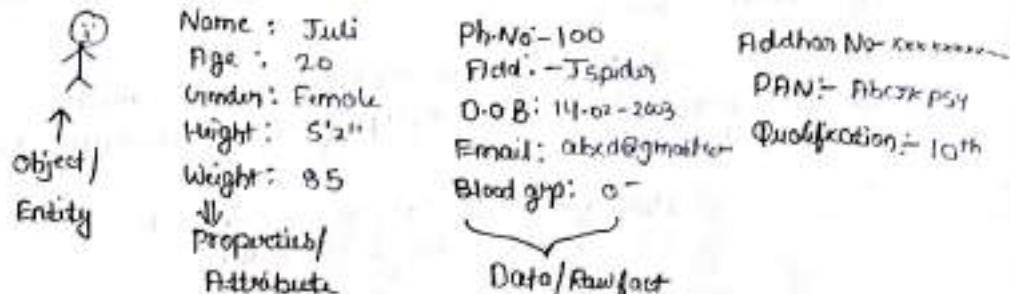


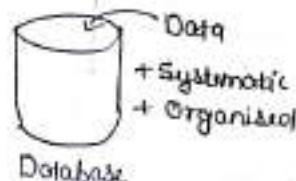
By :-
Jai Shankar Sir

SQL Structured Query Language

- Data:-
- Data is manifest which describes the properties of an object.
 - Properties is also known as Attribute.
 - Object is also known as Entity.



Database:- It is a place or medium in which we store the data in systematic and organised manner.



- In database we are performing one operation called CRUD operation

C	R	U	D
Create / Insert	READ / RETRIEVE	UPDATE / MODIFY	DELETE / DROP

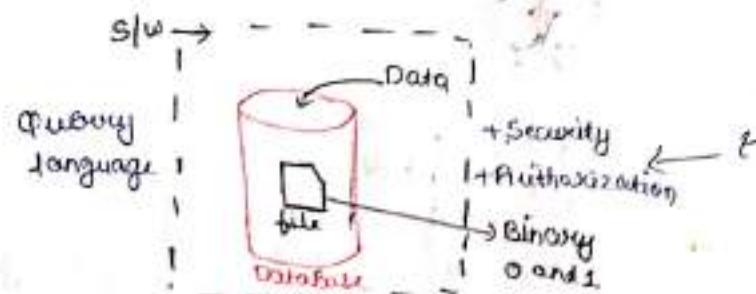
DBMS (Data Base Management System) -

- It is a software which is used to maintain and manage the Database.
- Security and Autorisation are the two main feature provided by DBMS software.
- In the DBMS software we store the data in file format.
- To communicate with DBMS software we use query language.

bit.ly/zoSoft WIN

↓
Software
to access query

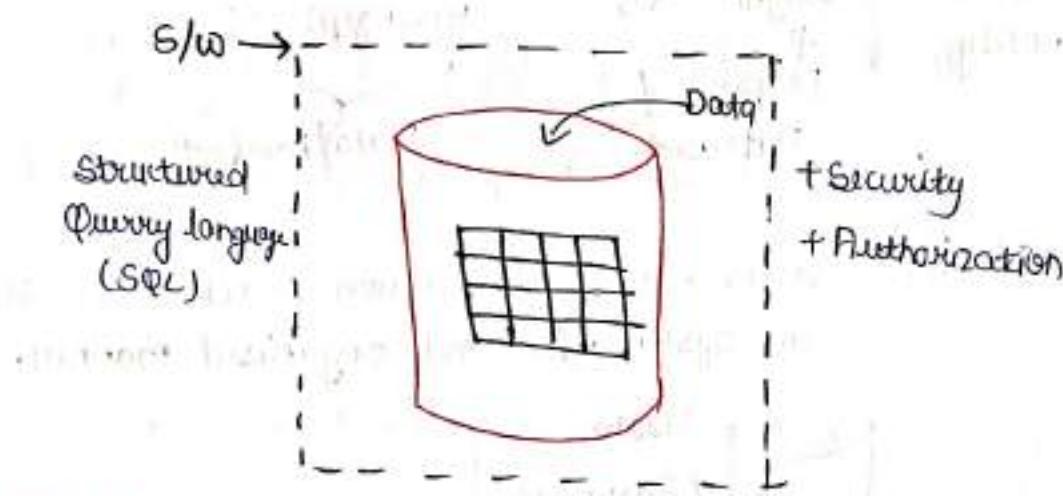
SQL
Structured Query Language



- * Drawbacks of DBMS →
 - Time consuming
 - Memory usage large.

RDBMS (Relational Database Management System)

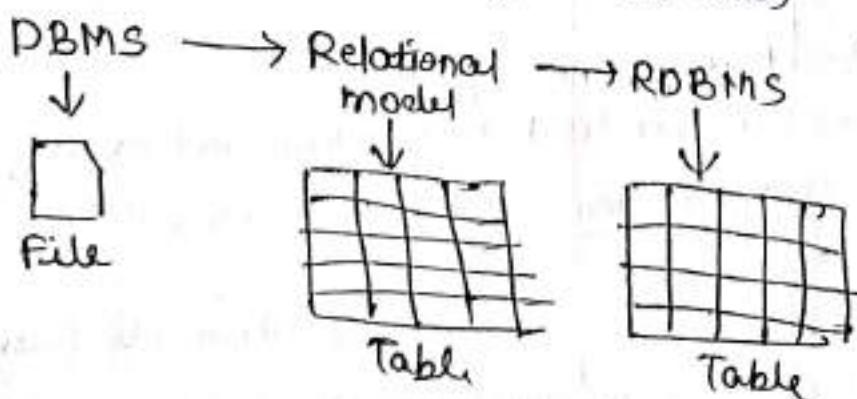
- RDBMS software is type of DBMS software which is used to maintain and manage the Database
- Security and Authorization are the two main features provided by RDBMS software
- In RDBMS software we store the Data in Table format
- We use Structured Query language (SQL) to communicate with RDBMS software



Relational Model:-

- In Relational model we store the data in the form of table.
- Any DBMS which follows relational model it will become RDBMS

Note:- Relational model was designed by one of the Data scientist called E.F.Codd
(Edgar Frank Codd)



→ Table → It is a combination of rows and columns.

Table consist of three types -

- (1) Column
- (2) Row
- (3) cell

(1) Column:

→ Vertical position of the table is known as column.

→ Column is also known as Attributes or fields

(2) Row:

→ Horizontal position of the table is known as Row.

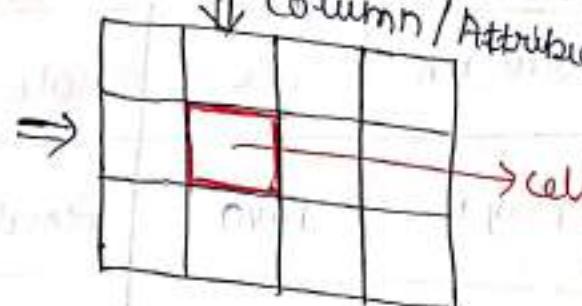
→ Row is also known as Records or Tuples

(3) Cell:

→ The smallest unit of the table is known as cell
or

→ Intersection of Rows and column is known as cell.

Row/
Record/
Tuple



Rules of E.F. CODD

- 1) The data enters in to the cell must be single value
- 2) According to E.F.CODD Rules we can store this data in multiple Tables if needed we can establishing the connection between the two tables with the help of key Attributes.

3) In RDBMS we store the data in the form of table, it includes meta data

Meta data :-

- Details about the data we called it as meta data
- Meta data's are auto generated.

4) The data can be validated in the two steps:-

- 1) By assigning data types
- 2) By assigning constraints.

Notes:-

- Data type are mandatory and constraints are optional.

Meta data



Name, Id, location, size, etc are data

Student

Photo	Student ID	Student Name	Course 1	Gender	Phone No.
	1	Saurabh	SQL	Male	9820451231
	2	Pooja	Java	Female	9342043211
	3	Gaurav	WebTech	Male	-----
	4	Karishma	SQL	Female	-----
	5	SAMEER	Python	Male	-----
	6	Raju	SQL	Male	-----

Datatype:- It is a type of data which is used to store in a particular memory location

Types of datatypes in SQL

- * CHAR
- * VARCHAR / VARCHAR2
- * DATE
- * NUMBER
- * LARGE OBJECT
 - CHARACTER LARGE OBJECT
 - BINARY LARGE OBJECT

- # CHAR:-
- CHAR datatype can accept capital A to Z small a to z , numbers from 0-9 , special character and Alpha Numerics → CHAR + NUM
 - Anything enclosed within single quotes we call it is char datatype.

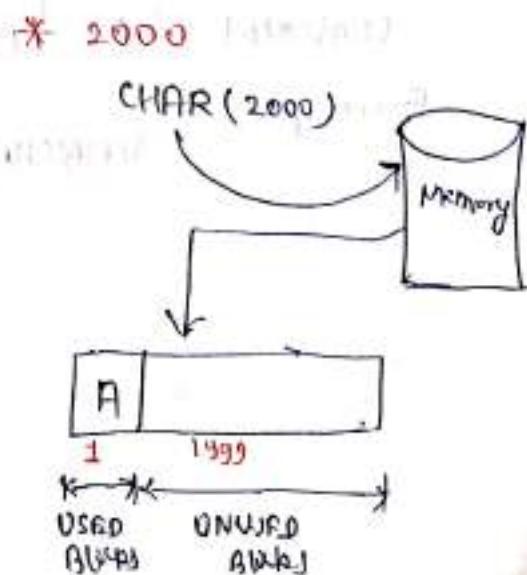
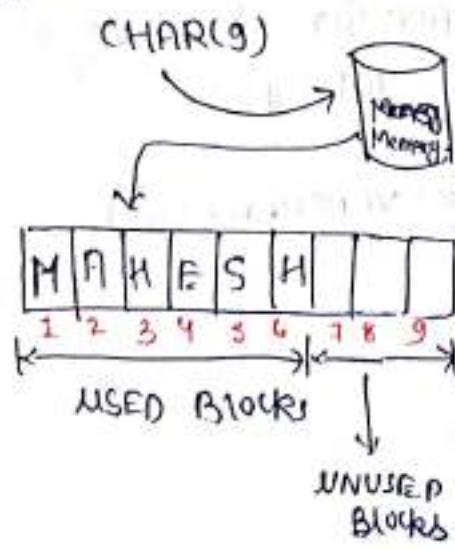
Syntax:-

CHAR(SIZE)

Example:- CHAR(9)

- Whenever we are using char datatype we should mention the size.
- Maximum size of CHAR datatype is '2000' character.
- char datatype follows fixed length memory allocation.

Example:-



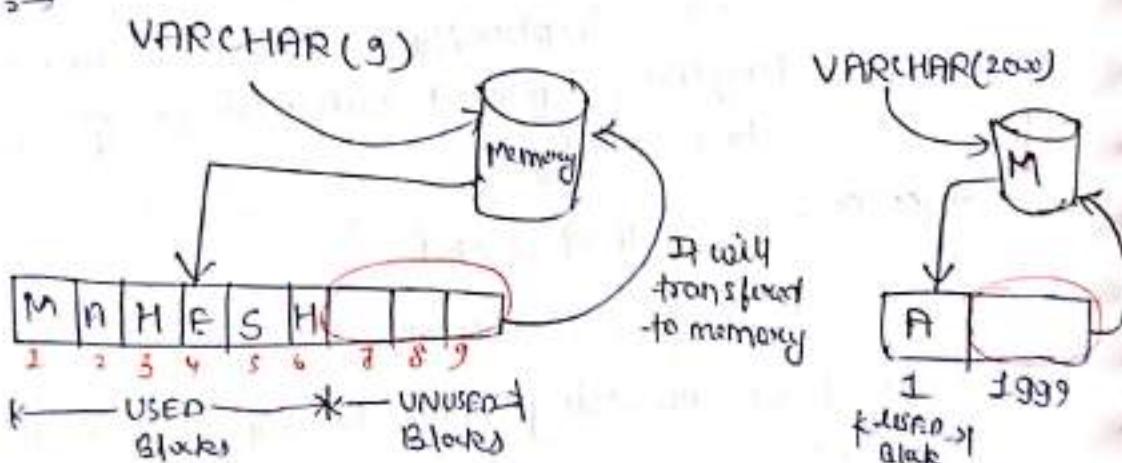
VARCHAR:

- VARCHAR datatype can accept capital 'A-to-Z', small 'a-to-z', numbers from '0-9', 'special characters' as well as 'Alpha Numerics'.
- Anything enclosed in single quotes we call it as VARCHAR datatype.

Syntax:→ VARCHAR(SIZE)

- Whenever we are using varchar datatype we should mention the size.
- The maximum size of varchar datatype is '2000' characters.
- VARCHAR datatype follows variable length memory allocation.

Example:→



VARCHAR2:

- VARCHAR2 datatype is the updation of VARCHAR datatype.
- The maximum size of VARCHAR2 datatype is '4000' characters.
- Whenever we are using VARCHAR datatype it will automatically converted into VARCHAR2 datatype.

Example:→

$$\text{VARCHAR}(30) \Rightarrow \text{VARCHAR2}(30)$$

DATE :-

- Date datatype is used to store the date in a particular format by oracle.

Syntax:- DATE

'DD-MON-YYYY' (or) 'DD-MON-yy'

'14-FEB-2024' (or) '14-Nov-25'

NUMBER :-

- It is used to store the numeric values.

Syntax:- NUMBER (PRECISION, SCALE)

PRECISION:- ◦ 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 value within the precision.
◦ SCALE is not mandatory.
◦ The default value of scale is zero.

Example:-

$$\text{NUMBER}(6,5) = \pm 4.36969 \\ P>S$$

$$\text{NUMBER}(7,3) = \pm 3.015248 \\ P>S$$

$$\text{NUMBER}(5,7) = \pm 0.069145 \\ P<S$$

$$\text{NUMBER}(4,6) = \pm 0.01357 \\ P<S$$

$$\text{NUMBER}(5,5) = \pm 2.6162 \\ P=S$$

$$\text{NUMBER}(4) = \pm 8.018 \\ P$$

Blocks are
created by
using the
greater value
of P or S

LARGE OBJECT :-

1) CHARACTER LARGE OBJECT:- It is used to store characters upto ~~upto~~ 4GB of size.

Syntax :- CLOB

{CLOB}

- 2) Binary Large Object :- It is used to store binary values of images, mp3, mp4, documents and etc. upto 4GB of size.

Syntax :- BLOB {BLOB}

CONSTRAINTS :-

- It is a rule given to the column for validation.

Types of Constraints :-

- * UNIQUE
- * NOT NULL
- * CHECK
- * PRIMARY KEY
- * FOREIGN KEY.

* UNIQUE :- UNIQUE is a constraint which cannot accept duplicated or repeated values.

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

* NULL :- It means empty.

* CHECK :- It is a constraint which is extra validation assigned to the column with some condition, if the condition is satisfied it will return true.

* PRIMARY KEY :- It is a constraint which is uniquely identify the records.

* Characteristics of primary key :-

- It cannot accept duplicated or repeated values.
- It cannot accept null values.
- Primary key is always a combination of unique and not null constraints.
- The table can accept only one column as a primary key.

- Primary key is not mandatory but is highly recommended
- * **FOREIGN KEY** :- It is a constraint which is used to establish the connection b/w two tables.

* Characteristics of foreign key :-

- It can accept duplicated or repeated values.
- It can accept NULL values.
- Foreign key is not a combination of unique and Not Null constraints.
- The table can accept more than one column as a foreign key.
- If a column has to become a foreign key it is mandatory to be primary key on its own table.
- It is present in child table but it belongs to parent table.
- IMP** ◦ It is also known as referential integrity constraints

REFERENTIAL INTEGRITY

EMP Table

Primary Key {NOTNULL} {UNIQUE}

column name → EMPID data type → VARCHAR (10)

CHECK(LENGTH(PHNO)>10)

NOTNULL
UNIQUE

Foreignkey

Foreign
key

DEPT Table

Primary Key

DEPTID

DNAME

LOC

	EMPID	ENAME	SALARY	PHNO.	DEPT NO.
1	MALINI	80	91. --	10	
2	KAVITA	10	24. --	20	
3	SOURABH		84. --	10	
4	VIVEK	50,000	92. --	30	
5	SVYASH	04	722. --	40	

CHILD

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).

DDL

DML

TCL

DCL

DQL

⇒ Data Query Language (DQL) :- It is used to retrieve or fetch the data from the database.

* It has four statements:-

- SELECT
- PROJECTION
- SELECTION
- JOINS

} all will
fetch the data from the table

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.

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

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

Q Write a query to display names of all the employees.

→ PROJECTION : It is used to retrieve or fetch the data from the table by selecting only the column is known as projection.

Syntax :-

SELECT * / [DISTINCT] col-NAME / EXPRESSION [ALIAS]
FROM TABLE-NAME;

ORDER OF EXECUTION :-

FROM → It will execute first
SELECT → It will execute Next

{ } this is used for optional things.
/ either we use * or Col-NAME or Expression }

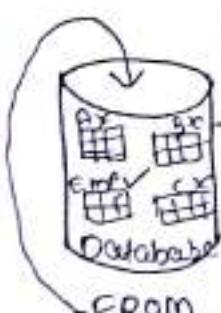
NOTE :-

- FROM clause will execute first
- In FROM clause we write the table-name as an argument
- The job of from clause is to go to the database and search for the table, put the table under execution.
- The job of from clause is completed.
- After the execution of from clause select clause will execute
- The job of select clause is to go to the table which is under execution and search for the column which we have mentioned and display.

Example:- Write a query to display names of all the employees

SELECT ENAME
FROM EMP;

o/p of FROM CLAUSE



Under Execution

EMP			
EMPNO	ENAME	SAL	DEPTNO
1	A	100	10
2	B	200	20
3	C	300	30
4	D	350	40
5	E	400	20
6	F	500	30

RESULT

ENAME
A
B
C
D
E
F

o/p of SELECT clause

Q. Write a query to display salaries of EMP

Ans \Rightarrow SELECT SAL
FROM EMP;

Q. Write a query to display details of all the EMP present in EMP table.

\Rightarrow To display all the details we have to use *

Ans \Rightarrow SELECT *
FROM EMP;

ASTERISK(*) : \Rightarrow Asterisk is used to select all the columns from the table.

SEMICOLON(;) : \Rightarrow It means end of the query.

Q. WAQTD NAME, SALARIES of the Employees.
 \downarrow
write a query to display.

SELECT ENAME, SAL
FROM EMP;

Q. WAQTD ENAME, SALARY, HIRE DATE, and COMMISSION of all the Employees.

SELECT ENAME, SAL, HIRE DATE, COMM
FROM EMP;

OR we can write as

FROM EMP } this will also work.
;

ASSIGNMENT QUESTIONS

1. WAQTD all the details from the EMPLOYEE table.

Ans \Rightarrow SELECT *
FROM EMP;

2. WAQTD NAMES of all the EMPLOYEES.

Ans \Rightarrow SELECT ENAMES
FROM EMP;

3. WAPTD NAME and SALARY given to all the EMPLOYEES.

Ans \Rightarrow SELECT ENAME, SAL
FROM EMP;

4. WAPTD NAME and COMMISSION given to all the EMPLOYEE

Ans \Rightarrow SELECT ENAME, COMM
FROM EMP;

5. WAPTD EMPLOYEE ID and DEPARTMENT NUMBER of all
the employees in EMP table.

Ans \Rightarrow SELECT EMPNO,
~~ENO~~, DEPTNO
FROM EMP;

6. WAPTD ENAME and HIREDATE of all the employees.

Ans \Rightarrow SELECT ENAME, HIREDATE
FROM EMP;

7. WAPTD NAME and DESIGNATION of all the employees.

Ans \Rightarrow SELECT ENAME, JOB
FROM EMP;

8. WAPTD NAME, JOB and SALARY given all the employee.

Ans \Rightarrow SELECT ENAME, JOB, SAL
FROM EMP;

9. WAPTD DNames present in Department table.

Ans \Rightarrow SELECT DEPTNAME, DNAME
FROM DEPT;

10. WAPTD DNAME and LOCATION present in DEPT table.

Ans \Rightarrow SELECT DNAME, LOC
FROM DEPT;

Q. What are all the details of employees.

⇒ SELECT *
FROM EMP;

Q. What are department no. of employees without repetition

SELECT DISTINCT DEPTNO.
FROM EMP;

DISTINCT CLAUSE :-

- DISTINCT clause is used to remove the duplicated or repeated values from the result table.
- Always we use DISTINCT clause as the first argument inside the Select clause.

Example:- What are DEPTNO of employees without repetition.

EMP	ENAME	DEPTNO
A	10	
B	20	
C	30	
A	10	
B	20	
C	30	

SELECT DISTINCT DEPTNO
FROM EMP;

DEPTNO
10
20
30

Q. What are

NAME and DEPTNO of employees without repetition.

SELECT DISTINCT ENAME, DEPTNO
FROM EMP;

Result

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

Note:- DISTINCT is optional

O means
repeated in
same column
so it will not
used in result
table

Q. What are ENAME, SALARY, DEPTNO of the employees without repetition.

SELECT DISTINCT ENAME, DEPTNO, SAL
FROM EMP;

Q WAP TO SALARY of employees.

```
SELECT SAL  
FROM EMP;
```

Q WAP TO Annual SALARY of employees.

EXPRESSION:- A statement which gives the result is known as expression.

Expression is a combination of operand and operator.

OPERAND:- This are the values that we pass.

OPERATOR:- This are the symbols which perform some operation on the operand.

Example:-

$$\begin{array}{c} \text{OPERANDS} \\ 100 * 12 = 1200 \\ \downarrow \\ \text{operator} \end{array}$$

Q. WAP TO Annual SALARY of employees.
Ans:-

```
SELECT SAL*12  
FROM EMP;
```

Q WAP TO Half year SALARY of employees.

```
⇒ SELECT SAL*6  
FROM EMP;
```

Q WAP TO NAMES of the Employees along with annual SAL.

```
SELECT ENAME, SAL*12  
FROM EMP;
```

Q WAP TO DETAILS of the employees along with annual SAL

```
SELECT *, SAL*12  
FROM EMP;
```

```
SELECT EMP.*, SAL*12  
FROM EMP;
```

{ * is used for displaying all the details but if more * are then then we use EMP.* for displaying all details }

Result



2 columns	3rd column
all the Details	SAL*12

One column is extra

Q. WHAT IS SAL OF THE EMPLOYEE WITH THE HIKE OF
OF 10%.

⇒ SELECT SAL + SAL * 10
 100
 FROM Emp;

Q. WHAT IS NAME AND SALARY OF THE EMPLOYEE ALONG WITH
SALARY WITH THE HIKE OF 15%.

SELECT FNAME, SAL, SAL + SAL * 15
 100
 FROM Emp;

Q. WHAT IS DETAILS OF THE EMPLOYEE ALONG WITH ANNUAL
SALARY WITH THE HIKE OF 12%.

~~SELECT EMP.* , SAL + SAL * 12 * 12~~
~~100~~
~~FROM Emp;~~
~~monthly SAL * 12 is not possible~~

✓ SELECT EMP.* , SAL * 12 + SAL * 12 * 12
 100
 FROM Emp;

Q. WHAT IS ANNUAL SALARY OF EMPLOYEE WITH THE HIKE OF
25%.

SELECT SAL * 12 + SAL * 12 * 25
 100
 FROM Emp;

ALIAS →

- IT IS THE ALTERNATE NAME GIVEN TO THE COLUMN
- WE CAN GIVE ALIAS NAME WITH OR WITHOUT
 USING AS.
- BY THREE WAYS WE CAN GIVE THE ALIAS NAME

ANNUALSAL

ANNUAL_SAL

"ANNUAL SAL"

Q. WHAT IS ANNUAL SALARY OF EMPLOYEE WITH THE ALIAS NAME
AS ANNUAL SAL

⇒ SELECT SAL * 12 AS "ANNUAL SAL"
 FROM Emp;

OR
SELECT SAL*12 AS ANNUAL_SAL
FROM EMP;

OR
SELECT SAL*12 AS ANNUALSAL
FROM EMP;

ASSIGNMENT QUESTION for EXPRESSION.

Q1 WAPTD NAMES of the EMPLOYEE along with their ANNUAL SALARY
SELECT ENAME, SAL*12 AS ANNUALSAL
FROM EMP;

Q2 WAPTD ENAMES and Job for all the employees with their
Half term SALARY.

SELECT ENAME, JOB,
SAL*6 AS HALFTERMSAL
FROM EMP;

Q3 WAPTD all the details of the employee along with an
ANNUALBONUS OF 2000

SELECT * FROM EMP,*, SAL*12+2000 AS ANNUALBONUS
FROM EMP;

Q.4. WAPTD NAME, SALARY and SALARY with A HIKE OF 10%

SELECT ENAME, SAL, SAL+SAL* $\frac{10}{100}$ AS HIKE10%
FROM EMP;

Q.5 WAPTD NAME AND SALARY with DEDUCTION of 25%.

SELECT ENAME, SAL-SAL* $\frac{25}{100}$ AS DEDUCTION25%
FROM EMP;

Q.6 WAPTD NAME and SALARY with Monthly HIKE of 50

SELECT ENAME, SAL+50 AS HIKE50
FROM EMP;

Q.7. WAPTD NAME and ANNUAL SALARY with DEDUCTION of 10%.

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

Q.8 WAPTD TOTAL SALARY given to each employee (SAL+WMM)

SELECT
FROM

↓
after
25 days

Q9 WHATD details of all the employees along with annual SALARY

```
SELECT EMP#, SAL*12  
FROM EMP;
```

Q10 WHATD NAMES and DESIGNATION along with 100 penalty in salary

```
SELECT ENAMES, JOB, SAL-100  
FROM EMP;
```

Q WHATD all the details of employee.

```
SELECT *  
FROM EMP;
```

Q. WHATD NAMES of all the employees

```
SELECT ENAME  
FROM EMP;
```

Q WHATD Department no. of all the employees

```
SELECT DEPTNO  
FROM EMP;
```

Q WHATD NAME and DEPTNO of all the employees

```
SELECT ENAME, DEPTNO  
FROM EMP;
```

#Q WHATD NAME and DEPTNO of the employees who are working in the DEPTNO 10

```
SELECT ENAME, DEPTNO  
FROM EMP;
```

WHERE DEPTNO=10;

SELECTION :-
It is process of subquery to fetch the data from the table by selecting both row as well as column. is known as selection

SYNTAX:-

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

ORDER OF EXECUTION:-

```
FROM  
↓  
WHERE  
↓  
SELECT
```

* WHERE clause and DEPTNO

NOTE :- • Where clause is used to filter the records.

- WHERE clause will execute after the execution of from clause.
- WHERE clause will execute row by row.
- In WHERE clause we can write the condition.
- We can write multiple conditions in where clause with the help of logical operator.

Example:- Q) What Name and Deptno of the Emp who are working in the DEPTNO DEPTNO 10

Ans1

```
SELECT ENAME, DEPTNO  
FROM EMP  
WHERE DEPTNO=10;
```

EMPNO	ENAME	DEPTNO
A	SAL	10
B	KEL	20
C	2000	30
D	3000	40
E	4000	10
F	5000	20
G	5000	10

EMPNO	ENAME	SAL	DEPTNO
A	SAL	1000	10
E	5000	5000	10



EMPNO	ENAME	SAL	DEPTNO
A	SAL	1000	10
E	5000	5000	10

EMP
alp of FROM clause

condition
DEPTNO=10

skip WHERE clause

Q) WHAT IS NAME AND JOB OF THE EMPLOYEES WHO ARE WORKING AS SALESMAN.

```
SELECT ENAME, JOB  
FROM EMP  
WHERE JOB = 'SALESMAN';
```

NOTE :-

- SQL IS NOT A CASE SENSITIVE LANGUAGE.
- THE RECORDS ARE CASE SENSITIVE.

∴ " SALESMAN " MUST
BE IN SINGLE QUOTES;
BUT IT IS NOT CASE

* Q) TO DISPLAY ALL THE TABLES FROM THE DATABASE.

```
SELECT *  
FROM TAB;
```

Q) WHAT IS NAME OF THE EMP WHO ARE WORKING IN DEPTNO 30

```
SELECT ENAME  
FROM EMP  
WHERE DEPTNO = 30;
```

Q) WHAT IS DETAILS OF THE EMP WHO ARE GETTING SALARY MORE THAN 2000

```
SELECT ENAME, *  
FROM EMP  
WHERE SAL > 2000;
```

Q) WHAT IS DETAILS OF THE EMP WHO HIRED AFTER 1980

```
SELECT *  
FROM EMP  
WHERE HIRE_DATE > '30-DEC-1980';
```

Q) WHAT IS DETAILS OF THE EMP WHO HIRED BEFORE 1982

```
SELECT *  
FROM EMP  
WHERE HIRE_DATE < '01-JAN-1982';
```

ASSIGNMENT QUESTION FOR WHERE CLAUSE

Q) WHAT IS THE ANNUAL SALARY OF THE EMPLOYEE WHOSE NAME IS SMITH

```
SELECT SAL * 12 AS ANNUAL  
FROM EMP  
WHERE ENAME = 'SMITH';
```

Q.2 WAP TO NAME of the employees working as CLERK

SELECT ENAME

FROM EMP

WHERE JOB = 'CLERK';

Q.3 WAP TO NAME SALARY of the employees who are working as SALESMAN.

SELECT SAL

FROM EMP

WHERE JOB = 'SALESMAN';

Q.4 WAP TO Details of the EMP who earns more than 2000

SELECT *

FROM EMP

WHERE SAL > 2000;

Q.5 WAP TO Details of the EMP whose NAMES is JONES

SELECT *

FROM EMP

WHERE ENAME = 'JONES';

Q.6 WAP TO Details of the EMP who was HIRED after 01-JAN-81

SELECT *

FROM EMP

WHERE HIREDATE > '01-JAN-81';

Q.7 WAP TO NAME and SAL along with annual salary if the annual salary is more than 12000

SELECT ENAME, SAL, SAL*12 AS ANNUAL

FROM EMP

WHERE SAL*12 > 12000;

Q.8 WAP TO EMPNO of the Employees who are worked in DEPT30

SELECT EMPNO

FROM EMP

WHERE DEPTNO = 30;

Q.9. WAP TO ENAME and HIREDATE if the are hired before 1981

SELECT ENAME, HIREDATE

FROM EMP

WHERE HIREDATE < '01-JAN-1981';

Q.10 WAP TO details of the emp working as MANAGER.

SELECT *

FROM EMP

WHERE JOB = 'MANAGER';

Q.11 WAPTD NAME and SALARY given to an Employee if employee earns a commission of Rupees 1400

```
SELECT ENAME, SAL  
FROM EMP  
WHERE COMM=1400;
```

Q.12. WAPTD Details of employee Having Comm more than SALARY

```
SELECT *  
FROM EMP  
WHERE COMM > SAL;
```

Q.13 WAPTD EMPNO of EMP HIRED Before the year 87

```
SELECT EMPNO  
FROM EMP  
WHERE HIREDATE < '01-JAN-87';
```

Q.14 WAPTD Details of Emp working as AN ANALYST

```
SELECT *  
FROM EMP  
WHERE JOB='ANALYST';
```

Q.15 WAPTD Details of Emp earning more than 2000 Rupee per month

```
SELECT *  
FROM EMP  
WHERE JSAL>2000;
```

Q. WAPTD all the details of employees.

```
SELECT *  
FROM Emp;
```

Q WAPTD details of the employees who are working as SALESMAN

```
SELECT *  
FROM Emp  
WHERE JOB='SALESMAN';
```

Q WAPTD Details of the employees who are working in the DEPTNO 30

```
SELECT *  
FROM EMP  
WHERE DEPTNO=30;
```

** Q WAPTD Details of the employees who are working as SALESMAN in the DEPTNo 30

```
SELECT *  
FROM EMP  
WHERE JOB='SALESMAN' AND DEPTNO=30;
```

OPERATOR :-

- * ARITHMETIC OPERATOR (+, -, *, /)
- * CONCATENATION OPERATOR (||)
- * COMPARISON OPERATOR (=, != (OP) < >)
- * RELATIONAL OPERATOR (<, >, ≤, ≥)
- * LOGICAL OPERATOR (AND, OR, NOT)
- * SPECIAL OPERATOR (IN, NOT IN, BETWEEN, ING, NOT BETWEEN, IS, ISNOT, LIKE, NOT LIKE)
- * SUB QUERY OPERATOR
↳ (ALL, ANY, EXISTS, NOT EXISTS)

CONCATENATION OPERATOR :-

- o It is used to join the strings.

Ex:- SELECT 'HI'||ENAME
FROM EMP
WHERE JOB='MANAGER';

'HI'||ENAME

HJ JONES
HI BLACK
HI CLARK

or
SELECT 'HI.'||ENAME
FROM EMP
WHERE JOB='MANAGER';

'HI.'||ENAME
HI-JONES
HI-BLACK
HI-CLARK

LOGICAL OPERATOR :-

- o It is used to write the multiple condition inside the where clause
- o We have three types of logical operator
 1. Logical AND
 2. Logical OR
 3. Logical NOT

1. LOGICAL AND OPERATOR

- o AND operator return true if all the condition are satisfied.

Ex:- Q. What details of the employees who are working as SALESMAN in the DEPTNO 30.

SELECT *

FROM EMP

WHERE JOB='SALESMAN' AND DEPTNO=30;

Q. WHAT DETAILS OF THE EMPLOYEES WHO ARE GETTING SALARY MORE THAN 1000 IN THE DEPT NO 10

⇒ SELECT *
FROM EMP
WHERE SAL > 1000 AND DEPTNO = 10;

⇒ If we use SAL > 6000 AND DEPTNO = 10;
then condition is false and it will return no rows selected.

2. OR OPERATOR :-

- OR OPERATOR returns true if any one of the condition is satisfied

Example:- Q. WHAT DETAILS OF THE EMPLOYEES WHO ARE WORKING AS SALESMAN OR IN DEPT NO 30

⇒ SELECT *
FROM EMP
WHERE JOB = 'SALESMAN' OR DEPTNO = 30;

Q. WHAT DETAILS OF THE EMPLOYEES WHO ARE WORKING IN DEPT NO 10 OR 20.

⇒ SELECT *
FROM EMP
WHERE DEPTNO = 10 OR DEPTNO = 20;

3. NOT OPERATOR :-

It is a inverse operator, the selected values will be rejected.

Example:- Q. WHAT NAME AND DEPTNO OF THE EMPLOYEES WHO ARE NOT WORKING IN THE DEPTNO 20

⇒ SELECT * ENAME, DEPTNO
FROM EMP
WHERE DEPTNO != 20;

WHERE DEPTNO <> 20;
OR

WHERE NOT DEPTNO = 20;

!= is
comparison
operator

<> is
comparison
operator

We can use
all the three
conditions.

{ != , <> , NOT }

all are same

ASSIGNMENT QUESTIONS FOR LOGICAL OPERATOR

Q1 WAP TO details of the EMP working as CLERK AND earning less than 1500

⇒ SELECT *
FROM EMP
WHERE JOB='CLERK' AND SAL<1500;

Q2 WAP TO NAME and HIREDATE of the EMP working as MANAGER in DEPTNO

⇒ SELECT & ENAME, HIREDATE
FROM EMP

WHERE JOB='MANAGER' AND DEPTNO=30;

Q3. WAP TO details of the EMP along with ANNUAL SAL if they are working in DEPTNO=30 as SALESMAN and their annual SALARY Has to be greater than 14000

⇒ SELECT EMP.* , SAL*12
FROM EMP

WHERE DEPTNO=30 AND JOB='SALESMAN' AND
SAL*12>14000;

Q4 WAP TO all the details of the EMP working in DEPT 30 OR as ANALYST

⇒ SELECT *
FROM EMP

WHERE DEPTNO=30 OR JOB='ANALYST';

Q5. WAP TO NAMES of the EMP whose salary is less than 1100 AND their designation is CLERK

⇒ SELECT ENAME
FROM EMP

WHERE SAL<1100 AND JOB='CLERK';

Q6. WAP TO NAME and SAL , ANNUALSAL AND DEPTNO If DEPTNO is 20 earning more than 1100 AND ANNUAL SAL exceeds 12000

⇒ SELECT ENAME, SAL, SAL*12, DEPTNO
FROM EMP

WHERE DEPTNO=20 AND SAL>1100 AND SAL*12>12000;

Q7. WAP TO EMPNO and NAMES of the EMP working as MANAGER in DEPTNO=20 .

⇒ SELECT EMPNO, ENAME
FROM EMP

WHERE JOB='MANAGER' AND DEPTNO=20;
WAP

Q.8 WAPTD details of EMP working in DEPT 20 OR 30

```
SELECT *  
FROM EMP  
WHERE DEPTNO=20 OR DEPTNO=30;
```

Q.9. WAPTD details of EMP working as ANALYST in DEPT 10.

```
SELECT *  
FROM EMP  
WHERE JOB='ANALYST' AND DEPTNO=10;
```

Q.10 WAPTD details of EMP working as president with SAL of Rupees 4000

```
SELECT *  
FROM EMP  
WHERE JOB='PRESIDENT' AND SAL=4000;
```

Q.11 WAPTD NAMES and DEPTNO, JOB of EMP working AS CLERK in DEPT 10 OR 20 .

```
SELECT ENAME, DEPTNO, JOB  
FROM EMP
```

WHERE JOB='CLERK' AND (DEPTNO=10 OR DEPTNO=20);

Q.12. WAPTD details of EMP working as CLERK OR MANAGER in DEPT 10

```
SELECT *  
FROM EMP  
WHERE (JOB='CLERK' OR JOB='CLERK' 'MANAGER')  
AND DEPTNO=10;
```

{ : } () is used for the operation to be first executed then internally

Q.13 WAPTD NAMES of EMP working in DEPT 10, 20, 30, 40

```
SELECT ENAME
```

```
FROM EMP  
WHERE DEPTNO=10 OR DEPTNO=20 OR  
DEPTNO=30 OR  
DEPTNO=40;
```

Q.14 WAPTD details of emp with EMPNO 7902, 7839.

```
SELECT *  
FROM EMP  
WHERE EMPNO=7902 OR EMPNO=7839;
```

Q.15 WAPTD details of EMP working as MANAGER or SALESMAN or CLERK

```
SELECT *  
FROM EMP  
WHERE JOB='MANAGER' OR JOB='SALESMAN' OR JOB='CLERK';
```

Q.16 WAPTD NAMES of EMP HIRED After 81 AND Before 87

```
SELECT ENAME  
FROM EMP  
WHERE HIREDATE > '31-DEC-81' AND HIREDATE <  
'01-JAN-87';
```

Q.17 WAPTD details of EMP earning more than 1250 but less than
3000

```
SELECT *  
FROM EMP  
WHERE SAL > 1250 AND SAL < 3000;
```

Q.18 WAPTD NAMES of EMP HIRED after 81 into DEPT10 OR 30

```
SELECT ENAME  
FROM EMP  
WHERE HIREDATE > '31-DEC-81' AND (DEPTNO=10  
OR DEPTNO=30);
```

Q.19. WAPTD NAMES of EMP along with ANNUAL SAL for the
EMP working as MANAGER OR CLERK into DEPT10 OR 30

```
SELECT ENAME, SAL*12  
FROM EMP  
WHERE (JOB='MANAGER' OR JOB='CLERK') AND  
(DEPTNO=10 OR DEPTNO=30);
```

Q.20 WAPTD all the details along with annual sal if SAL is

b/w 1000 and 4000 ANNUAL SAL more than 15000

SELECT * FROM EMP
X WHERE (SAL=1000 AND SAL=4000) AND (SAL*12>15000);

✓ WHERE (SAL>1000 AND SAL<4000) AND SAL*12>15000;

we should not use brackets here because all are AND

* Q. WAPTD Details of the EMP along with annual SAL and the
Employees who are working as 'SALESMAN', MANAGER,
ANALYST, CLERK, PRESIDENT in the DEPTNO 10, 20, 30, 40
AND who are getting SAL more than 1000 AND less than
3000 AND who hired after 1980 but before 1987.

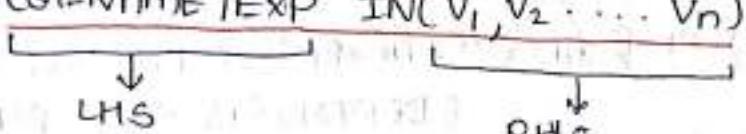
```
SELECT * , SAL*12  
FROM EMP  
WHERE
```

SPECIAL OPERATOR :-

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

* IN OPERATOR :- • It is a multivalue operator it can accept one value at LHS and multiple values at RHS
• We can use IN operator instead of using equal operator

SYNTAX :- Col-NAME / EXP IN(V₁, V₂ . . . V_n);



Example :- Q. Write a query to display NAME and DEPTNO of employees who are working in the DEPTNO 10, 20.

```
SELECT ENAME, DEPTNO  
FROM EMP  
WHERE DEPTNO IN(10,20);
```

Q. Write a query to display details of the emp who are working as Salesman, Manager, in the DEPTNO 30

```
SELECT *  
FROM EMP  
WHERE JOB IN ('SALESMAN', 'MANAGER') AND  
DEPTNO = 30;
```

Q. WAPTD NAME and SAL of the EMP who are working
SAL more than 1000 and working in the DEPTNO 10,
20, 30

```
SELECT ENAME, SAL  
FROM EMP
```

WHERE SAL > 1000 AND DEPTNO IN (10, 20, 30);

Q. WAPTD details of the EMP who are working in the
department No 20

```
SELECT *  
FROM EMP  
WHERE DEPTNO = 20;
```

```
OR  
DEPTNO IN 20;  
OR  
DEPTNO &IN (20);
```

} all are same

* NOT IN OPERATOR :-

- It is similar to IN operator instead of selected values will be rejected.

SYNTAX:- col-NAME | EXP NOTIN (V₁, V₂, ..., V_n)

Example:- Q WAPTD NAME and DEPTNO of the EMP
who are NOT working in the DEPTNO 10, 20.

```
SELECT ENAME, DEPTNO  
FROM EMP  
WHERE DEPTNO NOTIN (10, 20);
```

Q. WAPTD details of the EMP who are not working as
SALESMAN, MANAGER

```
SELECT *  
FROM EMP  
WHERE JOB NOTIN ('SALESMAN', 'MANAGER');
```

Q WAPTD details of the emp who are working as SALESMAN,
MANAGER, ANALYST, CLERK in the deptNo 10, 20, 30, and
getting salary more than 2000 less than 4000

```
SELECT *  
FROM EMP  
WHERE JOB IN('SALESMAN', 'MANAGER', 'ANALYST', 'CLERK')  
AND DEPTNO IN(10, 20, 30) AND SAL > 2000 AND  
SAL < 4000;
```

Q. WAQTID SAL of the emp who are getting salary more than 1000 and less than 4000

⇒ SELECT * SAL
FROM EMP
WHERE SAL > 1000 AND SAL < 4000;

* BETWEEN OPERATOR :- It is used whenever we have range of values.

Range between LOWER RANGE to HIGHER RANGE

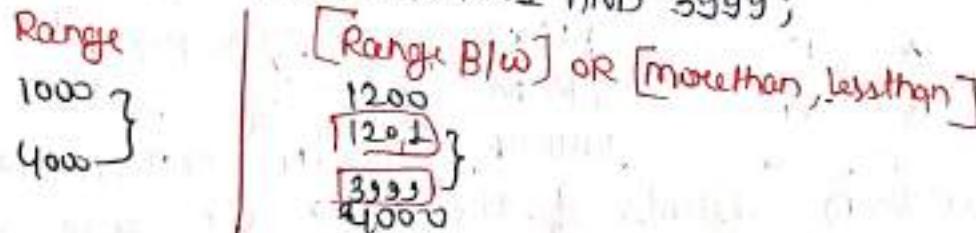
SYNTAX :- Col.NAME / EXP BETWEEN LOWER RANGE AND HIGHER RANGE;

Example:- Q. WAQTID NAME and SAL of the employees who are getting SAL in the Range of 1000 to 3000

⇒ SELECT ENAME, SAL
FROM EMP
WHERE SAL BETWEEN 1000 AND 3000;

Q. WAQTID NAME and SAL of the Emp who are getting SAL in the Range BETWEEN 1200 to 4000

⇒ SELECT ENAME, SAL
FROM EMP
X WHERE SAL BETWEEN 1200 AND 4000;
✓ WHERE SAL BETWEEN 1201 AND 3999;



Q. WAQTID Name ,SAL who are getting salary more than 1500 but less than 2500;

⇒ SELECT ENAME, SAL
FROM EMP

WHERE ~~SAL > 1500~~ SAL BETWEEN 1501 AND 2499;

Q. WAQTID details of the Emp who, HIREDATE in the Range of 1981 to 1987 .

SELECT *
FROM EMP

WHERE HIREDATE BETWEEN '01-JAN-81' AND
'31-DEC-87';

Q. WAPTD Details of the EMP who HIRED after 1981, before 1987.

```
→ SELECT *  
      FROM EMP  
     WHERE HIREDATE BETWEEN '01-JAN-82' AND '31-DEC-86';
```

Q. WAPTD details of the Emp

```
SELECT *  
  FROM EMP;
```

Q. WAPTD Name and HIREDATE of the EMP who are working as 'SALESMAN' in the deptno 10, 20, 30, and, getting salary in the range b/w 1000 to 4000

```
SELECT ENAME, HIREDATE  
  FROM EMP  
 WHERE JOB = 'SALESMAN' AND DEPTNO IN (10, 20, 30)  
       AND SAL BETWEEN 1000 AND 4000;
```

* NOT BETWEEN OPERATOR :-

- Not between operator is similar to between operator instead of selected values will be rejected

SYNTAX:- col-NAME / EXP NOT BETWEEN LOWER RANGE AND HIGHER RANGE;

Example:- Q. WAPTD NAME and SAL of the Emp who are not getting SAL in the range of 1000 to 3000

```
SELECT ENAME, SAL  
  FROM EMP  
 WHERE SAL NOT BETWEEN 1000 AND 3000;
```

Q. WAPTD NAME and HIREDATE of the Emp those who are not hired after 1981 before 1987

```
SELECT ENAME, SAL, HIREDATE  
  FROM EMP  
 WHERE HIREDATE NOT BETWEEN '01-JAN-82' AND  
       '31-DEC-86';
```

* IS OPERATOR :-

IS operator is used to compare only with null values.

SYNTAX :- col-name/exp IS NULL;

Example:- Q. What NAME AND comm of the employees who are not getting comm.

⇒ SELECT ENAME, Comm

FROM EMP

WHERE Comm IS NULL;

Q. What NAME, MGR (manager) of the employees those who are not having reporting MGR

⇒ SELECT ENAME, MGR

FROM EMP

WHERE MGR IS NULL;

* IS NOT OPERATOR :-

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

SYNTAX :- col-name/exp IS NOT NULL;

Ex:- Q. What NAME and comm of the EMP those who are getting comm.

⇒ SELECT ENAME, Comm

FROM EMP

WHERE Comm IS NOT NULL;

Q. What NAME, MGR, of the EMP those who are having reporting manager.

⇒ SELECT ENAME, MGR

FROM EMP

WHERE MGR IS NOT NULL;

* LIKE OPERATOR :-

LIKE operator is used whenever we have pattern matching

SYNTAX :- col-name/exp ^{LIKE} 'PATTERN-TO-MATCH';

Ex → Q. What NAME of the EMP whose name starts with A

A) SELECT ENAME
FROM EMP

B) WHERE ENAME LIKE 'A' ;

Special char used

(i) % - Percentile
(ii) _ - Underscore

```
SELECT ENAME  
FROM EMP  
WHERE ENAME LIKE 'A%';
```

Q. WHAT IS NAME OF EMP WHOSE NAMES ENDS WITH R.

```
SELECT ENAME
```

```
FROM EMP
```

```
WHERE ENAME LIKE '%R';
```

NOTE :-



% A % is used to display A's in name anywhere.

% AA % is used to display ~~two~~ ~~two~~ consecutive A's

% A % A % is used to display two A's in the name.

NOTE :- To achieve pattern matching we need two special characters
% and _

Q. WHAT IS NAME OF THE EMP THOSE WHOSE NAME CONSIST OF 5 CHARACTERS

```
SELECT ENAME
```

```
FROM EMP
```

```
WHERE ENAME LIKE '_____'; { : 5 underscores}
```

★ NOT LIKE OPERATOR :-

It is similar ^{to} like operator instead of selected values will be rejected.

SYNTAX :- Col-NAME/EXP NOT LIKE 'PATTERN - TO - MATCH';

Ex:- WHAT IS NAME OF THE EMP WHOSE NAME DOES NOT START WITH A.

```
SELECT ENAME
```

```
FROM EMP
```

```
WHERE ENAME NOT LIKE 'A%';
```

Assignment Questions for special operators

1) LIST all the emp whose commission is NULL

```
SELECT ENAME  
FROM EMP
```

WHERE comm IS NULL;

2) LIST all the emp who don't have a Reporting manager.

```
SELECT ENAME  
FROM EMP
```

WHERE MGR IS NULL;

3) LIST all the SALESMAN IN DEPT 30

```
SELECT ENAME  
FROM EMP
```

WHERE JOB = 'SALESMAN' AND DEPTNO = 30 ;
OR

JOB IN ('SALESMAN') AND DEPTNO IN (30);

4) LIST all the SALESMAN IN DEPT NO 30 and having SAL greater than 1500

```
SELECT ENAME, JOB  
FROM EMP
```

WHERE JOB IN ('SALESMAN') AND DEPTNO IN (30) AND
SAL > 1500;

5) LIST all EMP whose NAME starts with 'S' OR 'A'

```
SELECT ENAME  
FROM EMP
```

WHERE ENAME LIKE 'S%' OR ENAME LIKE 'A%';

6) LIST all the emp except those who are working in DEPT NO 10 & 20

```
SELECT ENAME  
FROM EMP
```

WHERE DEPTNO NOT IN (10, 20);

7) LIST the emp whose NAME does not start with 'S'.

```
SELECT ENAME  
FROM EMP
```

WHERE ENAME NOT LIKE 'S%';

8) LIST all the emp whose name Having Reporting manager in DEPTNO 10

X SELECT ENAME
FROM EMP
WHERE JOB IN ('MANAGER') AND
MGR AND DEPTNO IN (10);

✓ SELECT ENAME
FROM EMP
WHERE MGR IS NOT NULL AND DEPTNO IN (10);

9) LIST all the emp whose comm is NULL and working as CLERK

SELECT ENAME
FROM EMP

WHERE COMM IS NULL AND JOB = 'CLERK';

10) LIST all the emp who don't have a reporting manager in DEPTNO 10 OR 30

SELECT ENAME
FROM EMP

WHERE MGR IS NULL AND DEPTNO IN (10, 30);

11) LIST all the SALESMAN IN DEPT 30 with SAL more than 2450

SELECT ENAME
FROM EMP

WHERE JOB IN ('SALESMAN') AND DEPT NO IN (30)
AND SAL > 2450;

12) LIST all the ANALYST IN DEPT NO 20 and Having SALARY greater than 2500

SELECT ENAME
FROM EMP

WHERE JOB = 'ANALYST' AND DEPT NO = 20 AND
SAL > 2500;

13) LIST all the emp whose name starts with 'm' OR 'j'

SELECT ENAME
FROM EMP

WHERE ENAME LIKE 'm%.' OR ENAME LIKE 'j%.';

14) LIST all the emp with annual SAL except those who are working in DEPTNO 30

SELECT ENAME, SAL*12
FROM EMP

WHERE DEPT NO NOT IN (30);

- 15) LIST the emp whose NAME does not end with 'ES' OR 'R'
SELECT FNAME
FROM EMP
WHERE ENAME NOT LIKE '%ES' OR ENAME NOT LIKE '%R';
- 16) LIST all the emp who are Having supporting manager IN DEPT 10 along with 10% HIKE IN SALARY
SELECT FNAME
FROM EMP
WHERE MGR IS NOT NULL AND DEPT NO IN (10) AND SAL + SAL * $\frac{10}{100}$ AS 10% HIKE;
- 17) DISPLAY all the emp who are 'SALESMAN's Having 'E' AS the last BUT ONE character in ENAME but salary having exactly 4 character. → second last
SELECT FNAME
FROM EMP
WHERE JOB IN 'SALESMAN' AND ENAME LIKE '%_E%'
AND SAL + ~~10%~~ '---';
- 18) DISPLAY all the emp who are joined after year 81.
SELECT FNAME
FROM EMP
WHERE HIREDATE > '01-JAN-81' OR HIREDATE \geq '01-JAN-81';
OR
19) Display all the emp the emp who are joined IN FEB
SELECT ENAME, HIREDATE
FROM EMP
WHERE HIREDATE LIKE '01-FEB-%';
- 20) LIST the emp who are not working as manager and CLERKS in DEPT 10 and 20 with a SALARY in the Range of 1000 To 3000.
SELECT FNAME
FROM EMP
WHERE MGR IS NOT NULL AND JOB NOT IN ('MANAGER', 'CLERK')
AND DEPTNO IN (10, 20) AND SAL BETWEEN 1000 AND 3000;

Q. WAPTD emp name, SAL, HIREDATE, COMM, ANNALSAL, and those who are getting SAL more than 6000 and they are getting SAL in the range of 500 to 3000 they are working IN SALESMAN, MANAGER, ANALYST, CLERK, PRESIDENT in the DEPTNO 10, 20, 30, 40 those who are getting COMM and not having MGR and the ENAME starts with 'A' and ENAME ends with 'B' and those who are hired after 1959 and before 1970 those who are not getting SAL in the range between 3000 to 5000.

Ans \Rightarrow SELECT ENAME, SAL, HIREDATE, COMM, SALX12
FROM EMP
WHERE SALX12 > 6000 AND SAL BETWEEN 500 AND 3000 AND
JOB IN ('SALESMAN', 'MANAGER', 'ANALYST', 'CLERK', 'PRESIDENT')
AND DEPTNO IN (10, 20, 30, 40) AND COMM IS NOT NULL AND
MGR IS NULL AND ENAME LIKE 'A%' AND ENAME LIKE
'%B' AND HIREDATE BETWEEN '01-JAN-59' AND
'31-DEC-69' AND SAL NOT BETWEEN 3001 AND 4999;

Q. WHAT details of the EMP whose name starts with VOWELS

```
SELECT *
  FROM EMP
 WHERE ENAME LIKE 'A%'
   OR ENAME LIKE 'E%'
   OR ENAME LIKE 'I%'
   OR ENAME LIKE 'O%'
   OR ENAME LIKE 'U%' ;
```

Q. WHAT details of the EMP whose name starts with 'J'.

```
SELECT *
```

```
  FROM EMP
```

```
 WHERE ENAME LIKE 'J%' ;
```

Q. WHAT details of the EMP whose name ends with 'R'.

```
SELECT *
```

```
  FROM EMP
```

```
 WHERE ENAME LIKE '%R%' ;
```

* Q. WHAT details of the EMP whose name starts with '_'

```
SELECT *
```

```
  FROM EMP
```

```
 WHERE ENAME LIKE '_%' ;
```

ESCAPE CHARACTER :-

- ESCAPE character is used to remove the special behaviour of a special character and to treat it as a normal character.
- ESCAPE character must be written before the special char which has to be treated as a normal character.
- ESCAPE character must be define after using.
- We can use any character assign as an escape character but recommended characters to scope character are (! , \$, / , \)
- We use the concept of escape characters only to look for '%' and '_'.

SYNTAX:- Col-NAME/EXP LIKE/NOT LIKE 'PATTERN-TO-MATCH'
ESCAPE 'CHAR';

Ex:- JAI_SANKAR.K
DR_JONY%

_RUCHAAT102
% Tom_22_02
_VIRAT % KOHLI 18
% LIFT_WITHADI

Q. WAP TO NAMES that ends with '_'. .

SELECT ENAME
FROM EMP ESC

WHERE ENAME LIKE '% \$ _' ESCAPE '\$';

↓
Special character
↓
Normal character.
↓
ESCAPE character

{ we can use
\$, ! , / , \
any one of them }

Q. WAP TO NAMES that starts with '_'

SELECT ENAME
FROM ESC

WHERE ENAME LIKE '\$ _ %' ESCAPE '\$';
OR

' ! - % ' ESCAPE ' ! ';

Q. WHAT TO NAMES that start and ends with -

SELECT ENAME
FROM ESC

WHERE ENAME LIKE '/_ % /' ESCAPE '/';

↓ ↓ ↓ ↓
EC NC SC EC NC

Q. WHAT NAMES which has % in It.

SELECT ENAME
FROM ESC

WHERE ENAME LIKE '\$ % \$ % %' ESCAPE '\$';

↓ ↓ ↓ ↓
SC EC NC SC

Q. WHAT details of the emp whose name starts with _ and ends with %.

SELECT ENAME, *
FROM ESC

WHERE ENAME LIKE '_ % %' ESCAPE '\$';

Q. WHAT all the details of emp present in employee stable

SELECT *
FROM EMP;

Q. WHAT salary of all the emp

SELECT SAL
FROM EMP;

* Q. WHAT First maximum salary

SELECT max(SAL)
FROM EMP;

P
Multi
or
group
or
rank
SIN
I
W
I
.
Mu
.
L
⇒ L

Function :- Functions are block of code or list of instruction which is used to perform some specific task.

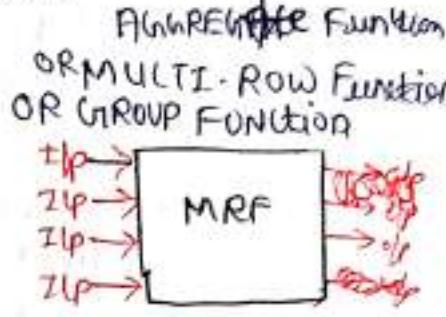
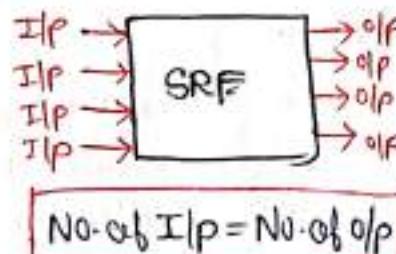
Function

USER DEFINE FUNCTION

Multi Row Function
or
Group function
or
AGGREGATE Function

IN BUILT FUNCTION

SINGLE ROW FUNCTION



SINGLE - ROW FUNCTION :-

- If we give one input to the single row function finally we will get single output
- If we give multiple ips to the single row function finally we will get multiple outputs
- No. of I/p = No. of o/p

After 20 days we will study SRF

MULTI ROW FUNCTION :-

- If we give one input to multi row function finally we will get single output.
- If we give multiple inputs to the multi view function finally we will get single output.
- No. of I/p = Single o/p

⇒ LIST OF MULTI ROW FUNCTION :-

- * MAX()
- * MIN()
- * SUM()
- * AVG()
- * COUNT()

- 1) MAX() :- It is used to obtain maximum value to present in the column.
- 2) MIN() :- It is used to obtain minimum value present in the column.
- 3) SUM() :- It is used to obtain sum of values present in the column.
- 4) AVG() :- It is used to obtain avg. of the values present in the column.
- 5) COUNT() :- It is used to obtain total no. of values present in the column.

NOTE :- Multi Row Function cannot accept null value.

- In SELECT clause we cannot use any other column name along with multirow function.
- We can use more than one multirow function inside the select clause.
- In WHERE clause we cannot use multirow function.
- COUNT is the only multirow function which can accept * as an argument.

Ex :- Q. WHAT IS max salary of emp.

SELECT max(SAL)
FROM Emp;

SAL	
1000	→ MIN
2000	→ AVG
3000	
4000	→ MAX
1000	→ SUM

Q. WHAT IS MINIMUM salary of emp.

SELECT MIN(SAL)
FROM Emp;

Q. WHAT IS Total salary of employees.

SELECT SUM(SAL)
FROM Emp;

Q. WHAT IS AVG salary of employees

SELECT AVG(SAL)
FROM Emp;

Q. WHAT IS No. of employees working in deptno. 10

SELECT COUNT(*)
FROM Emp
WHERE DEPT NO = 10;

OR

SELECT COUNT(DEPT NO)
FROM Emp
WHERE DEPT NO = 10;

col. Name

Q. What is the no. of employees, max, sal, & min(sal), total sal, avg sal of the employees who are working in the depno 30.

```
SELECT MAX(SAL), MIN(SAL), SUM(SAL), COUNT(*),  
AVG(SAL)  
FROM EMP  
WHERE DEPTNO = 30;
```

⇒ we cannot use any other column

```
SELECT ENAME, MAX(SAL) X
```

ASSIGNMENT ON MRF()

1 Q. What is the number of emp getting salary less than 2000 IN DEPTNO 10

```
SELECT COUNT(*)  
FROM EMP  
WHERE SAL < 200 AND DEPTNO = 10;
```

2 Q. What is the total salary needed to pay emp working as clerk.

```
SELECT SUM(SAL)  
FROM EMP  
WHERE JOB = 'CLERK';
```

3 Q. What is the average salary needed to pay all employees.

```
SELECT AVG(SAL)  
FROM EMP;
```

X WHERE

4 Q. What is the number of emp having 'A' as their first character.

```
SELECT COUNT(*)  
FROM EMP  
WHERE ENAME LIKE 'A%';
```

Q.5 What is the number of emp working as CLERK OR MANAGER

```
SELECT COUNT(*)  
FROM EMP  
WHERE JOB IN ('CLERK', 'MANAGER');
```

Q.13

Q.6 What is total salary needed to pay emp hired in FEB

```
SELECT SUM(SAL)
FROM EMP
WHERE HIREDATE LIKE '1% - FEB - %';
```

Q.7. What is number of emp reporting to MGR = 7839

```
SELECT COUNT(X)
FROM EMP
WHERE MGR = 7839;
```

Q.8. What is number of emp getting comm in DEPTNO 30

```
SELECT COUNT(X)
FROM EMP
WHERE COMM IS NOT NULL AND DEPTNO = 30;
```

Q.9 What is avg sal, total sal, number of emps and maximum salary given to employees working as president

```
SELECT AVG(SAL), SUM(SAL), COUNT(X), MAX(SAL)
FROM EMP
WHERE JOB = 'PRESIDENT';
```

Q.10 What is number of emp having 'A' in their names.

```
SELECT COUNT(X)
FROM EMP
WHERE ENAME LIKE '%A%';
```

Q.11 What is Number of emp having AND total salary needed to pay the emp. who have 2 consecutive L's in their names.

```
SELECT COUNT(X), SUM(SAL)
FROM EMP
WHERE ENAME LIKE '%LL%';
```

Q.12 What is number of departments present in emp table.

```
SELECT COUNT(DISTINCT(DEPTNO))
FROM EMP;
```

Q.13

Q.14

Q.15

Q.16

Q.17

Q.18

Q.13 WAPTD Number of Emp Having character 'z' in their Name

```
SELECT COUNT(*)  
FROM Emp  
WHERE ENAME LIKE '%z%';
```

Q.14 WAPTD Number of emp Having '\$' in their Names.

```
SELECT COUNT(*)  
FROM Emp  
WHERE ENAME LIKE '%$%'  
SELECT COUNT(*)  
FROM Emp  
WHERE ENAME LIKE '%$%' ESCAPE '$';
```

Q.15 WAPTD total sal given to emp working as clerks in Dept 30.

```
SELECT sum(SAL)  
From Emp  
WHERE JOB='CLERK' AND DEPTNO=30;
```

Q.16. WAPTD Maximum salary given to the emp working as analyst.

```
SELECT MAX(SAL)  
FROM Emp  
WHERE JOB='ANALYST';
```

Q.17 WAPTD No. of distinct salaries present in emp table

```
SELECT count(DISTINCT SAL)  
From Emp;  
WHERE
```

Q.18 WAPTD No. of Jobs present in emp table

```
SELECT COUNT(DISTINCT JOB SAL)  
From Emp;
```

Q.19 WHAT'S AVG SAL given to the clerk

```
SELECT AVG(SAL)  
FROM Emp
```

WHERE JOB = 'CLERK';

Q.20 WRITE TO minimum salary given to the emp who work
in dept 10 AS manager or a clerk.

```
SELECT MIN(SAL)  
FROM Emp
```

WHERE DEPT NO = 10 AND JOB IN ('MANAGER',
'CLERK');

Q. Write a query to display all the details present in dept
table

```
SELECT *  
FROM DEPT;
```

Q. WHAT'S no of employees , maximum salary , minimum salary,
AVG sal , total sal of the employees who are
working as salesman , manager , analyst , in the deptno
10 , 20 , 30 and getting sal more than 1000 and less than
4000 and employee names consist of 5 characters

```
SELECT MAX(SAL), COUNT(X), MIN(SAL), AVG(SAL), SUM(SAL)  
FROM Emp  
WHERE JOB IN ('SALESMAN', 'MANAGER', 'ANALYST') AND  
DEPT NO IN (10, 20, 30) AND SAL > 1000 AND  
SAL < 4000 AND ENAME LIKE '_____';
```

* Q. WHAT IS NO. OF EMP WORKING IN EACH DEPARTMENT

SELECT COUNT(X)

FROM EMP

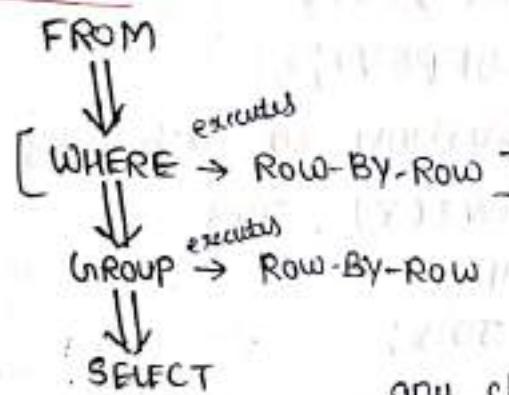
GROUP BY CLAUSE :-

It is used to group the records.

Syntax:- SELECT GROUP FUNCTION / GROUP BY EXPRESSION
 FROM TABLE-NAME
 [WHERE <FILTER CONDITION>]
 GROUP BY Col-NAME | EXP;

[] is not mandatory

ORDER of Execution:-



NOTE:-

- Group By clause executes after the execution of from clause.
- Group By clause executes row by row.
- Group By clause is used to group the records.
- After the execution of group by clause we will get the groups.
- Any clause executes after the execution of groupBy clause will executes group by group.

Example:-

EMP

Q. WHAT IS NO. OF EMP WORKING IN EACH DEPT

EMPNO	ENAME	SAL	DEPTNO
1	A	100	20
2	B	200	10
3	C	250	20
4	D	300	30
5	B	400	10
6	E	350	20
7	F	450	30

90 group			
1	A	100	20
3	C	250	20
6	E	350	20

=3

10 group			
2	B	200	10
5	B	400	10

=2

20 group			
4	D	300	30
7	F	450	30

=2

SELECT COUNT(X), DEPTNO

FROM Emp

GROUP BY DEPTNO;

RESULT

COUNT(X)	DEPTNO
3	20
2	10
2	30

Q. WAP TO no. of emp , max sal , who are working in each department

```
SELECT COUNT(X), DEPTNO, MAX(SAL)  
FROM EMP  
GROUP BY DEPT NO;
```

Q. WAP TO no. of emp who are getting sal more than 2000 in each department

```
SELECT COUNT(X), DEPTNO  
FROM EMP  
WHERE SAL > 2000  
GROUP BY DEPT NO;
```

Q. WAP TO no. of emp working in each job

```
SELECT COUNT(X), JOB  
FROM EMP  
GROUP BY JOB;
```

Q. WAP TO No. of emp , max sal , who are getting sal in each job

```
SELECT COUNT(X), MAX(SAL), JOB  
FROM EMP WHERE SAL IS NOT NULL  
GROUP BY JOB;
```

Assignment questions on ~~group~~ BY GroupBy

Q.1. WAP TO No. of emp working in each dept except president.

```
SELECT COUNT(X), DEPT NO  
FROM EMP  
WHERE JOB IS NOT IN 'PRESIDENT'  
GROUP BY DEPT NO;
```

Q.2. WAP TO Total sal needed to pay all the emp. in each job

```
SELECT SUM(SAL), JOB  
FROM EMP  
GROUP BY JOB;
```

Q.3 WHAT IS NO. OF EMP WORKING AS MANAGER IN EACH DEPT

```
SELECT COUNT(X), DEPTNO  
FROM EMP  
WHERE JOB = 'MANAGER'  
GROUP BY DEPTNO;
```

Q.4 WHAT IS AVG SAL NEEDED TO PAY ALL THE EMPLOYEES IN EACH DEPARTMENT EXCLUDING THE EMP. OF DEPTNO 20.

```
SELECT AVG(SAL), DEPTNO  
FROM EMP  
WHERE DEPT NOT IN 'DEPTNO 20'  
GROUP BY DEPTNO;
```

Q.5 WHAT IS NUMBER OF EMP. HAVING CHARACTER 'A' IN THEIR NAMES IN EACH JOB

```
SELECT COUNT(X), JOB  
FROM EMP  
WHERE ENAME LIKE '%A%'  
GROUP BY JOB;
```

Q.6 WHAT IS NO. OF EMP AND AVG SAL NEEDED TO PAY THE EMPLOYEES WHOSE SAL IS GREATER THAN 2000 IN EACH DEPT

```
SELECT COUNT(X), AVG(SAL), DEPTNO  
FROM EMP  
WHERE SAL > 2000  
GROUP BY DEPTNO;
```

Q.7 WHAT IS TOTAL SAL NEEDED TO PAY AND NO. OF SALESMAN IN EACH DEPT.

```
SELECT SUM(SAL), COUNT(X), DEPTNO  
FROM EMP WHERE JOB = 'SALESMAN'  
GROUP BY DEPTNO;
```

Q.8. WHAT ID n.o. of emp with their maximum salaries in each job

```
SELECT COUNT(*), MAX(SAL), JOB  
FROM EMP  
GROUP BY JOB ;
```

Q.9. WHAT ID max sal given to an emp working in each dept.

```
SELECT MAX(SAL), DEPT NO  
FROM EMP  
GROUP BY DEPT NO ;
```

Q.10. WHAT ID no. of times the sal present in emp table

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

Q. WHAT ID no. of employees working in each department and the employees are working as salesman, manager, ANALYST, in the department NO 10, 20, 30 and the employees names consist of 5 character and who are getting salary more than 1000 and hired after 1981.

```
SELECT COUNT(*), DEPTNO  
FROM EMP  
WHERE JOB IN ('SALESMAN', 'MANAGER', 'ANALYST') AND  
DEPTNO IN (10, 20, 30) AND ENAME LIKE '_____'  
AND SAL > 1000 AND HIRE DATE > '31-DEC-81';  
GROUP BY DEPTNO ;
```

* Q. WHAT ID no. of emp working in each department and there have atleast two emp in each department.

```
SELECT COUNT(*), DEPTNO  
FROM EMP
```

HAVING CLAUSE :- It is used to filter the groups.

SYNTAX :- SELECT GROUPFUNCTION / GROUPBY EXP

From Table Name

[WHERE < Filter - condition >]

GROUP BY Col-Name / EXP

HAVING < Group - FILTER - condition > ;

Order of execution :-

From

↓
[WHERE Row-By-Row]

↓
GROUP BY Row-By-Row

↓
HAVING GROUP-BY-GROUP

↓
SELECT GROUP-BY-GROUP

Example :-

NOTE :- • Having clause execute after the execution of group by clause.

- Having clause execute group by group.
- Having clause is used to filter the groups.
- Without using groupby clause we cannot use having clause.
- In having clause we can use multivalue functions.

Example :- Works A QTD no.of emp working in each dept and there has at least 2 employees in each dept.

Ans →

① → SELECT COUNT(X) , DEPTNO
 ② → From EMP
 ③ → Group By DEPTNO
 ④ → HAVING COUNT(X) ≥ 2;

EMPNO	ENAME	SAL	DEPTNO
1	A	1000	10
2	B	2000	20
3	C	3000	10
4	D	4000	10
5	E	5000	10
6	F	6000	20
7	G	7000	30
8	H	8000	30

Output of query by

10

↙

RESULT

1 A 1000 10	= [3 2 2] ✓
3 C 3000 10	
5 E 5000 10	

2 B 2000 20	= [2 2 2] ✓
6 F 6000 20	
7 G 7000 30	

COUNT(X)	DEPTNO
3	10
2	20
1	30

count(X) ≥ 2

Q. WHAT NO. OF EMP WORKING IN EACH JOB AND THERE HAVE AT LEAST 3 EMP IN EACH JOB.

SELECT COUNT(*) , JOB

FROM Emp

GROUP BY JOB

HAVING COUNT(*) ≥ 3;

Q. WHAT NO. OF EMP, MIN(SAL) AND MAX SAL OF THE EMP WHO ARE WORKING IN EACH DEPARTMENT AND THERE MAY SAL MORE THAN 4000.

S COUNT(*) , MIN(SAL) , MAX(SAL) , DEPT NO

F Emp

G-B DEPT NO

H MAX(SAL) > 4000;

Q. WHAT SALARIES WHICH ARE REPEATED

SELECT SAL

FROM Emp

GROUP BY SAL

HAVING COUNT(SAL) = 2;

↓
COUNT(SAL) ≥ 2;

Q. WHAT HIREDATE WHICH ARE REPEATED

SELECT HIREDATE

FROM Emp

GROUP BY HIREDATE

HAVING COUNT(HIREDATE) ≥ 2;

↓ OR

COUNT(*) ≥ 2;

Assignment Question of Having clause

- Q.1 WAP TO Deptno and no. of emp of working in each dept
if there are atleast 2 clerk in each Dept.
- ```
SELECT COUNT(*), DEPTNO
FROM EMP
GROUP BY DEPTNO
HAVING COUNT(*) >= 2;
```
- Q.2 WAP TO D.No and total salary needed to pay all emp in  
each dept if there are atleast 4 emp in each dept
- ```
SELECT SUM(SAL), DEPTNO, COUNT(*) AS COUNT
FROM EMP
GROUP BY DEPTNO
HAVING COUNT(*) >= 4;
```
- Q.3 WAP TO No. of emp earning sal more than 1200 in each job and the total sal needed to pay emp of each job must excess 3800
- ```
SELECT COUNT(*), JOB, SUM(SAL)
FROM EMP
WHERE SAL > 1200
GROUP BY JOB
HAVING SUM(SAL) > 3800;
```
- Q.4. WAP TO D.No and no. of emp working only if there are 2 emp working in each DFPT as manager
- ```
SELECT COUNT(*), DEPTNO
FROM EMP
WHERE JOB = 'MANAGER'
GROUP BY DEPTNO
HAVING COUNT(*) >= 2;
```

Q.5 WHATD JOB and max SAL of emp in each JOB if the
max sal exceeds 2600

```
SELECT MAX(SAL), JOB  
FROM Emp  
GROUP BY JOB  
HAVING MAX(SAL) > 2600;
```

Q.6 WHATD the SAL which are repeated in emp tabl.

```
SELECT SAL  
FROM Emp  
GROUP BY SAL  
HAVING COUNT(X) ≥ 2;
```

Q.7. WHATD the HIREDATE which are duplicated in emp
table

```
SELECT HIREDATE  
FROM Emp  
GROUP BY HIREDATE  
HAVING COUNT(X) ≥ 2;
```

Q.8. WHATD 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;
```

Q.9 WHATD deptno if there are atleast 3emp in each dept
whose name has char 'A' OR 'S'
SELECT DEPTNO
FROM Emp
WHERE ENAME LIKE '%A%' OR ENAME LIKE '%S%'

Q.10 WHATD min and maxsal of each job if minsal is more than 1000 and
maxsal is less than 5000

```
SELECT MAX(SAL), MIN(SAL), JOB  
FROM Emp  
GROUP BY JOB
```

HAVING MIN(SAL) > 1000 AND MAX(SAL) < 5000;

Q Difference b/w Where clause and Having clause.

WHERE

HAVING

- 1) Where clause is used to filter the records.
- 2) Where clause executes before the execution of group by clause.
- 3) Where clause executes row by row.
- 4) It cannot accept multirow function.
- 1) Having clause is used to filter the group.
- 2) Having clause executes after the execution of group by clause.
- 3) Having clause executes group by group.
- 4) It can accept multirow function.

Q. What no. of emp, min(sal), max(sal), total sal of each employees who were hired after 1950, and the employees getting salary more than 10. and then min(sal) should be more than 15 and max(sal) less than 5000 and then avg(sal) should be less than 50 and the emp who are working as 'Salesman', 'manager', 'analyst' in the dept no 10, 20, 30

```
SELECT COUNT(*), MIN(SAL), MAX(SAL), SUM(SAL), DEPTNO  
FROM EMP ENAME  
WHERE HIREDATE > '1950-12-31' AND SAL > 10 AND JOB IN  
('SALESMAN', 'MANAGER', 'ANALYST') AND  
DEPT NO IN (10, 20, 30)
```

GROUP BY ENAME DEPTNO

HAVING MIN(SAL) > 15 AND MAX(SAL) < 5000 AND
AVG(SAL) < 50;

Q WHATS Salaries of each emp

```
SELECT SAL  
FROM EMP;
```

Q WHATS names of all the emp.

```
SELECT ENAME  
FROM EMP;
```

ORDER BY CLAUSE:-

- It is used to arrange the records in ascending order or descending order.
- By default the compiler will arrange the records in ascending order.
- If we want to arrange the records in descending order we should use one key word called DESC

SYNTAX:-

```
SELECT GROUPFUNCTION / Group-By-Exp  
FROM T-N  
[ WHERE < FILTER CONDITION >  
  GROUP BY Col-Name / Exp  
  HAVING < GROUP FILTER CONDITION >  
  ORDER BY Col-NAME ; ]
```

ORDER of execution:-

```
From  
↓  
Where  
↓  
GroupBy  
↓  
Having  
↓  
Select  
↓  
OrderBy
```

Ex:- WORK TO Names of emp in ascending order

```
SELECT ENAME  
FROM EMP  
ORDER BY ENAME;
```

Q. WORK TO Sal of emp in ascending order

```
SELECT SAL  
FROM EMP  
ORDER BY SAL;
```

Q. WORK TO Name of the emp in descending order

```
SELECT ENAME  
FROM EMP  
ORDER BY ENAME DESC;
```

Q. WORK TO sal of the emp in descending order

```
SELECT SAL  
FROM EMP  
ORDER BY SAL DESC;
```

Q. WORK TO all the details of emp

```
SELECT *  
FROM EMP;
```

Q. WORK TO emp name , salary of the emp 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. WORK TO details of the emp who are getting sal more than 2000

```
SELECT *  
FROM EMP  
WHERE SAL > 2000;
```

Q. WAP TO details of the emp who are getting salary less than 4000.

```
SELECT *  
FROM EMP  
WHERE SAL < 4000;
```

* Q. WAP TO details of the emp who are getting salary more than 'ALLEN'.

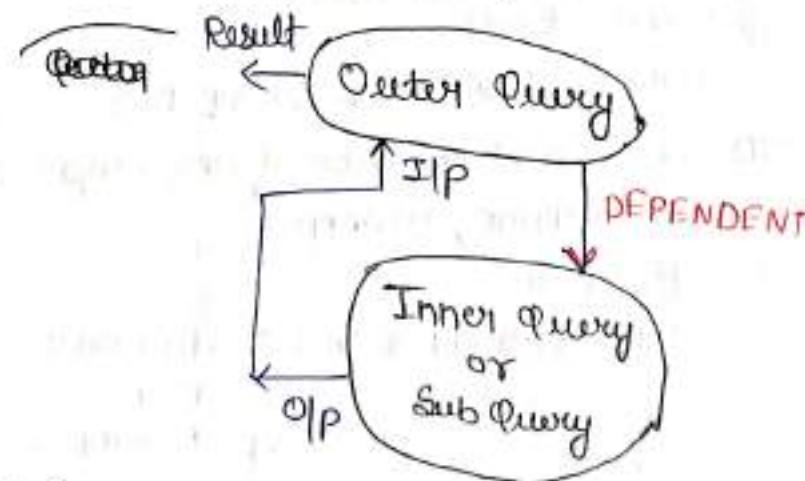
```
SELECT *  
FROM EMP
```

SUBQUERY :-

A query so written inside another query is known as sub query.

Q. Working process of subquery.

1. Inner query executes first and produce the output.
2. The output of inner query will be given as input to the outer query.
3. And then outer query will execute and produce the result.
4. Therefore outer query is completely dependent on inner query.



Q. When or why do we use subquery.
CASE 1

- Whenever we have unknown present in the query or a question to find that unknown we use subquery.

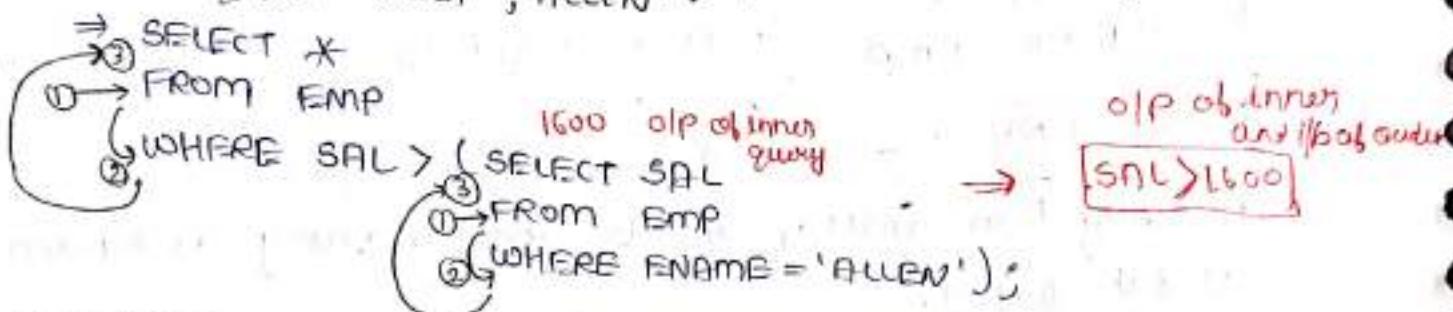
CASE 2:

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

SUBQUERY CASE 1 :-

- Whenever we have unknown present in the query or a question to find that unknown we used Subquery.

Ex:- Q. What are details of the emp who are getting salary more than after 'ALLEN'.



Q. What Name and salary of the emp those who are getting salary less than 'SCOTT'.

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL < (SELECT SAL
              FROM EMP
              WHERE ENAME = 'SCOTT');
```

Q. What NAME and HIREDATE of the emp those who hired after 1981

```
SELECT ENAME, HIREDATE
FROM EMP
WHERE HIREDATE > '31-DEC-1981';
```

Q. What NAME and HIREDATE of the emp who hired after JAMES

\$ ENAME, HIREDATE

F EMP

W HIREDATE > (\$ HIREDATE

F EMP

W ENAME = 'JAMES');

Q. What Name and department No of emp who are working in the same deptNo as KING

\$ ENAME, DEPTNO

F EMP

W DEPTNO = (\$ DEPTNO

F EMP

W ENAME = 'KING');

Q. What is Name and Comm of the emp who are getting comm more than ALLEN

S ENAME, COMM

F EMP

W COMM > (S COMM

F EMP

W ENAME = 'ALLEN');

Q. What is Name and Sal of the emp who are getting sal less than 2000 and getting sal more than 'SCOTT';

S ENAME, SAL

F EMP

W SAL < 2000 AND SAL > (S SAL

F EMP

W ENAME = 'SCOTT');

Assignment on SUBQUERY CASE I

1. What is name of the emp earning more than ADAMS

S ENAME

F EMP

W SAL > (S SAL

F EMP

W ENAME = 'ADAMS');

2. What is name and salary of the emp earning less than KING.

S ENAME, SAL

F EMP

W SAL < (S SAL

F EMP

W ENAME = 'KING');

3. What is Name and Deptno of the emp if they are working in the same DEPT as JONES

S NAME, DEPTNO

F EMP

W Job = (S Job

W DEPTNO = (S DEPTNO

F EMP

W Job

S NAME, DEPTNO

F EMP

W DEPTNO = (S DEPTNO

F EMP

W ENAME > 'JONES');

4. **W_HE_RT_O** Name and Job of all the emp working in the same designation as James.

S ENAME , JOB
F Emp

W JOB = (S JOB

F EMP

W ENAME = 'JAMES');

5. **W_HE_RT_O** empno and ENAME along with annual sal of all the emp if their annual sal is greater than wards annual salary.

S EMPNO , ENAME , SAL*12
F Emp

W SAL*12 > (S SAL*12

F Emp

W ENAME = 'WARD');

6. **W_HE_RT_O** Name and HIREDATE of the emp if they are hired after SCOTT.

S ENAME , HIREDATE
F Emp

W HIREDATE > (S HIREDATE

F Emp

W ENAME = 'SCOTT');

7. **W_HE_RT_O** NAME and HIREDATE of the emp if they are hired after the PRESIDENT

S ENAME , HIREDATE

F Emp

W HIREDATE > (S HIREDATE

F Emp

W ENAME = 'PRESIDENT');

8. **W_HE_RT_O** Name and SAL of the emp if they are earning sal less than the emp whose empno is 7839 .

S ENAME , SAL

F Emp

W SAL < (S SAL

F Emp

W EMPNO = 7839);

9. **W_HE_RT_O** all the details of the emp if the emp are hired Before MILLER

S *

F Emp

W HIREDATE < (S HIREDATE

F Emp

W ENAME = 'MILLER');

10. WANTED ENAME and EMPNO of the emp if emps are earning more than ALLEN
- g ENAME , EMPNO
F EMP
W SAL > (β SAL
F EMP
W ENAME = 'ALLEN');
11. WANTED ENAME and SAL of all the emp who are earning more than miller but less than ALLEN
- ⇒ g ENAME , SAL
F EMP
W SAL > (β SAL
F EMP
W ENAME = 'MILLER') AND \wedge \wedge (β SAL
F EMP
W ENAME = 'ALLEN');

⇒ β ENAME , SAL
F EMP
W SAL > (β SAL
F EMP
W ENAME = 'MILLER') AND SAL < (β SAL
F EMP
W ENAME = 'ALLEN');

12. WANTED all the details of the emp working in DEPTNO and working in the same DESIGNATION as SMITH.

⇒ β *
F EMP
W DEPTNO = 20 AND JOB = (β \$JOB
F EMP
W ENAME = 'SMITH');

13. WANTED all the details of the emp working as manager in the same DEPT as TURNER

⇒ β *
F EMP
W JOB = 'MANAGER' AND DEPTNO = (β DEPTNO
F EMP

- Q.14 WANTED Name and HIREDATE of the emp hired after 1980 and Before KING.

g ENAME , HIREDATE
F EMP
W HIREDATE > '31-DEC-1980' AND \wedge \wedge HIREDATE < (β HIREDATE
F EMP
W ENAME = 'KING');

Q.15 WHATD Name and sal along with annual sal for all employees whose sal is less than Blake and more than 3500

\$ ENAME , SAL , SAL*12
F EMP

W SAL < (\$ SAL

F EMP

W ENAME = 'BLAKE') AND SAL > 3500;

Q.16 WHATD all the details of emp who earn more than scott but less than KING.

\$ *

F EMP

W SAL > (\$ SAL

F EMP

W ENAME = 'SCOTT') AND SAL < (\$ SAL

F EMP

W ENAME = 'KING')

Q.17 WHATD Name of the emp whose name starts with 'A' and works in the same dept as BLACK

\$ ENAME

F EMP

W ENAME LIKE 'A%' AND DEPTNO = (\$ DEPTNO

F EMP

W ENAME = 'BLACK')

Q.18 WHATD Name and comm if employees earn commission and work in the same designation as SMITH .

\$ ENAME , comm

F EMP

W comm IS NOT NULL AND JOB = (\$ JOB

F EMP

W ENAME = 'SMITH')

Q.19 . WHATD Details of all the emp working as clerk in the same dept as TURNER.

\$ *

F EMP

W JOB = 'CLERK' AND DEPTNO = (\$ DEPTNO

F EMP

W ENAME = 'TURNER')

Q.20 WHATD ename ,sal and designation of the emp whose annual salary is more than SMITH and less than KING

\$ ENAME, SAL, JOB
F EMP
W SAL*12 > (\$ SAL*12
F EMP
W ENAME = 'SMITH') AND SAL*12 < (\$ SAL*12
F EMP
W ENAME = 'KING');

Q. Wanted details of the emp along with annual sal
and those who are working as 'salesman', 'MANAGER',
'CLERK', and working in the same depno as 'KING'
and getting sal more than 2000 but less than 'KING' and
the emp who hired after 1980 but before 'SCOTT' and
the emp who are getting comm more than 'TURNER'
and the emp not having MGR

~~select~~

\$ EMP.* , SAL*12
F Emp
W JOB IN ('SALESMAN', 'MANAGER', 'CLERK') AND DEPTNO = (\$ DEPTNO
F Emp
W ENAME = 'KING') AND SAL > 2000 AND SAL < (\$ SAL
F Emp
W ENAME = 'KING') AND HIREDATE > '31-DEC-1980'
AND HIREDATE < (\$ HIREDATE)
F Emp
W ENAME = 'SCOTT') AND Comm >
(\$ Comm
F Emp
W ENAME = 'TURNER')
AND MGR IS NULL;

Q. WAP TO SELECT all the details of 'ALLEN'.

S X

F EMP

W ENAME = 'ALLEN';

* Question DEPT NAME of 'ALLEN'.

S B NAME

F DEPT

W DEPTNO = (S DEPTNO)

SUBQUERY CNQS 2 :-

F EMP

W ENAME = 'ALLEN');

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

example:-

EMP

EMPNO	ENAME	SAL	DEPTNO
1	SMITH	1000	10
2	ALLEN	1500	20
3	MILLER	2000	10
4	SCOTT	3000	20
5	KING	2500	20

DEPT

DEPTNO	DNAME	LOC
10	D1	L1
20	D2	L2
30	D3	L3

Q. WAP TO DEPTNAME of ALLEN

S B DNAME

F DEPT

W DEPTNO = (S DEPTNO)

DEPTNO 20

F EMP

W ENAME = 'ALLEN');

Q. WAP TO location of KING

S LOC

F DEPT

W DEPTNO = (S DEPTNO)

F EMP

W ENAME = 'KING');

Q. WAP TO Name of the Emp who are working as in the location NEWYORK

S ENAME

F EMP

W DEPTNO = (S DEPTNO)

F EMP DEPT

W LOC = 'NEWYORK');

Q.WHATD Employee name, sal, hiredate of the emp who are working
in Research DEPT

S ENAME, SAL, HIREDATE

F EMP

W DEPTNO = (S DEPTNO

F DEPT)

W DNAME = 'RESEARCH');

Q. WHATD Name of the emp who are getting sal more than
MILLER

S ENAME

F EMP

W SAL > (S SAL

F EMP

W ENAME = 'MILLER');

Q. WHATD DNAME of the emp who are getting SAL more
than MILLER.

S DNAME

F DEPT

W DEPTNO IN (S DEPTNO

F EMP

W SAL > (S SAL

F Emp

W ENAME = 'MILLER');

Q. WHATD loc of the emp who hired after SCOTT

S LOC

F DEPT

W DEPTNO IN (S DEPTNO

F Emp

W HIREDATE > (S HIREDATE

who are working as SALESMAN

F Emp

W ENAME = 'SCOTT'));

S DNAME, Loc

F DEPT

W DEPTNO IN (S DEPTNO

F EMP

W JOB = 'SALESMAN');

Assignment of Subquery case :-

Q.21 WAQTD DNAME of emp whose name is SMITH

S DNAME

F DEPT

W DEPTNO IN (S DEPTNO F EMP

W ENAME = 'SMITH');

Q.22 WAQTD DName and LOC of the emp whose ENAME is KING

S DNAME, LOC

F DEPT

W DEPTNO IN (S DEPTNO

F EMP

W ENAME = 'KING');

Q.23 WAQTD Loc of the emp whose emp no is 7902

S LOC

F DEPT

W DEPTNO IN (S DEPTNO

F EMP

W EMPNO = 7902);

Q.24 WAQTD DNAME and LOC along with DEPTNO of the emp whose name ends with 'R'.

S DNAME, LOC, DEPTNO

F DEPT

W DEPTNO IN (S DEPTNO

F EMP

W ENAME LIKE '%.R');

Q.25 WAQTD DNAME of the emp whose designation is president

S DNAME

F DEPT

W DEPTNO IN (S DEPTNO

F EMP

W JOB = 'PRESIDENT');

Q.26 WAQTD name of the emp working in accounting Dept.

S ENAME

F EMP

W JOB IN (S DEPT

W DEPTNO IN (S DEPTNO

F DEPT

W DNAME = 'ACCOUNTING');

Q.27. WANTED ENAME and SAL of the emp who are working in the location CHICAGO

S ENAME , SAL

F EMP

W DEPTNO IN (S DEPTNO

F DEPT

W Loc='CHICAGO');

Q.28. WANTED details of the emp working in SALES

S *

F EMP

W DEPTNO IN (S DEPTNO

F DEPT

W DNAME='SALES');

Q.29. WANTED details of the emp along with ANNUAL salary if emp are working in New York.

S * EMP.X , SALX12

F EMP

W DEPTNO IN (S DEPTNO

F DEPT

W Loc='NEW YORK');

Q.30. WANTED Names of emp working in operations DEPT

S ENAME

F EMP

W DEPTNO IN (S DEPTNO

F DEPT

W DNAME='OPERATIONS');

Q.31. WANTED Names of the employees earning more than SCOTT in ACCOUNTING DEPT

S ENAME

F EMP

W SAL > (S SAL

F EMP

W ENAME='SCOTT') AND DEPTNO=(S DEPTNO

F DEPT

W DNAME='ACCOUNTING');

Q.32. WANTED Details of the emp working as manager in the location CHICAGO

S *

F EMP

W JOB='MANAGER' AND DEPTNO=(S DEPTNO

F DEPT

W Loc='CHICAGO');

Q.33. What is the name and sal of the emp earning more than king in the dept accounting.

S ENAME, SAL

F EMP

W SAL > (S SAL

F EMP

W ENAME = 'KING') AND DEPTNO IN (S DEPTNO

F DEPT

W DNAME = 'ACCOUNTING');

Q.34. What is the details of the emp. working as salesman in the department sales.

S *

F EMP

W JOB = 'SALESMAN' AND DEPTNO IN (S DEPTNO

F DEPT

W PNAME = 'SALES');

Q.35. What is the name ,SAL, JOB, HIREDATE of the emp .working in operations department and hired before K2N6

S ENAME, SAL, JOB, HIREDATE

F EMP

W DEPTNO IN (S DEPTNO

F DEPT

W DNAME = 'OPERATIONS') AND HIREDATE < (S HIREDATE

F EMP

W ENAME = 'KING');

Q.36. Display all the emp. whose dept name ending 'S'.

S ENAME

F EMP

W DEPTNO IN (S DEPTNO

F DEPT

W DNAME LIKE '%.S');

Q.37. What is the DNAME of the employees whose Names has character A 'A' in it.

S DNAME

F DEPT

W DNAME LIKE '%A%' AND DNAME < 'IN:)

W DEPTNO IN (S DEPTNO

F EMP

W ENAME LIKE '%A%');

Q38 WHATD DNAME and loc of the emp whose salary is Rupes 800.

S DNAME, LOC
F DEPT
W DEPTNO IN (S DEPTNO
F EMP
W SAL = 800);

Q39. WHATD DNAME of the emp. who earn comm.

S DNAME
F DEPT
W DEPTNO IN (S DEPTNO
F EMP
W COMM IS NOT NULL);

Q40. WHATD LOC of the emp if they earn comm in Dept 40

S LOC
F DEPT
W DEPTNO IN (S DEPTNO
F EMP
W COMM IS NOT NULL AND DEPTNO = 40);

Q. WHATD DName of emp whose name starts with An.

S DNAME
F DEPT
W DEPTNO IN (S DEPTNO
F EMP
W ENAME LIKE 'A%');

TYPES OF SUBQUERY :-

* SINGLE ROW Subquery

* MULTI ROW Subquery

Single Row Subquery :-

- If Subquery ~~return~~ returns exactly one value we call it as single row Subquery ~~return~~ exactly one value we can use either IN OPERATOR OR = OPERATOR to compare the values.
- If Subquery ~~return~~ returns more than one value we can use either IN OPERATOR OR = OPERATOR to compare the values.

Q. WHATD empname and deptno of emp

who are working in the same deptno as 'KINH'.

S ENAME, DEPTNO

F Emp

W DEPTNO IN (S DEPTNO

F DEPT Fmp

W ENAME = 'KINH');

Q. What's dname and loc of the emp whose empno is 7934

S DNAME, LOC

F DEPT

W DEPTNO IN (S DEPTNO

F Emp

W EMPNO = 7934);

2. Multi Row Subquery :-

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

Example:- Q. What's DeptName of the emp whose name starts with A

S DNAME

WF DEPT

W DEPTNO IN (S DEPTNO

F Emp

W ENAME LIKE 'A%');

Q. What's details of the emp whose deptname ends with 'S'.

S *

F EMP

W DEPTNO IN (S DEPTNO

F DEPT

W DNAME LIKE '%S');

Q. What's SAL of all the emp

S SAL

F Emp;

WF

Q. WHAT IS

NOTE :- It is difficult to identify whether the subquery ^{returns} ~~were~~ single row or multiRow so instead of using = to operator we should use IN operator.

Q. WHAT IS SAL OF all the emp.

S SAL

F EMP;

Q. WHAT IS ename and SAL of the employees who are getting sal more than JAMES

S ENAME, SAL

F Emp

W ~~EMP~~ SAL > (S SAL

F EMP

W ENAME = 'JAMES');

Q. WHAT IS Name and SAL of the emp who are getting SAL more than all the SALESMAN

S ENAME, SAL

F Emp

W SAL > (S SAL

F EMP

W JOB = 'SALESMAN');

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.

Example:-

Q. WHAT IS Name and SAL of the emp who are getting sal more than all the SALESMAN

S ENAME, SAL

F Emp

W SAL > ALL (S SAL

F EMP

W JOB = 'SALESMAN');

2. ANY :-
- Any operator is a special operator & it returns true if any one of the condition is satisfied.
 - Any operator is used along with relational operator.

Example:- Q. What Name and SAL of the emp who are getting sal more than any of the salesman.

S ENAME, SAL
 F EMP
 W SAL > ANY (S SAL
 F EMP
 W JOB='SALESMAN');

Assignment on types of sub Query.

Q.1. What Name of the emp earning sal more than the salesman

Ans - S ENAME
 F EMP
 W SAL > ALL (S SAL
 F EMP
 W JOB='SALESMAN');

Q.52. What details of the emp hired after all the clerks.

S *
 F EMP
 W HIREDATE > ALL (S HIREDATE
 F EMP
 W JOB='CLERK');

Q.53. What Name and salary for all the emp if they are earning less than atleast a MANAGER.

S ENAME, SAL
 F EMP
 W SAL < ANY (S SAL
 F EMP
 W JOB='MANAGER');

Q.54. What NAME and Hiredate of emp hired before all the manager

S ENAME, HIREDATE
 F EMP
 W HIREDATE < ALL (S HIREDATE
 F EMP
 W JOB='MANAGER');

Q.55. WAP TO Name of the employees hired after all the manager
and earning salary more than all the clerks.

S ENAME

F EMP

W HIREDATE > ALL (S HIREDATE

F Emp

W JOB='MANAGER') AND SAL > ALL (S SAL

FEMP

W JOB='CLERK')

Q.56 WAP TO details of the emp working as clerk and hired before
at least a salesman.

S *

F Emp

W JOB='CLERK' AND HIREDATE < ANY (S HIREDATE

F Emp

W JOB='SALESMAN');

Q.57 WAP TO details of the emp working as clerk and hired before
in ACCOUNTING OR SALES DEPT

S *

F EMP

W DEPTNO IN (S DEPTNO

F DEPT

W DNAME = 'ACCOUNTING' OR DNAME = 'SALES');

Q.58 WAP TO dept names of the emp with names SMITH, KING AND
MILLER.

S DNAME

F DEPT

W DEPTNO IN (S DEPTNO

F Fmp

W ENAME IN ('SMITH', 'KING', 'MILLER'));

Q.59. WAP TO details of emp working NEWYORK OR CHICAGO

S *

F Emp

W DEPTNO IN (S DEPTNO

F DEPT

✓ W LOC = 'NEW YORK' OR LOC = 'CHICAGO');

OR

✓ W LOC IN ('NEW YORK', 'CHICAGO');

- Q60. WHATD Emp Names if employee are hired after all the employee of DEPT 10.
- S FNAME
F Emp
- W HIREDATE > ALL(S HIREDATE
F Emp
W DEPT NO = 10);
- Q. WHATD details of the emp who hired after MILLER and getting sal more than 'SALESMAN' and the ENAME consist of 5 char.
- S *
F Emp
- W HIREDATE > (S HIREDATE
F Emp
W ENAME = 'MILLER') AND SAL >
ALL(S SAL
F Emp
W JOB = 'SALESMAN')
AND ENAME LIKE '_____';
- Q. WHATD Names of all the emp
- S ENAME
F Emp;
- Q. WHATD SAL of all the emp.
- S SAL
F Emp;
- Q. WHATD first maximum sal of emp.
- S MAX(SAL)
F Emp;
- Q. WHATD name of the emp who is getting first maximum salary
- X S MAX(SAL), ENAME
F Emp
GROUP BY ENAME;
- ✓
- S ENAME
F Emp
W SAL = (S MAX(SAL)
F EMP);

Q. What is the name and sal of the emp who are getting first max(sal)

S ENAME, SAL
F EMP
W SAL = (S MAX(SAL))
F EMP);

* Q. What is the name of the emp who are getting second maximum sal.

S ENAME
F EMP
W SAL = (S MAX(SAL))
F EMP
W SAL < (S MAX(SAL))
F EMP);

EMP	
ENAME	SAL
A	1000
B	2000
C	3000
D	4000
E	5000

* NESTED SUBQUERY

- A Subquery written inside another subquery is known as nested subquery.
- We can nested upto 255 subquery.

Example:- What is the name of emp who are getting second maximum salary.

EMP	
ENAME	SAL
A	1000
B	2000
C	3000
D	4000
E	5000

→ D
S ENAME
F EMP
W SAL = (S MAX(SAL))
SAL = 4000 → 4000
F EMP
W SAL < (S MAX(SAL))
SAL < 5000 → 5000
F EMP);

Remaining 4000
3000
2000
1000

Q. What is the name of
3rd max(sal) of emp

S ENAME
F EMP
W SAL = (S MAX(SAL))
F EMP
W SAL < (S MAX(SAL))
F EMP
W SAL < (S MAX(SAL))
F EMP));

Q. WAPTO DNAME of emp who are getting first min(SAL)

S DNAME

F DEPT

W DEPTNO IN (S DEPTNO

F EMP

W~~B~~ SAL = (S MIN(SAL))

F EMP));

Q. WAPTO loc of the emp who are getting second min(SAL)

S LOC

F DEPT

W DEPTNO IN (S DEPTNO

F EMP

W~~B~~ SAL = (S MIN(SAL))

F EMP

W SAL > (S MIN(SAL))

F EMP));

To find the min(SAL)

we have to use >

and for max(SAL) we use <

Q. WAPTO details of the emp who are getting 3rd max(SAL)

S *

F EMP

W SAL = (S MAX(SAL))

F EMP

W SAL < (S MAX(SAL))

F EMP

W SAL < (S MAX(SAL))

F EMP));

Q. WAPTO departmentName and loc of the emp who are getting
12th max(SAL)

S DEPT

S DNAME, LOC

F EMP

W SAL

S DEPT

Q.43. SELECT Name and HIREDATE of the emp hired before all the employees (First emp)

S FNAME, HIREDATE

F EMP

W HIREDATE = (S MIN(HIREDATE)
F EMP);

Q.44. SELECT Name and HIREDATE of the employee hired at the last

S FNAME, HIREDATE

F EMP

W HIREDATE = (S MAX(HIREDATE)
F EMP);

Q.45. SELECT Name, COMM of the emp who earns min comm.

S FNAME, COMM

F EMP

W COMM = (S MIN(COMM)
F EMP);

Q.46. SELECT Name, SAL AND COMM of the emp who earns max comm.

S FNAME, SAL, COMM

F EMP

W COMM = (S MAX(COMM)
F EMP);

Q.47. SELECT details of the emp who has greatest EMPNO

S *

F EMP

W EMPNO = (S MAX(EMPNO)
F EMP);

Q.48. SELECT details of the emp having the least HIREDATE.

S *

F EMP

W HIREDATE = (S MIN(HIREDATE)
F EMP);

Q.49. SELECT details of the emp earning least annual salary

S *

F EMP

W SALX12 = (S MIN(SALX12)
F EMP);

Q. 50 WAPTO Name, Annual salary of the emp if thou, annual salary is more than all the salesman.

S ENAME, SALX12

F EMP

W SALX12 > All (S SALX12

F EMP

W JOB = 'SALESMAN');

Q. 61 WAPTO 2nd MIN(SAL)

S MIN(SAL)

F EMP

W SAL > (S MIN(SAL)
F EMP);

Q. 62 WAPTO 5th Max(SAL)

S MAX(SAL)

F EMP

W SAL < (S MAX(SAL)

F EMP

W SAL < (S MAX(SAL)

F EMP

W SAL < (S MAX(SAL))

F EMP

W SAL < (S MAX(SAL))
F EMP));

Q. 63 WAPTO Name of the emp. earning 3rd max(SAL)

S ENAME

F EMP

W SAL = (S MAX(SAL))

F EMP

W SAL < (S MAX(SAL))

F EMP

W SAL < (S MAX(SAL))

F EMP));

Q. 64 WAPTO EmpNO of the emp. earning 2nd max(SAL)

S EMPNO

F EMP

W SAL = (S MAX(SAL))

F EMP

W SAL < (S MAX(SAL))

F EMP));

Q.65 WHEREO department of an emp getting 4th max sal

\$ DNAME

F DEPT

W DEPTNO IN (% DEPTNO

F EMP

W SAL < (% MAX(SAL))

F EMP))));

Q.66 WHEREO details of the emp who was hired 2nd

\$ *

F EMP

W HIREDATE = (% MIN(HIREDATE))

F EMP

W HIREDATE > (% MIN(HIREDATE))

F EMP));

Q.67 WHEREO Name of the emp hired before the last emp.

\$ ENAME

F EMP

W HIREDATE = (% MAX(HIREDATE))

F EMP

W HIREDATE < (% MAX(HIREDATE))

F EMP));

Q.68. WHEREO Loc of the emp who was hired first.

\$ LOC

F DEPT

W DEPTNO IN (% DEPTNO

F EMP

W HIREDATE = (% MIN(HIREDATE))

F EMP));

Q.69. WHEREO details of the emp earning 7th min(sal)

\$ *

F EMP

W SAL = (% MIN(SAL))

F EMP

W SAL > (\$ MIN(SAL))

F EMP

W SAL > (\$ MIN(SAL))

F EMP

W SAL > (\$ MIN(SAL))

F EMP))))));

Q. TO WAQTD DNAME of emp getting 2nd MAX(SAL).

S DNAME

F DEPT

W DEPTNO IN (\$ DEPTNO

F EMP

W SAL < (\$ MAX(SAL))

F EMP

W SAL < (\$ MAX(SAL))

F EMP));

Q WAQTD all the details of emp

S *

F EMP;

Q WAQTD all the details of SMITH

S *

F EMP

W ENAME = 'SMITH';

* Q WAQTD manager name of SMITH

S

X S ENAME

F EMP

Q WAQTD MGRNO of SMITH, W MGR = (\$ MGR

S MGR

F EMP

F EMP

W ENAME = 'SMITH');

W ENAME = 'SMITH';

* Q WAQTD manager name of SMITH

EMPLOYEE AND MANAGER RELATIONSHIP

1. WAQTD manager name of SMITH

& F MP

EMPNO	ENAME	MGR
1	SMITH	2
2	ALLEN	3
3	MILLER	4
4	KING	

→ ALLEN

→ S ENAME

→ F EMP

→ W EMPNO = (\$ MGR

EMPNO=2)

2

→ F EMP

→ W ENAME = 'SMITH');

Q. What's manager name of ALLEN

S ENAME

F EMP

W EMPNO = (S MGR

F EMP

W ENAME = 'ALLEN');

Q. What's manager's manager name of 'SMITH'.

S ENAME

F EMP

W EMPNO IN (S MGR

F EMP

W EMPNO = (S MGR

F EMP

W ENAME = 'SMITH');

Q. What's manager's manager's manager name of ALLEN

S ENAME

F EMP

W EMPNO = (S MGR

F EMP

W EMPNO = (S MGR

F EMP

W EMPNO = (S MGR

F EMP

W EMPNO

W ENAME = 'ALLEN');

Q. What's Name of the emp who are reporting ALLEN.

S ENAME

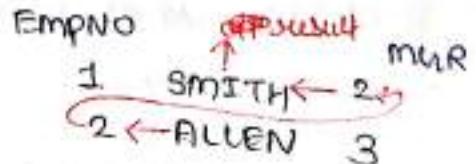
F EMP

W2 EMPNO

W MGR = (S EMPNO

F EMP

W ENAME = 'ALLEN');



Q. What's empnames who are reporting to KING

S ENAME

F EMP

W MGR = (S EMPNO

F EMP

W ENAME = 'KING');

NOTE:-

① To find MANAGER NAME

EMPNO = MGR

↓
SubQuery

② To find EMPNAME

MGR = EMPNO

↓
SubQuery.

Q.6 Q&A7D Name of the emp who are reporting to smith's
Manager

X
S ENAME
F EMP
W MGR = (S EMPNO
F EMP
W ENAME = 'SMITH') AND EMPNO = (S MGR
F Emp) +

✓
S ENAME
F EMP
W MGR > (S EMPNO
F EMP
W EMPNO = (S MGR
F EMP
OR
W ENAME = 'SMITH'));
S FNAME
F EMP
W MGR = (S MGR
F EMP
W ENAME = 'SMITH');

Assignment of emp and manager relation

Q.71 WAFD SMITHS reporting manager name.

S ENAME
F EMP
W EMPNO = (S MGR
F EMP
W ENAME = 'SMITH');

Q.72 WAFD ADAMS Manager's manager name.

S ENAME
F EMP
W EMPNO IN (S MGR
F EMP
W EMPNO = (S MGR
F EMP
W ENAME = 'ADAM'));

Q73. WHAT IS DNAME OF JONES'S MANAGER

S DNAME

F EMP DEPT

W DEPTNO IN (S DEPTNO

F EMP

W EMPNO = (% MGR

F EMP

W ENAME = 'JONES');

Q74. WHAT IS MILLER'S manager's salary

S SAL

F EMP

W EMPNO IN (S MGR

F EMP

W ENAME = 'MILLER');

Q75. WHAT IS LOC OF SMITH'S manager's manager.

S LOC

F DEPT

W DEPTNO IN (S DEPTNO

F EMP

W EMPNO = (% MGR

F EMP

W EMPNO = (% MGR

F EMP

W ENAME = 'SMITH');

S ENAME

F EMP

W MGR IN (% DEPNO)

F EMP

W ENAME = 'BLAKE');

Q77. WHAT IS NUMBER OF EMP. REPORTING TO KING

S ENAME COUNT(*)

F EMP

W MGR IN (% DEPNO)

F EMP

W ENAME = 'KING');

Q. 78. What details of the emp. reporting to JONES.

S *

F EMP

W MGR IN (S EMPNO

F EMP

W ENAME = 'JONES');

Blake

Q. 79. What is the ENAME of the emp. reporting to Black's manager.

S ENAME

F EMP

W MGR = (S MGR

F EMP

W ENAME = 'BLAKE');

S ENAME

F EMP

W MGR = (S MGR

F EMP

W EMPNO = (S MGR

F EMP

W ENAME =

'BLAKE');

Q. 80. What is the number of emp. reporting to Ford's manager.

S ENAMES (COUNT(X))

F EMP

W MGR = (S MGR

F EMP

W ENAME = 'FORD');

Q. What are the ENAMES of all the emp.

S ENAMES

F EMP;

Q. What are all the details of 'ALLEN'.

S *

F EMP

W ENAME = 'ALLEN');

Q. What is the deptname of ALLEN

S DNAME

F DEPT

W DEPTNO IN (S DEPTNO

F EMP

W ENAME = 'ALLEN');

* Q. What are the empname and deptname of the employees.

JOINS :-

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

Types of JOINS :-

- 1 * CROSS JOINS | CARTESIAN JOINS
- 2 * EQUI JOIN | INNER JOIN
- 3 * OUTER JOIN
 - ↳ * LEFT OUTER JOIN
 - * RIGHT OUTER JOIN
 - * FULL OUTER JOIN
- 4 * NATURAL JOIN
- 5 * SELF JOIN.

1. CROSS JOIN | CARTESIAN JOIN :-

In cartesian join the records from table one will be matched with all the records records from table two.

SYNTAX

ORACLE

```
SELECT Col-Name
From Table_Name1, Table_Name2 ;
```

ANSI

```
SELECT Col-Name
```

```
From Table_Name1 CROSS JOIN Table_Name2 ;
```

Example :-

Q) What Empname and DNAME of the employees

B) ENAME, DNAME

F) EMP, DEPT ;

OR ANSI

B) ENAME, DNAME

F) EMP CROSS JOIN DEPT ;

EMP

ENAME	DEPTNO
A	10
B	20
C	30

DEPT

DNAME	DEPTNO
D1	10
D2	20
D3	30

RESULT

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

ENAME	DNAME
A	D1
A	D2
B	D3
B	D1
B	D2
C	D3
C	D1
C	D2

- 2) INNER JOIN :- It is used to obtain only matching records
 Whenever we are using inner join we should write join condition.

SYNTAX:-

ORACLE

```
SELECT Col-NAME
  FROM T-N1 , TN-2
 WHERE <JOIN condition>;
```

ANSI

```
SELECT Col-Name
  From T-N1 , TN-2 INNER TN-2
 ON <JOIN condition>;
```

Example:- φ WAQTD
 EMP
 ENAME DEPTNO
 A 10
 B 20
 C .. 30

DNAME, and DEPTNO of the employees.

DEPT	
DNAME	DEPTNO
D1	10
D2	20
D3	30

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

Ans:- B ENAME , DNAME // ORACLE
 F EMP , DEPT
 W EMP.DEPTNO = DEPT. DEPTNO ;

OR ANSI

B ENAME , DNAME
 F EMP INNER DEPT
 W ON EMP.DEPTNO = DEPT. DEPTNO ;

Q. WAP TO salary and DNAME of emp

S SAL, DNAME

F EMP, DEPT

W EMP.DEPTNO = DEPT.DEPTNO ;

Q. WAP TO empname, sal, Hiredate, deptname, loc of all the employees

S SAL, HIREDATE, DNAME, LOC, FNAME

F EMP, DEPT

W EMP.DEPTNO = DEPT.DEPTNO ;

Q. WAP TO FNAME, SAL, and the DNAME of the employees who are getting sal more than 2000.

S FNAME, SAL, DNAME

F EMP, DEPT

W EMP.DEPTNO = DEPT.DEPTNO AND SAL > 2000 ;

Assignment on INNER JOIN

1. Name of the emp. and his loc aft. of all the employees

S FNAME, LOC

F EMP, DEPT

W EMP.DEPTNO = DEPT.DEPTNO ;

2. WAP TO DNAME and SAL for all the emp working IN accounting.

S DNAME, SAL

F DEPT, EMP

W DEPT.DEPTNO = EMP.DEPTNO AND DNAME = 'ACCOUNTING' ;

3. WAP TO DNAME and Annual sal for all employees whose sal is more than 2340

S DNAME, SAL * 12

F DEPT, EMP

W DEPT.DEPTNO = EMP.DEPTNO AND SAL > 2340 ;

4. WAP TO FNAME and DNAME for employees having character 'A' in their DNAME

S FNAME, DNAME

F EMP, DEPT

W EMP.DEPTNO = DEPT.DEPTNO AND DNAME LIKE '%A%' ;

5. WAP TO FNAME and DNAME for all the emp working as salesman.

S FNAME, DNAME

F EMP, DEPT

W EMP.DEPTNO = DEPT.DEPTNO AND JOB = 'SALESMAN' ;

Q.6) SELECT DNAME and JOB for all the emp whose JOB and DNAME Starts with character's

S DNAME, JOB

F DEPT, EMP

W DEPT. DEPTNO=EMP.DEPTNO AND JOB LIKE 'S%.' AND DNAME LIKE 'S%.'

Q.7) SELECT DNAME and MGR no. for emp reporting to 7839

S DNAME, MGR

F DEPT, EMP

W DEPT. DEPTNO=EMP.DEPTNO AND MGR = 7839;

Q.8) SELECT DNAME and HIREDATE for employees hired after 83 into ACCOUNTING OR RESEARCH DEPT

S DNAME, HIREDATE

F DEPT, EMP

W DEPT. DEPTNO=EMP.DEPTNO AND HIREDATE > '31-DEC-83' AND
& DNAME IN ('ACCOUNTING', 'RESEARCH');

Q.9) SELECT ENAME and DNAME of the emp. who are getting comm in
DEPT 10 OR 30

S ENAME, DNAME

F EMP, DEPT

W EMP. DEPTNO=DEPT. DEPTNO AND Comm IS NOT NULL AND
EMP.DEPTNO IN (10, 30);
when we have 2 deptno in same line we use
table name. ColName

Q.10) SELECT DNAME AND EMPNO for all the emp whose empno are
(7839, 7902) and are working in Loc NEW YORK

S DNAME, EMPNO

F DEPT, EMP

W EMP. DEPTNO=DEPT. DEPTNO AND EMPNO IN(7839, 7902)
AND LOC = 'NEW YORK';

3) OUTER JOINS :- It is used to obtain unmatched records.

(i) LEFT OUTER JOIN :- It is used to obtain unmatched records of left table along with matching records.

SYNTAX :-

ORACLE

```
SELECT Col.Name  
FROM TN1, TN2  
WHERE TN1.ColName = TN2.ColName (+);
```

ANSI

```
SELECT Col.Name  
FROM TN1 LEFT [OUTER] JOIN TN2  
ON <JOIN CONDITION>;
```

Example:-

Q) Wanted Name and DName of all the emp even though the employees don't work in any dept.

EMP	
ENAME	DEPTNO
A	10
B	Null
C	20
D	Null

LEFT

DEPT	
DNAME	DEPTNO
D ₁	10
D ₂	20
D ₃	30
D ₄	40

RIGHT

ENAME	EMP.DEPTNO	DNAME	DEPT.DEPTNO
A	10	D ₁	10
C	20	D ₂	20
B	Null	Null	Null
D	Null	Null	Null

ORACLE

```
S ENAME, DNAME  
F EMP, DEPT  
W EMP.DEPTNO = DEPT.DEPTNO (+);
```

ANSI

S ENAME, DNAME

```
F EMP LEFT OUTER JOIN DEPT  
ON EMP.DEPTNO = DEPT.DEPTNO;
```

Result

ENAME	DNAME
A	D ₁
C	D ₂
B	Null
D	Null

(iii) RIGHT OUTER JOIN :- It is used to obtain unmatched records of right table along with matching records.

SYNTAX :-

ORACLE

S Col-Name

F T.N₁, T.N₂

W T.N₁. Col-Name (+) = T.N₂. Col-Name ;

ANSI

S Col-Name

F T.N₁ RIGHT [OUTER] JOIN T.N₂
ON <JOIN Condition>;

Example:-

Want names and Dnames of all the emp even though there is no emp working in the dept.

EMP		DEPT	
ENAME	DEPTNO	DNAME	DEPTNO
A	10	D ₁	10
B	Null	D ₂	20
C	20	D ₃	30
D	Null	D ₄	40

LEFT RIGHT

ENAME	EMP.DEPENO	DNAME	DEPT.DEPENO
A	10	D ₁	10
C	20	D ₂	20
Null	Null	D ₃	30
Null	Null	D ₄	40

ORACLE

S ENAME, DNAME

F EMP, DEPT

W Emp.DEPENO (+) = DEPT.DEPENO;

ANSI

Result	
ENAME	DNAME
A	D ₁
C	D ₂
Null	D ₃
Null	D ₄

S ENAME, DNAME

F EMP RIGHT OUTER JOIN DEPT

ON Emp.DEPENO = DEPT.DEPENO;

(iii) FULL OUTER JOINS:-

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

SYNTAX:-

ANSI

```
SELECT colName  
FROM T_N1 FULL[OUTER] JOIN T_N2  
ON <Join-condition>.
```

Example

Q. What Names and Dnames of all the employees even though the employees don't work in any dept and dept having no. employees.

EMP		DEPT		Result	
ENAME	DEPTNO	DNAME	DEPTNO	ENAME	EMP.DEPTNO
A	10	D1	10	A	10
B	Null	D2	20	C	20
C	20	D3	30	B	Null
D	Null	D4	40	D	Null
				Null	Null
				Null	P ₃
				Null	P ₄

Q. ENAME, DNAME

F EMP.FULL JOIN DEPT

ON Emp. DEPTNO = DEPT. DEPTNO;

Result

ENAME	DNAME
A	D ₁
C	D ₂
B	Null
D	Null
Null	D ₃
Null	D ₄

4) NATURAL JOIN :-

- Whenever there is a relation b/w two tables it will act like inner join.
- Whenever there is no relation b/w two tables it will act like cross join.

Inner join

Syntax :-

ANSI

SELECT col.Name
FROM T-N₁ NATURAL JOIN T-N₂ ;

EMP	
ENAME	DEPTNO
A	10
B	20
C	30

DEPT	
DNAME	DEPTNO
D ₁	10
D ₂	20
D ₃	30

ENAME	DEPTNO	DNAME	DEPTNO
A	10	D ₁	10
B	20	D ₂	20
C	30	D ₃	30

Q. WRQTD ENAME, DNAME

RESULT :- Has a relation (INNER JOIN)

B ENAME, DNAME

F EMP NATURAL JOIN DEPT;

ENAME	DNAME
A	D ₁
B	D ₂
C	D ₃

Cross join

Syntax remain same.

EMP	
ENAME	DEPTNO
A	10
B	20
C	30

STU	
SNAME	SID
X	100
Y	101
Z	102

ENAME	DEPTNO	SNAME	SID
A	10	X	100
A	10	Y	101
A	10	Z	102
B	20	X	100
B	20	Y	101
B	20	Z	102
C	30	X	100
C	30	Y	101
C	30	Z	102

RESULT Has No relation (CROSS JOIN)

ENAME	SNAME
A	X
A	Y
A	Z
B	X
B	Y
B	Z
C	X
C	Y
C	Z

SELECT ENAME, SNAME
FROM EMP NATURAL JOIN STU;

Q. WAPTO all the details of employees.

S *

F EMP;

Q WAPTO ENAME, JOB, MGR, HIREDATE, SAL, COMM, DNAME, LOC,
and the employees who are working as SALESMAN,
MANAGER, CLERK, ANALYST in the DEPTNO 10,20,30 and
getting (SAL more than 10 , less than 5000 and those who hired
after 1980 and getting comm more than 2 .

S ENAME, JOB, MGR, HIREDATE, SAL, COMM, DNAME, LOC
F EMP, DEPT

W EMP. DEPTNO=DEPT.DEPTNO AND JOB IN ('SALESMAN', 'MANAGER',
'CLERK', 'ANALYST') AND DEPTNO IN (10,20,30) AND
SAL > 10 AND SAL < 5000 AND HIREDATE > '31-DEC-1980'
AND COMM > 2 ;

Q. WAPTO all the details of emp who are getting sal more
ALLEN less than KING and the emp hired after
MILLER and before ADAMS and those who are getting comm
more than TURNER but less than ALLEN and the
emp who are working in the DEPTNO 20,30 as
ANALYST, MANAGER, CLERK, SALESMAN and the emp who
are getting comm and the emp those who are not having
MGR and the emp names consist of 8 char .

S *

F EMP

W SAL > (S SAL)

F EMP

W ENAME = 'ALLEN' AND SAL < (S SAL)

F EMP

W ENAME = 'KING' AND HIREDATE > (S HIREDATE)

F EMP

W ENAME = 'MILLER' AND HIREDATE < (S HIREDATE)

F EMP

W ENAME = 'ADAMS' AND COMM > (S COMM)

F EMP

W ENAME = 'TURNER' AND COMM < (S COMM)

F EMP

W ENAME = 'ALLEN' AND DEPTNO IN (10,20,30) AND

JOB IN ('ANALYST', 'MANAGER', 'CLERK', 'SALESMAN') AND COMM IS NOT ^{NUL}
AND MGR IS NULL AND ENAME LIKE '-----'

Q. WHATD no of emp , max(SAL) , min(SAL) , sum(SAL) , AVH(SAL) of the employees who are working in each dept and getting sal > 1000 but SAL < 3000 and the employees who are working as SALESMAN, MANAGER, ANALYST, CLERK, in the DEPTNO 10,20,30 and the emp those who hired after 1965 those have ever not getting comm and not having the MGR.

S COUNT(X) , MAX(SAL) , MIN(SAL) , SUM(SAL) , AVH(SAL) , DEPTNO
F EMP
W SAL > 1000 AND SAL < 3000 AND JOB IN ('SALESMAN', 'MANAGER',
'ANALYST', 'CLERK') AND DEPTNO IN (10, 20, 30) AND
HIREDATE > '30-DEC-1965' AND COMM IS NULL AND
MGR IS NULL
GROUP BY DEPTNO ;

Q. WHATD MANAGER Name of MILLER .

S ENAME
F EMP
W RENAME = (S MGR
F EMP
W ENAME = 'MILLER');

Q. WHATD empname , emp.SAL , manager name , manager HIREDATE
manager.SAL , manager.manager name , manager's manager
SAL , manager's manager COMM and the emp who are
getting SAL < manager's manager and the manager's those
who hired after employee and the managers who are getting
SAL > employees and the emp who are working as manager, ANALYST,
and the managers who are working in the deptno (10, 20, 30) and the
managers' manager who are getting COMM less than manager and the
employees who are getting COMM < manager's manager and the
employee name consist of 4 char. And the managers name consist of
6 char and manager's manager name starts with 'A K' and

Q. WHAT IS MANAGER NAME OF MILLER

S. FNAME

F. FMP

W. FMPNO = (S.MLR

F. EMP

W. FNAME = 'MILLER');

* Q. WHAT IS Emp names and their Manager Name.

S.

4) SELF JOIN :-

1. Joining the same two tables is known as Self JOIN
OR

Joining a table by itself is known as SELF JOIN.

Q. When or why do we use SELF JOIN

- Whenever the data to be select is in the same table but present in the different records we use SELF JOIN

SYNTAX :-

ORACLE

SELECT Col. Name
FROM T.N₁ T₁, T.N₂ T₂
WHERE <JOIN-Condition>;

{T₁, T₂ are
alias name
to the table}

ANSI

SELECT Col. Name
FROM T.N₁ T, JUIN T.N₂ T₂
ON <Join-Condition>;

Q. WHAT IS ENAME and Manager Name of all the employee.

EMP (F₁)

EMP (F₂)

EMPNO	ENAME	MGR
1	A	2
2	B	3
3	C	4
4	D	

EMPNO	ENAME	MGR
1	A	2
2	B	3
3	C	4
4	D	

$$F_1.MGR = F_2.EMPNO$$



F ₁ .EMPNO	F ₂ .ENAME	F ₂ .EMPNO	F ₂ .ENAME	F ₁ .MGR	F ₂ .MGR
1	A				
2	B				
3	C				

Some
table
exists
Second

$E_1 \cdot EMPNO$	$E_1 \cdot ENAME$	$E_1 \cdot MGR$	$E_2 \cdot EMPNO$	$E_2 \cdot ENAME$	$E_2 \cdot MGR$
1	A	2	2	B	3
2	B	3	3	C	4
3	C	4	4	D	

↓
RESULT

$E_1 \cdot ENAME$	$E_2 \cdot ENAME$
A	B
B	C
C	D

Ans → $\$ E_1 \cdot ENAME, E_2 \cdot ENAME$

$F \text{ EMP } E_1, \text{ EMP } E_2$

$W \text{ } E_1 \cdot MGR = E_2 \cdot EMPNO;$

Q. WAP TO EmpName, SAL, managerName, managerSAL

$\$ E_1 \cdot ENAME, E_1 \cdot SAL, E_2 \cdot ENAME, E_2 \cdot SAL$

$F \text{ EMP } E_1, \text{ EMP } E_2$

$W \text{ } E_1 \cdot MGR = E_2 \cdot EMPNO;$

Q. WAP TO ENAME, EmpSAL, managerName, managerSAL and the emp who are getting sal more than 1000 and managers who are getting sal more than 2000

$\$ E_1 \cdot ENAME, E_1 \cdot SAL, E_2 \cdot ENAME, E_2 \cdot SAL$

$F \text{ EMP } E_1, \text{ EMP } E_2$

$W \text{ } E_1 \cdot MGR = E_2 \cdot EMPNO \text{ AND } E_1 \cdot SAL > 1000 \text{ AND } E_2 \cdot SAL > 2000;$

Q. WAP TO ENAME, E.HIREDATE, EmpSAL, EmpComm, managerName, managerComm, MGR.HIREDATE, and the employees who are getting comm and manager are hired after 1980

ORACLE $\$ E_1 \cdot ENAME, E_1 \cdot HIREDATE, E_2 \cdot SAL, E_1 \cdot COMM, E_2 \cdot ENAME, E_2 \cdot COMM, E_2 \cdot HIREDATE$

$F \text{ EMP } E_1, \text{ EMP } E_2$

$W \text{ } E_1 \cdot MGR = E_2 \cdot EMPNO \text{ AND } E_1 \cdot COMM \text{ IS NOTNULL AND } E_2 \cdot HIREDATE > '31-DEC-1980';$

ANSJ

g F₁.ENAME, F₁.HIREDATE, F₁.SAL, F₁.COMM, F₂.ENAME
 F₂.COMM, F₂.HIREDATE

F EMP F₁, EMP F₂

ON F₁.MGR = F₂.EMPNO AND F₁.COMM IS NOT NULL
 AND F₂.HIREDATE > '31-DEC-1980';
 OR we can use where clause in ANSI
 ON F₁.MGR = F₂.EMPNO
 WHERE F₁.COMM IS NOT NULL AND F₂.HIREDATE > '31-DEC-1980';

Assignment on SELF JOIN

1. what is name of the emp and his manager's name if employee is working as CLERK

g F₁.ENAME, F₂.ENAME

f EMP F₁, EMP F₂

w F₁.MGR = F₂.EMPNO AND F₁.JOB = 'CLERK');

2. what is Name of the emp and manager's designation if manager works in DEPT 10 or 20

g F₁.ENAME, F₂.JOB

f EMP F₁, EMP F₂

w F₁.MGR = F₂.EMPNO AND F₂.DEPTNO IN (10, 20)

Q.3. what is Name of the emp and manager's salary if emp and manager both earn more than 2300

g F₁.ENAME, F₂.SAL

f EMP F₁, EMP F₂

w F₁.MGR = F₂.EMPNO AND F₁.SAL > 2300 AND F₂.SAL > 2300;

Q.4. what is emp Name and manager's HIREDATE if employee was hired before 1982.

g F₁.ENAME, F₂.HIREDATE

f EMP F₁, EMP F₂

w F₁.MGR = F₂.EMPNO AND F₁.HIREDATE < '01-JAN-1982';

Q.5. what is emp Name manager's comm if emp work as salesman and manager works in DEPT 30.

g F₁.ENAME, F₂.COMM

f EMP F₁, EMP F₂

w F₁.MGR = F₂.EMPNO AND F₁.JOB = 'SALESMAN' AND

F₂.DEPTNO = 30;

Q.6. What is empName and manager name and their salaries if employee earns more than manager.

S E₁.ENAME , E₂.ENAME , E₁.SAL AS EMPSAL , E₂.SAL AS MGRSAL
F EMP E₁ , EMP E₂
W E₁.MGR = E₂.EMPNO AND E₁.SAL > E₂.SAL ;

Q.7 - What is EMP Name and his date manager name and his date if manager was hired before employee.

S E₁.ENAME , E₂.HIREDATE , E₂.ENAME , E₂.HIREDATE
F EMP E₁ , Emp E₂
W E₁.MGR = E₂.EMPNO AND E₂.HIREDATE < E₁.HIREDATE;

Q.8 - What is Emp Name and Manager Name if both are working in same job

S E₁.ENAME , E₂.ENAME
F EMP E₁ , EMP E₂
W E₁.MGR = E₂.EMPNO AND E₁.JOB = E₂.JOB ;

Q.9. What is EmpName and Manager Name if manager is working as actual manager.

S E₁.ENAME , E₂.ENAME
F EMP E₁ , EMP E₂
W E₁.MGR = E₂.EMPNO AND E₂.JOB = 'MANAGER' ;

Q.10. What is empName and manager name along with their annual salaries if employee works in DEPT10, 20 and MANAGER's sal is greater than employee salary.

S E₁.ENAME , E₁.SAL*12 , E₂.ENAME , E₂.SAL*12
F EMP E₁ , EMP E₂
W E₁.MGR = E₂.EMPNO AND E₂.JOB IN (10, 20)
AND E₂.SAL > E₁.SAL ;

Q.11. What is employees name and manager designation for all the employees.

S E₁.ENAME , E₂.JOB
F EMP E₁ , Emp E₂
W E₁.MGR = E₂.EMPNO ;

Q.12 WAPTD employee's name and manager's sal for all the employees if manager's sal ends with 50.

S E_1 .FNAME, E_2 .SAL

F EMP E_1 , EMPF₂

W E_1 .MGR = E_2 .EMPNO AND E_2 .SAL LIKE '%.50';

Q. WAPTD ENAME and Manager Name of all the employees.

~~S E_1 .ENAME; E_2 .ENAME~~

S E_1 .ENAME, E_2 .ENAME

F EMP E_1 , EMPF₂

W E_1 .MGR = E_2 .EMPNO;

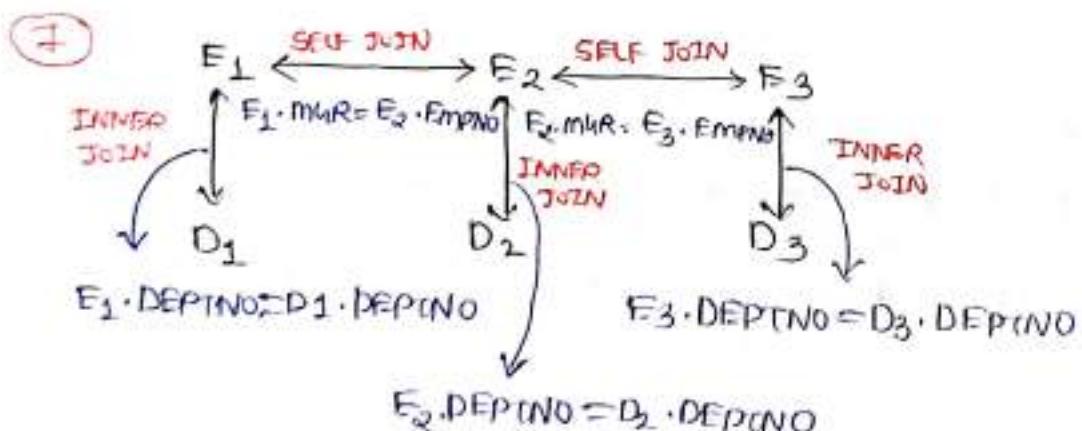
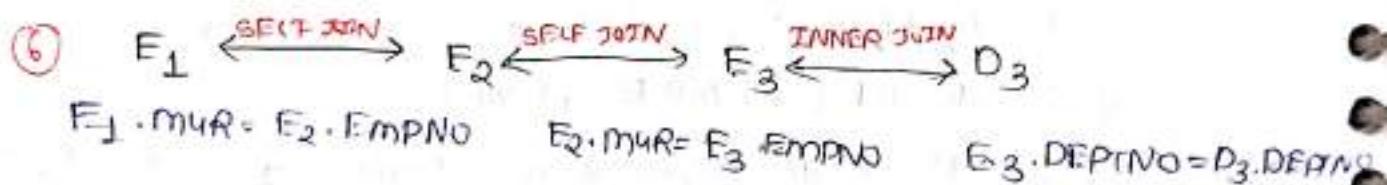
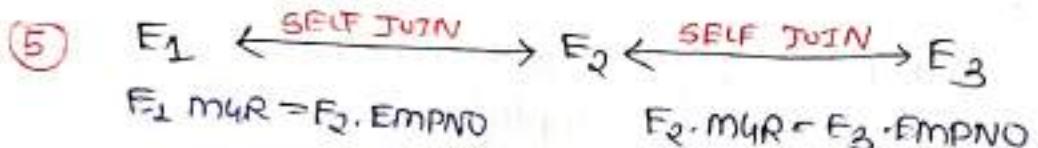
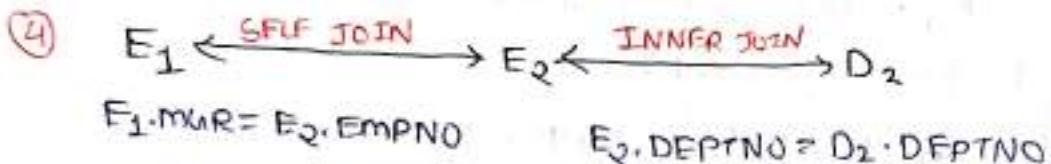
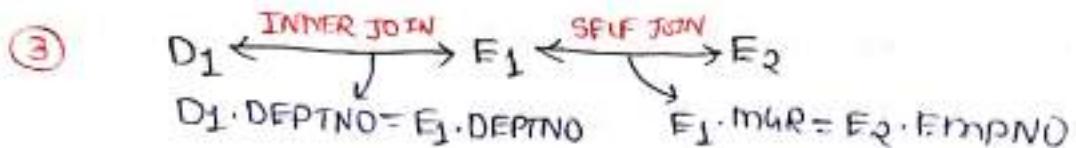
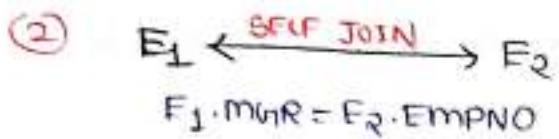
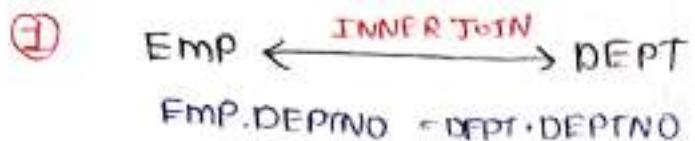
Q. WAPTD EmpName , Manager Name, Manager's manager name for all the employees .

S E_1 .ENAME, E_2 .ENAME, E_3 .ENAME

F EMPF₁, EMPF₂, EMPF₃

W E_1 .MGR = E_2 .EMPNO AND E_2 .MGR = E_3 .EMPNO;

JOINING MULTIPLE TABLES



- Q. WAP QTD ENAME , manager Name , manager's manager name
 S E1.ENAME , F2.ENAME , F3.ENAME
 F EMP E1 , EMP E2 , EMP E3
 W E1.MGR = E2.EMPNO AND E2.MGR = E3.EMPNO ;

Q. WAP TO ENAME , Manager Name, Manager's Manager Name ,
Manager's Manager's Manager name .

S E₁.ENAME , E₂.ENAME , E₃.ENAME , E₄.ENAME

F EMP E₁ , EMP E₂ , EMP E₃ , EMP E₄

W E₁.MGR = E₂.EMPNO AND E₂.MGR = E₃.EMPNO AND
E₃.MGR = E₄.EMPNO ;

Joining multiple tables.

1. WAP TO ENAME , MANAGER NAME and Loc for the emp
Working as CLERK

S E₁.ENAME , E₂.ENAME , ^{ORACLE} D₁
~~E₃~~.LOC

{ " we can
write D₃
in the place
of E₃ }

F EMP E₁ , EMP E₂ , DEPT ~~D₁~~

W E₁.MGR = E₂.EMPNO AND E₁.DEPTNO = ^{D₁} ~~E₃~~.DEPTNO
AND E₁.JOB = 'CLERK' ;

ANSI

S E₁.ENAME , E₂.ENAME , F₃.LOC

F EMP E₁ JOIN EMP E₂ INNER EMP E₃

ON E₁.MGR = E₂.EMPNO AND E₂.DEPTNO = E₃.DEPTNO
AND E₁.JOB = 'CLERK' ;

2. WAP TO ENAME , Manager's salary and Manager's location
manager working as president

S E₃.ENAME , E₂.SAL

ANSI - 1

S E₁.ENAME , E₂.ENAME , D₁.LOC

F EMP E₁ JOIN EMP E₂

ON E₁.MGR = E₂.EMPNO INNER JOIN DEPT D₁

ON E₁.DEPTNO = D₁.DEPTNO

WHERE E₁.JOB = 'CLERK' ;

Q.2. WANTED ENAME, MANAGER'S NAME, salary and manager's location if manager working as president.

S E₁.ENAME, E₂.SAL, D₁.LOC
F EMP E₁, EMP E₂, DEPT D₁
W E₁.MGR = E₂.EMPNO AND E₂.DEPTNO = D₁.DEPTNO;
AND E₂.JOB = 'PRESIDENT';

ANSI

S E₁.ENAME, E₂.SAL, D₁.LOC
F EMP E₁ JOIN EMP E₂
ON E₁.MGR = E₂.EMPNO INNER JOIN DEPT D₁
ON E₂.DEPTNO = D₁.DEPTNO
W E₂.JOB = 'PRESIDENT';

Q.3. WANTED ENAME, EMPSAL, MANAGER'S NAME, MANAGER'S SAL along with Emp DNAME if Emp EARNS more than MANAGER.

ORACLE

S E₁.ENAME, E₁.SAL, E₂.ENAME, E₂.SAL, D₁.DNAME
F EMP E₁, EMP E₂, DEPT D₁
W E₁.MGR = E₂.EMPNO AND E₁.DEPTNO = D₁.DEPTNO
AND E₁.SAL > E₂.SAL;

ANSI

S E₁.ENAME, E₁.SAL, E₂.ENAME, E₂.SAL, D₁.DNAME
F EMP E₁ JOIN EMP E₂
ON E₁.MGR = E₂.EMPNO INNER JOIN DEPT D₁
ON E₁.DEPTNO = D₁.DEPTNO
W E₁.SAL > E₂.SAL;

Q4. WAPTO ENAME, SAL, Manager's Name ,Manager's
DNAME If MANAGER works in NEWYORK with salary
more than 3000.

ORACLE

S E₁.ENAME, E₁.SAL, E₂.ENAME, D₂.DNAME
F EMP E₁, EMP E₂, DEPT D₂
W E₁.MGR = E₂.EMPNO AND E₂.DEPTNO = D₂.DEPTNO AND
E₂.JOB = 'M' D₂.LOC = 'NEWYORK' AND E₂.SAL > 3000;
ANSI

S E₁.ENAME, E₂.SAL, E₂.ENAME, D₂.DNAME
F EMP E₁ JOIN EMP E₂
ON E₁.MGR = E₂.EMPNO INNER JOIN DEPT D₂
ON E₂.DEPTNO = D₂.DEPTNO
W E₂.SAL > 3000 AND D₂.LOC = 'NEWYORK';

Q5. WAPTO ENAME ,EMP DNAME ,MANAGER'S Name ,MANAGER'S
DNAME if emp and manager works in same loc.

S E₁.ENAME, E₂.ENAME, D₁.DNAME, D₂.DNAME
F EMP E₁, EMP E₂, DEPT D₁, DEPT D₂
W E₁.MGR = E₂.EMPNO AND E₁.DEPTNO = D₁.DEPTNO AND
E₂.DEPTNO = D₂.DEPTNO AND D₁.LOC = D₂.LOC;
ANSI

S E₁.ENAME, E₂.ENAME, D₁.DNAME, D₂.DNAME
F EMP E₁ JOIN EMP E₂
ON E₁.MGR = E₂.EMPNO INNER JOIN DEPT D₁
ON E₁.DEPTNO = D₁.DEPTNO INNER JOIN DEPT D₂
ON E₂.DEPTNO = D₂.DEPTNO
W D₁.LOC = D₂.LOC;

Q.6. What is ENAME, EMP.HIREDATE, MANAGER'S HIREDATE, if manager was hired before the emp is to accounting dept.

ORACLE

S E₁.ENAME , E₁.HIREDATE , E₂.HIREDATE

F EMP E₁ , EMP E₂ , DEPT D₂

W E₁.MGR = E₂.EMPNO AND E₂.HIREDATE < E₁.HIREDATE
AND E₂.DEPTNO = D₂.DEPTNO AND D₂.DNAME = 'ACCOUNTING';

ANSI

S E₁.ENAME , E₁.HIREDATE , E₂.HIREDATE

F EMP E₁ JOIN EMP E₂

ON E₁.MGR = E₂.EMPNO INNER JOIN DEPT D₂

ON E₂.DEPTNO = D₂.DEPTNO

W E₂.HIREDATE < E₁.HIREDATE AND D₂.DNAME = 'ACCOUNTING';

Q.7 What is emp name manager's name along with manager's manager name.

S E₁.ENAME , E₂.ENAME , E₃.ENAME

F EMP E₁ , EMP E₂ , EMP E₃

W E₁.MGR = E₂.EMPNO AND AND E₂.MGR = E₃.EMPNO ;

ANSI

S E₁.ENAME , E₂.ENAME , E₃.ENAME

F EMP E₁ JOIN EMP E₂

ON E₁.MGR = E₂.EMPNO JOIN EMP E₃

ON E₂.MGR = E₃.EMPNO ;

Q.8. WRITE ENAME, DNAME, MANAGERNAME, his DEPT NAME if
MANAGER is HIRED before 1982.

S E₁.ENAME, E₂.ENAME, D₁.DNAME, D₂.DNAME

F EMP E₁, EMP E₂, DEPT D₁, DEPT D₂

W E₁.MGR = E₂.EMPNO AND E₁.DEPTNO = D₁.DEPTNO
AND E₂.DEPTNO = D₂.DEPTNO AND
E₂.HIREDATE < '31-DEC-82';
ANSI

S E₁.ENAME, E₂.ENAME, D₁.DNAME, D₂.DNAME

F EMP E₁ JOIN EMP E₂

W E₁.MGR = E₂.EMPNO INNER JOIN DEPT D₁

ON E₁.DEPTNO = D₁.DEPTNO INNER JOIN DEPT D₂
ON E₂.DEPTNO = D₂.DEPTNO

W E₂.HIREDATE < '31-DEC-82'

Q.9. WRITE EMP NAME with DNAME, MANAGER's Name
with Location along with manager's manager name
with loc.

S E₁.ENAME, E₂.ENAME, D₁.DNAME, D₂.LOC,
E₃.ENAME, D₃.LOC

F EMP E₁, EMP E₂, DEPT D₁, DEPT D₂, EMP E₃,
DEPT D₃

W E₁.MGR = E₂.EMPNO AND E₁.DEPTNO = D₁.DEPTNO
AND E₂.DEPTNO = D₂.DEPTNO AND E₂.MGR = E₃.EMPNO
AND E₃.DEPTNO = D₃.DEPTNO;
ANSI

S E₁.ENAME, E₂.ENAME, D₁.DNAME, D₂.LOC, E₃.ENAME, D₃.LOC

F EMP E₁ JOIN EMP E₂

ON E₁.MGR = E₂.EMPNO INNER JOIN DEPT D₁

ON E₁.DEPTNO = D₁.DEPTNO INNER JOIN DEPT D₂

ON E₂.DEPTNO = D₂.DEPTNO INNER JOIN EMP E₃

ON E₂.MGR = E₃.EMPNO INNER JOIN DEPT D₃

ON E₃.DEPTNO = D₃.DEPTNO;

Q10 WAPOTO EMP NAME , DNAME , MANAGER'S SAL with
DNAME along with MANAGER's MANAGER JOB with
DNAME if emp work in DEPTNO, MANAGER work
as PRESIDENT or actual manager and manager's
Manager salary must be greater or equal to
manager's salary.

S $E_1.\text{ENAME}, E_2.\text{SAL}, E_3.\text{JOB}, D_1.\text{DEPT DNAME},$
 $D_2.\text{DNAME}, D_3.\text{DNAME}$

F $\text{EMP } E_1, \text{EMP } E_2, \text{EMP } E_3, \text{DEPT } D_1, \text{DEPT } D_2,$
 $\text{DEPT } D_3$

W $E_1.\text{MGR} = E_2.\text{EMPNO} \text{ AND } E_2.\text{MGR} = E_3.\text{EMPNO} \text{ AND }$
 $E_1.\text{DEPT} = D_1.\text{DEPT} \text{ AND } E_2.\text{DEPT} = D_2.\text{DEPT} \text{ AND }$
 $E_3.\text{DEPT} = D_3.\text{DEPT} \text{ AND } E_1.\text{DEPTNO} \text{ IN } 10$
 $\text{AND } E_2.\text{JOB} \text{ IN } ('PRESIDENT', 'MANAGER')$
 $\text{AND } E_3.\text{SAL} \geq E_2.\text{SAL};$

ANSI

S $E_1.\text{ENAME}, E_2.\text{SAL}, E_3.\text{JOB}, D_1.\text{DNAME}, D_2.\text{DNAME},$
 $D_3.\text{DNAME}$

F $\text{EMP } E_1 \text{ JOIN } \text{EMP } E_2$
 $\text{ON } E_1.\text{MGR} = E_2.\text{EMPNO} \text{ JOIN } \text{EMP } E_3$
 $\text{ON } E_2.\text{MGR} = E_3.\text{EMPNO} \text{ INNER JOIN } \text{DEPT } D_1$
 $\text{ON } E_1.\text{DEPT} = D_1.\text{DEPT} \text{ JOIN } \text{DEPT } D_2$
 $\text{ON } E_2.\text{DEPT} = D_2.\text{DEPT} \text{ JOIN } \text{DEPT } D_3$
 $\text{ON } D_3.\text{DEPT} = D_3.\text{DEPT}$

W $E_1.\text{DEPTNO} = 10 \text{ AND } E_2.\text{JOB} \text{ IN } ('PRESIDENT',$
 $'MANAGER') \text{ AND } E_3.\text{SAL} \geq E_2.\text{SAL};$

$$\sqrt{-1} = \sqrt{(-1) \times (-1)}$$

Q. WORKTD emphname , DEPTNAME , Manager Name , Manager's DNAME
Manager's Manager Name , Manager's Manager location and
the emp who are working in deptNo 10 , and the
manager who are working in deptNo 20 and the
Manager manager who are getting sal more than 1000

S $F_1 \cdot \text{FNAME}$, $E_2 \cdot \text{FNAME}$, $F_3 \cdot \text{FNAME}$, $D_1 \cdot \text{DNAME}$,
 $D_2 \cdot \text{DNAME}$, $D_3 \cdot \text{LOC}$

F Emp F_1 , EMP E_2 , EMP F_3 , DEPT D_1 ,
DEPT D_2 , DEPT D_3

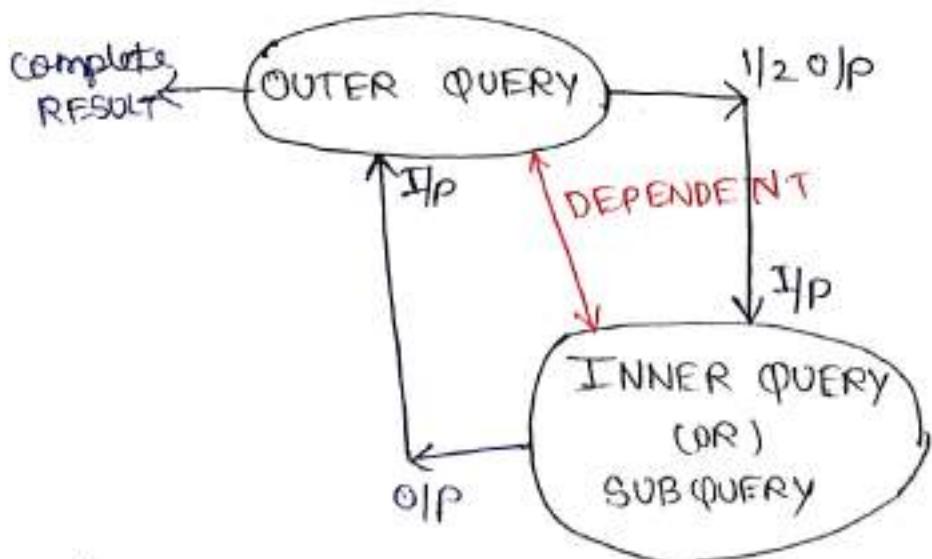
W $F_1 \cdot \text{MUR} = F_2 \cdot \text{EMPNO}$ AND $F_2 \cdot \text{MUR} = F_3 \cdot \text{EMPNO}$
AND $F_1 \cdot \text{DEPTNO} = D_1 \cdot \text{DEPTNO}$ AND $D_2 \cdot \text{DEPTNO} =$
 $F_2 \cdot \text{DEPTNO}$ AND $F_3 \cdot \text{DEPTNO} = D_3 \cdot \text{DEPTNO}$
AND $F_1 \cdot \text{DEPTNO} = 10$ AND $F_2 \cdot \text{DEPTNO} = 20$
AND $F_3 \cdot \text{SAL} > 1000$;

CORRELATED SUBQUERY :-

A query written inside another query and inner query and outer query are dependent on each other is known as correlated subquery.

Working Process of Correlated Subquery :-

- (i) Outer query will execute first and produces the partial output.
- (ii) The partial output of outer query will be passed as a input to the innerquery.
- (iii) And then inner query will execute and produces the output.
- (iv) The output of inner query will be given as input to the outer query.
- (v) And then outer query will execute and produces the result or complete result.
- (vi) Therefore the outer query and inner query are dependent of on each other.



Example:- want a QTD DNAME , in which emp are working

* DNAME
F DEPT D
W D·DEPTNO = (S^E·DEPTNO
F EMP E
W D·DEPTNO=E·DEPTNO) ;

- NOTE :-
- (i) Correlated subquery is working with the principle of both subquery as well as joins.
 - (ii) Whenever we are using correlated subquery we should write join condition.
 - (iii) The join condition should be written inside the inner query.

EMP

FNAME	DEPTNO
A	10
B	20
C	10, 20
D	40

DEPT

DEPTNO	DEPTNO
D1	10
D2	20
D3	30
D4	40

Q. What's DNAME in which emp are working.

Final
Step to answer
Outer query

10, 20, 30] Inner query o/p

S DNAME

F DEPT

DEPTNO IN (S DEPTNO)

10

20

30

40
Partial output

→ F Emp

Emp. DEPTNO = DEPT. DEPTNO

10 → 10

20 → 20

10 → 30

30 → 40

STEPS:
 1. S DNAME
 10=10
 10=20
 10=30
 10=40
 20=10
 20=20
 20=30
 20=40
 10=10
 10=20
 10=30
 10=40
 30=10
 30=20
 30=30
 30=40

Q. What's DNAME in which emp are not working

S DNAME

F DEPT NOT

W DEPTNO, IN (S DEPTNO)

F EMP

W Emp. DEPTNO = DEPT. DEPTNO;

EXISTS OPERATOR

⇒ It returns true if the subquery returns any value other than null.

Example:-

WHATD DNAME in which employees are working.

S DNAME

F DEPT

W DEPTNO EXISTS (S DEPTNO

F EMP

W EMP.DEPTNO = DEPT.DEPTNO

NOT EXISTS

⇒ NOT EXISTS operator returns true if subquery returns null.

Example:-

WHATD DNAME in which employees are not working

S DNAME

F DEPT

W DEPTNO NOT EXISTS (S DEPTNO

F EMP

W EMP.DEPTNO ≠ DEPT.DEPTNO;

SIMILAR Row FUNCTIONS:-

1) LENGTH()	10) MOD()	19) RANK()
2) CONCAT()	11) ROUND()	20) DENSE_RANK()
3) UPPER()	12) TRUNC()	21) CASE()
4) LOWER()	13) MONTHS_BETWEEN()	22) SYSDATE / CURRENT_DATE
5) INITCAP()	14) TO_CHAR()	/ CURRENT_DATE
6) REVERSE()	15) LAST_DAY()	
7) REPLACE()	16) NVL() (NULL VALUE Logic)	
8) SUBSTR()	17) To_DATE()	
9) INSTR()	18) ADD_MONTHS()	

1) LENGTH :-

- This function is used to count the no of char present in the given string.
- It will count all the characters including the space present within single quotes.

SYNTAX :-

LENGTH('STRING')

Example:-

SELECT LENGTH('SOMIT')
FROM DUAL;

> O/P
5

SELECT LENGTH('SU mIT')
FROM DUAL

> O/P
6

Q. What is * of the emp. having exactly 5 char in their name without using like operator.

{*: Dual is used to empty table present in DBMS with one row and one column}

X SELECT * , LENGTH(FNAME)
FROM EMP ;

→ 5 space.

✓
B *
F EMP

↳ LENGTH(FNAME) = 5 ;

Q. What details of the employees having more than 4 char in their names without using like operator.

B *
F EMP

↳ LENGTH(FNAME) > 4 ;

NOTE :-

DUAL TABLE :- It is a dummy table which is used to display the result.

2) CONCAT :-

06/09/23

- This function is used to join the given two strings
- It behaves like concatenation operator.

SYNTAX

CONCAT ('STRING1', 'STRING2')

\$ CONCAT ('USRAT', 'NNUSKA')
F DUAL;

Example:-

O/P

USRATTNNUSKA

3) UPPER :-

UPPER ('STRING')

Example

\$ UPPER ('jspiders')
F DUAL;

O/P

JSPIDERS

4) LOWER :-

- This function is used to convert the given string into lower case.

SYNTAX

LOWER ('STRING')

Example

\$ LOWER ('JSPIDERS')
F DUAL;

O/P

jspiders.

5) INITCAP :-

- This function is used to convert the given string first character into upper case.

SYNTAX

INITCAP ('STRING')

Example:-

\$ INITCAP ('j SPIDER')
F DUAL;

O/P

Jspider

6) REVERSE :-

This function is used to convert the given string into reverse order.

SYNTAX

REVERSE ('STRING')

Example

S REVERSE ('JSPIDERS')
F DUAL;

O/P

SREDIPSJ

7) REPLACE :-

This function is used to replace a given string with ^{the} new string.

SYNTAX

REPLACE('ORIGINAL-STR', 'OLD-STR', [NEW-STR])

Ex:-

SELECT REPLACE ('JSPIDERS', 'J', 'P')
FROM DUAL;
or
↳ O/P PSPIIDERS

S REPLACE ('JSPIDERS', 'J')
F DUAL;
↳ O/P SPIDERS

Q. WHAT IS no. of 'S' present in the JSPIDERS
X S LENGTH ('JSPIDERS')

S LENGTH ('JSPIDERS') - LENGTH (REPLACE ('JSPIDERS', 'S'))
F DUAL;
↳ O/P 2

Q. WHAT IS no. of 'E' present in Engineering.
S LENGTH ('ENGINEERING') - LENGTH (REPLACE ('ENGINEERING',

F DUAL;
↳ O/P 3

Q WHERE Name of the emp if character 'A' is present exactly once in the employee name.

S ENAME , LENGTH(ENAME) - LENGTH(REPLACE

S ENAME

F EMP

W LENGTH(ENAME) - LENGTH(REPLACE(ENAME, 'A')) = 1;

O/P →

Q WHERE Ename if char 'L' is present exactly twice in the Ename

S ENAME

F EMP

W LENGTH(ENAME) - LENGTH(REPLACE(ENAME, 'L')) = 2;

⇒ MOD :-

This function is used to obtain remainder of the given numbers

SYNTAX

MOD(M,N)

NOTE :-

- It is in the form of M/N

Example

S MOD(83,6)

O/P 5

F DUAL;

OR

S MOD(9,3)

O/P 0,

F DUAL;

⇒ ROUND :-

This function is used to round off a given number to the nearest value.

SYNTAX

ROUND (NUMBER, [SCALE])

Example.

ROUND(3.6) = 4

ROUND(3.8) = 4

ROUND(4.1) = 4

ROUND(4.5) = 5

\Rightarrow TRUNC :-

This function is used to give a number to the lower value.

SYNTAX

TRUNC (NUMBER, [SCALE])

Ex:-

TRUNC (3.6) = 3

TRUNC (3.8) = 3

TRUNC (4.1) = 4

TRUNC (5.4) = 5

\Rightarrow MONTHS_BETWEEN :-

This function is used to obtain the no. of months b/w the given two dates.

SYNTAX

MONTHS-BETWEEN (DATE1, DATE2)

Ex:-

S MONTHS-BETWEEN ('14-FEB-2020', '03-NOV-2023')
 F DUAL ; o/p
 -44.66 ..

S ROUND (MONTHS-BETWEEN ('14-FEB-20', '03-NOV-23'))
 F DUAL o/p
 -45.

S ABS (ROUND (MONTHS-BETWEEN ('14-FEB-20', '03-NOV-23')))
 F DUAL o/p
 45

single row function

ABS is absolute function used to make the value abs-value
or to display the abs value

\Rightarrow ADD_MONTHS :-

This function is used to add the no. of months for the given date.

SYNTAX

ADD_MONTHS (DATE, INTEGER)

Ex:-

S ADD_MONTHS(SYSDATE, 9)
F DUAL;

O/p

06-JUN-24

⇒ SYSDATE OR CURRENT_DATE :-

This function is used to obtain the current date of the system
(todays date).

Ex:- S SYSDATE

O/p

F DUAL;

06-SEP-23

or

S CURRENT_DATE

O/p

F DUAL;

06-SEP-23.

⇒ LAST_DAY :-

This function is used to obtain the last day of the month month in the given date.

SYNTAX

LAST_DAY(DATE)

Ex:-

S LAST_DAY(SYSDATE)

→ O/p

F DUAL;

30-SEP-23

or

S LAST_DAY('02-NOV-23')

→ O/p

F DUAL;

31-NOV-23

$\Rightarrow \text{SUBSTR}()$:-

This function is used to obtain a part of string from the given original string.

SYNTAX :-

SUBSTR

$\text{SUBSTR('ORIGINAL_STR', POSITION, [LENGTH])}$

-5 -4 -3 -2 -1
S U M I T
1 2 3 4 5

$\text{SUBSTR('SUMIT', 3, 2)} = MI$

$\text{SUBSTR('sumit', 4, 2)} = IT$

$\text{SUBSTR('sumit', 2)} = UMIT$

$\text{SUBSTR('sumit', -2, 1)} = I$

$\text{SUBSTR('sumit', -4, 2)} = UM$

Q. What is first 3 char of the ENAME

B SUBSTR(ENAME, 1, 3)
F EMP;
or

Q. What is last two char of the EMPName.

B SUBSTR(ENAME, -2, 2)
F EMP;
or

B SUBSTR(ENAME, -2)
F EMP;

Q. What is fixed half of the ENAMES

B SUBSTR(ENAME, 1, LENGTH(ENAME)/2)
F EMP;

Q. What is second half of the ENAMES

B SUBSTR(ENAME, LENGTH(ENAME)/2 + 1)
F EMP;

SMITH, 8/2 + 1

SMITH, 2 + 1

SMITH, 3

[ITH]

How to

Q. WHAT IS ENAME if the second character of the name is I

S SUBSTR (ENAME, 2, 1) = I

S ENAME

F EMP

W SUBSTR (ENAME, 2, 1) = 'I';

Q. WHAT IS * OF THE EMP IF THE LAST 3 CHAR OF THE JOB IS 'ERK'

S *

F EMP

W SUBSTR (JOB, -3 [3]) = 'ERK';

⇒ TO-DATE :-

This function is used to convert the given string into date format.

SYNTAX

TO-DATE ('DATE-STRING')

S TO-DATE (SYSDATE)+10

07-SEP-23

F DUAL;

O/P

17-SEP-23

Q. WHAT IS HIREDATE OF THE EMP BY ADDING +4

S KEEP TO-DATE ('HIREDATE') +4

F EMP;

$\Rightarrow \text{RANK}()$:-

- This function is used to assign the rank based on how the column is arranged.
- Rank function repeats the rank whenever we have same value and it also skips the ranks if we have repeated values.

SYNTAX :-

$\text{RANK}()$ OVER (ORDER BY col.Name ASC/DESC)

Example:-

\$ RANK() OVER(ORDER BY SAL DESC) RANK, SAL

F Emp;

ALIAS

Original Table.

SAL	RANK	SAL
2000	1	5000
1000	2	4000
3000	2	4000
2000	4	3000
5000	4	3000
4000	6	2000
3000	6	2000
4000	8	1000

Result Table.

DENSE_RANK() :-

- This function is used to assign the rank based on how the column is assigned.
- Dense-Rank function repeats the rank whenever we have same values and it will not skip the rank if we have repeated values.

SYNTAX :-

$\text{DENSE_RANK}()$ OVER (ORDER BY col.Name ASC/DESC)

Ex:-

\$ DENSE_RANK() OVER(ORDER BY SAL DESC) RANK, SAL
F Emp;

RANK	SAL
1	5000
2	4000
2	4000
3	3000
3	3000
4	2000
4	2000
5	1000

Result Table.

→ CASE :-

- This function is used to display the desired output if the condition is satisfied, satisfied.

SYNTAX :-

```
CASE WHEN <CONDITION> THEN 'RESULT 1'  
      WHEN <CONDITION> THEN 'RESULT 2'  
      WHEN <CONDITION> THEN 'RESULT 3'  
      ELSE 'RESULT 4'  
END ;
```

Example:-

```
↳ SAL, CASE WHEN SAL<1000 THEN 'LESS SAL'  
                WHEN SAL>2000 THEN 'MORE SAL'  
                ELSE 'OK OF SAL'  
              END  
P Emp  
OF ORDER BY SAL DESC ;
```

→ TO_CHAR():-

- This function is used to convert the given date to char or string format.

SYNTAX :-

```
TO_CHAR (DATE , 'FORMAT-MODEL')
```

Format models :-

- YYYY
- HH12
- YEAR
- MI
- YY
- SS
- MM
- DD
- MONTH
- HH12:MI:SS
- DD-MM-YYYY
- DD-MM-YY

Example:-

(1) $\$ \text{TO-CHAR}(\text{SYSDATE}, \& \text{MODEL})$

F DUAL;

ENTER value for model: 'DD-MM-YY'

(2) $\$ \text{TO-CHAR}(\text{SYSDATE}, \& \text{MODEL})$

F DUAL;

Enter value for model: 'DD-MM-YY, HH12:MI:SS' \Rightarrow

$\Rightarrow NVL()$:- (Null Value Logic)

It is used to overcome the drawbacks of NULL

Syntax :-

$NVL(\text{ARG}_1, \text{ARG}_2)$

ARG_1 :- column name or expression which may results NULL

ARG_2 :- Write a numeric value

Example:-

Q. Wanted total sal given to each emp

EMP

ENAME	SAL	COMM	
A	3000	NULL	= 3000
B	2500	500	= 3000
C	4000	NULL	= 4000

$\$ \text{SAL} + \text{NVL}(\text{COMM}, 0)$

F EMP;

Working :-

(1) $\Rightarrow 3000 + \text{NVL}(\text{NULL}, 0)$

$$\begin{array}{l} 3000 + 0 \\ \hline 3000 \end{array}$$

(2) $\Rightarrow 2500 + \text{NVL}(500, 0)$

$$\begin{array}{l} 2500 + 500 \\ \hline 3000 \end{array}$$

(3) $\Rightarrow 4000 + \text{NVL}(\text{NULL}, 0)$

$$\begin{array}{l} 4000 + 0 \\ \hline 4000 \end{array}$$

⇒ INSTR(STR) :- IN STR(STR)

This function is used to obtain the position in which the string is present in original string.

SYNTAX :-

INSTR('ORIGINAL STR', 'STR', POSITION, [OCCURANCE])

Ex:-

E N G I N E E R I N G
① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩ ⑪

g INSTR ('ENGINEERING', 'E', 1, 2) ⇒ 6
F DUAL;

g INSTR ('ENGINEERING', 'E', 2, 2) ⇒ 7

g INSTR ('ENGINEERING', 'I', 2, 2) ⇒ 6

g INSTR ('ENGINEERING', 'E', 3) ⇒ 6

Q. WANTED ENAME who have char A in their Names.

g ENAME

F Emp

w INSTR ('ENAME', 'A', 1) > 0;

Q. WANTED ENAME who have char L in at least twice in their name

g ENAME

X F Emp

w INSTR ('ENAME', 'L', 1) ~~>= 2~~ >= 2;

✓ g ENAME

F Emp

w INSTR (ENAME, 'L', 1, 2) > 0;

SET OPERATOR :-

It is used to combine the result of two select statements.

Types of SET OPERATOR

- * UNION
- * UNION ALL
- * INTERSECT
- * MINUS

- * UNION :-
- It is used to combine the result of two statements and it will remove the duplicate value
 - From both the query same no. of column should be present and the datatype should be same.

SYNTAX :-

```

SELECT Col_Name
FROM TABLE_NAME 1
UNION
SELECT Col_Name
FROM TABLE_NAME 2 ;
    
```

EMP

ENAME	DEPTNO
A	10
B	20
C	30
D	40

DEPT

DNAME	DEPTNO
D ₁	10
D ₂	50
D ₃	20
D ₄	60

RESULT
10
20
30
40
50
60

\$ DEPTNO

F Emp

UNION

\$ DEPTNO

F DEPT

* UNION ALL :-

It is used to combine the result of two statements and it will not remove the duplicate values.

SYNTAX :-

SELECT COL. NAME
FROM TABLE-NAME,
UNION ALL

SELECT COL-NAME
From TABLE -NAME 2

EMP	
ENAME	DEPTNO
A	10
B	20
C	30
D	40

DEPTNO	
DDNAME	DEPTNO
D1	10
D2	50
D3	20
D4	60

Result ₁	Result ₂
10	10
20	50
30	20
40	60

\$ DEPTNO
F Emp
UNION ALL
\$ DEPTNO
F DEPT;

RESULT
10
20
30
40
50
60
10
20

* INTERSECT :-

It is used to combine the result of two statements and it will give the common records b/w two statements.

SYNTAX:-

S col. NAME

F TABLE . NAME1 ;

INTERSECT

S col. NAME

F TABLE . NAME2

EMP

Ename	Deptno
A	10
B	20
C	30
D	40

DEPT	
DNAME	DEPTNO
D ₁	10
D ₂	20
D ₃	30
D ₄	40

10	10
20	20
30	20
40	60

RESULT

10

20

S DEPTNO

F Emp

INTERSECT

S DEPTNO

F DEPT;

MINUS

It is used to display the records present in the ~~two~~ result1 and the same records which are not present in the result2.

SYNTAX :

```

S col. NAME
F TABLE. NAME ;
MINUS
S col. NAME
F TABLE. NAME 2 ;

```

EMP

ENAME	DEPTNO
A	10
B	20
C	30
D	40

DEPT

DNNAME	DEPTNO
D1	10
D2	50
D3	20
D4	60

Result₁

10
20
30
40

Result₂

10
50
20
60

RESULT

30

40

S DEPTNO

F EMP

MINUS

S DEPTNO

F DEPT;

PSEUDO COLUMN

- PSEUDO COLUMNS are the false columns which are present in each and every table and must be called explicitly.
- We have two pseudo columns
 - ROWID
 - ROWNUM

- 1 ROWID :-
- ROWID is a 18 digit address of the record in the memory.
 - RowID's are generate at the time insertion of records.
 - Row ID's are unique.
 - RowID's cannot be deleted.
 - Row ID is static (it cannot be changed or constant).
 - Row ID is Quickest way to access or delete a record.

Example:-

→ \$ RowID, Rownum, DEPT.*
F DEPT;

ROWID	Rownum	DEPTNO	DNAME	LOCATION
1AAMFNAAIAQAAI	1			
2AAMFPAAEAAE	2			
3AAMFPAAEAAE	3			
4AAMFPAAEAAE	4			

⇒ \$ *
F Emp

W RowID = '1AAMFNAAIAQAAI';

EMPNO	ENAME	JOB
7834	MILLER	CLERK

⇒ \$ ROWNUM
F Emp

W ENAME = 'KING';

RowID

1AAMFPAAEAAIAQAAI

2. ROWNUM :-

- o Rownum is serial no. assigned to the result table.
- o Rownum is assigned at the time of execution.
- o Rownum is always starts with 1.
- o Rownum is dynamic (not constant).

EMP	ENAME	SAL
A	SAU	100
B	MO	200
C	MO	300
D	SAU	400
E	SAU	500

RESULT	RowNum	ENAME	SAL
+1	1	A	100
+1	2	B	200
+1	3	C	300
+1	4	D	400
+1	5	E	500

RESULT	RowNum	ENAME	SAL
+1	1	A	100
+1	2	B	200
+1	3	C	300
+1	4	D	400
+1	5	E	500

RESULT	RowNum	ENAME	SAL
+1	1	A	100
+1	2	B	200
+1	3	C	300
+1	4	D	400

① SELECT Rownum, ENAME, SAL
FROM EMP;
F Emp

② R Rownum, ENAME, SAL
RESLT

W Rownum < 5;

③

R Rownum, ENAME, SAL
EMP

W Rownum = 2; → To do this we have to make Rownum as static
Rownum=2

i 1 A 100 i=2 X

i 2 B 200 i=2 X

i 3 C 300 i=2 X

i 4 D 400 i=2 X

i 5 E 500 i=2 X

⇒ No Rows Selected

To make Rownum as static (constant)

- Step 1 :- Assign Rownum to a table and rename the column name as SLNO.

① $\$ ROWNUM AS SLNO ,ENAME ,SAL$
 $F EMP;$

↓ RESULT

SLNO	ENAME	SAL
1	A	100
2	B	200
3	C	300
4	D	400
5	E	500

- Step 2 :- Use the step 1 query in the from clause of outer query.

② $\$ *$

$F (\$ ROWNUM AS SLNO ,ENAME ,SAL$
 $F EMP);$

RESULT ↓

SLNO	ENAME	SAL
1	A	100
2	B	200
3	C	300
4	D	400
5	E	500

- Step 3 :- Use the SLNO condition in the where clause to get the expected output.

③

$\$ *$

$F (\$ ROWNUM AS SLNO ,ENAME ,SAL$
 $F EMP)$

W SLNO = 2;

↓ RESULT

SLNO	ENAME	SAL
2	B	200

Q. WAP TO even no. of EMPNUM

S *

F (SELECT RowNum AS SLNO , EMP.*
F EMP)

W MOD(SLNO, 2) = 0;

Q. WAP TO odd no. of EMPNUM

S *

F (\$ RowNum AS SLNO , EMP.*
F EMP)

W MOD(SLNO, 2) = 1;

Assignment on PSEUDO column.

Q.1. WAP TO PSEUDO column (RowID, RowNum) from Emp Table.

S RowNum, RowID

F EMP;

Q.2. WAP TO 2nd record from the table:

S *

F (\$ RowNum, RowID, Emp.*
F EMP)

W

S *

F (\$ RowNum AS SLNO , EMP.*
F EMP)

W SLNO = 2;

Q.3. WAP TO 1st 4 records from the emp table

S *

F EMP

W RowNum <= 4;

Q.4. WAP TO last record from the emp table

S *

F (\$ RowNum AS SLNO , EMP.*
F EMP)

W SLNO = (\$ COUNT(X)
F EMP);

Q.5 WAP/TD last but one record from the table.

S ENAME
F (SELECT ROWNUM AS SLNO , Emp.X
FROM EMP)
W SLNO = (% COUNT(X)
F EMP);

Q.6. Display even no. of records from emp table.

S *
F (% ROWNUM AS SLNO , Emp.*
F EMP)
W MOD(SLNO, 2) = 0;

Q.7 Display odd no. of records from emp table.

S *
F (% ROWNUM AS SLNO , Emp.*
F EMP)
W MOD(SLNO, 2) = 1;

Q.8. WAP/TD last 50% of Records from emp table.

S *
F rom (% ROWNUM AS SLNO , Emp.X
F EMP)
W HERE SLNO >= (% COUNT(X)/2
F EMP);

SLNO >= (% COUNT(X)/2
F EMP);

Q.9. WAP/TD is 50% (Half) of the records from emp table.

S *
F (% ROWNUM AS SLNO , Emp.X
F EMP)
W SLNO <= (% COUNT(X)/2
F EMP);

Q.10. What is 1st half (50%) of the records from Emp table and having char 'A' in their name.

S *

F (\$ RowNum AS SLNO , Emp.*
F Emp)

W SLNO <= (\$ COUNT(X)/2
F Emp)

AND ENAME LIKE '%.A%.%';

Q.11. What is 2nd half 50% of the record from emp table and having char 'S' as their second character in their job.

S *

F (\$ RowNum AS SLNO , Emp.*
F Emp)

W SLNO >= (\$ COUNT(X)/2
F Emp)

AND SUBSTR (JOB, 2,1) = 'S';

Q.12. What is 3rd, 4th, 5th record from the emp-table

S *

F (\$ RowNum AS SLNO , Emp.*
F Emp)

W SLNO IN (3,4,5);

Q. What are details of all the employees

S *

F EMP;

Q. What is salary of all the employees

S SAL

F Emp;

Q. What is first max sal.

S Max(SAL)

F EMP;

Q. What are details of the emp who are getting first min(sal)

S MIN(SAL) S *

F EMP;

W SAL = (\$ MIN(SAL)
F Emp);

Q. WAPTO 2nd max sal

WAPTO

~~S SAL~~
~~H EMP~~
S SAL < $\{ \}$ max(SAL)
F EMP

W SAL < ($\{ \}$ max(SAL)
F EMP);

Q. WAPTO details of the emp who are getting 2nd min(sal)

~~S EMP~~
W SAL = $\{ \}$ min(SAL)

F EMP

W SAL > ($\{ \}$ min(SAL)
F EMP);

Q. WAPTO 10th MAX(SAL)

S SAL

F (

To find n^{th} min and n^{th} max salaries using PSEUDO columns.

To find n^{th} max salary

STEP 1 :- Arrange all the distinct salaries in descending order

SAL
8000
10000
15000
20000
30000
20000
40000
50000
40000

① $\% \text{ DISTINCT SAL}$
 F EMP
 $\text{ORDER BY SAL DESC ;}$

↓

SAL
50000
40000
30000
20000
15000
10000

STEP 2 :- For the arrange table assign the serial no. by using ROWNUM

② $\text{SELECT ROWNUM AS SLNO, SAL}$
 $\text{FROM (SELECT DISTINCT SAL}$
 FROM EMP
 $\text{ORDER BY SAL DESC);}$

SLNO	SAL
1	50000
2	40000
3	30000
4	20000
5	15000
6	10000

STEP 3:- Use the serial no. as a condition for the table which we have created in the step 2.

③ $\% \text{ SAL}$
 $\text{F (%ROWNUM AS SLNO, SAL}$
 $\text{FROM (% DISTINCT SAL}$
 F EMP
 $\text{ORDER BY SAL DESC))}$
W. $SLNO = S;$

\Rightarrow RESULT

SAL
15000

Q.1 WAP TO TOP 3 MAX SALARIES.

S SAL
F (S ROWNUM AS SLNO, SAL
F EMP (S DISTINCT SAL
~~ORDF~~ F EMP
ORDER BY SAL DESC))
W SLNO < 4;

Q.2. WAP TO TOP 5 MIN salaries

S SAL
F (S ROWNUM AS SLNO, SAL
F (S DISTINCT SAL
F EMP
ORDER BY SAL))
W SLNO < 6;

Q.3. WAP TO 1,3,5 MIN SAL

S SAL
F (S ROWNUM AS SLNO, SAL
F (S DISTINCT SAL
F EMP
ORDER BY SAL))
W SLNO IN (1,3,5);

Q.4. WAP TO Bottom 3 MAX SAL / Top 3 MIN SAL

S SAL
F (S ROWNUM AS SLNO, SAL
F EMP (S DISTINCT SAL F EMP
O.B. SAL ~~DESC~~))
W SLNO < 4;

TOP
Bottom 3 MAXSAL
= Top 3 MIN

Q.5. WANTED details of the emp covering steps mentioned

S X

F (S SAL

F (S ROWNUM AS SLNO, SAL

F (S DISTINCT SAL

F EMP

OR BY. DESC))

W SLNO < 6);

S X

F EMP

W SAL IN (S SAL

F (S ROWNUM AS SLNO, SAL

F (S DISTINCT SAL

F EMP

O.B. ^{SLNO} DESC))

W SLNO < 6);

Q. WANTED dname and loc of the emp who are getting top 10 max (SAL)

~~S DNAME, LOC~~

~~F DEPT~~

~~W EMPNO IN (S EMPNO~~

~~S DNAME, LOC~~

~~F DEPT~~

~~W DEPTNO IN (S DEPTNO~~

F (S ROWNUM AS SLNO, SAL

F (S DISTINCT SAL

F EMP

O.B. DESAL DESC))

W SLNO < 11);

SQL :- Statements

Statements
can be classified
into 5 languages

- * DDL (Data definition language)
- * DML (Data manipulation language)
- * TCL (Transaction control language)
- * DCL (Data control language)
- * DQL (Data query language) ✓

1. Data Definition Language (DDL)

- This language is used to construct, modify and remove an object from the database.
- It has 5 statements
 - 1) CREATE
 - 2) RENAME
 - 3) ALTER
 - 4) TRUNCATE
 - 5) DROP

- * CREATE :-
- This statement is used to construct or build an object in the database.
 - Object refers to a table or view (Virtual table)

SYNTAX :-

```
CREATE TABLE TABLE_NAME
(
    COL_NAME1 DATATYPE CONSTRAINTS,
    COL_NAME2 DATATYPE CONSTRAINTS,
    COL_NAME3 DATATYPE CONSTRAINTS,
    ...
    COL_NAME N DATATYPE CONSTRAINTS
);
```

Example 1 :-

Table Name \Rightarrow INDIA

Col. Names \Rightarrow

JNO	PNAME	AGE	ROLE
NUMBER(4)	VARCHAR(30)	NUMBER(10)(2)	VARCHAR(30)
UN	NN	NN	NN
NN	NN	CHECK(AGE > 18)	NN
P.K.			

constraints

all are optional

UNIQUE

NOTNULL

CHECK

PRIMARY KEY

FOREIGN KEY

(P.S.)

CREATE TABLE INDIA

```

JNO NUMBER(4) PRIMARY KEY,
PNAME VARCHAR(30) NOTNULL,
AGE NUMBER(2) NOTNULL CHECK(AGE > 18),
ROLE VARCHAR(30) NOTNULL
);

```

Example 2 :-

Table Name \Rightarrow PAK

Col. Names \Rightarrow

NO	PNAME	PHNO	JNO
NUMBER(4)	VARCHAR(30)	NUMBER(10)	NUMBER(4)
UN	NN	NN	UN
NN	NN	NN	CHECK(LENGTH(PHNO) = 10)
PK			FK

constraints

optional

UNIQUE

NOTNULL

CHECK

PRIMARY KEY

FOREIGN KEY

SYNTAX for creating foreign key.

```
C  
CONSTRAINT COL_NAME_FK FOREIGN KEY (COL_NAME)  
REFERENCES PARENT_TABLE_NAME (COL_NAME)  
);
```

CREATE TABLE PAK

```
C  
NO NUMBER(4) PRIMARY KEY,  
PNAME VARCHAR(30) NOT NULL,  
PHNO NUMBER(10) UNIQUE NOTNULL CHECK(LENGTH(PHNO)=  
10),  
JNO NUMBER(4),  
CONSTRAINT JNO_FK FOREIGN KEY (JNO)  
REFERENCES INDIA (JNO)  
);
```

To Copy a TABLE :-

Syntax :-

CREATE TABLE TABLE_NAME
AS
SELECT STATEMENT;

Example 1

```
CREATE TABLE Emp1  
AS  
SELECT *  
FROM Emp  
WHERE Job = 'SALESMAN';
```

Example 2

```
CREATE TABLE Emp2  
AS  
SELECT *  
FROM EMP;
```

UPDATE TEMP2
SET Comm = 100;

TO CREATE A VIEW

VIEW :- It is a virtual table which is created on the result set of query.

- It doesn't occupy any memory.
- It can be created on any table
- Any changes on the original table will have an impact on the view created on the table.

SYNTAX :-

```
CREATE VIEW VIEW NAME
AS
SELECT STATEMENT;
```

Example:-1

```
CREATE VIEW SALES-MAN
AS
SELECT *
FROM EMP
WHERE JOB='SALESMAN';
```

⇒ To grant permission for updating view
SQL> CONNECT;

Enter User-name: SYSTEM
Enter Password: tiger
Connected

SQL> GRANT ~~VIEW~~ CREATE VIEW
 2 TO SCOTT;

Grant succeeded.

SQL> CONNECT

Enter User-name: SCOTT;
Enter Password: tiger

SQL> SHOW USER;
USER is "SCOTT" Connected

⇒ Now we can Create View

CREATE VIEW FROM SCOTT;
REVOKE CREATE VIEW FROM SCOTT;

REVOKE

Syntax

Difference b/w Table and View

TABLE

① CREATE TABLE EMP_T
AS
SELECT *
FROM EMP
WHERE SAL > 200;

EMP_T

EMPNO	FNAME	SAL
1	A	100
2	B	200
3	C	300
4	D	400
5	E	500

changes happen in Emp_T
will not affect the
original table

VIEW

① CREATE VIEW Emp_V
AS
SELECT *
FROM EMP
WHERE SAL > 200;

EMP

EMPNO	FNAME	SAL
1	A	100
2	B	200
3	C	300
4	D	400
5	E	500

Original table

EMPNO	FNAME	SAL
3	C	300
4	D	400
5	E	500

Changes happen in
view will affect the
original table

View also create separate
table from original

* 2) RENAME :-

This statement is used to change the name of an existing table.

SYNTAX:-

RENAME TABLE_NAME To NEW_TABLE_NAME

Example

RENAME INDIAN TO BHARAT;

To rename the table we can use rename directly
and to rename column we use rename under
alter

*3) ALTER :-

- This statement is used to modify the structure of the object.

SYNTAX

① To ADD the column

ALTER TABLE TABLE-NAME
ADD COL-NAME DATATYPE CONSTRAINTS;

Example 1

ALTER TABLE BHARAT
ADD PHNO NUMBER(10) NOTNULL UNIQUE CHECK(LENGTH(PHNO)=10);
Table altered

DESC BHARAT;

display the column details

Example 2

ALTER TABLE BHARAT
ADD SCORE NUMBER(3);

② To drop the column

SYNTAX :- ALTER TABLE TABLE-NAME
DROP COLUMN COLUMN-NAME;

Example:-

ALTER TABLE BHARAT
DROP COLUMN PHNO;

③ To RENAME the column

SYNTAX :- ALTER TABLE TABLE-NAME
RENAME COLUMN COL-NAME To NEW COL-NAME;

Example:-

ALTER TABLE BHARAT
RENAME COLUMN SCORES TO RUNS;
Table altered.
DESC BHARAT;

④ To Modify the DATATYPE :-

SYNTAX:- ALTER TABLE TABLE NAME
 MODIFY col.NAME DATATYPE ;

Example:-

ALTER TABLE BHARAT
MODIFY RUNS VARCHAR(3);

- * 4) TRUNCATE :-
- This statement is used to remove all the records permanently from the table.
 - It is used to empty the table.
 - It will not disturb the table structure.

SYNTAX:-

TRUNCATE TABLE TABLE NAME ;

Example:-

TRUNCATE TABLE PAK;

* 5) DROP :-

- This statement is used to remove the table from the database.

SYNTAX:-

DROP TABLE TABLE NAME ;

Example:-

DROP TABLE PAK;

To recover the table

* FLASH BACK

To recover the table
This stat is used to recover the table from the recycle bin,

SYNTAX :-

FLASH BACK TABLE TABLE NAME TO BEFORE DROP

FLASHBACK TABLE TABLE NAME TO BEFORE DROP ;

Example:- FLASHBACK TABLE PAK TO BEFORE DROP;

To Remove the table from Bin

* PURGE :- This stmt is used to remove the table from recyclebin.

SYNTAX:-

PURGE TABLE TABLE-NAME;

To use PURGE we have to drop first

Example:-

Drop TABLE PAK;

Table dropped.

Purge TABLE PAK;

Table purged.

2. Data Manipulation Language (DML)

- This statement is used to add, modify and remove the records from the table.

It has 3 statements

- 1) INSERT
- 2) UPDATE
- 3) DELETE

* 1) INSERT :-

This statement is used to Add the records into the given table.

SYNTAX 1:

INSERT INTO TABLE_NAME VALUES (V₁, V₂, V₃, ..., V_n)

Example:-

INSERT INTO BHARAT VALUES (18, 'VIRAT', 34,
'BATSMAN', 100);

SYNTAX 2:

INSERT INTO TABLE_NAME VALUES (& JNO COLUMN_NAME₁,
& COLNAME₂, & COLNAME₃, ..., & COLNAME_N)

Example:-

INSERT INTO BHARAT VALUES (& JNO, & PNAME, & AGE,
& ROLE, & RUNS);

Enter Value for JNO: 45

Enter Value for PNAME: 'ROHIT'

Enter Value for AGE: 36

Enter Value for ROLE: 'BATSMAN'

Enter Value for RUNS: 10

* 2) UPDATE :-

This statement is used to modify the existing values present in the table.

SYNTAX :-

UPDATE TABLE_NAME
SET COL-NAME₁=VALUE, COL-NAME₂=VALUE, ...
WHERE < FILTER CONDITION > ;

Example 1 :-

UPDATE BHARAT

SET JNO = 10

WHERE PNAME = 'ABD' ;

Example 2 :-

UPDATE BHARAT

SET PNAME = 'SACHIN', AGE = 45, ROLE = 'BATSMAN'

WHERE JNO = 10 ;

* 3) DELETE :-

This statement is used to remove a particular record from a given table.

SYNTAX :-

DELETE FROM TABLE-NAME
[WHERE <FILTER-CONDITION>] ;

Example :-

DELETE FROM BHARAT

WHERE JNO = 99 ;

or

DELETE FROM BHARAT ;

Difference b/w TRUNCATE and DELETE

TRUNCATE

- ① This statement belongs to DDL
- ② This stmt is used to remove all the records permanently from the table
- ③ We cannot recover the records after truncating
- ④ This stmt belongs to auto-commit language

DELETE

- ① This stmt belongs to DML
- ② This stmt is used to remove a particular record from the table
- ③ We can recover the deleted records before committing
- ④ There we have to use commit explicitly.

NOTE :-

- 1) DDL language is an Auto-commit language.
- 2) DML language is not an Auto-commit language.

3 Transaction Control Language (TCL)

- This language is used to control the transactions such as INSERT, UPDATE, and DELETE on the Database.

It has 3 statements

- 1) COMMIT
- 2) SAVE POINT
- 3) ROLL BACK

* 1) COMMIT :-

This statement is used to save the transaction in the database.

SYNTAX :-

COMMIT ;

* 2) SAVE POINT :-

This statement is used to mark the position in the database.

SYNTAX :-

SAVE POINT SAVEPOINT_NAME ;

* 3) ROLL BACK :-

- This statement is used to go back to the latest committed point.
- Rollback only gives us what is saved in the Database.

SYNTAX :-

ROLLBACK TO SAVEPOINT_NAME ;

Example :-

1
2
3
4
5
6

S1

1
2

SAVEPOINT S1;

3

4

SAVEPOINT S2;

5

6

```
graph TD; 1[1] --> 2[2]; 2 --> 3[3]; 3 --> 4[4]; 4 --> 5[5]; 5 --> 6[6]; S1[S1] --> 1; S1 --> 2; S2[S2] --> 4;
```

4 Data Control Language DCL

This statement is used to control the flow of the data b/w the user.

It has 2 statements

1) GRANT

2) REVOKE

* 1) GRANT :-

This stmt is used to give permission to the user.

Syntax :-

GRANT SQL STATEMENTS ON TABLE-NAME
TO USR-NAME

F

SQl STATEMENTS :-

SELECT
INSERT
UPDATE
DELETE

Example:-

: GRANT SELECT ON DEPT
TO HR;

* 2) REVOKE :-

This statement is used to take back the permission from the user.

Syntax :-

REVOKE SQL STATEMENT ON TABLE-NAME
FROM USR-NAME;

How to Connect or LOGIN as USER.

Commands:

CONNECT;

Enter the UserName; HR

Enter the password; Tiger

→ Connected

To check the current user.

SHOW USER

USER is "SCOTT"

SCOTT

DEPT

GRANT SELECT ON DEPT TO HR;
REVOKE SELECT ON DEPT FROM HR;

HR

SELECT *
FROM SCOTT.DEPT;

ATTRIBUTES:

- * KEY ATTRIBUTE / CANDIDATE KEY
- * NON-KEY ATTRIBUTE
- * PRIME KEY ATTRIBUTE
- * NON-PRIME KEY ATTRIBUTE
- * COMPOSITE KEY ATTRIBUTE
- * SUPER KEY ATTRIBUTE
- * FOREIGN KEY ATTRIBUTE

* KEY ATTRIBUTE

An attribute which is used to identify the records uniquely from a table is known as key attribute.

* NON-KEY ATTRIBUTE

All the attribute other than key attribute is known as Non-key Attribute.

* PRIME KEY ATTRIBUTE

Among the key attribute an attribute is chosen to be a main attribute to identify the records uniquely from the table is known as prime key attribute.

* NON-PRIME KEY ATTRIBUTE

All the key attribute other than prime key attributes is known as Non-prime key attribute.

* COMPOSITE KEY ATTRIBUTE

It is a combination of two or more non-key attribute which is used to identify the records uniquely from the table is known as composite key attribute.

* SUPER KEY ATTRIBUTE

The set of an key attribute is known as super key attribute.

* FOREIGN KEY ATTRIBUTE

It is an attribute which behaves as an attribute of another entity to represent the relationship is known as foreign key attribute.

FUNCTIONAL DEPENDENCY

There exists a dependency such that an attribute in a relation determines another attribute.

TYPES of functional dependency

- * Total functional dependency.
- * Partial functional dependency.
- * Transitive functional dependency.

Total FUNCTIONAL DEPENDENCY :-

If an attribute in a relation determines all the other attribute it is known as Total functional dependency.
OR

If all the attribute are dependent ^{on} a single attribute it is known as total functional dependency.

Partial FUNCTIONAL DEPENDENCY :-

There exists a dependency such that a part of composite key attribute determines another attribute uniquely is known as partial functional dependency.

TRANSITIVE FUNCTIONAL DEPENDENCY:-

There exists a dependency such that an attribute is determined by a Non-key attribute which is inter-depend by a key attribute is known as TRANSITIVE FUNCTIONAL DEPENDENCY.

REDUNDANCY

The Repetition of unwanted data is known as Redundancy.

ANOMALY:-

The side effects caused during DML operation is known as ANOMALY.

NORMALIZATION

It is a process of reducing a large table into smaller tables in order to remove Redundancy and anomalies by identifying their functional dependency is known as NORMALIZATION.

OR

The process of decomposing a large table into smaller table is known as NORMALIZATION.

OR

Reducing a table to its normal form is known as NORMALIZATION.

TYPES of NORMAL FORM

- 1) FIRST NORMAL FORM
- 2) SECOND NORMAL FORM
- 3) THIRD NORMAL FORM
- 4) BOYCE CODD NORMAL FORM

NORMAL FORM

↓
A table without redundancies and anomalies are said to be in normal form.

NOTE :- If any table is reduced to 3rd Normal form then the table is said to be NORMALIZED.

1) FIRST NORMAL FORM

- No-duplicate records
- Multi-value data should not be present.

2) SECOND NORMAL FORM

- Table should be in 1st Normal Form
- Table should not have partial functional dependency.

3) THIRD NORMAL FORM

- Table should be in 2nd Normal Form
- Table should not have transitive function dependency.