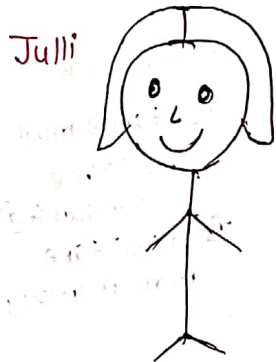


SQL (Structured Query Language)

SQL: SQL stands for Structured Query language.

Data: Data is raw fact which describes the Properties of an object.

- properties is also known as attributes
- Object is also known as Entity.



Object/
Entity

NAME: JULLI

AGE: 21

HEIGHT: 5.1

WEIGHT: 48

GENDER: FEMALE



Properties/Attributes

ADD : JSPIDER

PHONENO: 100

AADHNO: 0000 0000 0000

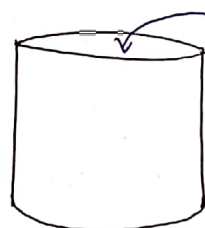
EMAIL: JULLI.ROMEO@gmail.com

DOB: 01-FEB-2003

Data = Raw Fact -

DATABASE:

① Database is a place or a medium in which we store the data in a systematic and organized manner.



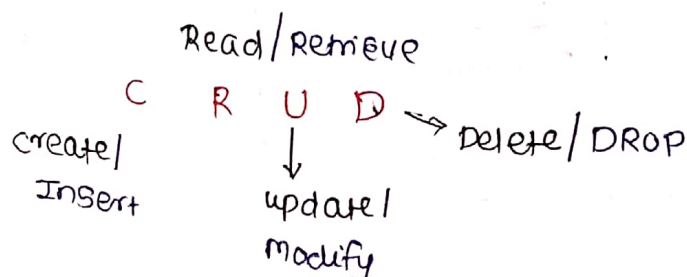
Database

Data

+Systematic

+Organized

In database we perform one operation called CRUD operation



* DBMS

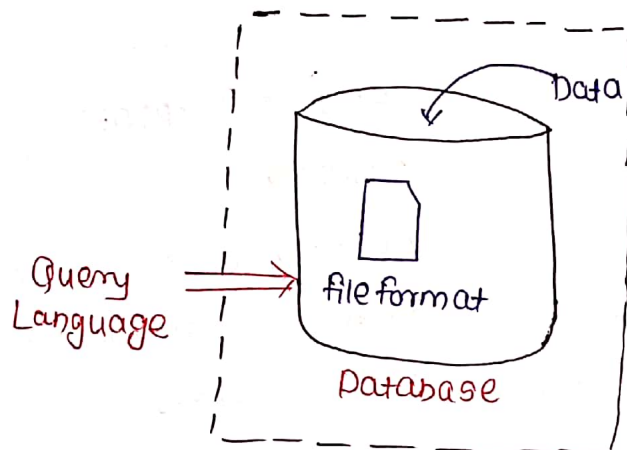
DBMS stands for Database Management System

DBMS is a sw which is used to maintain & manage the database.

Security and authorization are the 2 main features provided by DBMS.

In DBMS sw we store the data in the form of file format.

We use Query Language to communicate with DBMS software.



+ security

+ Authorization

* Drawback

① It takes more Time (Time consuming)

② It will take more memory

* RDBMS

RDBMS stands for relational Database Management System.

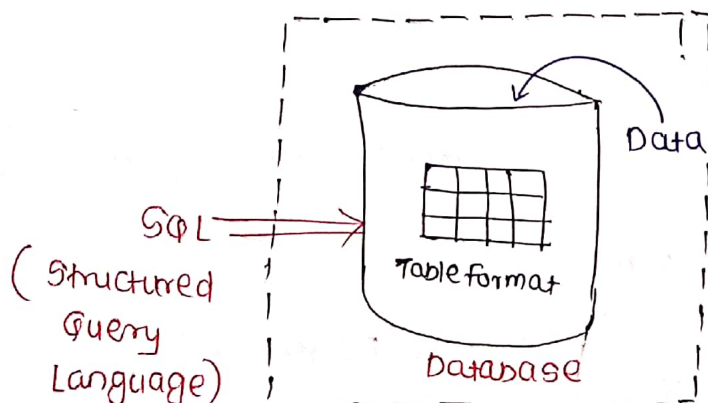
RDBMS is a sw ~~is~~ is also a type of DBMS software

RDBMS sw is used to maintain and manage the database.

Security and authorization are the two main features provided by RDBMS sw

In RDBMS sw we store the data in the form of Table format.

~~We~~ To Communicate RDBMS sw we used, structured Query Lang (SQL)



+ security

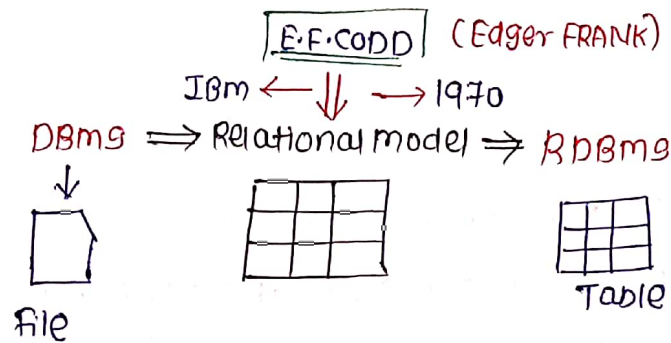
+ Authorization

* RELATIONAL MODEL

- ① Any DBMS slw which follows Relational model it will become RDBMS slw.
- ② In Relational model we store the data in the form of Table format.

[NOTE]:-

⊕ Relational model was proposed by one of the data Scientist called "E.F.Codd"



* TABLE:-

① The Combination of Rows and columns is known as Table.

② Table consist of Three Types.

- ⊕ Row
- ⊕ column
- ⊕ cell

① Row:- ⊕ Horizontal portion of Table is known as Row.

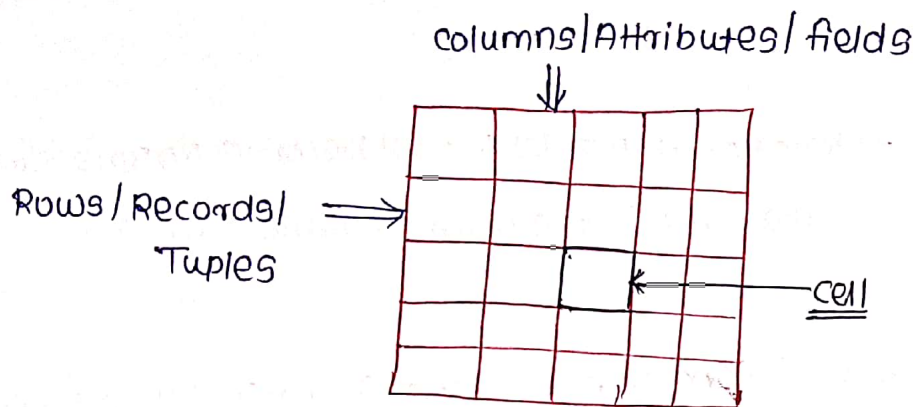
⊕ Row is also known as Records or Tuples.

② Column:- ⊕ Vertical portion of the table is known as column.

⊕ Column is also known as Attributes or fields.

③ cell:- ⊕ Smallest unit of the Table is known as cell. or

⊕ Intersection of Row and column is known as cell.



*E.F. CODD RULES:-

- ① The data enter into a cell must be a single value.
- ② According to E.F. Codd Rules we can store the data in multiple tables if needed we can establish the connection between the tables with the help of key attributes.
- ③ In RDBMS we store data in the form of tables, it includes metadata.
 - Ⓐ details about the data is known as metadata.
 - Ⓑ metadata's are autogenerated.
- ④ we can validate the data in 2 steps:
 - Ⓐ By Assigning datatypes
 - Ⓑ By Assigning constraints.

NOTE:- datatypes are mandatory whereas constraints are optional.

Student

IMAGE	Roll No	NAME	PHONE	COURSE1	AGE	COURSE2
	1	SHATYAM	9169056949	JAVA SQL	22	SQL
	2	PARTH	9893527575	SQL	21	
	3	PRIYA	100	JAVA	21	
	4	DIVYA	108	SQL	21	
	5	RAHUL	9175485991	SQL	23	
	6	AAYUSH	991	SQL	21	



NAME: IMG1234

TIME 8:25 AM

SIZE 2MB

DATE 5 FEB 2024

metadata

Data

CN

* DATATYPE :-

① Datatype is used to specify the type of data that will be store in a particular memory location.

* Datatypes in SQL:-

- ① char
- ② varchar/varchar2
- ③ DATE
- ④ NUMBER
- ⑤ LARGE OBJECT
 - * CHARACTER LARGE OBJECT
 - * BINARY LARGE OBJECT

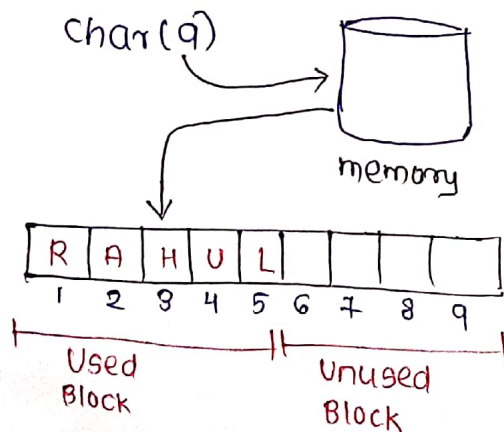
① CHAR :-

- ① CHAR datatype will accept A-Z, a-z, Numbers (0-9) and special characters as well as alphanumeric.
- ② characters must always be enclosed within single quotes ('')
- ③ syntax:- CHAR(size)
- ④ whenever we used char datatype we must mention size.

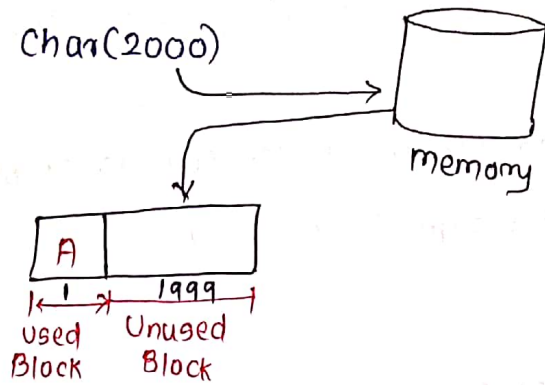
* Size :-

- ① It is used to specify the number of characters it can store.
- ② The maximum number of characters it can store is 2000 characters.
- ③ char datatype follows fixed length memory allocation.

Example:-



Example 2:-



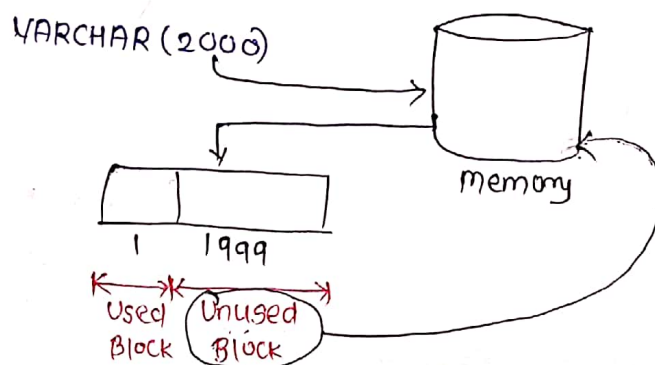
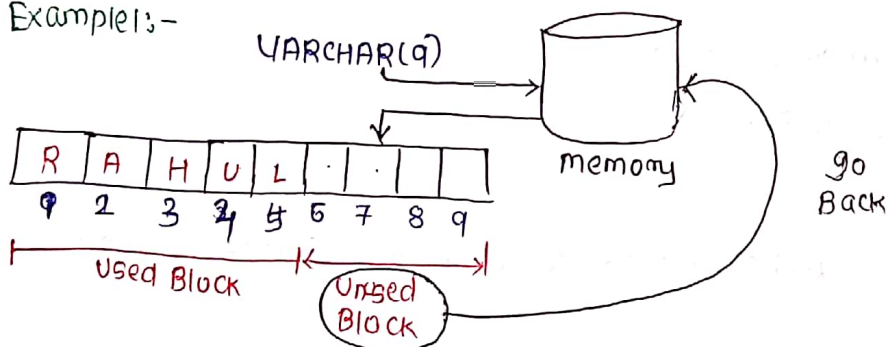
② VARCHAR DATATYPE:-

- ① VARCHAR Datatype will accept Capital (A-Z), Small (a-z), Numbers from 0-9 and special characters as well as Alphanumerics.
- ② characters must always be enclosed with ()
- ③ Syntax:- VARCHAR(size)
- ④ Whenever we used VARCHAR Datatype we must mention the size.

* Size:-

- ① It is used to specify the number of characters it can store.
- ② The maximum number of characters it can store is 2000 characters.
- ③ VARCHAR Datatype follows Variable length memory allocation.

Example 1:-



* VARCHAR2 DATATYPE:-

- ① VARCHAR2 datatype is an updated version of Varchar datatype.
- ② The maximum size of VARCHAR2 Datatype is 4000 characters.

* Syntax:- VARCHAR2(Size)

Example:- VARCHAR(20) → VARCHAR2(20)

[NOTE]:-

Whenever we are using VARCHAR Datatype it will automatically converted into VARCHAR2 Datatype.

* DATE

- ① Date Datatype is used to store Dates in a particular format.
- ② It used ORACLE Specified format.
- ③ Syntax : DATE

* Example 'DD-MON-YY' ② 'DD-MON-YY'
'07-FEB-2024' '14-FEB-24'

* NUMBER :

- ① It is used to store the numeric values.

* Syntax:- NUMBER(PRECISION, SCALE)

• Precision :

- ① It is used to determine the Number of digits used to store integer values

• Scale

- It is used to determine the number of digits used to store decimal values with Precision

- Scale is not mandatory

- The default value of scale is 0.

Example:-

Example:- NUMBER (5, 2) — ±99999
 $P > S$

② NUMBER (9, 1) — 999
 $P > S$

③ NUMBER (4, 4) — .7777
 $P = S$

④ NUMBER (2, 5) — .00099
 $P < S$

⑤ NUMBER (5) — 99999
 P

* LARGE OBJECT :-

① CHARACTER LARGE OBJECT (CLOB) :-

① It is used to store the characters upto 4GB of size.

* Binary LARGE OBJECT (BLOB):

- It is used to store the Binary values of Images, documents, etc upto 4GB of size)

CONSTRAINTS

① Constraints are the rule given to the column for validation

* Types of Constraints :-

- ① Unique
- ② Not null
- ③ check
- ④ Primary key
- ⑤ Foreign key

① UNIQUE :-

① UNIQUE is a constraint which cannot accept duplicated or repeated values.

② NOT NULL :-

- NOT NULL is a constraint which cannot accept null values.

NULL :- It means Empty.

③ CHECK :-

- CHECK is a constraint which is extra validation assign to the column with the condition. If the condition is satisfied it will return True

④ PRIMARY KEY :-

- PRIMARY KEY is a constraint which is uniquely identify the records from the Table.

* characteristics of Primary key :-

- ① primary key cannot accept duplicated or repeated values.
- ② primary key cannot accept null values.
- ③ primary key is always a combination of Unique + NOT NULL constraints
- ④ A Table can accept any one column as a primary key
- ⑤ Primary key is not mandatory but it is Highly recommended

⑤ FOREIGN KEY

① FOREIGN KEY is a constraint which is used to establish the connection between the Tables.

* Characteristics of Foreign Key:-

- ① FOREIGN KEY can accept duplicated or repeated values.
- ② Foreign key can accept null values.
- ③ FOREIGN KEY is not a combination of UNIQUE + NOT NULL CONSTRAINT.
- ④ A Table can accept more than one column as a Foreign Key.
- ⑤ If the column wants to become a foreign key ^{but} it is mandatory to be Primary key of its own Table.
- ⑥ Foreign key is present in child table but it belongs to Parent Table.
- ⑦ Foreign key is also known as Referential Integrity Constraints.

Employee:-

<u>PK</u> { UN NN	NN		UN NN	F.K
EMPID	ENAME	SAL	PHNO	DEPTNO
VARCHAR (10)	VARCHAR (20)	NUMBER (7)	NUMBER (10)	VARCHAR (5)
1	Aditya		85910341	20
2	Sumith	01	773018512	10
3	Sruthi	50	901813215	30
4	Lakshman	100	9004831780	40
5	Guresh	10000	86513281	10

↓
child

PK=F.K		
DNO	DNAME	LOC
VARCHAR (5)	VARCHAR (15)	VARCHAR (20)
10	D1	L1
20	D2	L2
30	D3	L3
40	D4	L4

↓
parent

* Overview of SQL Statements *

- ① Data Definition Language (DDL)
- ② Data Manipulation Language (DML)
- ③ Transaction Control Language (TCL)
- ④ Data Control Language (DCL)
- ⑤ Data Query Language (DQL)

* DATA QUERY LANGUAGE (DQL) :-

① This statement is used to retrieve or fetch the data from the Database.

② It has 4 statements :-

→ ① SELECT

→ ② PROJECTION

→ ③ SELECTION

→ ④ JOINS

* SELECT :-

It is a process of retrieve or fetch the data from the table and it will display.

* PROJECTION :-

- It is a process of retrieve or fetch the data from the table by selecting only the columns is known as 'PROJECTION'.

- In PROJECTION all the records present in a particular column or By default selected.

* SELECTION :-

- It is a process of retrieve or fetch the data from table by selecting both columns as well as rows is known as 'SELECTION'.

* JOINS :-

- It is a process of retrieve or fetch the data from the multiple table simultaneously is known as 'JOINS'.

* PROJECTION :-

- ① It is a process of retrieve or fetch the data from the table by selecting only the columns is known as PROJECTION
- ② In PROJECTION all the records present in a particular column or By default selected.

① WAG TO display Names of the Employees.

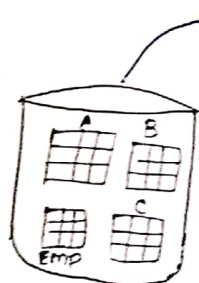
Syntax :- SELECT */ [DISTINCT] COL-NAME/ EXPRESSION [ALIAS]
FROM TABLE-NAME;

* Order of Execution FROM CLAUSE ①
↓
SELECT CLAUSE ②

* [NOTE] :- ① FROM CLAUSE will execute first

- ② In from clause we can write table-name as argument
- ③ The Job of from clause is to go to the database & search for the table and put the table under execution
- ④ After the Execution of from clause select clause will execute.
- ⑤ In SELECT CLAUSE we can write * or column-name or Expression as an argument.
- ⑥ The Job of SELECT CLAUSE is to go to the table which is under execution and display the column which we mention.

Example :- WAG TO Names of the Employees.



Database

EMPNO	ENAME	SAL	DEPTNO
1	A	100	10
2	B	200	30
3	C	300	20
4	D	400	10
5	E	500	30
6	F	600	20
7	G	700	30

OP OF FROM
CLAUSE

Under Execution

SELECT ENAME
FROM EMP;

ENAME
A
B
C
D
E
F
G

OP OF
SELECT CLAUSE

Q) WAQTD Salary of the Employees.

→ SELECT SAL
FROM EMP;

Q) WAQTD details of the employees.

→ SELECT *
FROM EMP;

*[NOTE):-

① SQL is NOT a case-sensitive language.

②

(*) Asteric :- It will display all the columns present in the table.

Semicolon(;) :- To end the statement

*To display the entire Tables present in database:-

→ SELECT *
FROM TAB;

WAQTD NAME & SAL OF the Employees.

→ SELECT ENAME, SAL
FROM EMP;

Q) WAQTD Employee Name, Hiredate, Department NO of the Employees.

→ SELECT NAME, HIREDATE, DEPTNO
FROM EMP;

ASSIGNMENTS

① WAQTD all the details from the Employee.

→ SELECT *
FROM EMP;

② WAQTD names of all the Employees.

→ SELECT ENAME
FROM EMP;

③ WAQTD NAME & SALARY given to all the Employees.

→ SELECT ENAME, SAL
FROM EMP;

④ WAQTD Name and Commission given to all the Employees.

→ SELECT ENAME, COMM
FROM EMP;

⑤ WAQTD Employee ID and Department No of all Employees.

→ SELECT EMPNO, DEPTNO
FROM EMP;

⑥ WAQTD Ename and HireDate of all the Employee

→ SELECT ENAME, HIREDATE
FROM EMP;

⑦ WAQTD Name and designation of all the Employees.

→ SELECT ENAME, JOB
FROM EMP;

⑧ WAQTD Name, Job and Salary given to all Employees.

→ SELECT ENAME, JOB, SAL
FROM EMP;

⑨ WAQTD DNAME Present in Department Table.

→ SELECT DNAME
FROM ~~EMP~~ DEPT;

⑩ WAQTD DName and location present in Department Table.

→ SELECT DNAME, LOC
FROM ~~EMP~~ DEPT;

Q) WAQTD Details of all the Employees.

→ SELECT *
FROM EMP;

Q) WAQTD Department No of the Employees.

→ SELECT DEPTNO
FROM EMP;

* Q) WAQTD Department No of the Employees without repetition.

→ SELECT DISTINCT DEPTNO
FROM EMP;

* DISTINCT CLAUSE :-

* ① Distinct clause is used to remove the duplicated or repeated values from the result table.

② Always we use DISTINCT clause as the first argument inside the SELECT CLAUSE.

Example :- WAQTD DeptNo of the Employees without repetition.

DEPTNAME	DEPTNO
A	10
B	20
C	30
A	40
B	20

SELECT DISTINCT DEPTNO
FROM EMP;

↓

Result →

DEPTNO
10
20
30
40

WAQTD EMPNAME & DEPTNO of the Employees without repetition.

→ SELECT DISTINCT EMPNAME, DEPTNO
FROM EMP;

DEPTNAME DNO

A 10
B 20
C 30
A 40
B 20

⇒

DeptName	DNO
A	10
B	20
C	30
A	40

Imp [NOTE]:

We can use multiple columns as an argument to DISTINCT clause
it will remove the combination of columns in which records are duplicated

Q) WQTD Employee Name, Sal, DeptNo of the Employees without repetition.

→ SELECT DISTINCT ENAME, SAL, DEPTNO
FROM EMP;

Q) WQTD Salary of the Employees.

→ SELECT SAL
FROM EMP;

Q) WQTD Annual Salary of the Employees.

→

* EXPRESSION:

① A statement which gives the result is known as Expression.

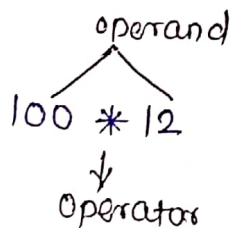
② Expression is combination of operand & operator.

⇒ operand:-

This are the values that we pass.

⇒ operator:-

This are the symbols which performs some operation on the operand.



⇒ Example:- WQTD Annual Salary of the Employees.

→ SELECT SAL * 12
FROM EMP;

Q) WAQTD Half term salary of the Employees.

→ SELECT SAL*6
FROM EMP;

Q) WAQTD Employee Names along with annual salary.

→ SELECT ENAME, SAL*12
FROM EMP;

Q) WAQTD details of the Employees.

→ SELECT *
FROM EMP;

Q) WAQTD details of the Employees along with annual salary.

→ SELECT * EMP.*, SAL*12
FROM EMP;

// when extra column want with all details.
→ EMP.*

Q) WAQTD salary of the Employees with the HIKE of 10%.

→ SELECT SAL + SAL * $\frac{10}{100}$
FROM EMP;

$$\begin{aligned} &\downarrow \\ &100 + 10 \cancel{\cancel{0}} \times \frac{10}{100} \\ &= 100 + 10 \\ &= 110 \end{aligned}$$

Q) WAQTD name and salary of the Employees with the HIKE of 12%.

→ SELECT ^{ENAME,} SAL + SAL * $\frac{12}{100}$
FROM EMP

Q) WAQTD name of the Employees along with annual salary with the HIKE of 20%.

→ SELECT ENAME, SAL*12 + SAL*12 + $\frac{20}{100}$
FROM EMP;

* ALIAS :-

- ① ALIAS is the alternate name given to the column in the result table.
- ② we can assign ALIAS Name with or without using AS.
- ③ ALIAS Name have to be a single string which is separated by an underscore (_) or Enclose with a " " Double quotes

NOTE - SAL*12

- ① ANNUAL SAL
- ② ANNUAL-SAL
- ③ "ANNUAL SAL"

Q) WAGTD Annual Salary of the employees by assigning alias Name as Annual Sal.

→ SELECT SAL*12 ~~AS~~ ANNUAL SAL
FROM EMP;

* ASSIGNMENT 2 *

① WAQTD Name of the Employees Along with their Annual Salary.

→ SELECT ENAME, SAL*12
FROM EMP;

② WAQTD Ename & Job for all the Employees with their Half term Salary.

→ SELECT ENAME, JOB, SAL*6
FROM EMP;

③ WAQTD all the details of Employees along with annual bonus of 2000.

→ SELECT EMP.*, SAL*12+2000
FROM EMP;

④ WAQTD Name and salary and salary with Hike of 10%.

→ SELECT ENAME, SAL, $SAL + SAL * \frac{10}{100}$
FROM EMP;

⑤ WAQTD Name and salary with deduction of 25%.

→ SELECT ENAME, $SAL - SAL * \frac{25}{100}$
FROM EMP;

⑥ WAQTD Name and salary with monthly Hike of 50

→ SELECT ENAME, SAL+50
FROM EMP;

⑦ WAQTD Name and Annual salary with deduction of 10%.

→ SELECT ENAME, $SAL*12 - SAL*12 * \frac{10}{100}$
FROM EMP;

⑧ WAQTD Total salary Given to Each Employee (SAL+COMM)

→ SELECT SAL+comm
FROM EMP;

⑨ WAQTD details of Employees along with Annual salary.

→ SELECT EMP.*, SAL*12
FROM EMP;

⑩ WAQTD Name & Designation along with 100 Penalty in salary.

→ SELECT ENAME, JOB, SAL-100 "penalty"
FROM EMP;

Q) WQTD Details of all the employees.

→ select *
from emp;

Q) WQTD Names of all the employees.

→ select ename
from emp;

Q) WQTD department no of the employees.

→ select deptno
from emp;

Q) WQTD Employee name and department no of the employees.

→ select ename, deptno
from emp;

Q) WQTD name and department no of the employees who are working in the department No 20.

→ select ename, deptno
from emp, ~~emp~~
where deptno = 20;

* Selection :-

① It is a process of retrieve or fetch the data by selecting both columns as well as rows known as selection.

• Syntax :-

```
SELECT COLUMN-NAME/EXPRESSION
FROM TABLE-NAME
WHERE <FILTER-CONDITION>;
```

• Order of Execution :-

```
FROM
↓
WHERE — Row-by-Row
↓
SELECT
```

[NOTE] :-

- ① where clause is used to filter the records
- ② where clause will execute ~~row~~ after the execution of from clause.
- ③ where clause will execute row-by-row.
- ④ In where clause we can write the condition.
- ⑤ we can write multiple conditions in where clause with the help of logical operators.

Example :-

Q) WAGTD Name and Department no of the employees who are working in the department No 20.

```
→ SELECT ENAME, DEPTNO
FROM EMP
WHERE DEPTNO = 20;
```


Output of from clause

EMPNO	ENAME	SAL	DEPTNO
1	A	1000	20
2	B	2000	10
3	C	3000	20
4	D	4000	30
5	E	5000	10
6	F	6000	20
7	G	7000	10

under execution



- 1 A 1000 20 = 20 ✓
- 2 B 2000 10 = 20 X
- 3 C 3000 20 = 20 ✓
- 4 D 4000 30 = 20 X
- 5 E 5000 10 = 20 X
- 6 F 6000 20 = 20 ✓
- 7 G 7000 10 = 20 X

Output of where clause

EMPNO	ENAME	SAL	DEPTNO
1	A	1000	20
3	C	3000	20
6	F	6000	20

Result

ENAME	DEPTNO
A	20
C	20
F	20

③ SELECT ENAME, DEPTNO
 ① FROM EMP
 ② WHERE DEPTNO = 20;

Q) WAQTD Name & Salary of the employees who are getting salary more than 2000.

```
→ select name, sal  
from emp  
where sal > 2000;
```

Q) WAQTD name and Job of the employees who are working as Salesman.

```
→ select ename, job  
from emp  
where job = 'Salesman';  
where job = 'SALESMAN';
```

[NOTE]: The records in SQL are CASE-SENSITIVE

Q) WAQTD details of the employees who are working as Analyst.

```
→ select *  
from emp  
where job = 'ANALYST';
```

Q) WAQTD Name and Hired date of the employees those who Hired after 12-FEB-1980.

```
→ select name, Hired date  
from emp  
where Hired date > '12-FEB-1980';
```

Q) WAQTD details of the employees who hired after 1981.

```
→ select *  
from emp  
where Hired date > '01-JAN-1981' '31-DEC-1981';
```

Q) WAQTD details of the employees those who hired before 1981.

```
→ select *  
from emp  
where Hired date < '01-JAN-1981';
```

* ASSIGNMENT 3 *

1) WAQTD Annual salary of the Employee whose name is 'SMITH'.
→ select sal*12
from Emp
where Ename='SMITH';

2) WAQTD Name of the Employees working as clerk.
→ select Ename
from Emp
where JOB='CLERK';

3) WAQTD Salary of the Employees who are working as salesman.
→ select sal
from Emp
where job='SALESMAN';

4) WAQTD details of the Emp who earns more than 2000.
→ select *
from Emp
where sal > 2000;

5) WAQTD details of the Employees whose name is JONES.
→ select *
from Emp
where Ename='JONES';

6) WAQTD Details of the Employees who was Hired after 01-JAN-81.
→ select *
from Emp
where Hired date > '01-JAN-81';

7) WAQTD Name and sal along with His Annual Salary. if the Annual salary more than 12000.
→ select ~~sal~~ Ename, sal, sal*12
from Emp
where sal*12 > 12000;

8) WAQTD Empno of the employees who are working in Dept 80.
select empno
from Emp
where deptno=80;

⑩ WAQTD Ename & Hired date if they are hired before 1981

→ select Ename, hireddate
from Emp
where hireddate < '01-JAN-1981';

⑪ WAQTD details of Employees working as manager.

→ select *
from Emp
where job = ~~'SALES'~~ 'MANAGER';

⑫ WAQTD Name and Salary given to an Employee If Employee earns a commission of Rupees 1400.

→ select Ename, sal
from Emp
where Comm = 1400;

⑬ WAQTD details of Employees Having commission more than salary.

→ select *
from Emp
where Comm > sal;

⑭ WAQTD Empno of Employees Hired before the year 1987.

→ select Empno
from Emp
where Hireddate < '01-JAN-1987';

⑮ WAQTD Details of Employees working as manager.

→ select *
from Emp
where job = ~~'SALES'~~ 'MANAGER';

⑯ WAQTD details of Employees Earning more than 2000 Rupees per month.

→ select *
from Emp
where sal > 2000;

Q) WAQTD details of all the employees

→ select *
from emp;

Q) WAQTD details of the employees along with annual salary with the Hike of 20% and salary with the deduction of 50% and the employees who are working in the deptno 20.

→ select emp.*, sal*12+sal*12* $\frac{20}{100}$, sal-sal*50/100
from emp
where deptno=20;

Q) WAQTD Name and department no of the employees who are working in the department No 30.

→ select name, deptno
from emp
where deptno=30;

Q) WAQTD name and job of the employees who are working as Salesman.

→ select ename, job
from emp
where job='SALESMAN';

Q) WAQTD details of the employees who are working as salesman in the department no 30

→ select *
from emp
where job='SALESMAN'

13/02/24

* OPERATORS *

- ① Arithmetic Operator (+, -, *, /)
- ② Concatenation operator (||)
- ③ Comparison operator (=, != (or) <>)
- ④ Relational operator (<, >, <=, >=)
- ⑤ Logical operator (AND, OR, NOT)
- ⑥ Special operator (IN, NOT IN, BETWEEN, NOT BETWEEN, IS, IS NOT, LIKE, NOT LIKE)
- ⑦ Sub-Query operator: - (ALL, ANY, EXISTS, NOT EXISTS)

* ② CONCATANATION OPERATOR:-

① It is used to Join the strings.

Example 1:-
SELECT 'MR.' || ENAME
FROM EMP
WHERE JOB = 'MANAGER';



o/p

Example 2:-
SELECT ENAME
FROM EMP
WHERE JOB = 'MANAGER'

ENAME

JONES

BLAKE

CLARK

@ Logical Operator:-

① Logical operator is used to write the multiple conditions inside the where clause

② It has 3 Types of Logical Operators:- ① Logical AND

② Logical OR

③ Logical NOT

* LOGICAL AND OPERATOR:-

① AND operator returns true if all the conditions are satisfied.

Ex:- WAQTD Details of the Employees who are working as salesman in deptno 30

→ select *
from Emp
where Job = 'SALESMAN' AND DEPTNO = 30;

② WAQTD Salary and department no of the Employees who are getting salary more than 1000 in the deptno 30.

→ select sal, deptno
from Emp
where sal > 1000 and deptno = 30;

* OR OPERATOR

- or operator returns true if any one condition is satisfied.

- Example:- WAQTD details of the Employees who are working as salesman or in the department Number 30

→ select *
from Emp
where Job = 'SALESMAN' OR DEPTNO = 30;

③ WAQTD Name and department no of Employees who are working in the deptno 10 or 20.

→ select name, deptno
from Emp
where deptno = 10 or deptno = 20;

③ NOT OPERATOR

- It is a Inverse operator of this selected names will be rejected.
- Ex:- WAGTD Name and deptno of the Employees who are not working in the department No 20.

→ (11) select Ename, deptno
from Emp
where deptno $\neq$ 20;

(12) select Ename, deptno
from Emp
where deptno $< >$ 20;

(13) select Ename, deptno
from Emp
where deptno not 20;

* ASSIGNMENT 4 *

① WAQTD details of the Employees working as clerk and Earning less than 1500.

→ select *
from emp
where job='CLERK' AND SAL < 1500;

② WAQTD Name and hireddate of the Employees working as manager in deptno 30.

→ select ename, hireddate
from emp
where job='MANAGER' and deptno=30;

③ WAQTD Details of the Employees along with annual salary if they are working in Deptno 30 as Salesman and their annual salary has to be > 14000

→ select emp.*, sal*12
from emp
where deptno=30 and job='SALESMAN' AND SAL*12 > 14000;

④ WAQTD names of the Employees whose salary less than 1100 & their designation is clerk.

→ select ename
from emp
where sal < 1100 and job='CLERK';

⑤ WAQTD all the details of the Employees working in deptno 30 or as analyst.

→ select *
from emp
where deptno=30 or job='ANALYST';

⑥ WAQTD name, sal, Annual sal and deptno if deptno is 20 Earning more than 1100 and annual salary exceeds 12000.

→ select ename, sal, sal*12, deptno
from emp
where deptno=20 and sal > 1100 and sal*12 > 12000;

⑦ WAQTD Empno and Names of the Employees working as manager in deptno 20.

```
→ select Empno, Ename  
   from Emp  
  where job='MANAGER' and deptno=20;
```

⑧ WAQTD details of the Employees working in deptno 20 or 30.

```
→ select *  
   from Emp  
  where deptno=20 or deptno=30;
```

⑨ WAQTD details of employees working as Analyst in Dept 10.

```
→ select *  
   from Emp  
  where job='ANALYST' AND deptno=10;
```

⑩ WAQTD details of the Employees working as president with salary of rupees 4000.

```
→ select *  
   from Emp  
  where job='PRESIDENT' and sal=4000;
```

⑪ WAQTD names and deptno, job of Employees working as Clerk ^{or} ~~in~~ manager deptno 10 or 20

```
→ select Ename, deptno, job  
   from Emp  
  where job='CLERK' or job='MANAGER' AND (deptno=10 or deptno=20);
```

⑫ WAQTD details of the Employees working as clerk or manager in deptno ¹⁰

```
→ select *  
   from Emp  
  where (job='CLERK' or job='MANAGER') AND DEPTNO=10;
```

⑬ WAQTD Names of the employee working in dept 10, 20, 30, 40.

```
→ select Ename  
   from Emp  
  where deptno=10 or deptno=20 or deptno=30 or deptno=40;
```


14) WAQTD details of the employees with Empno 7902, 7899.

```
→ select *  
  from Emp  
 where Empno = 7902 AND Empno = 7899;
```

15) WAQTD Details of the employees working as manager or salesman or clerk.

```
→ select *  
  from Emp  
 where job = 'MANAGER' or job = 'SALESMAN' or job = 'CLERK';
```

16) WAQTD Names of the employees Hired after 81 and before 87.

```
→ select Ename  
  from Emp  
 where Hired date > '31-DECJAN-1981' AND Hired date < '01-JAN-1987';
```

17) WAQTD details of Employees Earning more than 1250 but less than 3000.

```
→ select *  
  from Emp  
 where sal > 1250 AND sal < 3000;
```

18) WAQTD Names of the employees Hired after 81 into Dept 10 or 30.

```
→ select Ename  
  from Emp  
 where Hired date > '31-DEC-1981' AND Deptno = 10 or deptno = 30;
```

19) WAQTD Names of the employees along with annual salary for the employees working as manager or clerk into Dept 10 or 30.

```
→ select Ename, sal * 12  
  from Emp  
 where job = 'MANAGER' or job = 'CLERK' or deptno = 10 or deptno = 30;
```

20) WAQTD all the details along with annual salary if sal is between 1000 AND 4000, ANNUAL salary more than 15000.

```
→ select Emp.*, sal * 12 from Emp  
 where sal > 1000 and sal < 4000
```

Q) WATD details of all the Employees

→ Select *
from Emp;

Q) WATD details of the Employees along with annual salary with the Hike of 20% & the Employees who are working as salesman, manager, President, Analyst, clerk in Deptno 10, 20, 30 and the Employees who are getting salary more than 1000 but less than 6000 and the Employees are Hired after 1980 but before 1990 and the Employees who are getting Commission more than 10 RS but less than 3000 and the Employees who are getting salary more than Comm.

→ Select * Emp.*, Sal * 12 + Sal * 12 * 20 / 100
from Emp

where (Job = 'SALESMAN' or Job = 'MANAGER' or Job = 'PRESIDENT'
or Job = 'ANALYST' or Job = 'CLERK')

AND (Deptno = 10 or deptno = 20 or deptno = 30)

AND (Sal > 1000 and Sal < 6000)

AND (Hireddate > '31-DEC-1980' and Hireddate < '01-JAN-1990')

AND (Comm > 10 and Comm < 3000)

AND (Sal > Comm);

* SPECIAL OPERATORS *

1

- ① IN
- ② NOT IN
- ③ BETWEEN
- ④ NOT BETWEEN
- ⑤ IS
- ⑥ IS NOT
- ⑦ LIKE
- ⑧ NOT LIKE

① IN Operator :-

- ⊕ It is a multivalued operator it can accept one value at LHS & multiple values at RHS.
- ⊕ We can use IN operator instead of using = (Equal to) operator.

* Syntax - COL-NAME / EXPRESSION IN (N, M₁, M₂ -- V_n)

* Example :- WQTD Name and deptno of the employees those who are working in the department NO 10, 20, 30.

→ Select Ename, deptno
from Emp
where deptno IN (10, 20, 30);

② WQTD details of the employees those who are working in the deptno 30 as Salesman, manager,

→ select *
from Emp
where deptno=30 and JOB IN ('SALESMAN', 'MANAGER');

④ WAQTD details of the employees those who are working in dept No 20

→ select *
from emp
where deptno = 20;

select *
from emp
where deptno IN 20;

② NOT IN OPERATOR:-

① NOT IN operator is similar to IN operator instead of selected values will be rejected.

* Syntax:- $\text{COL-NAME} \text{ EXP NOT IN } (v_1, v_2, v_3 \dots v_n);$
LHS RHS

* Example:- WAQTD name and department no of the employees who are not working in the deptno 10, 30.

→ select deptno, empname, deptno
from emp
where deptno NOT IN (10, 30);

③ IS OPERATOR:-

① IS operator is used to compare with null.

* Syntax:- $\text{column-name} \text{ EXP IS NULL};$

* Example:- WAQTD name and commission of the employees who are not getting commission

→ select ename, comm
from emp
where comm IS null;

Q) WQTD name and MGR NO of the Employees who are not having MGR.

```
→ select Ename, MGR  
from Emp  
where MGR IS NULL;
```

IS NOT OPERATOR:-

① IS NOT operator is a similar to IS operator instead of selected values will be rejected.

* Syntax:- COLUMN-NAME / EXP IS NOT NULL;

* Example:- WQTD Name and Commission of the Employees who are getting the commission.

```
→ select Ename, Comm  
from Emp  
where COMM IS NOT NULL;
```

WQTD details of the Employees who are having MGR.

```
→ select *  
from Emp  
where MGR IS NOT NULL;
```

Q) WQTD details of all the Employees.

```
→ select *  
from Emp;
```

Q) WQTD details of the Employees along with annual Salary with the Hike of 10% and the Employees who are working as Salesman, Analyst in the deptno 10, 20, 30 and the Employees are getting Salary more than 1000 and the Employees who are having mgr and Not getting commission.

```
→ select Emp.*, Sal * 12 + Sal * 12 * 10 / 100;  
from Emp
```

→ SELECT Emp.*, Sal*12+Sal*12*10/100
FROM Emp

~~WHERE (job='SALESMAN' OR J~~

WHERE job IN ('SALESMAN', 'ANALYST')

AND DEPTNO IN (10, 20, 30)

AND SAL > 1000

AND MGR IS NOT NULL

AND COMM IS NULL;

BETWEEN OPERATOR:-

① Between Operator is used whenever we have range of values.

② Range Between lower range to Higher range.

* Syntax COL-NAME/EXP BETWEEN LOWER-RANGE AND HIGHER-RANGE;

* Example:- WAQTD NAME & Sal of the Employees who are getting salary more than 2000.

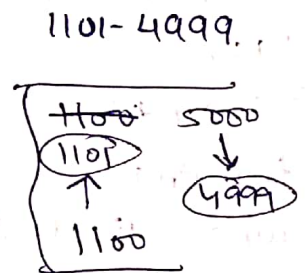
→ select ename, sal
from emp
where sal > 2000;

Q) WAQTD name and salary of the Employees who are getting salary in the range of 1000 to 4000.

→ select ename, sal
from emp
where sal BETWEEN 1000 AND 4000;

Q) WAQTD Name and salary of the Employees who are getting the salary in the range Between 1100 to 5000.

→ select ename, sal
from emp
where sal BETWEEN ~~1100~~ 1101 AND ~~5000~~ 4999;



Q) WAQTD Name & salary of the Employee who are getting salary in the range between 1500 to 5000.

→ select ename, sal
from emp
where sal BETWEEN 1501 AND 4999;

Q) WAQTD details of the Employees who are getting salary more than 1000 but less than 5000.

→ Select *
from Emp
where Sal BETWEEN 1001 AND 4999;

Q) WAQTD Name and Hiredate of the Employees who hired after 1980.

→ Select Name, Hiredate
from Emp
where hiredate > '31-DEC-1980';

Q) WAQTD name and Hiredate of the Employees who hired in the range of 1980 to 1987.

→ select Name, Hiredate
from Emp
where Hiredate BETWEEN '01-^{JAN}~~DEC~~-1980' AND '31-DEC-1987';

Q) WAQTD details of the Employees who hired after 1981 but before 1987.

→ Select *
from Emp
where Hiredate BETWEEN ~~'31-DEC-1981'~~ AND '01-JAN-1982'
AND '31-DEC-1986';

NOT BETWEEN:-

① Not Between operator is similar to Between operator. instead of Selected values will be rejected

* Syntax:- COL-NAME/EXPRES NOT BETWEEN LOWER-RANGE AND HIGH-RANGE;

* Example:-

WAQTD Name and salary of the Employees who are not getting salary in the range of 1000 to 4000.

→ select Ename, sal

from Emp

where sal NOT BETWEEN 1000 AND 4000;

* LIKE OPERATOR:-

- Like operator is used whenever we have pattern matching

* Syntax:- Column.name/Exp LIKE 'PATTERN-TO-MATCH';

* Example:-

① WAQTD Names of the Employees

→ select Ename

from Emp;

② WAQTD Names of the Employees whose name starts with character A.

→ select Ename

from Emp

where Ename LIKE 'A%';

NOTE:-

To Achieve pattern matching we need to Special Characters

① % (percentile)

② _ (underscore)

'A%'
 '%A'
 '%A%'
 '%A%A%'
 '%AA%'
 '-----'

Q) WRTD names of the Employees whose name ends with Character A.

→ select Ename

from Emp

where Ename like '%A';

*NOT LIKE:-

① Not like operator is a similar to like operator instead of selected values will be rejected.

→ Syntax:-

COL-NAME/EXP NOT LIKE 'PATTERN-TO-MATCH';

* Example:-

Q) WRTD Names of the Employees whose name does not starts with char A

→ select Ename

from Emp

where Ename NOT LIKE 'A%';

* ASSIGNMENT 5 *

Q1) list all the Employees whose commission is null.

```
→ select *  
  from Emp  
 where comm is null;
```

Q2) list all the Employees who don't have reporting manager.

```
→ select *  
  from Emp  
 where mgr is null;
```

Q3) list all the salesman in Dept no 30.

```
→ select *  
  from Emp  
 where job='SALESMAN' AND DEPTNO=30;
```

Q4) list all the salesman in deptno 30 and having salary greater than 1500.

```
→ select *  
  from Emp  
 where job='SALESMAN' AND DEPTNO=30 AND SAL>1500;
```

Q5) list all the Employees whose name starts with 'S' or 'A'.

```
→ select *  
  from Emp  
 where Ename Like 'S%' or Ename like 'A%';
```

Q6) list all the Employees except those who are working in Dept 10 & 20.

```
→ select *  
  from Emp  
 where deptno NOT IN (10, 20);
```

Q7) list the Employees whose name does not starts with 'S'.

```
→ select *  
  from Emp  
 where Ename NOT LIKE 'S%';
```

Q8) list all the Employees who are having reporting manager in Dept 10.

```
→ select *  
  from Emp  
 where mgr IS NOT NULL AND DEPTNO=10;
```

Q9) list all the employees whose comm is null & working as clerk.

```
→ select *  
  from emp  
 where comm IS NULL AND JOB='CLERK';
```

Q10) list all the employees who don't have reporting manager in deptno 10 or 30.

```
→ select *  
  from emp  
 where MGR IS NULL AND DEPTNO IN (10, 30);
```

Q11) list all the salesman in Dept 30 with sal more than 2450.

```
→ select *  
  from emp  
 where job='SALESMAN' AND DEPTNO=30 AND SAL>2450;
```

Q12) list all the analyst in Dept 20 and having salary greater than 2500.

```
→ select *  
  from emp  
 where job='ANALYST' AND DEPTNO=20 AND SAL>2500;
```

Q13) list all the employees whose name starts with 'M' or 'J'.

```
→ select *  
  from emp  
 where ENAME like 'M%' or ENAME like 'J%';
```

Q14) list all the employees with annual salary except those who are working in Dept 30.

```
→ select emp.*, sal*12  
  from emp  
 where DEPTNO NOT IN 30;
```

Q15) list the employees whose name does not end with 'ES' or 'R'.

```
→ select *  
  from emp  
 where ENAME NOT LIKE '%ES' or '%R'; AND  
 ENAME NOT LIKE '%R';
```


Q16) list all the employees who are having reporting manager in Dept 10 along with 10% of Hike in salary.

```
→ select * emp.*, SAL+SAL*10/100  
from emp  
where mgr not IS NOT NULL AND DEPTNO=10; AND
```

Q17) display all employees who are salesman Having 'E' as the last but one character in Ename but salary having exactly 4 character.

```
→ select *  
from emp  
where job='SALESMAN' AND ENAME LIKE '%E' AND SAL LIKE '----';
```

Q18) display all the employee who are joined after year 81.

```
→ select *  
from emp  
where Hiredate > '81-DEC-1981';
```

Q19) Display all the employees who are joined in FEB.

```
→ select *  
from emp  
where Hiredate like '%FEB%';
```

Q20) list all employees who are not working as manager & clerks in Dept 10 and 20 with a salary in range of 1000 to 3000.

```
→ select *  
from emp  
where JOB NOT IN ('MANAGER', 'CLERK') AND DEPTNO IN (10, 20)  
AND SAL BETWEEN 1000 AND 3000;
```

19/02/24

Q) WRTQ to display details of all the employees

→ select *
from emp;

Q) WRTQ Names of all the employees

→ select ename
from emp;

Q) WRTQ Names of the employees whose name starts with 'Vowels'.

→ select ename
from emp

where ename like 'A%' or ename like 'E%' or ename like 'I%'
or ename like 'O%' or ename like 'U%';

Q) WRTQ Names of the employees whose name starts with S or A.

→ select ename
from emp

where ename like 'S%' or ename like 'A%';

Q) WRTQ Names of the employees whose name does not start with A & S.

→ select ename
from emp

where ename not like 'S%' ^{AND} ~~AND~~ ename not like 'A%';

Q) WRTQ Names of the employee whose name does not start with vowels.

→ select ename
from emp

where ename not like 'A%' ^{AND} ~~AND~~ ename not like 'E%' ^{AND} ~~AND~~ ename not like 'I%'
^{AND} ~~AND~~ ename not like 'O%' ^{AND} ~~AND~~ ename not like 'U%';

Q) WAGTD all the details of Employees Present in the Employee table

→ select *
from Emp;

Q) WAGTD salary of the Employees.

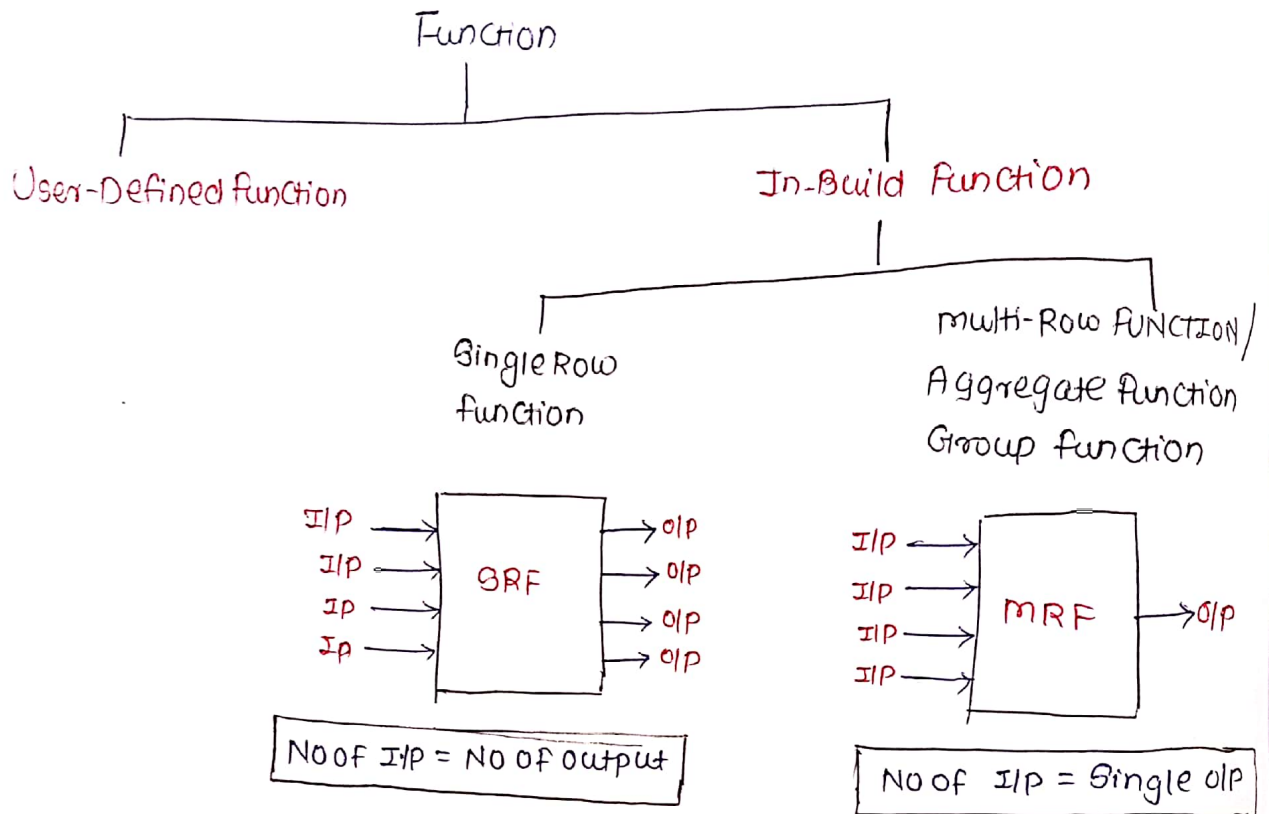
→ select sal
from Emp;

Q) WAGTD maximum salary of the Employees.

→ select
from Emp;

* FUNCTIONS *

① Functions are block of codes or list of instructions which is used to perform specific task.



* SINGLE ROW FUNCTION :-

- ① ⇒ If we give single Input to the single row function it will provide single output.
- ② If we give multiple Inputs to the single row function we will get multiple outputs
- ③ No of Inputs = No of OUTPUTS

* MULTI ROW FUNCTION :-

- ① If we give single Input to the multi-row function it will provide single o/p
- ② If we give multiple Inputs to the multi-row function it will return single output.

No of Inputs = Single output

* Group function

- ① max()
- ② min()
- ③ sum()
- ④ ~~count~~ avg()
- ⑤ count()

* max():

⇒ It is used to obtain maximum value present in the column.

* min():-

⇒ It is used to obtain the minimum value present in the column.

* sum():-

⇒ It is used to obtain the sum of value present in the column.

* avg():

It is used to obtain the average of values

* count()

It is used to obtain No of values present in the column.

NOTE:-

- ① Multi-Row function cannot accept null values
- ② Along with multi-row function we cannot use any other Column-name in the select clause.
- ③ In select clause we can use more than one multi-row function.
- ④ We cannot use multi-row functions in the where clause.
- ⑤ count is the only multi-row function which can accept * as an argument.

Example:- WAQTD maximum salary of the employees.

→ select max(sal)
from emp;

→

max(sal)
5000

2) WAQTD minimum salary of the employees.

```
→ select min(sal)
   from emp;
```

3) WAQTD Total salary of the employees.

```
→ select sum(sal)
   from emp;
```

4) WAQTD Average salary of the employees.

```
→ select avg(sal)
   from emp;
```

5) WAQTD No of employees working in the department NO 30.

```
→ select count(*)
   from emp
  where deptno=30;
```

```
select count(deptno)
   from emp
  where deptno=30;
```

* ASSIGNMENT ON MRF() *

① WAQTD No of Employees getting salary less than 2000 in DeptNO 10.

→ select count(*)

from Emp

where sal < 2000 AND Deptno = 10;

② WAQTD Total salary needed to pay Employees working as Clerk.

→ select sum(sal)

from Emp

where JOB = 'CLERK';

③ WAQTD average salary needed to pay all employees;

→ select avg(sal)

from Emp;

④ WAQTD No of Employees Having a 'A' as their first character.

→ select count(*)

from Emp

where ename like 'A%';

⑤ WAQTD No of Employees working as clerk or manager.

→ select count(*)

from Emp

where JOB IN ('CLERK', 'MANAGER');

⑥ WAQTD Total salary needed to pay Employees Hired in Feb.

→ select sum(sal)

from Emp

where ~~JOB~~ like '%FEB%';
hiredate

⑦ WAQTD No of Employees Reporting to 7839 (MGR).

→ select count(*)

from Emp

where MGR = 7839;

⑧ WAQTD No of Employees getting commission In DeptNo 30-

→ select count(*)

from emp

where comm IS NOT NULL AND DEPTNO = 30;

⑨ WAQTD Avg sal, Total sal, No of Employees And maximum salary working as President.

→ select avg(sal), ~~total~~ sum(sal), count(*), max(sal)

from emp

where JOB = 'PRESIDENT';

⑩ WAQTD No of Employees Having 'A' in their names-

→ select count(*)

from emp

where Ename like '%A%';

⑪ WAQTD No of Employees and total salary needed to pay the Employer who have 2 Consecutive L's in their Name.

→ select count(*), sum(sal)

from emp

where Ename like '%LL%';

⑫ WAQTD No of department present in Employee Table.

→ select count(Distinct deptno)

from emp;

⑬ WAQTD No of Employees Having character '1';

→ select count(*)

from emp

where Ename like '%1%';

⑭ WAQTD No of Employees Having 'S' in their names.

→

⑮ WAQTD Total salary given to employees working as clerk in Dept 30

→ select sum(sal)

from emp

where job = 'CLERK' AND DEPTNO = 30;

⑯ WAQTD maximum salary given to employees working as Analyst.

→ select max(sal)

from emp

where job = 'ANALYST';

⑰ WAQTD No of distinct salaries present in Employee Table

→ select count(DISTINCT sal)

from emp;

⑱ WAQTD No of Jobs present in Employee Table

→ select COUNT(JOB)

from emp;

⑲ WAQTD Avg salary given to the clerk.

→ select avg(sal)

from emp

where job = 'CLERK';

⑳ WAQTD minimum salary given to employees who works in DeptNo 10
~~and~~ as manager or clerk.

→ select min(sal)

from emp

where Deptno = 10 AND JOB IN ('CLERK', 'MANAGER');

21/02/24

Q) WATD details of all the employees.

→ select *
from Emp;

Q) WATD Name of the employee who are working in deptno 10

→ select ename
from Emp
where deptno=10;

Q) WATD NO of employees who are working in the dept no 10.

→ select count(*)
from Emp
where deptno=10;

Q) WATD NO of employees working in each department.

→ select COUNT(*)

* GROUP BY CLAUSE :-

① It is used to group the records.

* Syntax

```
SELECT GROUP FUNCTION / GROUP BY EXP  
FROM TABLE-NAME  
[WHERE < FILTER CONDITION >]  
GROUP BY COLUMN-NAME / EXP
```

* Order of Execution

```
FROM  
  ↓↓  
[WHERE — Row-by Row]  
  ↓↓  
GROUP BY — Row by row  
  ↓↓  
SELECT — group by group
```

NOTE :-

- ① Group by clause will execute after the execution of from clause.
- ② Group by clause will executes row by row.
- ③ Group by clause is used to group the records.
- ④ After the execution of group by clause we will get the groups.
- ⑤ Any clause which executes after the execution of group by clause it will executes group by group.

Example :- WQFTD NO of Employees working in each department.

```
⇒ SELECT COUNT(*), DEPTNO  
FROM EMP  
GROUP BY DEPTNO;
```

Emp:

EMPNO	ENAME	SAL	DEPTNO
1	A	1000	20
2	B	2000	30
3	C	3000	20
4	D	4000	10
5	E	5000	20
6	F	6000	30
7	G	7000	10

20	1 A 1000 20	=3
	3 C 3000 20	
	5 E 5000 20	
30	2 B 2000 30	=2
	6 F 6000 30	
10	4 D 4000 10	=2
	7 G 7000 10	

- ③ SELECT COUNT(*), DEPTNO
 ① FROM EMP
 ② GROUP BY DEPTNO;

Result

Count (*)	Deptno
3	20
2	30
2	10

Q) WAQTD NO of Employees, minimum salary of Employees who are working in each department.

```
→ select count(*), min(sal), deptno  
from Emp  
Group By deptno;
```

Q) WAQTD NO of Employees who are getting salary more than 2000 in each dept

```
→ select count(*), deptno  
from Emp  
where sal > 2000  
Group by deptno;
```

Q) WAQTD NO of Employees working in each job.

```
→ select count(*), job  
from Emp  
group by job;
```

* ASSIGNMENT ON GROUP BY *

① WAQTD No of Employees working in Each department Except president.

→ select count(*), DEPTNO

from Emp

where JOB NOT IN 'PRESIDENT'

GROUP BY DEPTNO;

<>

② WAQTD Total salary needed to pay all the Employees in Each job.

→ select sum(SAL), JOB

from Emp

GROUP BY JOB;

③ WAQTD No of Employees working as manager in Each department.

→ select count(*), DEPTNO

from Emp

where JOB = 'MANAGER'

GROUP BY DEPTNO;

④ WAQTD Average salary needed to pay all the Employees in Each Department Excluding the Employees of Deptno 20.

→ select avg(sal), Deptno

from Emp

where Dept NOT IN 20

GROUP BY DEPTNO;

⑤ WAQTD No of Employees having character 'A' in their names in Each job.

→ select count(*), JOB

from Emp

where ename like '%A%'

GROUP BY JOB;

⑥ WAQTD No of Employees And avg salary needed to pay the employees whose salary greater than 2000 in each dept

```
→ select count(*), Avg(sal), Deptno  
from Emp  
where sal > 2000  
Group by deptno;
```

⑦ WAQTD Total salary needed to pay & No of Emp Salesman in each dept

```
→ select sum(sal), deptno, count(*)  
from Emp  
where JOB = 'SALESMAN'  
Group by deptno;
```

⑧ WAQTD No of Employees with their maximum salaries in each Job.

```
→ select COUNT(*), max(sal), JOB  
from Emp  
Group by JOB;
```

⑨ WAQTD maximum salaries given to employee working in each dept.

```
→ select max(sal), deptno  
from Emp  
Group by deptno;
```

⑩ WAQTD No of Times the salaries present in Employee Table.

```
→ select count(*), sal  
from Emp  
Group by sal;
```

Q) WAQTD details of all the employees.

→ select *
from Emp;

Q) WAQTD department no of the employees.

→ select deptno
from Emp;

Q) WAQTD NO of employees who are working in the deptno 20

→ select count(*)
from Emp
~~from~~ where deptno = 20
~~group by deptno;~~

Q) WAQTD NO of employees who are working in each department

→ select count(*), deptno
from Emp
group by deptno;

Q) WAQTD NO of employees working in each department & their have at least 2 employees in each department.

→ select count(*), deptno
from Emp
group by deptno

* HAVING CLAUSE *

① having clause is used to filter the groups.

* SYNTAX

```
SELECT GROUP/FUNCTION/ GROUP BY EXP  
FROM TABLE_NAME  
WHERE <FILTER-CONDITION>  
GROUP BY COL_NAME  
HAVING <GROUP-FILTER-COND>
```

* ORDER OF EXECUTION:-

FROM
↓
[WHERE Row-by-Row]
↓
GROUP BY Row-by-Row
↓
HAVING ← Group-by-group
↓
SELECT group by group

* NOTE :-

- ① Having clause will execute after the execution of group by clause.
- ② Having clause will execute group by group
- Having clause is used to filter the group
- Without using group by clause we cannot use Having clause.
- In Having clause we can use multi-row function

* Ex

* Example:-

Q) WRTD NO of employees who are working in each dept & they have at least 2 employees in each department.

COUNT(*) ≥ 2

SELECT COUNT(*), DEPTNO
FROM EMP
GROUP BY DEPTNO
HAVING COUNT(*) >= 2;

COUNT(*)	DEPTNO
3	20
2	10
2	30

20 ⇒ 3

1	A	1000	20
3	C	9000	20
6	F	6000	20

10 ⇒ 2

2	B	2000	10
5	E	5000	10

30 ⇒ 2

4	D	4000	30
7	G	7000	30

Emp

EMPNO	ENAME	SAL	DEPTNO
1	A	1000	20
2	B	2000	10
3	C	9000	20
4	D	4000	30
5	E	5000	10
6	F	6000	20
7	G	7000	30

Q) WAQTD No of Employees who are working in each job & their have at least 2 employees in each job.

→ select count(*), job
from emp
group by job
having count(*) ≥ 2;

Q) WAQTD minimum salary, maximum salary, No of Employees who are working in each department and the employees are getting salary more than 1000 and maximum salary more than 2000.

→ select max(sal), min(sal), count(*), deptno
from emp where sal > 1000
group by deptno
having max(sal) > 2000;

Q) WAQTD Salaries which are repeated.

→ select count(*), sal from emp. group by sal having count(*) ≥ 2;		select sal from emp group by sal having count(*) ≥ 2;
---	--	--

Q) WAQTD Hired date which are duplicated

→ select count(*), Hire date
from emp
group by Hire date
having count(*) ≥ 2;

* ASSIGNMENT ON HAVING CLAUSE *

✓ ① WAQTD DNO, AND No of Employees working in each department if there are at least 2 clerk in each department.

```
→ select Deptno, count(*)  
    from Emp where job = 'CLERK'  
    group by deptno  
    having count(*) >= 2;
```

② WAQTD DNO and total salary needed to pay all Emp in each dept if there are at least 4 Emp in each department.

```
→ select deptno, sum(sal)  
    from Emp  
    group by deptno  
    having count(*) >= 4;
```

③ WAQTD No of employees earning sal > 1200 in each job.

```
→ select count(*), job  
    from Emp  
    where sal > 1200  
    group by job;
```

④ WAQTD Deptno & No of employees working only if there are 2 Emp working in each dept as manager

```
→ select count(*), deptno  
    from Emp  
    where job = 'MANAGER'  
    group by deptno  
    having count(*) >= 2;
```


Q5) WAQTD Job & max sal of emp in each job if the max sal exceeds 2600

```
→ select job, max(sal)
   from emp
   group by job
   having max(sal) > 2600;
```

Q6) WAQTD The salaries which are repeated in emp table

```
→ select count(*), sal
   from emp
   group by sal
   having count(*) >= 2;
```

Q7) WAQTD The Hiredate which are duplicated in emp table

```
select count(*), Hiredate
   from emp
   group by Hiredate
   having count(*) >= 2;
```

Q8) WAQTD Avg sal of each dept if avg sal is less than 3000.

```
→ select avg(sal), deptno
   from emp
   group by deptno
   having avg(sal) < 3000;
```

Q9) WAQTD Deptno if there are atleast 3 emp in each dept whose name has char 'A' or 'S'

```
→ select deptno, count(*)
   from emp where ename like '%A%' or ename like '%S%'
   group by deptno
   having count(*) >= 3;
```

Q10) WAQTD min & max salaries of each job if min sal is more than 1000 and max sal < 5000.

```
→ select min(sal), max(sal), job
   from emp
   group by job
   where having max(sal) < 5000 AND MIN(sal) > 1000;
```

① WAQTD details of all the employees.

→ select *
from emp;

② WAQTD No of records present in Employee table.

→ select count(*)
from emp;

③ WAQTD No of Employees, minimum salary of the employees who are working in the department No. 10

→ select count(*), min(sal)
from emp
where deptno = 10;

④ WAQTD minimum salary, maximum salary of the employees who are working in each department as manager & getting sal more than 1000.

→ select min(sal), max(sal), deptno
from emp
where job = 'MANAGER' and sal > 1000
GROUP BY deptno;

⑤ WAQTD No of Employees, total salary of the employees who are working in each job & their have at least 3 employees in each job and getting salary more than 1000.

→ select count(*), sum(sal), job
from emp
where sal > 1000
group by job
having count(*) >= 3;

Q) Diff b/w where clause and Having clause

where Clause

- ① where clause is used to filter the records
- ② where clause will execute before the execution of group by clause
- ③ where clause will execute Row-by-Row
- ④ where clause does not accept multi-row function

Having clause.

- ① Having clause is used to filter the groups.
- ② Having clause will execute after the execution of group by clause.
- ③ Having clause will execute group by group.
- ④ Having clause can accept multi-row-function

Q) WAQTD salary of all the employees.

→ select sal
from emp;

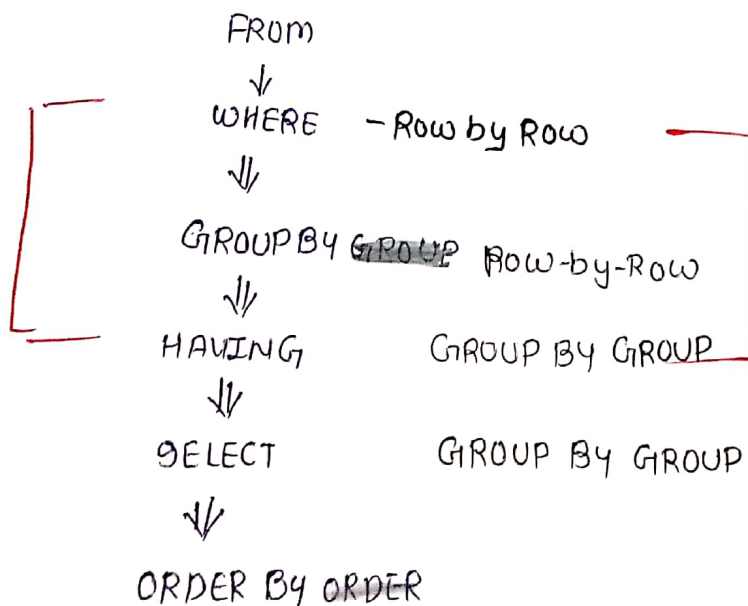
* ORDER BY CLAUSE :-

- ① Order by clause is used to arrange the records of the result table either in ascending or descending order.
- ② By default order by clause will arrange the records of the result table in ascending order.
- ③ If we want to arrange the records in descending order we have to use one keyword called desc.
- ④ Order By clause must be ~~return~~ written at last statement in the query.
- ⑤ Order by clause is only the clause which execute after the select clause.

* Syntax :-

```
SELECT GROUP-FUNCTION/ GROUP-EXP  
FROM TABLE-NAME  
[ WHERE <FILTER-CONDITION>  
  GROUP BY COLUMN-NAME/EXP  
  HAVING <GROUP-FILTER-COND> ]  
ORDER BY COL-NAME ASC/ DESC;
```

* Order of Execution :-



Example:- WAQTD Names of all the Employees

→ select Ename
from Emp;

Q) WAQTD Names of the Employees in Ascending order

→ select Ename
from Emp
order by Ename;

Q) WAQTD Names of the Employee in descending order;

→ select Ename
from Emp
order by Ename desc;

Q) WAQTD Name and Salary of the Employees who are working as Salesman, manager and the records should display in descending order based on salary column.

→ select Ename, Sal
from Emp
where JOB IN ('SALESMAN', 'MANAGER')
ORDER BY Sal DESC;

Q) WAQTD Name & Salary of the Employees the Names should be displaying Ascending order and Salary should display in descending order.

→ select Ename, Sal
from Emp
order by Ename , Sal desc;

Emp

ENAME	SAL
A	100
B	200
D	300
B	400
C	500
E	600

Result
⇒

ENAME	SAL
A	100
B	400
B	200
C	500
D	300
E	600

Q) WAQTD details of all the employees.

→ select *
from emp;

Q) WAQTD Names of all the employees.

→ select ename
from emp;

Q) WAQTD Names of the employees whose name starts with character A.

→ select ename
from emp
where ename like 'A%';

Q) WAQTD Names of the employees whose name ends with S.

→ select ename
from emp ^{like}
where ename like '%S';

Q) WAQTD Names of the employees whose name consist of character A.

→ select ename
from emp
where ename like '%A%';

Q) WAQTD details of the employees whose name consist of 5 characters.

→ select *
from emp
where ename like '_____';

Q) WAQTD Names of the employees whose name starts with '_'

→ select ename
from emp
where ename like '_%';

* ESCAPE CHARACTER:-

- ① ESCAPE character is used to remove the special Behaviour of a Special character and to treat it as Normal character.
- ② ESCAPE character must be return before the special character. which has to be treated as a Normal character.
- ③ ESCAPE character must be defined after using
- ④ we can use any character as an Escape character but recommended character for ESCAPE characters are (!, \$, /, \)
- ⑤ we can use the concept of ESCAPE character only to look for (%) & (-)

* Syntax:-

column_name/Exp like/Not like 'Pattern-to-match' ESCAPE 'CHAR'

* Example:-

```
% AKSHAY_KAMBLE%  
- CUTEBOY_ADITHYA%  
% VIRAJ SAWANT -  
- GOKU% ANIME -  
- PRASADPALVI -
```

Q) WAQTD Names of the employees whose name starts with '_'.

```
→ select Ename  
from Emp ESC  
where Ename like '$_%' ESCAPE '$'
```

Q) WAQTD Names of the employees whose names starts & ends with '-'

```
→ select Ename  
from Emp ESC  
where Ename like '$_-%_-' ESCAPE '$';
```

Q) WAP TO Names of the employee whose name consists of %

→ Select Ename

from emp

where Ename like '\$_%\$_' ESCAPE '\';

Q) WAP TO Names of the employees whose name ends with -

→ Select Ename

from emp

where Ename like '%\$_' ESCAPE '\';

* NOTE :-

① '\$_%' ESCAPE '\'

EC NC SC

Starts with - %

② '%\$_' ESCAPE '\'

SC EC NC

Ends with - %

③ '\$_%\$_' ESCAPE '\'

EC NC SC EC NC

Starts and ends with - %

④ '%\$_%' ESCAPE '\'

SC EC NC SC

- Consists - (% - %)

⑤ '\$_%\$_%' ESCAPE '\'

EC NC SC EC NC

Q) WAP TO details of all the employees.

→ select *
from Emp;

Q) WAP TO Name, salary of the employees.

→ select Ename, Sal
from Emp;

Q) WAP TO details of the employees who are getting salary in the range b/n 1000 to 4000.

→ select *
from Emp
where Sal Between 1001 AND 3999;

Q) WAP TO Name and salary of the employees who are getting salary more than 2000.

→ select Ename, Sal
from Emp
where Sal > 2000;

Q) WAP TO Name and salary of the employees who are getting salary more than 'MILLER'.

→ select Ename, Sal
from Emp
where Sal

* SUBQUERY:-

① A Query return inside another Query is known as subquery.

② * Working Principle of Subquery:-

let us consider 2 Queries

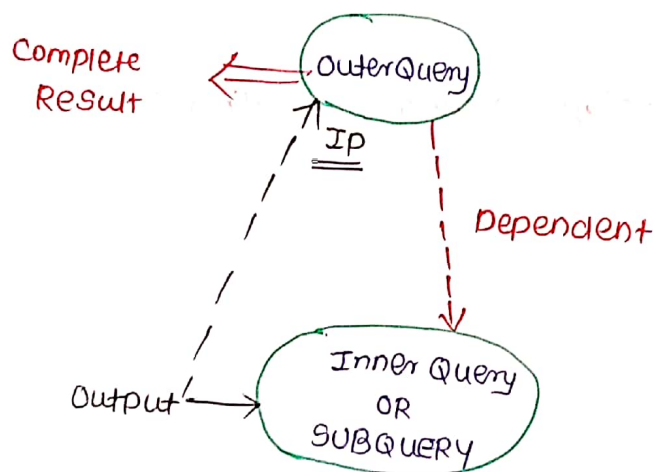
- ① Outer Query
- ② Inner Query.

⊕ Inner Query will Execute first and produce an output.

⊕ The output of Inner Query is given as an Input to the Outer Query.

⊕ The outer Query will Executes and generates the result.

⊕ Therefore the outer Query is completely dependent on Inner Query.



③ When or why we are use subqueries?

⊕ Case 1:

whenever we have unknown present in the Query to find that unknown we use subqueries

⊕ Case 2:-

whenever we have the data to be selected and the condition to be executed or present in the different table we use subquery.

* SUBQUERY CASE I :-

Whenever we have unknown present in the Query or a ~~Query~~ to find the unknown we use subquery.

Example :-

① WAQTD Name and Salary of the Employees who are getting salary more than MILLER:

Indirect Condition

→ Emp :-

ENAME	SAL
SMITH	1000
ALLEN	2000
SCOTT	3000
MILLER	4000
KING	5000

③ Select Ename, Sal

① from Emp

② where sal > (select sal

Sal > 4000

③

①

②

③

④

⑤

⑥

⑦

⑧

⑨

⑩

⑪

⑫

⑬

⑭

⑮

⑯

⑰

⑱

⑲

⑳

㉑

㉒

㉓

㉔

② WAQTD Name and Salary of the Employees who are getting salary less than ALLEN.

→ Select Ename, Sal

from Emp

where sal < (select sal

from Emp

where Ename = "ALLEN");

③ WAQTD Name and Hiredate of the Employees who ~~hired~~ hired after 1980.

→ select Ename, hiredate

from Emp

where hiredate > '31-DEC-1980';

④ WAQTD Name and Hiredate of the Employees who hired after SCOTT.

→ select Ename, Hiredate

from Emp

where Hiredate > (select hiredate

from Emp

where Ename = 'SCOTT');

Q) WQOTD Name and department no of the employees who are working in the same department as Adams.

```
→ select Ename, deptno  
from Emp  
where deptno = (select deptno  
from Emp  
where Ename = 'ADAMS');
```

Q) WQOTD Name and commission of the employees who are getting commission more than TURNER.

```
→ select Ename, Comm  
from Emp  
where Comm > (select Comm  
from Emp  
where Ename = 'TURNER');
```

Q) WQOTD details of the employees who are getting sal > 1000 AND less than KING.

```
→ select *  
from Emp  
where sal > 1000 AND sal < (select sal  
from Emp  
where Ename = 'KING');
```


* ASSIGNMENT ON SUBQUERY CASE I *

Q1) WQTD Names of the Employees Earning more than ADAMS.

```
→ select Ename  
   from Emp  
  where sal > (select sal  
               from Emp  
               where Ename = 'ADAMS');
```

Q2) WQTD Name and Salary of the Employees Earning less than King.

```
→ select Ename, sal  
   from Emp  
  where sal < (select sal  
               from Emp  
               where Ename = 'KING');
```

Q3) WQTD Name and deptno of the Employees if they are working in the same dept as Jones.

```
→ select depname, deptno  
   from Emp  
  where deptno = (select deptno  
                  from Emp  
                  where Ename = 'JONES');
```

Q4) WQTD Name and Job of all the Employees working in the same designation as James.

```
→ select name, job  
   from Emp  
  where job = (select job  
               from Emp  
               where Ename = 'James');
```

Q6) WAQTD Empno and Ename along with annual salary of all the Employees if their annual salary is greater than WARD's annual salary.

```
→ select Empno, Ename, sal*12
   from Emp
  where sal*12 > (select sal*12
                  from Emp
                 where ename='WARD');
```

Q7) WAQTD Name and hiredate of the Employees if they hired before SCOTT.

```
→ select Ename, hiredate
   from Emp
  where hiredate < (select hiredate
                    from Emp
                   where ename='SCOTT');
```

Q8) WAQTD Name, hiredate of the Employees if they hired after president.

```
→ select Ename, hiredate
   from Emp
  where hiredate > (select hiredate
                    from Emp
                   where job='PRESIDENT');
```

Q9) WAQTD Name, sal of the Employee if they are earning sal less than Employee whose Empno is 7839.

```
→ select Ename, sal
   from Emp
  where sal < (select sal
                from Emp
               where Empno=7839);
```

Q9) WQTD all the employees If the employees are Hired before miller.

```
→ select *  
  from emp  
  where hiredate < (select hiredate  
                    from emp  
                    where ename = 'MILLER');
```

Q10) WQTD Ename and Empno of the employees If employees are earning more than ALLEN.

```
→ select ename, empno  
  from emp  
  where sal > (select sal  
               from emp  
               where ename = 'ALLEN');
```

Q11) WQTD Ename & sal of all employees who are earning more than miller but less than miller.

```
→ select ename, sal  
  from emp  
  where sal > (select sal  
               from emp  
               where ename = 'MILLER') AND sal < (select sal  
                                                    from emp  
                                                    where ename = 'ALLEN');
```

Q12) WQTD all details of employees working in Dept 20 & working in same designation as smith.

```
→ select *  
  from emp  
  where deptno = 20 and job = (select job  
                              from emp  
                              where ename = 'SMITH');
```

Q13) WQTD all the details of employees working as manager ⁱⁿ same dept as TURNER.

```
→ select *  
  from emp  
  where job = 'MANAGER' AND deptno = (select deptno  
                                       from emp  
                                       where ename = 'TURNER');
```


Q14) WAQTD Name, hiredate of Employees hired after 1980 & before King.
→ Select Ename, hiredate
from Emp
where hiredate > '31-DEC-1980' AND Hiredate < (select hiredate
from Emp
where Ename = 'KING');

Q15) WAQTD Name, sal along with annua-sal for all employees whose sal < Blake and > 3500.
→ Select Ename, sal, sal * 12
from Emp
where sal > 3500 and sal < (select sal
from Emp
where Ename = 'BLAKE');

Q16) WAQTD Name of Emp whose name starts with A and works in same dept as BLAKE.
→ Select Ename
from Emp
where Ename like 'A%' AND DEPTNO = (select deptno
from Emp
where Ename = 'BLAKE');

Q17) WAQTD Name of Emp ~~whose~~ and comm if Emp earns comm and working in same designation as Smith.
→ Select Ename, comm
from Emp
where comm IS NOT NULL AND JOB = (select JOB
from Emp
where Ename = 'SMITH');

Q18) WAQTD all details of Emp who earns > SCOTT but less than King.
→ select *
from Emp
where sal > (select sal from Emp
where Ename = 'SCOTT') and sal < (select sal from Emp
where Ename = 'KING');

Q19) WAQTD details of Emp working as clerk in same dept TURNER.
→ select *
from Emp
where JOB = 'CLERK' AND DEPTNO = (select deptno from Emp where Ename = 'TURNER');

Q20) WAQTD Ename, sal, designation of Emp whose annual sal > Smith & less than King.
→ Select Ename, sal, JOB from Emp
where sal * 12 > (select sal * 12
from Emp
where Ename = 'SMITH') AND ~~select~~ sal < (select sal
from Emp
where Ename = 'KING');

Q) WAQTD details of all the Employees

→ select *
from Emp;

Q) WAQTD Name and Salary of the Employees who are getting salary more than 2000.

→ select Ename, sal
from Emp
where sal > 2000;

Q) WAQTD details of the Employees who are getting salary more than BLAKE

→ select *
from Emp
where sal > (select sal
from Emp
where Ename = 'BLAKE');

Q) WAQTD Employee Name, Job, Department No of the Employees who are getting salary less than ALLEN AND more than SCOTT & the Employees who hired after 1980 but before King and getting commission more than TURNER and the Employees who are working in the same department No as 'MILLER'.

→ select Ename, job, Deptno
from Emp
where sal < (select sal
from Emp
where Ename = 'ALLEN') and sal > (select sal
from Emp
where Ename = 'SCOTT')

and hiredate > '31-DEC-1980' and hiredate < (select hiredate
from Emp

where < (select hiredate
from Emp

where Ename = 'KING') AND comm > (select comm from Emp
where Ename = 'TURNER')

Q) WAQTD details of employees whose name is MILLER.

→ Select *
From Emp
Where Ename = 'MILLER';

Q) WAQTD deptno of the miller.

→ Select deptno
From Emp
Where Ename = 'MILLER';

Q) WAQTD deptname of the employee whose name is MILLER.

→ Select dept
From Emp
Where Ename = 'MILLER';

* SUBQUERY CASE II :-

① Whenever we have the data to be selected and condition to be executed or present in the different table then we use subquery.

* ex :- WAQTD DNAME of the Employee whose name is Miller.

EMPNO ENAME SAL DEPTNO

1	SMITH	1000	20
2	SCOTT	2000	10
3	ALLEN	3000	20
4	KING	4000	30
5	MILLER	5000	10
6	BLAKE	6000	20

DNO DNAME LOC

10	D1	L1
20	D2	L2
30	D3	L3

SELECT DNAME
FROM DEPT
WHERE DEPTNO = (SELECT DEPTNO
FROM EMP
WHERE ENAME = 'MILLER');

DeptNo = 10

③
①
②

② WAQTD Location of ALLEN.

→ select loc
from dept
where deptno = (select deptno
from emp
where ename = 'ALLEN');

Q) WAQTD Name of the employees who are working in the location Chicago.

```
→ select Ename  
   from Emp  
  where deptno = (select deptno  
                  from dept  
                  where loc = 'Chicago');
```

Q) WAQTD Name of the employee along with sal who are working in Sales department.

```
→ select Ename, sal  
   from Emp  
  where deptno = (select deptno  
                  from Emp dept  
                  where dname = 'SALES');
```

Q) WAQTD Name of the employees who are getting salary more than MILLER.

```
→ select Ename  
   from Emp  
  where sal > (select sal  
                from Emp  
                where Ename = 'MILLER');
```

Q) WAQTD dname of the employees who are getting salary more than MILLER.

```
→ select dname  
   from dept  
  where deptno = (select dname  
                    from Emp  
                    where sal > (select sal  
                                  from Emp  
                                  where Ename = 'MILLER'))
```


Q) WAP TO Location of the Employees who hired after ALLEN.

→ select loc
from dept

where deptno = (select deptno
from emp

where ~~ename = ALLEN~~ Hiredate > (select hiredate
from emp
where ename =
'ALLEN');

* ASSIGNMENT *

1

Q) WAP TO Dname of the Employees whose Name is SMITH.

→ select dname

from dept

where deptno = (select deptno
from emp
where ename = 'SMITH');

Q) WAP TO Dname and loc of the Employees whose Ename is king.

→ select dname, loc

from dept

where deptno = (select deptno
from emp
where ename = 'KING');

Q) WAP TO Loc of the Employees whose Employee No is 7902.

→ select loc

from dept

where deptno = (select deptno
from emp
where empno = 7902);

④ WAQTD Dname & loc along with Deptno of the Employee whose Name Ends with R.

```
→ select dname, loc, DeptNo  
from emp dept  
where deptno = (select deptno  
from emp  
where ename like '%R');
```

⑤ WAQTD Name of the Employee whose designation is president.

```
→ select dname  
from dept  
where deptno = (select deptno  
from emp  
where job = 'PRESIDENT');
```

⑥ WAQTD Names of the Employees working in Accounting department.

```
→ select ename  
from emp  
where deptno = (select deptno  
from dept  
where dname = 'ACCOUNTING');
```

(IN)
Ruo

⑦ WAQTD Ename, sal of Employees who are working in the location CHICAGO.

```
→ select Ename, sal  
from emp  
where deptno = (select deptno  
from dept  
where loc = 'CHICAGO');
```

⑧ WAQTD details of Employees working in Sales.

```
→ select *  
  from Emp  
 where deptno = (select deptno  
                  from Emp DEPT  
                  where dname = 'SALES');
```

@WAQTD details of Employees along with annual sal if Employees are working in 'NEWYORK'.

```
→ select Emp.*, sal * 12  
  from Emp  
 where deptno = (select deptno  
                  from dept  
                  where loc = 'NEW YORK');
```

⑩ WAQTD Names of the Employees working in operations department

```
→ select Ename  
  from Emp  
 where deptno = (select deptno  
                  from dept  
                  where dname = 'OPERATIONS');
```

* ASSIGNMENT ON CASE I & CASE II

Q1) WAQTD Names of the Employees Earning more than SCOTT in Accounting department

```
→ select Ename
   from Emp
  where sal > (select sal
               from Emp
              where Ename = 'SCOTT') AND DEPTNO = (select DEPTNO
                                                    from DEPT
                                                    where DNAME =
                                                    'ACCOUNTING');
```

Q2) WAQTD details of the Employees working as ~~clerk~~ manager in Location CHICAGO.

```
→ select *
   from Emp
  where job = 'MANAGER' AND DEPTNO = (select DEPTNO
                                       from DEPT
                                      where loc = 'CHICAGO');
```

Q3) WAQTD Name and salary of the Employees Earning more than King in the Dept Accounting.

```
→ select Ename, sal
   from Emp
  where sal > (select sal
               from Emp
              where Ename = 'KING') AND DEPTNO = (select DEPTNO
                                                    from DEPT
                                                    where DNAME = 'ACCOUNTING');
```


34) WAQTD details of Employees working as Salesman in the dept Sales.

→ select *

from Emp

where job = 'SALESMAN' AND DEPTNO = (SELECT DEPTNO

FROM DEPT

WHERE DNAME = 'SALES');

35) WAQTD Name, Sal, Job, Hiredate of Employees working in
operations department and Hired Before King.

→ select Ename, Sal, Job, Hiredate

from Emp

where DEPTNO = (select DEPTNO

FROM DEPT

WHERE DNAME = 'OPERATIONS') AND Hiredate < (select hiredate
from Emp

where ~~hiredate~~ Ename = 'KING');

36) display all the Employees whose department Names Ending 'S'.

→ select Ename

from Emp

where deptno IN (select deptno
from dept

where dname like '%S');

37) WAQTD DName of the Employees whose Name has character A init

→ select dname

from dept

where deptno IN (select deptno
from Emp

where Ename like '%A%');

Q8) WAFD Name & Loc of Employees whose salary of RS. 800

→ SELECT DNAME, LOC

FROM DEPT

WHERE DEPTNO = (SELECT DEPTNO

FROM EMP

WHERE SAL=800);

Q9) WAFD Dname of the Employee who Earn Commission.

→ select dname

from dept

where deptno = (select deptno

from emp

WHERE COMM IS NOT NULL);

Q10) WAFD Loc of Employee if they Earn Comm in Dept 40.

→ select LOC

from ~~EMP~~ DEPT

where deptno = (select deptno

from emp,

where COMM IS NOT NULL AND DEPTNO=40);

Q) WAFD details of all the employees.

→ select *
from emp;

Q) WAFD Names of all the employees.

→ select ename
from emp;

Q) WAFD details of the employees who are getting salary more than Scott.

→ select *
from emp
where sal > (select sal
from emp
where ename = 'SCOTT');

Q) WAFD emp Name, salary, deptno of the employees who are working in the location CHICAGO.

→ select ename, sal, deptno
from emp
where deptno = (select deptno
from dept
where loc = 'CHICAGO');

Q) WAFD Names of the employees whose name starts with character A.

→ select ename
from emp
where ename like 'A%';

Q) WAFD deptname of the employees whose name starts with character A.

→ select ~~dept~~ DNAME
from ~~emp~~ DEPT
where DEPTNO ^{IN} (select deptno
from ~~dept~~ emp
where ename like 'A%');

* TYPES OF SUBQUERY:-

① Single Row Subquery

② Multi-Row Subquery

① Single Row Subquery:-

- If the subquery returns exactly one value we called it as single row subquery
- If subquery return exactly one value we can use either (=) Equal to operator or IN operator to compare the values.

- Example:- WAQTQ Ename and deptno of the employees who are working in the same deptno as KING.

→ select ename, deptno

from emp

where deptno = (select deptno

from emp

where ename = 'KING');

② Multi-Row Subquery:-

- If subquery returns more than one value we called it as multi-row subquery
- If subquery returns more than one value we can use IN operator to compare the values.

- * Example:- WAQTQ Deptname of the employees whose Name starts with Character A.

→ select dname

from dept

where deptno IN (select deptno

from emp

where ename like 'A%');

NOTE:-

① It is difficult to identify the subquery returns single row or Multi-row
so Instead of using = (equal to operator) we should use IN operator.

Q) WQTD details of the employees who are getting sal more than 2000.

→ select *
from emp
where sal > 2000;

Q) WQTD Name and Salary of the employees who are getting salary more than BLAKE.

→ select ename, sal
from emp
where sal > (select sal
from emp
where ename = 'BLAKE');

Q) WQTD Name and salary of the employees who are getting salary more than Salesman.

→ select ename, sal
from emp
where sal > (select sal
from emp
where job = 'SALESMAN');

X Error

* SUBQUERY OPERATORS:-

① ALL

② ANY

① ALL

- all operator is a special operator it returns true if all the records are satisfied

- all operator is used along with relational operator

-
WAP TO Name and Sal of the employees who are getting Sal more than salesman

→ select Ename, Sal

from Emp

where Sal > all (select Sal

from Emp

where Job = 'SALESMAN');

② ANY

- any operator is a special operator it returns true if any of the records is satisfied.

- any operator is used along with relational operator.

* Example:-

WAP TO Name & Sal of the employees who are getting Sal more than any of the salesman.

→ select Ename, Sal

from Emp

where Sal > ANY (select Sal

from Emp

where Job = 'SALESMAN');

* ASSIGNMENT ON SUBQUERY *

Q1) WQTD Name of the Employees Earning Salary more than the
→ select Ename

'SALESMAN'.

from Emp

where sal > all (select sal

from Emp

where JOB = 'SALESMAN');

Q2) WQTD details of Employees hired after all the Clerks.

→ select *

from Emp

where hiredate > all (select hiredate

from Emp

where ~~ename~~ JOB = 'CLERK');

Q3) WQTD Name and salary for all employees if they are earning less
than at least a manager.

→ select Ename, sal

from Emp

where sal < ^{ALL} (select sal

from Emp

where JOB = 'MANAGER');

Q4) WQTD Name and hiredate of Employees hired Before all the
Managers.

→ select Ename, hiredate

from Emp

where hiredate < ~~ALL~~ ALL (select hiredate

from Emp

where job = 'MANAGER');

Q) WAQTD Names of Employees Hired after all the managers and
Earning Salary more than all the clerks.

→ select Ename

from Emp

where hiredate > ALL (select hiredate

from Emp

where JOB = 'MANAGER') AND sal > ALL (SELECT SAL

from Emp

where JOB = 'CLERK');

Q) WAQTD details of Employees working as clerk and Hired before
at least salesman.

→ select *

from Emp

where JOB = 'CLERK' AND HIREDATE < ANY (SELECT HIREDATE

from Emp

where job = 'SALESMAN');

Q) WAQTD details of Employees working in Accounting or SALES Dept

→ select *

from Emp

where Deptno IN (select deptno

from Dept

where DNAME IN ('ACCOUNTING', 'SALES'));

Q) WAQTD details of Employees working in NEW YORK or CHICAGO.

→ select *

from Emp

where deptno IN (select deptno

from Dept

where loc IN ('NEW YORK', 'CHICAGO'));

sq) WQTD details of Employees Department Names of Employees with Name Smith, King, Miller.

→ select dname

from dept

where deptno IN (select deptno

from emp

where Ename IN ('SMITH', 'KING', 'MILLER'));

@ WQTD Emp Names if Employees hired after all Employees of Dept 10.

→ select Ename

from emp

where Hiredat^{ALL}e ~~ALL~~ (select hiredate

from emp

where deptno=10);

Q) WQTD details of all the Employees.

→ select *

from emp;

Q) WQTD Names of all the Employees.

→ select Ename

from emp;

Q) WQTD Salary of all the Employees

→ select sal

from emp;

Q) WQTD Name and salary of the Employees.

→ select Ename, sal

from emp;

Q) WQTD Maximum salary of the Employee

→ select max(sal)

from emp;

Q) WQTD Name of the Employee who is getting first maximum sal.

→ ~~select ename~~ max(sal)
~~from emp;~~

select ename

from emp

where

Q) WQTD Name & salary of the Employees who are getting first minimum sal.

→ select ename, sal
from emp
where sal = (select min(sal)
from emp);

Q) WQTD Name & salary of the Employees who are getting second maximum salary

→ select ename, sal
from emp
where sal = (select max(sal)
from emp)

*Nested SubQuery

- A subquery ~~rewritten~~ ~~another~~ inside another subquery is known as nested subquery.
- we can nested upto 255 subqueries.

* Example:-

WAQTD Name and salary of the employees who are getting second maximum salary.

ENAME	SAL
A	1000
B	2000
C	3000
D	2500
E	4000

← 2nd max

← 1st max

SELECT ENAME, SAL
 FROM EMP
 WHERE SAL = (SELECT ³⁰⁰⁰max(sal)
 ^{C/3000}FROM EMP
 WHERE sal < (SELECT ⁴⁰⁰⁰max(sal)
 ^{SAL=3000}FROM EMP))

2 max sal &
1 query to
display Name &
sal.

Q) WAQTD 3rd maximum salary of the employee.

→ select Ename
 from Emp
 where SAL = (select max(sal)
 from emp
 where sal < (select max(sal)
 from emp
 where sal < (select max(sal)
 from emp)))

Select max(sal) ²⁵⁰⁰

from emp

where sal < (select max(sal) ³⁰⁰⁰

from emp

where sal < (select max(sal) ⁴⁰⁰⁰

sal < 4000 from emp));

Q) WQTD Department Name of the employees who are getting first minimum salary.

→ select dname

from dept

where deptno = (select deptno

from emp

where sal = (select min(sal) ¹⁰⁰⁰

from emp));

Q) WQTD location of the employees who are getting second minimum salary.

→ select loc

from dept

where deptno = (select deptno

from emp

where sal = (select min(sal)

from emp

where sal > (select min(sal)
from emp));

2 for relation

2 Queries

for connection

Q) WQTD 10th maximum salary.

→ select max(sal)

from emp

where sal = (select max(sal)

from emp

where sal < (select max(sal)

from emp

where sal < (select max(sal)

from emp

where sal < (select max(sal).

from emp

where sal < (select max(sal)

from emp

where sal < (select max(sal)

from emp

where sal < (select max(sal)

from emp

41) WAQTD Name of the employee earning maximum salary.

→ select Ename

from Emp

where sal = (select max(sal)
from Emp); where sal < (select max(sal)
from Emp);

42) WAQTD name of the employee earning minimum salary.

→ select Ename

from Emp

where sal = (select min(sal)
from Emp);

43) WAQTD Name, Hiredate of employees hired before all the employees (first Emp).

→ select Ename, hiredate

from Emp

where Hiredate = (select ~~max~~^{min}(hiredate)
from Emp);

44) WAQTD Name and hiredate of the employees hired at the last

→ select Ename, hiredate

from Emp

where hiredate = (select max(hiredate)
from Emp);

45) WAQTD name, Comm of employee who earning min comm.

→ select Ename, comm

from Emp

where comm = (select min(comm)
from Emp);

46) WAQTD name, sal, comm of employees earning maximum comm.

→ select Ename, sal, comm

from Emp

where comm = (select max(comm)
from Emp);

47) WAQTD details of employees who has greatest empno

```
→ select *  
  from emp  
 where empno = (select max(empno)  
                from emp);
```

48) WAQTD details of employees having ^{the} least hiredate.

```
→ select *  
  from emp  
 where hiredate = (select min(hiredate)  
                  from emp);
```

49) WAQTD details of employees earning least annual salary.

```
→ select *  
  from emp  
 where sal * 12 = (select min(sal * 12)  
                  from emp);
```

50) WAQTD Name, annual salary of the employees if their Annual salary is more than all salesman.

```
→ select ename, sal * 12  
  from emp  
 where sal * 12 > ALL (select sal * 12  
                      from emp  
                      where job = 'SALESMAN');
```

Q

Q) WAO TD 2nd minimum salary.

→

- Q) WAO TD 5th maximum salary. - 5 Qu

→ select max(sal)

from Emp

where sal < (select max(sal)

from Emp

where sal < (select max(sal)

from Emp

where sal < (select max(sal)

from Emp

where sal < (select max(sal)

from Emp;

Q

select Ename,

from Emp

where comm < (select max(comm)

from Emp

where comm < (select ^{max(comm)} ~~max(sal)~~

from Emp))

where comm < (select max(~~sal~~ comm)

from Emp)) ;

- 3 Qu
- 1

4)

```
select empno
```

```
from emp
```

```
where sal < (select max(sal)
```

```
from emp)
```

```
where sal < (select max(sal)
```

```
from emp));
```

5)

```
select dname
```

```
from dept
```

```
where deptno = (select deptno
```

```
from emp
```

```
where sal < (select max(sal)
```

```
from emp
```

```
where sal < (select max(sal)
```

```
from emp) where sal <
```

```
(select max(sal)
```

```
from emp
```

```
where sal < (select max(sal)
```

```
from emp
```

-2 connect
4th max - 4

6)

```
select *
```

```
from emp
```

```
where hiredate = (select min(hiredate)
```

```
from emp
```

```
where hiredate > (select min(hiredate)
```

```
from emp);
```

- select ename

from emp

where hiredate = (select max(hiredate)

from emp

where sal < (select max(sal)

from emp));

select loc

from dept

where deptno = (select deptno

from emp

where hiredate = (select min(hiredate)

from emp;

=> queries

- 7th min - 7 Queries
+ 1 Query.

Q) WQTD details of all the Employees.

→ select *
from Emp;

Q) WQTD Employee No of all the Employees.

→ select Empno
from Emp;

Q) WQTD Employee No, Employee Name, MGR of all the Employees.

→ select Empno, Ename, mgr
from Emp;

Q) WQTD mgr No of Smith.

→ select mgr
from Emp
where Ename = 'SMITH';

Q) WQTD ^{mgr}~~manager~~ name of Smith.

→ select Ename
from Emp
where Job = (

* Employee and Manager Relationship:-

① WAQTD manager Name of Smith.

→

Emp

EMPNO	ENAME	MGR
1	SMITH → 2	
2 ←	ALLEN	3
3	SCOTT	4
4	MILLER	5
5	KING	

→ select ^{ALLEN} Ename
 from Emp
 where Empno = (select ² mgr
 from Emp
 where Ename = 'SMITH');
Empno = 2

② WAQTD manager Name of ALLEN.

→ select Ename

from Emp.

where Empno = (select mgr
from Emp
where Ename = 'ALLEN');

③ WAQTD managers manager name of Smith.

→ select Ename

from Emp

where Empno = (select ^{ALLEN = 3} ~~name~~ mgr
Empno = 3

from Emp

where Empno = (select ² mgr
Empno = 2

from Emp

where Ename = 'SMITH');

= 3

Q) WQTD Name of the Employees who are reporting to ALLEN.

```
→ select Ename  
   from Emp  
   where mgr = (select Empno  
                from Emp  
                where Ename = 'ALLEN');
```

Q) WQTD Employee Name, Salary, Deptno of the employees who are reporting to SCOTT.

```
→ select Ename, sal, Deptno  
   from Emp  
   where mgr = (select Empno  
                from Emp  
                where Ename = 'SCOTT');
```

*NOTE :-

① To find MANAGER'S Name.

EmpNO = mgr
↓
subquery

② To find Employee Names.

mgr = EmpNO
↓
sub query.

Q) WAP TO Name of the Employees who are reporting to Smith's manager.

→ select ename

from emp

where mgr = (select ^{ALLEN=2} empno

from emp

where empno = (select ² mgr

empno=2)

from emp

where ename = 'SMITH')

MGR with MGR

select ename

from emp

where mgr = (select mgr

from emp

where ename = 'SMITH')

Q) WAQTD Smith reporting manager Name.

```
→ select Ename  
from Emp  
where Empno = ( select mgr  
from Emp  
where Ename = 'SMITH');
```

Q) WAQTD ADAMS manager's manager Name.

```
→ select Ename  
from Emp  
where deptno = ( select deptno  
from Emp  
where Empno = ( select Empno  
from Emp  
where Ename = 'ADAMS');
```

Q3) WAQTD DName of Jones manager.

```
→ select deptDname  
from dept  
where deptno = ( select deptno  
from Emp  
where Empno = ( select mgr  
from Emp  
where Ename = 'JONES'));
```

Q) WAQTD miller manager salary.

```
→ select Sal  
from Emp  
where Empno = ( select mgr  
from Emp  
where Ename = 'MILLER');
```

Q) Loc of Smith's managers, manager

→ Select Loc

from dept

where deptno = (select deptno
from Emp where Empno = (select mgr from
Emp where Empno = (select mgr
from Emp
where Ename = 'Smith'));

select Ename

from Emp

where mgr = (select Empno

from Emp

where Ename = 'BLAKE');

select count(*)

from Emp

where mgr = (select Empno

from Emp

where Ename = 'KING');

Q) WAQTD all the tables present in the SCOTT Database.
show users;
→ select *
from tabs

WAQTD details of all the employees.

→ select *
from emp;

Q) WAQTD Employee name, salary, hiredate, deptno of the employees who are working as Analyst, Manager, in the deptno 10, 20, 30.

→ select ename, hiredate, deptno
from emp

where JOB IN ('ANALYST', 'MANAGER') AND DEPTNO IN (10, 20, 30);

Q) WAQTD Salary of the employees.

→ select sal
from emp;

Q) WAQTD No of employees who are getting salary.

→ select count(*)
from emp
where sal IS NOT NULL;

Q) WAQTD No of employees, maximum salary of the employees who are working as managers in each department.

→ select count(*), max(sal), deptno

from emp

where JOB = 'MANAGER'

Group by Deptno;

Q) WAQTD manager's name of ALLEN.

→ select ename

from emp

where empno = (select mgr

from emp

where ename = 'ALLEN');

Q) WAQTD Employee Name, deptno of the employees who are reporting to king

```
→ select Ename, deptno  
   from emp  
   where mgr = (select Empno  
                from emp  
                where Ename = 'KING');
```

Q) WAQTD Employee Name, managers name, manager's manager Name of the Employee.

→ select

Q) WAQTD details of all the employees.

→ select *
 from emp

Q) WAQTD Names of all the employees.

→ select Ename
 from emp;

Q) WAQTD Department no of Allen

→ select deptno
 from emp
 where Ename = 'ALLEN';

Q) WAQTD department name of Allen.

→ select dname
 from dept
 where deptno = (select deptno
 from emp
 where Ename = 'ALLEN');

Q) WAQTD Employee name and department name of the employees

→ select Ename
 from emp

Joins :-

- It is a process of retrieve or fetch the data from the multiple tables simultaneously is known as Joins.

* Types of Joins :-

we have 5 types in joins in SQL.

① CARTESIAN JOIN / CROSS JOIN

② EQUI JOIN / INNER JOIN

③ OUTER JOIN

└─ ① LEFT OUTER JOIN

└─ ② RIGHT OUTER JOIN

└─ ③ FULL OUTER JOIN

④ NATURAL JOIN

⑤ SELF JOIN

⑥ Cross Join / Cartesian Join :-

- In Cartesian Join the records from table 1 will merge with all the records of table 2.

* Syntax :-

① Oracle :-
SELECT COL-NAME
FROM TABLE1, TABLE2

② ANSI :- American National Standard Institute

SELECT COLNAME
FROM TABLE1 CROSS JOIN TABLE2

Ex: WANT Employee Name & department name of the employees.

Emp ~~Dept~~ Dept

ENAME	DEPTNO	DNAME	DEPTNO
A	10	D1	10
B	20	D2	20
C	30	D3	30

*Oracle:-

SELECT ENAME, DNAME
FROM EMP, DEPT;

*ANSI :-

SELECT ENAME, DNAME
FROM EMP CROSS JOIN DEPT;

ENAME ~~deptno~~ deptno DNAME deptno

A	10	D1	10
A	10	D2	20
A	10	D3	30
B	20	D1	10
B	20	D2	20
B	20	D3	30
C	30	D1	10
C	30	D2	20
C	30	D3	30



ENAME DNAME

A	D1
A	D2
A	D3
B	D1
B	D2
B	D3
C	D1
C	D2
C	D3

* INNER JOIN / EQUI JOIN :-

- It is used to obtain the matching records
- whenever we are using inner join we should write Join Condition
- Join condition :-

* It is a condition on which 2 tables are merge

* Syntax of Join Condition :-

TABLE-N1.COI-NAME = TABLEN2.COI-NAME;

* ORACLE Syntax:

SELECT COI-NAME

FROM TABLE-NAME1, TABLENAME2

WHERE < JOIN-COND >

SELECT COI-NAME

FROM TABLENAME1 INNER JOIN TN2

ON < JOIN-COND >

Ex:- WAQTD Employee Name and department Name of the Employees.

Q) WQTD Employee Name, salary and the location of the Employees.

→ select Ename, Sal, loc

from Emp, dept

where Emp.deptno = dept.deptno;

Q) WQTD Employee Name, salary, department name of the Employees who are getting salary more than 2000.

→ select Ename, Sal, dname

from Emp, dept

where Emp.deptno = dept.deptno and Sal > 2000;

*ASSIGNMENT ON INNER

CLAUSE *

① Name of the Employees and his location of all the Employees

→ select Ename, loc

from Emp, dept

where Emp.deptno = dept.deptno;

② WQTD Dname and salary for all the Employee working in Accounting.

→ select dname, sal

from dept, Emp

where dept.deptno = Emp.deptno and dname = 'ACCOUNTING';

③ WQTD Dname and annual salary for all the Employees whose salary is more than 2340.

→ select dname, sal*12

from dept, Emp

where dept.deptno = Emp.deptno and sal > 2340;

4) WAQTD Ename and dname for all employees having Character 'A' in their DName.
→ select Ename, dname
from Emp, dept
where Emp.deptno = dept.deptno and dname like '%A%';

5) WAQTD Ename and dname for all employees working as Salesman.
→ select Ename, dname
from Emp, dept
where Emp.deptno = dept.deptno and Job = 'SALESMAN';

6) WAQTD Dname, Job for all employees whose Job and dname starts with character 'S'.
→ select dname, job
from Emp, dept
where Emp.deptno = dept.deptno and Job like 'S%' AND DNAME like 'S%';

7) WAQTD Dname and mgr No for employees Reporting to 7839.
→ select dname, mgr
from Emp, dept
where Emp.deptno = dept.deptno and mgr = 7839;

8) WAQTD Dname, hiredate for employees hired after 83 into accounting or research dept.
→ select dname, hiredate
from Emp, dept
where Emp.deptno = dept.deptno AND Hiredate > '31-DEC-1983'
AND DNAME IN ('ACCOUNTING', 'RESEARCH');

9) WAQTD Ename, dname of employees who are getting Commission in dept 10 or 30.
→ select Ename, dname
from Emp, dept
where Emp.deptno = dept.deptno and Comm is NOT NULL AND
~~DEPTNO (10, 30)~~ Emp.deptno (10, 30);

10) Write Dname and Employee for all the employees whose Empno are 7839, 7902 and are working in LOC New York.

→ select dname, empno
from emp, dept

where emp.deptno = dept.deptno and empno in (7839, 7902)
and LOC = 'NEW YORK';

* OUTER JOIN :-

- It is used to obtain the unmatched records

* LEFT OUTER JOIN :-

- It is used to obtain unmatched records of left table along with matching records.

Syntax :-

- ANSI

```
SELECT colName  
FROM TABLE1 LEFT [OUTER] JOIN TABLE2  
ON (JOIN Condition);
```

- ORACLE:

```
SELECT colName  
FROM T1, T2  
WHERE T1.colName = T2.colName (+);
```

Example :-

Wanted Employee Name and department name of the employees even though the employee is not having any department.

ENAME	DEPTNO	DNAME	DEPTNO	ENAME	DEPTNO	DNAME	DEPTNO
A	20	D1	10	A	20	D2	20
B	null	D2	20	C	10	D1	10
C	10	D3	30	B	null	null	null
D	null	D4	40	C	null	null	null

```
Select Ename, dname  
from Emp, dept  
where Emp.deptno = Dept.deptno(+);
```

↓
Result

ENAME	DNAME
A	D2
C	D1
B	null
D	null

Right Outer Join

- It is used to obtain the unmatched records of right table along with matching records.

ANSI:-

```
SELECT col-Name  
from TN1 RIGHT [OUTER] JOIN TN2  
ON <JOIN-COND>;
```

ORACLE:-

```
SELECT col-name  
from TN1, TN2  
where TN1.col-name (+) = TN2.col-name;
```

Ex:- WATD Employee Name and department name of the employees even though there is no employees in the department.

ENAME	DEPTNO	DNAME	DEPTNO
A	20	D1	10
B	null	D2	20
C	10	D3	30
D	null	D4	40

Emp Dept

ENAME	deptno	dname	deptno
A	20	D2	20
C	10	D1	10
null	null	D3	30
null	null	D4	40

↓ result

ENAME	DNAME
A	D2
C	D1
null	D3
null	D4

```
select Ename, dname  
from Emp, dept  
where Emp.deptno(+) = dept.deptno;
```

* FULL OUTER JOIN :-

It is used to obtain the unmatched records of both left and right table along with matching records.

Syntax :-

```
SELECT COL-NAME
FROM TN1 FULL [OUTER] JOIN TN2
ON <JOIN-COND>;
```

Example :- WAQTD Employee name and department name of the Employees even though the Employees not having any department and No Employees works in Department.

→ Emp

Dept

ENAME	Deptno	Dname	Deptno
A	20	D1	10
B	null	D2	20
C	10	D3	30
D	null	D4	40

ENAME	Deptno	Dname	deptno
A	20	D2	20
C	10	D1	10
B	null	null	null
D	null	null	null
null	null	D3	30
null	null	D4	40



```
Select Ename, dname
from Emp Full join DEPT
on Emp.deptno = dept.deptno;
```

ENAME	dname
A	D2
C	D1
B	null
D	null
null	D3
null	D4