

Assignment -2 Report submitted  
for Database Management System (UEC716)  
by

**Name of student: Prabhnoor singh**

**Roll number: 102115059**

**Group: 3NC3**

Submitted to  
**Ram Kishan Dewangan**



**THAPAR INSTITUTE**  
OF ENGINEERING & TECHNOLOGY  
(Deemed to be University)

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**  
**THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY, (A DEEMED TO**  
**BE UNIVERSITY), PATIALA, PUNJAB**  
**INDIA**  
**Jan-May 2023**

# ASSIGNMENT-2

**QUESTION 1: Create customer table based on the given information.**

**CODE:**

```
CREATE TABLE Customer(  
customer_name VARCHAR(15) NOT NULL UNIQUE PRIMARY KEY,  
customer_street VARCHAR(15),  
customer_city VARCHAR(15) NOT NULL  
);
```

**OUTPUT:**

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains the following SQL code:

```
1  
2 • use Customer;  
3 • CREATE TABLE Customer(  
4   customer_name VARCHAR(15) NOT NULL UNIQUE PRIMARY KEY,  
5   customer_street VARCHAR(15),  
6   customer_city VARCHAR(15) NOT NULL  
7   );  
8 • SELECT * FROM Customer;
```

The result grid shows the output of the query, which is an empty table with the following columns:

	customer_name	customer_street	customer_city
*	NULL	NULL	NULL

The interface also includes a toolbar with various icons for file operations, a "Limit to 1000 rows" dropdown, and a "Result Grid" tab. The bottom of the window shows a tab labeled "Customer 1" and an "Apply" button.

**QUESTION 2: Create branch table based on the given information.****CODE:**

```
CREATE TABLE Branch(  
branch_name VARCHAR(15) NOT NULL UNIQUE PRIMARY KEY,  
branch_city VARCHAR(15) NOT NULL,  
assets INTEGER(8) NOT NULL  
);
```

**OUTPUT:**

The screenshot shows a database management tool interface. At the top, there are tabs labeled q1, q2, q3, q4, q5, and q6. The q2 tab is active. Below the tabs is a toolbar with various icons. The main area displays SQL code:

```
1 • USE Branch;  
2 • CREATE TABLE Branch(  
3   branch_name VARCHAR(15) NOT NULL UNIQUE PRIMARY KEY,  
4   branch_city VARCHAR(15) NOT NULL,  
5   assets INTEGER(8) NOT NULL  
6 );  
7 • SELECT * FROM Branch;
```

Below the code editor is a section labeled "Result Grid". It contains a table with the following structure:

	branch_name	branch_city	assets
*	NULL	NULL	NULL

At the bottom of the interface, there is a tab labeled "Branch 1" with a close button (X).

**QUESTION 3: Create account table based on the given information.****CODE:**

```
CREATE TABLE Account(
account_number INTEGER(8) NOT NULL UNIQUE PRIMARY KEY,
branch_name VARCHAR(15) NOT NULL,
balance INTEGER(8) NOT NULL,
date DATE NOT NULL,
FOREIGN KEY(branch_name) REFERENCES Branch(branch_name)
);
```

**OUTPUT:**

The screenshot shows a SQL IDE with a query editor and a result grid. The query editor contains the following SQL code:

```
1 • USE Account;
2 • USE branch;
3 • CREATE TABLE Account(
4   account_number INTEGER(8) NOT NULL UNIQUE PRIMARY KEY,
5   branch_name VARCHAR(15) NOT NULL,
6   balance INTEGER(8) NOT NULL,
7   date DATE NOT NULL,
8   FOREIGN KEY(branch_name) REFERENCES Branch(branch_name)
9 );
10 • SELECT * FROM Account;
11
```

The result grid shows the output of the SELECT statement, which is an empty table with the following columns: account\_number, branch\_name, balance, and date. All cells in the first row are NULL.

**QUESTION 4: Create loan table based on the given information.****CODE:**

```
CREATE TABLE loan(
loan_number INTEGER(8) NOT NULL UNIQUE,
branch_name VARCHAR(15) NOT NULL,
amount INTEGER(8) NOT NULL,
PRIMARY KEY(loan_number),
FOREIGN KEY(branch_name) REFERENCES Branch(branch_name)
);
```

**OUTPUT:**

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor has tabs for q1, q2, q3, q4 (selected), q5, and q6. The query in q4 is as follows:

```
1 • USE loan;
2 • USE branch;
3 • CREATE TABLE loan(
4     loan_number INTEGER(8) NOT NULL UNIQUE,
5     branch_name VARCHAR(15) NOT NULL,
6     amount INTEGER(8) NOT NULL,
7     PRIMARY KEY(loan_number),
8     FOREIGN KEY(branch_name) REFERENCES Branch(branch_name)
9 );
10 • SELECT * FROM loan;
11
```

The result grid below the query editor shows the following data:

	loan_number	branch_name	amount
*	NULL	NULL	NULL

**QUESTION 5: Create depositor table based on the given information.**

**CODE:**

```
CREATE TABLE Depositor(
customer_name VARCHAR(15) NOT NULL UNIQUE,
account_number INTEGER(8) NOT NULL UNIQUE,
FOREIGN KEY(customer_name) REFERENCES customer(customer_name),
FOREIGN KEY(account_number) REFERENCES Account(account_number)
);
```

**OUTPUT:**

The screenshot shows a database IDE with a SQL editor and a result grid. The SQL editor contains the following code:

```

6  );
7
8  CREATE TABLE Account(
9      account_number INT(8) PRIMARY KEY
10 );
11
12 CREATE TABLE Depositor(
13     customer_name VARCHAR(15) NOT NULL UNIQUE,
14     account_number INTEGER(8) NOT NULL UNIQUE,
15     FOREIGN KEY(customer_name) REFERENCES customer(customer_name),
16     FOREIGN KEY(account_number) REFERENCES Account(account_number)
17 );
18 SELECT * FROM Depositor;
19

```

The result grid shows the output of the query:

	customer_name	account_number
*	NULL	NULL

**QUESTION 6: Create borrower table based on the given information.**

**CODE:**

```

CREATE TABLE Borrower (
    customer_name VARCHAR(15) NOT NULL UNIQUE,
    loan_number INT(8) NOT NULL UNIQUE,
    FOREIGN KEY (customer_name) REFERENCES Customer(customer_name),
    FOREIGN KEY (loan_number) REFERENCES Loan(loan_number) -- Corrected column name
);

```

**OUTPUT:**

q1 q2 q3 q4 q5 q6 x

Limit to 1000 rows

```
1 • USE Borrower;  
2 • USE Customer;  
3 • USE loan;  
4 • CREATE TABLE Borrower (  
5     customer_name VARCHAR(15) NOT NULL UNIQUE,  
6     loan_number INT(8) NOT NULL UNIQUE,  
7     FOREIGN KEY (customer_name) REFERENCES Customer(customer_name),  
8     FOREIGN KEY (loan_number) REFERENCES Loan(loan_number) -- Correct  
9 );  
10 • SELECT * FROM Borrower;
```

Result Grid

	customer_name	loan_number
*	NULL	NULL

Filter Rows: Edit: Export/Import: W