

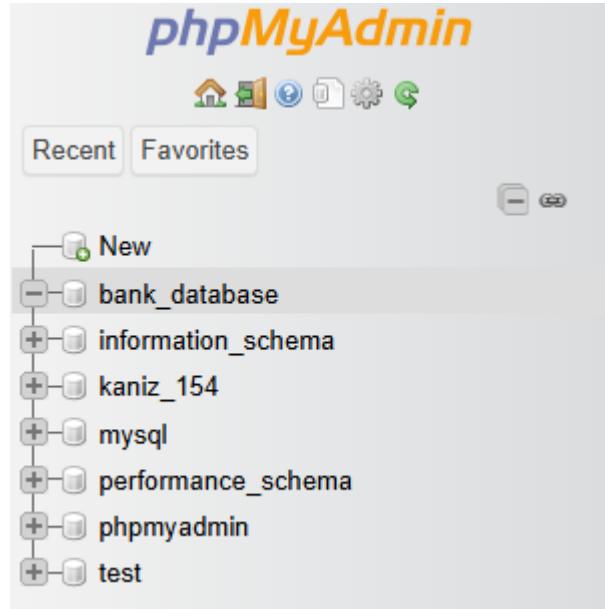
Lab 02: Bank Database

Name Kaniz Fatema

ID: 20245103154

1. Creating a new database:

```
CREATE DATABASE bank_database;
```



2. Create 6 tables in this DB named: account, branch, depositor, customer, loan, and borrower

Create Account:

```
CREATE TABLE account (account_number char(5), branch_name varchar(16), balance int(7));
CREATE TABLE branch (branch_name varchar(16), branch_city varchar(16), assets int(8));
CREATE TABLE depositor (customer_name varchar(16), account_number char(5));
CREATE TABLE customer (customer_name varchar(16), customer_street varchar(32), customer_city varchar(16));
CREATE TABLE loan (loan_number char(5), branch_name varchar(16), amount int(8));
CREATE TABLE borrower (customer_name varchar(16), loan_number char(8));
```

Table structure for table `account`:

account_number	branch_name	balance
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

3. Inserting data into the tables

Account:

```
INSERT INTO account VALUES("A-101", "Downtown", 500);
INSERT INTO account VALUES("A-102", "Perryridge", 400);
INSERT INTO account VALUES("A-201", "Brighton", 900);
INSERT INTO account VALUES("A-215", "Mianus", 700);
INSERT INTO account VALUES("A-217", "Brighton", 750);
INSERT INTO account VALUES("A-222", "Redwood", 700);
INSERT INTO account VALUES("A-305", "Round Hill", 350);
```

Table structure for table `account`:

account_number	branch_name	balance
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

Branch:

```
INSERT INTO branch VALUES("Brighton", "Brooklyn", 7100000);
INSERT INTO branch VALUES("Downtown", "Brooklyn", 9000000);
INSERT INTO branch VALUES("Mianus", "Horseneck", 4000000);
```

```
INSERT INTO branch VALUES("North Town", "Rye", 3700000);
INSERT INTO branch VALUES("Perryridge", "Horseneck", 1700000);
INSERT INTO branch VALUES("Pownal", "Bennington", 300000);
INSERT INTO branch VALUES("Redwood", "Palo Alto", 2100000);
INSERT INTO branch VALUES("Round Hill", "Horseneck", 8000000);
```

The screenshot shows the phpMyAdmin interface for the 'bank_database'. The left sidebar shows the database structure with tables: New, bank_database, account, borrower, branch, customer, depositor, loan, information_schema, kaniz_154, mysql, performance_schema, phpmyadmin, and test. The 'branch' table is selected. The main area displays the following data:

branch_name	branch_city	assets
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

Depositor:

```
INSERT INTO depositor VALUES("Hayes", "A-102");
INSERT INTO depositor VALUES("Johnson", "A-101");
INSERT INTO depositor VALUES("Johnson", "A-201");
INSERT INTO depositor VALUES("Jones", "A-217");
INSERT INTO depositor VALUES("Lindsay", "A-222");
INSERT INTO depositor VALUES("Smith", "A-215");
INSERT INTO depositor VALUES("Turner", "A-305");
```

The screenshot shows the phpMyAdmin interface for the 'bank_database'. The left sidebar shows the database structure with tables: New, bank_database, account, borrower, branch, customer, depositor, loan, information_schema, kaniz_154, mysql, performance_schema, phpmyadmin, and test. The 'depositor' table is selected. The main area displays the following data:

customer_name	account_number
Hayes	A-102
Johnson	A-101
Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305

Customer:

```
INSERT INTO customer VALUES("Adams", "Spring", "Pittsfield");
INSERT INTO customer VALUES("Brooks", "Senator", "Brooklyn");
INSERT INTO customer VALUES("Curry", "North", "Rye");
INSERT INTO customer VALUES("Glenn", "Sand Hill", "Woodside");
INSERT INTO customer VALUES("Green", "Walnut", "Stamford");
INSERT INTO customer VALUES("Hayes", "Main", "Harrison");
INSERT INTO customer VALUES("Johnson", "Alma", "Palo Alto");
INSERT INTO customer VALUES("Jones", "Main", "Harrison");
INSERT INTO customer VALUES("Lindsay", "Park", "Pittsfield");
INSERT INTO customer VALUES("Smith", "North", "Rye");
INSERT INTO customer VALUES("Turner", "Putnam", "Stamford");
INSERT INTO customer VALUES("Williams", "Nassau", "Princeton");
```

The screenshot shows the phpMyAdmin interface for a MySQL database named 'bank_database'. The left sidebar lists various schemas and tables. The 'customer' table is selected in the main area. The table structure is shown with columns: customer_name, customer_street, and customer_city. The data grid displays 12 rows of customer information. Below the table, there are 'Query results operations' buttons for Print, Copy to clipboard, Export, Display chart, and Create view. A 'Bookmark this SQL query' button is also present.

customer_name	customer_street	customer_city
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

Loan:

```
INSERT INTO loan VALUES("L-11", "Round Hill", 900);
INSERT INTO loan VALUES("L-14", "Downtown", 1500);
INSERT INTO loan VALUES("L-15", "Perryridge", 1500);
INSERT INTO loan VALUES("L-16", "Perryridge", 1300);
INSERT INTO loan VALUES("L-17", "Downtown", 1000);
INSERT INTO loan VALUES("L-23", "Redwoord", 2000);
INSERT INTO loan VALUES("L-93", "Mianus", 500);
```

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

loan_number	branch_name	amount
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Manus	500

Borrower:

```
INSERT INTO borrower VALUES("Adams", "L-16");
INSERT INTO borrower VALUES("Curry", "L-93");
INSERT INTO borrower VALUES("Hayes", "L-15");
INSERT INTO borrower VALUES("Johnson", "L-14");
INSERT INTO borrower VALUES("Jones", "L-17");
INSERT INTO borrower VALUES("Smith", "L-11");
INSERT INTO borrower VALUES("Smith", "L-23");
INSERT INTO borrower VALUES("Williams", "L-17");
```

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

customer_name	loan_number
Adams	L-16
Curry	L-93
Hayes	L-15
Johnson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

Primary Key after table creation

Account:

```
ALTER TABLE account ADD PRIMARY KEY(account_number);
```

The screenshot shows the phpMyAdmin interface for the 'account' table in the 'bank_database'. The table has columns: account_number, branch_name, and balance. The data is as follows:

account_number	branch_name	balance
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

Branch:

```
ALTER TABLE branch ADD PRIMARY KEY(branch_name);
```

The screenshot shows the phpMyAdmin interface for the 'branch' table in the 'bank_database'. The table has columns: branch_name, branch_city, and assets. The data is as follows:

branch_name	branch_city	assets
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

Depositor:

```
ALTER TABLE depositor ADD PRIMARY KEY(customer_name, account_number);
```

Showing rows 0 - 6 (7 total). Query took 0.0001 seconds.

	customer_name	account_number
Edit	Hayes	A-102
Edit	Johnson	A-101
Edit	Johnson	A-201
Edit	Jones	A-217
Edit	Lindsay	A-222
Edit	Smith	A-215
Edit	Turner	A-305

Customer:

```
ALTER TABLE customer ADD PRIMARY KEY(customer_name);
```

Showing rows 0 - 6 (7 total). Query took 0.0001 seconds.

	customer_name	account_number
Edit	Hayes	A-102
Edit	Johnson	A-101
Edit	Johnson	A-201
Edit	Jones	A-217
Edit	Lindsay	A-222
Edit	Smith	A-215
Edit	Turner	A-305

Loan:

```
ALTER TABLE loan ADD PRIMARY KEY(loan_number);
```

The screenshot shows the phpMyAdmin interface for the 'loan' table in the 'bank_database'. The table has columns: loan_number, branch_name, and amount. The data is as follows:

	loan_number	branch_name	amount
<input type="checkbox"/>	L-11	Round Hill	900
<input type="checkbox"/>	L-14	Downtown	1500
<input type="checkbox"/>	L-15	Perryridge	1500
<input type="checkbox"/>	L-16	Perryridge	1300
<input type="checkbox"/>	L-17	Downtown	1000
<input type="checkbox"/>	L-23	Redwood	2000
<input type="checkbox"/>	L-93	Manus	500

Borrower:

```
ALTER TABLE borrower ADD PRIMARY KEY(customer_name, loan_number);
```

The screenshot shows the phpMyAdmin interface for the 'borrower' table in the 'bank_database'. The table has columns: customer_name and loan_number. The data is as follows:

	customer_name	loan_number
<input type="checkbox"/>	Adams	L-16
<input type="checkbox"/>	Curry	L-93
<input type="checkbox"/>	Hayes	L-15
<input type="checkbox"/>	Johnson	L-14
<input type="checkbox"/>	Jones	L-17
<input type="checkbox"/>	Smith	L-11
<input type="checkbox"/>	Smith	L-23
<input type="checkbox"/>	Williams	L-17

Lab Tasks

1. Find the names of all branches in the **loan** relation

```
SELECT branch_name FROM loan;
```

The screenshot shows the phpMyAdmin interface for the 'loan' table. The table structure is displayed with the following data:

branch_name
Round Hill
Downtown
Perryridge
Redwood
Mianus

1. Find the names of all branches in the **loan** relation

```
SELECT branch_name FROM loan;
```

The screenshot shows the phpMyAdmin interface after executing the query. The results are displayed in the 'Query results operations' section:

branch_name
Round Hill
Downtown
Perryridge
Redwood
Mianus

2. Find all loan numbers for loans made at the **Perryridge** branch with loan amounts greater than 300

```
SELECT loan_number FROM loan WHERE branch_name="Perryridge" AND amount > 300;
```

The screenshot shows the phpMyAdmin interface for a database named 'bank_database'. The left sidebar lists tables: New, bank_database, account, borrower, branch, customer, depositor, and loan. The 'loan' table is selected. The main area displays a SQL query: `SELECT loan_number FROM loan WHERE branch_name='Perryridge' AND amount > 300;`. Below the query, the results show 2 rows. The results table has columns: loan_number, branch_name, amount, and date. The first row is L-15 (branch_name: Perryridge, amount: 350, date: 2023-01-01) and the second row is L-16 (branch_name: Perryridge, amount: 300, date: 2023-01-01). There are buttons for Edit, Copy, Delete, and Export.

3. Find all the loan numbers of the customers who has loan either Perryridge branch or Downtown branch

```
SELECT loan_number FROM loan WHERE branch_name IN("Perryridge", "Downtown");
```

The screenshot shows the phpMyAdmin interface for a database named 'bank_database'. The left sidebar lists tables: New, bank_database, account, borrower, branch, customer, depositor, and loan. The 'loan' table is selected. The main area displays a SQL query: `SELECT loan_number FROM loan WHERE branch_name IN("Perryridge", "Downtown");`. Below the query, the results show 3 rows. The results table has columns: loan_number, branch_name, amount, and date. The first row is L-14 (branch_name: Perryridge, amount: 300, date: 2023-01-01), the second row is L-15 (branch_name: Perryridge, amount: 350, date: 2023-01-01), and the third row is L-17 (branch_name: Downtown, amount: 300, date: 2023-01-01). There are buttons for Edit, Copy, Delete, and Export.

4. Find all the loan numbers of the customers who has loan either Perryridge branch or Downtown branch or Mianus branch

```
SELECT loan_number FROM loan WHERE branch_name IN("Perryridge", "Downtown", "Mianus");
```

The screenshot shows the phpMyAdmin interface for the 'loan' table. The left sidebar lists databases and tables, with 'loan' selected. The main area displays the first four rows of the 'loan' table. The rows are highlighted with a light blue background. The columns shown are 'loan_number' and 'branch_name'. The SQL query at the top is:

```
SELECT loan_number FROM loan WHERE branch_name IN("Perryridge", "Downtown", "Mianus");
```

5. Find the names of all customers who are not from **Stamford** or **Princeton** or **Harrison** city

```
SELECT customer_name FROM customer WHERE customer_city NOT IN("Stamford", "Princeton", "Harrison");
```

The screenshot shows the phpMyAdmin interface for the 'customer' table. The left sidebar lists databases and tables, with 'customer' selected. The main area displays the first seven rows of the 'customer' table. The rows are highlighted with a light blue background. The columns shown are 'customer_name' and 'customer_city'. The SQL query at the top is:

```
SELECT customer_name FROM customer WHERE customer_city NOT IN("Stamford", "Princeton", "Harrison");
```

6. Find the largest, minimum, and average account balance in the **account** relation

```
SELECT Max(balance), Min(balance), Avg(balance) FROM account;
```

Showing rows 0 - 0 (1 total). Query took 0.0001 seconds.

Max(balance)	Min(balance)	Avg(balance)
900	350	614.2857

7. Find the total number of customer from **customer** relation

```
SELECT Count(customer_name) FROM customer;
```

Showing rows 0 - 0 (1 total). Query took 0.0001 seconds.

Count(customer_name)
12

8. Find the loan number of those loans with loan amounts between 400 and 800

```
SELECT loan_number FROM loan WHERE amount > 400 AND amount < 800;
```

or

```
SELECT loan_number FROM loan WHERE amount BETWEEN 401 AND 799;
```

Showing rows 0 - 0 (1 total, Query took 0.0001 seconds.)

```
SELECT loan_number FROM loan WHERE amount BETWEEN 400 AND 800;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

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

Extra options

customer	loan_number
kaniz_154	L-93

Edit Copy Delete L-93

With selected: Edit Copy Delete Export

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

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Label: Let every user access this bookmark

Bookmark this SQL query

9. Find the names of all customers whose name start with G

```
SELECT customer_name FROM customer WHERE customer_name LIKE "G%";
```

Showing rows 0 - 1 (2 total, Query took 0.0002 seconds.)

```
SELECT customer_name FROM customer WHERE customer_name LIKE "G%";
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

customer_name
Glenn
Green

Edit Copy Delete Glenn

Edit Copy Delete Green

With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Label: Let every user access this bookmark

Bookmark this SQL query

10. Find the names of all customers whose name ends with s

```
SELECT customer_name FROM customer WHERE customer_name LIKE "%s";
```

The screenshot shows the phpMyAdmin interface for a database named 'bank_database'. The left sidebar lists various tables: New, bank_database, account, borrower, branch, customer, depositor, loan, information_schema, kaniz_154, mysql, performance_schema, phpmyadmin, and test. The 'customer' table is selected. The main area displays the results of the SQL query: `SELECT customer_name FROM customer WHERE customer_name LIKE "%k%"`. The results show five rows: Adams, Brooks, Hayes, Jones, and Williams. A red box highlights the results table.

customer_name
Adams
Brooks
Hayes
Jones
Williams

11. Find the names of all customers whose name has a o in 2nd position

```
SELECT customer_name FROM customer WHERE customer_name LIKE "_o%";
```

The screenshot shows the phpMyAdmin interface for a database named 'bank_database'. The left sidebar lists various tables. The 'customer' table is selected. The main area displays the results of the SQL query: `SELECT customer_name FROM customer WHERE customer_name LIKE "_o%"`. The results show two rows: Johnson and Jones. A red box highlights the results table.

customer_name
Johnson
Jones

12. Find the names of all customers whose name has a o in any position except 1st and last letter

```
SELECT customer_name FROM customer WHERE customer_name LIKE "%o%";
```

The screenshot shows the phpMyAdmin interface for a database named 'bank_database'. The left sidebar lists various tables: New, bank_database, account, borrower, branch, customer, depositor, loan, information_schema, kaniz_154, mysql, performance_schema, phpmyadmin, and test. The 'customer' table is selected. A query has been run: `SELECT customer_name FROM customer WHERE customer_name LIKE "%Kos%"`. The results show three rows: Brooks, Johnson, and Jones. The entire results table is highlighted with a red box.

customer_name
Brooks
Johnson
Jones

13. Find the length of the name of all customers from Customer relation

```
SELECT customer_name, Length(customer_name) FROM customer;
```

The screenshot shows the phpMyAdmin interface for a database named 'bank_database'. The left sidebar lists various tables. The 'customer' table is selected. A query has been run: `SELECT customer_name, Length(customer_name) FROM customer`. The results show 12 rows of customer names and their lengths. The entire results table is highlighted with a red box.

customer_name	Length(customer_name)
Adams	5
Brooks	6
Curry	5
Glen	5
Green	5
Hayes	5
Johnson	7
Jones	5
Lindsay	7
Smith	5
Turner	6
Williams	8

14. Find 1st three characters of each customer name from customer relation

```
SELECT customer_name, SUBSTR(customer_name, 1, 3) FROM customer;
```

phpMyAdmin

Server: 127.0.0.1 > Database: bank_database > Table: customer

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Show query box

Showing rows 0 - 11 (12 total, Query took 0.0001 seconds.)

SELECT customer_name, SUBSTR(customer_name, 1, 3) FROM customer;

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	customer_name	SUBSTR(customer_name, 1, 3)
<input type="checkbox"/>	Edit Copy Delete Adams	Ada
<input type="checkbox"/>	Edit Copy Delete Brooks	Bro
<input type="checkbox"/>	Edit Copy Delete Curry	Cur
<input type="checkbox"/>	Edit Copy Delete Glenn	Gle
<input type="checkbox"/>	Edit Copy Delete Green	Gre
<input type="checkbox"/>	Edit Copy Delete Hayes	Hay
<input type="checkbox"/>	Edit Copy Delete Johnson	Joh
<input type="checkbox"/>	Edit Copy Delete Jones	Jon
<input type="checkbox"/>	Edit Copy Delete Lindsay	Lin
<input type="checkbox"/>	Edit Copy Delete Smith	Smi
<input type="checkbox"/>	Edit Copy Delete Turner	Tur
<input type="checkbox"/>	Edit Copy Delete Williams	Wil

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations