LAB 1: Data Definition Language (DDL)

Data definition language is the specification notation for defining the database schema.

- Used by the DBA and database designers to specify the conceptual schema of a database.
- In many DBMSs, the DDL is also used to define internal and external schemas (views).

DDL compiler generates a set of tables stored in a data dictionary. SimplyData dictionary contains metadata (i.e., data about data)
The DDL provides the facilities to define:

- Database scheme
- Database tables
- Integrity constraints
  - Domain constraints
  - Referential integrity (references constraint in SQL)
  - Assertions
  - Triggers
  - Views
- Security and Authorization
- Modify the Scheme

The common DDL Commands are: CREATE to create a database or table, ALTER to alter table ,DROP to drop a table and so on.

CREATE
This command builds a new table and has a predefined syntax. The CREATE statement syntax is: **CREATE TABLE [TABLE NAME] ([column definitions] [table parameters]);**
Example: CREATE TABLE Employee (Employee ID int, First-name varchar(20), Last-name varchar(20));

ALTER
An alter command modified an existing database table. This command can add up the additional column, drop existing columns and even change the data type of columns involved in a database table.
The altered command syntax is:
ALTER object type object name parameters;
Example: ALTER TABLE Employee MODIFY First-name varchar(15);

DROP
Drop command is used to delete objects such as a table, index, or view. A DROP statement cannot be rolled back, so once an object is destroyed, there is no way to recover it.
DROP statement syntax is:
DROP object type object name;
Example: DROP TABLE Employee;

TRUNCATE
Similar to DROP, the TRUNCATE statement is used to quickly remove all records from a table. However, unlike DROP which completely deletes table, TRUNCATE preserves its full structure to be used later.

Truncate statement syntax is:
TRUNCATE TABLE table_name;
Example: TRUNCATE TABLE employee;

Problems:

a. **Create a table called EMP with the following structure.**
   Name Type
   ....
   EMPNO NUMBER (6)
   ENAME VARCHAR (20)
   JOB VARCHAR (10)
   DEPT NO NUMBER (3)
   SAL NUMBER (7,2)
   Allow NULL for all columns except ename and job

Queries:

- mysql -u root -p
- CREATE DATABASE LAB1;
- create table EMP (EMPNO INT, ENAME VARCHAR (20) NOT NULL, JOB VARCHAR (10) NOT NULL, DEPTNO INT, SAL INT);
- DESC EMP;

```
mysql> USE LAB1;
Database changed
mysql> create table EMP (EMPNO INT, ENAME VARCHAR(20) NOT NULL, JOB VARCHAR(10)
NOT NULL, DEPTNO INT, SAL INT);
Query OK, 0 rows affected (0.01 sec)

mysql> DESC EMP;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| EMPNO  | int         | YES  |     | NULL    |       |
| ENAME  | varchar(20) | NO   |     | NULL    |       |
| JOB    | varchar(10) | NO   |     | NULL    |       |
| DEPTNO | int         | YES  |     | NULL    |       |
| SAL    | int         | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

- mysql> INSERT INTO EMP VALUES('001', 'ROSHAN', 'DESIGNER', '003', '50000');
- mysql> INSERT INTO EMP VALUES('002', 'UKESH', 'DESIGNER', '002', '50000');
- mysql> INSERT INTO EMP VALUES('003', 'SUMIT', 'DESIGNER', '001', '50000');
- mysql> SELECT * FROM EMP;

```
mysql> INSERT INTO EMP VALUES('001', 'ROSHAN', 'DESIGNER', '003', '50000');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EMP VALUES('002', 'UKESH', 'DESIGNER', '002', '50000');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EMP VALUES('003', 'SUMIT', 'DESIGNER', '001', '50000');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM EMP;
+-------+--------+----------+--------+-------+
| EMPNO | ENAME  | JOB      | DEPTNO | SAL   |
+-------+--------+----------+--------+-------+
|     1 | ROSHAN | DESIGNER |      3 | 50000 |
|     2 | UKESH  | DESIGNER |      2 | 50000 |
|     3 | SUMIT  | DESIGNER |      1 | 50000 |
+-------+--------+----------+--------+-------+
3 rows in set (0.00 sec)
```

b. Add a column experience to the EMP table with data type number and allow null.

Queries:

- ALTER TABLE EMP
  -> ADD COLUMN EXPERIENCE INT AFTER JOB;

```
mysql> ALTER TABLE EMP
    -> ADD COLUMN EXPERIENCE INT AFTER JOB;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC EMP;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| EMPNO      | int         | YES  |     | NULL    |       |
| ENAME      | varchar(20) | NO   |     | NULL    |       |
| JOB        | varchar(10) | NO   |     | NULL    |       |
| EXPERIENCE | int         | YES  |     | NULL    |       |
| DEPTNO     | int         | YES  |     | NULL    |       |
| SAL        | int         | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

c. Modify the width of the job field of EMP table.

Queries:

- ALTER TABLE EMP
  -> MODIFY JOB VARCHAR (20);

```
mysql> ALTER TABLE EMP
    -> MODIFY JOB VARCHAR(20);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC EMP;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| EMPNO      | int         | YES  |     | NULL    |       |
| ENAME      | varchar(20) | NO   |     | NULL    |       |
| JOB        | varchar(20) | YES  |     | NULL    |       |
| EXPERIENCE | int         | YES  |     | NULL    |       |
| DEPTNO     | int         | YES  |     | NULL    |       |
| SAL        | int         | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

d. Create DEPT table with the following structure.

Name Type
....
DEPTNO NUMBER (2)
DNAME VARCHAR (10)
LOC VARCHAR (10)
DEPTNO as the primary key.

Queries:

- create table DEPT (DEPTNO INT, DNAME VARCHAR (10), LOC VARCHAR (10));
- ALTER TABLE DEPT ADD PRIMARY KEY (DEPTNO);

```
mysql> create table DEPT(DEPTNO INT, DNAME VARCHAR(10), LOC VARCHAR (10));
Query OK, 0 rows affected (0.02 sec)

mysql> ALTER TABLE DEPT ADD PRIMARY KEY (DEPTNO);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC DEPT;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| DEPTNO | int         | NO   | PRI | NULL    |       |
| DNAME  | varchar(10) | YES  |     | NULL    |       |
| LOC    | varchar(10) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

e. Create the EMP1 table with ename and empno, add constraints to check the empno value while entering (i.e.) empno>100.

Queries:

- CREATE TABLE EMP1 (EMPNO INT, ENAME VARCHAR (20), CHECK (EMPNO>100));

```
mysql> CREATE TABLE EMP1 (EMPNO INT, ENAME VARCHAR (20), CHECK (EMPNO>100));
Query OK, 0 rows affected (0.01 sec)

mysql> DESC EMP1;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| EMPNO | int         | YES  |     | NULL    |       |
| ENAME | varchar(20) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

- INSERT INTO EMP1 VALUES (‘001’, ‘ROSHAN’);
  ERROR 3819 (HY000): Check constraint 'emp1_chk_1' is violated.
- INSERT INTO EMP1 VALUES ('101', 'ROSHAN');
  Query OK, 1 row affected (0.00 sec)

- SELECT * FROM EMP1;

```
mysql> INSERT INTO EMP1 VALUES('001', 'ROSHAN');
ERROR 3819 (HY000): Check constraint 'emp1_chk_1' is violated.
mysql> INSERT INTO EMP1 VALUES('101', 'ROSHAN');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM EMP1;
+-------+--------+
| EMPNO | ENAME  |
+-------+--------+
|   101 | ROSHAN |
+-------+--------+
1 row in set (0.00 sec)
```

f. Drop a column experience to the emp table.

Queries:

- ALTER TABLE EMP
  -> DROP COLUMN EXPERIENCE;

```
mysql> ALTER TABLE EMP
    -> DROP COLUMN EXPERIENCE;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC EMP;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| EMPNO  | int         | YES  |     | NULL    |       |
| ENAME  | varchar(20) | NO   |     | NULL    |       |
| JOB    | varchar(20) | YES  |     | NULL    |       |
| DEPTNO | int         | YES  |     | NULL    |       |
| SAL    | int         | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

g. Truncate the EMP table and drop the dept table.

Queries:

- TRUNCATE TABLE EMP;

```
mysql> TRUNCATE TABLE EMP;
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT * FROM EMP;
Empty set (0.00 sec)
```

- DROP TABLE DEPT;
- SHOW TABLES;

```
mysql> DROP TABLE DEPT;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW TABLES;
+----------------+
| Tables_in_lab1 |
+----------------+
| EMP            |
| EMP1           |
+----------------+
2 rows in set (0.00 sec)
```