ADBMS PRACTICAL NO: 0

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:

```
SQL> CREATE TABLE student_prasanna(
  2  id NUMBER(5) PRIMARY KEY,
  3  Fname VARCHAR2(15),
  4  Lname VARCHAR2(15),
  5  Address VARCHAR2(50),
  6  DOB DATE);

Table created.
```

2] ALTER :

Code:

```
SQL> ALTER TABLE student_prasanna ADD Email VARCHAR2(55);

Table altered.
```

```
Name                                      Null?    Type
----------------------------------------- -------- ---------------
ID                                        NOT NULL NUMBER(5)
FNAME                                              VARCHAR2(15)
LNAME                                              VARCHAR2(15)
ADDRESS                                            VARCHAR2(50)
DOB                                                DATE
EMAIL                                              VARCHAR2(55)
```

3] TRUNCATE:

Code:

```
SQL> RENAME student_prasanna to student_vesit_prasanna;

Table renamed.
```

4] RENAME:

Code:

```
SQL> RENAME student_prasanna to student_vesit_prasanna;

Table renamed.
```

5] DROP :

Code:

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:

```
SQL> select * from student_prasanna;

        ID FNAME           LNAME           ADDRESS                                                 DOB
---------- --------------- --------------- ------------------------------------------------------- ---------
         1 Prasanna        Sawant          Kharghar                                                22-FEB-01
         2 Ghanshyam       Patel           Panvel                                                  02-AUG-01
         3 Abhishek        Singh           Nerul                                                   01-SEP-01
         4 Amey            Mhatre          Vashi                                                   21-JUN-01
         5 Kashyap         Sugandh         Chembur                                                 21-DEC-01
```

2] INSERT :

Code:

```
SQL> INSERT INTO student_prasanna VALUES(1, 'Prasanna', 'Sawant', 'Kharghar', '22-FEB-2001');

1 row created.
```

3] UPDATE:

Code:

```
SQL> UPDATE student_prasanna SET Lname = 'Patil' WHERE id = 4;

1 row updated.
```

4] DELETE:

Code:

```
SQL> DELETE FROM student_prasanna WHERE id = 4;

1 row deleted.

SQL> select * from student_prasanna;

        ID FNAME           LNAME           ADDRESS                                                DOB
---------- --------------- --------------- ------------------------------------------------------ ---------
         1 Prasanna        Sawant          Kharghar                                               22-FEB-01
         2 Ghanshyam       Patel           Panvel                                                 02-AUG-01
         3 Abhishek        Singh           Nerul                                                  01-SEP-01
         5 Kashyap         Sugandh         Chembur                                                21-DEC-01
```

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_prasanna (
    id NUMBER(5) PRIMARY KEY,
    Fname VARCHAR2(15),
    Lname VARCHAR(20),
    Address VARCHAR2(50),
    DOB DATE);

insert into student_prasanna values(1,'prasanna','sawant','kharghar','22-FEB-2001');
insert into student_prasanna values(2,'ghanshyam','patel','nerul','02-AUG-2002');
insert into student_prasanna values(3,'abhishek','singh','panvel','01-SEP-2000');
insert into student_prasanna values(4,'amey','mhatre','vashi','21-JUL-2001');
insert into student_prasanna values(5,'kashyap','shinde','chembur','27-DEC-2000');

desc student_prasanna;

select * from student_prasanna;
```

Output :

TABLE STUDENT_PRASANNA

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	prasanna	sawant	kharghar	22-FEB-01
2	ghanshyam	patel	nerul	02-AUG-02
3	abhishek	singh	panvel	01-SEP-00
4	amey	mhatre	vashi	21-JUL-01
5	kashyap	shinde	chembur	27-DEC-00

Download CSV

5 rows selected.

2] FOREIGN KEY:

Code :

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

CREATE TABLE products_prasanna (
    product_id numeric(10) not null,
    supplier_id numeric(10) not null,
    CONSTRAINT fk_supplier
    FOREIGN KEY (supplier_id)
    REFERENCES supplier_prasanna (supplier_id)
);
```

```
desc supplier_prasanna;
desc products_prasanna;

insert into supplier_prasanna values(1,'sai supplier','prasanna');
insert into supplier_prasanna values(2,'rahul supplier','ghanshyam');
insert into supplier_prasanna values(3,'shiv supplier','abhishek');
insert into supplier_prasanna values(4,'swami supplier','amey');
insert into supplier_prasanna values(5,'jay supplier','kashyap');

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

select * from supplier_prasanna;
select * from products_prasanna;
```

Output :

```
Table created.

Table created.
```

TABLE SUPPLIER_PRASANNA

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_PRASANNA

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

SUPPLIER_ID	SUPPLIER_NAME	CONTACT_NAME
1	sai supplier	prasanna
2	rahul supplier	ghanshyam
3	shiv supplier	abhishek
4	swami supplier	amey
5	jay supplier	kashyap

Download CSV

5 rows selected.

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

3] UNIQUE KEY :

Code:

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

desc employee_prasanna;
insert into employee_prasanna values(1,'prasanna');
insert into employee_prasanna values(1,'ghanshyam');

select * from employee_prasanna;
```

Output:

```
Table created.
```

TABLE EMPLOYEE_PRASANNA

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 a unique value.So, when the same value is inserted again
It throws an error.

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

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

EMPLOYEE_ID	EMPLOYEE_NAME
1	prasanna

4] DEFAULT KEY :

Code:

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

desc customers_prasanna;

insert into customers_prasanna(c_id, c_name) values(1,'prasanna');
insert into customers_prasanna(c_id, c_name, c_country) values(2,'ghanshyam','germany');

select * from customers_prasanna;
```

Output:

```
Table created.
```

TABLE CUSTOMERS_PRASANNA

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.

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	prasanna	INDIA
2	ghanshyam	germany

Download CSV

```
2 rows selected.
```

5] CHECK KEY :

Code:

```
CREATE TABLE elections_prasanna (
    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_prasanna;

insert into elections_prasanna values(1,'prasanna','22');
insert into elections_prasanna values(2,'ghanshyam','17');

select * from elections_prasanna;
```

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_PRASANNA

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.

1 row(s) inserted.

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

E_ID	E_NAME	E_AGE
1	prasanna	22

6] NOT NULL :

Code:

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

desc student_prasanna;

insert into student_prasanna values(1,'prasanna','sawant','kharghar','22-FEB-2001');
insert into student_prasanna values(2,'ghanshyam','patel','nerul','02-AUG-2002');

select * from student_prasanna;
```

Output :

```
Table created.
```

TABLE STUDENT_PRASANNA

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

```
5 rows selected.

1 row(s) inserted.

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

ID	FNAME	LNAME	ADDRESS	DOB
1	prasanna	sawant	kharghar	22-FEB-01

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