQL &gt; Select sum(comm)

2. from emp;

O/P sum(comm)

2200

SQL &gt; Select sum(comm+100)

2 from emp;

O/P summ(comm+100)

2600

Note:-

rule Cannot pass (write) an argument string/character in sum()

### ⑤ Average () :-

- \* This function is obtain used to obtain average value present in c.n/expr.

- \* It can pass (write) c.n/expr as an argument to average function.

$$\text{Avg} = \frac{\text{sum of all the values.}}{\text{total no of values.}}$$

Syntax:-

Avg (c.n/expr)

Practically

SQL &gt; Select avg(comm)

2 from emp;

O/P avg(comm)

550

SQL &gt; Select 2200/4

2 from dual;

O/P 2200/4

550

SQL &gt; Select avg(sal)

2. from emp;

O/P avg(sal)

2673.21429

### Note:-

If cannot pass (write) 'String/char' as an argument in the ~~MAX()~~ function.  
Avg

~~2/08/21~~

### Note of MRF :-

- \* All MRF will ~~not~~ write alias-name.
- \* ~~multi~~: All multi-row function (MRF) will ignore null values.
- \* All multi-row function (MRF) will accept duplicate values, if we want to ignore them we use distinct clause.
- \* We can write multiple MRF in single select statement.

### Example:-

```
SQL> Select Count(*), min(sal), max(sal)
      from emp;
```

O/P

COUNT(*)	MIN(SAL)	MAX(COMM)
14	800	1400

- \* All multi-row function (MRF) will accept single argument. or Single c.n.

### Example:-

```
SQL> Select min(sal, comm)
      from emp;
```

O/P:- Invalid number of arguments.

- \* We cannot write (pass) combination of c.n and MRF (multi-row function) in select statement.

If we want to write then we should go by "group by clause".

### Example:-

```
SQL> Select min(sal), Comm
      from emp;
```

O/P:- not a single-group group function.

\* We cannot pass (write) a multi-row-function (MRF) as a condition in where clause.

If we want to write then we should go "having clause".  
example:-

SQL> Select Count(\*), sum(sal), min(sal)  
from emp  
where min(sal) > 5000;

Ex:-  
group function is not allowed here.

\* MRF will execute group by group.

Note:-

Select (Sal), (min(Comm))  
↓              ↓  
C.N or MRF

when we are writing combination of C.N and MRF in the select clause/statement it will get confuse to which result/output want to give because.

\* C.N will when we are giving C.N as an argument in select clause/statement the C.N name will execute by row by row/record by record like wise but  
\* when we are giving MRF as an argument in select clause/statement the MRF will execute group by group.

so it throws error message

where min(sal) > 5000;  
↓              ↓  
group by group record by record.

We cannot use combination of C.N & MRF in Select clause or where clause.

display

Page No.: 113

youva

Date:

- ① WAP number of employee, minimum; maximum salary, maximum salary only the employees those who are working in department number 10, 20, 30
- ↓ condition

Select Count(\*), min(sal), max(sal)  
from emp  
where deptno in (10, 20, 30);

display

- ② WAP number of employees, minimum Commission, maximum Commission only the employee should to a manager number 7698, 7839, 7900.
- ↓ cond

Select count(\*), min(comm), max(comm)  
from emp  
where mgr in (7698, 7839, 7900);

display

- ③ WAP recent hiredate, oldest hiredate only the salary should be in a range of 1000 to 5000.
- ↓ condition

Select max(hiredate), min(hiredate)  
from emp  
where sal between 1000 and 5000;

Example:-

02-08-2021 → recent → Max() } we should decide comparing  
15-08-1947 → oldest → Min() } w/b years.

02-09-2021 → recent → Max() } if both years are equal  
15-08-2021 → oldest → min() } then we should compare with months.

02-09-2021 → recent → Max() } if both months and years are  
01-09-2021 → oldest → min() } same then we should compare the dates.

Note:- Both dates also equal then we consider recent and older as a same date.

# Assignment - 8

Page No.: 114  
 Date: youva

- (4) WAP no of employees, total salary, total Commission only the employee should be working as "salesman, analyst".
- display
- Condition

Select Count(\*), Sum(sal), Sum(comm)  
from emp

where Lower(job) in ('salesman', 'analyst');

- (5) WAP Average salary, Average Commission only the employee number should be in a range of 7000 to 7900.
- display
- Condition

Select Avg(sal), Avg(comm)

from emp

where empno between 7000 and 7900;

- (6) WAP minimum annual salary, max annual salary only the employee should be hired after a year 80.
- display
- Condition

Select min(sal\*12), max(sal\*12)

from emp

where hiredate like '%80'; hiredate >= '01-JAN-81';

- (7) WAP minimum annual Commission, maximum annual Commission total annual Commission only the employee name should begins 'a' or 's' character.
- display

Select min(<sup>comm</sup>sal\*12), max(comm\*12), sum(comm\*12)

from emp

where (lower(ename) like 'a%' or lower(ename) like 's%');

- (8) WAP total annual salary, total annual Commission, average annual salary, average annual Commission only employee who would be earning some Commission
- display

Select sum(sal \* 12), sum(comm \* 12), avg(sal \* 12), avg(comm \* 12)  
from emp

where comm is not null;

- (9) WQD minimum new salary with 1000rs increment, maximum new salary with 1000rs decrement except an employee who is working as president.
- display  
condition.

Select min(sal + 1000), max(sal - 1000)  
from emp

where lower(job) <> 'president';

- (10) WQD recent hiredate after one year, oldest hiredate before one year only the employee should be reporting to some manager number.
- display

Select max(hiredate + 365), min(hiredate - 365)

from emp

where mgr is not null;

(Note:-

After + 365

before - 365

- (11) WQD first employee number, last employee number only the employee should be earning some salary.

Select min(empno), max(empno)

from emp

where sal is not null;

Note:- After  $\rightarrow ( \geq )$ , before  $\rightarrow ( \leq )$  we can write in MRF.

After  $\rightarrow ( \geq )$ , before  $\rightarrow ( \leq )$  we can use in conditions to specify conditions on 'operators'

## Group by clause :-

- \* To write a combination of Cn and MRF in select clause or statement then we should go group by clause.
- \* By Default all multi-row function (MRF) will create single group and it will provide single record as output.
- \* If we want multiple records as output from MRF then we should create multiple groups.
- \* Inorder to get multiple groups we should go group by clause.

## Syntax:-

```

Select group-by expression --> (1)
from table-name --> (1)
where <cond/cond's> --> (2)
group by reference's --> (3)           c.n/expr/MRF
order by group-by-expression [ASC]/DESC ; --> (5)
                                         ↓ c.n/MRF/Expr
  
```

- \* The Combination of Cn/expr and MRF is called a group by expression.
- \* The Cn/expr present in group by clause is called as reference.
- \* whenever we are writing "single Cn/reference" in group by clause, then group by clause will create "main group".
- \* whenever we are writing "multiple Cn/expr/reference's" in group by clause, then group by clause will create "sub-group" inside "main group" it is also known as sub-grouping.

03/8/21

Sname	branch	Per
Dinga	CS	90
Dingi	EC	80
Penga	ME	40
Pengi	CS	85
Manga	CIVIL	35
Mangi	IS	65

Q) WQD branch wise no. of students.

(or)

WQD no. of students in each branch.

Select branch, Count (\*)  $\rightarrow$  3  
 from student  $\rightarrow$  1 group by express  
 group by branch;  $\rightarrow$  2  
 $\downarrow$  reference (c.n.)

CS	EC	ME	CIVIL
Dinga CS 90	Dingi EC 80	penga ME 40	mang Civil 35
pengi CS 85			

IS

mangi IS 65
-------------

Main group

O/P

branch	Count (*)
CS	2
EC	1
ME	1
CIVIL	1
IS	1

② WAD branch wise no of students, min percentage and maximum percentage except mechanical branch  
 display  
 & cond.

Select branch, count(\*), min(per), max(per)  
 from student

where lower(branch) != except 'ME'

group by branch;

③ WAD percentage wise no of students only the percentage  
 should be in a range if 40 to 90.  
 display  
 & cond

Select per, count(\*)

from student

where per between 40 and 90

group by per;

④ WAD branch wise total percentage, Average percentage, only  
 the student name should consists of 'a' character.  
 display  
 & cond

Select branch, sum(per), Avg(per)

from student

where lower(sname) like '%a%'

group by branch;

⑤ WAD department wise, no of employees, minimum salary,  
 maximum salary only the employees should be earning  
 some salary.  
 display  
 & condition

Select deptno, count(\*), min(sal), max(sal)

from emp

where sal is not null

group by deptno;

display

- ⑥ WQD designation wise recent hiredate, oldest hiredate only the employee should not be working for a department number 40.
- condition

Select job, max(hiredate), min(hiredate)

from emp

where deptno != 40

group by job;

display

- ⑦ WQD salary wise, number of employees, total salary only the employee should be earning four digit salary and display the output in descending order total salary.
- cond(1)      cond(2)

Select sal, count(\*), sum(sal)

from emp

where sal like '----'

group by sal

order by sum(sal) desc;

display

- ⑧ WQD manager number wise minimum annual salary, maximum annual salary only the designation contains of 'man' as a sub-string.
- cond

Select mgr, min(sal \* 12), max(sal \* 12)

from emp

where lower(job) like "%man%"

group by mgr;

display

- ⑨ WQD department wise total salary, Average salary only the manager number should be in a range of 7600 to 7900. and display the output on total salary in ascending order.
- cond(1)      cond(2)

# Assignment - 9

Page No.:

120

Date:

XOOUVX

Select deptno, sum(sal), Avg(sal)

from emp

where mgr between 7600 and 7900

group by deptno

order by sum(sal); (or) order by sum(sal) asc;

display

- (10) Who Designation wise / total annual salary, total annual Commission only employee should be earning salary more than 500.

↳ cond

Select job, sum(sal \* 12), sum(comm \* 12)

from emp

where sal > 500

group by job;

display

- (11) Who commission wise, no. of employees, recent hire date only the employee name should consists of 'a' or 'e' character.

↳ condition

Select comm, count(\*), max(hiredate)

from emp

where (lower(ename)) like '%a%' or lower(ename) like '%e%';

group by comm;

display

- (12) Who manager number wise, minimum commission, maximum commission only the employee should be earning some commission.

Select mgr, min(comm), max(comm)

from emp

where comm is not null

group by mgr;

Note:-

We cannot write char-name in group by clause.

display

Page No.: 121

youna

Date:

- (13) WQD annual salary wise, number of employees only employee number should be hired in a year 80 to 87.
- ↳ cond

Select (Sal\*12), Count (\*)  
from emp

where hiredate between '01-JAN-80' and '31-DEC-87'  
group by (Sal\*12);

Note:- We cannot write char-name in group by clause display.

- (14) WQD new salary with 1000Rs increment/wise average annual salary, Average annual Commission only employee number should be in a range of 7000 to 7900.
- ↳ Cond

Select (Sal+1000), Avg(Sal\*12), Avg(Comm\*12)  
from emp

where empno between 7000 and 7900  
group by (Sal+1000);

display

- (15) Wan department wise recent hiredate / after 6 months, oldest hiredate before 6 months only the employee should not hired in a month of february display the output in ascending based on deptno.
- ↳ cond①  
↳ cond②

Select deptno, max(Hiredate + 182), min(Hiredate - 182)  
from emp

where hiredate not like '%FEB%'

group by deptno

order by deptno; (or) order by deptno asc;

## rules of group by clause :-

\* whatever the C.n which is present in select clause same C.n should be present in group by clause.

example:-

```
Select (c1), MRF  
from table-name  
where <cond's>  
group by (c1);
```

\* whatever the C.n which is present in group by clause it may be or may not be present in select clause.

example:-

```
Select ( ) MRF  
from table-name  
where <cond's>  
group by (c1);
```

(or)

```
Select c1, ( ), MRF  
from table-name  
where <cond's>  
group by c1, (c2);
```

\* we cannot write MRF as a condition in where clause.

example:-

```
Select c1, MRF  
from table-name  
where MRF;
```

\* whenever we are writing both where clause and group by clause in a query, first where clause will be executed, based on condition it will retain some records, then retained records will given to

group by clause. and it will create group.

- \* Any clause will be execute as a "record by record / row by row" in a query.  
but after group by clause/statement present in a query then it will execute group by group.

~~04/18/21~~

### Sub-grouping :-

- \* whenever we are writing multiple c.n's / references in a group by clause, group by clause will create sub-group inside main group is called as sub-grouping.  
(or)

A group created inside another group is called as sub-grouping.

- \* whenever we are writing multiple references (c.n's / expr) in group by clause, the first column will create as main group and the next column will create sub-group inside main group.

- \* Group by clause will pass both main group and sub-group as an argument to select clause.

Student			
Sname	branch	per	
Dinga	CS	90	In QD branch wise under each
Dingi	EC	80	percentage no. of students.
penga	ME	40	
Pengi	CS	85	
Manga	civil	35	
Mangi	CS	90	

Select branch, per, Count (\*) ---> ③

from student ---> ①

group by branch, per; ---> ②

↓      ↓  
reference's

Main group      Sub group

Here "branch" is main group and "per" is sub-group.

CS	EC	ME	CIVIL
Dingi CS 90	Dingi EC 80	Penga ME 40	Manga Civil 35
Penga CS 85			
Mangi CS 90			

90	85	80	40	35
Dingi CS 90	Penga CS 85	Dingi EC 80	Penga ME 40	Manga Civil 35
Mangi CS 90				

Sub-group

O/P :-

branch    per    Count (\*)

CS      90      2

CS      85      1

EC      80      1

ME      40      1

CIVIL    35      1

display

- (2) WAP department wise / under each designation no of employees, total salary only the employee should be earning salary in a range up to 500 to 5000.

cond

Select deptno, job, count(\*), sum(sal)  
from emp

where sal between 500 and 5000

group by deptno, job;

Here deptno is main group, and job is sub-group.

display

- (3) WAP job wise, under each manager number under each department number, recent hiredate, old hiredate only the employee's should be reporting to some manager number.

by condition

Select job, mgr, deptno, max(hiredate), min(hiredate)

from emp

where mgr is not null

group by job, mgr, deptno;

to 'job'

Here 'job' is a main group of 'mgr' and 'mgr' is sub-group likewise 'mgr' is a main group for 'deptno' and 'deptno' is sub-group for 'mgr'.

## Having clause :-

- \* Having clause is used to "filter the record grouped records". because it should be present after the "group by clause" in a query.
- \* whenever we want to write MRF (multi row function) as a condition then we should go having clause.
- \* Having clause will execute "group by group".
- \* Having clause is a "dependent clause", we can't write a having clause if there is no group by clause.

## Syntax:-

```

Select group-by-expression --> 1 c.n / expr / MRF
from table-name --> 1
where <cond's> --> 2 c.n / expr
group by reference's --> 3
having <cond's> --> 4 MRF / c.n / Expr
order by group-by-expression [ASC] / DESC --> 5

```

	Student		display no branch wise / no of students only atleast 2 students should be stud ying in each branch. ↳ Condition
Sname	branch	Per	Select branch, count(*) --> 4
Dinga	CS	90	from student --> 1
Dingi	EC	80	group-by branch --> 2
Penga	ME	40	having count(*) >= 2 ; --> 3
Pengi	CS	85	
Manga	Civil	35	
Mangi	ME	65	

## Note:-

At least  $\rightarrow >$

At most  $\rightarrow \leq$

CS

Dinga	CS	90
Pengi	CS	85

Ec

Dingi	EC	80
-------	----	----

ME

Penga	ME	40
Mangi	ME	65

CIVIL

Manga	CIVIL	35
-------	-------	----

 $2 >= 2 \rightarrow T$  $1 >= 2 \rightarrow F$  $2 >= 2 \rightarrow F$  $1 >= 2 \rightarrow F$ 

If it is TRUE it will retain the group else if it is FALSE it will reject the group.

O/P:-	<u>branch</u>	<u>count(*)</u>
	CS	2
	ME	2

5/08/21

### Rules of having clause :-

- \* whenever we want to write can as a condition in having clause same can should be present in group by clause.

Ex:-

Select c1, MRF

from table 1

group by c1

having c1 and MRF;

- \* whatever the MRF present in having clause it may be are may not be present in select clause.

Ex:

Select c1, MRF

from table 1

where &lt;cond's&gt;

group by c1

having MRF;

- display

- ② WID branch wise no of students minimum percentage,  
max percentage only the percentage range of 35  
to 100 and minimum percentage more than 40.
- ↳ cond(1)                                  ↳ cond(2)

Select branch, Count(\*), min(per), max(per)  
 from emp

where per between 35 and 100.

group by branch

having min(per) > 40;

display

- ③ WID percentage wise no of students , Avg percentage  
only the student name should be exactly 5 char-  
acter and atleast 2 students should be present in  
each percentage.
- ↳ cond(1)                                  ↳ cond(2)

Select per, Count(\*), Avg(per)  
 from emp

where lower(sname) like '\_\_\_\_\_'

group by per

having Count(\*) >= 2;

display

- ④ WID department wise, number of employees total sala  
ry only the employee should be earning some  
salary and the total salary should be in a  
range of 500 to 10000.
- ↳ cond(1)                                  ↳ cond(2)

Select deptno, Count(\*), sum(sal)

from emp

where sal is not null

group by deptno

having sum(sal) between 500 and 10000.

⑤ WON designation wise, recent hire date, oldest hire date  
only the employee should not be working as president  
and the recent hire date should be in a year 81 to 87.

Select Job, max(hiredate), min(hiredate)  
 from emp

where lower(job) != 'president'

group by job

having max(hiredate) between '01-JAN-81' and '31-DEC-87';  
 (or)

Select Job, max(hiredate), min(hiredate)  
 from emp

where

group by job

having (lower(job) != 'president') and (max(hiredate) between  
 '01-JAN-81' and '31-DEC-87');

#### ⑥ Note:-

\* If a Condition with respect to column then it is better to go where clause.

\* When a condition with respect to MRF then it is better to go having clause.

⑦ Can we write column as a condition in having clause?

Ans Yes, but the same condition should be present in group by clause.

⑧ WON manager number wise, total salary, Average salary  
only employee should not be hired in a month January  
and minimum annual salary should be in a range of 5000 to 20000.

↳ cond(2)

Select mgr, sum(sal), Avg(sal)

from emp

where hiredate not like '%JAN%'

group by mgr

having min(sal\*12) between 5000 and 20000;

display

- ⑤ WAD annual salary wise, no of employees only employee should be working for a department number 10, 20, 30 and oldest hiredate should not be in a year 87, and display the output in ascending order based on annual salary.
- cond ③

Select (sal\*12), Count(\*)

from emp

where deptno in (10, 20, 30)

group by (sal\*12)

having min(hiredate) not like '%87'

order by sal\*12; (or) order by (sal\*12) asc;

is optional <sup>(ann-sal)</sup>

### Assignment - 10

display

- ① WAD department wise minimum/annual salary, max annual salary only the employee name should begins with 'a' or 's' character. and the <sup>cond ①</sup> min annual salary should be more than min annual commission.
- cond ②

Select deptno, min(sal\*12), max(sal\*12)

from emp

where lower(ename) like 'a%' or lower(ename) like 's%'

group by deptno

having max(sal\*12) > min(~~sal~~ comm \* 12)

display

- ② WAP Designation wise, recent hiredate, oldest hiredate only the employee should be reporting to some manager number and the <sup>second</sup> recent hiredate should be in a year 81.

Select job, max(hiredate), min(hiredate)  
from emp

where mgr is not null

group by job

having max(hiredate) like '%81';

display

- ③ WAP manager number wise number of employees,  
min comm, max comm only the employee should be  
working as salesman and the min salary should  
be in a range of <sup>cond①</sup> 500 to 5000.

cond①

Select mgr, Count(\*), min(comm), max(comm)  
from emp

where lower(job) = 'salesman'

group by mgr

having min(sal) between 500 and 5000;

display

- ④ WAP department wise no. of employees where at least  
2 clerks should be working in each department no.

Select deptno, count(\*) -> ⑤

from emp -> ① where lower(job) = 'clerk' -> ②

group by deptno -> ③

having count(\*) >= 2; -> ④

display

⑤ WAP department wise / under each designation total salary total Commission only. the designation should be man as a substring and the total salary should be less than 10000.

↳ cond ①

Select deptno, job, sum(sal), sum(comm)  
from emp

where lower(job) like '%man%'

group by deptno, job

-having sum(sal) < 10000;

display

⑥ WAP date of joining wise / number of employees only the employee name should be exactly five character and the total annual salary should be exactly 5 digit.

↳ cond ②

Select hiredate, count(\*)

from emp

where lower(ename) like ' - - - - '

group by hiredate

having sum(sal\*12) like ' - - - - ';

Difference b/w where clause and having clause:-

where clause

Having clause

\* It will filter un-grouped records. \* It will filter grouped records.

\* It will execute group by row by record by record.

\* It will execute group by group.

\* We cannot write MRF as a condition.

\* We can write MRF as a condition.

\* We can write any condition in where clause

\* We can write condition in having clause. only the condition which is present in group by clause.

- |   |  |
|---|--|
| * It will execute before group by clause  | * It will execute after group by clause.   |
| * we can write where clause if there is no group by clause present in query<br>(Independent clause) | * We cannot write having clause if there is <del>no</del> group by clause.<br>(dependent clause) |

Escape character: - (not required in SQL knowledge purpose only)

- \* The process of Converting Special character into ordinary character. Then we go ESCAPE character.
- \* Escape character can be any character which is present in Keyboard (except % - )
- \* One escape char will convert one special character into ordinary character.

How to write we will see some example's.

Q) Won emp details only the ename consists of '-'.

```
Select *
from emp
where ename like '%!-%'
escape '!';
(or)
```

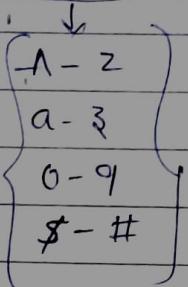
```
Select *
from emp
where ename like '%a-%'
escape 'a';
(or)
```

```
where ename like '%#-%'
escape '#';
(or)
```

```
where ename like '%q-%'
escape 'q';
(or)
```

Note:-

We want to write escape character before (%) or -) char



Then it will show the result/output which name has the special char (%) or -) present in the 'String/char' (or) C.n.

(2) W/o emp details only the ename consists of '%'.  
Select \* from emp

where ename like '% ! % %'

escape '!' ;

(3) W/o emp details only the ename consists of 2 consecutive '--'.

Select \* from emp

where ename like '% ! - ! - %'

escape '!' ;

(4) W/o emp details only the ename consists of 2 '%'.  
Select \*

from emp

where ename '% ! % % ! % %'

escape '!' ;

(5) W/o emp details only the ename last 2 char should be '\_-'.

Select \*

from emp

where ename like '% ! \_ - '

escape '!' ;

66/8/21

## Sub-Query:-

Query :-

- \* Query is a set of stmt's followed multiple cond's and which is used to interact with database is called as query.
- \* Query means which begins with select statement and followed by multiple conditions and terminating by semicolon (;) is called a query.

Example:-

```
Select * / c.n / enpr distinct alias-name
from table-name
where <cond's>;
```

## Sub-Query definition:-

A query inside another query is called as sub-query.

(or)

A query written inside another query is called as sub-query.

Sub-query can be written in two clause's / stmt's :-

① FROM clause

② WHERE clause

Syntax of sub-query in where clause:-

Syntax of Sub-query in where clause :-

Sel

→ Outer Query

Select \* / c.n

from table\_name

where <cond's> ( select \* / c.n

from table\_name

where <Cond's> ( select \* / c.n /

from table\_name

where <cond's>..... ) );

Outer Query / Inner Query

Inner query

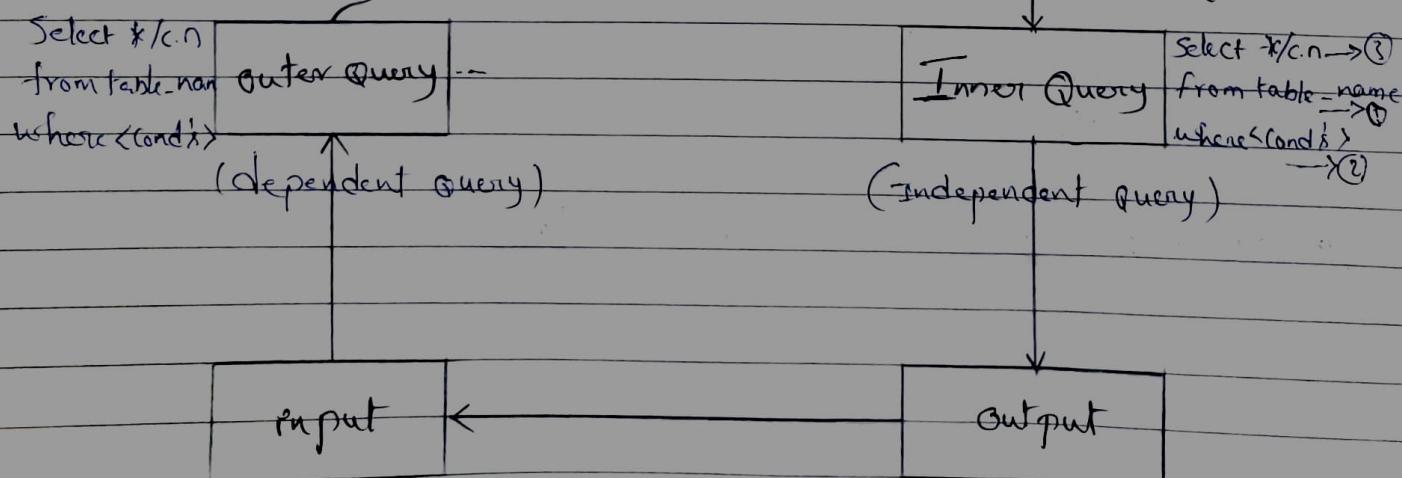
We Can write 255 INNER QUERY

Note :-

- \* A query which is present inner most part is called as Inner Query.
- \* A query which is present Outer most part is called as Outer Query.
- \* Generally sub-query is a Combination of "Inner query" and "Outer Query".
- \* Sub-query is also known as "nested sub-query" because we can write upto 255 inner query's.

Final output

Compiler gives control



Working Mechanism of sub-query.

- \* Compiler will give control to inner query. inner query will execute based on the condition and it will prepare output.
- \* The output of inner query will pass as an input to outer query.
- \* Outer query will execute based on the condition and input taken from inner query it will prepare a final output.
- \* Outer query is a dependent query and inner query is an independent query.

### Types of sub-query :-

- \* Based on the output of inner query sub-query is classified into two types.
  - (1) single row sub-query.
  - (2) Multi row sub-query.
- (1) single row sub-query :-
- \* If inner query generates exactly single record as output then it is called as single row sub-query.
- \* If inner query generates exactly single record as output then we can compare by using relational operators ( $<$ ,  $>$ ,  $<=$ ,  $>=$ ,  $=$ ,  $\neq$ ) and IN operator.

### (2) Multi-row sub-query :-

- If inner query generates more than one record as output then it is called multi-row sub-query.
- \* If inner query generates more than one record as output then we cannot compare by using relational operators so we should go (IN, NOT IN, ANY, ALL)

## Cases / when to go sub-query :-

Case 1:- To determine unknown Values.

Case 2:- Linking multiple tables by utilizing Common C.n b/w tables.

Case 3:- mapping Values present in same table but in different rows (nth max and nth min value)

Case 1:- To determine unknown values.

		emp		
	ename	job	deptno	.
Akash	salesman	20		
Smith	clerk	30		✓
Blake	salesman	10		
Miller	Analyst	30		
Turner	Clerk	30		
martin	salesman	10		

display (Q.Q)

① WAP emp details only the deptno should be same as  
 Smith deptno  
 ↗  
 ↘  
 unknown (In.Q)

Select \* ---> ③

from emp ---> ① "30" . "Inner query"

where deptno IN (select deptno ---> ②

deptno IN 30

20 = 30 ---> F

30 = 30 ---> T

10 = 30 ---> F

30 = 30 ---> T

30 = 30 ---> T

10 = 30 ---> F

from emp ---> ①  
 where lower(ename) = 'smith' ); ---> ②

Akash = Smith ---> F

Smith = Smith ---> T ✓ 30

Blake = Smith ---> F

Miller = Smith ---> F

Turner = Smith ---> F

Martin = Smith ---> F

Q/P:-

ename	Job	deptno
Smith	Clerk	30
Miller	Analyst	30
Turner	Clerk	30

~~#18121~~  
Note:-

- \* unknown value should be present in inner query.
- \* what they are asking you to display in outer query.
- \* It is always better to write INL rather than equal(=)

② W<sub>O</sub>D employee/ details only the salary should be more than miller salary.

```
Select *
from emp
where sal > (select sal
               from emp
               where lower(ename) = 'miller');
display (o. Q)
```

③ W<sub>O</sub>D employee/ details only the designation should be same as (blake designation)  $\rightarrow$  unknown (I. Q)

```
cond
Select *
from emp
where job in (select job
               from emp
               where lower(ename) = 'blake');
display (o. Q)
```

④ W<sub>O</sub>D employee /name, department number, date of joining only the date of joining should be same as 'Ford' date of joining  $\rightarrow$  cond  
unknown (I. Q).

Select ename, deptno, hiredate  
from emp

where hiredate in (Select hiredate  
from emp)

where lower(ename) = 'ford');

display (0.0)

⑤ WQD employee name / employee number, salary, commission  
only the employee salary should be more  
than 'Turner' commission.

Select ename, empno, sal, comm

from emp

where sal > (Select comm

from emp

where lower(<sup>ename</sup>job) = 'turner');

Note:-

\* what ever the C.N present in where Outer query  
where clause it should be "logically related"  
with inner query select C.N

where sal > (Select comm ✓

where empno = (Select mgr ✓

where Sal = (Select deptno X

where ename = (Select job X

display (0.0)

⑥ WQD employee details only the manager number sh  
ould be same as 'Smith' manager number.

Select \*

from emp

where mgr in (Select mgr

from emp

where lower(job) = 'smith');

(I.0)

display (0..Q)

- ⑦ W<sup>Q</sup>D employee details only the employee should be hired before Scott hiredate.

Select \*

from emp

where hiredate &lt; (select hiredate

from emp)

where lower(job) = 'Scott');

- ⑧ W<sup>Q</sup>D employee name, annual salary, annual commission only employee should be earning salary. and the annual salary should be more than 'allen' Annual comm.

Select ename, Sal\*12, ann-sal, comm\*12 ann-comm

from emp

where (sal is not null) and (sal\*12) &gt; (select comm\*12

from emp

where lower(ename)

= 'allen');

display (0..Q)      unknown (1..Q)

- ⑨ W<sup>Q</sup>D employee details only (the manager number should be same as (select employee number) and the employee name should be exactly '5' char.

Select \*

from emp

where ename like '\_\_\_\_' and mgr in (select empno

from emp

where lower(ename) = 'king');

(or)

Select \*

from emp

where mgr in (select empno

from emp)

where lower(ename) = 'king' and ename like 'k\_\_\_\_\_');

disby (0.0)

- (10) ~~W<sub>Q</sub>D~~ employee details only the employee should be hired in a year 81. and they should be hired after 'martin' hiredate.

Select \*

from emp

hiredate &gt;

where hiredate like '%81' and (select hiredate

from emp

where lower(ename) = 'martin');

- (11) ~~W<sub>Q</sub>D~~ employee details only the manager number should be in a range of 7000 to 7900 and their designation should be same as James, Blake designation.

Select \*

from emp

where mgr between 7000 and 7900 and job in

( select job

from emp

where lower(ename) in ('James', 'Blake'));

- (12) ~~W<sub>Q</sub>D~~ employee details only the employee number should be same as Ward mgr. and employee should not be hired on James hiredate.

Select \* from emp

where empno in

( select mgr

from emp

where lower(ename) = 'ward') and

(select hiredate

from emp

where lower(ename) = 'James');

⑬ W<sub>Q</sub>D employee details only the department number should be same as ward, miller department number and their manager number should be less than 'Ford' employee number, and display the output in descending order based on manager number.

Select \*

from emp

where deptno in (select deptno

from emp

where lower(ename) in ('ward', 'miller')

and mgr < (select empno

from emp

where lower(ename) = 'ford'

order by mgr desc;

⑭ W<sub>Q</sub>D employee details only the job should not be same as smith job and salary should be less than James salary.

Select \*

from emp <sup>not</sup>

where job \* in (select job

from emp

where lower(ename) = 'smith')

and ~~select~~ sal < (select sal

from emp

where lower(ename) = 'james');

q108121

display

⑮ W<sub>Q</sub>D ename, date of joining, comm only the employee number should not be in a range of 7600 to 7900 and their manager should be same as miller, James manager number

<sup>cond①</sup>

→ Inner Query  
Unknown

Select ename, hiredate, Comm  
from emp

where (empno not between 7600 and 7900)

and mgr in (select mgr  
from emp)

where lower(ename) in ('miller', 'gomes');

display

- ⑯ W<sub>QD</sub> employee details only the employee number should be same as allen, word manager number. and the designation should not be same as ('Smith designation')

↳ cond①

↳ unknown

Select \*

from emp

where empno in (select mgr

from emp

where lower(ename) in ('allen', 'blard')) and

job<sup>not</sup> in (select job

from emp

where lower(ename) = 'smith');

- ⑰ W<sub>QD</sub> employee details only the salary should be more than highest commission or maximum commission.

Select \*

from emp

where sal > (select max(comm)

from emp);

display

- ⑱ W<sub>QD</sub> employee details of max sal / 1st max sal / highest salary.

↳ condition

Select \*

from emp

where sal = (select max(sal)

from emp);

Note:-

when we are writing MRF in sub-query it is better to write relational operator (=) equal to instead of IN (special operator)

Assignment

- ① WQN employee name, employee number, department number, annual salary the department number should not be 40 and the annual salary should be less than miller annual salary.

Select ename, empno, deptno, sal\*12 as ann\_sal  
from emp

where deptno != 40 and sal\*12 < (Select sal\*12 as ann\_sal

from emp

where lower(ename) = 'miller' );

2

display

- ② WQN employee details only the employee should be hired in a year 80 to 87. and their designation should be same as king, blake designation and display the output in ascending order based on salary.

Select \*

from emp

where hiredate between '01-JAN-80' and '31-DEC-87'

and job in (Select job

from emp

where lower(ename) in ('king', 'blake'). )

order by sal ; (or)

order by sal asc ;

display

cond①

Date:

- ③ W<sup>Q</sup>D employee details only the employee should be hired before Smith hiredate and their commission should be more than blake salary.

→ cond②

Select \*

from emp

where hiredate &lt; (select hiredate

from emp

where lower(ename) = 'smith' )

and comm &gt; ( select sal

from emp

where lower(comm) = 'blake');

- ④ W<sup>Q</sup>D employee name, designation, salary, commission only the salary should be less than Average commission.

Select ename, job, sal, comm

from emp

where sal &lt; ( select avg(comm)

from emp );

- ⑤ W<sup>Q</sup>D employee details of 1st oldest hiredate.

Select \*

from emp

where hiredate = (select min(hiredate)

from emp );

Advance knowledge purpose of syllabus:-

- ⑥ W<sup>Q</sup>D number of employees, minimum salary, maximum salary only the date of joining should not be same as turnover date of joining.

Select count(\*), min(sal), max(sal)

from emp

where job != (select job  
from emp)

display where lower(ename) = 'turner');

- ⑤ WAD recent hiredate, oldest hiredate only the employee number should be more than Smith manager number.  
 ↓ condition      ↓ unknown (I.O)

Select max(hiredate), min(hiredate)  
from emp

where empno > (select mgr  
from emp  
where lower(ename) = 'smith');

display

- ⑥ WAD department wise, number of employees, maximum annual salary, minimum annual salary only employee should be hired before recent hiredate  
 ↓ cond

Select deptno, count(\*), max(sal \* 12), min(sal \* 12)  
from emp

where hiredate < (select max(hiredate)  
from emp)

group by deptno;

- ⑦ WAD designation wise total salary, Average salary, total commission only the employee name should be exactly 5 char. and the annual sal should be less than minimum annual commission.

Select job, sum(sal), Avg(sal), sum(comm)  
from emp

where (lower(ename) like '----') and sal \* 12 <  
(select min(comm \* 12)  
from emp).

group by job;

(10) w.QD manager number wise, number of employees, total salary, total annual salary only the employee should be reporting to some manager number and the salary should be less than king salary. and the total salary should be in a range of 500 to 10000.

Select mgr, count(\*), sum(sal), sum(sal \* 12)

from emp

where (mgr is not null) and sal < (select sal

from emp

where lower(ename) = 'king'

group by mgr

having sum(sal) between 500 and 10000;

(11) w.QD department wise / no of employees only the employee should be hired after oldest hiredate and at least two employee's should be working in each department and display the output in descending order based on department number

Select deptno, Count(\*)

from emp

where hiredate > (select min(hiredate)

from emp)

group by deptno

having Count(\*) >= 2

IMP order by deptno desc;

(12) W.QD employee details only the department number should be same as the employee's those are working as 'sales -man' department number.

Case 2 :- Linking multiple tables by utilizing Common Column between the tables.

emp			dept		
empno	ename	deptno	deptno	dname	loc
1	Dia	10	10	Accounting	New York
2	Dolly	30			
3	Dachu	20	20	Research	Dallas
4	Dingi	10	30	Sales	Chicago
5	Shashi	20	40	Operations	Boston

① WQD emp details only the employee should be working in 'accounting' department.

Here display(emp) → deptno ← condition(dept)  
 ↓  
 Common C.n

Select \* → "Outer Query"

from emp ---> ① "10" → "Inner Query"

where deptno in (select deptno ---> ③)

→ ② from dept ---> ①

deptno = 10 where lower(dname) = 'accounting') ; ---> ②

10 = 10 ---> T

30 = 10 ---> F

20 = 10 ---> F

10 = 10 ---> T

20 = 10 ---> F

accounting = accounting ---> T

research = accounting ---> F

sales = accounting ---> F

operations = accounting ---> F

O/P	empno	ename	deptno
	1	Dia	10
	4	Dingi	10

display (emp)

- ② WQD employee details only the employee should be located in 'chicago' location.  
 ↳ cond (dept)

Select \*

from emp

where deptno in (Select deptno

from dept

where lower(loc) = 'chicago' );

display (dept)

- ③ WQD department details only employee should be earning salary 1000 to 5000.  
 ↳ cond (emp)

Select \*

from dept

where deptno in (Select deptno

from emp

where sal between 1000 and 5000);

display (dept)

- ④ WQD dname,loc, only the employee should be hired in a year 81  
 ↳ condition (emp)

Select dname, loc

from dept

where deptno in (select deptno

from emp

where hiredate like '%81');

display (emp)

- ⑤ WQD employee name, designation, salary only the location should consist of 'new' as a sub-string.  
 ↳ cond (dept)

Select ename, job, sal

from emp

where deptno in (select deptno from dept

where lower(loc) like '%new%');

10/8/21

Page No.: 151

youva

Date:

display(emp)

- ⑥ W<sup>O</sup>D employee details only the employee name should consist of 'a' characters and they should not be working for a department <sup>cond(1)</sup> name "operations".  
↳ cond(2) (dept)

Select \*

from emp

where lower(ename) like '%a%' and deptno in

(select deptno

from dept

where lower(dname) != 'operations');

display(emp)

- ⑦ W<sup>O</sup>D ename, designation, salary only the employee should not be working as clerk and they should be located on 'Dallas', 'Chicago' location.  
↳ cond(1) (emp)  
↳ cond(2) (dept)

Select ename, job, sal

from emp

where lower(job) != 'clerk' and deptno in (select deptno

from dept

where lower(loc) in

('Chicago', 'Dallas'));

display(dept)

- ⑧ W<sup>O</sup>D department name, location only the department name should consist of two consecutive 'c' characters and they should be hired in a year '81'.  
↳ cond(1) (dept)  
↳ cond(2) (emp)

Select dname, loc

from dept

where lower(dname) like '%cc%' and deptno in

(select deptno

from emp

where hiredate like '%81');

(9) W/OD department details only the employees should not be located in Chicago location and their manager number should be 7698, and 7839.

Select \*

from dept

where lower(loc) != 'chicago' and deptno in

( select deptno  
from emp  
where mgr in (7698, 7839) ) ;

(10) W/OD employee name, employee number, date of joining only the employee number should be in a range of 7000 to 7900 and they should not be hired in a month Feb and the location should consist of two 'L' character

Select ename, empno, job

from emp

where (empno between 7000 and 7900) and

( hiredate not like '% FEB %') and deptno in

( select deptno

from dept

where lower(loc) like '% L % L %' ) ;

*(Advance Question  
Knowledge based)*

display(emp) — upto Syllabus — Over —

(11) W/OD employee details only the designation should be same as Miller designation and they should not be located in 'boston' location.  
cond(1)(emp)  
cond(2)(dept)

Select \*

from emp

where job in ( select job

from emp

where lower(ename) = 'miller') and deptno in

(Select deptno  
from dept  
where lower(loc) != 'boston');

- (12) WAP ename, department number, annual salary only the salary should be less than James salary and the department name should be exactly five char.  
 Select ename, deptno, sal\*12 as ann\_sal  
 from emp  
where sal < (select sal  
from emp  
where lower(ename) = 'james') and deptno in  
(select deptno  
from dept  
where dname like '---');

- (13) WAP employee details only the manager number should be same as adams manager number and they should be working for a department research, sales.

Select \*  
 from emp  
 where mgr in (select mg1  
 from emp  
 where lower(ename) = 'adams') and  
 deptno in (select deptno  
 from dept  
 where lower(dname) in ('research', 'sales'));