Practical - 1

Aim: To Implement various DDl, DML commands and constraints

Theory:

DDL(Data Definition Language): DDL or 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. Examples of DDL commands:

- CREATE – is used to create the database or its objects (like table, index, function, views, store procedure and triggers).
- DROP – is used to delete objects from the database.
- ALTER-is used to alter the structure of the database.
- TRUNCATE–is used to remove all records from a table, including all spaces allocated for the records.
- COMMENT –is used to add comments to the data dictionary.
- RENAME –is used to rename an object existing in the database

A] DDL(Date Definition Language):

1] CREATE :

Code:

```
CREATE TABLE student_sachin (
id NUMBER(5) PRIMARY KEY,
Fname VARCHAR2(15),
Lname VARCHAR2(20),
Address VARCHAR2(50),
DOB DATE);
```

Output:

```
Table created.
```

2] ALTER :

Code:

```
ALTER TABLE student_sachin ADD Email VARCHAR2(55);
desc student_sachin;
```

Output:

```
Table altered.
```

TABLE STUDENT_SACHIN

Column	Null?	Type
ID	NOT NULL	NUMBER(5,0)
FNAME	-	VARCHAR2(15)
LNAME	-	VARCHAR2(20)
ADDRESS	-	VARCHAR2(50)
DOB	-	DATE
EMAIL	-	VARCHAR2(55)

Download CSV

6 rows selected.

3] TRUNCATE:

Code:

```
TRUNCATE TABLE student_sachin;
```

Output:

```
Table truncated.
```

4] RENAME:

```
RENAME student_sachin TO student_vesit_sachin;
```

Output:

```
Statement processed.
```

5] DROP :

Code:

```
DROP TABLE student_vesit_sachin;
```

Output:

```
Table dropped.
```

B] DML(Data Manipulation Language)

DML(Data Manipulation Language): The SQL commands that deal with the manipulation of data present in databases belong to DML or Data Manipulation Language and this includes most of the SQL statements. Examples of DML:

- SELECT – is used to retrieve data from a database.
- INSERT – is used to insert data into a table.
- UPDATE – is used to update existing data within a table.
- DELETE – is used to delete records from a database table.

1] SELECT :

Code:

```
SELECT * FROM student_sachin;
```

Output:

ID	FNAME	LNAME	ADDRESS	DOB	EMAIL
1	sachin	bairi	ulhasnagar	13-MAY-99	-
2	edbin	pillai	kalyan	11-FEB-99	-
3	satyan	sisodia	diva	01-JAN-00	-
4	abhishek	yadav	thane	13-FEB-99	-
5	jhon	deo	ulhasnagar	15-NOV-99	-

Download CSV

5 rows selected.

2] INSERT :

Code:

```
insert into student_sachin values(1, 'sachin', 'bairi', 'ulhasnagar', '13-MAY-1999');
insert into student_sachin values(2, 'edbin', 'pillai', 'kalyan', '11-FEB-1999');
insert into student_sachin values(3, 'satyan', 'sisodia', 'diva', '1-JAN-2000');
insert into student_sachin values(4, 'abhishek', 'jadhav', 'thane', '13-FEB-1999');
insert into student_sachin values(5, 'jhon', 'deo', 'ulhasnagar', '15-NOV-1999');
```

Output:

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

3] UPDATE:

Code:

```
UPDATE student_sachin SET Lname = 'yadav' WHERE id = 4;
```

Output:

```
1 row(s) updated.
```

4] DELETE:

Code:

```
DELETE FROM student_sachin WHERE id = 4;
SELECT * FROM student_sachin;
```

Output:

```
1 row(s) deleted.
```

ID	FNAME	LNAME	ADDRESS	DOB	EMAIL
1	sachin	bairi	ulhasnagar	13-MAY-99	-
2	edbin	pillai	kalyan	11-FEB-99	-
3	satyan	sisodia	diva	01-JAN-00	-
5	jhon	deo	ulhasnagar	15-NOV-99	-

Download CSV

4 rows selected.

C] SQL Constraints

SQL constraints: It is used to specify rules for the data in a table. Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted. Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table. The following constraints are commonly used in SQL:

- NOT NULL - Ensures that a column cannot have a NULL value
- UNIQUE - Ensures that all values in a column are different

- PRIMARY KEY - A combination of NOT NULL and UNIQUE. Uniquely identifies each row in a table
- FOREIGN KEY - Uniquely identifies a row/record in another table
- CHECK - Ensures that all values in a column satisfies a specific condition
- DEFAULT - Sets a default value for a column when no value is specified

1] PRIMARY KEY:

Code:

```
CREATE TABLE student_sachin (
id NUMBER(5) PRIMARY KEY,
Fname VARCHAR2(15),
Lname VARCHAR2(20),
Address VARCHAR2(50),
DOB DATE);

insert into student_sachin values(1, 'sachin', 'bairi', 'ulhasnagar', '13-MAY-1999');
insert into student_sachin values(2, 'edbin', 'pillai', 'kalyan', '11-FEB-1999');
insert into student_sachin values(3, 'satyan', 'sisodia', 'diva', '1-JAN-2000');
insert into student_sachin values(4, 'abhishek', 'jadhav', 'thane', '13-FEB-1999');
insert into student_sachin values(5, 'jhon', 'deo', 'ulhasnagar', '15-NOV-1999');

desc student_sachin;

select * from student_sachin;
```

Output :

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

TABLE STUDENT_SACHIN

Column	Null?	Type
ID	NOT NULL	NUMBER(5,0)
FNAME	-	VARCHAR2(15)
LNAME	-	VARCHAR2(20)
ADDRESS	-	VARCHAR2(50)
DOB	-	DATE

Download CSV

5 rows selected.

ID	FNAME	LNAME	ADDRESS	DOB
1	sachin	bairi	ulhasnagar	13-MAY-99
2	edbin	pillai	kalyan	11-FEB-99
3	satyan	sisodia	diva	01-JAN-00
4	abhishek	jadhav	thane	13-FEB-99
5	jhon	deo	ulhasnagar	15-NOV-99

Download CSV

5 rows selected.

2] FOREIGN KEY:

Code :

```
CREATE TABLE supplier_sachin(
supplier_id numeric(10) not null,
supplier_name varchar2(50) not null,
contact_name varchar2(50),
CONSTRAINT supplier_sachin_pk PRIMARY KEY (supplier_id)
);

CREATE TABLE products_sachin (
product_id numeric(10) not null,
supplier_id numeric(10) not null,
CONSTRAINT fk_supplier
FOREIGN KEY (supplier_id)
REFERENCES supplier_sachin (supplier_id)
);

desc supplier_sachin;
desc products_sachin;

insert into supplier_sachin values(1, 'sai supplier','sachin');
insert into supplier_sachin values(2, 'rahul supplier','satyan');
insert into supplier_sachin values(3, 'shiv supplier','ronak');
insert into supplier_sachin values(4, 'swami supplier','jhon');
insert into supplier_sachin values(5, 'atharva supplier','leo');

insert into products_sachin values(1, 2);
insert into products_sachin values(2, 5);
insert into products_sachin values(3, 1);
insert into products_sachin values(4, 3);
insert into products_sachin values(5, 4);

select * from supplier_sachin;
select * from products_sachin;
```

Output :

```
Table created.


Table created.
```

TABLE SUPPLIER_SACHIN

Column	Null?	Type
SUPPLIER_ID	NOT NULL	NUMBER(10,0)
SUPPLIER_NAME	NOT NULL	VARCHAR2(50)
CONTACT_NAME	-	VARCHAR2(50)

Download CSV

3 rows selected.

TABLE PRODUCTS_SACHIN

Column	Null?	Type
PRODUCT_ID	NOT NULL	NUMBER(10,0)
SUPPLIER_ID	NOT NULL	NUMBER(10,0)

Download CSV
2 rows selected.

SUPPLIER_ID	SUPPLIER_NAME	CONTACT_NAME
1	sai supplier	sachin
2	rahul supplier	satyan
3	shiv supplier	ronak
4	swami supplier	jhon
5	atharva supplier	leo

Download CSV
5 rows selected.

PRODUCT_ID	SUPPLIER_ID
1	2
2	5
3	1
4	3
5	4

Download CSV
5 rows selected

3] UNIQUE KEY :

Code:

```
CREATE TABLE employee_sachin (
employee_id numeric (10) not null,
employee_name varchar2 (50) not null,
CONSTRAINT employee_id_unique UNIQUE (employee_id)
);

desc employee_sachin;
insert into employee_sachin values(1, 'narender');
insert into employee_sachin values(1, 'neel');

select * from employee_sachin;
```

Output:

```
Table created.
```

```
TABLE EMPLOYEE_SACHIN
```

Column	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(10,0)
EMPLOYEE_NAME	NOT NULL	VARCHAR2(50)

```
Download CSV
2 rows selected.
```

Unique Constraint is used to set unique value of the particular Field
In this example, the employee_id is set as unique value
So, when the same value is inserted again It throws an error.

```
1 row(s) inserted.

ORA-00001: unique constraint (SQL_ECBIIHWOQCDZHYBHZNZSHXYOL.EMPLOYEE_ID_UNIQUE) violated ORA-06512: at "SYS.DBMS_SQL",
line 1721
```

EMPLOYEE_ID	EMPLOYEE_NAME
1	narender

```
Download CSV
```

4] DEFAULT KEY :

Code:

```
CREATE TABLE customers_sachin (
c_id numeric(10) not null,
c_name varchar2(55) not null,
c_country varchar2(55) DEFAULT 'INDIA'
);

desc customers_sachin;

insert into customers_sachin(c_id, c_name) values(1, 'narender');
insert into csutomers_sachin(c_id, c_name, c_country) values(2, 'neel, 'germany');

select * from customers_sachin;
```

Output:

```
Table created.
```

TABLE CUSTOMERS_SACHIN

Column	Null?	Type
C_ID	NOT NULL	NUMBER(10,0)
C_NAME	NOT NULL	VARCHAR2(55)
C_COUNTRY	-	VARCHAR2(55)

Download CSV

3 rows selected.

```
1 row(s) inserted.
```

Default constraint is used to set the default value for a particular field.
In this example, the c_country is set to the default value of 'INDIA'
If we do not specify the value of the c_country then the default value will be 'INDIA'

C_ID	C_NAME	C_COUNTRY
1	sachin	INDIA

Download CSV

5] CHECK KEY :

Code:

```
CREATE TABLE elections_sachin (
e_id numeric(10) not null,
e_name varchar2(55) not null,
e_age numeric(10),
CONSTRAINT check_age CHECK (e_age >= 18)
);

desc elections_sachin;

insert into elections_sachin values(1, 'sachin', 22);
insert into elections_sachin values(2, 'ganesh', 15)

select * from elections_sachin;
```

Output:

The check constraint is used to check the condition If the condition is true then only allow to insert the values, else it will throw an error. In this example, we have checked the age of the

person who is eligible to vote. So, if the person's age is less than 18 then it will not insert the value in the table.

```
Table created.
```

```
TABLE ELECTIONS_SACHIN
```

Column	Null?	Type
E_ID	NOT NULL	NUMBER(10,0)
E_NAME	NOT NULL	VARCHAR2(55)
E_AGE	-	NUMBER(10,0)

```
Download CSV
3 rows selected.
```

```
ORA-02290: check constraint (SQL_IZGZXKBRMSEBAZVRAPTMQQIHV.CHECK_AGE) violated ORA-06512: at "SYS.DBMS_SQL", line 1721
```

6] NOT NULL :

Code:

```
CREATE TABLE student_sachin (
id NUMBER(5) PRIMARY KEY,
Fname VARCHAR2(15) NOT NULL,
Lname VARCHAR2(20),
Address VARCHAR2(50),
DOB DATE );

desc student_sachin;

insert into student_sachin values(1, 'sachin', 'bairi', 'kalyan', '10-MAY-1999');
insert into student_sachin values(2, '', 'jadhav', 'uran', '31-JAN-2000');

select * from student_sachin;
```

Output :

```
Table created.
```

TABLE STUDENT_SACHIN

Column	Null?	Type
ID	NOT NULL	NUMBER(5,0)
FNAME	NOT NULL	VARCHAR2(15)
LNAME	-	VARCHAR2(20)
ADDRESS	-	VARCHAR2(50)
DOB	-	DATE

Download CSV

5 rows selected.

```
1 row(s) inserted.
```

```
ORA-01400: cannot insert NULL into ("SQL_XBRHBKONXFOUDUJKEJLDXZSWT"."STUDENT_SACHIN"."FNAME") ORA-06512: at
"SYS.DBMS_SQL", line 1721
```

ID	FNAME	LNAME	ADDRESS	DOB
1	sachin	bairi	kalyan	10-MAY-99

Download CSV

Conclusion :

I have learned the basics of DML, DDL, SQL Constraints from this assignment.