UNIT-IV::Structured Query Language

1. Explain about SQL history.

ORACLE History:-

- 1970 – Dr. E.F Codd of IBM. He is also known as the father of relational databases. He described a relational model for databases.
- 1974 – D. Chamberlin of IBM. He defined a language called 'Structured English Query Language' (SEQUEL).
- 1976- A new version, SEQUEL/2, was defined in 1976 but name was later changed to SQL for legal reasons.
- SQL stands for **Structured Query Language**.
- It is designed for managing data in a relational database management system (RDBMS).
- SQL is a database language, it is used for creation of tables, accessing data and modifying rows, deletion.. etc.
- All DBMS like MySQL, Oracle, MS Access, Sybase and SQL Server use SQL as standard database language.
- Still 'SQL' is pronounced as 'SEQUEL'.

2. Explain about different SQL Data Types.

SQL Data Types

Data type represents the type of data in the database.

Data types are classified in four types.

1. **Number(P,S)**
2. **Char(N).**
3. **Varchar\varchar2(N).**
4. **Date.**

1. **Number(p,s):-**
   - ❖ It is used to represent the numeric information.
   - ❖ It is represents variable length of numeric data.
   - ❖ Where P is the precision, S is the scale

- P range is up to 32 digits for the oracle (7.x),38 digits for the oracle (8.0)

Syntax:

Column_name number(P,S);

Ex:-

Emp_salary number(6,2)

The value may be in the form of : 119000.00

2. **CHAR(N):-**

It is used to represent character information.

It is fixed length character data type supports static memory allocation.

The maximum limit for N is 256 bytes for oracle (7.x), 2000 characters for oracle(8.0).

Each byte represents the one character.

Ex:-

Emp_name char(20);

'ram' ---- ram is actually 3 bytes but total 20 bytes allocated

3. **Varchar/Varchar2(N):-**

It is also used to represent the character information.

It is a varying length character data type supports Dynamic Memory Allocation.

Maximum limit for N is 2000 bytes in oracle(7.x), 4000 bytes in oracle(8.0)

Ex:-

Emp_name char(20);

'ram' ---- only 3 bytes allocated

'murali krishna'------ only 14 bytes allocated

4. **DATE:-**

It is used to represent the data and time information.

7 bytes of Memory is occupied for the data type.

Standard format of the date data type is

'DD-MON-YYYY'

'DD-MM-YY and HH:MM:SS'

EX:- '10-JUN-2017'

3. Explain about imposition of constraints in SQL.

SQL Constraints:-

A constraint is a mechanism which is used to stop or restrict invalid data into the table.

Constraints can be divided into following two types,

- **Column level constraints :** limits only column data
- **Table level constraints :** limits whole table data

Constraints in SQL:

- ❖ NOT NULL
- ❖ UNIQUE
- ❖ CHECK
- ❖ DEFAULT
- ❖ PRIMARY KEY
- ❖ FOREIGN KEY

NOT NULL Constraint:-

- ❖ If you do not enter the value in any particular column in the table that column automatically take null.
- ❖ **Not null Constraints** does not allow entering null value on a particular column in a table.
- ❖ we can apply not null constraints on more than one column in same table.

Example:

```
SQL> create table first(sid number(3) not null,sname varchar(20));

Table created.

SQL> desc first;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 SID                                       NOT NULL NUMBER(3)
 SNAME                                              VARCHAR2(20)

SQL>
```

Unique Constraint:-

- ❖ **Unique Constraints** does not allow duplicate values.
- ❖ We can apply unique constraints on more than one column in same table.

Example:-

```
SQL> create table second(sid number(3) not null,sname varchar(20) unique);

Table created.

SQL> desc second;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SID                                       NOT NULL NUMBER(3)
 SNAME                                              VARCHAR2(20)

SQL> select * from second;

       SID SNAME
---------- --------------------
       101 murali
       103 vamsi
SQL> insert into second values(103,'murali');
insert into second values(103,'murali')
*
ERROR at line 1:
ORA-00001: unique constraint (SCOTT.SYS_C005218) violated
```

Check Constraints:-

- **Check Constraints** does not allow the values which are not satisfied the given condition.
- we can apply Check constraints on more than one column in same table. Check constraints allows entering null values.

Example:-

```
SQL> create table degree_student(sno number(3),sname varchar(20),age number(2) check(age>18));

Table created.

SQL> insert into degree_student values(101,'murali',19);

1 row created.

SQL> insert into degree_student values(102,'raju',18);
insert into degree_student values(102,'raju',18)
*
ERROR at line 1:
ORA-02290: check constraint (SCOTT.SYS_C005219) violated
```

Default Constraints

- Default Constraints is used to insert a default value into a column.
- The default value will be added to all new records, if no other value is specified.

Example:-

```
SQL> select * from emp1;

      E_ID E_NAME                      SAL CITY
---------- --------------- ---------- ------------------------------
       101 murali                  21000 singarayakonda
       102 krishna                 20000 ONGOLE

SQL> insert into emp1 (e_id,e_name,sal) values(103,'ramu',25000);

1 row created.

SQL> select * from emp1;

      E_ID E_NAME                      SAL CITY
---------- --------------- ---------- ------------------------------
       101 murali                  21000 singarayakonda
       102 krishna                 20000 ONGOLE
       103 ramu                    25000 ONGOLE
```

Primary Key Constraints

- It is used to define a key column of a table.
- Primary Key Constraints does not allow duplicate as well as null values.
- This constraint is supported with an index automatically.
- We cannot apply Primary Key constraints on more than one column in same table.

Primary key= NOT NULL + UNIQUE + INDEX

Example:-

```
SQL>  create table dept5
  2  (
  3  d_no number(3) primary key,
  4  dname varchar(20) not null unique,
  5  location varchar (20) default 'ongole'
  6  );

Table created.
```

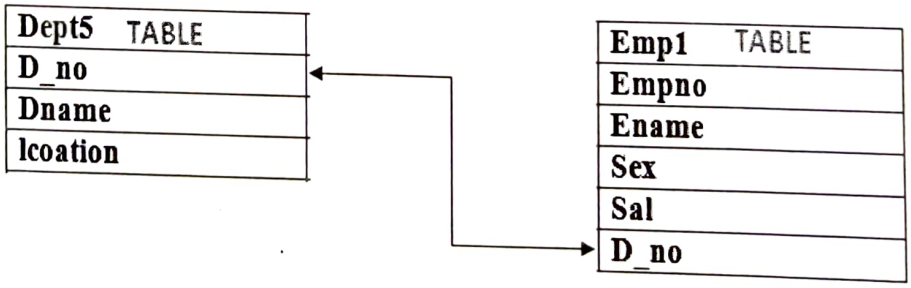
SQL FOREIGN KEY:-

- A foreign key a column that is used to establish a link between two tables.
- In simple words we can say that, a foreign key is one column in a table that is used to point primary key column in another table.
- It allows NULL and Duplicate values.
- It can be related to either primary key or unique constraint column of other table.

$$primary\ key/_{unique} \quad <------> foreign\ key$$

```
SQL> create table emp1
  2  (
  3  empno number(3) primary key,
  4  ename varchar(20) not null,
  5  sex char(1) check (sex in('m','f')),
  6  sal number(8,2),
  7  d_no number(3) references dept5 on delete cascade
  8  );

Table created.

SQL>
```

dependent on dept 5 table

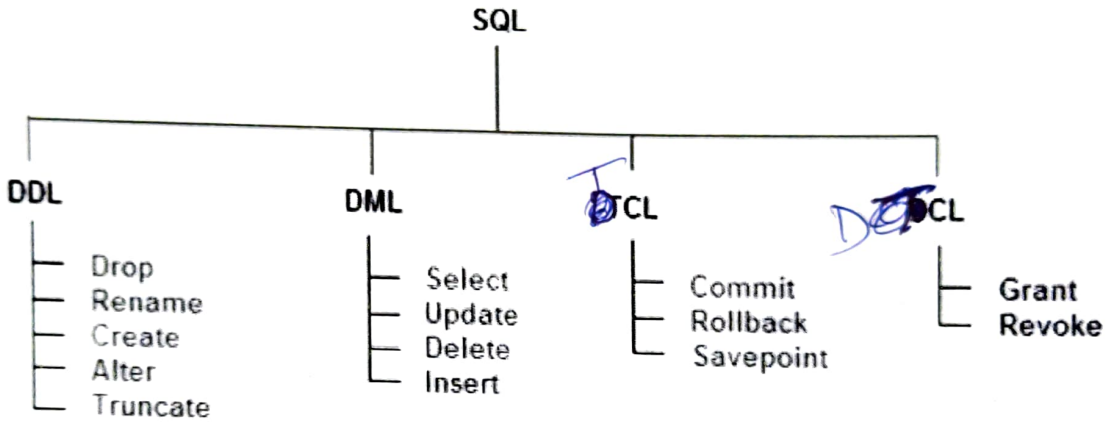


4. Explain about SQL commands?

SQL Commands are mainly classified into four types, which are DDL command, DML command, TCL command and DCL command.

SQL is mainly divided into four sub language

- **Data Definition Language(DDL)**
- **Data Manipulation Language(DML)**
- **Transaction Control Language(TCL)**
- **Data Control Language(DCL)**



DDL commands:-

DDL is means Data Definition Language, it is used to define the database schema and structures.

- ✓ CREATE – to create database and its objects like (table, index, views, and triggers).
- ✓ ALTER – it can change the structure of the existing database.
- ✓ DROP – delete objects from the database.
- ✓ TRUNCATE – remove all records from a table, including all spaces allocated for the records are removed.
- ✓ RENAME – rename an object.

DML commands:-

DML is means Data Manipulation Language which deals with data manipulation, and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE etc, and it is used to store, modify, retrieve, delete and update data in database.

- ✓ SELECT – retrieve data from the database.
- ✓ INSERT – insert data into a table.
- ✓ UPDATE – updates existing data within a table.
- ✓ DELETE – Delete all records from a database table.

DCL commands:-

DCL is means Data Control Language, by using the DCL commands the data administrator can manage the data control from other users in the database environment.

- ✓ GRANT – allow users access rights (permissions) to database.
- ✓ REVOKE – withdraw user's access rights (permissions) given by using the GRANT command.

SQL> Revoke privileges from user-name;
SQL> Revoke Create, Insert from x18317805 6;

1) SQL> Grant privileges to user-name;
2) SQL> Grant Create, Insert to x18317805 6;

DCL commands:-

TCL is means Transaction Control Language which deals with transaction within a database.

- ✓ COMMIT – commits a Transaction. SQL> Commit;
- ✓ ROLLBACK – rollback a transaction in case of any error occurs. SQL> Roll back;
- ✓ SAVEPOINT – to rollback the transaction making points within groups.

SQL> save point on number;
SQL> savepoint 5;

5. Explain about DDL commands.

DDL : Data Definition Language

All DDL commands are auto-committed. That means it saves all the changes permanently in the database.

Command	Description
Create	to create new table or database
Alter	for alteration
Truncate	delete data from table
Drop	to drop/delete a table
Rename	to rename a table

Create command

Create is a DDL command used to create data base objects.
database objects ex:- table, view and so on.

Creating a Table:-

- ✓ **Create Table** statement is used for create a table in database.
- ✓ We can specify names, data types and the sizes of the various columns.

Syntax

```
CREATE TABLE table_name
(
column_name1 data_type(size),
column_name2 data_type(size),
column_name3 data_type(size),
....
.....
);
```

Example

```
SQL> CREATE TABLE Employee
  2  (
  3  emp_id number(5),
  4  name varchar(20),
  5  salary number(10)
  6  );

Table created.

SQL> desc employee
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMP_ID                                             NUMBER(5)
 NAME                                               VARCHAR2(20)
 SALARY                                             NUMBER(10)

SQL>
```

Alter command

Alter Command is used to modify the structure of the table. Using this command we can perform four different operations.

This command contains four sub commands those are:

- **Alter modify**
- **Alter add**

- Alter rename
- Alter drop

Alter modify

Using this command we can increase or decrease the size of data type and also we can change the data type from old data type to new data type.

Syntax

```
ALTER TABLE table_name MODIFY
(
column 1 datatype(size),
column 2 datatype(size),
.....
.....
);
```

Example

```
SQL> alter table employee modify ( name varchar2(40));

Table altered.

SQL> desc employee
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMP_ID                                             NUMBER(5)
 NAME                                               VARCHAR2(40)
 SALARY                                             NUMBER(10)

SQL>
```

Alter add:-

This command is used to add a new column in the table. Using this command we can add more than one column to the existing.

Syntax

```
ALTER TABLE table_name ADD
(
column 1 datatype(size),
column 2 datatype(size),
...
...
);
```

Example

```
SQL> ALTER TABLE Employee ADD
  2  (
  3  emp_age number(5)
  4  );

Table altered.

SQL> desc employee
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMP_ID                                             NUMBER(5)
 NAME                                               VARCHAR2(40)
 SALARY                                             NUMBER(10)
 EMP_AGE                                            NUMBER(5)

SQL>
```

Note: We cannot realignment the table structure. Which means we cannot add the new column at required position in the table.

Alter rename:-

This command is used to change the column name from old column name to new column name.

Syntax

```
ALTER TABLE table_name RENAME COLUMN old_column_name TO new_column_name;
```

Example

```
SQL> ALTER TABLE EMPLOYEE RENAME COLUMN SALARY TO EMP_SALARY;

Table altered.

SQL> DESC EMPLOYEE
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMP_ID                                             NUMBER(5)
 NAME                                               VARCHAR2(40)
 EMP_SALARY                                         NUMBER(10)
 EMP_AGE                                            NUMBER(5)

SQL>
```

Note: We can not change more than one column name at a time.

Alter drop:-

This command is used to remove the column from existing table.

Syntax

```
ALTER TABLE table_name DROP column_name;
```

Example

```
SQL> ALTER TABLE EMPLOYEE DROP COLUMN EMP_SALARY;

Table altered.

SQL> DESC EMPLOYEE
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMP_ID                                             NUMBER(5)
 NAME                                               VARCHAR2(40)
 EMP_AGE                                            NUMBER(5)

SQL> |
```

RENAME (TABLE):-

SQL RENAME is used to change the name of a table. Sometimes, we choose non-meaningful name for the table. So it is required to be changed.

Let's see the syntax to rename a table from the database.

```
ALTER TABLE table_name
RENAME TO new_table_name;
```

Optionally, we can write following command to rename the table.

```
RENAME old_table _name To new_table_name;
```

```
SQL> ALTER TABLE EMPLOYEE RENAME TO EMP_TABLE;

Table altered.

SQL> DESC EMP_TABLEE
ERROR:
ORA-04043: object EMP_TABLEE does not exist


SQL> DESC EMP_TABLE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMP_ID                                             NUMBER(5)
 NAME                                               VARCHAR2(40)
 EMP_AGE                                            NUMBER(5)

SQL> DESC EMPLOYEE
ERROR:
ORA-04043: object EMPLOYEE does not exist


SQL> |
```

TRUNCATE :-

TRUNCATE command is used to delete complete data from the table. But , it cannot delete the table structure

Syntax:- SQL> truncate table <tablename>;

Ex:-

```
SQL> truncate table mpcs;
Table truncated.
SQL>
```

DROP:-

Drop command is used to remove the table from the database.

Syntax

```
drop table <table_name>;
```

Example

```
drop table Employee;
```

Note: Drop command do not drop table permanently from database. Once we drop the table then that table gone into recycle bin.

Permanently drop table from database

To drop table from recycle bin we can use **purge table <table_name>** command

Syntax

pruge table <table_name>;

Example

pruge table Employee;

Permanently drop table from database

To drop table permanently from database we can use following command

Syntax

drop table <table_name> pruge;

Example

drop table Employee pruge;

6. Explain about DML operations in SQL.

Data Manipulation Language (DML) statements are used for managing data in database. DML commands are not auto-committed. It means changes made by DML command are not permanent to database, it can be rolled back.

Command	Description
Insert	to insert a new row
Update	to update existing row
Delete	to delete a row

SQL Insert Command:-

Insert Command is used to insert new record in table. Using Insert Command we can insert a single or a multiple records in a table.

Insert data in table in two form, which is given below.

- **Insert data without specifying column name**
- **Insert data by specifies both the column names and the values to be inserted**

Insert data without specifying column name:-

Syntax:-

```
INSERT INTO table_name VALUES (value1,value2,value3,...);
```

Example:-

```
SQL> insert into employee values (101,'murali',31);

1 row created.

SQL> select * from employee;

    EMP_ID NAME                                               EMP_AGE
---------- -------------------------------------------- ----------
       101 murali                                               31
```

Note:- all character data must be represented in between the single codes.

Ex:- 'murali krishna'..etc

Insert data by specifies both the column names and the values to be inserted:-

Syntax:-

```
INSERT INTO table_name (column1,column2,column3,...)
VALUES (value1,value2,value3,...);
```

Example

```
SQL> insert into employee(emp_id,name) values (102,'siva');

1 row created.

SQL> select * from employee;

    EMP_ID NAME                           EMP_AGE
---------- -------------------- ----------
       101 murali                           31
       102 siva

SQL>
```

Select Command:-

Select command is used to retrieve the data from the table.

Retrieve complete data from the table:

```
SQL> select * from employee;

    EMP_ID NAME                 EMP_AGE     SALARY     DEPTNO
---------- -------------------- ---------- ---------- ----------
       101 murali                       31      15000         10
       102 siva                         30      10000         20
       103 raju                         31      16000         10
       104 krishna                      32      16000         30

SQL>
```

Retrieve required data from the table:

Query:- suppose if we want to see the employee details salary having 16000.

```
SQL> select * from employee where salary=16000;

    EMP_ID NAME                 EMP_AGE     SALARY     DEPTNO
---------- -------------------- ---------- ---------- ----------
       103 raju                         31      16000         10
       104 krishna                      32      16000         30

SQL>
```

SQL Update Command:-

Update command is used for modify the data in the table. Using this command we can modify all the records in the table and also we can modify some specific records in the table using "where clause". Using this command we can modify single column and also modify more than one column at a time.

Syntax

```
UPDATE table_name
SET column=value
WHERE column_name=some_value;
```

- ✓ Update column using where clause
- ✓ Using where clause we can update specific record from any existing table.

Example:-

```
SQL> select * from employee;

    EMP_ID NAME                 EMP_AGE
---------- -------------------- ----------
       101 murali                       31
       102 siva

SQL> update employee set emp_age=30 where name='siva';

1 row updated.

SQL> select * from employee;

    EMP_ID NAME                 EMP_AGE
---------- -------------------- ----------
       101 murali                       31
       102 siva                         30

SQL>
```

value is update with 30

SQL Delete Command:-

Delete command is used to delete the record from the table.

- Using Delete command you can delete all the records from the table without delete table.
- Using Delete command you can delete specific records from the table "using where clause"

Syntax:-

```
DELETE FROM table_name
WHERE some_column=some_value;
```

Example:-

```
SQL> select * from employee;

    EMP_ID NAME                 EMP_AGE
---------- -------------------- ----------
       101 murali                       31
       102 siva                         30

SQL> delete from employee where name='siva';

1 row deleted.

SQL> select * from employee;

    EMP_ID NAME                 EMP_AGE
---------- -------------------- ----------
       101 murali                       31

SQL>
```

Delete all record:-

You can also delete all the record from a table without deleting table.

Syntax

```
DELETE FROM table_name
      or
DELETE * FROM table_name;
```

Example

```
SQL> select * from employee;

    EMP_ID NAME                 EMP_AGE
---------- -------------------- ----------
       101 murali                       31

SQL> delete from employee;

1 row deleted.

SQL> select * from employee;

no rows selected

SQL>
```

How to Create a new user! -> SQL>Create user user-name identified by password;
SQL>Create user 9183178056 identified by *****;

7. Explain about table modification commands?

- ✓ **Alter is a table modification command.**
- ✓ **Alter Command** is used to modify the structure of the table.
- ✓ Using this command we can perform four different operations.
- ✓ This command contains four sub commands that is;
  - **Alter modify**
  - **Alter add**
  - **Alter rename**
  - **Alter drop**

Alter modify:-

Using this command we can increase or decrease the size of data type and also we can change the data type from old data type to new data type.

Syntax

```
ALTER TABLE table_name MODIFY
(
column 1 datatype(size),
column 2 datatype(size),
.....
);
```

```
SQL> alter table employee modify ( name varchar2(40));

Table altered.

SQL> desc employee
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMP_ID                                             NUMBER(5)
 NAME                                               VARCHAR2(40)
 SALARY                                             NUMBER(10)

SQL>
```

Alter add:-

This command is used to add a new column in the table. Using this command we can add more than one column to the existing.

Syntax

```
ALTER TABLE table_name ADD
(
column 1 datatype(size),
column 2 datatype(size),
.....
);
```

Example

```
SQL> ALTER TABLE Employee ADD
  2  (
  3  emp_age number(5)
  4  );

Table altered.

SQL> desc employee
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMP_ID                                             NUMBER(5)
 NAME                                               VARCHAR2(40)
 SALARY                                             NUMBER(10)
 EMP_AGE                                            NUMBER(5)

SQL>
```

Note: We cannot add the new column at required position in the table.

Alter rename:-

This command is used to change the column name from old column name to new column name.

Syntax

```
ALTER TABLE table_name RENAME COLUMN old_column_name TO new_column_name;
```

Example

```
SQL> ALTER TABLE EMPLOYEE RENAME COLUMN SALARY TO EMP_SALARY;

Table altered.

SQL> DESC EMPLOYEE
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMP_ID                                             NUMBER(5)
 NAME                                               VARCHAR2(40)
 EMP_SALARY                                         NUMBER(10)
 EMP_AGE                                            NUMBER(5)

SQL>
```

Note: We can not change more than one column name at a time.

Alter drop:-

This command is used to remove the column from existing table.

Syntax

```
ALTER TABLE table_name DROP column_name;
```

Example

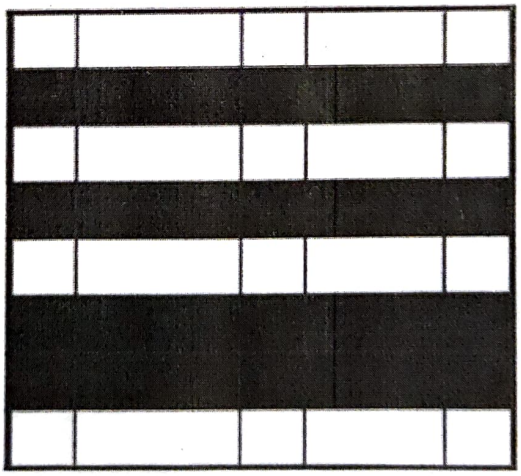
```
SQL> ALTER TABLE EMPLOYEE DROP COLUMN EMP_SALARY;

Table altered.

SQL> DESC EMPLOYEE
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMP_ID                                             NUMBER(5)
 NAME                                               VARCHAR2(40)
 EMP_AGE                                            NUMBER(5)

SQL>
```

8. Explain about selection, Projection operations in SQL?

Selection:-

It is a process of selecting the all or required records from the particular table.

Table : Selection



Syntax:-

```
Select * from <table name> where <condition>;
```

Example:-

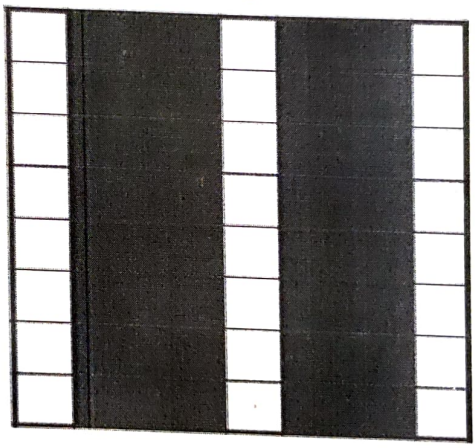
```
SQL> select * from employee;

    EMP_ID NAME                 EMP_AGE
---------- -------------------- ----------
       101 murali                       31
       102 siva                         30
       103 raju                         31
```

```
SQL> select * from employee where emp_id=101 or emp_id=103;

    EMP_ID NAME                 EMP_AGE
---------- -------------------- ----------
       101 murali                       31
       103 raju                         31

SQL>
```

Projection:-

It is a process of selecting one or more columns from the particular table.

Table : Projection



Syntax:-

```
Select <column 1,...column N> from <table name>;
```

Example:-

```
SQL> select * from employee;

    EMP_ID NAME                           EMP_AGE
---------- -------------------- ----------
       101 murali                          31
       102 siva                            30
       103 raju                            31

SQL> select name,emp_age from employee;

NAME                           EMP_AGE
-------------------- ----------
murali                              31
siva                                30
raju                                31

SQL> select emp_id,emp_age from employee;

    EMP_ID     EMP_AGE
---------- ----------
       101          31
       102          30
       103          31
```

Combination of selection and projection in SQL:

- ✓ It is a process of accessing the data from the both columns and rows
- ✓ We can get the information from the particular columns based on the particular condition.

```
SQL> select * from employee;

    EMP_ID NAME                           EMP_AGE
---------- -------------------- ----------
       101 murali                          31
       102 siva                            30
       103 raju                            31

SQL> select emp_id,name from employee where emp_age=31;

    EMP_ID NAME
---------- --------------------
       101 murali
       103 raju

SQL>
```

9. Explain about SQL aggregate functions.

Aggregate function:-

- ➢ These functions are applied on group of values.
- ➢ They consider entire column content as one input source.
- ➢ They ignore NULL values.

Some aggregate functions are:

SQL aggregate function	Description
AVG	Average value of the column.
COUNT	Total Number of records.
MAX	Maximum value of the column.
MIN	Minimum value of the column.
SUM	Sum of the column.

Avg:-

Avg function gives average of given column supports with numbers only.

Syntax:-

Select avg(number type column) from <table name>;

Example:-

```
SQL> select * from employee;

    EMP_ID NAME                 EMP_AGE     SALARY
---------- -------------------- ---------- --------
       101 murali                       31    15000
       102 siva                         30    10000
       103 raju                         31    10000

SQL> select avg(salary) from employee;

AVG(SALARY)
-----------
 11666.6667

SQL>
```

Count:-

Count function gives number of rows in given column supports with all types of data.

Syntax:-

Select count(column name) from <table name>;

Example:-

```
SQL> select * from employee;

    EMP_ID NAME                 EMP_AGE     SALARY
---------- -------------------- ---------- ----------
       101 murali               31          15000
       102 siva                 30          10000
       103 raju                 31          10000

SQL> select count(name) from employee;

COUNT(NAME)
-----------
          3

SQL>
```

Max:-

Max gives the largest value of given column supports with number and date values only.

Syntax:-

Select max(column name) from <table name>;

Example:-

```
SQL> select * from employee;

    EMP_ID NAME                 EMP_AGE     SALARY
---------- -------------------- ---------- ----------
       101 murali               31          15000
       102 siva                 30          10000
       103 raju                 31          10000

SQL> select max(salary) from employee;

MAX(SALARY)
-----------
      15000

SQL>
```

Min:-

Min gives the lowest value of given column supports with number and date values only.

Syntax:-

Select min(column name) from <table name>;

Example:-

```
SQL> select * from employee;

    EMP_ID NAME                 EMP_AGE     SALARY
---------- -------------------- ---------- ----------
       101 murali                       31      15000
       102 siva                         30      10000
       103 raju                         31       9000

SQL> select min(salary) from employee;

MIN(SALARY)
-----------
       9000

SQL> |
```

Sum:-

Sum function is used to find the total of given column supports with number only.

Syntax:-

Select sum(column name) from <table name>;

Example:-

```
SQL> select * from employee;

    EMP_ID NAME                 EMP_AGE     SALARY
---------- -------------------- ---------- ----------
       101 murali                       31      15000
       102 siva                         30      10000
       103 raju                         31       9000

SQL> select sum(salary) from employee;

SUM(SALARY)
-----------
      34000

SQL> |
```

10. Explain about table truncation.

The SQL **TRUNCATE TABLE** command is used to delete complete data from an existing table.

You can also use DROP TABLE command to delete complete table but it would remove complete table structure form the database and you would need to re-create this table once again if you wish you store some data.

Syntax-

The basic syntax of a **TRUNCATE TABLE** command is as follows.

```
TRUNCATE TABLE table_name;
```

Example:-

```
SQL> select * from second;

        ID NAME
---------- --------------------
         2 raju
         3 vamsi

SQL> truncate table second;

Table truncated.

SQL> select * from second;

no rows selected
```

Difference between truncate and delete:-

	Truncate	Delete
1	It is DDL Command	It is DML Command
2	It is used for permanent deletion.	It is used for temporary deletion.
3	We cannot delete the specific record.	We can delete the specific records.
4	We cannot roll back the DDL commands	We can ROLL BACK DML commands.

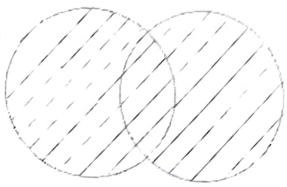
11. Explain about set operations.

SQL SET Operators are behaving same like mathematical sets, these sql set operators are classified into four types, which is given below;

- **Union**
- **Union all**
- **Minus**
- **Intersect**

Union:-

Union operator returns all the value from the entire table. But it do not include duplicate values, it means it not return duplicate values.

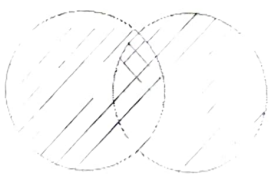


Example

```
SQL> select * from first;

        ID NAME
---------- --------------------
         1 murali
         2 raju

SQL> select * from second;

        ID NAME
---------- --------------------
         2 raju
         3 vamsi

SQL> select * from first union select * from second;

        ID NAME
---------- --------------------
         1 murali
         2 raju
         3 vamsi
```

Union all

Union all operator returns all the value from the entire table including duplicate values.

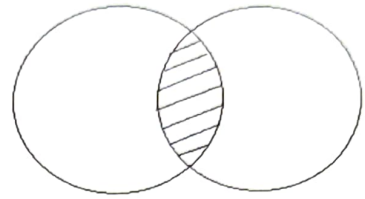


Example

```
SQL> select * from first union all select * from second;

        ID NAME
---------- --------------------
         1 murali
         2 raju
         2 raju
         3 vamsi

SQL>
```

Intersect:-

Intersect operator return the common value from all the variables.

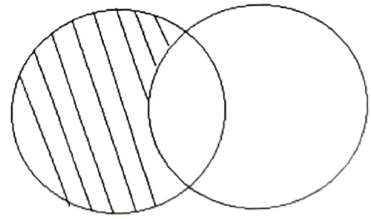


Example

```
SQL> select * from first intersect select * from second;

        ID NAME
---------- --------------------
         2 raju

SQL>
```

Minus

Minus operator return the values which are not available in first table but not available in second table.



Example

```
SQL> select * from first minus select * from second;

        ID NAME
---------- --------------------
         1 murali

SQL> select * from second minus select * from first;

        ID NAME
---------- --------------------
         3 vamsi

SQL>
```

12. Explain about table view in sql.

- **View** is a logical database object, which contains the logical representation of data. In a simple word View is the logical or virtual table.
- If you perform DML operation of view table the same operation automatically reflected to corresponding base table and vice-versa.
- **Note:** Once you drop the base table the corresponding view will not be drop, but it will become invalid.

View will become invalid in three cases:-

- Drop the base table.
- When the change the table name.
- When we modify the structure of the base table.

Syntax:-

```
CREATE VIEW view_name as SELECT * FROM table_name;
```

```
SQL> create view v1 as select * from employee;

View created.

SQL> insert into v1 values ( 104,'krishna',32,16000);

1 row created.
```

Drop a View:-

We can delete the view by using the drop command.

Syntax

```
DROP VIEW view_name;
```

Example:-

```
SQL> drop view v1;

View dropped.

SQL>
```

13. Explain about sub-queries in SQL.

- A subquery is a SQL query within a query.
- Subqueries are nested queries that provide data to another query.
- Subqueries can return individual values or a list of records
- Subqueries must be enclosed with parenthesis

There are a few rules that subqueries must follow –

- Subqueries must be enclosed within parentheses.
- A subquery can have only one column in the SELECT clause.
- An ORDER BY command cannot be used in a subquery, although the main query can use an ORDER BY
- Subqueries that return more than one row can only be used with multiple value operators such as the IN operator.
- A subquery cannot be immediately enclosed in a set function.

Example:-

```
SQL> select * from employee where salary=(select max(salary) from employee);

    EMP_ID NAME                 EMP_AGE     SALARY    DEPT_NO
---------- -------------------- ---------- ---------- ----------
       103 raju                         31      16000         10
       104 krishna                      32      16000         30

SQL> select * from employee;

    EMP_ID NAME                 EMP_AGE     SALARY    DEPT_NO
---------- -------------------- ---------- ---------- ----------
       101 murali                       31      15000         10
       102 siva                         30      10000         20
       103 raju                         31      16000         10
       104 krishna                      32      16000         30

SQL>
```

14. Explain about JOINS in SQL?

Join is used to retrieve the data from more than one table in a single query

There are following joins are available in SQL:-

1) Equi join.
2) Cartesian join[cross join].
3) Non-equi join.
4) Outer join.
5) Self join.

1) Equi join:-

- ✓ It is used to retrieve the data from more than 1 table based on "equality" condition.
- ✓ Tables must have common column between them to apply this join.
- ✓ If N tables are joined N-1 conditions are applied.

SQL> select name,d_name from emp,dept where emp.dno=dept.dno;
–THETA STYLE

SQL> select name,d_name from emp inner join dept on emp.dno=dept.dno;
–ANSI STYLE

Example:-

```
SQL> select * from employee;

    EMP_ID NAME                 EMP_AGE     SALARY     DEPTNO
---------- -------------------- ---------- ---------- ----------
       101 murali                       31      20000         10
       102 siva                         30      15000         20
       103 raju                         31      21000         10
       104 krishna                      32      21000         30

SQL> select * from dept;

    DEPTNO D_NAME
---------- --------------------
        10 computer
        20 physics
        30 chemistry
       102 jhon

SQL> select name,d_name from employee, dept where employee.deptno=dept.deptno;

NAME                 D_NAME
-------------------- --------------------
raju                 computer
murali               computer
siva                 physics
krishna              chemistry
```

THETA STYLE

```
SQL> select name,d_name from employee inner join dept on employee.deptno=dept.deptno;

NAME                 D_NAME
-------------------- --------------------
raju                 computer
murali               computer
siva                 physics
krishna              chemistry
```

ANSI STYLE

2) Cartesian join:-[cross join]

- ✓ It is used to retrieve the data from the multiple tables without any conditions.
- ✓ It is used to retrieve data analysis report.
- ✓ No need to have common column between tables to apply this type of join.

Example:-

```
SQL> SET PAGESIZE 400;
SQL> select name,d_name from employee, dept;

NAME                 D_NAME
-------------------- --------------------
murali               computer
murali               physics
murali               chemistry
murali               jhon
siva                 computer
siva                 physics
siva                 chemistry
siva                 jhon
raju                 computer
raju                 physics
raju                 chemistry
raju                 jhon
krishna              computer
krishna              physics
krishna              chemistry
krishna              jhon

16 rows selected.
```

3) Non-Equi join:-

- ✓ It is used to retrieve the data from multiple tables based on other condition but not equal.
- ✓ Non Equi Join uses all comparison operators except the equal (=) operator like !=, >=, <=, <, >.

```
SQL> select name,d_name from employee, dept where employee.deptno != dept.deptno;

NAME                 D_NAME
-------------------- --------------------
murali               physics
murali               chemistry
murali               jhon
siva                 computer
siva                 chemistry
siva                 jhon
raju                 physics
raju                 chemistry
raju                 jhon
krishna              computer
krishna              physics
krishna              jhon

12 rows selected.
```

4) Outer Join – This join returns all the rows from one table and only those rows from second table which meets the condition.

Outer join is classified into 3 types:

a. Left Outer Join

b. Right Outer Join

c. Full Outer Join

a. **Left Outer Join:–**

- ✓ Returns all the rows from left table(ie, first table) and rows that meet the condition from second table.
- ✓ All the rows from left table and only matching rows from the right table.

```
SQL> select name, d_name from employee, dept where employee.deptno=dept.deptno(+);

NAME                 D_NAME
-------------------- --------------------
raju                 computer
murali               computer
siva                 physics
krishna              chemistry

SQL> select name, d_name from employee left outer join dept on employee.deptno=dept.deptno;

NAME                 D_NAME
-------------------- --------------------
raju                 computer
murali               computer
siva                 physics
krishna              chemistry
```

b).Right Outer Join:-

- ✓ Returns all the rows from right table(ie, second table) and rows that meet the condition from first table.
- ✓ All rows from right table and only matching rows from the left table.

```
SQL> select name, d_name from employee, dept where employee.deptno(+)=dept.deptno;

NAME                 D_NAME
-------------------- --------------------
murali               computer
siva                 physics
raju                 computer
krishna              chemistry
                     jhon

SQL> select name, d_name from employee right outer join dept on employee.deptno=dept.deptno;

NAME                 D_NAME
-------------------- --------------------
murali               computer
siva                 physics
raju                 computer
krishna              chemistry
                     jhon
```

c). Full Outer Join :–

- ✓ Combines the results of both left and right outer joins.
- ✓ Non-matching records from both the tables will be left blank.

```
SQL> select name, d_name from employee full outer join dept on employee.deptno=dept.deptno;

NAME                 D_NAME
-------------------- --------------------
raju                 computer
murali               computer
siva                 physics
krishna              chemistry
                     jhon
```