Date:- 12/08/25

Task - 2

Generating design of other Traditional Data Base Model

Aim:- To generate design of other traditional database model and implement DDL Commands of SQL with samples.

DATA DEFINITION LANGUAGE (DDL):-

Definition: DDL Commands are used to define modify or delete the database objects such as tables.

DDL Queries:-

SQL

```
CREATE TABLE Customer(
        Cust_ID INT PRIMARY KEY,
        Cust_Name VARCHAR(100),
        Phone_No VARCHAR(20),
        City VARCHAR(50),
        Amount_Paid DECIMAL(10,2)
);
```

Output: Table created successfully.

2. DESCRIBE (or) DESC:-

Definition:- Displays the structure of a table (column names and data types)

Query:

SQL

```
DESC Customer;
```

Output:-

Cust_ID	int
Cust_Name	Varchar (100)
Phone_No	Varchar (20)
City	Varchar (50)
Amount_Paid	decimal (10,2)

3. DROP TABLE:

Query:

```
DROP TABLE Customer;
```

Output

Table dropped successfully.

4. ALTER TABLE: (Adds fields in a table)

Query

```
ALTER TABLE Customer
ADD COLUMN Email VARCHAR(100);
```

Output:

"Commands completed successfully."

II DML Queries:-

Insert

Query:

Sql

```
INSERT INTO Customer (Cust_ID, Cust_Name, Phone_No,
City, Amount_Paid) VALUES (1, 'John Doe', '123-456-7890',
'New York', 100.00);
```

Output:-

1 row inserted to customer

*SELECT :- (Retrives data from one or more tables)

Query

Sql

```
SELECT * FROM Customer;
```

Output:-

Cust_ID	Cust_Name	Phone_No	City	Amount_Paid
1	John Doe	123-456-7890	New York	100.00
2	Smith	987-654-321	Chicago	200.00
3	Krish	555-123-4567	America	50.00

*UPDATE

Query

Sql

```
UPDATE Customer
SET Amount_Paid = 250.00
WHERE Cust_ID = 1;
```

Output

1 row updated

*DELETE

Query

Sql

```
DELETE FROM Customer
WHERE Cust_ID = 2;
```

Output

1 row deleted;

VEL TECH	
EX NO.	2
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	0
RECORD (5)	
TOTAL (20)	10
IGN WITH DATE	

19/8

Result:- Therefore, DDL and DML commands using My SQL has been implemented successfully.

Aim:- To design and Implement a database for a mobile phone purchase and Billing Manggement System that manages information about customer, Bill, logic, mobile

Steps:- 1. Identify Entities

- Customer
- Bill
- logic
- Mobile

2. Identify Attributes

Customer → Cust_Name, Cust_ID, Cust_phoneNo, Cust_city, Cust_amount paid.

Bill → Price, Bid, Cost Name

logic → Admin ID, password

mobile → Mobile_Name, mobile price, mobile ID

3. Relationships

- Customer-Mobile → (many-to-many) A customer can purchase multiple mobiles.
- Customer-Bill → (one-to-many) A customer can have bills and bills with one customer
- Mobile-login → (one-to-many) A mobile is associated with one login can be maltiple mobiles.

```
CREATE TABLE Customers (
        Cust_ID VARCHAR(255) PRIMARY KEY,
        Cust_Name VARCHAR(255) NOT NULL,
        Cust_Phone_no VARCHAR(20) NOT NULL,
        Cust_city VARCHAR (255) NOT NULL
        Cust_amount_paid DECIMAL (10,2) NOT NULL
    );
```

```
CREATE TABLE Bill (
        Bill_ID VARCHAR(255) PRIMARY KEY,
        Price DECIMAL(10,2) NOT NULL,
        Cust_Name VAACHAR(255) NOT NULL,
        FOREIGN KEY (Cust_Name) REFERENCES
                Customes (Cust_Name)
    );
```

```
CREATE TABLE MOBILE(
        mobile_ID VARCHAR(255) PRIMARY KEY,
        mobile_Name VARCHAR(255) NOT NULL,
        mobile_price DECIMAL(10,2) NOT NULL,
        Phone_ID VARCHAR(255) NOT NULL,
        FOREIGN KEY (Phone_ID) REFERENCES Phone
                (Phone_ID)
    );
```

```
CREATE TABLE Admin (
        Login_ID VARCHAR(255) PRIMARY KEY,
        Admin_ID VARCHAR(255) NOT NULL,
        Password VARCHAR(255) NOT NULL,
    );
```

Constraints

1. Primary keys:
   - Login_ID in Login
   - Cust_ID in Customer
   - Bid in Bill
   - Phone_ID in Mobile
2. Foreign keys:
   - Cust_Name in Bill
     reference Cust_Name in Customer
   - Phone_ID in Mobile is a foreign key

VEL TECH	
EX NO.	26
FERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	2
RECORD (5)	+
TOTAL (20)	12
SIGN WITH DATE	19/8

Result: