### EX.NO:1 IMPLEMENTATION OF DDL COMMANDS

**AIM:**

To execute and verify the Data Definition Language commands.

### PROCEDURE

STEP 1: Start

STEP 2: Create the table with its essential attributes.

STEP 3: Execute different Commands and extract information from the table.

STEP 4: Stop

### DDL COMMANDS:

1. The Create Table Command: -

It defines each column of the table uniquely. Each column has minimum of three attributes, a name, data type and size.

### Syntax:

Create table <table name> (<col1> <datatype>(<size>),<col2> <datatype>(<size>));

Ex: create table emp(empno number(4) primary key, ename char(10));

Table created.

1. Modifying the structure of tables.
   1. Add new columns

### Syntax:

Alter table <tablename> add(<new col><datatype>(size),<new col><datatype>(size));

Ex: alter table emp add(sal number(7,2));

Table altered.

Name Null? Type

EMPNO NOT NULL NUMBER(4) ENAME CHAR(10)

SAL NUMBER(7,2)

1. Dropping a column from a table.

### Syntax:

Alter table <tablename> drop column <col>;

Ex : alter table emp drop column sal;

Table altered.

Name Null? Type

EMPNO NOTNULL NUMBER(4)

ENAME CHAR(10)

1. Modifying existing columns.

### Syntax:

Alter table <tablename> modify(<col><newdatatype>(<newsize>)); Ex: alter table emp modify(ename varchar2(15));

Table altered.

Name Null? Type

EMPNO NOT NULL NUMBER(4) ENAME VARCHAR2(15)

1. Renaming the tables

### Syntax:

Rename <oldtable> to <new table>;

Ex: rename emp to emp1;

Table renamed.

Name Null? Type

EMPNO NOT NULL NUMBER(4) ENAME VARCHAR2(15)

1. Truncating the tables.

### Syntax:

Truncate table <tablename>;

Ex: truncate table emp1;

Table truncated.

Name Null? Type

EMPNO NOT NULL NUMBER(4)

ENAME VARCHAR2(15)

1. Destroying tables.

### Syntax:

Drop table <tablename>;

Ex: drop table emp1;

Table dropped.

### CONSTRAINTS:

Create table tablename (column\_name1 data\_ type constraints, column\_name2 data\_ type constraints …)

### Example:

Create table stud1(sname varchar2(20) not null, rollno number(10) not null,dob date not null);

### DOMAIN INTEGRITY

**Example:** Create table cust(custid number(6) not null, name char(10));

Alter table cust modify (name not null);

### ENTITY INTEGRITY

Primary Key Constraint:

Example: Create table stud2(regno number(6) primary key, name char(20));

### RESULT:

Thus the DDL commands have been executed successfully.

### EX.NO:2 IMPLEMENTATION OF DML COMMANDS

**AIM:**

To execute and verify the DML and commands.

## PROCEDURE

STEP 1: Start

STEP 2: Create the table with its essential attributes. STEP 3: Insert the record into table

STEP 4: Update the existing records into the table STEP 5: Delete the records in to the table

STEP 6: Stop

### DML COMMANDS

DML commands are the most frequently used SQL commands and is used to query and manipulate the existing database objects. Some of the commands are Insert, Select, Update, Delete.

**Insert Command:** This is used to add one or more rows to a table. The values are separated by commas and the data types char and date are enclosed in apostrophes. The values must be entered in the same order as they are defined.

**Select Commands**: It is used to retrieve information from the table. It is generally referred to as querying the table. We can either display all columns in a table or only specify column from the table.

**Update Command**: It is used to alter the column values in a table. A single column may be updated or more than one column could be updated.

**Delete command**: After inserting row in a table we can also delete them if required. The delete command consists of a from clause followed by an optional where clause.

Q1: create table emp(empno number(6) primary key,ename varchar2(20),job varchar2(13),deptno number(3),sal number(7,2));

Table created.

Q2: Insert more than a record into emp table using a single insert command.

insert into emp values(1,'Mathi','AP',1,10000)

1 row created.

insert into emp values(2,'Arjun','ASP',2,12000)

1 row created.

insert into emp values(3,'Gugan','ASP',1,12000)

1 row created.

Q3: Update the emp table to set the salary of all employees to Rs15000/- who are working as ASP

SQL> select \* from emp;

EMPNO ENAME JOB DEPTNO SAL

1. Mathi AP 1 10000
2. Arjun ASP 2 12000
3. Gugan ASP 1 12000

SQL> update emp set sal=15000 where job='ASP'; 2 rows updated.

SQL> select \* from emp;

EMPNO ENAME JOB DEPTNO SAL

1. Mathi AP 1 10000
2. Arjun ASP 2 15000
3. Gugan ASP 1 15000

insert into emp values(4,'Karthik','Prof',2,30000)

1row created.

insert into emp values(5,'Akalya','AP',1,10000)

1 row created.

insert into emp values(6,'suresh','lect',1,8000)

1 row created.

SQL> select \* from emp;

EMPNO ENAME JOB DEPTNO SAL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Mathi | AP | 1 | 10000 |
| 2 | Arjun | ASP | 2 | 15000 |
| 3 | Gugan | ASP | 1 | 15000 |
| 4 | Karthik | Prof | 2 | 30000 |
| 5 | Akalya | AP | 1 | 10000 |
| 6 | suresh | lect | 1 | 8000 |

6 rows selected.

Q4: Create a pseudo table employee with the same structure as the table emp and insert rows into the table using select clauses.

SQL> create table employee as select \* from emp;

Table created.

SQL> desc employee; Name Null? Type

EMPNO NUMBER(6)

ENAME NOT NULL VARCHAR2(20) JOB NOT NULL VARCHAR2(13) DEPTNO NUMBER(3)

SAL NUMBER(7,2)

Q5: select employee name, job from the emp table

SQL> select ename, job from emp;

ENAME JOB

Mathi AP

Arjun ASP

Gugan ASP

Karthik Prof

Akalya AP

suresh lect 6 rows selected.

Q6: Delete only those who are working as lecturer

SQL> select \* from emp;

EMPNO ENAME JOB DEPTNO SAL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Mathi | AP |  | 1 |  | 10000 |
| 2 | Arjun | ASP |  | 2 |  | 15000 |
| 3 | Gugan | ASP |  | 1 |  | 15000 |
| 4 | Karthik | Prof |  | 2 |  | 30000 |
| 5 | Akalya | AP |  | 1 |  | 10000 |
| 6 | suresh | lect |  | 1 |  | 8000 |

6 rows selected.

SQL> delete from emp where job='lect'; 1 row deleted.

SQL> select \* from emp;

EMPNO ENAME JOB DEPTNO SAL

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | Mathi |  | AP | 1 | 10000 |
| 2 | Arjun |  | ASP | 2 | 15000 |
| 3 | Gugan |  | ASP | 1 | 15000 |
| 4 | Karthik |  | Prof | 2 | 30000 |
| 5 | Akalya |  | AP | 1 | 10000 |

Q7: List the records in the emp table orderby salary in ascending order. SQL> select \* from emp order by sal;

EMPNO ENAME JOB DEPTNO SAL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Mathi |  | AP |  | 1 | 10000 |
| 5 | Akalya |  | AP |  | 1 | 10000 |
| 2 | Arjun |  | ASP |  | 2 | 15000 |
| 3 | Gugan |  | ASP |  | 1 | 15000 |
| 4 | Karthik |  | Prof |  | 2 | 30000 |

Q8: List the records in the emp table orderby salary in descending order.

SQL> select \* from emp order by sal desc;

EMPNO ENAME JOB DEPTNO SAL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 4 | Karthik | Prof |  | 2 |  | 30000 |
| 2 | Arjun | ASP |  | 2 |  | 15000 |
| 3 | Gugan | ASP |  | 1 |  | 15000 |
| 1 | Mathi | AP |  | 1 |  | 10000 |
| 5 | Akalya | AP |  | 1 |  | 10000 |

Q9: Display only those employees whose deptno is 1. SQL> select \* from emp where deptno=1;

EMPNO ENAME JOB DEPTNO SAL

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Mathi | AP | 1 10000 |
| 3 | Gugan | ASP | 1 15000 |
| 5 | Akalya | AP | 1 10000 |

Q10: Display deptno from the table employee avoiding the duplicated values. SQL> select distinct deptno from emp;

DEPTNO

1

2

### IMPLEMENTATION OF DATA AND BUILT IN FUNCTIONS IN SQL CHARACTER/STRING FUNCTION:

SQL> select upper('hai') from dual;

UPP

--- HAI

SQL> select lower('HAI') from dual; LOW

---

hai

SQL> select initcap(‘hello world') from dual; INITCAP('Hello’)

Hello World

SQL> select ltrim(' hai') from dual;

LTR

---

hai

SQL> select rtrim('hai ')from dual; RTR

---

hai

SQL> select rtrim(' hai ')from dual; RTRIM('

hai

SQL> select concat('SRM',' university')from dual;

SRM university

SQL> select length('SRM’)from dual; LENGTH('SRM')

12

SQL> select replace('SRM university', 'SRM','Anna')from dual;

Anna university

SQL> select substr('SRM', 7,6)from dual;

SUBSTR

lingam

SQL> select rpad('hai',3,'\*')from dual; RPAD('

hai\*\*\*

SQL> select lpad('hai',3,'\*')from dual; LPAD('

\*\*\*hai

SQL> select replace('Dany','y','ie')from dual; REPLACE

Danie

SQL> select translate('cold','ld','ol')from dual; TRANSL

Cool

**DATE & TIME FUNCTION**

SQL> select sysdate from dual; SYSDATE

07-APR-10

SQL> select round(sysdate)from dual; ROUND(SYS

07-APR-10

SQL> select add\_months(sysdate,3)from dual; ADD\_MONTH

07-JUL-10

SQL> select last\_day(sysdate)from dual; LAST\_DAY(

30-APR-10

SQL> select sysdate+20 from dual; SYSDATE+2

27-APR-10

SQL> select next\_day(sysdate,'tuesday')from dual; NEXT\_DAY(

13-APR-10

**NUMERIC FUNCTION**

SQL> select round(15.6789) from dual; ROUND(15.6789)

16

SQL> select ceil(23.20)from dual; CEIL(23.20)

24

SQL> select floor(34.56)from dual; FLOOR(34.56)

34

SQL> select trunc(15.56743)from dual; TRUNC(15.56743)

15

SQL> select sign(-345)from dual; SIGN(-345)

-1

SQL> select abs(-70)from dual; ABS(-70)

70

**MATH FUNCTION:**

SQL> select abs(45) from dual; ABS(45)

45

SQL> select power(10,12) from dual; POWER(10,12)

1.000E+12

SQL> select mod(11,5) from dual; MOD(11,5)

1

SQL> select exp(10) from dual; EXP(10)

22026.466

SQL> select sqrt(225) from dual; SQRT(225)

15

### SET OPERATORS QUERIES:

SQL> create table dept(dno number(10),dname varchar(10),loc varchar(10)); Table created.

SQL> insert into dept values(10,'inventory','hyd'); 1 row created.

SQL> insert into dept values(20,'finance','bglr'); 1 row created.

SQL> insert into dept values(30,'HR','mumbai'); 1 row created.

SQL> select \* from dept; DNO DNAME LOC

10 inventory hyd

20 finance bglr

30 HR mumbai

Q1: Display all the dept numbers available with the dept and emp tables avoiding duplicates.

### Solution:

SQL> select deptno from emp union select deptno from dept; DEPTNO

1

2

12

30

40

Q2: Display all the dept numbers available with the dept and emp tables. SQL> select deptno from emp union all select deptno from dept; DEPTNO

1

2

2

1

12

1

2

30

40

9 rows selected.

Q3: Display all the dept numbers available in emp and not in dept tables and vice versa. SQL> select deptno from emp minus select deptno from dept;

DEPTNO

12

SQL> select deptno from dept minus select deptno from emp; DEPTNO

30

40

### RESULT

Thus the DML commands was performed successfully and executed.