Practical number 1

Aim:- Study of various DDL, DML and DCL commands.

Theory:- SQL commands are instructions that are used to communicate with the database. It is also used to perform specific tasks, functions and queries of data. SQL can perform tasks like create tables and add data to tables, drop the table, modify the table, set permission for users.

1. Data Definition Language (DDL):- DDL changes the structure of the table, like creating tables, deleting tables, altering the table, etc. All the commands of DDL are auto committed, which means it permanently saves all the changes in the database.
    - ➢ Some DDL commands are as follows:-
        - CREATE:- It is used to create a new table in the database.

          Syntax:- CREATE TABLE TABLE_NAME (CO;OUMN_NAME DATATYPE [....]);

          Example:- CREATE TABLE EMPLOYEE (Name VARCHAR(20), Email VARCHAR2(100) DOB, DATE);

        - DROP:- It is used to delete both the structures and records stored in the table.

          Syntax:- DROP TABLE TABLE_NAME

          Example:-DROP TABLE STUDENT;

        - ALTER:- It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probability to add an inner attribute.

          Syntax:- To add a new column in table

          ALTER TABLE TABLE_NAME ADD COLOUMN_NAME COLUMN DEFINITION

          To modify containing column in table

ALTER TABLE TABLE_NAME MODIFY COLOUMN_DEFINITIONS...);

Example:- ALTER TABLE STU ROLLNO ADD (ADDRESS VARCHAR (20));

ALTER TABLE STU ROLL NO MODIFY (NAME VARCHAR (20));

- TRUNCATE:- It is used to delete all the rows from the table and free the space containing the table.

Syntax:- TRUNCATE TABLE TABLE_NAME

Example:- TRUNCATE TABLE EMPLOYEE;

2. Data Manipulation Language(DML):- Commands are used to modify the database. It is responsible for all forms of changes in the database. The command of DML is not auto committed, which means it can't permanently save all the changeS in the database. They can be rolled back.

   How are some commands that come under DML Insert, Update, Delete, etc.

   - INSERT:- The insert statement is a SQL query. It is used to insert data into the row of a table.

     Syntax:- INSERT INTO TABLE NAME (col1, col2, col3, ...colN) VALUES (value1, value2, value3, ... valueN);

     Or INSERT INTO TABLE_NAME VALUES (value1, value2, value3, ... valueN);

     Example:- INSERT INTO javapoint (Author Subject) VALUES ("Sahil", "DBMS");

   - UPDATE:- This command is used to update or modify the value of a column in a table.

     Syntax:- UPDATE table_name SET [column_name= value1....column_name N=value N] (where condition)

   - DELETE:- It is used to remove one or more rows from a table.

     Syntax:- DELETE FROM Table_name (where condition)

Example:- DELETE FROM EMPLOYEE WHERE NAME = “SAHIL”;

3. Data Control Language(DCL):- DCL commands are used to grant and take back authority from any database user.
   - ➢ Some dcl commands are as follows:-

- GRANT:- It is used to give user access privilege to database

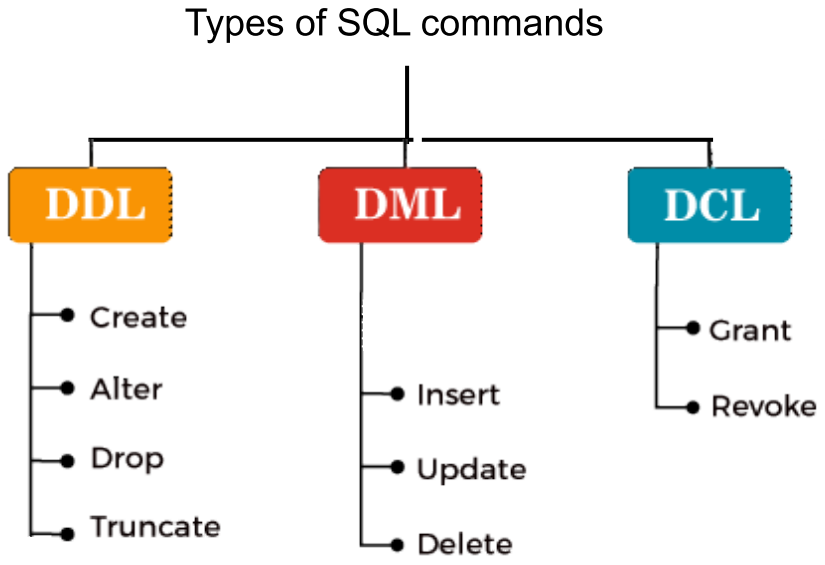
  Example:- GRANT SELECT UPDATE ON MY_TABLE TO SOME USER, ANOTHER USER

- REVOKE:- It is used to take back permission from user

  Example:- REVOKE SELECT UPDATE ON MY_TABLE FROM USER1, USER2;

Conclusion:- Hence, we have successfully studied the DDL DML and DCL commands.

Diagram (Left-hand side) :-



Practical No.2

Aim: Implementation of DDL and DML commands in SQL.

Theory:

CREATE: CREATE is used to create a new table in the database.

```
CREATE TABLE Employee (
    Employee_id   INT,
    NAME   VARCHAR (20),
    PHONE_NO   VARCHAR(10),
    CITY    VARCHAR (20),
    SALARY    INT,
    DEPARTMENT   VARCHAR (20));
```

INSERT: The INSERT statement is a SQL query. It is used to insert data into the row of a table.

```
INSERT INTO Employee (Employee_id, NAME, PHONE_NO, CITY, SALARY, DEPARTMENT)
    VALUES
    (1001, 'John Doe', 123456789, 'New York', 60000, 'IT'),
    (1002, 'Susan Grant',7896478521,'San Fran', 70000, 'Sales'),
    (1003, 'Bob Smith',9874569785,'Chicago', 65000,'Marketing'),
    (1004, 'Alice Lee', 9856742587,'Los Angeles', 85000,'HR'),
```

(1005, ‘Mike Wang’,8657848965,’Seattle’,75000, ’IT’),

(1006, ‘Emily Tan ’, 807037722,’Boston’, 65000,’Sales’),

(1007, ‘Chris Kim’, 9423438778,’Atlanta’, 55000,’ IT’),

(1008, ‘Sarah Lin’,9690897749,’Alaska’, 58000,’IT’),

(1009,’David Warner’, 9699089774,’Miami’, 60000,’Sales’),

(1010,’Shane Watson’,9854827722,’Boston’, 65000,’Sales’);

Employee_id	NAME	PHONE_NO	CITY	SALARY	DEPARTMENT
1001	John Doe	123456789	New York	60000	IT
1002	Susan Grant	7896478521	San Fran	70000	Sales
1003	Bob Smith	9874569785	Chicago	65000	Marketing
1004	Alice Lee	9856742587	Los Angeles	85000	HR
1005	Mike Wang	8657848965	Seattle	75000	IT
1006	Emily Tan	807037722	Boston	65000	Sales
1007	Chris Kim	9423438778	Atlanta	55000	IT
1008	Sarah Lin	9690897749	Alaska	58000	IT
1009	David Warner	9699089774	Miami	60000	Sales
1010	Shane Watson	9854827722	Boston	65000	Sales

UPDATE: This command is used to update or modify the value of a column in a table.

UPDATE Employee SET NAME=” Marc Harris” WHERE Employee_id=1002;

Employee_id	NAME	PHONE_NO	CITY	SALARY	DEPARTMENT
1001	John Doe	123456789	New York	60000	IT
1002	Marc Harris	7896478521	San Fran	70000	Sales
1003	Bob Smith	9874569785	Chicago	65000	Marketing
1004	Alice Lee	9856742587	Los Angeles	85000	HR
1005	Mike Wang	8657848965	Seattle	75000	IT
1006	Emily Tan	807037722	Boston	65000	Sales
1007	Chris Kim	9423438778	Atlanta	55000	IT
1008	Sarah Lin	9690897749	Alaska	58000	IT
1009	David Warner	9699089774	Miami	60000	Sales
1010	Shane Watson	9854827722	Boston	65000	Sales

DELETE: It is used to remove one or more rows from a table.

DELETE FROM Employee WHERE Employee_id=1010;

Employee_id	NAME	PHONE_NO	CITY	SALARY	DEPARTMENT
1001	John Doe	123456789	New York	60000	IT
1002	Marc Harris	7896478521	San Fran	70000	Sales
1003	Bob Smith	9874569785	Chicago	65000	Marketing
1004	Alice Lee	9856742587	Los Angeles	85000	HR
1005	Mike Wang	8657848965	Seattle	75000	IT
1006	Emily Tan	807037722	Boston	65000	Sales
1007	Chris Kim	9423438778	Atlanta	55000	IT
1008	Sarah Lin	9690897749	Alaska	58000	IT
1009	David Warner	9699089774	Miami	60000	Sales

ALTER: It is used to alter the structure of the database . This change could be either to modify the characteristics of an existing attribute or probability to add an inner attribute.

ALTER TABLE Employee ADD COLUMN EMAIL VARCHAR(50);

Employee_id	NAME	PHONE_NO	CITY	SALARY	DEPARTMENT	EMAIL
1001	John Doe	123456789	New York	60000	IT	NULL
1002	Marc Harris	7896478521	San Fran	70000	Sales	NULL
1003	Bob Smith	9874569785	Chicago	65000	Marketing	NULL
1004	Alice Lee	9856742587	Los Angeles	85000	HR	NULL
1005	Mike Wang	8657848965	Seattle	75000	IT	NULL
1006	Emily Tan	807037722	Boston	65000	Sales	NULL
1007	Chris Kim	9423438778	Atlanta	55000	IT	NULL
1008	Sarah Lin	9690897749	Alaska	58000	IT	NULL
1009	David Warner	9699089774	Miami	60000	Sales	NULL

DROP: It is used to delete both the structure and record stored in a table.

ALTER TABLE Employee DROP EMAIL;

Employee_id	NAME	PHONE_NO	CITY	SALARY	DEPARTMENT
1001	John Doe	123456789	New York	60000	IT
1002	Marc Harris	7896478521	San Fran	70000	Sales
1003	Bob Smith	9874569785	Chicago	65000	Marketing
1004	Alice Lee	9856742587	Los Angeles	85000	HR
1005	Mike Wang	8657848965	Seattle	75000	IT
1006	Emily Tan	807037722	Boston	65000	Sales
1007	Chris Kim	9423438778	Atlanta	55000	IT
1008	Sarah Lin	9690897749	Alaska	58000	IT
1009	David Warner	9699089774	Miami	60000	Sales

QUERY 1: How to update salary with 70000 from table whose name is Mike Wang?

UPDATE Employee SET SALARY=1500000 WHERE NAME=" Mike Wang";

Employee_id	NAME	PHONE_NO	CITY	SALARY	DEPARTMENT
1001	John Doe	123456789	New York	60000	IT
1002	Marc Harris	7896478521	San Fran	70000	Sales
1003	Bob Smith	9874569785	Chicago	65000	Marketing
1004	Alice Lee	9856742587	Los Angeles	85000	HR
1005	Mike Wang	8657848965	Seattle	1500000	IT
1006	Emily Tan	807037722	Boston	65000	Sales
1007	Chris Kim	9423438778	anta	55000	IT
1008	Sarah Lin	9690897749	Alaska	58000	IT
1009	David Warner	9699089774	Miami	60000	Sales

Query 2: How to update all employee salaries with 80,000?

UPDATE Employee SET SALARY=80000;

Employee_id	NAME	PHONE_NO	CITY	SALARY	DEPARTMENT
1001	John Doe	123456789	New York	80000	IT
1002	Marc Harris	7896478521	San Fran	80000	Sales
1003	Bob Smith	9874569785	Chicago	80000	Marketing
1004	Alice Lee	9856742587	Los Angeles	80000	HR
1005	Mike Wang	8657848965	Seattle	80000	IT
1006	Emily Tan	807037722	Boston	80000	Sales
1007	Chris Kim	9423438778	Atlanta	80000	IT
1008	Sarah Lin	9690897749	Alaska	80000	IT
1009	David Warner	9699089774	Miami	80000	Sales

Conclusion: Hence from this we studied the implementation of DDl and DML commands in SQL.

Practical No.3

Aim: To Implement the Arithmetic and Comparison operations in SQL commands.

Theory:

Arithmetic Operations in SQL

We can use various arithmetic operators on the data stored in the tables. The SQL arithmetic operators perform the operation on the numerical columns in the table.

```
CREATE TABLE emp(
    emp_id int,
    emp_name varchar(20),
    emp_salary decimal(10,0),
    emp_bonus decimal(10,0),
    emp_position varchar(30)
);
 INSERT INTO emp
(emp_id,emp_name,emp_salary,emp_bonus,emp_position)
VALUES
(1,"Marcus",85536.02,4853.86,"Manager"),
(2,"Brown",70459.21,6714.3,"Developer"),
(3,"Raj",69391.18,5969.81,"Developer"),
(4,"Jeet",63004.42,4278.16,"Developer"),
(5,"Kanak",60640.31,3257.26,"Data Analyst"),
(6,"Rahil",64127.68,3231.88,"Marketing"),
(7,"Neel",65005.86,6643.76,"Marketing"),
(8,"Dev",54812.43,7646.25,"Tester"),
(9,"Jay",50171.48,2795.77,"Tester"),
(10,"Vijay",57838.68,2764.35,"Tester");
```

emp_id	emp_name	emp_salary	emp_bonus	emp_position
1	Marcus	85536	4854	Manager
2	Brown	70459	6714	Developer
3	Raj	69391	5970	Developer
4	Jeet	63004	4278	Developer
5	Kanak	60640	3257	Data Analyst
6	Rahil	64128	3232	Marketing
7	Neel	65006	6644	Marketing
8	Dev	54812	7646	Tester
9	Jay	50171	2796	Tester
10	Vijay	57839	2764	Tester

We can perform addition, subtraction, multiplication and division operations on the data stored in the SQL table.

Addition:

It is used to perform addition operations on the data items, items include either single column or multiple columns.

SELECT * , (emp_salary+emp_bonus) AS total_salary FROM emp;

emp_id	emp_name	emp_salary	emp_bonus	emp_position	total_salary
1	Marcus	85536	4854	Manager	90390
2	Brown	70459	6714	Developer	77173
3	Raj	69391	5970	Developer	75361
4	Jeet	63004	4278	Developer	67282
5	Kanak	60640	3257	Data Analyst	63897
6	Rahil	64128	3232	Marketing	67360
7	Neel	65006	6644	Marketing	71650
8	Dev	54812	7646	Tester	62458
9	Jay	50171	2796	Tester	52967
10	Vijay	57839	2764	Tester	60603

Subtraction:

It is used to perform subtraction operations on the data items, items include either single column or multiple columns.

SELECT * ,(emp_salary-5000) AS excluding_fundsFROM emp;

emp_id	emp_name	emp_salary	emp_bonus	emp_position	excluding_fund
1	Marcus	85536	4854	Manager	80536
2	Brown	70459	6714	Developer	65459
3	Raj	69391	5970	Developer	64391
4	Jeet	63004	4278	Developer	58004
5	Kanak	60640	3257	Data Analyst	55640
6	Rahil	64128	3232	Marketing	59128
7	Neel	65006	6644	Marketing	60006
8	Dev	54812	7646	Tester	49812
9	Jay	50171	2796	Tester	45171
10	Vijay	57839	2764	Tester	52839

Multiplication: It is used to perform multiplication operations on the data items.

SELECT * ,(emp_salary * 0.18) AS tax FROM emp;

emp_id	emp_name	emp_salary	emp_bonus	emp_position	tax
1	Marcus	85536	4854	Manager	15396.48
2	Brown	70459	6714	Developer	12682.62
3	Raj	69391	5970	Developer	12490.38
4	Jeet	63004	4278	Developer	11340.72
5	Kanak	60640	3257	Data Analyst	10915.20
6	Rahil	64128	3232	Marketing	11543.04
7	Neel	65006	6644	Marketing	11701.08
8	Dev	54812	7646	Tester	9866.16
9	Jay	50171	2796	Tester	9030.78
10	Vijay	57839	2764	Tester	10411.02

Division: It is used to get a reminder when one data is divided by another.

SELECT *,(emp_bonus/100) AS %_of_bonus FROM emp;

emp_id	emp_name	emp_salary	emp_bonus	emp_position	%_of_bonus
1	Marcus	85536	4854	Manager	48.5400
2	Brown	70459	6714	Developer	67.1400
3	Raj	69391	5970	Developer	59.7000
4	Jeet	63004	4278	Developer	42.7800
5	Kanak	60640	3257	Data Analyst	32.5700
6	Rahil	64128	3232	Marketing	32.3200
7	Neel	65006	6644	Marketing	66.4400
8	Dev	54812	7646	Tester	76.4600
9	Jay	50171	2796	Tester	27.9600
10	Vijay	57839	2764	Tester	27.6400

Comparison Operators

Greater than or equal to Operator:

It returns the rows / tuples which have the value of the attributes greater or equal than given value.

SELECT * FROM emp WHERE emp_salary >= 70000;

	emp_id	emp_name	emp_salary	emp_bonus	emp_position
▶	1	Marcus	85536	4854	Manager
	2	Brown	70459	6714	Developer

Less than or equal to Operator:

It returns the rows / tuples which have the value of the attributes lesser or equal than the given value.

SELECT * FROM emp WHERE emp_salary <= 60000;

	emp_id	emp_name	emp_salary	emp_bonus	emp_position
▶	8	Dev	54812	7646	Tester
	9	Jay	50171	2796	Tester
	10	Vijay	57839	2764	Tester

Equal to Operator:

It returns the row / tuples which have the same value of the attribute

SELECT * FROM emp WHERE emp_position =”Developer”;

	emp_id	emp_name	emp_salary	emp_bonus	emp_position
▶	2	Brown	70459	6714	Developer
	3	Raj	69391	5970	Developer
	4	Jeet	63004	4278	Developer

Not equal to Operator:

It return the row / tuples which is not same or equal to value of the attribute

SELECT * FROM emp WHERE emp_position <> “Tester”;

	emp_id	emp_name	emp_salary	emp_bonus	emp_position
▶	1	Marcus	85536	4854	Manager
	2	Brown	70459	6714	Developer
	3	Raj	69391	5970	Developer
	4	Jeet	63004	4278	Developer
	5	Kanak	60640	3257	Data Analyst
	6	Rahil	64128	3232	Marketing
	7	Neel	65006	6644	Marketing

Conclusion: Hence from this we studied the implementation of Arithmetic and Comparison operators in SQL.