

Ex. No: 1

SQL BASIC COMMANDS

Date:

AIM:

To write SQL queries to execute basic SQL commands.

QUERIES:

1. Create table

Query:

```
CREATE TABLE emp
(
    empno NUMBER,
    empname VARCHAR2(255),
    DOB DATE,
    salary NUMBER,
    designation VARCHAR2(20)
);
```

Output:

Table created.

2. Insert values

Query:

```
INSERT INTO emp VALUES(100,'John','4.21.1994', 50000,'Manager');
```

```
INSERT INTO emp VALUES(101,'Greg','6.20.1994',25000,'Clerk');
```

Output:

2 rows inserted

3. Display values

Query:

```
SELECT * FROM emp;
```

Output:

EMPNAM				
EMPNO	E	DOB	SALARY	DESIGNATION
100	John	04/21/1994	50000	Manager

101	Greg	06/20/1994	25000	Clerk
-----	------	------------	-------	-------

Query:

```
SELECT empname,salary FROM emp;
```

Output:

EMPNAME	SALARY
John	50000
Greg	25000

4. Modify values

Query:

```
UPDATE emp SET salary = salary + 1000;
```

Output:

2 row(s) updated.

Query:

```
SELECT * FROM emp;
```

Output:

EMPNO	EMPNAME	DOB	SALARY	DESIGNATION
100	John	04/21/1994	51000	Manager
101	Greg	06/20/1994	26000	Clerk

5. Delete values

Query:

```
DELETE FROM emp WHERE empno = 100;
```

Output:

1 row(s) deleted.

Query:

```
SELECT * FROM emp;
```

Output:

EMPNO	EMPNAME	DOB	SALARY	DESIGNATION
101	Greg	06/20/1994	26000	Clerk

RESULT:

Thus the basic SQL queries were successfully executed and verified.

Ex. No: 2 DATA DEFINITION LANGUAGE (DDL)

Date :

AIM:

To write the SQL queries using DDL Commands with and without constraints.

DDL STATEMENTS

- CREATE TABLE
- ALTER TABLE
- DROP TABLE

SYNTAX:

1. Create Table

The CREATE TABLE statement is used to create a relational table

```
CREATE TABLE table_name
(
    column_name1 data_type [constraints],
    column_name1 data_type [constraints],
    column_n
    ame1 data_type [constraints],
```

```
);
```

2. Alter Table

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table

a. To Add a column

```
ALTER TABLE table_name ADD column_name datatype
```

b. To delete a column in a table

```
ALTER TABLE table_name DROP (column_name)
```

c. To change the data type of a column in a table

```
ALTER TABLE table_name MODIFY(column_name datatype )
```

3. Drop Table

Used to delete the table permanently from the storage

```
DROP TABLE table_name
```

QUERIES:

1. CREATE THE TABLE (with no constraint)

Query:

```
CREATE TABLE emp
(
    empno NUMBER,
    empname VARCHAR2(25),
    dob DATE,
    salary NUMBER,
    designation VARCHAR2(20)
);
```

Output:

Table Created

Query:

```
DESC emp;
```

Output:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>EMP</u>	<u>EMPNO</u>	NUMBER	22	-	-	-	-	-	-
	<u>EMPNAME</u>	VARCHAR2	255	-	-	-	-	-	-
	<u>DOB</u>	DATE	7	-	-	-	-	-	-
	<u>SALARY</u>	NUMBER	22	-	-	-	-	-	-
	<u>DESIGNATION</u>	VARCHAR2	20	-	-	-	-	-	-

2. ALTER THE TABLE

a. ADD

// To alter the table emp by adding new attribute department

Query:

```
ALTER TABLE emp ADD department VARCHAR2(50);
```

Output:

Table Altered

Query:

```
DESC emp;
```

Output:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>EMP</u>	<u>EMPNO</u>	NUMBER	22	-	-	-	-	-	-
	<u>EMPNAME</u>	VARCHAR2	255	-	-	-	-	-	-
	<u>DOB</u>	DATE	7	-	-	-	-	-	-
	<u>SALARY</u>	NUMBER	22	-	-	-	-	-	-
	<u>DESIGNATION</u>	VARCHAR2	20	-	-	-	-	-	-
	<u>DEPARTMENT</u>	VARCHAR2	50	-	-	-	-	-	-

b. MODIFY

//To alter the table emp by modifying the size of the attribute department

Query:

```
ALTER TABLE emp MODIFY (department VARCHAR2(100));
```

Output:

Table Altered

Query:

```
DESC emp;
```

Output:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>EMP</u>	<u>EMPNO</u>	NUMBER	22	-	-	-		-	-
	<u>EMPNAME</u>	VARCHAR2	255	-	-	-		-	-
	<u>DOB</u>	DATE	7	-	-	-		-	-
	<u>SALARY</u>	NUMBER	22	-	-	-		-	-
	<u>DESIGNATION</u>	VARCHAR2	20	-	-	-		-	-
	<u>DEPARTMENT</u>	VARCHAR2	100	-	-	-		-	-

c. DROP

// To alter the table emp by deleting the attribute department

Query:

```
ALTER TABLE emp DROP(department);
```

Output:

Table Altered

Query:

```
DESC emp;
```

Output:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>EMP</u>	<u>EMPNO</u>	NUMBER	22	-	-	-		-	-
	<u>EMPNAME</u>	VARCHAR2	255	-	-	-		-	-
	<u>DOB</u>	DATE	7	-	-	-		-	-
	<u>SALARY</u>	NUMBER	22	-	-	-		-	-
	<u>DESIGNATION</u>	VARCHAR2	20	-	-	-		-	-

d. RENAME

// To alter the table name by using rename keyword

Query:

```
ALTER TABLE emp RENAME TO emp1 ;
```

Output:

Table Altered

Query:

```
DESC emp1;
```

Output:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>EMP1</u>	<u>EMPNO</u>	NUMBER	22	-	-	-	-	-	-
	<u>EMPNAME</u>	VARCHAR2	255	-	-	-	-	-	-
	<u>DOB</u>	DATE	7	-	-	-	-	-	-
	<u>SALARY</u>	NUMBER	22	-	-	-	-	-	-
	<u>DESIGNATION</u>	VARCHAR2	20	-	-	-	-	-	-
	<u>DEPARTMENT</u>	VARCHAR2	100	-	-	-	-	-	-

3. DROP

//To delete the table from the database

Query:

```
DROP TABLE emp1;
```

Output:

Table Dropped

Query:

```
DESC emp1;
```

Output:

Object to be described could not be found.

CONSTRAINT TYPES:

- NOT NULL

- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK
- DEFAULT

QUERIES:

1. CREATE THE TABLE

Query:

```
CREATE TABLE student
(
    studentID NUMBER PRIMARY KEY,
    sname VARCHAR2(30) NOT NULL,
    department CHAR(5),
    sem NUMBER,
    dob DATE,
    email_id VARCHAR2(20) UNIQUE,
    college VARCHAR2(20) DEFAULT 'MEC'
);
```

Output:

Table created.

Query:

```
DESC student;
```

Output:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>STUDENT</u>	<u>STUDENTID</u>	NUMBER	22	-	-	1	-	-	-
	<u>SNAME</u>	VARCHAR2	30	-	-	-	-	-	-
	<u>DEPARTMENT</u>	CHAR	5	-	-	-	✓	-	-
	<u>SEM</u>	NUMBER	22	-	-	-	✓	-	-
	<u>DOB</u>	DATE	7	-	-	-	✓	-	-

<u>EMAIL ID</u>	VARC HAR2	20	-	-	-	✓	-	-
<u>COLLE GE</u>	VARC HAR2	20	-	-	-	✓	'ME C'	-

Query:

```
CREATE TABLE exam
(
    examID NUMBER ,
    studentID NUMBER REFERENCES student(studentID),
    department CHAR(5) NOT NULL,
    mark1 NUMBER CHECK (mark1<=100 and mark1>=0),
    mark2 NUMBER CHECK (mark2<=100 and mark2>=0),
    mark3 NUMBER CHECK (mark3<=100 and mark3>=0),
    mark4 NUMBER CHECK (mark4<=100 and mark4>=0),
    mark5 NUMBER CHECK (mark5<=100 and mark5>=0),
    total NUMBER,
    average NUMBER,
    grade CHAR(1)
);
```

Output:

Table created.

//To alter the table student by adding new constraint to the examID attribute

Query:

```
ALTER TABLE student ADD CONSTRAINT pr
PRIMARY KEY (examid);
```

Output:

Table altered.

2. CREATE THE TABLE USING COMPOSITE PRIMARY KEY

Create the following table with the attributes reg_no and stu_name as primary key.

stu_details (reg_no, stu_name, DOB, address, city)

Query:

```
CREATE TABLE stu_details
(
    reg_no number,
    stu_name varchar2(30),
    DOB date,
    address varchar2(30),
    city char(30),
    primary key(reg_no, stu_name)
);
```

Output:

Table created.

Query:

```
DESCstu_details
```

Output:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
-------	--------	-----------	--------	-----------	-------	-------------	----------	---------	---------

<u>STU D</u> <u>ETAILS</u>	<u>REG</u> <u>NO</u>	NUMB ER	22	-	-	1	-	-	-
	<u>STU</u> <u>NAME</u>	VARC HAR2	30	-	-	2	-	-	-
	<u>DOB</u>	DATE	7	-	-	-	✓	-	-
	<u>ADDR</u> <u>ESS</u>	VARC HAR2	30	-	-	-	✓	-	-
	<u>CITY</u>	CHAR	30	-	-	-	✓	-	-

RESULT:

Thus the SQL queries using DDL Commands with and without constraints were successfully executed and verified.