Green University of Bangladesh
Department of Computer Science and Engineering(CSE)
Faculty of Sciences and Engineering
Semester: (Spring, Year:2024), B.Sc. in CSE (Day)

LAB REPORT NO #03
Course Title: Database Lab
Course Code: CSE 210   Section: 221_D9

Experiment Name: Modifying MySQL databases and Updating Data in MySQL Table

Student Details

	Name	ID
1.	Jahidul Islam	221002504

Lab Date	: 09 – 03 – 2024	
Submission Date	: 16 – 03 – 2024	
Course Teacher's Name	: Md. Nazmus Shakib	

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status	
Marks: ………………………………	Signature:.....................
Comments:..	Date:..............................

1. TITLE OF THE LAB EXPERIMENT:

Modifying MySQL databases and Updating Data in MySQL Table

Create This following Bank Database.

branch (branch_name, branch_city, assets)
customer (customer_id, customer_name, customer_city)
account (account_number, branch_name, balance)
loan (loan_number, branch_name, amount)
depositor (customer_name, account_number)
borrower (customer_name, loan_number)

- Tables are placed according to parent and child relationship
- Create above table considering PRIMARY KEY and FOREIGN KEY.
- Data type for amount and balance are INTEGER otherwise VARCHAR(13).
- Insert records into your table.
- Add column Email in customer relation and Set the value.
- Change the name of column name customer_city and modify the data type of column assets

2. OBJECTIVES:

After complementing this lab experiment, we will gain practical knowledge and the outcomes of this experiment are

- To gain the advance knowledge for modifying and updating MySQL databases.

3. PROCEDURE:

To successfully complete the outcomes, we have done the following.
1. Database Design:
   Database Name: Bank, consisting of 6 tables.
2. Table Cration:
   a. Branch has one primary key.

   b. customer has 1 primary key: customer_id

Field	Type	Null	Key	Default	Extra
customer_id	int(11)	NO	PRI	NULL	
customer_name	varchar(100)	YES		NULL	
customer_city	varchar(100)	YES		NULL	

c. account has 1 primary key

Field	Type	Null	Key	Default	Extra
account_number	int(11)	NO	PRI	NULL	
branch_name	varchar(100)	YES	MUL	NULL	
balance	decimal(15,2)	YES		NULL	

d. borrower table has 2 columns.

Field	Type	Null	Key	Default	Extra
customer_name	varchar(100)	YES	UNI	NULL	
loan_number	int(11)	NO	PRI	NULL	

e. loan with 3 columns and 1 primary key

Field	Type	Null	Key	Default	Extra
loan_number	int(11)	NO	PRI	NULL	
branch_name	varchar(100)	YES	MUL	NULL	
amount	decimal(15,2)	YES		NULL	

4. IMPLEMENTATION

Here's I have included all the code we need to obtain all the outcomes of this experiment.

Codes:
CREATE DATABASE bank;
use bank;

Now we need to create 6 tables: branch, customer, account, loan, depositor, borrower.

Table creation:

CREATE TABLE branch(branch_name VARCHAR(100),branch_city VARCHAR(100),assets DECIMAL(15, 2),PRIMARY KEY(branch_name)
);

CREATE TABLE customer(customer_id INT PRIMARY KEY,customer_name VARCHAR(100),customer_city VARCHAR(100)
);
CREATE TABLE ACCOUNT(account_number INT PRIMARY KEY,branch_name VARCHAR(100),balance DECIMAL(15, 2),FOREIGN KEY(branch_name) REFERENCES branch(branch_name)
);
CREATE TABLE loan(loan_number INT PRIMARY KEY,branch_name VARCHAR(100),amount DECIMAL(15, 2),FOREIGN KEY(branch_name) REFERENCES branch(branch_name)
);
CREATE TABLE depositor(customer_name VARCHAR(100),account_number INT,
  FOREIGN KEY(customer_name) REFERENCES customer(customer_name),
  FOREIGN KEY(account_number) REFERENCES ACCOUNT(account_number)
);
CREATE TABLE borrower(customer_name VARCHAR(100),loan_number INT,
  FOREIGN KEY(customer_name) REFERENCES customer(customer_name),
  FOREIGN KEY(loan_number) REFERENCES loan(loan_number)
);

a. **Insertion in the table:**
   i. **Branch Table:**
      INSERT INTO branch (branch_name, branch_city, assets) VALUES
      ('Dhaka Branch', 'Dhaka', 1000000.00),
      ('Chittagong Branch', 'Chittagong', 1500000.50),
      ('Sylhet Branch', 'Sylhet', 2000000.75),
      ('Khulna Branch', 'Khulna', 1200000.25),
      ('Rajshahi Branch', 'Rajshahi', 1800000.80),
      ('Barisal Branch', 'Barisal', 2200000.90);

   ii. **Account Table:**
      INSERT INTO account (account_number, branch_name, balance) VALUES
      (1001, 'Dhaka Branch', 50000.00),
      (1002, 'Chittagong Branch', 75000.50),
      (1003, 'Sylhet Branch', 100000.75),
      (1004, 'Khulna Branch', 60000.25),
      (1005, 'Rajshahi Branch', 90000.80),
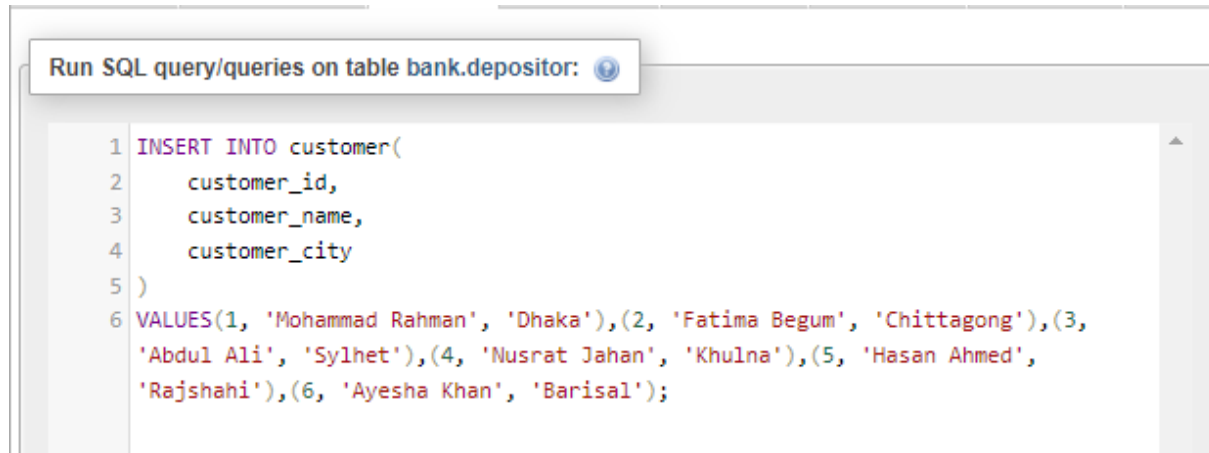   (1006, 'Barisal Branch', 110000.90);
   iii. Customer Table:

INSERT INTO customer (customer_id, customer_name, customer_city) VALUES
(1, 'Mohammad Rahman', 'Dhaka'),
(2, 'Fatima Begum', 'Chittagong'),
(3, 'Abdul Ali', 'Sylhet'),
(4, 'Nusrat Jahan', 'Khulna'),
(5, 'Hasan Ahmed', 'Rajshahi'),
(6, 'Ayesha Khan', 'Barisal');

Run SQL query/queries on table bank.depositor:

```
1  INSERT INTO customer(
2      customer_id,
3      customer_name,
4      customer_city
5  )
6  VALUES(1, 'Mohammad Rahman', 'Dhaka'),(2, 'Fatima Begum', 'Chittagong'),(3,
   'Abdul Ali', 'Sylhet'),(4, 'Nusrat Jahan', 'Khulna'),(5, 'Hasan Ahmed',
   'Rajshahi'),(6, 'Ayesha Khan', 'Barisal');
```

```
1  INSERT INTO branch (branch_name, branch_city, assets) VALUES
2  ('Dhaka Branch', 'Dhaka', 1000000.00),
3  ('Chittagong Branch', 'Chittagong', 1500000.50),
4  ('Sylhet Branch', 'Sylhet', 2000000.75),
5  ('Khulna Branch', 'Khulna', 1200000.25),
6  ('Rajshahi Branch', 'Rajshahi', 1800000.80),
7  ('Barisal Branch', 'Barisal', 2200000.90);
8
```

iv.   borrower table:
INSERT INTO borrower (customer_name, loan_number) VALUES
('Mohammad Rahman', 10001),
('Fatima Begum', 10002),
('Abdul Ali', 10003),
('Nusrat Jahan', 10004),
('Hasan Ahmed', 10005),
('Ayesha Khan', 10006);

v.   loan table:

```
 1  INSERT INTO loan(
 2      loan_number,
 3      branch_name,
 4      amount
 5  )
 6  VALUES(10001, 'Dhaka Branch', 5000.00),(
 7      10002,
 8      'Chittagong Branch',
 9      7500.50
10  ),(10003, 'Sylhet Branch', 10000.75),(10004, 'Khulna Branch', 6000.25),
    (10005, 'Rajshahi Branch', 9000.80),(10006, 'Barisal Branch', 11000.90);
```

b. Modifying exiting table.
   i.   Add column Email in customer table.

   ii. Change customer_city to address

Field	Type	Null	Key	Default	Extra
customer_id	int(11)	NO	PRI	NULL	
customer_name	varchar(100)	YES		NULL	
address	varchar(100)	YES		NULL	
Email	varchar(255)	YES		NULL	

   iii. Modified assets column to DECIMAL with two-point float position.

```
 1  ALTER TABLE
 2      branch MODIFY COLUMN assets DECIMAL(18, 2);
```

5. OUTPUT:

We can see the data of our earlier created database uing the following command.

Showing rows 0 - 5 (6 total, Query took 0.0001 seconds.)

SELECT * FROM `account`

Profiling [Edit inline

Show all | Number of rows: 25 ∨   Filter rows: Search this table   S

Extra options

←T→				account_number	branch_name	balance
☐	✎ Edit	⊞ Copy	⊝ Delete	1001	Dhaka Branch	50000.00
☐	✎ Edit	⊞ Copy	⊝ Delete	1002	Chittagong Branch	75000.50
☐	✎ Edit	⊞ Copy	⊝ Delete	1003	Sylhet Branch	100000.75
☐	✎ Edit	⊞ Copy	⊝ Delete	1004	Khulna Branch	60000.25
☐	✎ Edit	⊞ Copy	⊝ Delete	1005	Rajshahi Branch	90000.80
☐	✎ Edit	⊞ Copy	⊝ Delete	1006	Barisal Branch	110000.90

↑ ☐ Check all   With selected: ✎ Edit   ⊞ Copy   ⊝ Delete   ⊞ Export

---

Showing rows 0 - 5 (6 total, Query took 0.0002 seconds.)

SELECT * FROM `branch`

Profiling [Edit inline] [

Show all | Number of rows: 25 ∨   Filter rows: Search this table   Sort

Extra options

←T→				branch_name	branch_city	assets
☐	✎ Edit	⊞ Copy	⊝ Delete	Barisal Branch	Barisal	2200000.90
☐	✎ Edit	⊞ Copy	⊝ Delete	Chittagong Branch	Chittagong	1500000.50
☐	✎ Edit	⊞ Copy	⊝ Delete	Dhaka Branch	Dhaka	1000000.00
☐	✎ Edit	⊞ Copy	⊝ Delete	Khulna Branch	Khulna	1200000.25
☐	✎ Edit	⊞ Copy	⊝ Delete	Rajshahi Branch	Rajshahi	1800000.80
☐	✎ Edit	⊞ Copy	⊝ Delete	Sylhet Branch	Sylhet	2000000.75

↑ ☐ Check all   With selected: ✎ Edit   ⊞ Copy   ⊝ Delete   ⊞ Export

6. ANALYSIS AND DISCUSSION:

7. SUMMARY:

The lab experiment is successfully completed on creating and inserting data in the database with the given Lab Task.

The data insertion has reference integrity by referencing existing data from another tables. The lab aimed to demonstrate the implementation of integrity constraints and foreign key relationships in MySQL databases while updating existing tables and data in it.

Data types are chosen differently, with DECIMAL though the given instruction was to choose INTEGER type for numerical values like amount and balance, and VARCHAR for textual data.