**Lab Assignment:**

• DDL Commands: Create/Alter/Drop/Grant/Revoke

**1.Create a database using ‘CREATE’ command**

CREATE DATABASE CDAC;

**2.To use the created data base and creation of table in the selected data use the ‘use database name’ command and** create a table using Create command:

**use CDAC;**

create table test(name varchar(20), age int, address varchar(20));

**3.To fetch the table and its structure use the ‘Select’ command**

select \*from table\_name;

**4. To Insert data in the created table use ‘Insert’ Command**

Syntax: INSERT INTO table\_name (column1, column2, ...) VALUES (value1, value2, ...);

• DML Commands: Select/Insert/Update/Delete/Truncate

• DCL Commands: Rollback Commit

• Create new User named ‘dbda’, Grant all the privileges and Perform following Queries.

• Create Table ‘Books’ using proper data types which contain columns (name, author,price, writer)

**4. Use Describe command to check the table schema:**

Describe test;

5. The **SQL DROP DATABASE** statement is used to delete an existing database along with all the data such as tables, views, indexes, stored procedures, and constraints.

**DROP** DATABASE DatabaseName;

You can drop multiple databases using the SQL DROP

**DROP DATABASE testDB3, testDB4;**

1. **verify whether the databases are created or not**

SHOW DATABASES;

**To show the list of tables firstly select the database:**

**use cdac;**

**SHOW TABLES;**

1. **To Rename the table in database:**

**Syntax:** RENAME TABLE **oldTable\_name** to **New\_Tablename;**

**Ex:** RENAME TABLE **test** to **employe;**

1. **To add any new colum in the table using ALTER command**

**Syntax:** ALTER TABLE **table\_name**

ADD **field\_name** datatype**;**

**select \*from employe;**

**Example:** ALTER TABLE **employe**

ADD **Email varchar(255);**

**select \*from employe;**

1. **ALTER TABLE - DROP COLUMN from a table:**

**Syntax:** ALTER TABLE **table\_name**

DROP COLUMN **column\_name;**

**Example: ALTER TABLE employe**

**DROP COLUMN Email;**

1. **ALTER TABLE - RENAME COLUMN in a table:**

**Synatx: ALTER TABLE table\_name**

**RENAME COLUMN old\_name to new\_name;**

**Example:** ALTER TABLE **employe**

RENAME COLUMN **address to mobile;**

1. **ALTER TABLE - ALTER/MODIFY DATATYPE:** To change the data type of a column in a table:

**Syntax: ALTER TABLE table\_name**

**ALTER COLUMN column\_name new\_datatype;**

**Example: ALTER TABLE employe**

**MODIFY COLUMN age varchar(10);**

1. **SQL Constraints**

SQL constraints are used to specify rules for the data in a table.

**The following constraints are commonly used in SQL:**

* NOT NULL - Ensures that a column cannot have a NULL value

**Case 1: At the time of Table creation:**

CREATE TABLE course (ID int NOT NULL, LastName varchar(255) NOT NULL, FirstName varchar(255) NOT NULL, Age int);

**Case 2: if table is created then we can also apply constraints on it**

**Syntax: ALTER TABLE Tabele\_Name**

**MODIFY COLUMN col\_name datatype constraint;**

**Example: ALTER TABLE course**

**MODIFY COLUMN Age int NOT NULL;**

* UNIQUE - Ensures that all values in a column are different

**Case 1: On single column field:**

**CREATE TABLE course (** ID int NOT NULL, LastName varchar(255) NOT NULL,

FirstName varchar(255), Age int, UNIQUE (ID)**);**

**Case 2:** to define a UNIQUE constraint on multiple columns, use the following SQL syntax:

CREATE TABLE Persons (

ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Age int,

CONSTRAINT UNIQUE (ID,LastName)

);

* PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table.

CREATE TABLE course (

ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Age int,

**PRIMARY KEY** (ID)

);

**If table is created then:**

ALTER TABLE **course**  
ADD PRIMARY KEY (ID);

* **FOREIGN KEY** - Prevents actions that would destroy links between tables

CREATE TABLE Orders (

OrderID int NOT NULL,

OrderNumber int NOT NULL,

PersonID int,

PRIMARY KEY (OrderID),

FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)

);

* **CHECK** - Ensures that the values in a column satisfies a specific condition

CREATE TABLE Persons (

ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Age int,

**CHECK** (Age>=18)

);

* **CREATE INDEX** - Used to create and retrieve data from the database very quickly

**Syntax:** CREATE INDEX ind\_name

ON Table\_name (col\_name);

Example: CREATE INDEX ind\_name

ON  **employe** (name);