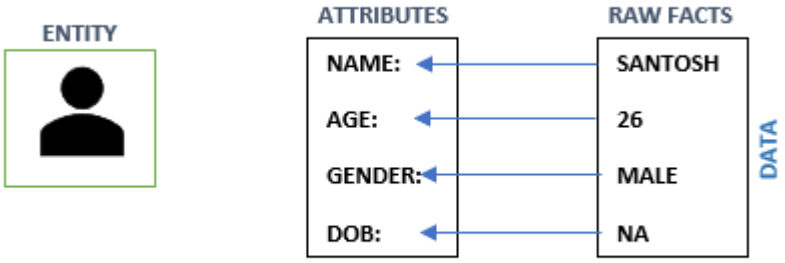
SQL:ORACLE DATABASE

TRAINER: MOHAMMAD RIZA WASHIQ

DATA

Data is a *raw fact* which describes the *attributes* of an *entity*.



DATABASE

Database is a *place* where we can *store the data* in a systematic & organized manner.

Basic Operations performed on Database are;

1. Create / Insert
2. Read / Retrieve
3. Update / Modify
4. Delete / Modify

Generally known as "**C R U D**" operations.

DBMS (Database Management System)

- This is a *software* we *use* to maintain & manage the database.
- It provides two factors:
  - Security
  - Authorization
- We use Query Language to *communicate* with DBMS.

RDBMS (Relational Database Management System)

- It is a type of DBMS software where we *store* the *data* in the *form* of tables.
- We *use* SQL (Structured Query Language) to *communicate* with RDBMS.
- RDBMS follows *the theory* of "**Relational Model**".

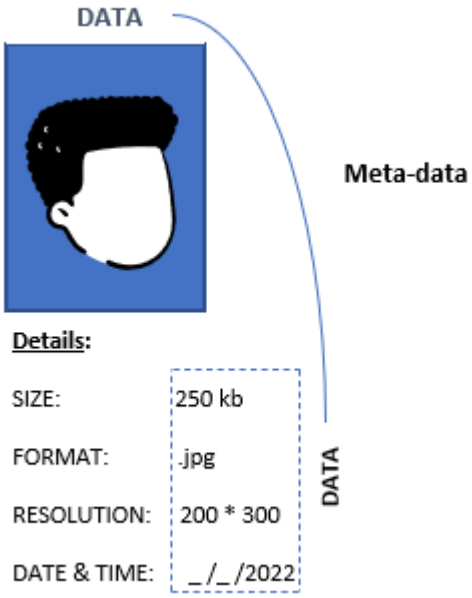
Relational Model

It is a theory *designed* by a Data Scientist named E.F. Codd which says that data should be stored in the form of tables.

Rules of E.F. Codd

1. The data *stored* in the cell *must* be a single value data.

2. In RDBMS, we *store* everything in the form of tables *including* meta data (The details about the data are meta data).
3. According to E.F. Codd, *we can store data in multiple tables*. If needed, we can establish *connection between two* tables using **key attributes.**
4. We can validate the data *entered* into the table in 2 steps,
    a. By assigning **datatypes.**
    b. By assigning **constraints.**



DATATYPES

Datatypes are *used* to determine what type or kind of data will be *stored* in *particular* memory location.

Datatypes in SQL

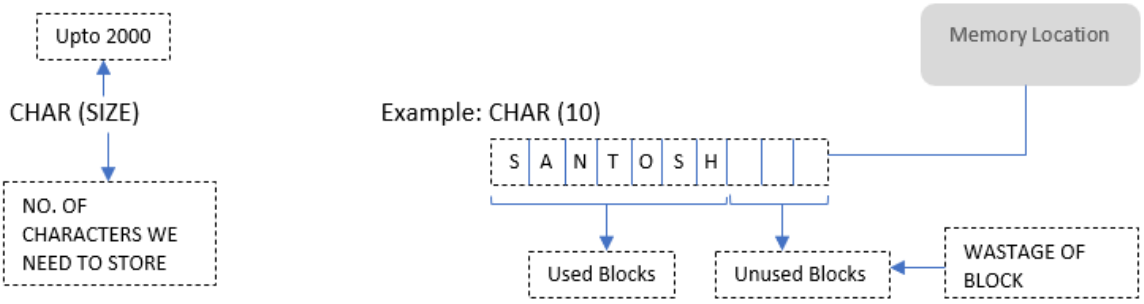
1. CHAR
2. VARCHAR
3. NUMBER
4. DATE
5. LARGE OBJECT
    a. Character Large Object (CLOB)
    b. Binary Large Object (BLOB)

Note: **SQL** is *not* a case sensitive.

CHAR

It is a datatype in which we can store characters from A-Z, a-z, 0-9 & special characters (#, $, ...)

Syntax:

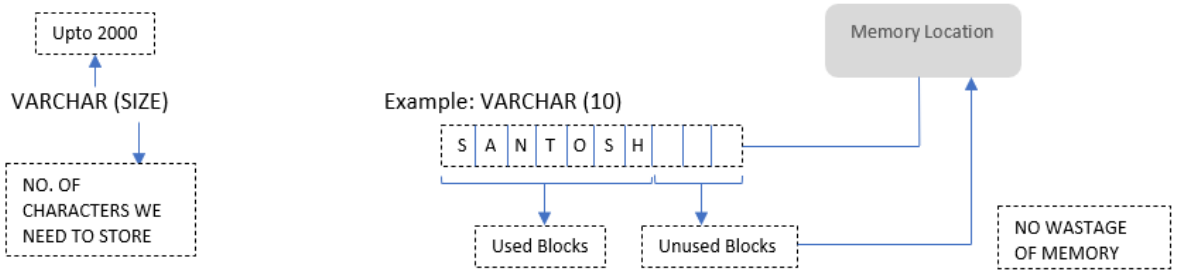


- The maximum character we can store in CHAR datatype is 2000.
- There is *wastage of memory* in CHAR datatype.
- CHAR is also known as '**Fixed Length Memory Allocation**'.

VARCHAR

In VARCHAR, we can pass arguments like A-Z, a-z, 0-9 & special characters (#, $, ...)

Syntax:



- The maximum character we can store in VARCHAR datatype is 2000.
- In VARCHAR, there is no wastage of memory.
- VARCHAR is also known as '**Variable Length Memory Allocation**'.

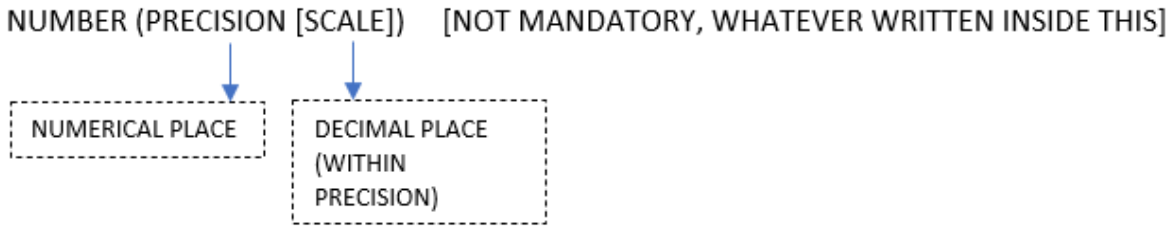
VARCHAR2

It is an updated version of VARCHAR & we can store upto 4000 characters.

NUMBER

- It is a datatype which is used to store the numbers.
- We can pass 2 arguments (PRECISION & SCALE)

Syntax:



NUMBER (3)	+ -	9 9 9
NUMBER (5,2)	+ -	9 9 9 . 9 9
NUMBER (7,2)	+ -	9 9 9 9 9 . 9 9
NUMBER (2,7)	+ -	0 . 0 0 0 0 0 9 9
NUMBER (2,2)	+ -	0 . 9 9

DATE

Syntax: DATE

The 2 oracle specified date formats are:

a. 'DD-MON-YY' (**MON**, denotes the *first 3 characters of the month*)
b. 'DD-MON-YYYY'

Dates should *always* be enclosed within single quotes.

Example: '06-SEP-87' or, '06-SEP-1987'

LARGE OBJECT

a. **CLOB (Character Large Object)**
This is used to store the characters upto 4GB of size.
Syntax: CLOB

b. **BLOB (Binary Large Object)**
This is used to store binary number of images, videos, files, etc. upto 4GB size.
Syntax: BLOB

CONSTRAINTS

Constraints are the conditions that are *assigned* on a particular column to validate the *data*.

Types of Constraints

1. UNIQUE

2. NOT NULL
3. CHECK
4. PRIMARY KEY
5. FOREIGN KEY

UNIQUE

It is a constraint that is assigned to a particular column which cannot *accept* the repeated values.

NOT NULL

It is a constraint we assign to a particular column which cannot be a NULL or, records which are mandatory.

CHECK

It is a constraint we assign for a particular column for *extra validation*.

PRIMARY KEY

- It is a constraint we assign to a particular column to *identify the records* uniquely from the table.
- Primary Key *must* be unique & it *cannot* be a NULL.
- It is better to have *only* one primary key in table.

FOREIGN KEY

- It is a constraint we assign to a column to *establish connection* between two tables.
- Foreign Key need *not* be unique & it *can* be NULL.
- Only primary key *can* become a foreign key in *another table*.
- We *can* have any number of *foreign keys* in a table.

ATTRIBUTES

1. KEY ATTRIBUTES / CANDIDATE KEY
2. NON-KEY ATTRIBUTE
3. PRIME-KEY ATTRIBUTE
4. NON-PRIME KEY ATTRIBUTE
5. COMPOSITE KEY ATTRIBUTE
6. SUPERKEY ATTRIBUTE
7. FOREIGN KEY ATTRIBUTE

KEY ATTRIBUTES/ CANDIDATE KEY

An attribute which is *used* to identify a *record* uniquely from the table is called Key Attribute.

NON-KEY ATTRIBUTE

All the attributes *except* Key Attributes are referred as Non-Key Attributes.

PRIME KEY ATTRIBUTES

Among the Key Attributes, an attribute is chosen to be the main attribute to identify the record uniquely from the table.

NON-PRIME KEY ATTRIBUTE

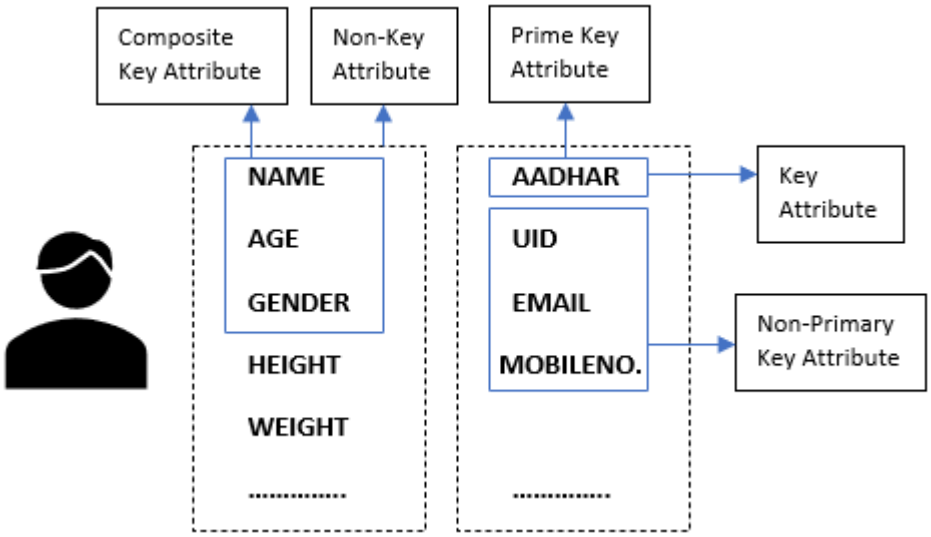
All the key attributes except Prime-Key Attribute is referred as Non-Prime Key Attribute.

COMPOSITE KEY ATTRIBUTE

It is a combination of two or more Non-Key Attributes, which is used to identify the record uniquely from the table.

SUPER KEY ATTRIBUTE

It is the set of all the key attributes.



STATEMENTS IN SQL

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

DATA QUERY LANGUAGE (DQL)

- This statement is *used* to retrieve the data from database.
- There are 4 statements:
    1. SELECT
    2. PROJECTION
    3. SELECTION
    4. JOINS

SELECT

This statement is *used* to retrieve the data from *database* & display it.

PROJECTION

- This statement is *used* to retrieve the data from the *database* by selecting *only* column.
- *All* the *values* in the column will be *selected* by default.

SELECTION

This statement is *used* to retrieve the data from *database* by selecting *both* columns as well as records.

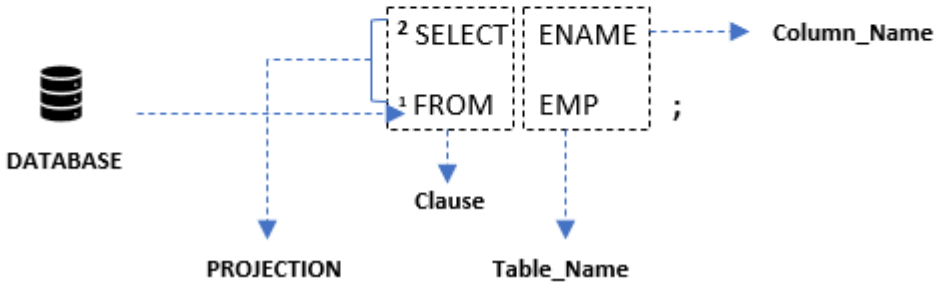
JOINS

This statement is *used* to retrieve the data from multiple tables simultaneously.

EMP

EMPID	ENAME	JOB	SALARY
101	NIKHIL	DEVELOPER	$200
102	SANTOSH	TEST ENGINEER	$100
103	PAVITRA	UI/UX	$350

Write a Query to Display names of all the employees.



ENAME

NIKHIL

SANTOSH

PAVITRA

NOTE:

- '**FROM**' clause *starts* the **execution**.
- '**FROM**' clause, we can *pass* table name as an *argument*.
- The *job* of '**FROM**' clause is to go to the database & search for the *table* & put the table under execution.
- '**SELECT**' clause **executes** *after* the **execution** of '**FROM**' clause.
- For '**SELECT**' clause, we can pass asterisk (*), column name & expressions as an argument.
- The job of '**SELECT**' clause is to go to *the table* which is *under execution* & **select the data** & **display**.
- '**SELECT**' clause is responsible for the **result table**.

Q1. WAQTD job & salary of all the employees?

SELECT JOB, SAL

FROM EMP;

NOTE:

If we want to display more than one column, we have to separate the columns by using comma (,).

Q2. WAQTD name & job of all the employees?

SELECT ENAME, JOB

FROM EMP;

NOTE:

We have to write the same column name or the table name i.e., present in the database.

Q3. WAQTD all the details from EMP table?

SELECT EMPNO, ENAME, JOB, SAL, MGR, COMM, HIREDATE, DEPTNO

FROM EMP;

or,

SELECT *

FROM EMP;

```
SQL> SELECT *
  2  FROM EMP;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

14 rows selected.
```

ASTERISK (*): It is used "*To select all the columns*".

SEMI-COLON (;): It is used to determine "*The end of the statement*".

Q3. WAQTD the tables present in the database?

SELECT *

FROM TAB;

```
SQL> SELECT *
  2  FROM TAB;

TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
DEPT                           TABLE
EMP                            TABLE
BONUS                          TABLE
SALGRADE                       TABLE
```

Q4. WAQTD name & salary given to all the employees?

SELECT ENAME, SAL

FROM EMP;

Q5. WAQTD employee ID & department no. of all the employees in EMP table?

SELECT EMP NO., DEPT NO.

FROM EMP;

Q6. WAQTD employee name & hire date of all the employees?

SELECT ENAME, HIREDATE

FROM EMP;

Q7. WAQTD name & designation of all the employees?

SELECT ENAME, JOB

FROM EMP;

Q8. WAQTD name & annual salary given to all the employees?

SELECT ENAME, SAL*12

FROM EMP;

Q9. WAQTD name & half-term salary of all the employees?

SELECT ENAME, SAL*6

FROM EMP;

Q10. WAQTD name, salary & salary with 25% hike?

SELECT ENAME, SAL, SAL + $\frac{25}{100}$*SAL

FROM EMP;

Q11. WAQTD name, salary & also salary with 12% deduction?

SELECT ENAME, SAL, SAL – $\frac{12}{100}$*SAL

FROM EMP;

Syntax:

SELECT */ [DISTINCT] COLUMN_NAME/ EXPRESSIONS [ALIAS]

FROM TABLE_NAME;

Order of Execution:

1. FROM
2. SELECT

EXPRESSIONS

A statement which gives us result is known as *expressions*.

Expressions consists of 2 types:

1. **Operand**

   Operand consists of 2 types:

   A. COLUMN_NAME
   B. Literals (Direct Values)

   Literals are of 3 types:

   a. Number literals
   b. Character literals
   c. Date literals

2. **Operators**

NOTE:

'Character Literal**'** & **'**Date literal**'** should be enclosed within *single quotes*.

ALIAS

- It is *an* alternative name given to a column or, *an* expression in the result table.
- Alias name can be used with or, without using '**AS**' *keyword*.
- Alias name should be a ***single word*** or, a ***string enclosed within double quotes***.

- Alias is *not mandatory* but *recommended* to provide.

For Example:

SAL*12 AS ANNUALSAL or, SAL*12 ANNUALSAL

SAL*12 AS ANNUAL_SAL or, SAL*12 ANNUAL_SAL

SAL*12 AS "ANNUAL SAL" or, SAL*12 "ANNUAL SAL"

NOTE:

To display all the details of the table along with other column of different attribute we use the following,

SELECT TABLE_NAME.*, EXPRESSION [ALIAS]

For Example:

Q12. WAQTD all the details of an employee along with the annual salary?

SELECT EMP.*, SAL*12 ANNUAL_SAL

FROM EMP;

```
SQL> SELECT EMP.*, SAL*12 ANNUALSAL
  2  FROM EMP;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO  ANNUALSAL
---------- ---------- --------- ---------- --------- ---------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20       9600
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30      19200
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30      15000
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20      35700
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30      15000
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30      34200
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10      29400
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20      36000
      7839 KING       PRESIDENT            17-NOV-81       5000                    10      60000
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30      18000
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20      13200
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30      11400
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20      36000
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10      15600

14 rows selected.
```

DISTINCT

- To *remove* **repeated** values or, **duplicated** values in result table, we use 'DISTINCT' clause.
- For **DISTINCT** clause, we can pass **COLUMN_NAME** or, an **expression** as an argument.
- **DISTINCT** clause *should be used* as the **first argument** in the SELECT clause.
- We can *pass* multiple columns for DISTINCT clause.
- It *removes* the **combination** of *duplicates* from all the columns.

Q13. WAQTD only different salaries given to employees?

SELECT DISTINCT SAL

FROM EMP;

```
SQL> SELECT DISTINCT SAL
  2  FROM EMP;

       SAL
----------
      2450
      5000
      1300
      1250
      2850
      2975
      1100
      3000
       800
      1600
      1500
       950

12 rows selected.
```

Q14. WAQTD the different designations that are present in the employee table?

SELECT DISTINCT JOB

FROM EMP;

Q15. WAQTD different Department number as well as salaries that are present in the table?

SELECT DISTINCT DEPTNO., SAL

FROM EMP;

```
SQL> SELECT DISTINCT DEPTNO, SAL
  2  FROM EMP;

    DEPTNO        SAL
---------- ----------
        10       5000
        20        800
        20       2975
        10       2450
        30       1250
        20       3000
        30        950
        20       1100
        10       1300
        30       1600
        30       2850
        30       1500

12 rows selected.
```

SELECTION

- '**WHERE**' clause is *used* to **filter** the records.
- '**WHERE**' clause *executes* **Row-by-Row**.

- We can *pass* filter condition to the 'WHERE' clause.
- We can *pass* multiple conditions to 'WHERE' clause using **logical operators.**

Syntax:

SELECT */ [DISTINCT] COLUMN_NAME/ EXPRESSIONS [ALIAS]

FROM TABLE_NAME

WHERE <FILTER_CONDITIONS>;

Order of Execution:

1. FROM
2. WHERE
3. SELECT

Q16. WAQTD details of an employee working as a 'SALESMAN'?

SELECT *

FROM EMP

WHERE JOB='SALESMAN';

```
SQL> SELECT *
  2  FROM EMP
  3  WHERE JOB='SALESMAN';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
```

Q17. WAQTD names of an employee working in department number 20?

SELECT ENAME

FROM EMP

WHERE DEPTNO =20;

Q18. WAQTD details of an employee who are earning more than Rs.3000?

SELECT *

FROM EMP

WHERE SAL>3000;

Q19. WAQTD all the details of an employee along with the annual salary if the annual salary is more than Rs.14000?

SELECT EMP.*, SAL*12 ANNUAL_SAL

FROM EMP

WHERE SAL*12 > 14000;

```
SQL> SELECT EMP.*, SAL*12 ANNUAL_SAL
  2  FROM EMP
  3  WHERE SAL*12>14000;

     EMPNO ENAME      JOB              MGR HIREDATE          SAL       COMM     DEPTNO ANNUAL_SAL
---------- ---------- --------- ---------- --------- ---------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN         7698 20-FEB-81       1600        300         30      19200
      7521 WARD       SALESMAN         7698 22-FEB-81       1250        500         30      15000
      7566 JONES      MANAGER          7839 02-APR-81       2975                    20      35700
      7654 MARTIN     SALESMAN         7698 28-SEP-81       1250       1400         30      15000
      7698 BLAKE      MANAGER          7839 01-MAY-81       2850                    30      34200
      7782 CLARK      MANAGER          7839 09-JUN-81       2450                    10      29400
      7788 SCOTT      ANALYST          7566 19-APR-87       3000                    20      36000
      7839 KING       PRESIDENT             17-NOV-81       5000                    10      60000
      7844 TURNER     SALESMAN         7698 08-SEP-81       1500          0         30      18000
      7902 FORD       ANALYST          7566 03-DEC-81       3000                    20      36000
      7934 MILLER     CLERK            7782 23-JAN-82       1300                    10      15600

11 rows selected.
```

NOTE:

We *can not* write the **ALIAS NAME** in **WHERE** clause.

Q20. WAQTD name, designation & salary of 'SCOTT'?

SELECT ENAME, JOB, SAL

FROM EMP

WHERE ENAME='SCOTT';

Q21. WAQTD names of an employee working as a 'MANAGER'?

SELECT ENAME

FROM EMP

WHERE JOB='MANAGER';

Q22. WAQTD names of an employee hired after 1982?

SELECT ENAME

FROM EMP

WHERE HIREDATE>='01-JAN-1983';

Q23. WAQTD details of an employee who are working as a 'CLERK' & earning less than 1000?

SELECT *

FROM EMP

WHERE JOB='CLERK' AND

SAL<1000;

```
SQL> SELECT *
  2  FROM EMP
  3  WHERE JOB='CLERK' AND
  4                SAL<1000;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK            7902 17-DEC-80        800                    20
      7900 JAMES      CLERK            7698 03-DEC-81        950                    30
```

OPERATORS

- Arithmetic Operator (+, -, *, /)
- Comparison Operator (=, !=)
- Relational Operator (<, >, <=, >=)
- Logical Operator (AND, OR, NOT)
- Concatenation Operator (||)
- Special Operator (IN, NOT IN, BETWEEN, NOT BETWEEN, LIKE, NOT LIKE, IS, IS NOT)
- Sub-query (ALL, ANY, EXISTS, NOT EXISTS)

LOGICAL OPERATOR

'AND' OPERATOR (Binary Multiplication)

- 'AND' operator *returns* TRUE If *both* the *conditions* are TRUE.
- 'AND' operator should always be used between conditions.

'OR' OPERATOR (Binary Addition)

- 'OR' operator *returns* TRUE If *any one* of the *conditions* is satisfied.
- 'OR' operator should always be used between conditions.

Q24. WAQTD details of an employee working as a 'MANAGER' & earning less than Rs.1500?

```
SELECT *
FROM EMP
WHERE JOB='MANAGER' AND
        SAL<1500;
```

Q25. WAQTD names of an employee working as a 'SALESMAN' in department no. 30?

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

Q26. WAQTD details of an employee working as a 'MANAGER' or earning less than Rs.2000.

```
SELECT *
```

FROM EMP

WHERE JOB=’MANAGER’ OR

SAL<2000;

```
SQL> SELECT *
  2  FROM EMP
  3  WHERE JOB='MANAGER' OR
  4   SAL<2000;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

11 rows selected.
```

Q27. WAQTD names of an employee working in department 10, 20.

SELECT ENAME

FROM EMP

WHERE DEPTNO=10 OR

DEPTNO=20;

Q28. WAQTD all the details along with annual salary if employee is working as ‘SALESMAN’ & earning less than Rs.1250 & annual salary more than Rs.14,000.

SELECT EMP.*, SAL*12 ANNUAL_SAL

FROM EMP

WHERE JOB=’SALESMAN’ AND

SAL<1250 AND

SAL*12>14000;

Q29. WAQTD details of an employee working as a ‘MANAGER’ in department no. 30.

SELECT *

FROM EMP

WHERE JOB=’MANAGER’ AND

DEPTNO=30;

Q30. WAQTD details of an employee as an ‘ANALYST’ & earning Rs.3000.

```
SELECT *
FROM EMP
WHERE JOB='ANALYST' AND
        SAL<3000;
```

Q31. WAQTD names of an employee working as a 'CLERK' & hired before 1981.

```
SELECT ENAME
FROM EMP
WHERE JOB='CLERK' AND
HIREDATE<='31-DEC-1980';
```

Q32. WAQTD details of an employee working as a 'PRESIDENT' or earning more than Rs.3000.

```
SELECT *
FROM EMP
WHERE JOB='PRESIDENT' OR
        SAL>3000;
```

Q33. WAQTD names of an employee hired after 1982 into department no. 20.

```
SELECT ENAME
FROM EMP
WHERE HIREDATE>='01-JAN-1983' AND
        DEPTNO=20;
```

Q34. WAQTD all the details of an employee along with the annual salary if employee working as a 'CLERK' in department no. 20 & annual salary must be less than Rs.15000.

```
SELECT EMP.*, SAL*12 ANNUAL_SAL
FROM EMP
WHERE JOB='CLERK' AND
        DEPTNO=20 AND
        SAL*12<15000;
```

Q35. WAQTD details of an employee working as 'PRESIDENT' or 'SALESMAN' or 'ANALYST'.

```
SELECT *
FROM EMP
WHERE JOB='PRESIDENT' OR
        JOB='SALESMAN' OR
```

```
    JOB='ANALYST';
```

Q36. WAQTD names of an employee hired after 1981 & earning more than Rs.2000.

```
SELECT ENAME
FROM EMP
WHERE HIREDATE>='01-JAN-1981' AND
    SAL>2000;
```

Q37. WAQTD details of an employee working as 'ANALYST' or working in department no. 10.

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

Q38. WAQTD names of an employee hired after 1981 & before 1987.

```
SELECT ENAME
FROM EMP
WHERE HIREDATE>='01-JAN-1982' AND
    HIREDATE<='31-DEC-1986';
```

Q39. WAQTD names of an employee hired in the year 1982.

```
SELECT ENAME
FROM EMP
WHERE HIREDATE>='01-JAN-1982' AND
    HIREDATE<='31-DEC-1982';
```

'IN' OPERATOR

'IN' operator is a multi-valued operator in which we can *pass* multiple values at RHS.

- 'IN' operator *returns* TRUE If *any one* of the *conditions* is satisfied.
- 'IN' operator allows the value present at the LHS to be compared with all the value present at RHS.

Syntax:

```
COLUMN_NAME/ EXPRESSION IN (V1, V2, V3, ……., Vn);
```

Q40. WAQTD details of an employee working in department no. 10,20,30,40,50,60.

```
SELECT *
```

FROM EMP

WHERE DEPTNO IN (10,20,30,40,50,60);

Q41. WAQTD names of an employee working as 'MANAGER' or 'SALESMAN' or 'ANALYST'.

SELECT ENAME

FROM EMP

WHERE JOB IN ('MANAGER', 'SALESMAN', 'ANALYST');

```
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE JOB IN ('MANAGER', 'SALESMAN', 'ANALYST');

ENAME
----------
ALLEN
WARD
JONES
MARTIN
BLAKE
CLARK
SCOTT
TURNER
FORD
```


'NOT IN' OPERATOR

'NOT IN' operator is similar to 'IN' operator but it rejects the value instead of selecting it.

Syntax:

COLUMN_NAME/ EXPRESSION NOT IN (V1, V2, V3, ……, V_n)

Q42. WAQTD details of an employee excluding the employees working in department no. 10 or 20.

SELECT *

FROM EMP

WHERE DEPTNO NOT IN (10,20);

Q43. WAQTD details of an employee working in department no. 20 or 30 except 'SALESMAN' or 'CLERK'.

SELECT *

FROM EMP

WHERE DEPTNO IN (20,30) AND

JOB NOT IN ('SALESMAN', 'CLERK');

```
SQL> SELECT *
  2  FROM EMP
  3  WHERE DEPTNO IN (20,30) AND
  4   JOB NOT IN ('SALESMAN', 'CLERK');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
```

'BETWEEN' OPERATOR

We *use* 'BETWEEN' operator whenever we have a range.

- '**BETWEEN**' operator works *including* the **range values**.
- **The range** *cannot* be **interchanged**.

Syntax:

COLUMN_NAME / EXPRESSIONS BETWEEN LOWER_RANGE AND HIGHER_RANGE;

Q44. WAQTD details of an employee hired in the year 1982.

SELECT *

FROM EMP

WHERE HIREDATE BETWEEN '01-JAN-1982' AND '31-DEC-1982';

```
SQL> SELECT *
  2  FROM EMP
  3  WHERE HIREDATE BETWEEN '01-JAN-1982' AND '31-DEC-1982';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
```

Q45. WAQTD name & hire date of an employee hired after 1981 & before 1987.

SELECT ENAME, HIREDATE

FROM EMP

WHERE HIREDATE BETWEEN '01-JAN-1982' AND '31-DEC-1986';

```
SQL> SELECT ENAME, HIREDATE
  2  FROM EMP
  3  WHERE HIREDATE BETWEEN '01-JAN-1982' AND '31-DEC-1986';

ENAME      HIREDATE
---------- ---------
MILLER     23-JAN-82
```

Q46. WAQTD details of an employee earning more than Rs.200 and hired after 1982.

```
SELECT *
FROM EMP
WHERE SAL>200 AND
        HIREDATE>='01-JAN-1983';
```

'LIKE' OPERATOR

'**LIKE**' operator is used whenever we need to "***Match the pattern***".

Syntax:

```
COLUMN_NAME/ EXPRESSION LIKE 'PATTERN_TO_MATCH';
```

To achieve the pattern matching we use special characters such as

1. **PERCENTILE (%):** It can accept any *character* any number of times or, no character.
2. **UNDERSCORE (_):** It can accept any character but *only* once.

Q47. WAQTD details of an employee whose name starts with 'S'.

```
SELECT *
FROM EMP
WHERE ENAME LIKE 'S%';
```

```
SQL> SELECT *
  2  FROM EMP
  3  WHERE ENAME LIKE 'S%';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
```

Q48. WAQTD details of an employee whose name ends with 'S'.

```
SELECT *
FROM EMP
WHERE ENAME LIKE '%S';
```

```
SQL> SELECT *
  2  FROM EMP
  3  WHERE ENAME LIKE '%S';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
```

Q49. WAQTD details of an employee who are having exactly 5 characters.

SELECT *

FROM EMP

WHERE ENAME LIKE '_ _ _ _ _';

```
SQL> SELECT *
  2  FROM EMP
  3  WHERE ENAME LIKE '_____';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30

8 rows selected.
```

Q50. WAQTD names of an employee whose name does not start with 'S'.

SELECT ENAME

FROM EMP

WHERE ENAME NOT LIKE 'S%';

```
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE ENAME NOT LIKE 'S%';

ENAME
----------
ALLEN
WARD
JONES
MARTIN
BLAKE
CLARK
KING
TURNER
ADAMS
JAMES
FORD
MILLER

12 rows selected.
```

Q51. WAQTD employee name if the employee having character 'A' in the second place.

SELECT ENAME

FROM EMP

WHERE ENAME LIKE '_A%';

Q52. WAQTD details of an employee having 'A' in 2nd last place.

SELECT *

FROM EMP

WHERE ENAME LIKE '%A_';

Q53. WAQTD details of an employee having 'S' in the last place.

SELECT *

FROM EMP

WHERE ENAME LIKE '%S';

Q54. WAQTD names of an employee having 'E' in the 4th place.

SELECT ENAME

FROM EMP

WHERE ENAME LIKE '_ _ _E%';

```
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE ENAME LIKE '___E%';

ENAME
----------
ALLEN
JONES
JAMES
```

Q55. WAQTD names of an employee having character 'A' in first place & 'S' in the last place.

SELECT ENAME

FROM EMP

WHERE ENAME LIKE 'A%S';

Q56. WAQTD name, salary if they were hired in the year 82.

SELECT ENAME, SAL

FROM EMP

WHERE HIREDATE LIKE '%82';

```
SQL> SELECT ENAME, SAL
  2  FROM EMP
  3  WHERE HIREDATE LIKE '%82';

ENAME             SAL
---------- ----------
MILLER           1300
```

Q57. WAQTD name, salary if they are earning 3 digits salary.

SELECT ENAME, SAL

FROM EMP

WHERE SAL LIKE '_ _ _';

```
SQL> SELECT ENAME, SAL
  2  FROM EMP
  3  WHERE SAL LIKE '___';

ENAME             SAL
---------- ----------
SMITH             800
JAMES             950
```

Q58. WAQTD details of an employee who are having character 'A' in first place or employees if they are having character 'S' in the last place.

SELECT *

FROM EMP

WHERE ENAME LIKE ‘A%’ OR

ENAME LIKE ‘%S’;

Q59. WAQTD names of an employee if they were having ‘A’ in the first place & working in department no. 10 or 20.

SELECT ENAME

FROM EMP

WHERE ENAME LIKE ‘A%’ AND

DEPTNO IN (10,20);

Q60. WAQTD name, salary if they are earning 4 digits salary.

SELECT ENAME, SAL

FROM EMP

WHERE SAL LIKE ‘_ _ _ _’;

Q61. WAQTD names of an employee, hire date if they are hired in the year 81 & earning salary more than Rs.2000.

SELECT ENAME, HIREDATE

FROM EMP

WHERE HIREDATE LIKE ‘%82’ AND

SAL>2000;

Q62. WAQTD details of an employee if they having string ‘MAN’ in their job.

SELECT *

FROM EMP

WHERE JOB LIKE ‘%MAN%’;

```
SQL> SELECT *
  2  FROM EMP
  3  WHERE JOB LIKE '%MAN%';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30

7 rows selected.
```

Q63. WAQTD details of an employee if they are having 'A' in first place, 'D' in second place & 'S' in last place in their names.

```
SELECT *
FROM EMP
WHERE ENAME LIKE 'AD%S';
```

Q64. WAQTD names of an employee working in department no. 30 or 40 & earning more than 2000.

```
SELECT ENAME
FROM EMP
WHERE DEPTNO IN (30,40) AND
      SAL >2000;
```

Q65. WAQTD details of an employee working as 'SALESMAN' or 'CLERK' or 'MANAGER' in department no. 10.

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

Q66. WAQTD details of an employee working as 'SALESMAN' or 'MANAGER' in department no. 20 or 30 & hired in the year 83.

```
SELECT *
FROM EMP
WHERE JOB IN ('SALESMAN', 'MANAGER') AND
      DEPTNO IN (20,30) AND
      HIREDATE BETWEEN '01-JAN-83' AND '31-DEC-83';  [or, HIREDATE LIKE '%83';]
```

Q67. WAQTD details of an employee working as 'SALESMAN' or 'PRESIDENT' in department no. 20 & hired after 1983.

```
SELECT *
FROM EMP
WHERE JOB IN ('SALESMAN', 'PRESIDENT') AND
      DEPTNO=20 AND
      HIREDATE>='01-JAN-1984';
```

Q68. WAQTD all details of an employee along with annual salary if annual salary is more than 14000 & working as 'SALESMAN' or 'ANALYST' in department no. 20 or 30 & hired in the year 1982.

```
SELECT EMP.*, SAL*12 ANNUAL_SAL
```

FROM EMP

WHERE SAL*12>14000 AND

JOB IN (‘SALESMAN’, ‘ANALYST’) AND

DEPTNO IN (20,30) AND

HIREDATE LIKE ‘%1982’;

Q69. WAQTD details of an employee working in department no. 20 or 30 & hired after 1981 before 1988.

SELECT *

FROM EMP

WHERE DEPTNO IN (20,30) AND

HIREDATE BETWEEN ’01-JAN-1982’ AND ’31-DEC-1987’;

Q70. WAQTD names of an employee working in department no. 10 or 20 except ‘MANAGER’ or ‘ANALYST’ & hired after 1981 except the employees hired in the year 1987.

SELECT ENAME

FROM EMP

WHERE DEPTNO IN (10,20) AND

JOB NOT IN (‘MANAGER’, ‘ANALYST’) AND

HIREDATE>=’01-JAN-1982’ AND

HIREDATE NOT BETWEEN ’01-JAN-1987’ AND ’31-DEC-1987’;

```
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE DEPTNO IN (10,20) AND
  4   JOB NOT IN ('MANAGER', 'ANALYST') AND
  5   HIREDATE>='01-JAN-1982' AND
  6   HIREDATE NOT BETWEEN '01-JAN-1987' AND '31-DEC-1987';

ENAME
----------
MILLER
```

Q71. WAQTD all the details of an employee along with annual salary if the annual salary is more than Rs.15000 in department 10 or 20 except employees hired in the year 1982.

SELECT EMP.*, SAL*12 ANNUAL_SAL

FROM EMP

WHERE SAL*12>15000 AND

DEPTNO IN (10,20) AND

HIREDATE NOT BETWEEN '01-JAN-1982' AND '31-DEC-1982';

Q72. WAQTD all the detail of an employee hired after 1980 before 1988 except the employees hired in the year 1982.

SELECT *

FROM EMP

WHERE HIREDATE BETWEEN '01-JAN-1981' AND '31-DEC-1987' AND

HIREDATE NOT BETWEEN '01-JAN-1982' AND '31-DEC-1982'; [or, HIREDATE NOT LIKE '%82';]

'IS' OPERATOR

'**IS**' operator is used only to compare with **NULL**.

Syntax:

COLUMN_NAME/ EXPRESSION IS NULL;

Q73. WAQTD names of an employee who doesn't get any commission.

SELECT ENAME

FROM EMP

WHERE COMM IS NULL;

```
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE COMM IS NULL;

ENAME
----------
SMITH
JONES
BLAKE
CLARK
SCOTT
KING
ADAMS
JAMES
FORD
MILLER

10 rows selected.
```

Q74. WAQTD names of an employee who doesn't get any salary.

SELECT ENAME

FROM EMP

WHERE SAL IS NULL;

Q75. WAQTD names of an employee who gets commission.

```
SELECT ENAME

FROM EMP

WHERE COMM IS NOT NULL;
```

```
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE COMM IS NOT NULL;

ENAME
----------
ALLEN
WARD
MARTIN
TURNER
```

Q76. WAQTD names of an employee working as a ''PRESIDENT or 'MANAGER' except the employees working in department no. 30 & hired after 1980 before 1987 except the employees hired in the year 1982 & employees must be earning more than Rs.3000 & name has second character as 'I' & he does not get any commission.

```
SELECT ENAME

FROM EMP

WHERE JOB IN ('PRESIDENT', 'MANAGER') AND

        DEPTNO NOT IN 30 AND

        HIREDATE BETWEEN '01-JAN-1981' AND '31-DEC-1986' AND

        HIREDATE NOT LIKE '%82' AND

        SAL>3000 AND

        ENAME LIKE '_I%' AND

        COMM IS NULL;
```

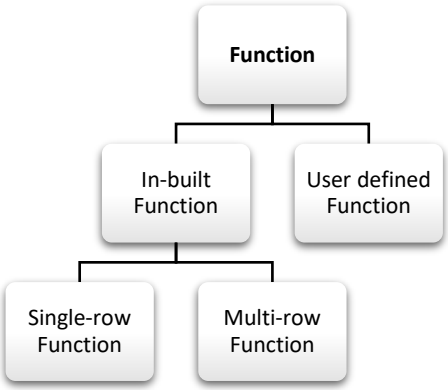
Q77. WAQTD names of an employee working as 'SALESMAN' or 'CLERK' in department no. 20 or 30 & hired in the month of September & earns the commission of Rs.1400 & name has 2nd character 'A' & 2nd last character 'I'.

```
SELECT ENAME

FROM EMP

WHERE JOB IN ('SALESMAN', 'CLERK') AND

        DEPTNO IN (20,30) AND

        HIREDATE LIKE '%SEP' AND

        COMM=1400 AND
```

ENAME LIKE '_A%I_';

FUNCTIONS

It is a list of instructions that are used to perform the specific task.



SINGLE-ROW FUNCTION

- Single-row function executes row-by-row.
- It takes one input executes & generates one output, then goes to the next.
- If we pass 'n' number of inputs to single-row function, it returns 'n' number of output.

There are some commonly used Single-row Functions listed Below:

- **DUAL:** It is a dummy table that is *present* in all the database to *perform* any operations.
- **LOWER ():** This function is *used* to convert the *string* from uppercase to lowercase.

  Syntax: LOWER ('STRING')

  Example: SELECT LOWER ('HI! NISHA')

  FROM DUAL; //**output:** hi! nisha

- **UPPER ():** This function is *used* to convert the *string* from lowercase to uppercase.

  Syntax: UPPER ('string')

  Example: SELECT UPPER ('hi! nisha')

  FROM DUAL; //**output:** HI! NISHA

- **INITCAP ():** This function is *used* to convert the initial characters into uppercase.

  Syntax: INITCAP ('STRING')

  Example: SELECT INITCAP ('HI! NISHA')

  FROM DUAL; //**output:** Hi! nisha

- **REVERSE ():** This function is *used* to reverse a string.

  Syntax: REVERSE ('STRING')

Example: SELECT REVERSE ('HI! NISHA')

FROM DUAL; //**output:** AHSIN !IH

- **LENGTH ():** This function is *used* to find the number of characters in a *given* string.

  Syntax: LENGTH ('STRING')

  Example: SELECT LENGTH ('HI! NISHA')

  FROM DUAL; //**output:** 9

- **SUBSTR ():** This function is *used* to extract the part of the string from the *given* original string.

  Syntax: SUBSTR ('ORIGINAL_STRING', POSITION, [LENGTH])

  Example:

  -9 -8 -7 -6 -5 -4 -3 -2 -1

  1 2 3 4 5 6 7 8 9

  B E N G A L U R U

  SUBSTR ('BENGALURU',1,3) // **BEN**

  SUBSTR ('BENGALURU', 3, 2) // **NG**

  SUBSTR ('BENGALURU', 1, 1) // **B**

  SUBSTR ('BENGALURU', 1) // **BENGALURU**

  SUBSTR ('BENGALURU', 10, 2) // **_ _**

  SUBSTR ('BENGALURU', 8) // **RU**

  SUBSTR ('BENGALURU', -3, 2) // **UR**

  SUBSTR ('BENGALURU', -6, 3) // **GAL**

  SUBSTR ('BENGALURU', -4) // **LURU**

- **SYSDATE:** It is *used* to find the date present in the system.

  Syntax: SELECT SYSDATE

  FROM DUAL;

```
SQL> SELECT SYSDATE
  2    FROM DUAL;

SYSDATE
---------
11-SEP-22
```

- **SYSTIMESTAMP:** This function is *used* to find the date & time *along* with the time zone.

  Syntax: SELECT SYSTIMESTAMP

FROM DUAL;

```
SQL> SELECT SYSTIMESTAMP
  2                 FROM DUAL;

SYSTIMESTAMP
-----------------------------------
11-SEP-22 01.56.45.791000 AM +05:30
```

- **TO_CHAR():** This function is *used* to convert the given date to string format.

  Syntax: TO_CHAR (DATE, 'FORMAT_MODELS');

  FORMAT TOOLS

  1. YEAR
  2. YYYY
  3. YY
  4. MONTH
  5. MON
  6. MM
  7. DAY
  8. DY
  9. DD
  10. D
  11. HH24
  12. HH12
  13. MI
  14. SS

- TO_CHAR(SYSDATE, 'YEAR') //**Twenty Twenty Two**
- TO_CHAR(SYSDATE, 'YYYY') //**2022**
- TO_CHAR(SYSDATE, 'YY') //**22**
- TO_CHAR(SYSDATE, 'MONTH') //**SEPTEMBER**
- TO_CHAR(SYSDATE, 'MON') //**SEP**
- TO_CHAR(SYSDATE, 'MM') //**09**
- TO_CHAR(SYSDATE, 'DAY') //**SATURDAY** (***NOT RECOMMENDED***)
- TO_CHAR(SYSDATE, 'DY') //**SAT** (***PREFERRABLE***)
- TO_CHAR(SYSDATE, 'DD') //**10**
- TO_CHAR(SYSDATE, 'D') //**7**
- TO_CHAR(SYSDATE, 'HH24') //**9**
- TO_CHAR(SYSDATE, 'HH12') //**9**
- TO_CHAR(SYSDATE, 'MI') //**32**
- TO_CHAR(SYSDATE, 'SS') //**Prints the current 'seconds' time of the system**

- **INSTR():** This function is *used* to *obtain* index value of the substring which is *present* in the original string. (Index value- position of the character)

  Syntax:

  INSTR ('ORIGINAL_STRING', 'SUB_STR', POSITION, [N^{TH} OCCURENCE])

Nth OCCURRENCE – Number of times it is present.

Example: B_1 A_2 N_3 A_4 N_5 A_6 **//Subscript represents the index value**

INSTR ('BANANA', 'A', 1, 1) //2 **> 0**

INSTR ('BANANA', 'A', 1, 2) //4 **> 0**

INSTR ('BANANA', 'A', 3, 1) //4 **> 0**

INSTR ('BANANA', 'N', 3, 1) //3 **> 0**

INSTR ('BANANA', 'N', 4) //5 **> 0**

INSTR ('BANANA', 'B', 1) //1 **> 0**

INSTR ('BANANA', 'NA', 1, 1) //3 **> 0**

INSTR ('BANANA', 'AN', 1, 2) //4 **> 0**

INSTR ('BANANA', 'Z', 2, 1) //0 **= 0**

INSTR ('BANANA', 'A', 2, 4) //0 **= 0**

**** Here, > 0 (*is a condition if occurrence is greater than 0*) & = 0 (*is a condition if occurrence is equal to 0*).**

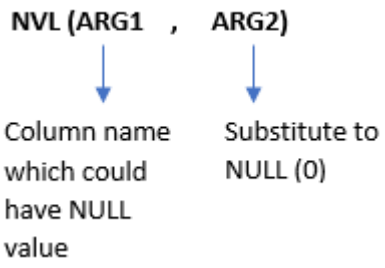
- **NVL (NULL VALUE LOGIC)**

  It can *accept* 2 arguments.

  In ARG1, we must *write* a COLUMN_NAME or EXPRESSION that *can* be NULL.

  In ARG2, we must *write* a value that *can* be substituted in place of NULL.

  If ARG1 is NOT NULL, **NVL** *returns* same value *present* in ARG.



- **MOD ():** It is a function which is *used* to find the modulus of a given number.

  Syntax: MOD (m, n)

  Example:

MOD (7, 2)

```
    3
  ------
2 | 7
    6
  ------
    1
```

MOD (8, 2)

```
    4
  ------
2 | 8
    8
  ------
    0
```

- **REPLACE ():** This function is *used* to replace the substring with new string in *given* original string.
  Syntax: REPLACE ('ORIGINAL_STRING', 'SUB_STR', ['NEW_STR'])

  Example: 1. SELECT REPLACE ('QSPIDERS', 'S', 'J')

  FROM DUAL; //**output:** JSPIDERS

  2. SELECT REPLACE ('QSPIDERS', 'Q')

  FROM DUAL; //**output:** SPIDERS

MULTI-ROW FUNCTION

- Also, known as '***Group Functions***' or '***Aggregate Functions***'.
- It takes all the inputs *at once* aggregates (combines) it & generates one output.
- If you provide 'n' number of inputs to multi-row function, it *returns* a single output.

**** In interview, interviewer never use the name multi-row function, they will simply ask what is Group Function.**

Note:

- **Multi-row functions** can *accept* only a *single argument* that is a COLUMN_NAME or, an EXPRESSION.
- **MAX ()** & **MIN ()** functions can be *used* for all the *following* datatypes problems. **i.e.,** CHAR, VARCHAR, NUMBER & DATE.
- **SUM ()** & **AVG ()** functions can *only* take NUMBER COLUMN as an *argument*.
- Multi-row functions will *ignore* the **NULL value**.
- We ***can't*** use multi-row functions in **WHERE** clause.
- We ***can't*** use any **COLUMN_NAME** with **multi-row functions** in **SELECT** clause.
- **COUNT ()** is the *only* **multi-row function** to which we can **pass** *Asterisk (*)* as an **argument**.

LIST OF MULTI-ROW FUNCTIONS

- MAX ()
- MIN ()
- SUM ()
- AVG ()

- COUNT ()

Q78. WAQTD the maximum salary in EMP table.

SELECT MAX (SAL)

FROM EMP;

```
SQL> SELECT MAX (SAL)
  2  FROM EMP;

  MAX(SAL)
----------
      5000
```

Q79. WAQTD the maximum salary in Department number 10.

SELECT MAX (SAL)

FROM EMP

WHERE DEPTNO=10;

Q80. WAQTD minimum salary in EMP table.

SELECT MIN (SAL)

FROM EMP;

```
SQL> SELECT MIN (SAL)
  2  FROM EMP;

  MIN(SAL)
----------
       800
```

Q81. WAQTD number of employees earning Rs.3000 in Department number 20.

SELECT COUNT (*)

FROM EMP

WHERE SAL>3000 AND

DEPTNO=20;

```
SQL> SELECT COUNT (*)
  2  FROM EMP
  3  WHERE SAL>3000 AND
  4  DEPTNO=20;

  COUNT(*)
----------
         0
```

Q82. WAQTD number of employees earning more than Rs.3000.

SELECT COUNT (*)

FROM EMP

WHERE SAL>3000;

Q83. WAQTD maximum salary given to the manager.

SELECT MAX (SAL)

FROM EMP

WHERE JOB IN ('MANAGER');

Q84. WAQTD number of employees hired in the year 1987.

SELECT COUNT (*)

FROM EMP

WHERE HIREDATE LIKE '%1987';

Q85. WAQTD average salary given to the 'PRESIDENT'.

SELECT AVG (SAL)

FROM EMP

WHERE JOB IN ('PRESIDENT');


```
SQL> SELECT AVG (SAL)
  2  FROM EMP
  3  WHERE JOB IN ('PRESIDENT');

  AVG(SAL)
----------
      5000
```

Q86. WAQTD number of employees working as a 'MANAGER' or, 'SALESMAN'.

SELECT COUNT (*)

FROM EMP

WHERE JOB IN ('MANAGER', 'SALESMAN');

Q87. WAQTD number of employees hired in the month of December.

SELECT COUNT (*)

FROM EMP

WHERE HIREDATE LIKE '%DEC%';

Q88. WAQTD maximum salary given to the employee hired in the month of February.

SELECT MAX (SAL)

FROM EMP

WHERE HIREDATE LIKE '%FEB%';

Q89. WAQTD total salary given to an employee who is having two consecutive 'T' in their name.

SELECT SUM (SAL)

FROM EMP

WHERE ENAME LIKE '%TT%';

```
SQL> SELECT SUM (SAL)
  2  FROM EMP
  3  WHERE ENAME LIKE '%TT%';

  SUM(SAL)
----------
      3000
```

Q90. WAQTD total salary given to the employees whose name ends with character 'N'.

SELECT SUM (SAL)

FROM EM

WHERE ENAME LIKE '%N';

Q91. WAQTD number of employees whose name starts with character 'A'.

SELECT COUNT (*)

FROM EMP

WHERE ENAME LIKE 'A%';

Q92. WAQTD number of employees earning more than Rs.3000 in Department number 10 & Name must start with 'K' & ends with 'G'.

SELECT COUNT (*)

FROM EMP

WHERE SAL>3000 AND

DEPTNO=10 AND

ENAME LIKE 'K%G';

'CONCATENATION' OPERATOR

This operator is *used* to join the *given* two strings.

Q93. WAQTD the output in the following format,

a. **'MR. ABC YOUR SALARY IS RS.XYZ'**

SELECT 'MR.'||ENAME||' YOUR SALARY IS RS.'||SAL

FROM EMP;

```
SQL> SELECT 'MR.'||ENAME||' YOUR SALARY IS RS.'||SAL
  2  FROM EMP;

'MR.'||ENAME||'YOURSALARYISRS.'||SAL
----------------------------------------------------
MR.SMITH YOUR SALARY IS RS.800
MR.ALLEN YOUR SALARY IS RS.1600
MR.WARD YOUR SALARY IS RS.1250
MR.JONES YOUR SALARY IS RS.2975
MR.MARTIN YOUR SALARY IS RS.1250
MR.BLAKE YOUR SALARY IS RS.2850
MR.CLARK YOUR SALARY IS RS.2450
MR.SCOTT YOUR SALARY IS RS.3000
MR.KING YOUR SALARY IS RS.5000
MR.TURNER YOUR SALARY IS RS.1500
MR.ADAMS YOUR SALARY IS RS.1100
MR.JAMES YOUR SALARY IS RS.950
MR.FORD YOUR SALARY IS RS.3000
MR.MILLER YOUR SALARY IS RS.1300

14 rows selected.
```

b. **'MR. A YOUR SAL IS RS. B AND YOUR ANNUAL SALARY IS RS.C'**

SELECT 'MR.'||ENAME||' YOUR SAL IS RS.'||SAL||' AND YOUR ANNUAL SALARY IS RS.'||SAL*12

FROM EMP;

```
SQL> SELECT 'MR.'||ENAME||' YOUR SAL IS RS.'||SAL||' AND YOUR ANNUAL SALARY IS RS.'||SAL*12
  2  FROM EMP;

'MR.'||ENAME||'YOURSALISRS.'||SAL||'ANDYOURANNUALSALARYISRS.'||SAL*12
--------------------------------------------------------------------------------
MR.SMITH YOUR SAL IS RS.800 AND YOUR ANNUAL SALARY IS RS.9600
MR.ALLEN YOUR SAL IS RS.1600 AND YOUR ANNUAL SALARY IS RS.19200
MR.WARD YOUR SAL IS RS.1250 AND YOUR ANNUAL SALARY IS RS.15000
MR.JONES YOUR SAL IS RS.2975 AND YOUR ANNUAL SALARY IS RS.35700
MR.MARTIN YOUR SAL IS RS.1250 AND YOUR ANNUAL SALARY IS RS.15000
MR.BLAKE YOUR SAL IS RS.2850 AND YOUR ANNUAL SALARY IS RS.34200
MR.CLARK YOUR SAL IS RS.2450 AND YOUR ANNUAL SALARY IS RS.29400
MR.SCOTT YOUR SAL IS RS.3000 AND YOUR ANNUAL SALARY IS RS.36000
MR.KING YOUR SAL IS RS.5000 AND YOUR ANNUAL SALARY IS RS.60000
MR.TURNER YOUR SAL IS RS.1500 AND YOUR ANNUAL SALARY IS RS.18000
MR.ADAMS YOUR SAL IS RS.1100 AND YOUR ANNUAL SALARY IS RS.13200
MR.JAMES YOUR SAL IS RS.950 AND YOUR ANNUAL SALARY IS RS.11400
MR.FORD YOUR SAL IS RS.3000 AND YOUR ANNUAL SALARY IS RS.36000
MR.MILLER YOUR SAL IS RS.1300 AND YOUR ANNUAL SALARY IS RS.15600

14 rows selected.
```

'GROUP BY' CLAUSE

- 'GROUP BY' clause is *use* to *group the records*.
- It *executes* row-by-row.

- For 'GROUP BY' clause, we can *pass* COLUMN_NAME or, an EXPRESSION as an *argument*.
- We can write 'GROUP BY' expression *along* with the multi-row function in 'SELECT' clause.

'GROUP BY' EXPRESSION

- Any COLUMN_NAME or, an EXPRESSION that is *written* in 'GROUP BY' clause is known as 'GROUP BY' expression.
- *After* the *execution* of 'GROUP BY' clause, it *creates* group & *anything* that *executes after* 'GROUP BY' clause *executes* '**GROUP-BY-GROUP**'.

Syntax:

SELECT GROUP_BY_EXPRESSION / GROUP_FUNCTION

FROM TABLE_NAME

[WHERE <filter-condition>]

GROUP BY COLUMN_NAME / EXPRESSION

Order of Execution:

1. FROM
2. WHERE (*if used*) [row-by-row]
3. GROUP BY [row-by-row]
4. SELECT [group-by-group]

Q94. WAQTD maximum salary & DEPTNO in each Department number.

SELECT MAX (SAL), DEPTNO

FROM EMP

GROUP BY DEPTNO;

```
SQL> SELECT MAX (SAL), DEPTNO
  2  FROM EMP
  3  GROUP BY DEPTNO;

  MAX(SAL)     DEPTNO
---------- ----------
      2850         30
      3000         20
      5000         10
```

Q95. WAQTD average salary of employees in each job.

SELECT JOB, AVG (SAL)

FROM EMP

GROUP BY JOB;

```
SQL> SELECT JOB, AVG (SAL)
  2  FROM EMP
  3  GROUP BY JOB;

JOB         AVG(SAL)
--------- ----------
CLERK         1037.5
SALESMAN        1400
PRESIDENT       5000
MANAGER   2758.33333
ANALYST         3000
```

Q96. WAQTD number of employees in each Department if the employees are earning more than Rs.2000.

SELECT COUNT (*), DEPTNO

FROM EMP

WHERE SAL>2000

GROUP BY DEPTNO;

```
SQL> SELECT COUNT (*), DEPTNO
  2  FROM EMP
  3  WHERE SAL>2000
  4  GROUP BY DEPTNO;

  COUNT(*)     DEPTNO
---------- ----------
         1         30
         3         20
         2         10
```


Q97. WAQTD number of employees working in each department except 'PRESIDENT'.

SELECT COUNT (*), DEPTNO

FROM EMP

WHERE JOB NOT IN ('PRESIDENT')

GROUP BY DEPTNO;

Q98. WAQTD number of employees working as 'MANAGER' in each department.

SELECT COUNT (*), DEPTNO

FROM EMP

WHERE JOB IN ('MANAGER')

GROUP BY DEPTNO;

Q99. WAQTD average salary needed to pay to all the employees excluding the employees working in department number 20 in each job.

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

Q100. WAQTD number of employees having character 'A' in their name in each job.

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

'HAVING' CLAUSE

- We use '**HAVING**' clause to **filter** the ***group***.
- We can *pass* **multi-row function** condition in '**HAVING**' clause.
- It executes **group-by-group**.
- If you are using '**HAVING**' clause it must be used ***after*** the '**GROUP BY**' clause.
- '**HAVING**' clause **executes** *after* the **execution** of '**GROUP BY**' clause.

Q101. WAQTD number of employees working in each department if there are at least 2 employees in each department.

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

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

  COUNT(*)     DEPTNO
---------- ----------
         6         30
         5         20
         3         10
```

Q102. WAQTD total salary needed to pay all the employees in each job.

```
SELECT SUM (SAL), JOB
FROM EMP
```

GROUP BY JOB;

Q103. WAQTD number of employees & average salary needed to pay the employees whose salary is greater than Rs.2000 in each department.

SELECT COUNT (*), AVG (SAL), DEPTNO

FROM EMP

WHERE SAL>2000

GROUP BY DEPTNO;

Q104. WAQTD number of employees & total salary given to all the salesman in each department.

SELECT COUNT (*), SUM (SAL), DEPTNO

FROM EMP

WHERE JOB IN ('SALESMAN')

GROUP BY DEPTNO;

Q105. WAQTD number of employees with their maximum salaries in each job.

SELECT COUNT (*), MAX (SAL), JOB

FROM EMP

GROUP BY JOB;

Q106. WAQTD maximum salaries given to an employee working in each department

SELECT MAX (SAL), DEPTNO

FROM EMP

GROUP BY DEPTNO;

Q107. WAQTD number of times salaries are present in employee table.

SELECT COUNT (*), SAL

FROM EMP

GROUP BY SAL;

Q108. WAQTD the number & total salary needed to pay all employees in each department if there are at least 4 employees in each department.

SELECT COUNT (*), SUM (SAL), DEPTNO

FROM EMP

GROUP BY DEPTNO

HAVING COUNT (*)>=4;

Q109. WAQTD number of employees earning salary more than Rs.1200 in each job & the total salary needed to pay employee of each job must exceed Rs.3800

SELECT COUNT (*), JOB

FROM EMP

WHERE SAL>1200

GROUP BY JOB

HAVING SUM (SAL)>3800;

```
SQL> SELECT COUNT (*), JOB
  2  FROM EMP
  3  WHERE SAL>1200
  4  GROUP BY JOB
  5  HAVING SUM (SAL)>3800;

  COUNT(*) JOB
---------- ---------
         4 SALESMAN
         1 PRESIDENT
         3 MANAGER
         2 ANALYST
```

Q110. WAQTD DEPTNO & number of employees working only if there are 2 employees working in each department as Manager.

SELECT COUNT (*), DEPTNO

FROM EMP

WHERE JOB IN ('MANAGER')

GROUP BY DEPTNO

HAVING COUNT (*) = 2;

Q111. WAQTD to display repeated salaries in the EMP table.

SELECT SAL

FROM EMP

GROUP BY SAL

HAVING COUNT (*) >= 2; [or, COUNT (*) > 1;]

```
SQL> SELECT SAL
  2  FROM EMP
  3  GROUP BY SAL
  4  HAVING COUNT (*) >= 2;

       SAL
----------
      1250
      3000
```

Q112. WAQTD the hire date which are repeated in EMP table.

SELECT HIREDATE

FROM EMP

GROUP BY HIREDATE

HAVING COUNT (*) > 1;

```
SQL> SELECT HIREDATE
  2  FROM EMP
  3  GROUP BY HIREDATE
  4  HAVING COUNT (*) > 1;

HIREDATE
---------
03-DEC-81
```

'ORDER BY' CLAUSE

It is *used* to sort the records in ascending or descending order.

- '**ORDER BY**' clause must be *written* as **last clause** in the statement.
- '**ORDER BY**' clause *executes* **after** the '**SELECT**' clause.
- By *default*, '**ORDER BY**' clause *sort* the records in ascending order.
- We can pass COLUMN_NAME or EXPRESSION as an *argument* in 'ORDER BY' clause.
- We can *pass* **ALIAS** name in '**ORDER BY**' clause.

Syntax:

SELECT GROUP_BY_EXPRESSION / GROUP_FUNCTION

FROM TABLE_NAME

[WHERE <filter-condition>]

[GROUP BY COLUMN_NAME / EXPRESSION]

[HAVING <group_filter_condition>]

ORDER BY COLUMN_NAME [ASC] / [DESC];

Order of Execution:

1. FROM
2. WHERE [if used] (row-by-row)
3. GROUP BY [if used] (row-by-row)
4. HAVING [if used] (group-by-group)
5. SELECT (group-by-group)
6. ORDER BY

Q113. WAQTD salary in descending order.

SELECT SAL

FROM EMP

ORDER BY SAL DESC;

```
SQL> SELECT SAL
  2  FROM EMP
  3  ORDER BY SAL DESC;

       SAL
----------
      5000
      3000
      3000
      2975
      2850
      2450
      1600
      1500
      1300
      1250
      1250
      1100
       950
       800

14 rows selected.
```


Q114. WAQTD annual salary in ascending order.

SELECT SAL*12 ANNUALSAL

FROM EMP

ORDER BY ANNUALSAL ASC;

SUB-QUERY

A *query* written *inside* another *query* is known as Sub-query.

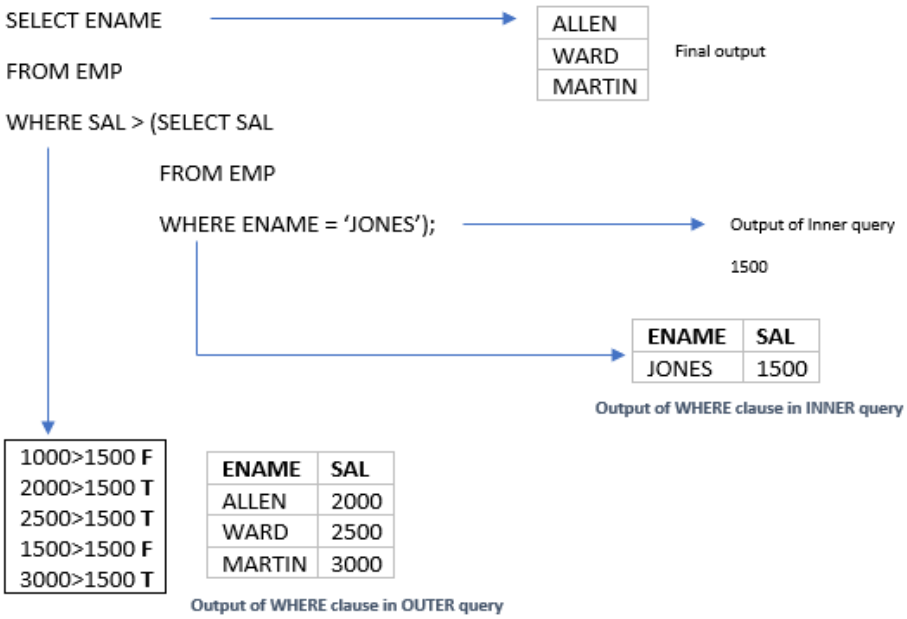
Why should we go for sub-queries?

Case 1. Whenever we have the unknown values, we go for sub-queries.

EMP

ENAME	SAL
SMITH	1000
ALLEN	2000
WARD	2500
JONES	1500
MARTIN	3000

WAQTD names of an employee earning more than 'JONES'.



Working Procedure

- Here, we will be having minimum of 2 queries.
  - Outer query
  - Inner query / Sub-query
- Inner query will *execute* first & *generate* the output.
- The *output generated* by the inner query will be *given* as input to the outer query.
- The outer query will execute & generate the output.
- This output will be the result.
- By this we can say, the outer query is *dependent* on inner query.

TYPES OF SUB-QUERY

1. Single-row Sub-query
2. Multi-row Sub-query

SINGLE-ROW SUB-QUERY

If, inner query *returns exactly* one output to the outer query, we call it as Single-row Sub-query.

MULTI-ROW SUB-QUERY

- If, inner query *returns more than* one output to the outer query, we call it as Muti-row Sub-query.
- We can achieve multi-row sub-query by *using* 'ALL' & 'ANY' operator.

Q115. WAQTD names of an employee who are earning less than 'ADAMS'.

```
SELECT ENAME
FROM EMP
WHERE SAL > (SELECT SAL
             FROM EMP
             WHERE ENAME = 'ADAMS');
```

```
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE SAL > (SELECT SAL
  4               FROM EMP
  5                            WHERE ENAME = 'ADAMS');

ENAME
----------
ALLEN
WARD
JONES
MARTIN
BLAKE
CLARK
SCOTT
KING
TURNER
FORD
MILLER
```

Q116. WAQTD details of an employee who are working in same department as that of 'KING'.

```
SELECT *
FROM EMP
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
                 WHERE ENAME = 'KING');
```

Q117. WAQTD details of an employee who are working in same designation as that of 'SCOTT'.

```
SELECT *
FROM EMP
WHERE JOB IN (SELECT JOB
              FROM EMP
              WHERE JOB = 'SCOTT');
```

Q118. WAQTD name & hire date of an employee hired after 'FORD.'

```
SELECT ENAME, HIREDATE
FROM EMP
WHERE HIREDATE > (SELECT HIREDATE
                  FROM EMP
                  WHERE ENAME = 'FORD');
```

Q119. WAQTD details of an employee earning more than 'JONES' & less than 'KING'.

```
SELECT *
FROM EMP
WHERE SAL > (SELECT SAL
             FROM EMP
             WHERE ENAME = 'JONES') AND
      SAL < (SELECT SAL
             FROM EMP
             WHERE ENAME = 'KING');
```

Q120. WAQTD names of an employee who are working as a 'MANAGER' & earning more than 'CLARK'.

```
SELECT ENAME
FROM EMP
WHERE JOB = 'MANAGER' AND
      SAL > (SELECT SAL
             FROM EMP
             WHERE ENAME = 'CLARK');
```

Q121. WAQTD details of an employee working as a 'SALESMAN' in same department as that of 'JONES'.

```
SELECT *
FROM EMP
WHERE JOB = 'SALESMAN' AND
      DEPTNO IN (SELECT DEPTNO
                 FROM EMP
                 WHERE ENAME = 'JONES');
```

Q122. WAQTD no. of employee working as 'PRESIDENT' in same department of that of 'KING'.

```
SELECT COUNT (*)
```

```
FROM EMP
WHERE JOB = 'PRESIDENT' AND
        DEPTNO IN (SELECT DEPTNO
                        FROM EMP
                        WHERE ENAME = 'KING');
```

Q123. WAQTD maximum salary given to the employee working as 'CLERK' in the same department as that of 'SMITH'.

```
SELECT MAX (SAL)
FROM EMP
WHERE JOB = 'CLERK' AND
        DEPTNO IN (SELECT DEPTNO
                        FROM EMP
                        WHERE ENAME = 'SMITH');
```

Q124. WAQTD name of an employee earning more than 'JONES' but less than 'KING'.

```
SELECT ENAME
FROM EMP
WHERE SAL > (SELECT SAL
                FROM EMP
                WHERE ENAME = 'JONES') AND
        SAL < (SELECT SAL
                FROM EMP
                WHERE ENAME = 'KING');
```

Q125. WAQTD no. of employees hired in the month of 'APRIL' in the same department as that of 'SCOTT'.

```
SELECT COUNT (*)
FROM EMP
WHERE HIREDATE LIKE '%APR%' AND
        DEPTNO IN (SELECT DEPTNO
                        FROM EMP
                        WHERE ENAME = 'SCOTT');
```

Q126. WAQTD name of an employee working as 'MANAGER' in same department as that of 'MARTIN'.

```
SELECT ENAME
FROM EMP
WHERE JOB = 'MANAGER' AND
        DEPTNO IN (SELECT DEPTNO
                FROM EMP
                WHERE ENAME = 'MARTIN');
```

Q127. WAQTD name of an employee hired after 'JAMES' before 'SCOTT'.

```
SELECT ENAME
FROM EMP
WHERE HIREDATE > (SELECT HIREDATE
                FROM EMP
                WHERE ENAME = 'JAMES') AND
        HIREDATE < (SELECT HIREDATE
                FROM EMP
                WHERE ENAME = 'SCOTT');
```

Q128. WAQTD name of an employee working as a 'MANAGER' in same department as that of 'KING' & earning more than 'MILLER'.

```
SELECT ENAME
FROM EMP
WHERE JOB = 'MANAGER' AND
        DEPTNO IN (SELECT DEPTNO
                FROM EMP
                WHERE ENAME = 'KING') AND
        SAL > (SELECT SAL
                FROM EMP
                WHERE ENAME = 'MILLER');
```

Q129. WAQTD no. of employees working as a 'CLERK' in same department as that of 'ADAMS' & hired in the year 1980.

```
SELECT COUNT (*)
FROM EMP
WHERE JOB = 'CLERK' AND
        DEPTNO IN (SELECT DEPTNO
```

```
                FROM EMP
                WHERE ENAME = 'ADAMS') AND
        HIREDATE LIKE '%80';
```

Q130. WAQTD ENAME of employees earning more than 'ADAMS'.

```
SELECT ENAME
FROM EMP
WHERE SAL > (SELECT SAL
                FROM EMP
                WHERE ENAME = 'ADAMS');
```

Q131. WAQTD ENAME, SAL of the employee earning less than 'KING'.

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

Q132. WAQTD ENAME, DEPT of the employees if they are working in the same department as 'JONES'.

```
SELECT ENAME, DEPTNO
FROM EMP
WHERE DEPTNO IN (SELECT DEPTNO
                FROM EMP
                WHERE ENAME = 'JONES');
```

Q133. WAQTD ENAME, JOB of all the employees working in the same designation as 'JAMES'.

```
SELECT ENAME, JOB
FROM EMP
WHERE JOB IN (SELECT JOB
                FROM EMP
                WHERE ENAME = 'JAMES');
```

Q134. WAQTD EMPNO, ENAME along with ANNUALSAL of all the employees if their ANNUALSAL is greater than 'WARD' annual salary.

```
SELECT EMPNO, ENAME, SAL*12 ANNUALSAL
FROM EMP
```

WHERE SAL*12 > (SELECT SAL*12

FROM EMP

WHERE ENAME = ‘WARD’);

```
SQL> SELECT EMPNO, ENAME, SAL*12 ANNUALSAL
  2  FROM EMP
  3  WHERE SAL*12 > (SELECT SAL*12
  4                                  FROM EMP
  5                                  WHERE ENAME = 'WARD');

     EMPNO ENAME      ANNUALSAL
---------- ---------- ----------
      7499 ALLEN           19200
      7566 JONES           35700
      7698 BLAKE           34200
      7782 CLARK           29400
      7788 SCOTT           36000
      7839 KING            60000
      7844 TURNER          18000
      7902 FORD            36000
      7934 MILLER          15600

9 rows selected.
```

Q135. WAQTD ENAME, HIREDATE of the employees if they are hired before ‘TURNER’.

SELECT ENAME, HIREDATE

FROM EMP

WHERE HIREDATE < (SELECT HIREDATE

FROM EMP

WHERE ENAME = ‘TURNER’);

Q136. WAQTD ENAME, HIREDATE of the employees if they are hired after the ‘PRESIDENT’.

SELECT ENAME, HIREDATE

FROM EMP

WHERE HIREDATE > (SELECT HIREDATE

FROM EMP

WHERE JOB = ‘PRESIDENT’);

Q137. WAQTD ENAME & SAL of the employees if they are earning SAL less than the employee whose EMPNO is 7839.

SELECT ENAME, SAL

FROM EMP

WHERE SAL < (SELECT SAL

```
        FROM EMP
        WHERE EMPNO = 7839);
```

Q138. WAQTD all the details of an employee if the employees are hired before 'MILLER'.

```
SELECT *
FROM EMP
WHERE HIREDATE < (SELECT HIREDATE
                  FROM EMP
                  WHERE ENAME = 'MILLER');
```

Q139. WAQTD ENAME & EMPNO of the employees if employees are earning more than 'ALLEN'.

```
SELECT ENAME, EMPNO
FROM EMP
WHERE SAL > (SELECT SAL
        FROM EMP
        WHERE ENAME = 'ALLEN');
```

Q140. WAQTD no. of employees hired after 'KING'.

```
SELECT COUNT (*)
FROM EMP
WHERE HIREDATE > (SELECT HIREDATE
                  FROM EMP
                  WHERE ENAME = 'KING');
```

Q141. WAQTD total salary given to the employees working in the same department as of 'WARD'.

```
SELECT SUM (SAL) TOTALSAL
FROM EMP
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM EMP
                 WHERE ENAME = 'WARD');
```

Q142. WAQTD ENAME & SAL of the employees who are earning more than 'MILLER' but less than 'ALLEN'.

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > (SELECT SAL
```

```
            FROM EMP
            WHERE ENAME = 'MILLER') AND
      SAL < (SELECT SAL
            FROM EMP
            WHERE ENAME = 'ALLEN');
```

Q143. WAQTD ENAME & SAL of the employees who are earning more than Rs.1000 but less than Rs.3000.

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > 1000 AND
      SAL < 3000;
```

Q144. WAQTD all the detail of the employees working in department 20 & working in the same designation as 'SMITH'.

```
SELECT *
FROM EMP
WHERE JOB IN (SELECT JOB
            FROM EMP
            WHERE DEPTNO = 20) AND
      JOB IN (SELECT JOB
            FROM EMP
            WHERE ENAME = 'SMITH');
```

Q145. WAQTD all the details of an employee working as 'MANAGER' in the same department as that of 'TURNER'.

```
SELECT *
FROM EMP
WHERE JOB = 'MANAGER' AND
      DEPTNO IN (SELECT DEPTNO
                  FROM EMP
                  WHERE ENAME = 'TURNER');
```

Q146. WAQTD ENAME & HIREDATE of an employees hired after 1980 & before 'KING'.

```
SELECT ENAME, HIREDATE
FROM EMP
```

```
WHERE HIREDATE >= '01-JAN-1981' AND
        HIREDATE < (SELECT HIREDATE
                    FROM EMP
                    WHERE ENAME = 'KING');
```

```
SQL> SELECT ENAME, HIREDATE
  2  FROM EMP
  3  WHERE HIREDATE >= '01-JAN-1981' AND
  4   HIREDATE < (SELECT HIREDATE
  5                                          FROM EMP
  6                                          WHERE ENAME = 'KING');

ENAME      HIREDATE
---------- ---------
ALLEN      20-FEB-81
WARD       22-FEB-81
JONES      02-APR-81
MARTIN     28-SEP-81
BLAKE      01-MAY-81
CLARK      09-JUN-81
TURNER     08-SEP-81

7 rows selected.
```

Q147. WAQTD ENAME & SAL along with annual salary for all employees whose SAL is less than 'BLAKE' or employees earning more than 3500.

```
SELECT ENAME, SAL, SAL*12 ANNUALSAL
FROM EMP
WHERE SAL < (SELECT SAL
             FROM EMP
             WHERE ENAME = 'BLAKE') OR
      SAL > 3500;
```

Q148. WAQTD all the details of employees who earn more than 'SCOTT' but less than 'KING'.

```
SELECT *
FROM EMP
WHERE SAL > (SELECT SAL
             FROM EMP
             WHERE ENAME = 'SCOTT') AND
      SAL < (SELECT SAL
             FROM EMP
             WHERE ENAME = 'KING');
```

Q149. WAQTD ENAME of the employees whose name starts with 'A' & works in the same department as 'BLAKE'.

```
SELECT ENAME
FROM EMP
WHERE ENAME LIKE 'A%' AND
        DEPTNO IN (SELECT DEPTNO
                        FROM EMP
                        WHERE ENAME = 'BLAKE');
```

Q150. WAQTD ENAME & COMM if employees earn commission & work in the same designation as 'SMITH'.

```
SELECT ENAME, COMM
FROM EMP
WHERE COMM IS NOT NULL AND
        JOB IN (SELECT JOB
                FROM EMP
                WHERE ENAME = 'SMITH');
```

Q151. WAQTD details of all the employees working as 'CLERK' in the same department as 'TURNER'.

```
SELECT *
FROM EMP
WHERE JOB = 'CLERK' AND
        DEPTNO IN (SELECT DEPTNO
                        FROM EMP
                        WHERE ENAME = 'TURNER');
```

Q152. WAQTD ENAME, SAL & JOB of the employees whose annual salary is more than 'SMITH' & less than 'KING'.

```
SELECT ENAME, SAL, JOB
FROM EMP
WHERE SAL*12 > (SELECT SAL*12
                        FROM EMP
                        WHERE ENAME = 'SMITH') AND
        SAL*12 < (SELECT SAL*12
                        FROM EMP
```

WHERE ENAME = ‘KING’);

```
SQL> SELECT ENAME, SAL, JOB
  2  FROM EMP
  3  WHERE SAL*12 > (SELECT SAL*12
  4                                  FROM EMP
  5                                  WHERE ENAME = 'SMITH') AND
  6   SAL*12 < (SELECT SAL*12
  7                                  FROM EMP
  8                                  WHERE ENAME = 'KING');

ENAME              SAL JOB
---------- ---------- ---------
ALLEN            1600 SALESMAN
WARD             1250 SALESMAN
JONES            2975 MANAGER
MARTIN           1250 SALESMAN
BLAKE            2850 MANAGER
CLARK            2450 MANAGER
SCOTT            3000 ANALYST
TURNER           1500 SALESMAN
ADAMS            1100 CLERK
JAMES             950 CLERK
FORD             3000 ANALYST
MILLER           1300 CLERK

12 rows selected.
```


Case 2. Whenever the data to be found & condition to be executed are present in different tables, we go for sub-queries.

EMP

ENAME	DEPTNO
NIKHIL	10
KAVITHA	20
PRIYA	10
ARCHANA	30
RANJITHA	20

DEPT

DEPTNO	DNAME
10	D1
20	D2
30	D3

WAQTD DNAME OF KAVITHA.

⁶SELECT DNAME → Final output- **D2**

⁴FROM DEPT

⁵WHERE DEPTNO = (³SELECT DEPTNO

¹FROM EMP

²WHERE ENAME = ‘KAVITHA’);

DEPTNO	DNAME
20	D2

ENAME	DEPTNO
KAVITHA	20

Q132. WAQTD department name of 'TURNER'.

```
SELECT DNAME
FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM EMP
                 WHERE ENAME = 'TURNER');
```

Q133. WAQTD location of an employee whose name is 'MILLER'.

```
SELECT LOC
FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM EMP
                 WHERE ENAME = 'MILLER');
```

Q134. WAQTD names of the employee working in 'NEW YORK'.

```
SELECT ENAME
FROM EMP
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
                 WHERE LOC = 'NEW YORK');
```


Q135. WAQTD number of employees working in 'RESEARCH' department.

```
SELECT COUNT (*)
FROM EMP
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
                 WHERE DNAME = 'RESEARCH');
```

Q136. WAQTD location of an employee whose name ends with character 'E'.

```
SELECT LOC
FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM EMP
                 WHERE ENAME LIKE '%E');
```

Q137. WAQTD names of an employee earning more than 'MILLER' in 'ACCOUNTING' department.

```
SELECT ENAME
FROM EMP
WHERE SAL > (SELECT SAL
            FROM EMP
            WHERE ENAME = 'MILLER') AND
      DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
                 WHERE DNAME = 'ACCOUNTING');
```

Q138. WAQTD name of an employee working as a 'MANAGER' in 'DALLAS'.

```
SELECT ENAME
FROM EMP
WHERE JOB = 'MANAGER' AND
      DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
                 WHERE LOC = 'DALLAS');
```

Q139. WAQTD name of an employee working in same designation as that of 'CLARK' in 'CHICAGO'.

```
SELECT ENAME
FROM EMP
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
                 WHERE LOC = 'CHICAGO') AND
      JOB IN (SELECT JOB
              FROM EMP
              WHERE ENAME = 'CLARK');
```

Q140. WAQTD no. of employees working in same designation as that of 'KING' in 'NEW YORK' & hired in the year 1981.

```
SELECT COUNT (*)
FROM EMP
WHERE HIREDATE LIKE '%81' AND
      DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
```

```
                WHERE LOC = 'NEW YORK') AND
        JOB IN (SELECT JOB
                FROM EMP
                WHERE ENAME = 'KING');
```

Q141. WAQTD names of an employee working as an 'ANALYST' in same department as that of 'JONES' & his location must have 2 consecutive 'L'.

```
SELECT ENAME
FROM EMP
WHERE JOB = 'ANALYST' AND
        DEPTNO IN (SELECT DEPTNO
                FROM DEPT
                WHERE LOC LIKE '%LL%') AND
        DEPTNO IN (SELECT DEPTNO
                 FROM EMP
                 WHERE ENAME = 'JONES');
```

Q142. WAQTD names of an employee who is earning the maximum salary.

```
SELECT ENAME
FROM EMP
WHERE SAL = (SELECT MAX (SAL)
            FROM EMP);
```

```
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE SAL = (SELECT MAX (SAL)
  4               FROM EMP);

ENAME
----------
KING
```

Q143. WAQTD names of an employee who is earning the minimum salary.

```
SELECT ENAME
FROM EMP
WHERE SAL = (SELECT MIN(SAL)
            FROM EMP);
```

```
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE SAL = (SELECT MIN(SAL)
  4                          FROM EMP);

ENAME
----------
SMITH
```

Q144. WAQTD name of an employee who was hired first.

```
SELECT ENAME
FROM EMP
WHERE HIREDATE = (SELECT MIN(HIREDATE)
                  FROM EMP);
```

NESTED SUB-QUERY

- A sub-query *written* inside *another* sub-query is known as Nested sub-query.
- We can nest up to ***255*** sub-queries.

Q145. WAQTD names of an employee who is earning second maximum salary.

```
SELECT ENAME
FROM EMP
WHERE SAL = (SELECT MAX (SAL)
             FROM EMP
             WHERE SAL < (SELECT MAX (SAL)
                          FROM EMP));
```

```
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE SAL = (SELECT MAX (SAL)
  4               FROM EMP
  5                          WHERE SAL < (SELECT MAX (SAL)
  6                          FROM EMP));

ENAME
----------
SCOTT
FORD
```

Q146. WAQTD name of an employee earning third minimum salary.

```
SELECT ENAME
```

```
FROM EMP
WHERE SAL = (SELECT MIN (SAL)
             FROM EMP
             WHERE SAL > (SELECT MIN (SAL)
                          FROM EMP
                          WHERE SAL > (SELECT MIN (SAL)
                                       FROM EMP)));
```

```
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE SAL = (SELECT MIN (SAL)
  4               FROM EMP
  5               WHERE SAL > (SELECT MIN (SAL)
  6               FROM EMP
  7                WHERE SAL > (SELECT MIN (SAL)
  8                FROM EMP)));

ENAME
----------
ADAMS
```


Q147. WAQTD location of an employee earning second minimum salary.

```
SELECT LOC
FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM EMP
                 WHERE SAL IN (SELECT MIN (SAL)
                               FROM EMP
                               WHERE SAL > (SELECT MIN (SAL)
                                            FROM EMP)));
```

```
SQL> SELECT LOC
  2  FROM DEPT
  3  WHERE DEPTNO IN (SELECT DEPTNO
  4             FROM EMP
  5             WHERE SAL IN (SELECT MIN (SAL)
  6               FROM EMP
  7               WHERE SAL > (SELECT MIN (SAL)
  8               FROM EMP)));

LOC
-------------
CHICAGO
```

Q148. WAQTD name of an employee earning fourth minimum salary.

```
SELECT ENAME
FROM EMP
WHERE SAL = (SELECT MIN (SAL)
            FROM EMP
            WHERE SAL > (SELECT MIN (SAL)
                        FROM EMP
                        WHERE SAL > (SELECT MIN (SAL)
                                    FROM EMP
                                    WHERE SAL > (SELECT MIN (SAL)
                                                FROM EMP))));
```

Q149. WAQTD name of an employee earning third maximum salary.

```
SELECT ENAME
FROM EMP
WHERE SAL = (SELECT MAX (SAL)
            FROM EMP
            WHERE SAL < (SELECT MAX (SAL)
                        FROM EMP
                        WHERE SAL < (SELECT MAX (SAL)
                                    FROM EMP)));
```

Q150. WAQTD details of an employee who are earning more than all the 'SALESMAN'.

```
SELECT *
FROM EMP
WHERE SAL = ALL (SELECT MAX (SAL)
                FROM EMP
                WHERE JOB = 'SALESMAN');
```

```
SQL> SELECT *
  2  FROM EMP
  3  WHERE SAL > ALL(SELECT SAL
  4                  FROM EMP
  5                  WHERE JOB = 'SALESMAN');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20

6 rows selected.
```

Q151. WAQTD name & salary of an employee if they are earning more than at least a 'MANAGER'.

SELECT ENAME, SAL

FROM EMP

WHERE SAL > ANY (SELECT SAL

FROM EMP

WHERE JOB = 'MANAGER');

Q152. WAQTD names of an employee hired after all the 'MANAGER' & earning more than all the 'CLERK'.

SELECT ENAME

FROM EMP

WHERE HIREDATE = ALL (SELECT HIREDATE

FROM EMP

WHERE JOB = 'MANAGER') AND

SAL > ALL (SELECT SAL

FROM EMP

WHERE JOB = 'CLERK');

```
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE HIREDATE = ALL (SELECT HIREDATE
  4        FROM EMP
  5       WHERE JOB = 'MANAGER') AND
  6   SAL > ALL (SELECT SAL
  7                           FROM EMP
  8                           WHERE JOB = 'CLERK');

no rows selected
```

Q153. WAQTD DNAME of the employees whose name is 'SMITH'.

```
SELECT DNAME
FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM EMP
                 WHERE ENAME='SMITH');
```

Q154. WAQTD DNAME & LOC of the employee whose ENAME is 'KING'.

```
SELECT DNAME, LOC
FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM EMP
                 WHERE ENAME='KING');
```

Q155. WAQTD LOC of the employee whose employee number is 7902.

```
SELECT LOC
FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM EMP
                 WHERE EMPNO = 7902);
```


Q156. WAQTD DNAME & LOC along with DEPTNO of the employee whose name ends with 'R'.

```
SELECT DNAME, LOC, DEPTNO
FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM EMP
                 WHERE ENAME LIKE '%R');
```

Q157. WAQTD DNAME of the employee whose designation is 'PRESIDENT'.

```
SELECT DNAME
FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM EMP
                 WHERE JOB = 'PRESIDENT');
```

Q158. WAQTD names of the employee working in a 'ACCOUNTING' department.

```
SELECT ENAME
```

```
FROM EMP
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
                WHERE DNAME = 'ACCOUNTING');
```

Q159. WAQTD ENAME & salaries of the employee who are working in the location 'CHICAGO'.

```
SELECT ENAME, SAL
FROM EMP
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
                 WHERE LOC = 'CHICAGO');
```

Q160. WAQTD details of the employee working in 'SALES'.

```
SELECT *
FROM EMP
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
                 WHERE DNAME = 'SALES');
```

Q161. WAQTD details of the employee along with annual salary if employees are working in 'NEW YORK'.

```
SELECT EMP.*, SAL*12 ANNUALSAL
FROM EMP
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
                 WHERE LOC = 'NEW YORK');
```

Q162. WAQTD names of employee working in 'OPERATIONS' department.

```
SELECT ENAME
FROM EMP
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
                 WHERE DNAME = 'OPERATIONS');
```

Q163. WAQTD names of the employees earning more than 'SCOTT' in 'ACCOUNTING' department.

```
SELECT ENAME
```

```
FROM EMP
WHERE SAL > (SELECT SAL
             FROM EMP
             WHERE ENAME = 'SCOTT') AND
      DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
                 WHERE DNAME = 'ACCOUNTING');
```

Q164. WAQTD details of the employees working as the 'MANAGER' in the location 'CHICAGO'.

```
SELECT *
FROM EMP
WHERE JOB = 'MANAGER' AND
      DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
                 WHERE LOC = 'CHICAGO');
```

Q165. WAQTD ENAME & SAL of the employees earning more than 'KING' in the department 'ACCOUNTING'.

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > (SELECT SAL
             FROM EMP
             WHERE ENAME = 'KING') AND
      DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
                 WHERE DNAME = 'ACCOUNTING');
```

Q166. WAQTD details of the employees working as 'SALESMAN' in the department 'SALES'.

```
SELECT *
FROM EMP
WHERE JOB = 'SALESMAN' AND
      DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
                 WHERE DNAME = 'SALES');
```

Q167. WAQTD ENAME, SAL, JOB, HIREDATE of the employees working in 'OPERATIONS' department & hired before 'KING'.

```
SELECT ENAME, SAL, JOB, HIREDATE
FROM EMP
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
                 WHERE DNAME = 'OPERATIONS') AND
      HIREDATE < (SELECT HIREDATE
                  FROM EMP
                  WHERE ENAME = 'KING');
```

Q168. WAQTD display all the details of an employee whose department name ending 'S'.

```
SELECT *
FROM EMP
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM DEPT
                 WHERE DNAME LIKE '%S');
```

Q169. WAQTD DNAME of the employees whose name has character 'A' in it.

```
SELECT DNAME
FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM EMP
                 WHERE ENAME LIKE '%A%');
```

Q170. WAQTD DNAME & LOC of the employees whose salary is Rs.800.

```
SELECT DNAME, LOC
FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM EMP
                 WHERE SAL = 800);
```

Q171. WAQTD DNAME of the employees who earn commission.

```
SELECT DNAME
FROM DEPT
```

```
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM EMP
                 WHERE COMM IS NOT NULL);
```

Q172. WAQTD LOC of the employees if they earn commission in department 40.

```
SELECT LOC
FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM EMP
                 WHERE COMM IS NOT NULL) AND
      DEPTNO = 40;
```

Q173. WAQTD details of the employees hired after all the 'CLERKS'.

```
SELECT *
FROM EMP
WHERE HIREDATE > ALL (SELECT HIREDATE
                      FROM EMP
                      WHERE JOB = 'CLERK');
```

Q174. WAQTD ENAME & HIREDATE of employees hired before all the 'MANAGER'S.

```
SELECT ENAME, HIREDATE
FROM EMP
WHERE HIREDATE < ALL (SELECT HIREDATE
                      FROM EMP
                      WHERE JOB = 'MANAGER');
```

Q175. WAQTD details of the employees working as 'CLERK' & hired before at least a 'SALESMAN'.

```
SELECT *
FROM EMP
WHERE JOB = 'CLERK' AND
      HIREDATE < ANY (SELECT HIREDATE
                      FROM EMP
                      WHERE JOB = 'SALESMAN');
```

****EMPLOYEE-MANAGER RELATIONSHIP BASED QUERY**

Q176. WAQTD 'SMITH' reporting manager's name.

```
SELECT ENAME
FROM EMP
WHERE EMPNO IN (SELECT MGR
                FROM EMP
                WHERE ENAME = 'SMITH');
```

```
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE EMPNO IN (SELECT MGR
  4                  FROM EMP
  5          WHERE ENAME = 'SMITH');

ENAME
----------
FORD
```

Q177. WAQTD name of 'ALLEN' managers.

```
SELECT ENAME
FROM EMP
WHERE EMPNO IN (SELECT MGR
                FROM EMP
                WHERE ENAME = 'ALLEN');
```

Q178. WAQTD SAL of 'ADAMS' manager.

```
SELECT SAL
FROM EMP
WHERE EMPNO IN (SELECT MGR
                FROM EMP
                WHERE ENAME = 'ADAMS');
```

Q179. WAQTD name of 'ADAMS' manager's manager.

```
SELECT ENAME
FROM EMP
WHERE EMPNO IN (SELECT MGR
                FROM EMP
                WHERE EMPNO IN (SELECT MGR
                                FROM EMP
```

WHERE ENAME = ‘ADAMS’));

```
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE EMPNO IN (SELECT MGR
  4                  FROM EMP
  5        WHERE EMPNO IN (SELECT MGR
  6                        FROM EMP
  7                        WHERE ENAME = 'ADAMS'));

ENAME
----------
JONES
```

Q180. WAQTD LOC of ‘JONES’ manager.

SELECT LOC

FROM DEPT

WHERE DEPTNO IN (SELECT DEPTNO

FROM EMP

WHERE EMPNO IN (SELECT MGR

FROM EMP

WHERE ENAME = ‘JONES’));

Q181. WAQTD DNAME of ‘SMITH’ manager’s manager.

SELECT DNAME

FROM DEPT

WHERE DEPTNO IN (SELECT DEPTNO

FROM EMP

WHERE EMPNO IN (SELECT MGR

FROM EMP

WHERE EMPNO IN (SELECT MGR

FROM EMP

WHERE ENAME = ‘SMITH’)));

```
SQL> SELECT DNAME
  2  FROM DEPT
  3  WHERE DEPTNO IN (SELECT DEPTNO
  4                   FROM EMP
  5        WHERE EMPNO IN (SELECT MGR
  6                                    FROM EMP
  7                                    WHERE EMPNO IN (SELECT MGR
  8                                                    FROM EMP
  9                                                    WHERE ENAME = 'SMI
TH')));

DNAME
--------------
RESEARCH
```

Q182. WAQTD names of the employees reporting to 'BLAKE'.

SELECT ENAME

FROM EMP

WHERE MGR IN (SELECT EMPNO

FROM EMP

WHERE ENAME = 'BLAKE');

Q183. WAQTD number of employees reporting to 'KING'.

SELECT COUNT (*)

FROM EMP

WHERE MGR IN (SELECT EMPNO

FROM EMP

WHERE ENAME = 'KING');

```
SQL> SELECT COUNT (*)
  2  FROM EMP
  3  WHERE MGR IN (SELECT EMPNO
  4                   FROM EMP
  5                                WHERE ENAME = 'KING');

  COUNT(*)
----------
         3
```

Q184. WAQTD details of an employee reporting to 'JONES'.

SELECT *

FROM EMP

WHERE MGR IN (SELECT EMPNO

FROM EMP

WHERE ENAME = 'JONES');

```
SQL> SELECT *
  2  FROM EMP
  3  WHERE MGR IN (SELECT EMPNO
  4                FROM EMP
  5                               WHERE ENAME = 'JONES');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
```

Q185. WAQTD names of an employee reporting to 'BLAKE' manager.

```
SELECT ENAME
FROM EMP
WHERE MGR IN (SELECT MGR
              FROM EMP
              WHERE ENAME = 'BLAKE');
```

Q186. WAQTD names of an employee who are earning more than 'ADAMS' manager.

```
SELECT ENAME
FROM EMP
WHERE SAL > (SELECT SAL
             FROM EMP
             WHERE EMPNO IN (SELECT MGR
                             FROM EMP
                             WHERE ENAME = 'ADAMS'));
```

Q187. WAQTD DNAME of 'ADAMS' manager's manager.

```
SELECT DNAME
FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM EMP
                 WHERE EMPNO IN (SELECT MGR
                                 FROM EMP
                                 WHERE EMPNO IN (SELECT MGR
                                                 FROM EMP
                                                 WHERE ENAME = 'ADAMS')));
```

Q188. WAQTD number of employees reporting to 'FORD' manager.

```
SELECT COUNT (*)
FROM EMP
WHERE MGR IN (SELECT MGR
              FROM EMP
              WHERE ENAME = 'FORD');
```

JOINS

It is *used* to retrieve the data from multiple tables *simultaneously*.

TYPES OF JOINS

1. Cartesian Join or, Cross Joins
2. Inner Join or, Equijoin
3. Outer Join
   a. Left outer join
   b. Right outer join
   c. Full outer join
4. Self-Join
5. Natural Join

CARTESIAN JOIN

- In cartesian join, the *records* from table 1 will *merge* with all the *records* of table 2.
- Number of Columns in resultant table will be the summation of Column 1 & Column 2.
- Number of records in resultant table will be the product of records present in Table 1 & Table 2.
- In cartesian join, we will be *getting* error records.

WAQTD BNAME & GNAME.

```
SELECT BNAME, GNAME
FROM BOY, GIRL;
```

BOY

M=3, P=3

BID	BNAME	GID
101	ROMEO	201
102	VIRAT	202
103	RANBIR	203

GIRL

N=2, Q=3

GID	GNAME
201	JULIET
202	ANUSKHA
203	ALIA

M+N: 3+2 = 5

P*Q: 3*3 = 9

OUTPUT OF FROM CLAUSE

BID	BNAME	GID	GID	GNAME
	BOY		GIRL	
101	ROMEO	201	201	JULIET
101	ROMEO	201	202	AUSKHA
101	ROMEO	201	203	ALIA
102	VIRAT	202	201	JULIET
102	VIRAT	202	202	ANUSKHA
102	VIRAT	202	203	ALIA
103	RANBIR	203	201	JULIET
103	RANBIR	203	202	ANUSKHA
103	RANBIR	203	203	ALIA

FINAL OUTPUT

BNAME	GNAME
ROMEO	JULIET
ROMEO	ANUSKHA
ROMEO	ALIA
VIRAT	JULIET
VIRAT	ANUSKHA
VIRAT	ALIA
RANBIR	JULIET
RANBIR	ANUSKHA
RANBIR	ALIA

Syntax:

1. **ANSI**: SELECT COLUMN_NAME
   FROM TABLE_NAME1 CROSS JOIN TABLE_NAME2;

2. **ORACLE**: SELECT COLUMN_NAME
   FROM TABLE_NAME1, TABLE_NAME2;

INNER JOIN

- We *use* inner joins to *obtain* the matched records or, the records which are in pair.
- We *use* join condition to *obtain* the matched records.
  Join Condition: It is a condition in which we ***merge two tables*** to get the ***matched records***.

Example: **EMP.DEPTNO = DEPT.DEPTNO**

WAQTD BNAME & GNAME.

```
SELECT ENAME, GNAME
FROM BOY, GIRL
WHERE BOY.GID = GIRL.GID;
```

JOIN CONDITION

BID	BNAME	GID	GID	GNAME
101	ROMEO	201	201	JULIET
101	ROMEO	201	202	AUSKHA
101	ROMEO	201	203	ALIA
102	VIRAT	202	201	JULIET
102	VIRAT	202	202	ANUSKHA
102	VIRAT	202	203	ALIA
103	RANBIR	203	201	JULIET
103	RANBIR	203	202	ANUSKHA
103	RANBIR	203	203	ALIA

OUTPUT OF FROM CLAUSE

BID	BNAME	GID	GID	GNAME
101	ROMEO	201	201	JULIET
102	VIRAT	202	202	AUSKHA
103	RANBIR	203	203	ALIA

OUTPUT OF WHERE CLAUSE

Syntax:

1. **ANSI**: SELECT COLUMN_NAME
   FROM TABLE_NAME1 INNER JOIN TABLE_NAME2;
   ON <JOIN_CONDITION>;
   Example: SELECT *
   FROM EMP INNER JOIN DEPT
   ON EMP.DEPTNO = DEPT.DEPTNO;

2. **ORACLE**: SELECT COLUMN_NAME
   FROM TABLE_NAME1, TABLE_NAME2;
   WHERE <JOIN_CONDITION>;
   Example: SELECT *
   FROM EMP, DEPT
   WHERE EMP.DEPTNO = DEPT.DEPTNO;

QUESTIONS BASED ON JOINS & SUB-QUERY

Q189. WAQTD ENAME, DNAME of all the employees.

```
SELECT ENAME, DNAME
FROM EMP, DEPT
WHERE EMP.DEPTNO = DEPT.DEPTNO;
```

```
SQL> SELECT ENAME, DNAME
  2  FROM EMP, DEPT
  3  WHERE EMP.DEPTNO = DEPT.DEPTNO;

ENAME      DNAME
---------- --------------
SMITH      RESEARCH
ALLEN      SALES
WARD       SALES
JONES      RESEARCH
MARTIN     SALES
BLAKE      SALES
CLARK      ACCOUNTING
SCOTT      RESEARCH
KING       ACCOUNTING
TURNER     SALES
ADAMS      RESEARCH
JAMES      SALES
FORD       RESEARCH
MILLER     ACCOUNTING

14 rows selected.
```

Q190. WAQTD ENAME, SAL & LOC of all the employees.

```
SELECT ENAME, SAL, LOC
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO;
```

Q191. WAQTD ENAME, DEPTNO & DNAME of all the employees.

```
SELECT ENAME, D.DEPTNO, DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO;
```

Q192. WAQTD ENAME, DEPTNO & DNAME of employees working in DEPTNO 20.

```
SELECT ENAME, E.DEPTNO, DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
      DEPTNO = 20;
```

Q193. WAQTD ENAME, DNAME of employees who are earning less than Rs.2000.

```
SELECT ENAME, DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
      SAL < 2000;
```

Q194. WAQTD ENAME, LOC of employees working in 'DALLAS'.

```
SELECT ENAME, LOC
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        LOC = 'DALLAS';
```

Q195. WAQTD ENAME, SAL & DNAME of employees whose name starts with character 'A' & DNAME ends with character 'S'.

```
SELECT ENAME, SAL, DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO
        ENAME LIKE 'A%' AND
        DNAME LIKE '%S';
```

Q196. WAQTD ENAME, SAL, DNAME of all the employees who are earning more than 'SCOTT' in 'ACCOUNTING' department.

```
SELECT ENAME, SAL, DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        SAL > (SELECT SAL
                FROM EMP
                WHERE ENAME = 'SCOTT') AND
        DEPT = 'ACCOUNTING';
```

Q197. WAQTD number of employees hired before 'ALLEN' in 'RESEARCH' department using joins.

```
SELECT COUNT (*)
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        HIREDATE <= (SELECT HIREDATE
                        FROM EMP
                        WHERE ENAME = 'ALLEN') AND
        DEPT = 'RESEARCH';
```

Q198. WAQTD maximum SAL given to the employees working in same designation as that of 'BLAKE' in 'DALLAS'.

```
SELECT MAX (SAL)
```

```
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        JOB IN (SELECT JOB
                FROM EMP
                WHERE ENAME = 'BLAKE') AND
        LOC = 'DALLAS';
```

Q199. WAQTD ENAME, HIREDATE & DNAME of employees hired in the month of 'FEB' & his DNAME must have second character as 'A'.

```
SELECT ENAME, HIREDATE, DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        HIREDATE LIKE '%FEB%' AND
        DNAME LIKE '_A%';
```

Q200. WAQTD ENAME earning more than 'MILLER' in 'NEW YORK' using both Sub-query & joins.

Sub-Query Method:

```
SELECT ENAME
FROM EMP
WHERE SAL > (SELECT SAL
                FROM EMP
                WHERE ENAME = 'MILLER') AND
        DEPTNO IN (SELECT DEPTNO
                        FROM DEPT
                        WHERE LOC = 'NEW YORK');
```

Join Method:

```
SELECT ENAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        SAL > (SELECT SAL
                FROM EMP
                WHERE ENAME = 'MILLER') AND
        LOC = 'NEW YORK';
```

Q201. WAQTD ENAME & DNAME of employee who is having exactly 4 characters in his name & his DNAME should have 2 consecutive 'CC'.

```
SELECT ENAME, DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        ENAME LIKE '_ _ _ _'
        DNAME LIKE '%CC%';
```

Q202. WAQTD DNAME, ENAME & LOC of employees hired before 1981 in 'DALLAS'.

```
SELECT DNAME, ENAME, LOC
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        HIREDATE <= '31-JAN-1980' AND
        LOC = 'DALLAS';
```

Q203. WAQTD ENAME, JOB, LOC of employees working in same designation as that of 'JONES' in 'CHICAGO'.

```
SELECT ENAME, JOB, LOC
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        JOB IN (SELECT JOB
                FROM EMP
                WHERE ENAME = 'JONES') AND
        LOC = 'CHICAGO';
```

Q204. WAQTD ENAME, LOC of employees who is searching same SAL as that of 'SCOTT' in 'DALLAS' & he was hired in the month of 'DEC'.

```
SELECT ENAME, LOC
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        SAL = (SELECT SAL
                FROM EMP
                WHERE ENAME = 'SCOTT') AND
        LOC = 'DALLAS' AND
        HIREDATE LIKE '%DEC%';
```

Q205. WAQTD ENAME, DNAME of employees whose designation ends with string 'MAN' & his name must have 2 consecutive 'LL'.

```
SELECT ENAME, DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        JOB LIKE '%MAN' AND
        ENAME LIKE '%LL%';
```

Q206. WAQTD ENAME, DEPTNO & DNAME of employees working in same designation as that of 'SMITH' in 'NEW YORK'.

```
SELECT ENAME, D.DEPTNO, DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        JOB IN (SELECT JOB
                FROM EMP
                WHERE ENAME = 'SMITH') AND
        LOC = 'NEW YORK';
```

Q207. WAQTD ENAME & DNAME of employees hired after 1980 into 'RESEARCH' department & working as an 'ANALYST'.

```
SELECT ENAME, DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        HIREDATE >= '01-JAN-1981' AND
        DEPT = 'RESEARCH' AND
        JOB = 'ANALYST';
```

Q208. WAQTD ENAME, SAL, LOC of employees whose SAL ends with 50 in 'CHICAGO' & his name must start with 'M' & ends with 'N'.

```
SELECT ENAME, SAL, LOC
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        SAL LIKE '%50' AND
        LOC = 'CHICAGO' AND
        ENAME LIKE 'M%' AND ENAME LIKE '%N';
```

Q209. WAQTD ENAME, DNAME of employees earning less than 'JAMES' & must be hired in the year 1980 in 'DALLAS'.

```
SELECT ENAME, DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        SAL < (SELECT SAL
                FROM EMP
                WHERE ENAME = 'JAMES') AND
                HIREDATE LIKE '%80' AND
        LOC = 'DALLAS';
```

Q210. WAQTD ENAME & DEPTNO & DNAME of employees working as 'SALESMAN' or 'MANAGER' in DEPTNO 20 or 30 & he must be earning more than 'WARD' in 'SALES' or 'RESEARCH' department & he must get commission.

```
SELECT ENAME, D.DEPTNO, DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        JOB IN ('SALESMAN', 'MANAGER') AND
        E.DEPTNO IN (20,30) AND
        SAL > (SELECT SAL
                FROM EMP
                WHERE ENAME = 'WARD') AND
        DNAME IN ('SALES', 'RESEARCH') AND
        COMM IS NOT NULL;
```

Q211. WAQTD ENAME of the employees & their LOC of all the employees.

```
SELECT ENAME, LOC
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO;
```

Q212. WAQTD DNAME & SAL for all the employee working in 'ACCOUNTING' department.

```
SELECT DNAME, SAL
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        DNAME = 'ACCOUNTING';
```

Q213. WAQTD DNAME & ANNUALSAL for all employees whose SAL is more than Rs.2340.

```
SELECT DNAME, SAL*12 ANNUALSAL
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        SAL > 2340;
```

Q214. WAQTD ENAME & DNAME for employees having character 'A' in their DNAME.

```
SELECT ENAME, DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        DNAME LIKE '%A%';
```

Q215. WAQTD ENAME & DNAME for all the employees working as 'SALESMAN'.

```
SELECT ENAME, DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        JOB IN ('SALESMAN');
```

Q216. WAQTD DNAME & JOB for all the employees whose JOB & DNAME starts with character 'S'.

```
SELECT DNAME, JOB
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        JOB LIKE 'S%' AND
        DNAME LIKE 'S%';
```

Q217. WAQTD DNAME & MGR for employees reporting to 7839.

```
SELECT DNAME, MGR
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        MGR = 7839;
```

Q218. WAQTD DNAME & HIREDATE for employees hired after 83 into 'ACCOUNTING' or 'RESEARCH' department.

```
SELECT DNAME, HIREDATE
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
```

```
        HIREDATE >= '01-JAN-1984' AND
        DNAME IN ('ACCOUNTING', 'RESEARCH');
```

Q219. WAQTD ENAME & DNAME of the employees who are getting COMM in DEPTNO 10 or 30.

```
SELECT ENAME, DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        COMM IS NOT NULL AND
        E.DEPTNO IN (10, 30);
```

Q220. WAQTD DNAME & EMPNO for all the employees whose EMPNO are (7839, 7902) & are working in LOC = 'NEW YORK'.

```
SELECT DNAME, EMPNO
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        EMPNO IN (7839, 7902) AND
        LOC = 'NEW YORK';
```

Q221. WAQTD ENAME & DNAME who are earning more than 'SMITH'.

```
SELECT ENAME, DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
        SAL > (SELECT SAL
                FROM EMP
                WHERE ENAME = 'SMITH');
```

SELF-JOIN

It is *used* to join the same two tables or, the table itself.

Why do we use self-join?

If the data to be selected & condition to be executed are *present* in same table *but* different records we go for self-join.

WAQTD employee name along with his manager's name.

```
SELECT E1.ENAME, E2.ENAME
FROM EMP E1, EMP E2
WHERE E1.MGR = E2.EMPNO;
```

EMP E1

EMPNO	ENAME	MGR
1	SMITH	2
2	ALLEN	3
3	WARD	5
4	JAMES	5
5	KING	

EMP E2

EMPNO	ENAME	MGR
1	SMITH	2
2	ALLEN	3
3	WARD	5
4	JAMES	5
5	KING	

OUTPUT OF WHERE CLAUSE

E1			E2		
EMPNO	ENAME	MGR	EMPNO	ENAME	MGR
1	SMITH	2	2	ALLEN	3
2	ALLEN	3	3	WARD	5
3	WARD	5	5	KING	
4	JAMES	5	5	KING	

FINAL OUTPUT

ENAME	ENAME
SMITH	ALLEN
ALLEN	WARD
WARD	KING
JAMES	KING

Syntax:

1. **ANSI**: SELECT COLUMN_NAME
   FROM TABLE_NAME T1 JOIN TABLE_NAME T2;
   ON <JOIN_CONDITION>;
   Example: SELECT *
   FROM EMP E1 JOIN EMP E2
   ON E1.MGR = E2.EMPNO;

2. **ORACLE**: SELECT COLUMN_NAME
   FROM TABLE_NAME T1, TABLE_NAME T2;
   WHERE <JOIN_CONDITION>;
   Example: SELECT *
   FROM EMP E1, EMP E2
   WHERE E1.MGR = E2.EMPNO;

QUESTIONS BASED ON SELF-JOIN

Q222. WAQTD employees SAL & managers SAL.

```
SELECT E1.SAL EMP_SAL, E2.SAL MGR_SAL
FROM EMP E1, EMP E2
WHERE E1.MGR = E2.EMPNO;
```

```
SQL> SELECT E1.SAL EMP_SAL, E2.SAL MGR_SAL
  2  FROM EMP E1, EMP E2
  3  WHERE E1.MGR = E2.EMPNO;

   EMP_SAL    MGR_SAL
---------- ----------
       800       3000
      1600       2850
      1250       2850
      2975       5000
      1250       2850
      2850       5000
      2450       5000
      3000       2975
      1500       2850
      1100       3000
       950       2850
      3000       2975
      1300       2450

13 rows selected.
```

Q223. WAQTD employees name & managers name of if employee is working in DEPTNO 20.

SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME

FROM EMP E1, EMP E2

WHERE E1.MGR = E2.EMPNO AND

E1.DEPTNO = 20;

Q224. WAQTD employee name, manager's name if manager is working as 'PRESIDENT'.

SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME

FROM EMP E1, EMP E2

WHERE E1.MGR = E2.EMPNO AND

E2.JOB = 'PRESIDENT';

Q225. WAQTD employee name, employee SAL, manager name, manager SAL if employee is earning less than 1000.

SELECT E1.ENAME EMP_NAME, E1.SAL EMP_SAL, E2.ENAME MGR_NAME, E2.SAL MGR_SAL

FROM EMP E1, EMP E2

WHERE E1.MGR = E2.EMPNO AND

E1.SAL < 1000;

Q226. WAQTD employee name, employee HIREDATE, manager's name, manager's HIREDATE if employees is hired after 1980 & manager hired before 1987.

SELECT E1.ENAME EMP_NAME, E1.HIREDATE EMP_HIRED, E2.ENAME MGR_NAME, E2.HIREDATE MGR_HIRED

```
FROM EMP E1, EMP E2
WHERE E1.MGR = E2.EMPNO AND
        E1.HIREDATE >= '01-JAN-1981' AND
        E2.HIREDATE <= '31-DEC-1986';
```

Q227. WAQTD name of the employee & his manager's name if employee is working as 'CLERK'.

```
SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME
FROM EMP E1, EMP E2
WHERE E1.MGR = E2.EMPNO AND
        E1.JOB = 'CLERK';
```

Q228. WAQTD ENAME, MGR designation if manager is working in DEPTNO 10 or 20.

```
SELECT E1.ENAME EMP_NAME, E2.JOB MGR_JOB
FROM EMP E1, EMP E2
WHERE E1.MGR = E2.EMPNO AND
        E2.DEPTNO IN (10,20);
```

Q229. WAQTD employee name & manager name if employee is hired before 1982.

```
SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME
FROM EMP E1, EMP E2
WHERE E1.MGR = E2.EMPNO AND
        E1.HIREDATE <= '31-DEC-1981';
```

Q230. WAQTD employee name, manager's name if employee & manager both earn more than 2300.

```
SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME
FROM EMP E1, EMP E2
WHERE E1.MGR = E2.EMPNO AND
        E1.SAL > 2300 AND E2.SAL > 2300;
```

Q231. WAQTD employee name, employee SAL, manager name, manager SAL if employee is earning more than his manager.

```
SELECT E1.ENAME EMP_NAME, E1.SAL EMP_SAL, E2.ENAME MGR_NAME, E2.SAL MGR_SAL
FROM EMP E1, EMP E2
WHERE E1.MGR = E2.EMPNO AND
        E1.SAL > E2.SAL;
```

```
SQL> SELECT E1.ENAME EMP_NAME, E1.SAL EMP_SAL, E2.ENAME MGR_NAME, E2.SAL MGR_SAL
  2  FROM EMP E1, EMP E2
  3  WHERE E1.MGR = E2.EMPNO AND
  4   E1.SAL > E2.SAL;

EMP_NAME      EMP_SAL MGR_NAME      MGR_SAL
---------- ---------- ---------- ----------
SCOTT            3000 JONES            2975
FORD             3000 JONES            2975
```

Q232. WAQTD employee name, employee HIREDATE, manager name, manager HIREDATE if employee is hired before his manager.

```
SELECT E1.ENAME EMP_NAME, E1.SAL EMP_SAL, E2.ENAME MGR_NAME, E2.SAL MGR_SAL
FROM EMP E1, EMP E2
WHERE E1.MGR = E2.EMPNO AND
        E1.HIREDATE < E2.HIREDATE;
```

Q233. WAQTD employee name & his managers name if employee is hired in the year 1980 and manager is hired in the year 1981.

```
SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME
FROM EMP E1, EMP E2
WHERE E1.MGR = E2.EMPNO AND
        E1.HIREDATE LIKE '%80' AND
        E2.HIREDATE LIKE '%81';
```

Q234. WAQTD employee name & managers name if employee & manager both hired in the year 1987.

```
SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME
FROM EMP E1, EMP E2
WHERE E1.MGR = E2.EMPNO AND
        E1.HIREDATE LIKE '%87' AND
        E2.HIREDATE LIKE '%87';
```

Q235. WAQTD employee name & manager name if employee & manager both hired in the month of DEC.

```
SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME
FROM EMP E1, EMP E2
WHERE E1.MGR = E2.EMPNO AND
        E1.HIREDATE LIKE '%DEC%' AND
        E2.HIREDATE LIKE '%DEC%';
```

Q236. WAQTD employee name & manager name if employee is earning more than 2900 & manager is earning more than 3000.

```
SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME
FROM EMP E1, EMP E2
WHERE E1.MGR = E2.EMPNO AND
      E1.SAL > 2900 AND
      E2.SAL > 3000;
```

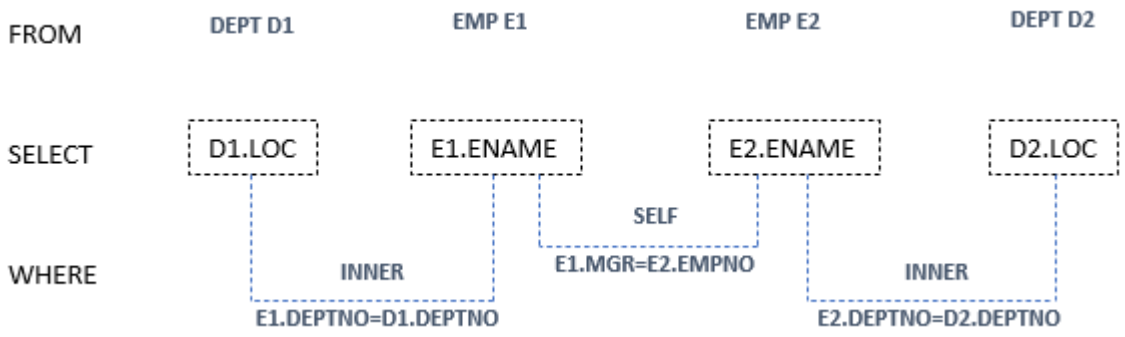
Q237. WAQTD employee name, manager name If employee is working as ANALYST & manager is working as actual manager.

```
SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME
FROM EMP E1, EMP E2
WHERE E1.MGR = E2.EMPNO AND
      E1.JOB = 'ANALYST' AND
      E2.JOB = 'MANAGER';
```

Q238. WAQTD employee name & manager's name if employee is earning less than 1000 & manager in department number 30.

```
SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME
FROM EMP E1, EMP E2
WHERE E1.MGR = E2.EMPNO AND
      E1.SAL < 1000 AND
      E2.DEPTNO = 30;
```

Q239. WAQTD employee name, employee LOC, manager's name, & manager's LOC.




```
SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME, D1.LOC EMP_LOC, D2.LOC MGR_LOC
FROM EMP E1, EMP E2, DEPT D1, DEPT D2
WHERE E1.MGR = E2.EMPNO AND
```

E1.DEPTNO = D1.DEPTNO AND

E2.DEPTNO = D2.DEPTNO;

```
SQL> SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME, D1.LOC EMP_LOC, D2.LOC MGR_LOC
  2  FROM EMP E1, EMP E2, DEPT D1, DEPT D2
  3  WHERE E1.MGR = E2.EMPNO AND
  4   E1.DEPTNO = D1.DEPTNO AND
  5  E2.DEPTNO = D2.DEPTNO;

EMP_NAME   MGR_NAME   EMP_LOC       MGR_LOC
---------- ---------- ------------- -------------
SMITH      FORD       DALLAS        DALLAS
ALLEN      BLAKE      CHICAGO       CHICAGO
WARD       BLAKE      CHICAGO       CHICAGO
JONES      KING       DALLAS        NEW YORK
MARTIN     BLAKE      CHICAGO       CHICAGO
BLAKE      KING       CHICAGO       NEW YORK
CLARK      KING       NEW YORK      NEW YORK
SCOTT      JONES      DALLAS        DALLAS
TURNER     BLAKE      CHICAGO       CHICAGO
ADAMS      SCOTT      DALLAS        DALLAS
JAMES      BLAKE      CHICAGO       CHICAGO
FORD       JONES      DALLAS        DALLAS
MILLER     CLARK      NEW YORK      NEW YORK

13 rows selected.
```

Q240. WAQTD employee name, employee LOC, manager name, manager LOC if employee is working as a 'CLERK' & manager is working in 'DALLAS'.

SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME, D1.LOC EMP_LOC, D2.LOC MGR_LOC

FROM EMP E1, EMP E2, DEPT D1, DEPT D2

WHERE E1.MGR = E2.EMPNO AND

E1.DEPTNO = D1.DEPTNO AND

E2.DEPTNO = D2.DEPTNO AND

E1.JOB='CLERK' AND D2.LOC='DALLAS';

Q241. WAQTD employee name, employee LOC, manager name, manager LOC if employee working as 'SALESMAN' in 'SALES' department & manager is working as actual manager in 'CHICAGO'.

SELECT E1.ENAME EMP_NAME, D1.LOC EMP_LOC, E2.ENAME MGR_NAME, D2.LOC MGR_LOC

FROM EMP E1, EMP E2, DEPT D1, DEPT D2

WHERE E1.MGR = E2.EMPNO AND

E1.DEPTNO = D1.DEPTNO AND

E2.DEPTNO = D2.DEPTNO AND

E1.JOB = 'SALESMAN' AND D1.DNAME = 'SALES' AND

E2.JOB = 'MANAGER' AND D2.LOC = 'CHICAGO';

Q242. WAQTD employee name, employee DNAME, manager name, manager DNAME if employee is hired in the year 81 working as 'CLERK' & manager working in 'SALES' department.

```
SELECT E1.ENAME EMP_NAME, D1.DNAME EMP_DNAME, E2.ENAME MGR_NAME, D2.DNAME
MGR_DNAME

FROM EMP E1, EMP E2, DEPT D1, DEPT D2

WHERE E1.MGR = E2.EMPNO AND

        E1.DEPTNO = D1.DEPTNO AND

        E2.DEPTNO = D2.DEPTNO AND

        E1.HIREDATE LIKE '%81' AND E1.JOB = 'CLERK' AND D2.DNAME = 'SALES';
```

Q243. WAQTD employee name, employee LOC, manager name, manager LOC if employee is working in DEPTNO 10 or 20 and hired after 1982 & manager is working as actual manager in 'RESEARCH' department.

```
SELECT E1.ENAME EMP_NAME, D1.LOC EMP_LOC, E2.ENAME MGR_NAME, D2.LOC MGR_LOC

FROM EMP E1, EMP E2, DEPT D1, DEPT D2

WHERE E1.MGR = E2.EMPNO AND

        E1.DEPTNO = D1.DEPTNO AND

        E2.DEPTNO = D2.DEPTNO AND

        E1.DEPTNO IN (10,20) AND E1.HIREDATE >= '01-JAN-1983' AND

        E2.JOB = 'MANAGER' AND D2.DNAME = 'RESEARCH';
```


Q244. WAQTD employee name, employee LOC, manager name, manager LOC if employee is hired after 'JONES' into 'SALES' department & manager is earning less than 'KING' in 'CHICAGO'.

```
SELECT E1.ENAME EMP_NAME, D1.LOC EMP_LOC, E2.ENAME MGR_NAME, D2.LOC MGR_LOC

FROM EMP E1, EMP E2, DEPT D1, DEPT D2

WHERE E1.MGR = E2.EMPNO AND

        E1.DEPTNO = D1.DEPTNO AND

        E2.DEPTNO = D2.DEPTNO AND

        E1.HIREDATE = (SELECT E1.HIREDATE

                        FROM EMP E1

                        WHERE E1.ENAME = 'JONES') AND D1.DNAME = 'SALES' AND

        E2.SAL < (SELECT E2.SAL

                FROM EMP E2

                WHERE E2.ENAME = 'KING') AND D2.LOC = 'CHICAGO';
```

Q245. WAQTD employee name, employee DNAME, manager name, manager DNAME if employee is earning more than 'ALLEN' in 'ACCOUNTING' department & manager is working as a 'PRESIDENT' in 'NEW YORK'.

```
SELECT E1.ENAME EMP_NAME, D1.DNAME EMP_DNAME, E2.ENAME MGR_NAME, D2.DNAME
MGR_DNAME

FROM EMP E1, EMP E2, DEPT D1, DEPT D2

WHERE E1.MGR = E2.EMPNO AND

        E1.DEPTNO = D1.DEPTNO AND

        E2.DEPTNO = D2.DEPTNO AND

        E1.SAL > (SELECT E1.SAL

                FROM EMP E1

                WHERE E1.NAME = 'ALLEN') AND D1.DNAME = 'ACCOUNTING' AND

        D2.JOB = 'PRESIDENT' AND D2.LOC = 'NEW YORK';
```

Q246. WAQTD employee name, manager name, along with manager's manager name.

```
SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME, E3.ENAME MGR_MGR_NAME

FROM EMP E1, EMP E2, EMP E3

WHERE E1.MGR = E2.EMPNO AND

        E2.MGR = E3.EMPNO;
```

```
SQL> SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME, E3.ENAME MGR_MGR_NAME
  2  FROM EMP E1, EMP E2, EMP E3
  3  WHERE E1.MGR = E2.EMPNO AND
  4   E2.MGR = E3.EMPNO;

EMP_NAME   MGR_NAME   MGR_MGR_NA
---------- ---------- ----------
SMITH      FORD       JONES
ALLEN      BLAKE      KING
WARD       BLAKE      KING
MARTIN     BLAKE      KING
SCOTT      JONES      KING
TURNER     BLAKE      KING
ADAMS      SCOTT      JONES
JAMES      BLAKE      KING
FORD       JONES      KING
MILLER     CLARK      KING

10 rows selected.
```

Q247. WAQTD DEPTNO & 2nd maximum SAL in each department.

```
SELECT A.DEPTNO DEPT_NAME, MAX(A.SAL) MAX_SAL

FROM EMP A, EMP B
```

```
WHERE A.DEPTNO=B.DEPTNO AND
        A.SAL<B.SAL
GROUP BY A.DEPTNO
ORDER BY A.DEPTNO;
```

```
SQL> SELECT A.DEPTNO DEPT_NAME, MAX(A.SAL) MAX_SAL
  2  FROM EMP A, EMP B
  3  WHERE A.DEPTNO=B.DEPTNO AND
  4  A.SAL<B.SAL
  5  GROUP BY A.DEPTNO
  6  ORDER BY A.DEPTNO;

 DEPT_NAME    MAX_SAL
---------- ----------
        10       2450
        20       2975
        30       1600
```

Q248. WAQTD employee name, manager name, manager's manager name along with their DNAME.

```
SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME, E3.ENAME MGR_MGR_NAME, D1.DNAME,
D2.DNAME, D3.DNAME
FROM EMP E1, EMP E2, EMP E3, DEPT D1, DEPT D2, DEPT D3
WHERE E1.MGR = E2.EMPNO AND
        E2.MGR = E3.EMPNO AND
        E1.DEPTNO = D1.DEPTNO AND
        E2.DEPTNO =D2.DEPTNO AND
        E3.DEPTNO = D3.DEPTNO;
```

```
SQL> SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME, E3.ENAME MGR_MGR_NAME, D1.DNAME, D2.DNAME, D3.DNAM
E
  2  FROM EMP E1, EMP E2, EMP E3, DEPT D1, DEPT D2, DEPT D3
  3  WHERE E1.MGR = E2.EMPNO AND
  4   E2.MGR = E3.EMPNO AND
  5   E1.DEPTNO = D1.DEPTNO AND
  6   E2.DEPTNO =D2.DEPTNO AND
  7   E3.DEPTNO = D3.DEPTNO;

EMP_NAME   MGR_NAME   MGR_MGR_NA DNAME          DNAME          DNAME
---------- ---------- ---------- -------------- -------------- --------------
SMITH      FORD       JONES      RESEARCH       RESEARCH       RESEARCH
ALLEN      BLAKE      KING       SALES          SALES          ACCOUNTING
WARD       BLAKE      KING       SALES          SALES          ACCOUNTING
MARTIN     BLAKE      KING       SALES          SALES          ACCOUNTING
SCOTT      JONES      KING       RESEARCH       RESEARCH       ACCOUNTING
TURNER     BLAKE      KING       SALES          SALES          ACCOUNTING
ADAMS      SCOTT      JONES      RESEARCH       RESEARCH       RESEARCH
JAMES      BLAKE      KING       SALES          SALES          ACCOUNTING
FORD       JONES      KING       RESEARCH       RESEARCH       ACCOUNTING
MILLER     CLARK      KING       ACCOUNTING     ACCOUNTING     ACCOUNTING

10 rows selected.
```

Q249. WAQTD employee name, managers name & manager's manager name along with their DNAME if employee is earning more than 1000 & manager earns more than 'ALLEN' & Manager's manager working in 'NEW YORK' or 'CHICAGO'.

```
SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME, E3.ENAME MGR_MGR_NAME, D1.DNAME
EMP_DNAME, D2.DNAME MGR_DNAME, D3.DNAME MGR_MGR_DNAME

FROM EMP E1, EMP E2, EMP E3, DEPT D1, DEPT D2, DEPT D3

WHERE E1.MGR = E2.EMPNO AND

        E2.MGR = E3.EMPNO AND

        E1.DEPTNO = D1.DEPTNO AND

        E2.DEPTNO =D2.DEPTNO AND

        E3.DEPTNO = D3.DEPTNO AND

        E1.SAL > 1000 AND

        E2.SAL > (SELECT E2.SAL

                FROM EMP E2

                WHERE E2.ENAME = 'ALLEN') AND

        D3.LOC IN ('NEW YORK', 'CHICAGO');
```

Q250. WAQTD employee name, managers name & manager's manager name along with their LOC if the employees hired before 'MARTIN' & manager working in 'ACCOUNTING' or 'SALES' department & manager's manager earning SAL more than 'SMITH'.

```
SELECT E1.ENAME EMP_NAME, E2.ENAME MGR_NAME, E3.ENAME MGR_MGR_NAME, D1.LOC
EMP_LOC, D2.LOC MGR_LOC, D3.LOC MGR_MGR_LOC

FROM EMP E1, EMP E2, EMP E3, DEPT D1, DEPT D2, DEPT D3

WHERE E1.MGR = E2.EMPNO AND

        E2.MGR = E3.EMPNO AND

        E1.DEPTNO = D1.DEPTNO AND

        E2.DEPTNO = D2.DEPTNO AND

        E3.DEPTNO = D3.DEPTNO AND

        E1.HIREDATE < (SELECT E1.HIREDATE

                        FROM EMP E1

                        WHERE E1.ENAME = 'MARTIN') AND

        D2.DNAME IN ('ACCOUNTING', 'SALES') AND

        E3.SAL > (SELECT E3.SAL

                FROM EMP E3
```

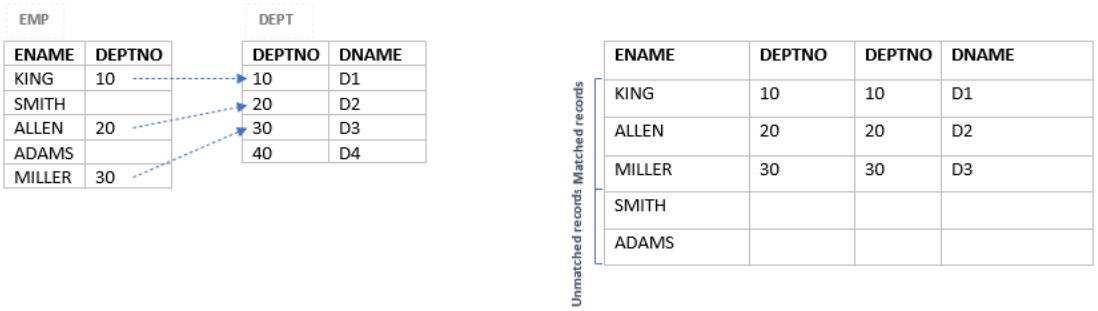
WHERE E3.ENAME = 'SMITH');

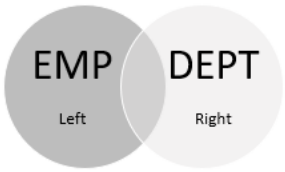
OUTER JOIN

In outer join, we *get* the matched records *along* with the unmatched records.

- **LEFT OUTER JOIN:** In left outer join, we get the unmatched records along with the matched records from left table.

  SELECT *
  FROM EMP E, DEPT D
  WHERE E.DEPTNO = D.DEPTNO (+)





Syntax:

ANSI: SELECT COLUMN_NAME
FROM TABLE_NAME1 LEFT [OUTER] JOIN TABLE_NAME2
ON <JOIN_CONDITION>;

Example: SELECT *
FROM EMP E LEFT OUTER JOIN DEPT D
ON E.DEPTNO = D.DEPTNO;

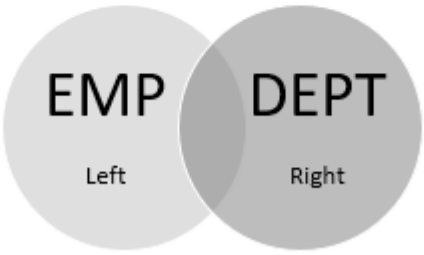
ORACLE: SELECT COLUMN_NAME
FROM TABLE_NAME1, TABLE_NAME2
WHERE TABLE_NAME1.COLUMN_NAME = TABLE_NAME2.COLUMN_NAME (+);

Example: SELECT *
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO (+);

- **RIGHT OUTER JOIN:** It is used to get the unmatched records along with the matched records from right table.

```
SELECT *
FROM EMP E, DEPT D
WHERE E.DEPTNO (+) = D.DEPTNO;
```

ENAME	DEPTNO	DEPTNO	DNAME
KING	10	10	D1
ALLEN	20	20	D2
MILLER	30	30	D3
		40	D4



Syntax:

ANSI: SELECT COLUMN_NAME
FROM TABLE_NAME1 RIGHT [OUTER]
JOIN TABLE_NAME2
ON <JOIN_CONDITION>;

Example: SELECT *
FROM EMP E RIGHT OUTER JOIN DEPT D
ON E.DEPTNO = D.DEPTNO;

ORACLE: SELECT COLUMN_NAME
FROM TABLE_NAME1, TABLE_NAME2
WHERE TABLE_NAME1.COLUMN_NAME (+) = TABLE_NAME2.COLUMN_NAME;

Example: SELECT *
FROM EMP E, DEPT D
WHERE E.DEPTNO (+) = D.DEPTNO;

- **FULL OUTER JOIN:** It is used to get the unmatched records along with the matched records from both the table.

	ENAME	DEPTNO	DEPTNO	DNAME
Matched records	KING	10	10	D1
	ALLEN	20	20	D2
	MILLER	30	30	D3
Unmatched records	NULL	NULL	40	D4
	SMITH	NULL	NULL	NULL
	ADAMS	NULL	NULL	NULL

Syntax:

ANSI: SELECT COLUMN_NAME
FROM TABLE_NAME1 FULL [OUTER] JOIN TABLE_NAME2
ON <JOIN_CONDITION>;

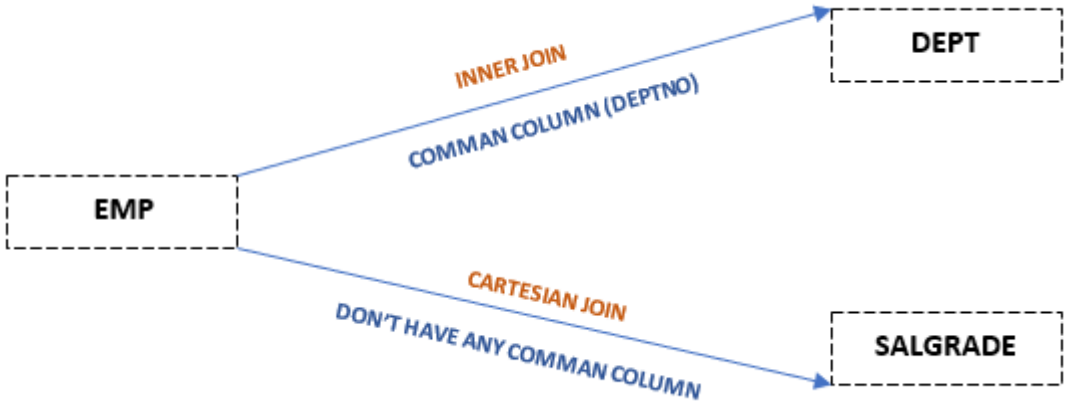
Example: SELECT *
FROM EMP E FULL OUTER JOIN DEPT D
ON E.DEPTNO = D.DEPTNO;

NATURAL JOIN

- In Natural Join, we *won't* be *writing* any join condition.
- If there are similar columns between 2 table, we will be *getting* the output of inner join.
- If there are no similar columns, we will be *getting* the output of cartesian join.

***Why do we go for Natural Join?**

Whenever we don't know the structure of a table we go for natural join.



Syntax:

ANSI: SELECT COLUMN_NAME

FROM TABLE_NAME1 NATURAL JOIN TABLE_NAME2;

Example: **1.** SELECT *

FROM EMP NATURAL JOIN DEPT;

2. SELECT *

FROM EMP NATURAL JOIN SALGRADE;

PSEUDO-COLUMN

Pseudo Column is the false columns that are *present* in each & every table & it must be *called* Explicitly.

TYPES OF PSEUDO-COLUMN

- ROWID
- ROWNUM

ROWID: ROWID is an 18-digit *address* in which *records* are present or, stored in a memory.

NOTE:

- ROWID is one of the ways to access or, delete the record.
- ROWID is unique.
- ROWID is *generated* at the time of insertion of records.
- ROWID *cannot* be inserted, updated or deleted.
- Empty table will *not* be having ROWID.
- ROWID is static in nature (constant).
- ROWID can be *used* to identify a record uniquely from the table when there is *no* key attribute or primary key.

```
SQL> SELECT ROWID, EMP.*
  2  FROM EMP;

ROWID                 EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
------------------ ---------- ---------- --------- ---------- --------- ---------- ---------- ------
AAAMFPAAEAAAAAgAAA       7369 SMITH      CLERK           7902 17-DEC-80        800                    20
AAAMFPAAEAAAAAgAAB       7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
AAAMFPAAEAAAAAgAAC       7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
AAAMFPAAEAAAAAgAAD       7566 JONES      MANAGER         7839 02-APR-81       2975                    20
AAAMFPAAEAAAAAgAAE       7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
AAAMFPAAEAAAAAgAAF       7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
AAAMFPAAEAAAAAgAAG       7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
AAAMFPAAEAAAAAgAAH       7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
AAAMFPAAEAAAAAgAAI       7839 KING       PRESIDENT            17-NOV-81       5000                    10
AAAMFPAAEAAAAAgAAJ       7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
AAAMFPAAEAAAAAgAAK       7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
AAAMFPAAEAAAAAgAAL       7900 JAMES      CLERK           7698 03-DEC-81        950                    30
AAAMFPAAEAAAAAgAAM       7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
AAAMFPAAEAAAAAgAAN       7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

14 rows selected.
```

ROWNUM

ROWNUM acts as serial number to the *result table*.

- ROWNUM is *used* as record number that is assigned to the *result table*.

- ROWNUM is dynamic in *nature* (Keeps on changing).
- ROWNUM is *generated* at the time of execution.
- ROWNUM *always starts* with 1.
- ROWNUM *cannot* be duplicated.
- ROWNUM gets incremented *after* it is assigned.
- ROWNUM *changes* because it *depends* on the result table.

Q251. WAQTD the first five records from EMP table.

```
SELECT *
FROM EMP
WHERE ROWNUM <= 5;
```

Q252. WAQTD first three records from EMP table.

```
SELECT *
FROM EMP
WHERE ROWNUM <=3;
```

Q253. WAQTD the first seven records from EMP table.

```
SELECT *
FROM EMP
WHERE ROWNUM <=7;
```

Q254. WAQTD the first record from EMP table.

```
SELECT *
FROM EMP
WHERE ROWNUM = 1;
```

Q255. WAQTD the first four records from EMP table.

```
SELECT *
FROM EMP
WHERE ROWNUM <=4;
```

Q256. WAQTD 3rd record from EMP table. (Conceptual Question)

a. To make ROWNUM as static.

```
SELECT ROWNUM, EMP.*
FROM EMP;
```

EMP

ENAME	DEPTNO
SMITH	20
KING	10
MARTIN	30
MILLER	10
SCOTT	20

ROWNUM	ENAME	DEPTNO
1	SMITH	20
2	KING	10
3	MARTIN	30
4	MILLER	10
5	SCOTT	20

b. Change the ROWNUM to any other name by using alias SLNO.

```
SELECT ROWNUM AS SLNO, EMP.*
FROM EMP;
```

SLNO	ENAME	DEPTNO
1	SMITH	20
2	KING	10
3	MARTIN	30
4	MILLER	10
5	SCOTT	20

c. Use this sub-query in FROM clause of outer query.

```
SELECT *
FROM (SELECT ROWNUM AS SLNO, EMP.*
      FROM EMP);
```

SLNO	ENAME	DEPTNO
1	SMITH	20
2	KING	10
3	MARTIN	30
4	MILLER	10
5	SCOTT	20

d. In the outer query, use the alias name in the condition.

```
SELECT *
FROM (SELECT ROWNUM AS SLNO, EMP.*
      FROM EMP)
WHERE SLNO = 3;
```

```
SQL> SELECT *
  2                       FROM (SELECT ROWNUM AS SLNO, EMP.*
  3                                    FROM EMP)
  4                       WHERE SLNO = 3;

      SLNO      EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- ---------- --------- ---------- --------- ---------- ---------- ----------
         3       7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
```

Q257. WAQTD 5th record from the EMP table.

```
SELECT *
FROM (SELECT ROWNUM SLNO, EMP.*
      FROM EMP)
WHERE SLNO = 5;
```

Q258. WAQTD ENAME, SAL from 7th record.

```
SELECT ENAME, SAL
FROM (SELECT ROWNUM SLNO, EMP.*
      FROM EMP)
WHERE SLNO = 7;
```

Q259. WAQTD 1st, 3rd, 5th, & 8th records from EMP table.

```
SELECT *
FROM (SELECT ROWNUM SLNO, EMP.*
      FROM EMP)
WHERE SLNO IN (1, 3, 5, 8);
```

```
SQL> SELECT *
  2  FROM (SELECT ROWNUM SLNO, EMP.*
  3                FROM EMP)
  4  WHERE SLNO IN (1, 3, 5, 8);

      SLNO      EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- ---------- --------- ---------- --------- ---------- ---------- ----------
         1       7369 SMITH      CLERK           7902 17-DEC-80        800                    20
         3       7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
         5       7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
         8       7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
```

Q260. WAQTD ENAME of 8th, 7th & 6th record.

```
SELECT ENAME
```

```
FROM (SELECT ROWNUM, EMP.*
        FROM EMP)
WHERE SLNO IN (8, 7, 6);
```

Q261. WAQTD top 3 records from the EMP table.

```
SELECT *
FROM EMP
WHERE ROWNUM <=3;
```

```
SQL> SELECT *
  2  FROM EMP
  3  WHERE ROWNUM <=3;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
```

Q262. WAQTD last 3 records from EMP table. (Important)

```
SELECT *
FROM (SELECT ROWNUM SLNO, EMP.*
        FROM EMP
        ORDER BY SLNO DESC)
WHERE ROWNUM<=3;
```

```
SQL> SELECT *
  2  FROM (SELECT ROWNUM SLNO, EMP.*
  3        FROM EMP
  4        ORDER BY SLNO DESC)
  5  WHERE ROWNUM<=3;

      SLNO      EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- ---------- --------- ---------- --------- ---------- ---------- ----------
        14       7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
        13       7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
        12       7900 JAMES      CLERK           7698 03-DEC-81        950                    30
```

Q263. WAQTD 3rd maximum salary. (Using ROWNUM method)

```
SQL> SELECT SAL
  2  FROM EMP;

       SAL
----------
       800
      1600
      1250
      2975
      1250
      2850
      2450
      3000
      5000
      1500
      1100
       950
      3000
      1300

14 rows selected.
```

```
SQL> SELECT SAL
  2  FROM EMP
  3  ORDER BY SAL DESC;

       SAL
----------
      5000
      3000
      3000
      2975
      2850
      2450
      1600
      1500
      1300
      1250
      1250
      1100
       950
       800

14 rows selected.
```

```
SQL> SELECT DISTINCT SAL
  2  FROM EMP
  3  ORDER BY SAL DESC;

       SAL
----------
      5000
      3000
      2975
      2850
      2450
      1600
      1500
      1300
      1250
      1100
       950
       800

12 rows selected.
```

```
SQL> SELECT ROWNUM SLNO, SAL
  2  FROM (SELECT DISTINCT SAL
  3        FROM EMP
  4        ORDER BY SAL DESC);

      SLNO        SAL
---------- ----------
         1       5000
         2       3000
         3       2975
         4       2850
         5       2450
         6       1600
         7       1500
         8       1300
         9       1250
        10       1100
        11        950
        12        800

12 rows selected.
```

```
SQL> SELECT SAL
  2  FROM (SELECT ROWNUM SLNO, SAL
  3        FROM (SELECT DISTINCT SAL
  4              FROM EMP
  5              ORDER BY SAL DESC))
  6  WHERE SLNO = 3;

       SAL
----------
      2975
```

Syntax for Nth MAX(SAL)

```
SELECT SAL
FROM (SELECT ROWNUM SLNO, SAL
        FROM (SELECT DISTINCT SAL
                FROM EMP
                ORDER BY SAL DESC))
WHERE SLNO = N;
```

Syntax for Nth MIN(SAL)

```
SELECT SAL
FROM (SELECT ROWNUM SLNO, SAL
        FROM (SELECT DISTINCT SAL
```

```
            FROM EMP
            ORDER BY SAL ASC))
WHERE SLNO = N;
```

Q264. WAQTD 7th minimum salary.

```
SELECT SAL
FROM (SELECT ROWNUM SLNO, SAL
      FROM (SELECT DISTINCT SAL
            FROM EMP
            ORDER BY SAL ASC))
WHERE SLNO = 7;
```

Q265. WAQTD name of an employee who is getting 5th Minimum salary.

```
SELECT ENAME
FROM EMP
WHERE SAL IN (SELECT SAL
              FROM (SELECT ROWNUM SLNO, SAL
                    FROM (SELECT DISTINCT SAL
                          FROM EMP
                          ORDER BY SAL ASC))
              WHERE SLNO = 5);
```

```
SQL>  SELECT ENAME
  2  FROM EMP
  3  WHERE SAL IN (SELECT SAL
  4                FROM (SELECT ROWNUM SLNO, SAL
  5                      FROM (SELECT DISTINCT SAL
  6                            FROM EMP
  7                            ORDER BY SAL ASC))
  8                WHERE SLNO = 5);

ENAME
----------
MILLER
```

Q266. WAQTD names of an employee who are earning 2nd maximum salary.

```
SELECT ENAME
FROM EMP
WHERE SAL IN (SELECT SAL
```

```
        FROM (SELECT ROWNUM SLNO, SAL
                FROM (SELECT DISTINCT SAL
                        FROM EMP
                        ORDER BY SAL DESC))
        WHERE SLNO = 2);
```

Q267. WAQTD DNAME of an employee earning 4th minimum salary.

```
SELECT DNAME
FROM EMP, DEPT
WHERE EMP.DEPTNO = DEPT.DEPTNO AND
    SAL IN (SELECT SAL
            FROM (SELECT ROWNUM SLNO, SAL
                    FROM (SELECT DISTINCT SAL
                            FROM EMP
                            ORDER BY SAL ASC))
            WHERE SLNO = 4);
```


DATA MANIPULATION LANGUAGE (DML)

There are 3 statements in DML:

1. INSERT
2. UPDATE
3. DELETE

```
SQL> SELECT *
  2  FROM BONUS;

no rows selected
```

Here, BONUS table has no records.

Consider this BONUS table, as an example. **INSERT**, **UPDATE** & **DELETE** the records in SQL.

BONUS

ENAME	JOB	SAL	COMM
HARSH	SALESMAN	5000	300
BHUVI	MANAGER	7000	200
HARDIK	CLERK	3000	0
NAWAZ	PRESIDENT	9000	500

INSERT

INSERT is used to *enter the records* to a particular table.

Syntax:

1. INSERT INTO TABLE_NAMES VALUES (V1, V2, V3, ……, V_N);
   Example: INSERT INTO BONUS VALUES ('HARSH', 'SALESMAN', 5000, 300);

```
SQL> INSERT INTO BONUS VALUES ('HARSH', 'SALESMAN', 5000, 300);

1 row created.

SQL> SELECT *
  2  FROM BONUS;

ENAME      JOB              SAL       COMM
---------- --------- ---------- ----------
HARSH      SALESMAN        5000        300
```

2. INSERT INTO TABLE_NAME (COL1, COL2, COL3, ……, COL_N)
   VALUES (V1, V2, V3, ……., V_N);

   Example: INSERT INTO BONUS (ENAME, JOB, SAL, COMM)

   VALUES ('BHUVI', 'MANAGER', 7000, 200);

Use **COMMIT** at the end, to save the inserted values of the table. If not used, the inserted data in the table *gets* erased *once* SQL is closed.

```
SQL> INSERT INTO BONUS (ENAME, JOB, SAL, COMM)
  2                                   VALUES ('BHUVI', 'MANAGER', 7000, 200);

1 row created.

SQL> COMMIT
  2  /

Commit complete.

SQL> SELECT *
  2  FROM BONUS;

ENAME      JOB              SAL       COMM
---------- --------- ---------- ----------
HARSH      SALESMAN        5000        300
BHUVI      MANAGER         7000        200
```

3. INSERT INTO TABLE_NAME (COL1, COL2, COL3, ……, COL_N)
   VALUES (&COL1, &COL2, &COL3, ……, &COL_N);

   Example: INSERT INTO BONUS (ENAME, JOB, SAL, COMM)

   VALUES (&ENAME, &JOB, &SAL, &COMM);

```
SQL> INSERT INTO BONUS (ENAME, JOB, SAL, COMM)
  2                                  VALUES (&ENAME, &JOB, &SAL, &COMM);
Enter value for ename: 'HARSH'
Enter value for job: 'SALESMAN'
Enter value for sal: 5000
Enter value for comm: 300
old   2:                                  VALUES (&ENAME, &JOB, &SAL, &COMM)
new   2:                                  VALUES ('HARSH', 'SALESMAN', 5000, 300)

1 row created.

SQL> /
Enter value for ename: 'BHUVI'
Enter value for job: 'MANAGER'
Enter value for sal: 7000
Enter value for comm: 200
old   2:                                  VALUES (&ENAME, &JOB, &SAL, &COMM)
new   2:                                  VALUES ('BHUVI', 'MANAGER', 7000, 200)

1 row created.

SQL> /
Enter value for ename: 'HARDIK'
Enter value for job: 'CLERK'
Enter value for sal: 3000
Enter value for comm: 0
old   2:                                  VALUES (&ENAME, &JOB, &SAL, &COMM)
new   2:                                  VALUES ('HARDIK', 'CLERK', 3000, 0)

1 row created.

SQL> /
Enter value for ename: 'NAWAZ'
Enter value for job: 'PRESIDENT'
Enter value for sal: 9000
Enter value for comm: 500
old   2:                                  VALUES (&ENAME, &JOB, &SAL, &COMM)
new   2:                                  VALUES ('NAWAZ', 'PRESIDENT', 9000, 500)

1 row created.


SQL> COMMIT
  2  /

Commit complete.

SQL> SELECT *
  2  FROM BONUS;

ENAME      JOB               SAL       COMM
---------- --------- ---------- ----------
HARSH      SALESMAN        5000        300
BHUVI      MANAGER         7000        200
HARDIK     CLERK           3000          0
NAWAZ      PRESIDENT       9000        500
```

UPDATE

UPDATE is used to *update the existing records* of a table.

Syntax:

UPDATE TABLE_NAME

SET COL1 = V1, COL2 = V2, ……, $COL_N = V_N$

[WHERE <FILTER_CONDITION>]

Example: UPADTE BONUS

SET SAL = 4000

WHERE ENAME = 'HARSH';

```
SQL> UPDATE BONUS
  2  SET SAL=4000
  3  WHERE ENAME='HARSH';

1 row updated.

SQL> SELECT *
  2  FROM BONUS;

ENAME      JOB              SAL       COMM
---------- --------- ---------- ----------
HARSH      SALESMAN         4000        300
BHUVI      MANAGER          7000        200
HARDIK     CLERK            3000          0
NAWAZ      PRESIDENT        9000        500
```


DELETE

DELETE is used to *delete the records* from a table.

Syntax:

DELETE

FROM TABLE_NAME

[WHERE <FILTER_CONDITION>];

Example: DELETE

FROM BONUS

WHERE ENAME = 'NAWAZ';

```
SQL> DELETE
  2      FROM BONUS
  3      WHERE ENAME = 'NAWAZ';

1 row deleted.

SQL> SELECT *
  2  FROM BONUS;

ENAME      JOB              SAL       COMM
---------- --------- ---------- ----------
HARSH      SALESMAN         4000        300
BHUVI      MANAGER          7000        200
HARDIK     CLERK            3000          0
```

Q268. WAQTD maximum salary of employee working as 'PRESIDENT' OR 'SALESMAN' OR, 'MANAGER' & earning more than FORD & his name must end with character 'G' or, 'S' & working in 'NEW YORK'.

```
SELECT MAX(SAL)
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND
      E.JOB IN ('PRESIDENT', 'SALESMAN', 'MANAGER') AND
      E.SAL > (SELECT E.SAL
               FROM EMP E
               WHERE E.ENAME = 'FORD') AND
      E.ENAME LIKE '%G' OR E.ENAME LIKE '%S' AND
      D.LOC = 'NEW YORK';
```

Q269. WAQTD DNAME of employees whose SLNO is 7.

```
SELECT DNAME
FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO
                 FROM (SELECT ROWNUM SLNO, EMP.*
                       FROM EMP)
                 WHERE SLNO = 7);
```

```
SQL> SELECT DNAME
  2  FROM DEPT
  3  WHERE DEPTNO IN (SELECT DEPTNO
  4                                    FROM (SELECT ROWNUM SLNO, EMP.*
  5                                          FROM EMP)
  6                                   WHERE SLNO = 7);

DNAME
--------------
ACCOUNTING
```

Q270. WAQTD name of employees working as 'SALESMAN' in same DEPT as that of BLAKE & he gets commission & his MANAGER hired in month of 'MAY' in 'CHICAGO' & manager's manager earning more than 'SCOTT' in 'NEW YORK'.

SELECT E1.ENAME

FROM EMP E1, EMP E2, EMP E3, DEPT D1, DEPT D2, DEPT D3

WHERE E1.MGR = E2.EMPNO AND

E2.MGR = E3.EMPNO AND

E1.DEPTNO = D1.DEPTNO AND

E2.DEPTNO = D2.DEPTNO AND

E3.DEPTNO = D3.DEPTNO AND

E1.JOB = 'SALESMAN' AND

E1.DEPTNO IN (SELECT E1.DEPTNO

FROM EMP E1

WHERE E1.ENAME = 'BLAKE') AND

E1.COMM IS NOT NULL AND E2.HIREDATE LIKE '%MAY%' AND D2.LOC = 'CHICAGO' AND

E3.SAL > (SELECT E3.SAL

FROM EMP E3

WHERE E3.ENAME = 'SCOTT' AND

D3.LOC = 'NEW YORK');

DATA DEFINATION LANGUAGE (DDL)

There are 5 statements:

1. CREATE
2. RENAME
3. ALTER
4. TRUNCATE
5. DROP

CREATE

It is used to create a table.

Syntax:

```
CREATE TABLE TABLE_NAME
(
        COLUMN_NAME_1 DATATYPE NOT NULL/ [NULL],
        COLUMN_NAME_2 DATATYPE NOT NULL/ [NULL],
        .
        .
        COLUMN_NAME_n DATATYPE NOT NULL/ [NULL],
        CONSTRAINT CONSTRAINT_REF_NAME UNIQUE (COLUMN_NAME),
        CONSTRAINT CONSTRAINT_REF_NAME CHECK (CONDITIONS),
        CONSTRAINT CONSTRAINT_REF_NAME PRIMARY KEY (COLUMN_NAME),
        CONSTRAINT CONSTRAINT_REF_NAME FOREIGN KEY (COLUMN_NAME),
        REFRENCES PARENT_TABLE_NAME (COLUMN_NAME)
);
```


Let's CREATE a table named STUDENT,

```
CREATE TABLE STUDENT
(
        SID NUMBER (3) NOT NULL,
        NAME VARCHAR (40) NOT NULL,
        BRANCH VARCHAR (30) NOT NULL,
        PERC NUMBER (3) NOT NULL,
        CONSTRAINT PERCCH CHECK (PERC>0),
        CONSTRAINT SIDPK PRIMARY KEY (SID)
);
```

```
SQL> CREATE TABLE STUDENT
  2  (
  3   SID NUMBER (3) NOT NULL,
  4   NAME VARCHAR (40) NOT NULL,
  5   BRANCH VARCHAR (30) NOT NULL,
  6   PERC NUMBER (3) NOT NULL,
  7   CONSTRAINT PERCCH CHECK (PERC>0),
  8   CONSTRAINT SIDPK PRIMARY KEY (SID)
  9  );

Table created.

SQL> SELECT *
  2  FROM TAB;

TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
DEPT                           TABLE
EMP                            TABLE
BONUS                          TABLE
SALGRADE                       TABLE
STUDENT                        TABLE
```

To Describe the COLUMN_NAMES of a table.

```
SQL> DESC STUDENT
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SID                                       NOT NULL NUMBER(3)
 NAME                                      NOT NULL VARCHAR2(40)
 BRANCH                                    NOT NULL VARCHAR2(30)
 PERC                                      NOT NULL NUMBER(3)
```

RENAME

It is used to change the name of a table.

Syntax:

RENAME CURENT_TABLE_NAME TO NEW_TABLE_NAME;

Example: RENAME STUDENT TO STUD;

ALTER

It is a statement which is used to modify the table.

Syntax:

1. **TO ADD A COLUMN:**
   ALTER TABLE_NAME
   ADD COLUMN_NAME DATATYPE [NULL/NOT NULL];

   Example: ALTER TABLE STUD
   ADD MOBILENO NUMBER (0) NOT NULL;

2. **TO DROP A COLUMN:**

ALTER TABLE TABLE_NAME
DROP COLUMN_NAME;

Example: ALTER TABLE STUD
DROP COLUMN MOBILENO;

3. **TO CHANGE THE DATATYPE:**

ALTER TABLE TABLE_NAME
MODIFY COLUMN_NAME NEW_DATATYPE;

Example: ALTER TABLE STUD
MODIFY NAME VARCHAR (30);

4. **TO CHANGE THE NOT NULL CONSTRAINT:**

ALTER TABLE TABLE_NAME
MODIFY COLUMN_NAME EXISTING DATATYPE NULL/ NOT NULL;

Example: ALTER TABLE STUD
MODIFY BRANCH VARCHAR (10) NULL;

5. **TO RENAME THE COLUMN:**

ALTER TABLE TABLE_NAME
RENAME COLUMN CURRENT_NAME TO NEW_NAME;

Example: ALTER TABLE STUD
RENAME MOBILENO TO MOB_NO;

TRUNCATE

It is used to delete all the records from a table permanently.

Syntax:

TRUNCATE TABLE TABLE_NAME;

DROP

It is used to drop a table.

Syntax:

DROP TABLE TABLE_NAME

1. **TO RECOVER THE TABLE (ONLY IN ORACLE):**

Syntax:

FLASHBACK TABLE TABLE_NAME

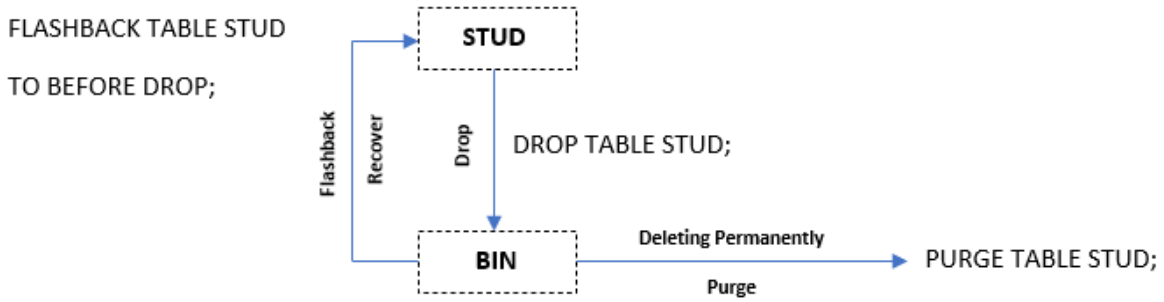
TO BEFORE DROP

[RENAME TO NEW_NAME];

2. **TO DROP THE TABLE FROM RECYCLE BIN**

Syntax:

PURGE TABLE TABLE_NAME;



TRANSACTION CONTROL LANGUAGE (TCL)

There are 3 statements:

1. COMMIT
2. SAVEPOINT
3. ROLLBACK

COMMIT

It is used to save the records.

Syntax: COMMIT;

SAVEPOINT

It is used to give the checkpoint to a particular operation.

Syntax: SAVEPOINT SAVEPOINT_NAME;

ROLLBACK

It is used as a UNDO operation.

Syntax: ROLLBACK;

ROLLBACK TO

It is *used* to go to the particular SAVEPOINT.

Syntax: ROLLBACK TO SAVEPOINT_NAME;

DATA CONTROL LANGUAGE (DCL)

It has 2 statements:

1. GRANT
2. REVOKE

GRANT

It is *used* to give the permission to another database to access the data.

Syntax:

GRANT SQL_STATEMENTS ON TABLE_NAME

TO USER_NAME;

Example: GRANT SELECT ON EMPLOYEES

TO SCOTT;

REVOKE

It is *used* to take back that permission.

Syntax:

REVOKE SQL_STATEMENTS ON TABLE_NAME

FROM USER_NAME;

Example: REVOKE SELECT ON EMPLOYEES

FROM SCOTT;

Q271. WAQTD first character of names of all the employees.

SELECT SUBSTR (ENAME, 1, 1)

FROM EMP;

```
SQL> SELECT SUBSTR (ENAME, 1, 1)
  2  FROM EMP;

S
-
S
A
W
J
M
B
C
S
K
T
A
J
F
M

14 rows selected.
```

Q272. WAQTD first 3 characters of the name of all the employees.

SELECT SUBSTR (ENAME, 1, 3)

FROM EMP;

Q273. WAQTD details of an employee whose name starts with character 'A' using single row function.

SELECT *

FROM EMP

WHERE SUBSTR (ENAME, 1, 1) = 'A';

```
SQL> SELECT *
  2  FROM EMP
  3  WHERE SUBSTR (ENAME, 1, 1) = 'A';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN         7698 20-FEB-81       1600        300         30
      7876 ADAMS      CLERK            7788 23-MAY-87       1100                    20
```

Q274. WAQTD details of an employee whose designation ends with character 'N'.

SELECT *

FROM EMP

WHERE SUBSTR (JOB, -1, 1) = 'N';

Q275. WAQTD names of an employee whose name starts with character 'A' or 'S'

SELECT ENAME

FROM EMP

WHERE SUBSTR (ENAME, 1, 1) IN (‘A’, ‘S’);

***Q276. WAQTD details of an employee whose name starts with vowels.**

SELECT *

FROM EMP

WHERE SUBSTR (ENAME, 1, 1) IN (‘A’, ‘E’, ‘I’, ‘O’, ‘U’);

Q277. WAQTD names of an employee along with number of characters they are having in their name.

SELECT ENAME, LENGTH (‘ENAME’)

FROM EMP;

```
SQL> SELECT ENAME, LENGTH ('ENAME')
  2  FROM EMP;

ENAME      LENGTH('ENAME')
---------- ---------------
SMITH                    5
ALLEN                    5
WARD                     5
JONES                    5
MARTIN                   5
BLAKE                    5
CLARK                    5
SCOTT                    5
KING                     5
TURNER                   5
ADAMS                    5
JAMES                    5
FORD                     5
MILLER                   5

14 rows selected.
```


Q278. WAQTD number of employees having exactly 4 characters in his name using single row function.

SELECT COUNT (*)

FROM EMP

WHERE LENGTH (ENAME) = 4;

Q279. WAQTD name in uppercase, designation in lowercase if their name has five characters & getting 3-digit salary.

SELECT UPPER (ENAME), LOWER (JOB)

FROM EMP

WHERE LENGTH (ENAME) = 5 AND LENGTH (SAL) = 3;

```
SQL> SELECT UPPER (ENAME), LOWER (JOB)
  2  FROM EMP
  3  WHERE LENGTH (ENAME) = 5 AND LENGTH (SAL) = 3;

UPPER(ENAM LOWER(JOB
---------- ---------
SMITH      clerk
JAMES      clerk
```

****Q280. WAQTD first character & last character of employee name in uppercase & rest of the characters in lowercase.**

SELECT UPPER(SUBSTR(ENAME,1,1)) || LOWER(SUBSTR(ENAME,2,LENGTH(ENAME)-2)) || UPPER(SUBSTR(ENAME,-1,1))

FROM EMP;

```
SQL> SELECT UPPER(SUBSTR(ENAME,1,1)) || LOWER(SUBSTR(ENAME,2,LENGTH(ENAME)-2)) || UPPER(SUBSTR(ENAME
,-1,1))
  2  FROM EMP;

UPPER(SUBST
-----------
SmitH
AlleN
WarD
JoneS
MartiN
BlakE
ClarK
ScotT
KinG
TurneR
AdamS
JameS
ForD
MilleR

14 rows selected.
```

****Q281. WAQTD first half of employee name in uppercase & second half in lowercase & reverse.**

SELECT UPPER(SUBSTR(ENAME,1,2)) || REVERSE(LOWER(SUBSTR(ENAME,3,LENGTH(ENAME)-1)))

FROM EMP;

Q282. WAQTD details of an employee hired on SATURDAY.

SELECT *

FROM EMP

WHERE TO_CHAR (HIREDATE, ‘DY’) = ‘SAT’;

Q283. WAQTD details of an employee hired on FRIDAY, SATURDAY & SUNDAY.

SELECT *

FROM EMP

WHERE TO_CHAR (HIREDATE, ‘DY’) IN (‘FRIDAY’, ‘SATUDAY’, ‘SUNDAY’);

Q284. WAQTD details of an employee hired in a month of OCTOBER, NOVEMBER & DECEMBER.

```
SELECT *
FROM EMP
WHERE TO_CHAR (HIREDATE, 'MON') IN ('OCT', 'NOV', 'DEC');
```

OR,

```
SELECT *
FROM EMP
WHERE TO_CHAR (HIREDATE, 'MM') IN (10, 11, 12);
```

Q285. WAQTD details of an employee if their name having character 'A' using Single Row Function.

```
SELECT *
FROM EMP
WHERE INSTR (ENAME, 'A', 1, 1) > 0;
```

Q286. WAQTD details of an employee who is having at least two 'A' in his name using single row function.

```
SELECT *
FROM EMP
WHERE INSTR (ENAME, 'A', 1, 2) > 0;
```

Q287. WAQTD total salary given to each employee.

```
SELECT ENAME, SAL, COMM, SAL+NVL (COMM, 0)
FROM EMP;
```

***Explanation of Execution**:*

1600 + NVL (300, 0) = 1600 + 300 = 1900

5000 + NVL (NULL, 0) = 5000 + 0 = 5000

1500 + NVL (0,0) = 1500 + 0 = 1500

```
SQL> SELECT ENAME, SAL, COMM, SAL+NVL(COMM,0)
  2  FROM EMP;

ENAME             SAL       COMM SAL+NVL(COMM,0)
---------- ---------- ---------- ---------------
SMITH             800                       800
ALLEN            1600        300            1900
WARD             1250        500            1750
JONES            2975                      2975
MARTIN           1250       1400            2650
BLAKE            2850                      2850
CLARK            2450                      2450
SCOTT            3000                      3000
KING             5000                      5000
TURNER           1500          0            1500
ADAMS            1100                      1100
JAMES             950                       950
FORD             3000                      3000
MILLER           1300                      1300

14 rows selected.
```

FUNCTIONAL DEPENDENCY

Let us consider the relation '**R**' with 2 attributes '**X**' & '**Y**' respectively in which attribute '**X**' determines attribute '**Y**'.

OR, in other words, '**Y**' is *dependen*t on '**X**', there exists Functional Dependency.

R → {X, Y}

X → Y

Y is dependent on **X**.

TYPES OF FUNCTIONAL DEPENDENCY

1. Total Functional Dependency
2. Partial Functional Dependency
3. Transitive Functional Dependency

TOTAL FUNCTIONAL DEPENDENCY

If all the attribute in a relation is *determined* by a single attribute which *is* a Key Attribute, then there exists Total Functional Dependency.

Example: Let us consider, a relation with 4 attributes **A**, **B**, **C** & **D**.

In which '**A**' is a **Key Attribute**.

R → {A, B, C, D}

A → B

A → C

A → D

There exists Total Functional Dependency.

A → {B, C, D}

PARTIAL FUNCTIONAL DEPENDENCY

For a partial functional dependency to exists their *must* a Composite Key Attribute.

One of the Attribute in Composite Key relation determines another attribute separately & this is known as Partial Functional Dependency.

Example: Let us consider a relation 'R' with 4 attributes **A**, **B**, **C** & **D**.

In which '**A**' & '**B**' are **Composite Key Attributes**.

R → {A, B, C, D}

(A, B) → (C, D)

B → {C}

There exists Partial Functional Dependency.

TRANSITIVE FUNCTIONAL DEPENDENCY

If an attribute is determined by a non-key attribute which *in turn* is determined by a Key Attribute, then there exists Transitive Functional Dependency.

Example: Let us consider a relation with 4 attributes **A**, **B**, **C** & **D**.

In which '**A**' is a **Key Attribute**.

R → {A, B, C, D}

A → B

D → C

A → D

A → C

There exists Transitive Functional Dependency.

***DIFERRENCE BETWEEN TRUNCATE, DROP & DELETE

	TRUNCATE	DROP	DELETE
1.	TRUNCATE is used to delete all the records from a table permanently.	DROP is used to delete the table.	DELETE is used to delete a particular record from a table.

2.	We can't get back the table that is truncated.	We can get back the table that is dropped by using FLASHBACK.	We can get back the records that is deleted by using ROLLBACK.
3.	**SYNTAX:** TRUNCATE TABLE table_name	**SYNTAX:** DROP TABLE table_name	**SYNTAX:** DELETE FROM TABLE WHERE [conditions]

***NORMALIZATION

- It is process of reducing a larger table into smaller table.
- In order to remove the redundancy & anomaly by identifying their functional dependency.

LEVELS OF NORMAL FORM

1. First Normal Form (1NF)
2. Second Normal Form (2NF)
3. Third Normal Form (3NF)
4. Boyce-Codd Normal Form (BCNF)

NOTE:

A table is said to be normalized, if we reduce the table to 3rd normal form.

1NF

A table is said to be in 1st Normal Form, if it satisfies the following conditions

- A table should not consist of multi-valued data.
- A table should not have Duplicate or, Repeated values.

R → {EMPNO, ENAME, SAL, COMM, DEPTNO, DNAME, LOC}

STUDENT

ID	NAME	SKILLS
101	GOUTHAM	JAVA, SQL
102	PAWAN	SQL
103	SANTOSH	MT, SQL
104	ASHUTOSH	SQL, MT
102	PAWAN	JAVA

STUDENT

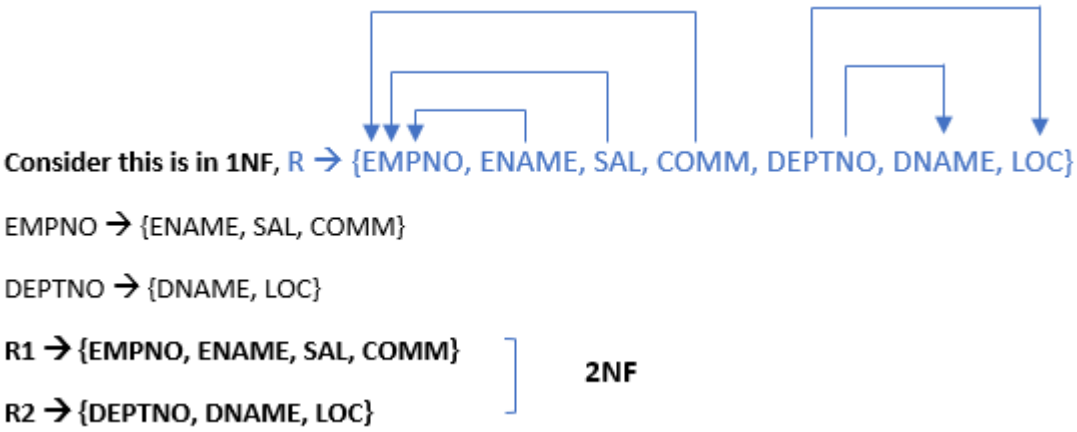
ID	NAME	SKILL_1	SKILL_2
101	GOUTHAM	JAVA	SQL
102	PAWAN	SQL	JAVA
103	SANTOSH	MT	SQL
104	ASHUTOSH	SQL	MT
102	PAWAN	JAVA	

1NF

2NF

A table is said to be in 2nd normal form, if it satisfies the following conditions

- A table should be in 1st normal form.
- The table should not have partial functional dependency.



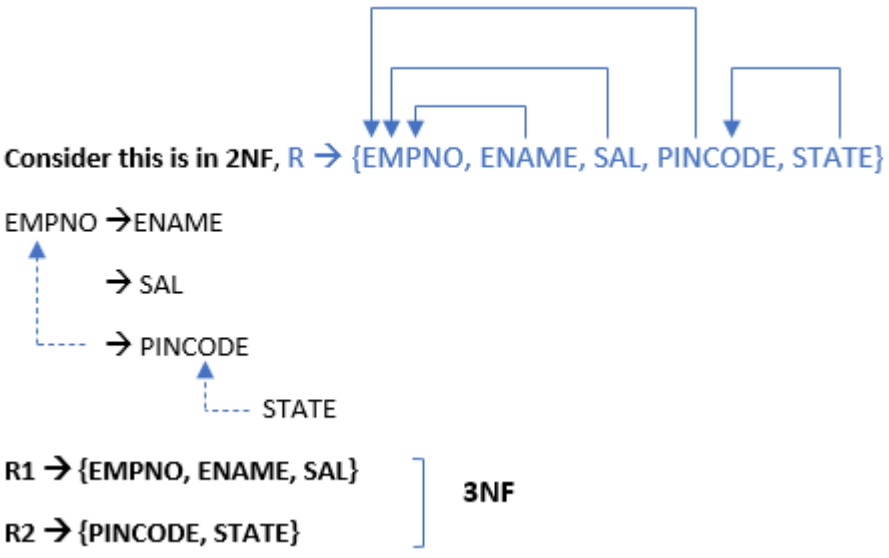
NOTE:

If the table consists of partial functional dependency, then the attributes responsible are removed from the table.

3NF

A table is said to be in 3rd normal form, if it satisfies the following conditions

- A table should be in 2nd normal form.
- The table should not have transitive functional dependency.



NOTE:

If the table consists of transitive functional dependency, then the attributes responsible are removed from the table.