12)8/25 Task-2

Generating Design of Other Traditional database mode

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

Data Definition Language(DDL)

Definition: DDL commands are used to define modify or delete the database objects such as table

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

Describe (or) Desc:

Displays the structure of a table (column name and data type)

Query:

```
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 successfuly
```

4) Alter Table: (Adds fields in a table)

Query

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

Output:

```
"commands completed succesfully".
```

DML Queries:

Insert:

Query:

sql

```
INSERT INto customer(cust_ID, Cust_Name,
phone_No, city, Amount_paid) VALUES ('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	JohnDoe	123-456-7890	Newyork	100.00
2	Smith	987-654-321	chicago	200.00
3	Krish	555-123-4567	America	50.00

* UPDATE:

Query

sql

```
UPDATE Customers

SET Amount -Paid = 250.00

WHERE-cust_ID =1;
```

Output

```
1row updated
```

* DELETE

Query

sql

```
DELETE from customers
WHERE cust_ID = 2;
```

output

```
1row deleted;
```

VEL TECH	
EX NO.	2
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	0
RECORD (5)	–
TOTAL (20)	10
SIGN WITH DATE	17/8

Result:

Therefore, DDL and DML commands using My SQL has been implemented successfully

12/8/25 Task-2b

Aim: To design and Implement a database for a database for a mobile phone purchase and Billing Management 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_amountpaid

Bill -> price, Bid, costName

Logic -> Admin ID, password

mobile -> Mobile_Name, mobile price, mobileID

3) Relationships

=> customer_moblie -> (many-to-many) A customer can purchase multiple mobiles

=> customer - Bill -> (one-to-many) A customer can have Bills and bill is with one custom

=> Mobile - login -> (one-to-many) A mobile is associated with one login can be multiple mobiles.

```
CREATE TABLE customer(
    cust_ID VARCHAR(255) PRIMARYKEY,
    cust_Name VARCHAR(255) NOTNULL,
    cust_Phone_no VARCHAR(20) NOTNULL,
    cust_city VARCHR(255) NOT NULL,
    cust_amount_paid DECIMAL (10,2) NOTNULL
);
```

```
CREATETABLE Bill(
  Bill_ID  VARCHAR(255) PRIMARYKEY,
 Price DECIMAL(10,2)  Not NULL,
 cust_Name VARCHAR(255) NOT NULL,
  FOREIGNKEY (cust_Name) REFERENCES
                 customer(cust_Name)

 );
CREATE TABLE MOBILE(
     mobile_ID VARCHAR(255) PRIMARYKEY,
     mobile_Name VARCHAR(255) NOT NULL,
     mobile_Price DECIMAL(10,2) NOTNULL,
     phone_ID VARCHAR(255) NOT NULL,
     FOREIGNKEY (Phone ID) REFERENCES phone
                              (phone_ID)

 );
    CREATETABLE. Admin(
        Login_ID VARCHAR(255) PRIMARY, KEY
        Adim_ID VARCHAR (255)  NOT NULL,
        Passoword 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.	2.1
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	3
RECORD (5)	
TOTAL (20)	13
SIGN WITH DATE	20/8/25

Result:

Thus the Design implement and a Database management system for the mobile phone has been implemented succesfully.