SQL

Application/Software:

Software is a collection of programs which will help to performed human task.

Types of applications

- Web Application
- Client-Server Application
- Stand-Alone Application

1) Web Application

- Web Applications are those applications which are accessed with thorough browsers.
- To use these applications, we should use URL(Uniform Resource Locator) of these applications and internet connectivity is must.
- To use these applications, we need not to download this application on local machine or system.
- E.g.: www.facebook.com
  www.amazon.com

2) Client-Server Application

- Server is nothing but a machine which are used with configuration when application is installed and accessed by multiple users at a time.
- In Client-server application developers develop two Software:
  - **Client software:**
  - **Server software:**
- To use these applications, we have to install clint software on local machine and use it. Internet is must for using these applications.
- E.g.: WhatsApp, Instagram, etc.

3) Stand-Alone Software:

To use these stand-alone applications, we have to download it and install it on the local machine, but to use It, internet is not required.

Stand-alone applications do not contain any centralized database that means these applications does not have database layer.

E.g.: Microsoft office, calculator.

Data:

Data is called as raw data or single unit.

OR

Data describe attribute of entity.

		Name :	Sumit	
Entity → Attributes →		Age :	22	← **Data**
		DOB :	7/06/2003	
		Gender :	male	

Data is classifying in two types:

Structured data

Unstructured data

1. **Structured data:**

   Data which is stored in specific structure is known as structured data.

   E.g.: data in the form of table.

2. **Unstructured data:**

   Data which is stored without maintaining special structure is known as unstructured data.

   E.g.: data in the form of word, PDF, text, etc.

   Note: SQL is used to handle structured data only.

Information:

Information is called as collection of data or processed data.

e.g.: Pune is a beautiful city.

Database:

Database is a place where we store the data in organized way for easy retravel.

DBMS (Database Management System):

It is a system/ software which is used to interact with the data stored in the database for data manipulation using query language.

To manage data, we have to perform CRUD operations.

Classification:

- Flat file DBMS
- Hierarchical DBMS
- Network DBMS
- RDBMS

i. **Flat file DBMS**

Here data is stored in the terms of files and folders.

Advantages:

Easy to store data.

Disadvantage:

Data duplication.

ii. **Hierarchical DBMS:**

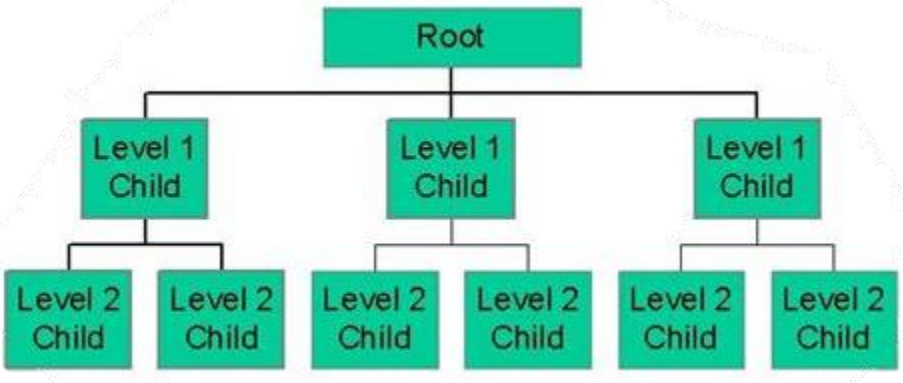
Here data is stored in tree structure or parent-child relation.

Advantages:

No Data duplication.

Disadvantage:

Time consuming.



iii. **Network DBMS**

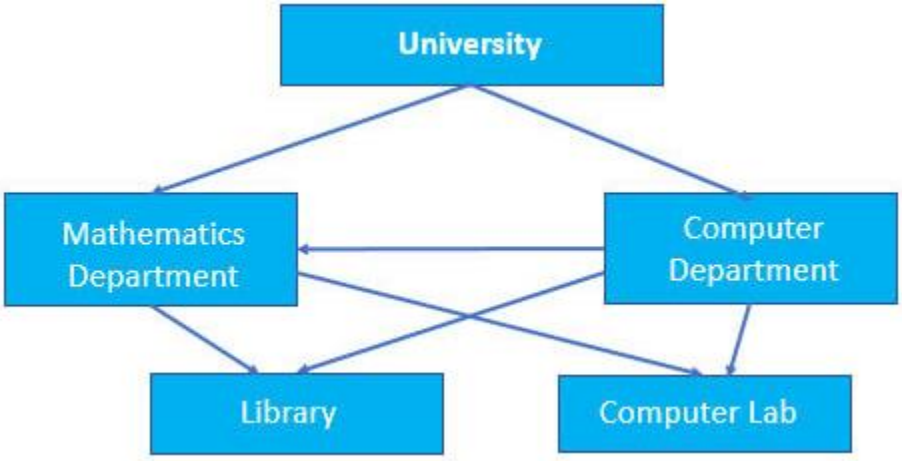
Here data is stored in network style i.e. all data nodes are connected to each other.

Advantages:

Less time consuming.

Disadvantage:

Complex Stucture.



iv. RBDMS (Relational DBMS)

- It is a system/ software which is used to interact with the data stored in database for data manipulation using query language with relational model.
- Any DBMS that follows relational model becomes RDBMS or we can say any dbms which follows rules of EF codd it becomes RDBMS.

Following the rules of EF code:

- Data should be stored in the form of table.
- Cell should be atomic in nature.
- Null should be independent of data types.
- We should able to establish relation between tables.

What is a table ?

Table is a logical organization of data which is consists of rows and columns.

Column:

A column is also known as attributes fields. A column is used to represent property of all the entities.

Row:

Row is also known as record as record or tuple.

A row is used to represent properties of an individual entity.

Cell:

Cell is the smallest unit of a table in Which we stores the data.

The interaction of rows and columns generate cells.

Data Types:

Data types are the types of data which we apply on the column of a table.

Following are the types of data :

- Number (int, float)
- Char
- Varchar
- Date

Constraints

Constraints are the condition which we will apply on the columns of the table for extra validation of records. SQL constraints are used to specify rules for the data in a table.

Data Integrity:

Data integrity is used to maintain accuracy, correctness and consistency of the table

Null:

- Null is missing value or unknown values.
- Null is neither zero nor black place.
- All the null values are unique.
- Memory is not allocated to null values.
- Any arithmetic operations performed with the null; it results in null only.
- e.g.: 100 + null = null.

Number:

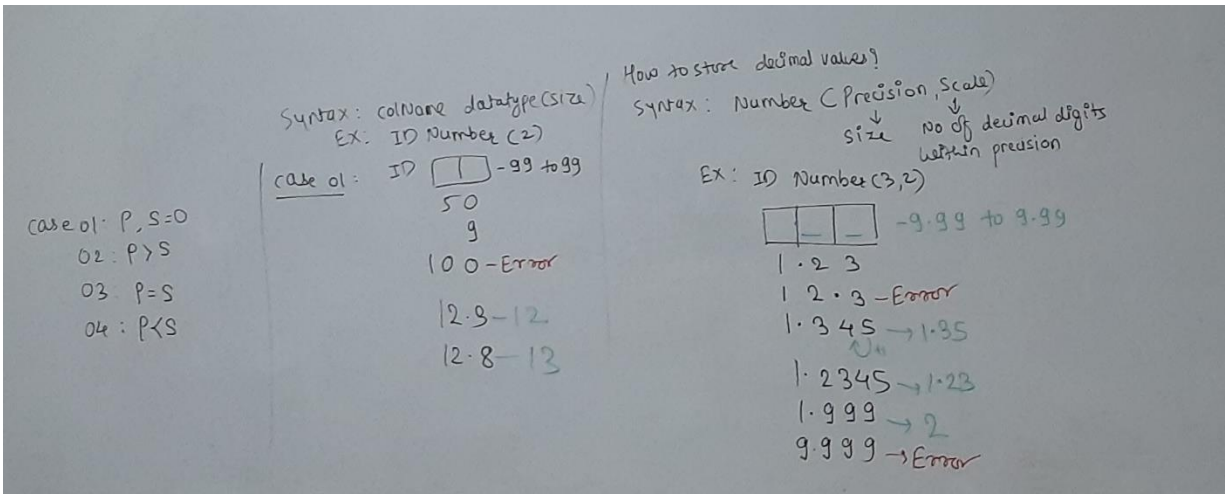
- Number datatype is used to store numeric values
- Number datatype can accept two parameters
    i. Precision
    ii. Scale

Precision:

- Precision is used to determine number of digits used to store integer values.
- Precision values we can give up to 38.

Scale:

- Scale is used to determine number of digits used to store decimal values within precision block.
- Scale value we can give up to 127



Case 4:

ID number (2 , 3)

0. □□

0.023

0.23 →Error

.456

.099

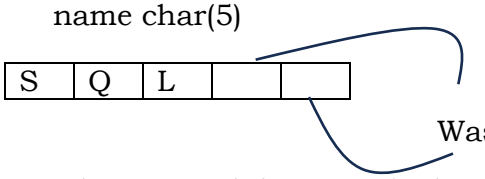
.0999 → Error

Number of zeros = scale – precision

Char:

Char is used to store alpha-numeric data

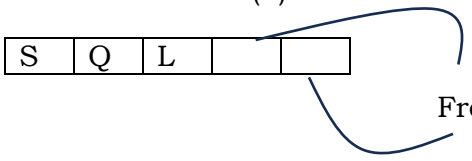
Example: name char(5)



Varchar

Varchar is used to store alphanumeric data

Example: name varchar(5)



Difference between char and varchar

Char	Varchar
Char is a fixed memory allocation	Varchar is a variable memory allocation
Unused memory location in char is waste	Unused memory location in varchar is free
Unused memory location in char is occupied by blank spaces	Unused memory location in varchar is occupied by blank spaces
Size of char is 2000	Size of varchar is 4000

Date

Date datatype is used to insert the date related values.

E.g.: Hire date, date of admission, date of joining.

Default format of date is **DD-MOY-YY**

```
SQL> CREATE TABLE STUD(NAME VARCHAR(5),JOINING_DATE DATE);

Table created.

SQL> INSERT INTO STUD VALUES('A', '31-DECEMBER-2023');

1 row created.

SQL> INSERT INTO STUD VALUES('A', '31-DEC-23');

1 row created.

SQL> INSERT INTO STUD VALUES('A', '31-DECEMBER-23');

1 row created.

SQL> SELECT * FROM STUD;

NAME  JOINING_D
----- ---------
A     31-DEC-23
A     31-DEC-23
A     31-DEC-23
```

Constraints

Constraints are the condition which we will apply on the columns of the table for extra validation of records

Types of constraints:

- Not Null
- Unique
- Primary Key
- Foreign Key
- Check
- Default

Not Null:

- Not null constraint will not allow any null value to get inserted in a column of a table
- We can insert duplicate records.
- Not null constraints we can apply on multiple columns of a table
- E.g.: CREATE TABLE STUD(ID NUMBER(2), NAME VARCHAR(10) NOT NULL);

```
SQL> insert into stud values(1,'A');

1 row created.

SQL> select * from stud;

        ID NAME
---------- ----------
         1 A
```

Unique:

- Unique constraint will not allow any duplicate record to get inserted in a column of a table
- We can insert multiple null values.
- We can apply on multiple columns of a table.
- E.g.: CREATE TABLE STUD(ID NUMBER(2), NAME VARCHAR(10) NOT NULL, C_NO NUMBER(10) UNIQUE);

```
SQL> INSERT INTO STUD VALUES(2,'B',12456);
INSERT INTO STUD VALUES(2,'B',12456)
*
ERROR at line 1:
ORA-00001: unique constraint (SCOTT.SYS_C008045) violated


SQL> SELECT * FROM STUD;

        ID NAME                 C_NO
---------- ---------- ----------
         1 A                   12456
```

Primary key:

- Primary key is used for unique identification of records
- Primary key is a combination of **unique + not null.**
- **Primary key** we can apply only on one column of a table.
- Creation of a primary key is not mandatory but it is recommended to create a primary key in a table.

E.g.: CREATE TABLE STUD (ID NUMBER(2) PRIMARY KEY,

NAME VARCHAR(10) NOT NULL,

CONTACT_NO NUMBER(10) UNIQUE);

	ID	NAME	CONTACT_NO
	1	A	123
ERROR →	2	B	123
	2	B	124
ERROR →	2	C	NULL
ERROR →	NULL	D	125
	5	E	126

Difference between primary key and unique key

Primary Key	Unique Key
Primary key we can create only once	Unique key we can create multiple's
Primary key will not allow null values	Unique key will allow multiple null values

Check

Check is a user defined constraint

Check constraint Will not allow to record which never satisfy the condition

E.g.: CREATE TABLE STUD(ID NUMBER(2) PRMARY KEY,

NAME VARCHAR(10) NOT NULL,

MARKS NUMBER(2) CHECK(MARKS>55));

ID	NAME	MARKS
1	A	56
2	B	56
3	C	55
4	D	NULL
5	E	100

CREATE TABLE STUD(ID NUMBER(2) PRMARY KEY,

NAME VARCHAR(10) NOT NULL,

CONTACT_NO NUMBER(10) CHECK(LENGTH(CONTACT_NO) = 10 UNIQUE NOT NULL

WAQTD all the details from the employee table.

```
SQL> select * 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.
```

WAQTD names of all the employees

```
SQL> select ename from emp;

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

14 rows selected.
```

WAQTD names and salary given to all the employees

```
SQL> select ename, sal from emp;

ENAME                SAL
---------- ----------
SMITH                800
ALLEN               1600
WARD                1250
JONES               2975
MARTIN              1250
BLAKE               2850
CLARK               2450
SCOTT               3000
KING                5000
TURNER              1500
ADAMS               1100
JAMES                950
FORD                3000
MILLER              1300

14 rows selected.
```

WAQTD name and commission given to all the employees.

```
SQL> select ename, comm from emp;

ENAME               COMM
---------- ----------
SMITH
ALLEN                300
WARD                 500
JONES
MARTIN              1400
BLAKE
CLARK
SCOTT
KING
TURNER                 0
ADAMS
JAMES
FORD
MILLER

14 rows selected.
```

WAQTD employee ID and dept no of all the employees in employee table.

```
SQL> select empno , deptno from emp;

     EMPNO     DEPTNO
---------- ----------
      7369         20
      7499         30
      7521         30
      7566         20
      7654         30
      7698         30
      7782         10
      7788         20
      7839         10
      7844         30
      7876         20
      7900         30
      7902         20
      7934         10

14 rows selected.
```

WAQTD ename and hiredate of all the employees

```
SQL> select ename, hiredate from emp;

ENAME      HIREDATE
---------- ---------
SMITH      17-DEC-80
ALLEN      20-FEB-81
WARD       22-FEB-81
JONES      02-APR-81
MARTIN     28-SEP-81
BLAKE      01-MAY-81
CLARK      09-JUN-81
SCOTT      19-APR-87
KING       17-NOV-81
TURNER     08-SEP-81
ADAMS      23-MAY-87
JAMES      03-DEC-81
FORD       03-DEC-81
MILLER     23-JAN-82

14 rows selected.
```

WAQTD name and designation of all the employees

```
SQL> select ename, job from emp;

ENAME      JOB
---------- ---------
SMITH      CLERK
ALLEN      SALESMAN
WARD       SALESMAN
JONES      MANAGER
MARTIN     SALESMAN
BLAKE      MANAGER
CLARK      MANAGER
SCOTT      ANALYST
KING       PRESIDENT
TURNER     SALESMAN
ADAMS      CLERK
JAMES      CLERK
FORD       ANALYST
MILLER     CLERK

14 rows selected.
```

WAQTD name, job, salary given to all the employees.

```
SQL> select ename, job, sal from emp;

ENAME      JOB              SAL
---------- --------- ----------
SMITH      CLERK             800
ALLEN      SALESMAN         1600
WARD       SALESMAN         1250
JONES      MANAGER          2975
MARTIN     SALESMAN         1250
BLAKE      MANAGER          2850
CLARK      MANAGER          2450
SCOTT      ANALYST          3000
KING       PRESIDENT        5000
TURNER     SALESMAN         1500
ADAMS      CLERK            1100
JAMES      CLERK             950
FORD       ANALYST          3000
MILLER     CLERK            1300

14 rows selected.
```

WAQTD dname present in department table.

```
SQL> select dname from dept;

DNAME
--------------
ACCOUNTING
RESEARCH
SALES
OPERATIONS
```

WAQTD dname and location present in dept table.

```
SQL> select dname , loc from dept;

DNAME          LOC
-------------- -------------
ACCOUNTING     NEW YORK
RESEARCH       DALLAS
SALES          CHICAGO
OPERATIONS     BOSTON
```

Operators:

Operators are symbols which are used to perform specific operations.

Operands:

Operands are the inputs which are given as the argument to the operators.

Types of operators:

Arithmetic operators: (+, -, * /)

Arithmetic operators used for performing mathematical operations with the data present inside our table. We are using the arithmetic operator with the numeric data type only.

Comparison operators: (=, !=, <>)

SQL Comparison Operators are **used to compare two values and check if they meet the specific criteria**

Relational operators: (<, >, <=, >=)

Relational operators compare two expressions or values and return a Boolean result.

Logical operators (AND, OR, NOT)

- **AND , OR** operator is used to filter the record based on more than one condition.
- **AND** operator selects the record if all the conditions given by **AND** operator are true
- **OR** operator selects the record if any one of the conditions given by **OR** operator is true
- **NOT** operator is used to inverse the result of a query.

Concatenation operators FUNCT(||)

Concatenation operators are used for joining character strings.

Special operators (IN, NOT IN, BETWEEN, LIKE, NOT LIKE, IS, IS NOT)

IN : It is a multi-valued operator which can accept multiple values at the RHS.

IN operator returns true if any of the condition is satisfied at RHS.

NOT IN: It is similar to IN operator, instead of selecting the values it rejects the values.

BETWEEN :

- BETWEEN operator is used to display the record is a given range
- It Is used whenever we have ranges of values

NOT BETWEEN: It is opposite of between operator.

LIKE:

- LIKE operator is used for pattern matching
- Inside pattern we are using two special characters those are known as wildcard characters or like operator's special characters.

IS

- **IS** operator is used to check whether values of the column are null;
- **IS** operator is specially made for null value comparison.

Sub-Query operators (ALL, ANY, EXISTS, NOT EXISTS)

ALL

The ALL operator compares a column value or literal value with the result of a subquery that returns a single-column values. It return true if all the values meets the condition.

ANY

The SQL ANY operators is used to perform a comparison between a single value and a range of values returned by the subquery. It return true if any of the values meets the condition.

Arithmetic Operators:

WAQTD incremented salary of all the employees by Rs. 100.

```
SQL> select ename, sal Old_sal, sal+100 New_sal from emp;

ENAME          OLD_SAL    NEW_SAL
---------- ---------- ----------
SMITH             800        900
ALLEN            1600       1700
WARD             1250       1350
JONES            2975       3075
MARTIN           1250       1350
BLAKE            2850       2950
CLARK            2450       2550
SCOTT            3000       3100
KING             5000       5100
TURNER           1500       1600
ADAMS            1100       1200
JAMES             950       1050
FORD             3000       3100
MILLER           1300       1400

14 rows selected.
```

WAQTD decrement salary of the employees who are working in dept 20 by Rs 50.

```
SQL> select sal old_sal, sal-50 as Decr_sal from emp where deptno=20;

   OLD_SAL   DECR_SAL
---------- ----------
       800        750
      2975       2925
      3000       2950
      1100       1050
      3000       2950
```

WAQTD annual salary of all salesmen.

```
SQL> select emp.*, sal*12 Annual_Salary from emp where job='SALESMAN';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO ANNUAL_SALARY
---------- ---------- --------- ---------- --------- ---------- ---------- ---------- -------------
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30         19200
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30         15000
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30         15000
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30         18000
```

WAQTD 10% hike in the salary of all clerk.

```
SQL>  SELECT ENAME, SAL + SAL*10/100 HIKE_SAL FROM EMP WHERE JOB ='CLERK';

ENAME      HIKE_SAL
---------- ----------
SMITH             880
ADAMS            1210
JAMES            1045
MILLER           1430
```

Relational operators: (<, >, <=, >=)

WAQTD of employees who are earning more than 1000.

```
SQL> select * from emp where sal>1000;
```

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
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
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

```
12 rows selected.
```

WAQTD Of employees who are earning less than 3000.

```
SQL> select * from emp where sal < 3000;
```

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.
```

WAQTD of employees who learns at least 1250.

```
SQL>  select * from emp where sal>=1250;

     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
      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
      7902 FORD       ANALYST          7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK            7782 23-JAN-82       1300                    10

11 rows selected.
```

WAQTD of employees who earns at most 3000.

```
SQL> select * from emp where sal<=3000;

     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
      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

13 rows selected.
```

WAQTD of employees who are not clerk

```
SQL> select * from emp where job !='CLERK';

     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
      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
      7902 FORD       ANALYST          7566 03-DEC-81       3000                    20

10 rows selected.
```

WAQTD Who are not working in dept 20.

```
SQL> SELECT * FROM EMP WHERE DEPTNO<>20;

     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
      7698 BLAKE      MANAGER          7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER          7839 09-JUN-81       2450                    10
      7839 KING       PRESIDENT             17-NOV-81       5000                    10
      7844 TURNER     SALESMAN         7698 08-SEP-81       1500          0         30
      7900 JAMES      CLERK            7698 03-DEC-81        950                    30
      7934 MILLER     CLERK            7782 23-JAN-82       1300                    10
```

WAQTD List all salesman

```
SQL> SELECT * FROM EMP 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
```

Logical Operators:

AND , OR operator is used to filter the record based on more than one condition.

AND operator selects the record if all the conditions given by **AND** operator are true

OR operator selects the record if any one of the conditions given by **OR** operator is true

NOT operator is used to inverse the result of a query.

WAQTD list all the clerks who are earning more than 1000.

```
SQL> select * from emp where job='CLERK' and sal>1000;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
```

WAQTD list the employees of dept 10,20.

```
SQL> select * from emp where deptno=10 or deptno=20;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      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
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

8 rows selected.
```

WAQTD employees who are earning more than 1250 but less than 3000.

```
SQL> select * from emp where sal>1250 and sal<3000;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      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
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

6 rows selected.
```

WAQTD employees who are hired after 81 and before 87.

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

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

WAQTD list all the salesman and managers.

```
SQL> select * from emp where job='SALESMAN' OR JOB='MANAGER';

     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.
```

```
SQL> select * from emp where not deptno=20;

     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
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
```

Special operators:

IS

IS operator is used to check whether values of the column are null;

IS operator is specially made for null value comparison.

Syntax: select */colName form tableName

Where colName is null/is not null;

IS NULL : It returns true if it is able to get null value in columns.

IS NOT NULL: It returns true if it is not able to get null value in columns.

WAQTD employees who are not earning commission:

Select * from emp where comm is null;

Select * from emp where comm=0 or comm is null;

WAQTD employees who's commission is not null:

Select ename from emp where comm is not null;

IN Operator:

In operator is used to compare one LHS with multiple RHS.

IN : It is a multi-valued operator which can accept multiple values at the RHS.

IN operator returns true if any of the condition is satisfied at RHS.

Syntax: select */col_name from table_name where col_name in(val1, val2,…);

NOT IN: It is similar to IN operator, instead of selecting the values it rejects the values.

Syntax: select */col_name from table_name where col_name NOT IN (val1, val2,…);

WAQTD employees who are working in deptno 10,20;

```
SQL> select * from emp where deptno=10 or deptno=20;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      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
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

8 rows selected.
```

```
SQL> select * from emp where deptno in(10,20);

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      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
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

8 rows selected.
```

WAQTD employees who are not working in deptno 10,20.

```
SQL> select * from emp where deptno<> 10 and deptno <>20;

     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
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30

6 rows selected.
```

```
SQL> select * from emp where deptno not in(10,20);

     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
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30

6 rows selected.
```

BETWEEN :

- BETWEEN operator is used to display the record is a given range
- It Is used whenever we have ranges of values

Syntax: select */col_name from table_name where col_name between val1 and val2;

NOT BETWEEN: It is opposite of between operator.

Syntax:

select */col_name from table_name where col_name between lower_range and Higher_range;

ex:

```
SQL> SELECT * FROM EMP WHERE ENAME BETWEEN 'A' AND 'SZ';

     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
      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
      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

12 rows selected.

SQL> SELECT * FROM EMP WHERE ENAME BETWEEN 'ADAMS' AND 'SMITH';

     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
      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
      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

12 rows selected.
```

```
SQL> select * from emp where hiredate between '01-JAN-82' AND '31-DEC-86';

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

SQL> select * from emp where sal between 1250 and 3000;

     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
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

10 rows selected.
```

LIKE

- LIKE operator is used for pattern matching
- Inside pattern we are using two special characters those are known as wildcard characters or like operator's special characters.

Following is the two wildcards:

'%' : Used to compare zero or multiple characters.

'_': used to compare one char at a time.

Syntax:

select */col_name from table_name where col_name like 'pattern' escape 'escape_char';

Optional

Ex:

Name starts with 'A' : ename like 'A%'

Name ends with 'N' : ename like '%N'

Name starts with 'A'

And ends with'N': ename like 'A%N'

Name contains 'A'

anywhere in name: ename like '%A%'

name contains exactly

4 characters: ename like'____'

Name contains 'A'

At 2nd position: ename like '_A%'

Name contains 'I'

At 2nd last position: ename like '%I_'

Escape:

Escape character is used in pattern string to indicate that any wildcard character that occurs after escape character should be treated as regular character.

select * from emp where ename like '%@%%' escape '@';

Concat operator:

Concat operator is used to join two or more string together

Syntax:

ARGS1 || ARGS2

```
SQL> SELECT 'MY NAME IS ' || ENAME FROM EMP;

'MYNAMEIS'||ENAME
--------------------
MY NAME IS SMITH
MY NAME IS ALLEN
MY NAME IS WARD
MY NAME IS JONES
MY NAME IS MARTIN
MY NAME IS BLAKE
MY NAME IS CLARK
MY NAME IS SCOTT
MY NAME IS KING
MY NAME IS TURNER
MY NAME IS ADAMS
MY NAME IS JAMES
MY NAME IS FORD
MY NAME IS MILLER

14 rows selected.
```

```
SQL> SELECT 'QSP ' || 'JSP ' || 'HADAPSAR' FROM DUAL;

'QSP'||'JSP'||'H
----------------
QSP JSP HADAPSAR
```

```
SQL> SELECT 'MY NAME IS ' || ENAME FROM EMP;

'MYNAMEIS'||ENAME
--------------------
MY NAME IS SMITH
MY NAME IS ALLEN
MY NAME IS WARD
MY NAME IS JONES
MY NAME IS MARTIN
MY NAME IS BLAKE
MY NAME IS CLARK
MY NAME IS SCOTT
MY NAME IS KING
MY NAME IS TURNER
MY NAME IS ADAMS
MY NAME IS JAMES
MY NAME IS FORD
MY NAME IS MILLER

14 rows selected.
```

All Operator:

- All operator is a special operator which has to be used along with relational operator which will compare the value present at the LHS with all the values present at RHS
- All operator returns true if all the values at the RHS have satisfied

Assignment on operators:

1. **List all the employees whose commission is null**

```
SQL> select * from emp where comm is null;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
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
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

2. **List all the employees who don't have a reporting manager.**

```
SQL>  select * from emp where mgr is null;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10

3. **Display name which has 8 characters in which 3rd character is A, 6th character is 'B' and last character is 'C';**

```
SQL> select * from emp where ename like '__A__B_C';
```

4. **List all the salesman in dept no 30 and having salary greater than 1500.**

```
SQL> SELECT * FROM EMP WHERE DEPTNO=30 AND SAL>1500;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30

5. **List all the employees whose name starts with 'S' or 'A'.**

```
SQL> SELECT * FROM EMP WHERE ENAME LIKE 'A%' OR ENAME LIKE 'S%';
```

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
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20

6. List all the employees except those who are working in dept 10 & 20.

```
SQL> SELECT * FROM EMP WHERE DEPTNO NOT IN(10,20);

     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
      7698 BLAKE      MANAGER          7839 01-MAY-81       2850                    30
      7844 TURNER     SALESMAN         7698 08-SEP-81       1500          0         30
      7900 JAMES      CLERK            7698 03-DEC-81        950                    30
```

7. List all the employees whose name does not start with 'S'.

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

     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
      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
```

8. WAQTD the emp who doesn't get salary but gets commission.

SQL> SELECT * FROM EMP WHERE SAL IS NULL AND COMM IS NOT NULL;

9. List all the employees whose commission is null and working as clerk.

```
SQL> SELECT * FROM EMP WHERE COMM IS NULL AND JOB='CLERK';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK            7902 17-DEC-80        800                    20
      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
```

10. WAQTD name and hiredate of all the employees except the employee hired in 2020.

SQL> SELECT ENAME, HIREDATE FROM EMP WHERE HIREDATE IN ('01-JAN-2020' , '31-DEC-2020');

11. List all the salesman in dept 30 with sal more than 2450.

SQL> SELECT * FROM EMP WHERE JOB='SALESMAN' AND DEPTNO=30 AND SAL>2450;

12. List all the analyst in dept number 20 and having salary greater than 2500.

```
SQL> SELECT * FROM EMP WHERE JOB='ANALYST' AND DEPTNO=20 AND SAL>2500;

     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
```

13. List all the employees whose name start with 'M' OR 'J'.

```
SQL> SELECT * FROM EMP WHERE ENAME LIKE 'M%' OR ENAME LIKE 'J%';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
```

14. List all the employees with annual salary except those who are working in dept 30.

```
SQL> SELECT EMP.*, SAL*12 ANNUAL_SAL FROM EMP
  2  WHERE DEPTNO<>30;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO ANNUAL_SAL
---------- ---------- --------- ---------- --------- ---------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20       9600
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20      35700
      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
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20      13200
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20      36000
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10      15600
```

15. List the employees whose name does not end with 'ES' or 'R'.

```
SQL> SELECT * FROM EMP WHERE ENAME NOT LIKE 'ES%' OR ENAME NOT LIKE 'R';
```

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

16. List the employees who are having reporting managers along with 10% hike in salary.

```
SQL> SELECT EMP.*, SAL+SAL*10/100 HIKED_SAL FROM EMP WHERE MGR IS NOT NULL;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	HIKED_SAL
7369	SMITH	CLERK	7902	17-DEC-80	800		20	880
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30	1760
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30	1375
7566	JONES	MANAGER	7839	02-APR-81	2975		20	3272.5
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	1375
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30	3135
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10	2695
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20	3300
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	1650
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20	1210
7900	JAMES	CLERK	7698	03-DEC-81	950		30	1045
7902	FORD	ANALYST	7566	03-DEC-81	3000		20	3300
7934	MILLER	CLERK	7782	23-JAN-82	1300		10	1430

17. Display all the employees who are salesman having 'E' as the second 'last character in ename salary having 4 characters.

```
SQL> SELECT * FROM EMP WHERE JOB='SALESMAN' AND ENAME LIKE '%E_' AND SAL LIKE '____';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30

18. Display all the employees who are joined after year 81.

```
SQL> SELECT * FROM EMP WHERE HIREDATE >'31-DEC-1981';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
```

19. Display all the employees who joined in feb.

```
SQL>  SELECT * FROM EMP WHERE HIREDATE LIKE '%FEB%';

     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
```

20. List the employees who are not working as managers and clerks in dept 10 and 20 with a salary in the range of 1000 to 3000.

```
SQL> SELECT * FROM EMP
  2  WHERE JOB IN('MANAGER','CLERK') AND
  3  DEPTNO IN(10,20) AND
  4  SAL BETWEEN 1000 AND 3000;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
```

21. List the employees whose salary not in the range of 1000 to 2000 and working in dept 10,20 or 30 except all salesman.

```
SQL> SELECT * FROM EMP WHERE JOB<>'SALESMAN' AND SAL NOT BETWEEN 1000 AND 2000 AND DEPTNO IN(10,20,3
0);

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      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
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
```

22. List the department names which are having letter 'O' in their location as well as their department names.

```
SQL> SELECT * FROM DEPT WHERE DNAME LIKE '%O%' AND LOC LIKE '%O%';

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        40 OPERATIONS     BOSTON
```

23. Display all the employees whose job has string 'MAN' in it.

```
SQL> SELECT * FROM EMP 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
```

24. List the employees who are hired after 82 and before 87.

```
SQL>  SELECT * FROM EMP WHERE HIREDATE >'31-DEC-1982' AND HIREDATE<'01-JAN-1987';
```

25. WAQTD all the details of employees hired in November and December .

```
SQL> SELECT * FROM EMP
  2  WHERE
  3  HIREDATE LIKE '%NOV%' OR HIREDATE LIKE '%DEC%';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL        COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK            7902 17-DEC-80        800                    20
      7839 KING       PRESIDENT             17-NOV-81       5000                    10
      7900 JAMES      CLERK            7698 03-DEC-81        950                    30
      7902 FORD       ANALYST          7566 03-DEC-81       3000                    20
```

26. List all the employees names and commission for those employees who earn commission more than their salary

```
SQL> SELECT ENAME, COMM FROM EMP
  2  WHERE
  3  SAL < COMM;

ENAME             COMM
---------- ----------
MARTIN            1400
```

27. WAQTD name and salary of all the employees if their annual salary ends with '0';

```
SQL> SELECT ENAME, SAL, SAL*12 ANNUAL_SAL FROM EMP
  2  WHERE
  3  SAL*12 LIKE '%0';

ENAME              SAL ANNUAL_SAL
---------- ---------- ----------
SMITH              800       9600
ALLEN             1600      19200
WARD              1250      15000
JONES             2975      35700
MARTIN            1250      15000
BLAKE             2850      34200
CLARK             2450      29400
SCOTT             3000      36000
KING              5000      60000
TURNER            1500      18000
ADAMS             1100      13200
JAMES              950      11400
FORD              3000      36000
MILLER            1300      15600
```

28. WAQTD name of the employee having at least 2 L's in his name.

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

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

29. WAQTD name of the employees who starts with starts with a vowel.

```
SQL> SELECT ENAME FROM EMP
  2  WHERE
  3  ENAME LIKE 'A%' OR ENAME LIKE 'E%' OR ENAME LIKE 'I%'
  4  OR ENAME LIKE 'O%' OR ENAME LIKE '%U';

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

Functions:

- Functions are the programs which contains set of instructions.
- Functions are used to perform some specific operation

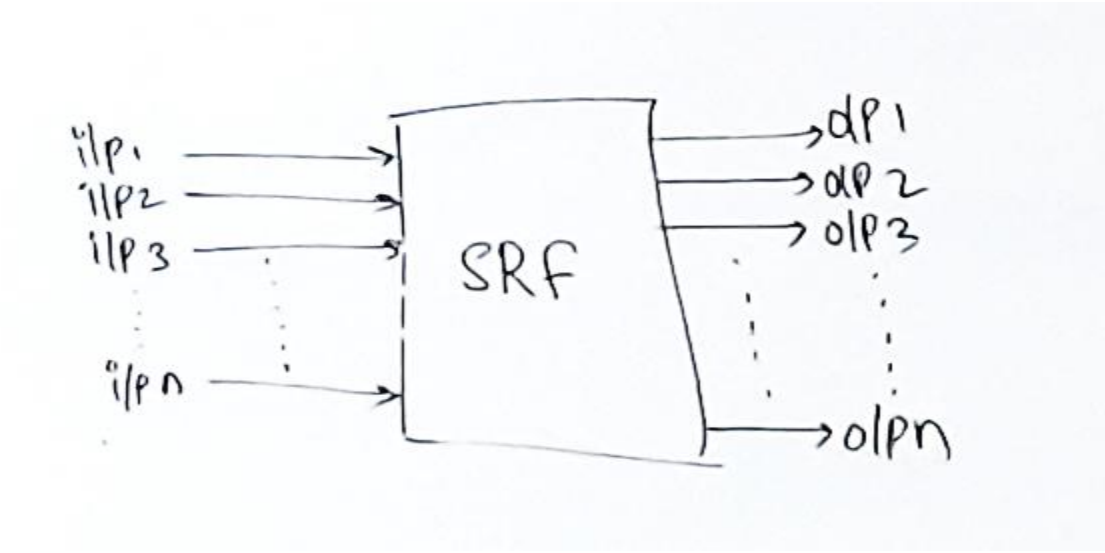
Types:

Single row function

Multi- row function

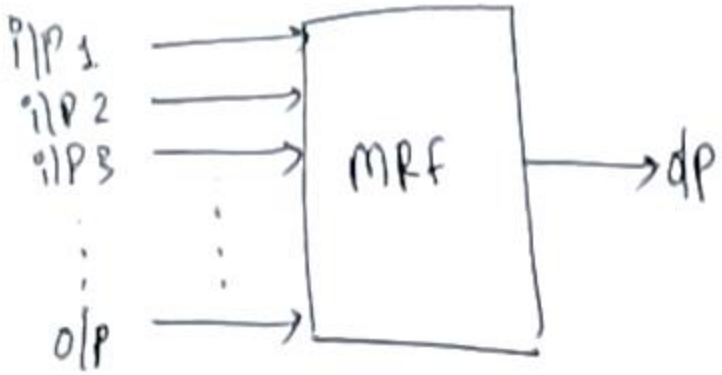
Single row function

- Single row function are those functions which accepts N number of inputs and generate N number of output.
- Single row functions accept first argument, executes set of instructions and generate the output.
- Then it goes to the second input, execute set of instructions and generate the output and the same process is repeated for all N number of inputs. Therefore, we can say single row function executes row by row.



Multi-row functions

- Multi-row functions are those functions which can accept multiple inputs but it can generate only one output
- Multi-row functions execute all the inputs in one shot and generate the single output
- Therefore, we can say multi-row function executes group by group.



SQL Functions

Single Row Functions

String		Number	General	Conversion	Date
Case_Manipulation	Char_manipulation				
Upper()	Length()	ceil()	NVL	TO_NUMBER()	SYSDATE
Lower()	Replace()	Floor()	NVL2()	TO_DATE()	ARITHMATIC WITH SYSDATE
InitCap()	Reverse()	Round()	DECODE()	TO_CHAR()	SYSTIMESTAMP
	Instr()	Trunc()	COALESCE		MONTHS_BETWEEN
	Substr()	ABS()	NULLIF		ADD_MONTHS
	Trim()	Power()			NEXT_DAY
	Ltrim(), Rtrim()	SQRT()			LAST_DAY
	Lpad()	mod()			
	Rpad()				
	Concat()				

Multi-row Functions / Group fun /Aggregate fun

SUM
MAX
MIN
AVG
COUNT

Multi-Row Functions

MAX:

MAX function will return highest value or largest character.

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

  MAX(SAL)
----------
      5000
SQL> SELECT MAX(HIREDATE) FROM EMP;

MAX(HIRED
---------
23-MAY-87

SQL> SELECT MAX(ENAME) FROM EMP;

MAX(ENAME)
----------
WARD
```

MIN:

MIN function will return lowest value of smallest character.

```
SQL> SELECT MIN(SAL), MIN(HIREDATE), MIN(ENAME) FROM EMP;


  MIN(SAL) MIN(HIRED MIN(ENAME)
---------- --------- ----------
       800 17-DEC-80 ADAMS
```

SUM:

SUM function will return summation of all the values of given column.

```
SQL> SELECT SUM(SAL) FROM EMP;


  SUM(SAL)
----------
     29025
```

COUNT

COUNT function will return number of records of number of rows of a column or table

COUNT function will not count null values

```
SQL> SELECT COUNT(DEPTNO) FROM EMP;

COUNT(DEPTNO)
-------------
          14



SQL> SELECT COUNT(COMM) FROM EMP;

COUNT(COMM)
-----------
          4

SQL> SELECT COUNT(*) FROM EMP;

  COUNT(*)
----------
        14
```

AVG

AVG function will return average value of given column

```
SQL> SELECT AVG(SAL) FROM EMP WHERE JOB='CLERK';


  AVG(SAL)
----------
    1037.5
```

1. **WAQTD number of employees getting salary less than 2000 in DEPTNO**

```
SQL> select count(*) from emp where sal<2000 and deptno=20;


  COUNT(*)

----------

         2
```

2. WAQTD total salary needed to pay employees working as clerk

```
SQL> select sum(sal) from emp where job='CLERK';

  SUM(SAL)
----------
      4150
```

3. WAQTD number of employees having ‘A’ as their first character

```
SQL> select count(*) from emp where ename like 'A%';

  COUNT(*)
----------
         2
```

4. WAQTD total salary needed to pay employee hired in feb

```
SQL> select sum(sal) from emp where hiredate LIKE '%FEB%';

  SUM(SAL)
----------
      2850
```

5. WAQTD number of employees reporting to 7839

```
SQL> SELECT COUNT(ENAME) FROM EMP WHERE MGR=7839;

COUNT(ENAME)
------------
           3
```

6. WAQTD number of employees having ‘A’ in their names

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

  COUNT(*)
----------
         7
```

7. WAQTD number of departments present in employee table

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

COUNT(DISTINCTDEPTNO)
---------------------
                    3
```

8. WAQTD number of employees earning commission

```
SQL> SELECT COUNT(*) FROM EMP WHERE COMM IS NOT NULL AND COMM > 0

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

9. WAQTD DISTINCT SALARY OF EMPLOYEES

```
SQL> SELECT COUNT(DISTINCT SAL) FROM EMP;

COUNT(DISTINCTSAL)
------------------
                12
```

10. WAQTD number of employees not earning commission

```
SQL> SELECT COUNT(*) FROM EMP WHERE COMM IS NULL OR COMM = 0;

  COUNT(*)
----------
        11
```

Single row functions:

1. **String**

- **Case manipulation function:**

Lower: It will return character of a string in small case letter.

Syntax: select lower(colname) from tblename;

Q: Display name of smith in small case letter.

```
SQL> select lower(ename) from emp where ename = 'SMITH';

LOWER(ENAM
----------
smith
```

Upper: It will return character of a string in capital letter.

```
SQL> select upper('qspiders') from dual;

UPPER('Q
--------
QSPIDERS
```

InitCap: It will return initial character capital and rest of the characters in small case in lower case.

```
SQL> select initcap(ename) from emp;

INITCAP(EN
----------
Smith
Allen
Ward
Jones
Martin
Blake
Clark
Scott
King
Turner
Adams
James
Ford
Miller

14 rows selected.
```

- **Character manipulation function:**

Length(): it will return number of characters of given string

```
SQL> select length(ename), ename from emp;


LENGTH(ENAME) ENAME
------------- ----------
            5 SMITH
            5 ALLEN
            4 WARD
            5 JONES
            6 MARTIN
            5 BLAKE
            5 CLARK
            5 SCOTT
            4 KING
            6 TURNER
            5 ADAMS
            5 JAMES
            4 FORD
            6 MILLER
```

Display name of employee whose name having exactly 4 characters

```
SQL> select ename from emp where length(ename) = 4;

ENAME
----------
WARD
KING
FORD
```

Reverse(): reverse function is used to reverse the characters of given string.

```
SQL> select reverse('abc') from dual;

REV
---
cba

SQL> select reverse('123') from dual;

REV
---
321
```

Replace (): **Replace** function is used to replace the characters of given string.

Syntax: Replace(String , char to replace, char to be replaced by)

Last parameter is optional by, default it will act as remove.

```
SQL> select replace('qsiders','s','$') from dual;

REPLACE
-------
q$ider$

SQL> select replace('qspiders','spider','*') from dual;

REP
---
q*s

SQL> select replace('qspiders',
  2  'qspiders', 'qsp') from dual;

REP
---
qsp

SQL>  select replace('qspiders','S','$') from dual;

REPLACE(
--------
qspiders

SQL>  select replace('qspiders','s') from dual;

REPLAC
------
qpider


SQL> SELECT REPLACE(REPLACE('QSPIDERS','I','!'),'E','@') FROM DUAL;

REPLACE(
--------
QSP!D@RS
```

Substr :

Substring is used to fetch substring out of the main string

Syntax: SUBSTR(STRING, POSITION, LENGTH)

Note:

negative number in start position indicates that searching of the character will goes from reverse direction (Right to Left)

Last parameter is optional, by default it will display the string till end.

```
SQL> select substr('qspiders',1,3) from dual;

SUB
---
qsp

SQL> select substr('qspiders',4,2) from dual;

SU
--
id

SQL> select substr('qspiders',2) from dual;

SUBSTR(
-------
spiders

SQL> select substr('qspiders',-3,2) from dual;

SU
--

SQL> select substr('qspiders',-1,2), substr('qspiders',-1,1), substr('qspiders',-1) from dual;

S S S
- - -
s s s

SQL> select substr('qspiders',0,2) from dual;

SU
--
qs

SQL> select substr('qspiders',3,-2) from dual;

S
```

Instr:

Instring is used to identify position of a character

Syntax: instr(string , char , start_pos , no_of_occurence)

Note:

- Last two parameters are optional
- By default start position and number of occurrences will be considered as 1 , 1.
- Negative number in start position indicates that searching of the characters will goes from reverse direction
- If start position is 0 then function will return 0 only.
- Number of occurrences should not be zero or negative numbers, otherwise function will return error.

Examples:

```
SQL> select instr('qspiders','s',1,1) from dual;

INSTR('QSPIDERS','S',1,1)
-------------------------
                        2

SQL> select instr('qspiders','s',1,2) from dual;

INSTR('QSPIDERS','S',1,2)
-------------------------
                        8

SQL> select instr('qspiders','s',1,3) from dual;

INSTR('QSPIDERS','S',1,3)
-------------------------
                        0

SQL> select instr('qspiders','s',2,1) from dual;

INSTR('QSPIDERS','S',2,1)
-------------------------
                        2

SQL> select instr('qspiders','s',3,2) from dual;

INSTR('QSPIDERS','S',3,2)
-------------------------
                        0

SQL> select instr('qspiders','s',-2,1) from dual;

INSTR('QSPIDERS','S',-2,1)
--------------------------
```

```
                     2

SQL> select instr('qspiders','s',-2,2) from dual;

INSTR('QSPIDERS','S',-2,2)
--------------------------
                     0

SQL> select instr('qspiders','s') from dual;

INSTR('QSPIDERS','S')
---------------------
                    2

SQL> select instr('qspiders','s',3) from dual;

INSTR('QSPIDERS','S',3)
-----------------------
                     8

SQL> select instr('qspiders','s',0,2) from dual;

INSTR('QSPIDERS','S',0,2)
-------------------------
                     0

SQL> select instr('qspiders','s',1,0) from dual;
select instr('qspiders','s',1,0) from dual
                                *
ERROR at line 1:
ORA-01428: argument '0' is out of range


SQL> select instr('qspiders','s',1,-1) from dual;
select instr('qspiders','s',1,-1) from dual
                                 *
ERROR at line 1:
ORA-01428: argument '-1' is out of range
```

Display name of emp where name stars with ‘A’
Substr:

```
SQL> SELECT ENAME FROM EMP
  2  WHERE SUBSTR(ENAME,1,1) = 'A';

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

Instr:

```
SQL>  SELECT ENAME
  2   FROM EMP WHERE INSTR(ENAME,'A') = 1;

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

Display name of emp whose name ends with ‘N’
Substr:

```
SQL> SELECT ENAME FROM EMP
  2  WHERE SUBSTR(ENAME,-1,1) = 'N';

ENAME
----------
ALLEN
MARTIN
```

Instr:

Display name of emp whose name contains ‘A’ at 2nd position
Substr:

```
SQL> SELECT ENAME FROM EMP
  2  WHERE SUBSTR(ENAME,2,1) = 'A';

ENAME
----------
WARD
MARTIN
JAMES
```

Instr:

```
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE INSTR(ENAME,'A') = 2;

ENAME
----------
WARD
MARTIN
JAMES
```

Display name of emp whose name contains 'I' at 2nd last position

SUBSTR:

```
SQL> SELECT ENAME FROM EMP
  2  WHERE SUBSTR(ENAME, -2, 1) = 'I';

ENAME
----------
MARTIN
```

Instr:

Display name of emp whose name contains 'A' anywhere in name

Substr:

Instr:

Concat:

```
SQL> select concat( concat(ename,'('),concat(job,')')) from emp;

CONCAT(CONCAT(ENAME,'
--------------------
SMITH(CLERK)
ALLEN(SALESMAN)
WARD(SALESMAN)
JONES(MANAGER)
MARTIN(SALESMAN)
BLAKE(MANAGER)
CLARK(MANAGER)
SCOTT(ANALYST)
KING(PRESIDENT)
TURNER(SALESMAN)
ADAMS(CLERK)
JAMES(CLERK)
FORD(ANALYST)
MILLER(CLERK)


SQL> select ename || '(' || job || ')' from emp;

ENAME||'('||JOB||')'
--------------------
SMITH(CLERK)
ALLEN(SALESMAN)
WARD(SALESMAN)
JONES(MANAGER)
MARTIN(SALESMAN)
BLAKE(MANAGER)
CLARK(MANAGER)
SCOTT(ANALYST)
KING(PRESIDENT)
TURNER(SALESMAN)
ADAMS(CLERK)
JAMES(CLERK)
FORD(ANALYST)
MILLER(CLERK)
```

Number Functions

1. **Ceil :**

Ceil function will return nearest highest value.

```
SQL> select ceil(-123.25) from dual;

CEIL(-123.25)
-------------
        -123

SQL> select ceil(12.00) from dual;

CEIL(12.00)
-----------
         12

SQL> select ceil(12.01) from dual;

CEIL(12.01)
-----------
         13

SQL> select ceil(12.99) from dual;

CEIL(12.99)
-----------
         13

SQL> select ceil(12.40) from dual;

CEIL(12.40)
-----------
         13
```

2. **Floor:**

Floor function will return nearest lowest value.

```
SQL> select floor(-123.45) from dual;

FLOOR(-123.45)
--------------
         -124
SQL> select floor(12.99) from dual;

FLOOR(12.99)
------------
          12

SQL> select floor(12.60) from dual;
```

```
FLOOR(12.60)
------------
          12
```

Round:

The ROUND function in SQL is used to round a numeric value to a specified number of decimal places.

```
SQL> select round(123.543,-1) from dual
  2  ;

ROUND(123.543,-1)
-----------------
            120

SQL>
SQL> select round(123.34) from dual;

ROUND(123.34)
-------------
          123

SQL> select round(123.345,2) from dual;

ROUND(123.345,2)
----------------
          123.35

SQL> select round(123.345,1) from dual;

ROUND(123.345,1)
----------------
           123.3

SQL> select round(123.543,0) from dual;

ROUND(123.543,0)
----------------
            124

SQL> select round(123.543,-1) from dual;

ROUND(123.543,-1)
-----------------
            120

SQL> select round(160.23,-2) from dual;

ROUND(160.23,-2)
----------------
            200
```

```
SQL> select round(160.23,-3) from dual;

ROUND(160.23,-3)
----------------
               0

SQL> select round(555.555,-3) from dual;

ROUND(555.555,-3)
-----------------
            1000
```

Truncate:

The TRUNCATE function in SQL is used to truncate a numeric value to a specified number of decimal places without rounding. This means that it simply cuts off the digits beyond the specified decimal places, rather than rounding them.

```
SQL> select trunc(555.555) from dual;

TRUNC(555.555)
--------------
           555

SQL> select trunc(555.555,2) from dual;

TRUNC(555.555,2)
----------------
          555.55

SQL> select trunc(555.555,-2) from dual;

TRUNC(555.555,-2)
-----------------
            500

SQL> select trunc(555.555,-3) from dual;

TRUNC(555.555,-3)
-----------------
               0
```

ABS:

The ABS function in SQL is used to return the absolute value of a numeric expression. The absolute value of a number is its non-negative value, regardless of whether the original number is positive or negative.

```
SQL> select abs(-10) from dual;

  ABS(-10)
----------
        10

SQL> select abs(010.0100) from dual;

ABS(010.0100)
-------------
        10.01
```

Sqrt

The SQRT function in SQL is used to calculate the square root of a numeric expression. The square root of a number is a value that, when multiplied by itself, gives the original number.

```
SQL> select sqrt(144) from dual;

 SQRT(144)
----------
        12

SQL> select mod(12,2) from dual;

 MOD(12,2)
----------
         0

SQL> select mod(12,2) from dual;

 MOD(12,2)
----------
         0

SQL>  select mod(13,2) from dual;

 MOD(13,2)
----------
         1
```

Power:

The POWER function in SQL is used to raise a numeric expression to a specified power. This function takes two arguments: the base number and the exponent to which the base number is raised.

```
SQL> select power(12,2) from  dual;

POWER(12,2)
-----------
        144

SQL> select power(5,3) from dual;

POWER(5,3)
----------
       125



SQL> select power(12,0) from dual;

POWER(12,0)
-----------
          1
```

General functions:

NVL

NVL Function is used substitute alternate value for null

Syntax: NVL(ARGS1 , ARGS2)

If value of first value is null then function will return second argument.

If value of first argument is not null then function will return value of first argument only.

Display total salary (sal + comm) of all the employees

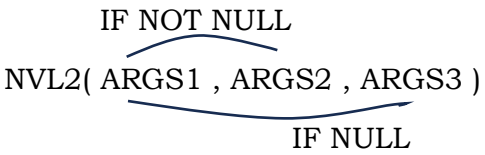
```
SQL> select ename, sal, comm , sal+nvl(comm,0) Total_Sal from emp;

ENAME             SAL       COMM  TOTAL_SAL
---------- ---------- ---------- ----------
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
```

NVL2

- NVL2 is enhancement over NVL function
- NVL2 is used to substitute alternate values for null also for not null.

  Syntax:



- If value of first argument is null then function will return value of a third argument.
- If value of first argument is not null then function will return value of second argument.

WAQTD total salary of all the employees after incrementing commission by rs 100 of all the employees

```
SQL> select ename, sal, comm, sal+nvl2(comm,comm+100,100) from emp;

ENAME             SAL       COMM SAL+NVL2(COMM,COMM+100,100)
---------- ---------- ---------- ----------------------------
SMITH             800                                   900
ALLEN            1600        300                       2000
WARD             1250        500                       1850
JONES            2975                                  3075
MARTIN           1250       1400                       2750
BLAKE            2850                                  2950
CLARK            2450                                  2550
SCOTT            3000                                  3100
KING             5000                                  5100
TURNER           1500          0                       1600
ADAMS            1100                                  1200
JAMES             950                                  1050
FORD             3000                                  3100
MILLER           1300                                  1400
```

Decode :

Decode function will acts like if else statement

Syntax:

Decode(col_name, search_value, Result,{default});

optional

- Decode function will compare search value against column values
- If equality exist function will return its respective result
- If equality does not exists then function will return default value if provided otherwise function will return null

WAQTD incremented salary of all the managers by rs 100 and analyst by rs 50.

```
SQL> select ename, job, sal,
  2  decode(job,'MANAGER',sal+100,'ANALYST',sal+50,sal) incre_sal
  3  from emp;

ENAME      JOB              SAL  INCRE_SAL
---------- --------- ---------- ----------
SMITH      CLERK             800        800
ALLEN      SALESMAN         1600       1600
WARD       SALESMAN         1250       1250
JONES      MANAGER          2975       3075
MARTIN     SALESMAN         1250       1250
BLAKE      MANAGER          2850       2950
CLARK      MANAGER          2450       2550
SCOTT      ANALYST          3000       3050
KING       PRESIDENT        5000       5000
TURNER     SALESMAN         1500       1500
ADAMS      CLERK            1100       1100
JAMES      CLERK             950        950
FORD       ANALYST          3000       3050
MILLER     CLERK            1300       1300
```

```
SQL> select ename, job, sal,
  2  decode(job,'MANAGER',sal+100,'ANALYST', sal+50) incre_sal
  3  from emp;

ENAME      JOB              SAL  INCRE_SAL
---------- --------- ---------- ----------
SMITH      CLERK             800
ALLEN      SALESMAN         1600
WARD       SALESMAN         1250
JONES      MANAGER          2975       3075
MARTIN     SALESMAN         1250
BLAKE      MANAGER          2850       2950
CLARK      MANAGER          2450       2550
SCOTT      ANALYST          3000       3050
KING       PRESIDENT        5000
TURNER     SALESMAN         1500
ADAMS      CLERK            1100
JAMES      CLERK             950
FORD       ANALYST          3000       3050
MILLER     CLERK            1300
```

Coalesce:

Coalesce is similar to NVL function

But to the coalesce function we can give a list of arguments and function will return first not null argument from given list.

Example:

```
SQL> select coalesce(null,null,'a','b') from dual;

C
-
a

SQL> select coalesce('a',null,'b') from dual;

C
-
a
```

Nullif:

Nullif function compares value of two arguments if both are equal then function will return null.

Otherwise function will return value of first arguments.

Example:

```
SQL> select nullif('sunday','monday') from dual;

NULLIF
------
sunday

SQL> select nullif('monday','monday') from dual;

NULLIF
------
```

Date functions

1. **SYSDATE**
   Sysdate will return current system date.

   We can perform arithmetic operations with sysdate i.e. we can add or subtract number of days of given sysdate.

   Example:

   SQL> select sysdate from dual;

   SYSDATE

   12-JUN-24

   SQL> select sysdate+1 from dual;

   SYSDATE+1

   13-JUN-24

2. **Systimestamp**

   It will return current system date, time in millisecond and timezone

   Example:

   SQL> select systimestamp from dual;

   SYSTIMESTAMP

   12-JUN-24 11.07.20.447000 AM +05:30

3. **Add_months:**

   It will add or subtract number of months from given date.

   Example:

   SQL> select add_months(sysdate,2) from dual;

   ADD_MONTH

   12-AUG-24

   SQL> select add_months('12-AUG-24',-1) FROM DUAL;

   ADD_MONTH

   12-JUL-24

4. **Month_between:**

   It will return number of months between two given dates

   Example:

   SQL> select months_between(sysdate,'12-APR-23') FROM DUAL;

   MONTHS_BETWEEN(SYSDATE,'12-APR-23')

   14
   SQL> select months_between(sysdate,'12-JUL-24') FROM DUAL;

   MONTHS_BETWEEN(SYSDATE,'12-JUL-24')

   -1

5. **Next_day:**

   It will return upcoming date of given day.

   SQL> SELECT NEXT_DAY(SYSDATE,'WED') FROM DUAL;

   NEXT_DAY(

   19-JUN-24

   SQL> SELECT NEXT_DAY(SYSDATE,'THU') FROM DUAL;

   NEXT_DAY(

   13-JUN-24

6. **Last_date:**

   It will return last day of given month.

   SQL> SELECT LAST_DAY(SYSDATE) FROM DUAL;

   LAST_DAY(

   30-JUN-24

   SQL> SELECT LAST_DAY('01-FEB-23') FROM DUAL;

   LAST_DAY(

   28-FEB-23

WAQTD HOW MANY DAYS LEFT TO COMPLETE THE MONTH.

```
SQL> SELECT LAST_DAY(SYSDATE)- SYSDATE FROM DUAL;

LAST_DAY(SYSDATE)-SYSDATE
-------------------------
                       18
```

WAQTD THE EXPERIENCE OF THE SMITH;

```
SQL> SELECT ename , trunc(MONTHS_BETWEEN(SYSDATE,HIREDATE) / 12)
  2   AS NO_OF_YRS_EXP FROM EMP;

ENAME      NO_OF_YRS_EXP
---------- -------------
SMITH                 43
ALLEN                 43
WARD                  43
JONES                 43
MARTIN                42
BLAKE                 43
CLARK                 43
SCOTT                 37
KING                  42
TURNER                42
ADAMS                 37
JAMES                 42
FORD                  42
MILLER                42
```

Conversion Functions

1. **To_number:**
   - To_number function converts character values into number.
   - If a string been converted non-numeric data then function will return non-numeric data.

   Example:

   SQL> select to_number('123') from dual;

   TO_NUMBER('123')

   123

2. **To_date:**

- To_date function takes character values as input and return a date which is same to date format.
- To_date function allows the user to enter a date in any format,but function will return a date which is similar to date datatype only.

Examples:

SQL> select to_date('31-december-2023') from dual;

TO_DATE('

31-DEC-23

SQL> select to_date('11-12-23','dd-mm-yy') from dual;

TO_DATE('

11-DEC-23

3. **To_char:**

   To_char function is used to covert a numeric or date input to character type with a different format.

Examples:

```
SQL> select to_char(to_date('01-may-24'),'dd-month-yyyy') from dual;

TO_CHAR(TO_DATE('
-----------------
01-may      -2024


SQL> select to_char(sal,'$9999.99') from emp;

TO_CHAR(S
---------
  $800.00
 $1600.00
 $1250.00
 $2975.00
 $1250.00
 $2850.00
 $2450.00
 $3000.00
 $5000.00
 $1500.00
 $1100.00
  $950.00
 $3000.00
 $1300.00


SQL> select to_char(sysdate,'dd-month-yyyy') from dual;

TO_CHAR(SYSDATE,'
-----------------
12-june     -2024
```

Abbriviation	**Description**
MM	Month of year 01,02,03,…12
MONTH	month in characters (i.e. january)
MON	JAN,FEB
DDD	Day of year in numbers (i.e.365)
DD	Day of the month in numbers (i.e. 28)
D	Day of the week in numbers (i.e. 7)
YEAR	Year in character
YY	Year with 2 numbers
YYYY	Year with 4 Numbers
WW	Week number of the year (i.e. 1)
W	Week number of month (i.e. 5)
SP	Spelled format
TH	ST,ND,RD,TH
HH	Hour number of the day (1-12)
HH24	Hour number of the day with 24 hours notation
AM,PM	AM OR PM
MI,SS	Number of minutes and seconds
DAY	Day of the week in character (i.e. Monday)
DY	Day of the week in short character description (i.e. SUN)

Test on functions

1. **List all the employees whose name contains only 4 characters**

SQL> select ename from emp where length(ename)=4;

ENAME

WARD

KING

FORD

2. **List all the employees whose job is having 7 characters**

SQL> select ename, job from emp where length(job)=7;

ENAME	JOB
JONES	MANAGER
BLAKE	MANAGER
CLARK	MANAGER
SCOTT	ANALYST
FORD	ANALYST

3. **Find out how many times letter ‘s’ occurs in qspiders**

SQL> select length('qspiders') - length(replace('qspiders','s')) from dual;

LENGTH('QSPIDERS')-LENGTH(REPLACE('QSPIDERS','S'))

2

4. **List all the employees whose job having last 3 characters as ‘man’.**

SQL> SELECT ENAME, JOB FROM EMP WHERE SUBSTR(JOB,-3) = 'MAN';

ENAME	JOB
ALLEN	SALESMAN
WARD	SALESMAN
MARTIN	SALESMAN
TURNER	SALESMAN

5. List all the employees whose job is having first 3 characters as ‘man’.

```
SQL> SELECT ENAME, JOB FROM EMP WHERE SUBSTR(JOB , 1 ,3) = 'MAN';

ENAME      JOB
---------- ---------
JONES      MANAGER
BLAKE      MANAGER
CLARK      MANAGER
```

6. List all the names whose name is having exactly 1 ‘L’.

```
SQL> SELECT ENAME FROM EMP
  2  WHERE LENGTH(ENAME) - LENGTH(REPLACE(ENAME, 'L')) = 1;

ENAME
----------
BLAKE
CLARK
```

7. Display all the names which are having letter ‘O’.

```
SQL> SELECT ENAME FROM EMP
  2  WHERE LENGTH(ENAME) - LENGTH(REPLACE(ENAME, 'O')) > 0;

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

8. Display the output as shown below.
Scott working as a clerk earns 3000 in dept 20.

```
SQL> SELECT ENAME || ' WORKING AS A ' || JOB || ' EARNS '
|| SAL || ' IN DEPT ' || DEPTNO FROM EMP WHERE ENAME = 'SMITH';

ENAME||'WORKINGASA'||JOB||'EARNS'||SAL||'INDEPT'||DEPTNO
--------------------------------------------------------------------------------
SMITH WORKING AS A CLERK EARNS 800 IN DEPT 20
```

9. Calculate number of L in string 'HELLLLL'.

```
SQL> SELECT LENGTH('HELLLLL') - LENGTH(REPLACE('HELLLLL', 'L')) FROM DUAL;


LENGTH('HELLLLL')-LENGTH(REPLACE('HELLLLL','L'))
------------------------------------------------
                                               5
```

10. Display all the employees whose job has a string 'man'.

```
SQL> SELECT ENAME, JOB FROM EMP
  2  WHERE LENGTH(JOB) - LENGTH(REPLACE(JOB,'MAN')) > 0;

ENAME      JOB
---------- ---------
ALLEN      SALESMAN
WARD       SALESMAN
JONES      MANAGER
MARTIN     SALESMAN
BLAKE      MANAGER
CLARK      MANAGER
TURNER     SALESMAN
```

11. Display all the employees whose job starts with string 'man'.

```
SQL> SELECT ENAME, JOB FROM EMP WHERE SUBSTR(JOB , 1 ,3) = 'MAN';

ENAME      JOB
---------- ---------
JONES      MANAGER
BLAKE      MANAGER
CLARK      MANAGER
```

12. Display all the employees whose job ends with string 'man'.

```
SQL> SELECT ENAME, JOB FROM EMP WHERE SUBSTR(JOB,-3) = 'MAN';

ENAME      JOB
---------- ---------
ALLEN      SALESMAN
WARD       SALESMAN
MARTIN     SALESMAN
TURNER     SALESMAN
```

13. Display first 3 characters of ename in lower case and rest everything in uppercase

```
SQL> SELECT CONCAT(LOWER(SUBSTR(ENAME,1,3)), UPPER(SUBSTR(ENAME,4))) FROM EMP;

CONCAT(LOW
----------
smiTH
allEN
warD
jonES
marTIN
blaKE
claRK
scoTT
kinG
turNER
adaMS
jamES
forD
milLER
```

14. Display the result from emp table as shown below

Smith is a clerk and gets salary 800.

```
SQL> SELECT CONCAT(CONCAT(CONCAT(CONCAT(ENAME , ' IS A ' ) , JOB),' AND GETS SALARY '),
SAL) FROM EMP WHERE ENAME = 'SMITH';

CONCAT(CONCAT(CONCAT(CONCAT(ENAME,'ISA'),JOB),'ANDGETSSALARY'),SAL)
--------------------------------------------------------------------------------
SMITH IS A CLERK AND GETS SALARY 800
```

15. List the employees hired on a Wednesday

SQL> SELECT ENAME, NEXT_DAY(HIREDATE,'WED') FROM EMP;

```
SQL> SELECT * FROM EMP WHERE TO_CHAR(HIREDATE,'DAY') = 'WEDNESDAY';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
```

16. List the employees hired on a leap year

```
SQL> SELECT * FROM EMP
  2  WHERE MOD(TO_CHAR(HIREDATE,'YYYY'),4) = 0;

    EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
--------- ---------- --------- ---------- --------- ---------- ---------- ----------
     7369 SMITH      CLERK          7902 17-DEC-80        800                    20
```

17. List the employees hired on Sunday in the month of may

```
SQL> SELECT * FROM EMP
  2  WHERE TO_CHAR(HIREDATE,'DY-MON')= 'SUN-MAY';
```

18. WAQTD enames of the employees who earns salary in multiples of 3.

```
SQL> select * from emp where mod(sal,3)=0;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
```

19. WAQTD details of the employee who have odd eid.

```
SQL> select * from emp where mod(empno,2) <> 0;

     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
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
```

20. WAQTD names of the employees if they have char 'A' present at least thrice in their names

```
SQL> select * from emp where instr(ename,'A',1,3) >0;

no rows selected
```

21. WAQTD name of the employees having 8 characters in the name and 3rd and 7th character is '_'.

```
SQL> select ename from emp
  2  where length(ename)=8 and
  3        substr(ename,3,1) = 1 and
  4        substr(ename,7,1) = 1;

no rows selected
```

22. Display 1st half of ename in lower case and 2nd half in reverse.

```
SQL> select concat(lower(substr(ename, 1 , length(ename)/2))
  2  ,reverse(substr(ename,-3,length(ename)/2))) from emp;

CONCAT(LOWER(
-------------
smTI
alEL
waRA
joEN
marNIT
blKA
clRA
scTO
kiNI
turREN
adMA
jaEM
foRO
milREL
```

23. Display last 3 char ename if name contains 'A' anywhere in ename.

24. Display name of emp whose name starts with vowels.

25. Display as 'CEO' in mgr column of emp who don't have reporting manager

```
SQL> SELECT ENAME,  decode(mgr, NULL, 'CEO', MGR) FROM EMP;

ENAME      DECODE(MGR,NULL,'CEO',MGR)
---------- ----------------------------------------
SMITH      7902
ALLEN      7698
WARD       7698
JONES      7839
MARTIN     7698
BLAKE      7839
CLARK      7839
SCOTT      7566
KING       CEO
TURNER     7698
ADAMS      7788
JAMES      7698
FORD       7566
MILLER     7782


SQL> SELECT DECODE(ENAME, (SELECT ENAME FROM EMP WHERE MGR IS NULL),'SAURABH',ENAME),
  2  NVL (TO_CHAR(MGR),'CEO') FROM EMP;

DECODE(ENA NVL(TO_CHAR(MGR),'CEO')
---------- ----------------------------------------
SMITH      7902
ALLEN      7698
WARD       7698
JONES      7839
MARTIN     7698
BLAKE      7839
CLARK      7839
SCOTT      7566
SAURABH    CEO
TURNER     7698
ADAMS      7788
JAMES      7698
FORD       7566
MILLER     7782
```

26. Display name of emp without 1st and last char of ename.

27. Display output as given below
"2 nd of oct"

```
SQL>  SELECT TO_CHAR(TO_DATE('02-OCT-2000'),'DTH " OF " MON') FROM DUAL;

TO_CHAR(TO_D
------------
2ND  OF  OCT
```

Statements

DDL	**Create, Alter, Drop, Truncate, Rename**
DML	**Insert, Update, Delete**
DQL	**Select**
TCL	**Commit, Rollback, Savepoint**
DCL	**Grant, Revoke**

DDL(Data Definition language):

- Data Definition Language actually consists of the SQL commands that can be used to define the database schema.
- It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.
- DDL is a set of SQL commands used to create, modify, and delete database structures but not data.
- These commands are normally not used by a general user, who should be accessing the database via an application.

1. **Create :**

   Create is used to create the structure of a table.

   Case 1: How to create table

   Syntax: **create table Table_name(Col_name1 datatype(size),**

   Col_name2 datatype(size), contraints,...);

   Create table Employee

   Empno-primary key

   Ename-not null

   Sal-check(sal>0)

   Contact_no-unique

   Mgr-fk

   :

```
SQL> create table employee
  2  (empno number(4) primary key, ename varchar(20) not null,
  3  sal number(6) check ( sal>0),
  4  contact_no number(10) check(length(contact_no) = 10) unique,
  5  mgr number(4) references employee(empno),
  6  city varchar(10) default('PUNE'));
```

EMPNO	ENAME	SAL	CONTACT	MGR	CITY
1	A	100	123	1	PUNE
2	B	100	124		PUNE
3	C	100	125	1	PUNE
4	D	100	126	2	PUNE

How to create duplicate table

Syntax:

Create table New_table_name as select */col_name from Old_table_name;

Ex: create table employee1 as select * from employee;

Note: In duplicate table(employee1) all columns, records and not null contraints will copy from original table (employee).

2. Drop:

Drop is used to delete all the records as well as structure of a table

Syntax: drop table tablename;

Ex: drop table student;

3. **Truncate**

Truncate is used to delete all the records without disturbing structure of a table

Syntax: truncate table tablename;

Ex: truncate table stud1;

Note:

After drop table will moved to recycle bin to display the date from the recycle bin we will use:

Syntax: show recyclebin; //to display tables of recyclebin.

- **To restore table from recycle bin:**

Syntax: flashback table table_name to before frop;

Ex: flashback table stud to before drop;

- **To delete table form recycle bin we will use purge**.

Case 01: to delete one table from recyclebin.

Syntax: purge table table_name;

Ex: purge table stud;

Case 02: to delete all tables from recyclebin.

Syntax: purge recyclebin;

Case 03: to purger table

Syntax: drop table table_name purge;

4. Alter:

Alter is used for Modifying the existing database objects.

Case 1: Add column.

Syntax: alter table tablename add colname datatype(size);

Ex: alter table stud add city varchar(10);

Case 2: Modify Column.

Syntax: alter table tablename modify colname datatype(size);

Ex: alter table stud modify city char(8);

Case 3: rename column

Syntax: alter table tableName rename column oldColName to NewColName;

Ex: alter table stud rename column city to addr;

Case 4: drop column

Syntax: alter table tableName drop column colName;

Ex: alter table stud drop column addr;

Case 5: Add constraints

Syntax: alter table tableName add constraint_name constraint_type(colName);

Ex: alter table stud add constraint pk primary key(id);

Case 6: Drop constraints

Syntax: alter table tableName drop constraint constraint_name;

Ex: alter table stud drop constraint pk;

Default:

alter table stud add city varchar(10) default('PUNE');

alter table stud modify city varchar(10) default(null);

Not Null:

alter table stud modify name varchar(10) not null;

alter table stud modify name null;

Foreign Key:

Alter table stud_marks add constraint fk foreign key(id) references stud(id);5.

5. Rename:

Rename is used to change the name of table

Syntax: rename old_tableName to new_tableName;

Ex: rename stud to student;

DML (Data Manipulation Language):

- DML (Data Manipulation Language) is a subset of SQL (Structured Query Language) used for inserting, updating, deleting, and merging data within a database.
- The primary DML statements are INSERT, UPDATE, DELETE.

1. Insert:

Insert in used to insert the record in a columns of a table

Case 1: Insert record in all columns

Syntax: insert into table_name values(val01,val02,…);

EX: insert into stud values(1,'A');

Insert into stud values(null,'B');

Case 2: insert record in selected columns

Syntax: insert into table_name (col_name1,col_name2,..) values(val01,val02,…);

Ex: insert into stud(id) values(3);

Case 3: copy data from one table to another.

Syntax: insert into target_table select * from source_table;

Ex: insert into stud1 select * from stud;

Stud1(target_table) and stud(source_table);

Insert into stud1(name) select name from stud;

Insert into stud1(name) select name from stud where id=2;

Case 4: insert data into multiple tables;

Syntax: insert all

into target_table1 values(val01,val02)

into target_table2 values(val01,val02)

select * from dual;

Ex: insert all

into stud values(4, 'A')

into student values(4, 'B')

select * from dual;

2. Update:

Update is used to update the records in a column of a table.

Case 1: Update all records

Syntax: update table_name set col_name = value;

Ex: update stud set city = 'PUNE';

Case 2: Update selected record.

Syntax: update table_name set colname = value where condition;

Ex: update stud set name = 'C' where id = 3;

3. Delete:

Delete is used to delete all the records or selected records without disturbing structure of a table.

Case 1: Delete all the records

Syntax: delete from table_name;

Ex: delete from stud;

Case 2: to delete selected records

Syntax: delete from table_name where consition;

Ex: delete from stud where id = 2;

ID	Name
1	A
2	B

To delete single cell from table;

Update stud

Set name = null

Where name = 'B';

Difference between drop, truncate and delete

Drop	Truncate	Delete
DDL	DDL	DML
Used to delete all the records also structure of the table	Used to delete all the records of the table without deleting its structure	Used to delete all/selected records without deleting structure of the table
Restore is possible using flashback	Restore is not possible	Restore is possible using rollback

DQL (Data query language) :

- DQL (Data Query Language) is a subset of SQL (Structured Query Language) used primarily for querying data from a database.
- The most common DQL statement is the SELECT statement, which retrieves data from one or more tables.

Syntax:

```
SELECT column1, column2, ...
FROM table_name
WHERE condition
GROUP BY column
HAVING condition
ORDER BY column;
```

TCL(Transaction control language) :

- TCL (Transaction Control Language) is a subset of SQL (Structured Query Language) used to manage transactions in a database.
- Transactions are a sequence of operations performed as a single logical unit of work, and TCL commands ensure the integrity and consistency of data during transactions.
- The primary TCL commands are COMMIT, ROLLBACK, and SAVEPOINT.

1. **COMMIT: Permanently saves all changes made in the current transaction.**

```
-- Begin transaction
START TRANSACTION;

-- Perform some DML operations
UPDATE employees SET salary = salary + 1000 WHERE department = 'Sales';
INSERT INTO employees (first_name, last_name, department, salary) VALUES ('Jane', 'Smith',
'Marketing', 60000);

-- Commit the transaction
COMMIT;
```

2. **ROLLBACK: Undoes all changes made in the current transaction.**

```
-- Begin transaction
START TRANSACTION;

-- Perform some DML operations
UPDATE employees SET salary = salary + 1000 WHERE department = 'Sales';
DELETE FROM employees WHERE first_name = 'Jane' AND last_name = 'Smith';

-- Rollback the transaction
ROLLBACK;
```

3. **SAVEPOINT: Sets a point within a transaction to which you can later roll back.**

```
-- Begin transaction
START TRANSACTION;

-- Perform some DML operations
INSERT INTO employees (first_name, last_name, department, salary) VALUES ('John', 'Doe', 'Sales',
50000);

-- Set a savepoint
```

```
SAVEPOINT savepoint1;

-- Perform more DML operations
UPDATE employees SET salary = salary + 1000 WHERE department = 'Sales';

-- Rollback to the savepoint
ROLLBACK TO SAVEPOINT savepoint1;

-- Commit the transaction
COMMIT;
```

DCL (Data Control language)

DCL (Data Control Language) is a subset of SQL (Structured Query Language) used to control access to data in a database. The primary DCL commands are GRANT and REVOKE.

1. **GRANT**: Gives specific privileges to users or roles.

```
-- Grant SELECT privilege on employees table to user john
GRANT SELECT ON employees TO john;

-- Grant INSERT and UPDATE privileges on employees table to role manager
GRANT INSERT, UPDATE ON employees TO manager;

-- Grant ALL privileges on employees table to user admin
GRANT ALL PRIVILEGES ON employees TO admin;
```

2. **REVOKE**: Removes specific privileges from users or roles.

```
-- Revoke SELECT privilege on employees table from user john
REVOKE SELECT ON employees FROM john;

-- Revoke INSERT and UPDATE privileges on employees table from role manager
REVOKE INSERT, UPDATE ON employees FROM manager;

-- Revoke ALL privileges on employees table from user admin
REVOKE ALL PRIVILEGES ON employees FROM admin;
```

Order by :

- **Order by** clause is used to display the record in ascending or descending order.
- By default order by clause will select ascending order to arrange the records.
- Order by clause will execute after select clause

Syntax: select */col_name from table_name

Order by col_name/col_number asc/desc;

Ex: select * from emp order by sal desc;

Select * from emp order by sal asc;

Select * from emp order by 5;

Order of execution

3.Select	5.Select
1.From	1.from
2.Where	2.where
4.Order by	3.group by
	4.Having
	6.Order by

List the number of employees whose job is salesman working for newyork and Chicago.

```
SQL> select ename, sal from emp
  2  order by hiredate;

ENAME                SAL
---------- ----------
SMITH                800
ALLEN               1600
WARD                1250
JONES               2975
BLAKE               2850
CLARK               2450
TURNER              1500
MARTIN              1250
KING                5000
JAMES                950
FORD                3000
MILLER              1300
SCOTT               3000
ADAMS               1100
```

```
SQL> select * from emp where job
  2   not in('SALESMAN','ANALYST') and sal > 1000 order by DEPTNO DESC;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
```

```
SQL> SELECT  EMP.* , SAL + SAL * 25 /100 HIKE
  2  FROM EMP
  3  WHERE SUBSTR(ENAME,1,1) NOT IN ('A','B')
  4  ORDER BY HIKE;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO       HIKE
---------- ---------- --------- ---------- --------- ---------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20       1000
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30     1187.5
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30     1562.5
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30     1562.5
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10       1625
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30       1875
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10     3062.5
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20    3718.75
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20       3750
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20       3750
      7839 KING       PRESIDENT            17-NOV-81       5000                    10       6250
```

Write a query to display

Sub Query

SubQuery is the query written inside another query.

Nested SubQuery:

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

Working principal of subquery:

- Inner query executes first and generates the output.
- The generated output of inner query is given as the input to outer query. Now outer query executes completely and generate the result of main query.
- Therefore, we can say that outer query depends on inner query.

Why to use subquery/ when to use

1. When we have unknown data in the query then we will go with the subquery.

- E.g.: Display details of employee who is working as smith manager.

```
Select *
From emp
Where empno =
(select mgr from emp where ename = 'SMITH');
```

2. When query is on one table and data needed to execute that query is present in another table than we will go with subquery.

- E.g.: Display location of smith.

```
Select loc
From dept
Where deptno =
(select deptno from emp where ename = 'SMITH');
```

Types of subquery:

1. **Single row subquery:**

   If subquery returns exactly one record then we will call it single row subquery.

   E.g.:

   Display dept name of smith.

```
Select dname
From dept
Where deptno =
(select deptno from emp where ename = 'SMITH');
```

2. Multi-row subquery

If a subquery return more than one record then we will can it multi row subquery.

E.g.:

Display dept name of employee whose employee name ends with 'R'

```
Select dname
From dept
Where deptno in
(select deptno from emp where ename like '%R');
```

1. WAQTD names of the employees earning more than adams.

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

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

11 rows selected.
```

2. WAQTD name and deptno of the employees if they are working in the same dept as jones.

```
SQL> SELECT ENAME , DEPTNO FROM EMP
  2  WHERE DEPTNO=(SELECT DEPTNO FROM EMP WHERE ENAME = 'JONES');

ENAME           DEPTNO
---------- ----------
SMITH               20
JONES               20
SCOTT               20
ADAMS               20
FORD                20
```

3. **WAQTD name and job of all the employees working in the same designation as james.**

```
SQL> SELECT ENAME,JOB FROM EMP
  2  WHERE JOB=(SELECT JOB FROM EMP WHERE ENAME ='JAMES');

ENAME      JOB
---------- ---------
SMITH      CLERK
ADAMS      CLERK
JAMES      CLERK
MILLER     CLERK
```

4. **WAQTD empno and ename along with annual salary of all the employees if their annual salary is greater than ward's annual salary.**

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

     EMPNO ENAME          SAL*12
---------- ---------- ----------
      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.
```

5. **WAQTD name and sal of the employees if they are hired before scott.**

```
SQL> SELECT ENAME, HIREDATE FROM EMP
  2  WHERE HIREDATE<(SELECT HIREDATE FROM EMP WHERE ENAME='SCOTT');

ENAME      HIREDATE
---------- ---------
SMITH      17-DEC-80
ALLEN      20-FEB-81
WARD       22-FEB-81
JONES      02-APR-81
MARTIN     28-SEP-81
BLAKE      01-MAY-81
CLARK      09-JUN-81
KING       17-NOV-81
TURNER     08-SEP-81
JAMES      03-DEC-81
FORD       03-DEC-81

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

6. **WAQTD name and sal of the employee if they are earning sal less than the employee whos empno us 7839.**

```
SQL> SELECT ENAME, SAL FROM EMP
  2  WHERE SAL<(SELECT SAL FROM EMP WHERE EMPNO=7839);

ENAME                SAL
---------- ----------
SMITH               800
ALLEN              1600
WARD               1250
JONES              2975
MARTIN             1250
BLAKE              2850
CLARK              2450
SCOTT              3000
TURNER             1500
ADAMS              1100
JAMES               950
FORD               3000
MILLER             1300

13 rows selected.
```

7. **WAQTD ename and salary of all the employees who are earning more than miller but less than allen.**

```
SQL> select ENAME, SAL FROM EMP
  2  WHERE SAL >
  3  (SELECT SAL FROM EMP WHERE ENAME='MILLER') AND
  4  SAL < (SELECT SAL FROM EMP WHERE ENAME='ALLEN');

ENAME                SAL
---------- ----------
TURNER             1500
```

8. **WAQTD all the details of the employees working in dept 20 and working in the same designation as smith.**

```
SQL> SELECT * FROM EMP WHERE DEPTNO=20
  2  AND
  3  JOB = (SELECT JOB FROM EMP WHERE ENAME ='SMITH');

     EMPNO ENAME      JOB              MGR HIREDATE          SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK            7902 17-DEC-80         800                    20
      7876 ADAMS      CLERK            7788 23-MAY-87        1100                    20
```

9. **WAQTD name and hiredate of the employees hired after 1980 and before king.**

```
SQL> SELECT ENAME, HIREDATE FROM EMP WHERE
  2  HIREDATE > '31-DEC-1980' AND
  3  HIREDATE < (SELECT HIREDATE FROM EMP 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
```

10. WAQTD name and comm if employees earn commission and work in the same designation as smith and less than king.

```
SQL> SELECT ENAME, COMM FROM EMP WHERE
  2  COMM IS NOT NULL AND
  3  JOB = (SELECT JOB FROM EMP WHERE ENAME = 'SMITH') AND
  4  COMM < (SELECT COMM FROM EMP WHERE ENAME = 'KING');
```

11. WAQTD dname of the employees whose name is smith.

```
SQL> SELECT DNAME FROM DEPT WHERE
  2  DEPTNO = (SELECT DEPTNO FROM EMP WHERE ENAME = 'SMITH');

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

12. **WAQTD dname and loc of the employee whose ename is king**.

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

DNAME          LOC
-------------- -------------
ACCOUNTING     NEW YORK
```

13. WAQTD loc of the emp whose employee number if 7902.

```
SQL> SELECT LOC FROM DEPT WHERE
  2  DEPTNO = (SELECT DEPTNO FROM EMP WHERE EMPNO = 7902);

LOC
-------------
DALLAS
```

14. WAQTD dname and loc along with deptno of the employees whose name ends with 'H'.

```
SQL>  SELECT DNAME, LOC, DEPTNO FROM DEPT WHERE
  2   DEPTNO = (SELECT DEPTNO FROM EMP WHERE ENAME LIKE '%H');

DNAME          LOC                DEPTNO
-------------- ------------- ----------
RESEARCH       DALLAS                 20
```

15. WAQTD dname of the employee whose designation is president.

```
SQL> SELECT DNAME FROM DEPT WHERE
  2  DEPTNO = (SELECT DEPTNO FROM EMP WHERE JOB = 'PRESIDENT');

DNAME
--------------
ACCOUNTING
```

16. WAQTD dname of the employee working in accounting department

```
SQL> SELECT * FROM EMP WHERE DEPTNO =
  2  (SELECT DEPTNO FROM DEPT WHERE DNAME = 'ACCOUNTING');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
```

17. WAQTD ename and salaries of the employees who are working in the location Chicago.

```
SQL> SELECT ENAME, SAL
  2  FROM EMP
  3  WHERE DEPTNO =
  4  (SELECT DEPTNO FROM DEPT
  5  WHERE LOC = 'CHICAGO');

ENAME                SAL
---------- ----------
ALLEN              1600
WARD               1250
MARTIN             1250
BLAKE              2850
TURNER             1500
JAMES               950
```

18. WAQTD details of the employees working in sales.

```
SQL>  SELECT *  FROM EMP WHERE
  2  DEPTNO = (SELECT DEPTNO FROM DEPT WHERE DNAME='SALES');

     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
      7698 BLAKE      MANAGER          7839 01-MAY-81       2850                    30
      7844 TURNER     SALESMAN         7698 08-SEP-81       1500          0         30
      7900 JAMES      CLERK            7698 03-DEC-81        950                    30
```

19. WAQTD details of the emp along with annual salary if employees are working in new York.

```
SQL> SELECT EMP.*, SAL*12 ANNUAL_SAL FROM EMP
  2  WHERE DEPTNO= (SELECT DEPTNO FROM DEPT WHERE LOC = 'NEW YORK');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL        COMM     DEPTNO ANNUAL_SAL
---------- ---------- --------- ---------- --------- ---------- ---------- ---------- ----------
      7782 CLARK      MANAGER          7839 09-JUN-81       2450                    10      29400
      7839 KING       PRESIDENT             17-NOV-81       5000                    10      60000
      7934 MILLER     CLERK            7782 23-JAN-82       1300                    10      15600
```

20. WAQTD names of the employees earning more than scott in accounting dept.

```
SQL> SELECT ENAME FROM EMP WHERE
  2  SAL >
  3  (SELECT SAL FROM EMP WHERE ENAME = 'SCOTT')
  4  AND
  5  DEPTNO =
  6  (SELECT DEPTNO FROM DEPT WHERE DNAME = 'ACCOUNTING');

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

21. WAQTD details of employees working as manager in the location of Chicago

```
SQL>  SELECT * FROM EMP
  2   WHERE JOB = 'MANAGER'
  3  AND
  4  DEPTNO =
  5  (SELECT DEPTNO FROM DEPT WHERE LOC = 'CHICAGO');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL        COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7698 BLAKE      MANAGER          7839 01-MAY-81       2850                    30
```

22. WAQTD name, sal, job , hiredate of the employees working in operations department and hired before king.

```
SQL>   SELECT ENAME, SAL, HIREDATE, JOB FROM EMP
  2   WHERE DEPTNO =
  3   (SELECT DEPTNO FROM DEPT WHERE DNAME ='OPERATIONS')
  4  AND
  5  HIREDATE < (SELECT HIREDATE FROM EMP WHERE ENAME ='KING');
```

23. WAQTD second max salary.

```
SQL> select * from emp
  2  where
  3  sal =
  4  (select max(sal) from emp where sal <
  5  (select max(sal) from emp));

     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
```

24. WAQTD name of the employee getting 3rd max salary.

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

ENAME               SAL
---------- ----------
JONES              2975
```

25. WAQTD smith's reporting manager's name.

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

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

26. WAQTD adams manager's manager name.

```
SQL> SELECT ENAME FROM EMP
  2  WHERE EMPNO = (SELECT MGR FROM EMP WHERE EMPNO = (SELECT MGR FROM EMP WHERE ENAME = 'ADAMS'));

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

27. WAQTD number of employees reporting to king.

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

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

28. WAQTD enames of the employees reporting to black's manager.

```
SQL> SELECT ENAME FROM EMP WHERE
  2  MGR = (SELECT MGR FROM EMP WHERE ENAME = 'BLAKE');

ENAME
----------
JONES
BLAKE
CLARK
```

Test on Sub-query

1. **Display all the employees whose department names ending ‘s’.**

```
SQL> SELECT * FROM EMP WHERE
  2  DEPTNO IN(SELECT DEPTNO FROM DEPT WHERE SUBSTR(DNAME,-1) = 'S');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7521 WARD       SALESMAN         7698 22-FEB-81       1250        500         30
      7844 TURNER     SALESMAN         7698 08-SEP-81       1500          0         30
      7499 ALLEN      SALESMAN         7698 20-FEB-81       1600        300         30
      7900 JAMES      CLERK            7698 03-DEC-81        950                    30
      7698 BLAKE      MANAGER          7839 01-MAY-81       2850                    30
      7654 MARTIN     SALESMAN         7698 28-SEP-81       1250       1400         30

6 rows selected.
```

2. **WAQTD the employees who is having maximum sal in dept name ‘accounting’.**

```
SQL> SELECT ENAME FROM EMP
  2  WHERE SAL = (SELECT MAX(SAL) FROM EMP WHERE DEPTNO =
  3  (SELECT DEPTNO FROM DEPT WHERE DNAME='ACCOUNTING'));

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

3. **WAQTD the dept name who is having highest commission.**

```
SQL> SELECT DNAME FROM DEPT WHERE
  2   DEPTNO = (SELECT DEPTNO FROM EMP WHERE COMM =
  3   (SELECT MAX(COMM) FROM EMP));

DNAME
--------------
SALES
```

4. **WAQTD the employee name whose dept name has 2nd character as ‘o’.**

```
SQL> SELECT ENAME FROM EMP
  2  WHERE DEPTNO =
  3  (SELECT DEPTNO FROM DEPT WHERE DNAME LIKE '%_O');

no rows selected
```

5. WAQTD all the employee whose dept number is same as scott

```
SQL> select * from emp
  2  where deptno =
  3  (select deptno from emp where ename = 'SCOTT');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
```

6. WAQTD all the employees in operations and accounting dept.

```
SQL> SELECT * FROM EMP WHERE
  2  DEPTNO = (SELECT DEPTNO FROM DEPT WHERE DNAME = 'OPERATIONS')
  3  OR
  4  DEPTNO =  (SELECT DEPTNO FROM DEPT WHERE DNAME = 'ACCOUNTING');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
```

7. List all the employees who has salary greater than miller.

```
SQL> SELECT * FROM EMP WHERE
  2  SAL > (SELECT SAL FROM EMP WHERE ENAME = 'MILLER');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      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
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20

8 rows selected.
```

8. WAQTD all the employee whose job not same as allen and salary is greater than martin.

```
SQL> SELECT * FROM EMP WHERE
  2  JOB <> (SELECT JOB FROM EMP WHERE ENAME ='ALLEN')
  3  AND SAL > (SELECT SAL FROM EMP WHERE ENAME = 'MARTIN');

     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
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

7 rows selected.
```

9. Display all the employees who is having location is same as adams manager.

```
SQL> select * from emp
  2  where deptno =
  3  (select deptno from dept where loc =
  4  (select loc from dept where deptno =
  5  (select deptno from emp where empno =
  6  (select mgr from emp where ename = 'ADAMS')
  7  )));

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
```

10. Display the job, manager number of employee who Is working for jones

```
SQL> SELECT JOB,MGR FROM EMP
  2  WHERE MGR =
  3  (SELECT EMPNO FROM EMP
  4  WHERE ENAME = 'JONES');

JOB              MGR
--------- ----------
ANALYST         7566
ANALYST         7566
```

11. Display employee details who are reporting to blake and have commission without using null or not null.

```
SQL> SELECT * FROM EMP
  2  WHERE MGR IN
  3  (SELECT EMPNO FROM EMP WHERE ENAME = 'BLAKE')
  4  AND
  5  COMM >= 0;

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

12. Display all the employee whose department is sales and who is earning some commission i.e commission is not null or zero and who is hired before the last person

```
SQL> SELECT * FROM EMP WHERE
  2  DEPTNO IN
  3  (SELECT DEPTNO FROM DEPT WHERE DNAME = 'SALES')
  4  AND
  5  COMM > 0
  6  AND HIREDATE < (SELECT MAX(HIREDATE) FROM EMP);

     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
```

13. Display dept names of employee whose salary is greater than average salary of all the clerks.

```
SQL> SELECT DNAME FROM DEPT
  2  WHERE DEPTNO IN
  3  (SELECT DEPTNO FROM EMP WHERE SAL >
  4  (SELECT AVG(SAL) FROM EMP WHERE JOB IN ('CLERK')));

DNAME
--------------
SALES
RESEARCH
ACCOUNTING
```

14. Display employee names who is having minimum salary in dept research.

```
SQL> SELECT * FROM EMP
  2  WHERE SAL =
  3  (SELECT MIN(SAL) FROM EMP WHERE DEPTNO =
  4  (SELECT DEPTNO FROM DEPT WHERE DNAME = 'RESEARCH'));

     EMPNO ENAME      JOB              MGR HIREDATE          SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
```

ANY and ALL operator.

ALL Operator:

All operator is a special operator which has to be used along with relational operator which will compare the value present at the LHS with all the value of RHS.

All operator returns true if all the values at the RHS have satisfied the condition.

Any Operator:

Any operator is a special operator which has to be used along with relational operator which will compare the value present at the LHS with all the value present of RHS.

Any operator returns true if one of the values at the RHS have satisfied the condition.

NOTE:

IN and NOT IN can be used as a substitute for = & != operator only, ‘

not for relational operators (>,< ,<=,>=).

Display the number of employees who work for research dept and their sal is lesser than one of the sal in dept 10.

```
SQL> SELECT COUNT(*) FROM EMP WHERE
  2  DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE DNAME = 'RESEARCH')
  3  AND
  4  SAL < ANY(SELECT SAL FROM EMP WHERE DEPTNO = 10);

  COUNT(*)
----------
         5
```

WAQTD name and salary of the employee earning more than allen and blake.

```
SQL> SELECT ENAME, SAL FROM EMP
  2  WHERE
  3  SAL > ALL(SELECT SAL FROM EMP WHERE ENAME IN ('ALLEN','BLAKE'));

ENAME             SAL
---------- ----------
JONES            2975
SCOTT            3000
KING             5000
FORD             3000
```

WAQTD name and salary of the employee earning more than the employee of dept 20.

```
SQL> SELECT ENAME, SAL FROM EMP
  2   WHERE SAL >
  3   ALL(SELECT SAL FROM EMP WHERE DEPTNO = 20 );

ENAME                SAL
---------- ----------
KING               5000
```

WAQTD name and salary of the employee if employee were hired before a clerk.

```
SQL> SELECT ENAME, SAL FROM EMP
  2  WHERE HIREDATE < ANY(SELECT HIREDATE FROM EMP WHERE JOB IN ('CLERK'));

ENAME                SAL
---------- ----------
SMITH                800
ALLEN               1600
WARD                1250
JONES               2975
BLAKE               2850
CLARK               2450
TURNER              1500
MARTIN              1250
KING                5000
JAMES                950
FORD                3000
MILLER              1300
SCOTT               3000

13 rows selected.
```

Group By

Note :

- In select clause we can write only multirow functions along with that we can write a column which is used with group by clause.
- Group by clause executes row by row.
- After group by clause, we will get group of similar data, therefore any clause executes after group by clause will execute group by group.

Assignment question on group by

1. **WAQTD number of employees working in each dept except president.**

```
SQL> select count(*) from emp
  2  where job <> 'PRESIDENT'
  3  group by deptno;

  COUNT(*)
----------
         6
         5
         2
```

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

```
SQL> select sum(sal), job from emp
  2  group by job;

  SUM(SAL) JOB
---------- ---------
      4150 CLERK
      5600 SALESMAN
      5000 PRESIDENT
      8275 MANAGER
      6000 ANALYST
```

3. **WAQTD number of employees working as manager in each dept.**

```
SQL> select count(*) , deptno from emp
  2  where job='MANAGER'
  3  group by deptno;

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

4. **WAQTD avg salary needed to pay all the employees in each department excluding the employees of deptno 20.**

```
SQL> select avg(sal), deptno from emp
  2  where deptno<>20
  3  group by deptno;

  AVG(SAL)     DEPTNO
---------- ----------
1566.66667         30
2916.66667         10
```

5. **WAQTD number of employees and avg salary in greater than 2000 in each dept.**

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

  COUNT(*)   AVG(SAL)
---------- ----------
         1       2850
         3 2991.66667
         2       3725
```

6. **WAQTD number of employees having character 'A' In their names in each job.**

```
SQL> select count(*)
  2  from emp where
  3  ename like '%A%'
  4  group by job;

  COUNT(*)
----------
         3
         2
         2
```

7. WAQTD total salary needed to pay and number of salesmen in each dept.

```
SQL> SELECT SUM(SAL) , COUNT(ENAME) FROM EMP
  2  WHERE JOB = 'SALESMAN'
  3  ORDER BY DEPTNO;

  SUM(SAL) COUNT(ENAME)
---------- ------------
      5600            4
```

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

```
SQL> select count(ename), max(sal) from emp
  2  group by job;

COUNT(ENAME)   MAX(SAL)
------------ ----------
           4       1300
           4       1600
           1       5000
           3       2975
           2       3000
```

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

```
SQL> select max(sal), deptno from emp
  2  group by deptno;

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

10. WAQTD number of times the salaries present in employee table.

```
SQL> select count(*), sal from emp
  2  group by sal;

  COUNT(*)        SAL
---------- ----------
         1       2450
         1       5000
         1       1300
         2       1250
         1       2850
         1       2975
         1       1100
         2       3000
         1        800
         1       1600
         1       1500

  COUNT(*)        SAL
---------- ----------
         1        950
```

Having clause:

- Having clause is used to filter the groups.
- All multi-row functions condition, we can write in having clause.
- Having clause executes after group by clause, therefore having clause will execute group by group.
- Having clause is similar to where clause by where clause filters the records and having clause filters the group.
- In having clause, we can also write multiple conditions by using logical operators.

Difference between where clause and having clause.

Where	**Having**
Where clause will not allow multi-row functions	Having clause will allow multi-row functions
Where clause executes row by row	Having clause executes group by group.
Where clause is used to filter records/rows.	Having clause is used to filter groups.
Where clause executes before group by clause	Having clause executes after where clause

1) EMP

ename	sal	deptno
A	100	10
B	200	20
C	300	20
D	100	10
E	300	30
F	600	10
G	700	30

2 AFTER WHERE

ename	sal	deptno
A	100	10
B	200	20
C	300	20
D	100	10
F	600	10

3) AFTER GROUP BY

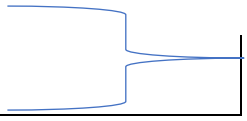
ename	sal	deptno
A	100	10
D	100	10
F	600	10
B	200	20
C	300	20

having
max(sal) > 400
600 > 400 ?

300>400 ?

4) AFTER HAVING BY

ename	sal	deptno
A	100	10
D	100	10
F	600	10



5) SELECT MAX(SAL)

600

RESULT

MAX(SAL)	DEPTNO
600	10

Questions on having

WAQTD the designation in which there are at least 2 employees present.

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

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

WAQTD names that are repeated exactly twice.

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

no rows selected
```

WAQTD number of employees working in each dept having at least 2 employees character 'A' AND 'S' on their names.

```
SQL> SELECT COUNT(*), DEPTNO FROM EMP
  2  WHERE ENAME LIKE '%A%' OR ENAME LIKE '%S%'
  3  GROUP BY DEPTNO
  4  HAVING COUNT(*) > = 2;

  COUNT(*)     DEPTNO
---------- ----------
         5         30
         4         20
```

WAQTD job and total salary of each job, if the total salary of each job is greater than 3450.

```
SQL> SELECT JOB , SUM(SAL) FROM EMP
  2  GROUP BY JOB
  3  HAVING SUM(SAL) > 3450.
  4  ;

JOB          SUM(SAL)
--------- ----------
CLERK            4150
SALESMAN         5600
PRESIDENT        5000
MANAGER          8275
ANALYST          6000
```

Questions on group by and having clause

1. **WAQTD total salary needed to pay each job in employee table.**

```
SQL> SELECT SUM(SAL) FROM EMP
  2   GROUP BY JOB;

  SUM(SAL)
----------
      4150
      5600
      5000
      8275
      6000
```

2. **WAQTD THE HIREDATE ON WHICH AT LEAST 3 EMPLOYEES WHERE HIRED.**

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

no rows selected
```

3. **WAQTD NUMBER OF EMPLOYEES WORKING IN EACH DEPT AND ITS AVERAGE SALARY EXCLUDING ALL THE EMPLOYEES WHOSE SALARY IS LESS THAN THEIR COMMISSION.**

```
SQL> SELECT COUNT(ENAME), AVG(SAL) FROM EMP
  2  WHERE SAL > COMM
  3  GROUP BY DEPTNO;

COUNT(ENAME)   AVG(SAL)
------------ ----------
           3       1450
```

4. **WAQTD THE DEPARTMENT NUMBER HAS MORE THAN 2 EMPLOYEES AND TOTAL AMOUNT REQUIRED TO PAY THE MONTHLY SALARY OF ALL THE EMPLOYEES IN THAT DEPT SHOULD BE MORE THAN 9000**

```
SQL> SELECT DEPTNO
  2  FROM EMP
  3  GROUP BY  DEPTNO
  4   HAVING COUNT(*) > 2 AND SUM(SAL) > 9000;

    DEPTNO
----------
        30
        20
```

5. **WAQTD THE SALARIES WHICH HAS REPEATATION IN THE SAL COLUMN OF EMPLOYEE TABLE.**

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

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

6. **WAQTD THE EMPLOYEE NAMES ONLY IF MORE THAN ONE PERSON IN THE EMPLOYEE OF THE COMPANY HAS SAME NAME.**

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

no rows selected
```

7. **WAQTD the department number whose average salary is between 2500 and 3000.**

```
SQL> SELECT DEPTNO
  2  FROM EMP
  3  GROUP BY DEPTNO
  4  HAVING AVG(SAL) BETWEEN 2500 AND 3000;

    DEPTNO
----------
        10
```

8. **WAQTD the number of employees only if they are working as manager or analyst and their annual salary should end with zero, in each dept.**

```
SQL>  SELECT COUNT(*) FROM EMP
  2   WHERE JOB IN ('MANAGER','ANALYST') AND SAL*12 LIKE '%0'
  3  GROUP BY DEPTNO;


  COUNT(*)
----------
         1
         3
         1
```

9. **WAQTD no of clerks working in each dept.**

```
SQL> SELECT COUNT(*) FROM EMP
  2  WHERE JOB = 'CLERK'
  3  GROUP BY DEPTNO;

  COUNT(*)
----------
         1
         2
         1
```

10. **WAQTD highest salary given to a manager in each dept.**

```
SQL> SELECT MAX(SAL)
  2  FROM EMP
  3  WHERE JOB='MANAGER'
  4  GROUP BY DEPTNO;


  MAX(SAL)
----------
      2850
```

2975

2450

11. WAQTD no of times the salaries have repeated in the emp table.

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

```
COUNT(SAL)
----------
         2
         2
```

12. WAQTD deptno and number of employes working in each dept except those working in dept 10

```
SQL>  SELECT COUNT(*), DEPTNO
  2   FROM EMP
  3  WHERE DEPTNO<> 10
  4  GROUP BY DEPTNO;

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

13.

```
SQL> SELECT COUNT(ENAME)
  2  FROM EMP
  3  WHERE COMM IS NOT NULL
  4  ORDER BY DEPTNO;
```

```
COUNT(ENAME)
------------
           4
```

14. WAQTD number of employees getting salary more than 1600 excluding all the managers in each dept.

```
SQL> SELECT COUNT(*)
  2  FROM EMP
  3  WHERE JOB <> 'MANAGER' AND SAL > 1600
  4  GROUP BY DEPTNO;


  COUNT(*)
----------
         2
         1
```

15)

```
SQL> SELECT AVG(SAL) FROM EMP
  2  WHERE MGR IS NOT NULL
  3  GROUP BY JOB;


  AVG(SAL)
----------
    1037.5
      1400
2758.33333
      3000
```

16)

```
SQL> SELECT COUNT(*), DEPTNO,TO_CHAR(HIREDATE,'DY')
  2  FROM EMP
  3  GROUP BY DEPTNO,TO_CHAR(HIREDATE,'DY')
  4  HAVING COUNT(*) > 1;
```

```
  COUNT(*)     DEPTNO TO_
---------- ---------- ---
         2         30 FRI
         2         10 TUE
         2         20 THU
```

17)

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

  COUNT(*)
----------
         2
         2
```

18)

```
SQL> SELECT MAX(SAL)
  2  FROM EMP
  3  WHERE ENAME NOT LIKE 'K%'
  4  GROUP BY JOB;

  MAX(SAL)
----------
      1300
      1600
      2975
      3000
```

19)

20)

21)

22)

```
SQL> select loc
  2  from dept d , emp e
  3  where d.deptno = e.deptno
  4  group by loc
  5  having count(ename) > 4;

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

23)

24)

25)

```
SQL>  select dname
  2   from dept
  3   where deptno =
  4   (select deptno from emp
  5   where job= 'SALESMAN'
  6  group by deptno
  7  having count(job) >= 3);

DNAME
--------------
SALES
```

Joins

The process of retrieving the data from multiple tables simultaneously, is known as joins

When to use joins?
If you want to select the columns from two different tables in one result table then we will go with joins.

Types of joins:
1. **Cartesian join/cross join**
2. **Inner join/equi join**
3. **Outer join**
    a. **Left outer join**
    b. **Right outer join**
    c. **Full outer join**
4. **Natural join**
5. **Self join**

1. Cross join

In cross join, record from table one will be merge with all the records of another table

EMP

ENAME	DEPTNO
A	20
B	30
C	10

Dept

DNAME	DEPTNO
D1	10
D2	20
D3	30

Number of columns in the result table : will be equivalent to the summation of columns present in the both the tables.

Number of col = Number of col T1 + Number of col T2

= 2 + 2

= 4 columns

Number of rows in the result table : will be equivalent to the product of number of rows present in the both tables.

Number of rows = Number of rows T1 x Number of rows T2

= 3 x 3

= 6 rows

Syntax: ANSI(AMERICAN NATIONAL STANDARD INSTITUTE)

Select */(col_name from table 1 cross join table 2);

Syntax: select */column_name from table_name1, table_name2;

Example:

WAQTD ename and dept name for all the employees.

Ans:

Select ename, dname from emp, dept;

Select ename, dname from emp cross join dept;

Result table

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

2. Inner join

Inner join is a type of a join where we are getting matched records from both tables by writing join condition.

Join Condition

It is the condition on which two tables are merged

E.g.: e.deptno = d.deptno

1. ANSI

Select */column_name from table_name1 inner join table_name2

On <join_condition>;

Select ename,dname from emp e inner join dept d on e.deptno = d.deptno;

2. Oracle

Select */column_name from table_name1, table_name2

Where <join_condition>

Select ename,dname from emp,dept where e.deptno = d.deptno;

Questions on joins:

1. **Name Of the employee and his location of all the employees.**

```
SQL> select ename,loc
  2  from emp,dept
  3  where emp.deptno = dept.deptno;

ENAME      LOC
---------- -------------
CLARK      NEW YORK
KING       NEW YORK
MILLER     NEW YORK
JONES      DALLAS
FORD       DALLAS
ADAMS      DALLAS
SMITH      DALLAS
SCOTT      DALLAS
WARD       CHICAGO
TURNER     CHICAGO
ALLEN      CHICAGO
JAMES      CHICAGO
BLAKE      CHICAGO
MARTIN     CHICAGO

14 rows selected.
```

2. **Write a query to display dname and salary for all the employees working in accounting.**

```
SQL> select dname, sal
  2  from emp e,dept d
  3   where e.deptno = d.deptno and dname='ACCOUNTING';

DNAME                 SAL
-------------- ----------
ACCOUNTING            2450
ACCOUNTING            5000
ACCOUNTING            1300
```

3. **Write a query to display dname and annual salary for all the employees. Whose salary is more than 2340**

```
SQL> SELECT ENAME, SAL*12 ANNULA_SAL
  2   FROM EMP E, DEPT D
  3  WHERE E.DEPTNO = D.DEPTNO AND SAL > 2340;

ENAME      ANNULA_SAL
---------- ----------
JONES           35700
BLAKE           34200
CLARK           29400
SCOTT           36000
KING            60000
FORD            36000

6 rows selected.
```

4. **Write a query to display ename and dname for all the employees having character 'A' in their dname.**

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

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

14 rows selected.
```

5. **Write a query to display ename and dname for all the employees working as salesman.**

```
SQL> SELECT ENAME,DNAME
  2   FROM EMP E, DEPT D
  3  WHERE E.DEPTNO = D.DEPTNO AND JOB = 'SALESMAN';

ENAME      DNAME
---------- --------------
ALLEN      SALES
TURNER     SALES
MARTIN     SALES
WARD       SALES
```

6. **WAQTD dname and job for all the employees whose job and dname starts with character 'S'.**

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

DNAME          JOB
-------------- ---------
SALES          SALESMAN
SALES          SALESMAN
SALES          SALESMAN
SALES          SALESMAN
```

7. **WAQTD dname and mgr no for employee reporting to 7839.**

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

DNAME                MGR
-------------- ----------
RESEARCH             7839
SALES                7839
ACCOUNTING           7839
```

8. **WAQTD dname and hiredate for employees hired after 83 into accounting or research dept.**

```
SQL> SELECT DNAME, HIREDATE
  2  FROM EMP E, DEPT D
  3  WHERE  E.DEPTNO = D.DEPTNO AND HIREDATE > '31-DEC-1983'
  4   AND DNAME IN ('ACCOUNTING','RESEARCH');

DNAME          HIREDATE
-------------- ---------
RESEARCH       23-MAY-87
RESEARCH       19-APR-87
```

9. WAQTD ename and dname of the employees who are getting comm in dept 10 or 30.

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

ENAME      DNAME
---------- --------------
ALLEN      SALES
WARD       SALES
MARTIN     SALES
TURNER     SALES
```

10. WAQTD dname and empno for all the employee whose empno are 7839,7902 and are working in loc new York.

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

DNAME                EMPNO
-------------- ----------
ACCOUNTING           7839
```

11. WAQTD loc and average salary given for each location by excluding all the employees whose second char is A in their name.

```
SQL>  SELECT LOC, AVG(SAL)
  2  FROM EMP E,DEPT D
  3  WHERE E.DEPTNO=D.DEPTNO
  4   AND ENAME NOT LIKE '_A%'
  5  GROUP BY LOC;

LOC            AVG(SAL)
-------------- ----------
NEW YORK       2916.66667
CHICAGO        1983.33333
DALLAS               2175
```

.

12. **WAQTD name of the emp and his loc if employee is working as manager and working under the employee whos empno is 7839**

```
SQL> SELECT ENAME, LOC
  2   FROM EMP E,DEPT D
  3   WHERE E.DEPTNO=D.DEPTNO
  4  AND JOB = 'MANAGER'
  5  AND MGR = 7839;

ENAME      LOC
---------- -------------
JONES      DALLAS
BLAKE      CHICAGO
CLARK      NEW YORK
```

13. **WAQTD dname and employee id of all the employees who are clerk and having reporting managers.**

```
SQL> SELECT DNAME, EMPNO
  2  FROM EMP E,DEPT D
  3  WHERE E.DEPTNO=D.DEPTNO
  4  AND JOB = 'CLERK'
  5  AND MGR IS NOT NULL;

DNAME                EMPNO
-------------- ----------
ACCOUNTING           7934
RESEARCH             7369
RESEARCH             7876
SALES                7900
```

14. **WAQTD dname and total salary given to that dept if there are atleat 4 employees working for each dept.**

```
SQL> SELECT DNAME, SUM(SAL)
  2  FROM EMP E,DEPT D
  3  WHERE E.DEPTNO=D.DEPTNO
  4  GROUP BY DNAME
  5  HAVING COUNT(ENAME) >=4;

DNAME            SUM(SAL)
-------------- ----------
RESEARCH            10875
SALES                9400
```

15. WAQTD dname and number of employees working in each dept only If there are manager or clerks.

```
SQL> SELECT DNAME, COUNT(*)
  2  FROM EMP E, DEPT D
  3  WHERE E.DEPTNO = D.DEPTNO
  4  AND JOB IN('MANAGER','CLERK')
  5  GROUP BY DNAME;

DNAME          COUNT(*)
-------------- ----------
ACCOUNTING              2
RESEARCH                3
SALES                   2
```

Outer Join

Outer join in used to obtain unmatched record from the tables.

Types of outer joins:

1. Left outer join
2. Right outer join
3. Full outer join

Emp

EMPNO	ENAME	DEPTNO
1	A	10
2	B	20
3	C	10
4	D	20
5	E	40

DEPT

DEPTNO	DNAME
10	ACCOUNTING
20	RESEARCH
30	SALES

1. **Left outer join**

   It is the type of a join where we will get matched record from both the tables and unmatched records from left table only that means, left join is used to obtain unmatched all record from left table along with matched record from both the tables.

ANSI syntax: select */col_name from table1 left join table2.

On table1.col_name = table2.col_name;

Example: select ename,dname

From emp e left join dept d

On e.deptno = d.deptno;

Oracle syntax: select * / col_name

From table1, table2

Where table1.col_name = table2.col_name (+);

Example: select ename,dname

From dept d, emp e

Where e.deptno = d.deptno(+);

Left Join

ENAME	DNAME
A	ACCOUNTING
B	RESEARCH
C	ACCOUNTING
D	RESEARCH

E	

Right outer Join

Right join is a type of join where we will get matched records from both tables and unmatched records from right table only, that means right join is used to obtain unmatched records from right table along with matched records of both tables.

ANSI syntax: select */col_name from table1 right join table2.

On table1.col_name = table2.col_name;

Example: select ename,dname

From emp e right join dept d

On e.deptno = d.deptno;

Oracle syntax: select * / col_name

From table1, table2

Where table1.col_name(+) = table2.col_name;

Example: select ename,dname

From dept d, emp e

Where e.deptno(+) = d.deptno;

Right join

ENAME	**DNAME**
A	ACCOUNTING
B	RESEARCH
C	ACCOUNTING
D	RESEARCH
	SALES

Full outer join

It is the type of join where we will get matched records of both table and unmatched records from left and right table, that means it is used to obtain unmatched records of both left and right table along with matched records.

ANSI syntax: select */col_name from table1 full join table2.

On table1.col_name = table2.col_name;

Example: select ename,dname

From emp e full join dept d

On e.deptno = d.deptno;

Full join

ENAME	DNAME
A	ACCOUNTING
B	RESEARCH
C	ACCOUNTING
D	RESEARCH
E	
	SALES

WAQTD names and dnames of all the employees even though the employees don't work in any dept.

```
SQL> select ename, dname
  2  from emp e left join dept d
  3  on e.deptno = d.deptno;

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

WAQTD names and dnames of all the employees even though the there are no employees in a dept.

```
SQL> select ename, dname
  2  from emp e right join dept d
  3  on e.deptno = d.deptno;

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

WAQTD names and dnames of all the employees and depts even though the employees don't work in any dept and a deptn having no employees.

```
SQL>   select ename , dname
  2  from emp e full join dept d
  3  on e.deptno = d.deptno;

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

15 rows selected.
```

Natural Join

- Natural join is a type of a join where we are getting matched records from both tables.
- Natural join is similar to inner join but in inner join to fetch common records we have to write join condition and in natural join we are getting matched records for that we are not writing any join condition
- Natural join will consider common column name from both tables to join two tables naturally.
- If there is no common column in two tables then the output will be like cross join.
- To call common column, we can't use **table_name.col_name**
- e.g. **emp.deptno** not allowded

Case 1

demo1

id	name	city
1	A	pune
2	B	mumbai
3	C	pune
4	D	nagpur

demo2

id	marks	city
1	50	pune
2	51	mumbai
3	52	pune
4	53	nagpur

select * from demo1 natural join demo2

id	city	name	marks
1	pune	A	50
2	mumbai	B	51
3	pune	C	52
4	nagpur	D	53

Case 2

demo1

id	name	city
1	A	pune
2	B	mumbai
3	C	pune
4	D	nagpur

demo2

id	marks	addr
1	50	pune
2	51	mumbai
3	52	pune
4	53	nagpur

select * from demo1 natural join demo2

id	name	city	marks	addr
1	A	pune	50	pune
2	B	mumbai	51	mumbai
3	C	pune	52	pune
4	D	nagpur	53	nagpur

Case 3

demo1

id	name	city
1	A	pune
2	B	mumbai
3	C	pune
4	D	nagpur

demo2

id	marks	city
1	50	pune
2	51	mumbai
3	52	pune
4	53	kolhapur

select * from demo1 natural join demo2

id	city	name	marks
1	pune	A	50
2	mumbai	B	51
3	pune	C	52

Self Join:

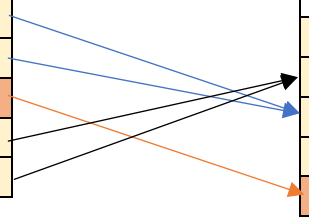
Joining a table by itself is known as self join.

Why ? / when ?

“ Whenever the data to select is in the same table but present in different records we use self join”.

E1 (EMP)

empno	ename	mgr
1	A	3
2	B	3
3	SMITH	5
4	SCOTT	2
5	FORD	2



E2 (MGR)

empno	ename	mgr
1	A	3
2	B	3
3	SMITH	5
4	SCOTT	2
5	FORD	2

Syntax:

Oracle:

```
Select column_name
from table_name t1 , table_name t2
where < join_condition >;
```

Example:

```
Select e1.ename “emp_Name”, e2.ename “mgr_Name”
From emp e1,emp e2
where e1.mgr = e2.empno;
```

ANSI:

```
Select column_name
From table_name1 t1 join Table_name2 t2
On < join_condition >;
```

Example:

```
Select e1.ename “emp_Name”, e2.ename “mgr_Name”
From emp1 join emp e2
On e1.mgr = e2.empno;
```

Emp_Name	Mgr_Name
A	SMITH
B	SMITH
SMITH	FORD

SCOTT	B
FORD	B

Assignment on self joins:

1. **WAQTD name of the employee and his manager's name if employee working as clerk.**

```
SQL> select e1.ename , e2.ename
  2  from emp e1 , emp e2
  3  where e1.mgr = e2.empno and
  4  e1.job = 'CLERK';

ENAME      ENAME
---------- ----------
JAMES      BLAKE
MILLER     CLARK
ADAMS      SCOTT
SMITH      FORD
```

2. **WAQTD name of the employee and manager's designation if manager works in dept 10 or 20.**

```
SQL>  select e1.ename , e2.job
  2   from emp e1 , emp e2
  3   where e1.mgr = e2.empno
  4  and e2.deptno in (10,20);

ENAME      JOB
---------- ---------
FORD       MANAGER
SCOTT      MANAGER
MILLER     MANAGER
ADAMS      ANALYST
BLAKE      PRESIDENT
JONES      PRESIDENT
CLARK      PRESIDENT
SMITH      ANALYST

8 rows selected.

SQL>
```

3. **WAQTD name of the employee and managers salary if employee and manager both earn more than 2300.**

```
SQL> select e1.ename , e2.sal
  2  from emp e1 , emp e2
  3  where e1.mgr = e2.empno
  4  and
  5  e1.sal > 2300 and e2.sal > 2300;

ENAME                SAL
---------- ----------
SCOTT              2975
FORD               2975
BLAKE              5000
JONES              5000
CLARK              5000
```

4. **WAQTD employee name and manager's hiredate if employee was hired before 1982.**

```
SQL> select e1.ename , e2.hiredate
  2  from emp e1 , emp e2
  3  where e1.mgr = e2.empno
  4  and
  5  e1.hiredate < '01-JAN-1982';

ENAME      HIREDATE
---------- ---------
SMITH      03-DEC-81
ALLEN      01-MAY-81
WARD       01-MAY-81
JONES      17-NOV-81
MARTIN     01-MAY-81
BLAKE      17-NOV-81
CLARK      17-NOV-81
TURNER     01-MAY-81
JAMES      01-MAY-81
FORD       02-APR-81

10 rows selected.
```

5. **WAQTD employee name and manager's comm if employee works as salesman and manager works in dept 30.**

```
SQL> select e1.ename , e2.comm
  2  from emp e1 , emp e2
  3  where e1.mgr = e2.empno
  4  and e1.job = 'SALESMAN'
  5  and e2.deptno = 30;

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

6. **WAQTD employee name and managers name and their salaries if employee earns more than manager.**

```
SQL> select e1.ename , e1.sal, e2.ename, e2.sal
  2   from emp e1 , emp e2
  3   where e1.mgr = e2.empno
  4  and e1.sal > e2.sal;

ENAME               SAL ENAME               SAL
---------- ---------- ---------- ----------
FORD              3000 JONES             2975
SCOTT             3000 JONES             2975
```

7. **WAQTD Employee name and hiredate, managers name and hiredate if manager was hired before employee.**

```
SQL>  select e1.ename, e1.hiredate, e2.ename, e2.hiredate
  2  from emp e1 , emp e2
  3  where e1.mgr = e2.empno
  4  and e2.hiredate < e1.hiredate;

ENAME      HIREDATE  ENAME      HIREDATE
---------- --------- ---------- ---------
FORD       03-DEC-81 JONES      02-APR-81
SCOTT      19-APR-87 JONES      02-APR-81
TURNER     08-SEP-81 BLAKE      01-MAY-81
JAMES      03-DEC-81 BLAKE      01-MAY-81
MARTIN     28-SEP-81 BLAKE      01-MAY-81
MILLER     23-JAN-82 CLARK      09-JUN-81
ADAMS      23-MAY-87 SCOTT      19-APR-87
```

8. **WAQTD employee name and managers name of both are working in same job.**

```
SQL> select e1.ename , e2.ename
  2   from emp e1 , emp e2
  3   where e1.mgr = e2.empno
  4  and e1.job = e2.job;

no rows selected
```

9. **WAQTD employee name and manager name if manager is working as manager.**

```
SQL> select e1.ename, e2.ename
  2  from emp e1 , emp e2
  3  where e1.mgr = e2.empno
  4  AND e2.job = 'MANAGER';

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

8 rows selected.
```

10. **WAQTD employee name and manager name along with their annual salaries if employee works in dept 10,20 and manager's sal is greater than employee salary.**

```
SQL> select e1.ename,  e1.sal*12 Annual_sal,
  2  e2.ename , e2.sal*12 Annual_sal
  3    from emp e1 , emp e2
  4    where e1.mgr = e2.empno
  5   and e1.deptno in (10,20)
  6   and e2.sal > e1.sal;

ENAME      ANNUAL_SAL ENAME      ANNUAL_SAL
---------- ---------- ---------- ----------
MILLER          15600 CLARK           29400
ADAMS           13200 SCOTT           36000
CLARK           29400 KING            60000
JONES           35700 KING            60000
SMITH            9600 FORD            36000
```

11. **WAQTD employee name and manager's designation for all the employees**.

```
SQL> select e1.ename , e2.job
  2  from emp e1 , emp e2
  3  where e1.mgr = e2.empno;

ENAME      JOB
---------- ---------
FORD       MANAGER
SCOTT      MANAGER
TURNER     MANAGER
ALLEN      MANAGER
WARD       MANAGER
JAMES      MANAGER
MARTIN     MANAGER
MILLER     MANAGER
ADAMS      ANALYST
BLAKE      PRESIDENT
JONES      PRESIDENT
CLARK      PRESIDENT
SMITH      ANALYST

13 rows selected.
```

12. WAQTD employee name and manager's salary for all the employees If manager's salary ends with 50.

```
SQL> select e1.ename , e2.sal
  2  from emp e1 , emp e2
  3  where e1.mgr = e2.empno
  4  and e2.sal like '%50';

ENAME                SAL
---------- ----------
TURNER             2850
ALLEN              2850
WARD               2850
JAMES              2850
MARTIN             2850
MILLER             2450

6 rows selected.
```

13. Display the number of employees who are getting salary less than the black's manager.

```
SQL>  select count(e3.ename)
  2   from emp e1, emp e2, emp e3
  3  where e1.mgr = e2.empno
  4  and e1.ename = 'BLAKE'
  5  and e3.sal < e2.sal;

COUNT(E3.ENAME)
---------------
             13
```

14. Display 2nd least salary from employee table.

15. List all the dept name and location of all the salesman manager's manager.

```
SQL>  select dname, loc
  2   from emp e1, emp e2, emp e3, dept d
  3   where e1.job = 'SALESMAN'
  4   AND e1.mgr = e2.empno
  5   and e2.mgr = e3.empno;

DNAME          LOC
-------------- -------------
ACCOUNTING     NEW YORK
RESEARCH       DALLAS
SALES          CHICAGO
OPERATIONS     BOSTON
ACCOUNTING     NEW YORK
RESEARCH       DALLAS
SALES          CHICAGO
OPERATIONS     BOSTON
ACCOUNTING     NEW YORK
RESEARCH       DALLAS
SALES          CHICAGO
OPERATIONS     BOSTON
ACCOUNTING     NEW YORK
RESEARCH       DALLAS
SALES          CHICAGO
OPERATIONS     BOSTON

16 rows selected.
```

1. **Display name and sal of emp working as Blake's manager.**

```
SQL>  select e2.ENAME,  e2.sal
  2  from emp e1, emp e2
  3  where e1.mgr = e2.empno
  4  and e1.ename = 'BLAKE';

ENAME                SAL
---------- ----------
KING               5000
```

2. **Display name of employee working under king.**

```
SQL> select e1.ename
  2  from emp e1, emp e2
  3  where e1.mgr = e2.empno
  4  and
  5  e2.ename ='KING';

ENAME
----------
BLAKE
JONES
CLARK
```

3. **Display ename and dname of emp who is working as ford's manager;**

```
SQL> select e2.ename, d.dname
  2  from emp e1 ,emp e2 , dept d
  3  where e1.mgr = e2.empno
  4  and
  5  e1.ename = 'FORD'
  6  AND e2.deptno = d.deptno;

ENAME      DNAME
---------- --------------
JONES      RESEARCH
```

4. **Display details of adam's manager and loc of manager's manager.**

```
SQL> select e2.* , loc
  2  from emp e1, emp e2 , emp e3 ,dept d
  3  where e1.ename = 'ADAMS'
  4  AND e1.mgr = e2.empno
  5  and e2.mgr = e3.empno
  6  and e3.deptno = d.deptno;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO LOC
---------- ---------- --------- ---------- --------- ---------- ---------- ---------- -------------
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20 DALLAS
```

5. Display name of emp as well as his manager's manager loc and sal.

```
SQL> select e1.ename , e3.sal , d.loc
  2  from emp e1,emp e2, emp e3, dept d
  3  where e1.mgr = e2.empno
  4  and e2.mgr = e3.empno
  5  and e3.deptno = d.deptno;

ENAME                SAL LOC
---------- ---------- -------------
SMITH              2975 DALLAS
ALLEN              5000 NEW YORK
WARD               5000 NEW YORK
MARTIN             5000 NEW YORK
SCOTT              5000 NEW YORK
TURNER             5000 NEW YORK
ADAMS              2975 DALLAS
JAMES              5000 NEW YORK
FORD               5000 NEW YORK
MILLER             5000 NEW YORK

10 rows selected.
```

6. Display details of emp who is working under smith's manager's manager.

```
SQL> select e4.*
  2  from emp e1, emp e2, emp e3, emp e4
  3  where e1.ename = 'SMITH'
  4  AND e1.mgr = e2.empno
  5  and e2.mgr = e3.empno
  6  and e3.empno = e4.mgr;

     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
```

7. Display name and loc of emp who is working as blake's manager.

```
SQL> select e2.ename, d.loc
  2  from emp e1,emp e2 , dept d
  3  where e1.mgr = e2.empno
  4  and e2.deptno = d.deptno
  5  and e1.ename = 'BLAKE';

ENAME      LOC
---------- -------------
KING       NEW YORK
```

8. Display details of employee who was hired before smith's manager was hired.

```
SQL> select e2.*
  2  from emp e1, emp e2, emp e3
  3  where e1.ename = 'SMITH'
  4  and e1.mgr = e3.empno
  5  and e2.hiredate < e3.hiredate;

     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
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30

9 rows selected.
```

9. Display ename of emp who are earning more than his manager.

```
SQL> select e1.ename
  2  from emp e1 , emp e2
  3  where e1.mgr = e2.empno
  4  and e1.sal > e2.sal;

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

10. Display name of manager whose employees are earning salary more than 30000 per year.

```
SQL> select distinct e2.ename
  2  from emp e1 , emp e2
  3  where e1.mgr = e2.empno
  4  and e1.sal*12 > 30000;

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

Mix joins questions

1. **List ename, job, annual salary, deptno, dname who earn 30000 per year and who are not clerks.**

```
SQL> select ename, job, sal* 12 Annual_sal , e.deptno
  2  from emp e, dept d
  3  where e.deptno = d.deptno
  4  AND sal*12 > 30000
  5  and e.job <> 'CLERK';

ENAME      JOB       ANNUAL_SAL     DEPTNO
---------- --------- ---------- ----------
JONES      MANAGER        35700         20
BLAKE      MANAGER        34200         30
SCOTT      ANALYST        36000         20
KING       PRESIDENT      60000         10
FORD       ANALYST        36000         20

SQL>
```

2. **List out all the employees by name and employee number along with their manager's name and employee number.**

```
SQL> select e1.ename,  e1.empno , e2.ename, e2.empno
  2   from emp e1, emp e2
  3   where e1.mgr = e2.empno;

ENAME           EMPNO ENAME           EMPNO
---------- ---------- ---------- ----------
FORD             7902 JONES            7566
SCOTT            7788 JONES            7566
TURNER           7844 BLAKE            7698
ALLEN            7499 BLAKE            7698
WARD             7521 BLAKE            7698
JAMES            7900 BLAKE            7698
MARTIN           7654 BLAKE            7698
MILLER           7934 CLARK            7782
ADAMS            7876 SCOTT            7788
BLAKE            7698 KING             7839
JONES            7566 KING             7839
CLARK            7782 KING             7839
SMITH            7369 FORD             7902

13 rows selected.
```

3. Ename, dname even if there no employee working in a particular dept.

```
SQL> select ename, dname
  2  from emp e right join dept d
  3  on d.deptno = e.deptno;

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

15 rows selected.
```

4. Display the department name along with total salary in each department.

```
SQL>  select dname , sum(sal)
  2   from emp e, dept d
  3   where e.deptno = d.deptno
  4   group by dname;

DNAME          SUM(SAL)
-------------- ----------
ACCOUNTING           8750
RESEARCH            10875
SALES                9400
```

5. **Display employee name and department name for each employee.**

```
SQL> select ename, dname
  2  from emp e, dept d
  3  where e.deptno = d.deptno;

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

14 rows selected.
```

6. **Display all the dept names irrespective of any employee working in it or not. If an employee is working display his name**

```
SQL> select ename, dname
  2  from emp e right join dept d
  3  on d.deptno = e.deptno;

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

15 rows selected.
```

7. **WAQTD employee name, job, dname, location of all employees who are working as actual manager and works at Chicago**

```
SQL> select ename, job, dname, loc
  2  from emp natural join dept
  3  where job in ('MANAGER')
  4  AND LOC = 'CHICAGO';

ENAME      JOB       DNAME          LOC
---------- --------- -------------- -------------
BLAKE      MANAGER   SALES          CHICAGO
```

8. **List the departments names in which employees hired between 1st jan 1981 and 31st dec 1982 with sal more than 1800.**

```
SQL> select DISTINCT dname
  2  from emp natural join dept
  3  where hiredate between '01-JAN-1981' AND '31-DEC-1982'
  4   AND SAL > 1800;

DNAME
--------------
ACCOUNTING
RESEARCH
SALES
```

9. **Display 2nd least salary from employee table.**

```
SQL> select min(e2.sal) from emp e1 , emp e2
  2  where e1.sal < e2.sal;

MIN(E2.SAL)
-----------
        950
```

10. **Display dept name of the employee who earn min salary and have no reporting manager.**

```
SQL> select dname
  2  from emp natural join dept
  3  where sal = (select min(sal) from emp)
  4  and mgr is null;

no rows selected
```

11. Display dept name , loc of all the employees who are reporting to smith.

```
SQL>  select dname, loc
  2   from emp e1 , emp e2 , dept d
  3   where e2.ename = 'SMITH'
  4   AND e1.mgr = e2.empno
  5  and e1.deptno = d.deptno;

no rows selected
```

12. List all the dept name and loc of all the salesman manager's manager.

```
SQL> select distinct dname, loc
  2  from emp e1, emp e2, emp e3, dept d
  3  where e1.job = 'SALESMAN'
  4  AND e1.mgr = e2.empno
  5  and e2.mgr = e3.empno
  6  and e3.deptno = d.deptno;

DNAME          LOC
-------------- -------------
ACCOUNTING     NEW YORK
```

13. List all the employees who are working in research dept and they are manager.

```
SQL> select e.*
  2  from emp e , dept d
  3  where e.deptno = d.deptno
  4  and e.job = 'MANAGER'
  5  AND e.deptno = (select deptno from dept where dname = 'RESEARCH');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
```

14. Display the number of employees who are getting salary less than the blake's manager.

```
SQL> select count(e1.ename)
  2  from emp e1 , emp e2, emp e3
  3  where e2.ename = 'BLAKE'
  4  and e2.mgr = e3.empno
  5  and e1.sal < e3.sal;

COUNT(E1.ENAME)
---------------
             13
```

15. List the employee deptname and loc of all the employees who are analyst , reporting to blake

```
SQL>  select dname, loc
  2   from dept d, emp e1, emp e2
  3   where e2.ename = 'BLAKE'
  4   AND e1.job = 'ANALYST'
  5   AND e1.deptno = d.deptno;

DNAME          LOC
-------------- -------------
RESEARCH       DALLAS
RESEARCH       DALLAS
```

16. **Display the employee names, hiredate , comm of ford's manager.**

```
SQL> select e2.ename, e2.hiredate, e2.comm
  2  from emp e1, emp e2
  3  where e1.ename = 'FORD'
  4  and e1.mgr = e2.empno;

ENAME      HIREDATE         COMM
---------- --------- ----------
JONES      02-APR-81
```

17. Display ename, dname of all the employees whose salary less than avg sal of dept 30

```
SQL> select ename, dname
  2  from  emp e, dept d
  3  where e.deptno = d.deptno
  4  and  sal < (select avg(sal) from emp where deptno = 30);

ENAME      DNAME
---------- --------------
SMITH      RESEARCH
WARD       SALES
MARTIN     SALES
TURNER     SALES
ADAMS      RESEARCH
JAMES      SALES
MILLER     ACCOUNTING

7 rows selected.
```

18. Display ename, dname of the employees whose name starts with S.

```
SQL> select ename, dname
  2  from emp e natural join dept d
  3  where ename like 'S%';

ENAME      DNAME
---------- --------------
SMITH      RESEARCH
SCOTT      RESEARCH
```

19. Display ename, dname of all the employees who are working for jones

```
SQL> select e1.ename , dname , loc
  2  from emp e1, emp e2 , dept d
  3  where e2.ename = 'JONES'
  4  AND e1.mgr = e2.empno
  5  and e1.deptno = d.deptno;

ENAME      DNAME          LOC
---------- -------------- -------------
FORD       RESEARCH       DALLAS
SCOTT      RESEARCH       DALLAS
```

20. List ename who are not having any employee in it.

```
SQL> SELECT d.dname
  2  FROM dept d
  3  LEFT JOIN emp e ON d.deptno = e.deptno
  4  WHERE e.empno IS NULL;

DNAME
--------------
OPERATIONS
```

21. Display employee name along with their manager name.

```
SQL> select e1.ename, e2.ename
  2  from emp e1, emp e2
  3  where e1.mgr = e2.empno;

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

13 rows selected.
```

22. Display location name of the employee who earn commission

```
SQL> select distinct loc
  2  from emp e natural join dept d
  3  where e.comm is not null;

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

23. Display dept name of the employee who earn min salary and have no reporting manager.

```
SQL> select dname
  2  from emp natural join dept
  3  where sal = (select min(sal) from emp)
  4  and mgr is null;

no rows selected
```

rownum

```
SQL> select *
  2  from
  3  (select * from emp order by rownum desc)
  4  where rownum <=3;

    EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
     7934 MILLER     CLERK          7782 23-JAN-82       1300                  10
     7902 FORD       ANALYST        7566 03-DEC-81       3000                  20
     7900 JAMES      CLERK          7698 03-DEC-81        950                  30

SQL> select * from emp 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

SQL> select * from emp where rownum = 1;

    EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
     7369 SMITH      CLERK          7902 17-DEC-80        800                  20
SQL> select * from (select * from emp order by rownum desc)
  2  where rownum = 1;

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

```
SQL> select sal
  2  from (select rownum as sal_no, sal
  3        from (select distinct sal from emp order by sal desc))
  4  where sal_no = 4;

       SAL
----------
      2850
```

```
select sal
from (select rownum as sal_no, sal
from (select distinct sal from emp order by sal desc))
where sal_no >= 2 and sal_no <=5;

       SAL
----------
      3000
      2975
      2850
      2450
```

Normalization is database design technique .it is organized in a such a way that it reducesredundancy and dependency.

Normalization is the process of splitting the bigger table into many small tables withoutchanging its functionality.

Advantages

1) it reduces the redundancy (unnecessary repeatation of data)
2) avoids problem due to delete anamoly (inconsistency)

Normalization is a step-by-step process and in each step, we have to perform some activities.

STEPS IN NORMALIZATION

1) 1NF – 1st Normal form
2) 2NF – 2nd Normal form
3) 3NF – 3rd Normal form

1st Normal Form (1NF)

In this Normal Form, we tackle the problem of atomicity. Here atomicity means values in thetable should not be further divided. In simple terms, a single cell cannot hold multiple values

Employee	Age	Department
Melvin	32	Marketing, Sales
Edward	45	Quality Assurance
Alex	36	Human Resource

:

Employee	Age	Department
Melvin	32	Marketing
Melvin	32	Sales
Edward	45	Quality Assurance
Alex	36	Human Resource

Employee ID	Employee Name	Phone Number	Salary
1EDU001	Alex	+91 8553206126 +91 9449424949	60,131
1EDU002	Barry	+91 8762989672	48,302
1EDU003	Clair	+91 9916255225	22,900
1EDU004	David	+91 6363625811 +91 8762055007	81,538

In the above table, we can clearly see that the Phone Number column has two values. Thus it violated the 1st NF. Now if we apply the 1st NF to the above table we get the below table as the result.

Employee ID	Employee Name	Phone Number	Salary
1EDU001	Alex	+91 8553206126	60,131
1EDU001	Alex	+91 9449424949	60,131
1EDU002	Barry	+91 8762989672	48,302
1EDU003	Clair	+91 9916255225	22,900
1EDU004	David	+91 6363625811	81,538
1EDU004	David	+91 8762055007	81,538

2NF (Second Normal Form) Rules

Rule 1- Be in 1NF

- Rule 2- Single Column Primary Key

The first condition in the 2nd NF is that the table has to be in 1st NF.The table also should not contain partial dependency.

Example:

Sample Products table:

productID	Product	Brand

1	Monitor	Apple
2	Monitor	Samsung
3	Scanner	HP
4	Head phone	JBL

Product table following 2NF:

Products Category table:

brandID	Brand
1	Apple
2	Samsung
3	HP
4	JBL

Brand table:

productID	Product
11	Monitor
22	Scanner
33	Head phone

3rd Normal Form (3NF)

The same rule applies as before i.e, the table has to be in 2NF before proceeding to 3NFThe other condition is there should be no transitive dependency.

It is also used to achieve the data integrity.

Student Id	Student Name	Subject Id	Subject	Address
1DT15ENG01	Alex	15CS11	SQL	Goa
1DT15ENG02	Barry	15CS13	JAVA	Bengaluru
1DT15ENG03	Clair	15CS12	C++	Delhi
1DT15ENG04	David	15CS13	JAVA	Kochi

In the above table, Student IDdetermines Subject ID, and Subject ID determines Subject. Therefore, Student ID determines Subject via Subject ID**.** This implies that we have a transitive functional dependency, and this structure doesnot satisfy the third normal form.

Now if we apply the 3rd NF to the above table we get the below table as the result.

Student Id	Student Name	Subject Id	Address
1DT15ENG01	Alex	15CS11	Goa
1DT15ENG02	Barry	15CS13	Bengaluru
1DT15ENG03	Clair	15CS12	Delhi
1DT15ENG04	David	15CS13	Kochi

Subject Id	Subject
15CS11	SQL
15CS13	JAVA
15CS12	C++

Example:02

EMP_ID	EMP_NAME	EMP_ZIP	EMP_STATE	EMP_CITY
222	Harry	201010	UP	Noida
333	Stephan	02228	US	Boston

444	Lan	60007	US	Chicago
555	Katharine	06389	UK	Norwich
666	John	462007	MP	Bhopal

EMPLOYEE_DETAIL table:

Here, EMP_STATE & EMP_CITY dependent on EMP_ZIP and EMP_ZIP dependenton EMP_ID. The non-prime attributes (EMP_STATE, EMP_CITY) transitively

dependent on EMP_ID. It violates the rule of third normal form.

That's why we need to move the EMP_CITY and EMP_STATE to the new

<EMPLOYEE_ZIP> table, with EMP_ZIP as a Primary key.

EMPLOYEE table:

EMP_ID	EMP_NAME	EMP_ZIP
222	Harry	201010
333	Stephan	02228
444	Lan	60007
555	Katharine	06389
666	John	462007

EMPLOYEE_ZIP table:

EMP_ZIP	EMP_STATE	EMP_CITY
201010	UP	Noida

02228	US	Boston
60007	US	Chicago
06389	UK	Norwich
462007	MP	Bhopal

Views:

- A view is a representation of another table or combination of tables or views
- A view contains no data itself.
- A view derives its data from the tables on which it is based. These tables are called base tables.
- Base tables might in turn be actual tables or might be views themselves.
- All operations performed on a view actually affect the base table of the view.
- You can use views in almost the same way as tables. You can query, update, insert into, and delete from views, just as you can standard tables.

Creating Views:

To create a view in your schema, you must have the CREATE VIEW privilege.

Step 1: Login as Username :System ,Password :tiger

Step 2: grant all privilege to scott;

Step 3: Login as username : Scott, password :tiger

SQL> create view v1 as select ename,deptno from emp;

View created.

SQL> select * from v1;

ENAME	DEPTNO
SMITH	20
ALLEN	30
WARD	30
JONES	20
MARTIN	30
BLAKE	30
CLARK	10
SCOTT	20
KING	10
TURNER	30
ADAMS	20

SQL> create view v2 as select ename ,dname from emp natural join dept;

View created.

```
SQL> select * from v2;
ENAME      DNAME
---------- --------------
SMITH      RESEARCH
ALLEN      SALES
WARD       SALES
JONES      RESEARCH
MARTIN     SALES
BLAKE      SALES
CLARK      ACCOUNTING
SCOTT      RESEARCH

SQL> create view v3 as select ename,loc from v1 natural join dept;
View created.

SQL> select * from v3;
ENAME      LOC
---------- -------------
SMITH      DALLAS
ALLEN      CHICAGO
WARD       CHICAGO
JONES      DALLAS
MARTIN     CHICAGO
BLAKE      CHICAGO
CLARK      NEW YORK
SCOTT      DALLAS
KING       NEW YORK
```

****To drop View****

Syntax: Drop view view_name;

Eg: SQL> drop view v3;

View dropped.

****To Revoke All privilege from Scott**

Step 4: Login as system(username) and tiger(password)

Step 5: SQL> revoke all privilege from scott;

Revoke succeeded.

Step 6: Exit from system.

ROWID

- For each row in the database, the ROWID pseudocolumn returns the address of the row.
- It is the identity or address of a row.
- It is representative of the allocation of physical memory.
- Rowid values have several important uses:

- They are the fastest way to access a single row.
- They can show you how the rows in a table are stored.
- They are unique identifiers for rows in a table.

You can use the ROWID pseudocolumn in the SELECT and WHERE clause of a query, these pseudocolumn values are not actually stored in the database. You cannot insert, update, or delete a value of the ROWID pseudocolumn.

Eg: SQL> SELECT * FROM DEMO80;

ID	NAME	CITY
1	A	
2	B	

SQL> SELECT ROWID FROM DEMO80;

```
ROWID
------------------
AAANLZAAEAAAAKNAAA
AAANLZAAEAAAAKNAAB

* UPDATE DEMO80 SET  CITY='MUMBAI' WHERE ROWID='AAANLZAAEAAAAKNAAB';
1 row updated.

SQL> SELECT * FROM DEMO80;
       ID NAME       CITY
---------- ---------- ----------
        1 A
        2 B          MUMBAI

SQL> select * from demo80 where ROWID='AAANLZAAEAAAAKNAAB';
       ID NAME       CITY
---------- ---------- ----------
        2 B          MUMBAI
```

ROWNUM The RowId pseudocolumn returns a number indicating the order in which Oracle selects the row from a table

- The first row selected has a rownum of 1, the second has 2, and so on.
- Is a temporarily assigned sequence to a row.
- Is representative of the sequence allocated to any data retrieval
- You can use rownum to limit the number of rows returned by a query, as in this example:

eg: Select * from emp where rownum<10;

SQL> select rowid,rownum,ename from emp;

ROWID	ROWNUM	ENAME
AAAMfPAAEAAAAAgAAA	1	SMITH
AAAMfPAAEAAAAAgAAB	2	ALLEN
AAAMfPAAEAAAAAgAAC	3	WARD
AAAMfPAAEAAAAAgAAD	4	JONES
AAAMfPAAEAAAAAgAAE	5	MARTIN
AAAMfPAAEAAAAAgAAF	6	BLAKE
AAAMfPAAEAAAAAgAAG	7	CLARK
AAAMfPAAEAAAAAgAAH	8	SCOTT
AAAMfPAAEAAAAAgAAI	9	KING
AAAMfPAAEAAAAAgAAJ	10	TURNER
AAAMfPAAEAAAAAgAAK	11	ADAMS
AAAMfPAAEAAAAAgAAL	12	JAMES
AAAMfPAAEAAAAAgAAM	13	FORD
AAAMfPAAEAAAAAgAAN	14	MILLER

14 rows selected.

ATTRIBUTES

---

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

ATTRIBUTES

----

1. KEY ATTRIBUTE

AN ATTRIBUTE WHICH IS USED TO IDENTIFY A RECORD UNIQUELY FROM THE TABLE IS CALLED KEY ATTRIBUTE.

2. **NON-KEY ATTRIBUTE**

ALL THE ATTRIBUTES EXCEPT KEY ATTRIBUTES ARE REFFERED AS NON-KEY ATTRIBUTES.

3. **PRIME-KEY ATTRIBUTES**

AMONG THE KEY ATTRIBUTES AN ATTRIBUTE IS CHOOSEN TO BE THE MAIN ATTRIBUTE TO IDENTIFY THE RECORD UNIQUELY FROM THE TABLE.

4. **NON- PRIME KEY ATTRIBUTE**

ALL THE KEY ATTRIBUTES EXCEPT PRIME KEY ATTRIBUTE IS REFFERED NON- PRIME KEY ATTRIBUTE.

5. **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.

6**. SUPER KEY ATTRIBUTE**

IT IS THE SET OF ALL THE KEY ATTRIBUTES.

7. FOREIGN KEY ATTRIBUTE

IT BEHAVES AS AN ATTRIBUTE OF ANOTHER ENTITY TO REPRESENT THE RELATION.

FUNCTIONAL DEPENDENCY

LET US CONSIDER THE RELATION 'R' WITH TWO ATTRIBUTES 'X' AND 'Y'
RESPECTIVELY. IN WHICH ATTRIBUTE 'X' DETEMINES ATTRIBUTE 'Y'.

or IN OTHER WORDS, 'Y' IS DEPENDENT 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

1. **TOTAL FUNCTIONAL DEPENDENCY**

IF ALL THE ATTRIBUTES IN A RELATION ARE DETEMINED BY A SINGLE ATTRIBUTE
WHICH IS A KEY ATTRIBUTE, THEN THERE EXISTS TOTAL FUNCTIONAL DEPENDENCY.

- IN TOTAL FUNCTIONAL DEPENDENCY THERE ARE NO ANOMALY AND REDUNDANCY.

ANOMALY - THESE ARE THE SIDE EFFECTS WHICH ARE CAUSED DURING THE DML OPERATIONS.

REDUNDANCY - THESE ARE THE REPEATED or DUPLICATE.

EG:- 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 IS K.A

A--> B

A--> C

A--> D

THEIR EXISTS T.F.D.

A--> { B,C,D }.

2. PARTIAL FUNCTIONAL DEPENDENCY

FOR A PARTIAL FUNCTIONAL DEPENDENCY TO EXIST THERE MUST BE A

COMPOSITE KEY ATTRIBUTE.

ONE OF THE ATTRIBUTE IN COMPOSITE KEY RELATION DETERMINES ANOTHER ATTRIBUTE SEPERATELY,

AND THIS IS KNOWN AS PARTIAL FUNCTIONAL DEPENDENCY.

IN PARTIAL FUNCTIONAL DEPENDENCY WE HAVE REDUNDANCY AND ANOMALY.

EG:- 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.K.A

(A,B)-->(C,D)
B--> {C}

THEIS EXIST P.F.D.

3. TRANSITIVE FUNCTIONAL DEPENDENCY

IF AN ATTRIBUTE IS DETERMINED BY A NON-KEY ATTRIBUTE WHICH INTERN IS DETERMINED BY A KEY ATTRIBUTE,THEN THERE EXIST TRANSITIVE FUNCTIONAL DEPENDENCY.

IN TRANSITIVE FUNCTIONAL DEPENDENCY WE HAVE REDUNDANCY AND ANOMALY.

EG:- LET US CONSIDER A RELATION WHITH FOUR ATTRIBUTES A,B,C, & D
IN WHICH A IS A KEY ATTRIBUTES .

R--> { A,B,C,D }

A IS K.A.

A--> B
A--> D
D--> C
A--> C
THIER EXIST T.F.D

Relationships in SQL

- We can manipulate the date stored in database using SQL if the data is stored in structured format.
- RDBMS is the management system where data is stored in structed format (i.e is table format)
- If data is stored in tables then to maintain correctness ,consistency of table we should able to establish the relations between tables.

    - ❖ Following are the types of relations
        1. One to one relation
        2. One to many or many to one relation
        3. Many to many relation

1) One to One relation

- Two tables can have one to one relation

-In case of one to one relation , single record in Table A is related to the single record in Table B and vice-versa.

If table is person_details(pid,Name, age, address, contact no, Aadhar no.).

In the above example, we have stored the aadhar card number in the same table 'Person', but it would be better if we will create another table for the 'Aadhar card' because the aadhar card number may be sensitive data and should be hidden from others.

For example, If there are two table,

Person-(Pid,Name, age, address, contact no.)

Aadhar card- (Pid, Aadhar no.).

So each person can have only one aadhar card, and the single aadhar card belongs to only one person.

Consider a table of Employee as shown below:

Table A:

emp_id	emp_name	emp_address
001	Claira Anderson	113, Zaraiah Road, TX 77001
003	Marc Doe	34343, Palm Jumeriah Road, Dubai 990039
005	Bruce Quilt	23, Santa Cruiz Road, NY 44303

Now, you can place the employee address in a separate table as shown below:

Employee:

emp_id	emp_name	emp_address_id
001	Claira Anderson	901
003	Marc Doe	903
005	Bruce Quilt	905

Employee Address:

emp_address_id	emp_address
901	113, Zaraiah Road, TX 77001
903	34343, Palm Jumeriah Road, Dubai 990039
905	23, Santa Cruiz Road, NY 44303

2) One to Many Relationship:

-Any single rows of the Table A can be related to one or more rows of the Table B

-If two tables are having one to many to relation then automatically many to one relation will get achieved.

For example, if there are two tables,

Table A: Customer (Cust_id(pk),C_Name,Contact_no)

Cust_id	C_name	Contact_n o
1	A	123
2	B	124

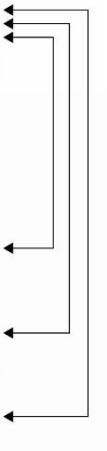
Table B: Account (Acc_id(pk),Acc_no,Cust_id(Fk))

Acc_id	Acc_no	Cust_id
1	1122	1
2	1133	1
3	1144	2

then each customer can have more than one account, and also, each account is owned by one customer only.

Customer_id	Customer_name	Customer_n0
111	Maria Danzie	1199229921
222	Alex Brat	3939637382
333	Sania Martini	8249283563

Order_id	Order_amount	Customer_id
10001	1200	222
10002	2000	333
10003	4500	222
10004	1220	111
10005	3550	222



In this example, there is a one-to-many relationship if you will observe from perspective ofthe Customers.

As shown above, the customer_id - 222 is related to the three different order_id.

In a similar way, there is a many-to-one relationship between the tables if you will observe fromthe perspective of the Orders table.

3) Many-to-Many Relationship

Two tables can have many to many relationship

If tables are having many to many relationship then we have to create separate table for mapping theirrelation.

A many-to-many relationship exists between the tables if a one record of the Table A is related to one ormore records of the Table B and a one record in the Table B is related to one or more records of the
Table A.

For example, consider the two tables

Table A: student table (stud_id,Sname)

Table B: courses table. (Course_id,C_Name)

A particular student may enroll himself in one or more than one course, while a course also may haveone or more students.

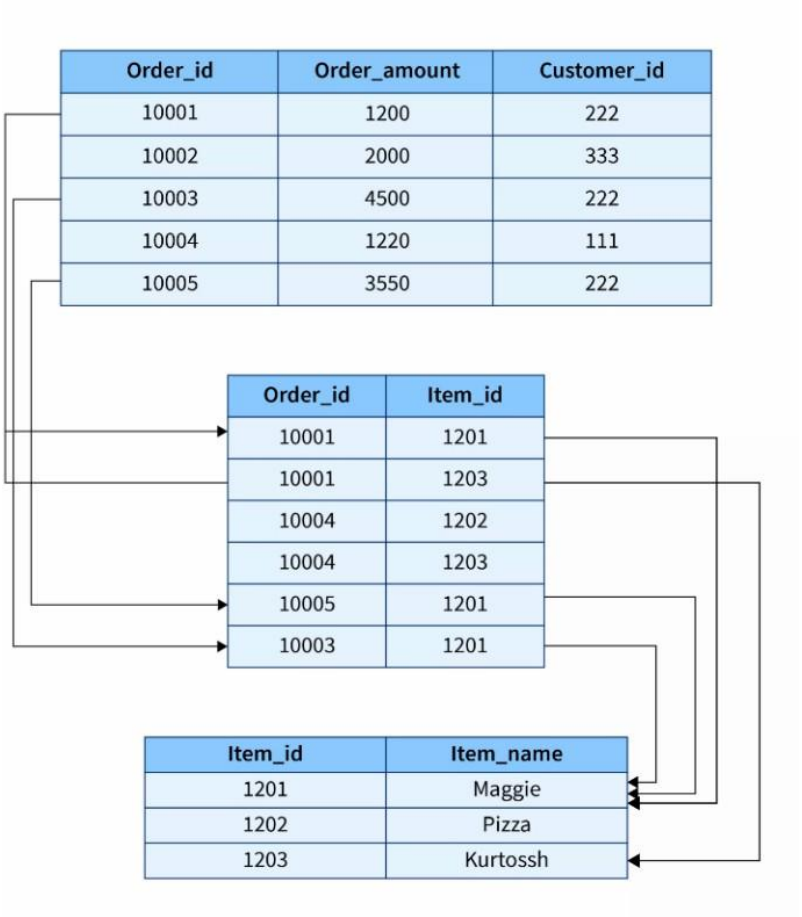
Student

stud_id	sname
1	A
2	B

course

course_id	cname
10	java
20	sql

sid	cid
1	10
2	20
1	10
2	20



Order_id	Order_amount	Customer_id
10001	1200	222
10002	2000	333
10003	4500	222
10004	1220	111
10005	3550	222

Order_id	Item_id
10001	1201
10001	1203
10004	1202
10004	1203
10005	1201
10003	1201

Item_id	Item_name
1201	Maggie
1202	Pizza
1203	Kurtossh