



## Project Final Report

Only for course Teacher						
		Needs Improvement	Developing	Sufficient	Above Average	Total Mark
<b>Allocate mark &amp; Percentage</b>		<b>25%</b>	<b>50%</b>	<b>75%</b>	<b>100%</b>	<b>25</b>
Understanding	<b>3</b>					
Analysis	<b>4</b>					
Implementation	<b>8</b>					
Report Writing	<b>10</b>					
<b>Total obtained mark</b>						

**Semester: Spring ...../ Fall - 2024**

**Course Code: SE 224                          Course Name: Database System Lab**

**Course Teacher Name: Tapushe Rabaya Toma**

**Designation: Assistant Professor**

Student Name	Student ID	Section
Sohan Ahemmed	0242310005341266	D-1
Niher Ronjon Pramanik	0242310005341102	D-1

**Submission Date: ...06.... /...12.../...2024....**

# BANKING SYSTEM

## Scenario:

Bank is structured into branches, each having a unique identifier, a name, and an address. Each branch serves a diverse group of customers across different locations and offers a variety of banking services.

Each customer of Bank has a unique identifier along with personal information including a first and last name, date of birth, gender, and contact number. A customer can open one or more bank accounts at their chosen branch and can perform various transactions and access banking services through these accounts.

A bank account has a unique identifier and a current balance. Each account is linked to one active card, which has its own unique number, expiration date, and a status indicating if it is blocked. An account can also be used to initiate loans and make transactions.

Each card is associated with one bank account and allows customers to access funds or make purchases. The card includes important details such as a unique card number, expiration date, and a status to indicate whether it is blocked due to security or other reasons.

The loan system tracks each loan issued by the bank. Each loan has a unique identifier, the loan type (e.g., personal loan, student loan, etc.), the total loan amount, the amount repaid, and start and due dates. Additionally, each loan type includes a base amount, interest rate, and description to help guide customers on loan options and repayment terms.

To manage its financial activities, the bank keeps detailed transaction records. Each transaction has a unique identifier, a description, amount, and date. Transactions may include deposits, withdrawals, and payments made by the customer, which are recorded for future reference and tracking.

In addition to core banking operations, Bank provides various services to its customers, such as credit cards, loans, and online banking. Each service has a unique identifier, a name, a brief description, and a service fee if applicable.

## Entity Design and Structure:

"BRANCHES" Table

Field Name	Type	Constraints
branch_id	Int	Primary Key
branch_name	Varchar2(50)	Not Null
address	Varchar2(100)	Not Null
contact_number	Varchar2(15)	Not Null
manager_id	Int	Foreign Key

## "CUSTOMERS" Table

Field Name	Type	Constraints
customer_id	Int	Primary Key
first_name	Varchar2(20)	Not Null
last_name	Varchar2(20)	Not Null
date_of_birth	Date	Not Null
gender	Varchar2(10)	Not Null
contact_number	Varchar2(15)	Not Null
email	Varchar2(50)	
address	Varchar2(100)	Not Null
branch_id	Int	Foreign Key

## "ACCOUNTS" Table

Field Name	Type	Constraints
account_id	Int	Primary Key
customer_id	Int	Foreign Key
branch_id	Int	Foreign Key
account_type	Varchar2(20)	Not Null
balance	Decimal(15, 2)	Not Null
created_date	Date	Not Null
status	Varchar2(10)	Not Null

## "CARDS" Table

Field Name	Type	Constraints
card_id	Int	Primary Key
account_id	Int	Foreign Key
card_number	Varchar2(20)	Unique
expiration_date	Date	Not Null
is_blocked	Boolean	Not Null
card_type	Varchar2(10)	Not Null

## "LOANS" Table

Field Name	Type	Constraints
loan_id	Int	Primary Key
account_id	Int	Foreign Key
loan_type	Varchar2(20)	Not Null
amount	Decimal(15, 2)	Not Null
amount_paid	Decimal(15, 2)	
interest_rate	Decimal(4, 2)	

start_date	Date	Not Null
due_date	Date	Not Null

### "TRANSACTIONS" Table

Field Name	Type	Constraints
transaction_id	Int	Primary Key
account_id	Int	Foreign Key
transaction_type	Varchar2(20)	Not Null
amount	Decimal (15, 2)	Not Null
transaction_date	Date	Not Null

### "EMPLOYEES" Table

Field Name	Type	Constraints
employee_id	Int	Primary Key
branch_id	Int	Foreign Key
name	Varchar2(50)	Not Null
position	Varchar2(20)	Not Null
salary	Decimal(10, 2)	Not Null
contact_number	Varchar2(15)	

### "ATM\_LOCATIONS" Table

Field Name	Type	Constraints
atm_id	Int	Primary Key
location	Varchar2(100)	Not Null
branch_id	Int	Foreign Key
status	Varchar2(10)	Not Null

### "ATM\_TRANSACTIONS" Table

Field Name	Type	Constraints
atm_transaction_id	Int	Primary Key
atm_id	Int	Foreign Key
card_id	Int	Foreign Key
transaction_type	Varchar2(20)	Not Null
amount	Decimal(15, 2)	
transaction_date	Date	Not Null

## "SERVICES" Table

Field Name	Type	Constraints
service_id	Int	Primary Key
service_name	Varchar2(50)	Not Null
availability_status	Varchar2(10)	Not Null

**1. Branches:** This entity holds information about each branch of the bank.

- branch\_id: A unique identifier for each branch.
- branch\_name: The name of the branch.
- address: The physical address of the branch.
- contact\_number: The primary contact number for the branch.
- manager\_id: The ID of the employee who manages the branch.

**2. Customers:** This entity stores combined customer details including personal and contact information.

- customer\_id: A unique identifier for each customer.
- first\_name: The first name of the customer.
- last\_name: The last name of the customer.
- date\_of\_birth: The birthdate of the customer.
- gender: The gender of the customer.
- contact\_number: The contact number of the customer.
- email: The email address of the customer.
- address: The residential address of the customer.
- branch\_id: The branch where the customer is registered.

**3. Accounts:** This entity tracks customer accounts, including account type, balance, and status.

- account\_id: A unique identifier for each account.
- customer\_id: The ID of the customer who owns the account.
- branch\_id: The branch where the account is held.
- account\_type: The type of account (e.g., Savings, Checking).
- balance: The current balance in the account.
- created\_date: The date when the account was opened.
- status: The current status of the account (e.g., Active, Closed).

**4. Cards:** This entity tracks bank cards linked to customer accounts.

- card\_id: A unique identifier for each card.
- account\_id: The account linked to the card.
- card\_number: The unique card number.

- expiration\_date: The expiration date of the card.
- is\_blocked: A boolean indicating if the card is blocked.
- card\_type: The type of card (e.g., Debit, Credit).

**5. Loans:** This entity contains details about loans issued to customers.

- loan\_id: A unique identifier for each loan.
- account\_id: The account linked to the loan.
- loan\_type: The type of loan (e.g., Personal, Student).
- amount: The total loan amount.
- amount\_paid: The total amount repaid on the loan.
- interest\_rate: The interest rate on the loan.
- start\_date: The date the loan was issued.
- due\_date: The loan's due date for repayment.

**6. Transactions:** This entity logs transactions made on customer accounts, such as deposits, withdrawals, and payments.

- transaction\_id: A unique identifier for each transaction.
- account\_id: The account involved in the transaction.
- transaction\_type: The type of transaction (e.g., Deposit, Withdrawal).
- amount: The amount of money involved in the transaction.
- transaction\_date: The date when the transaction took place.

**7. Employees:** This entity contains information about employees working at each branch of the bank.

- employee\_id: A unique identifier for each employee.
- branch\_id: The branch where the employee works.
- name: The name of the employee.
- position: The job title or role of the employee (e.g., Teller, Manager).
- salary: The salary of the employee.
- contact\_number: The contact information of the employee.

**8. ATM Locations:** This entity holds information about the bank's ATMs and their locations.

- atm\_id: A unique identifier for each ATM.
- location: The physical location of the ATM.
- branch\_id: The branch responsible for the ATM.
- status: The operational status of the ATM (e.g., Active, Out of Service).

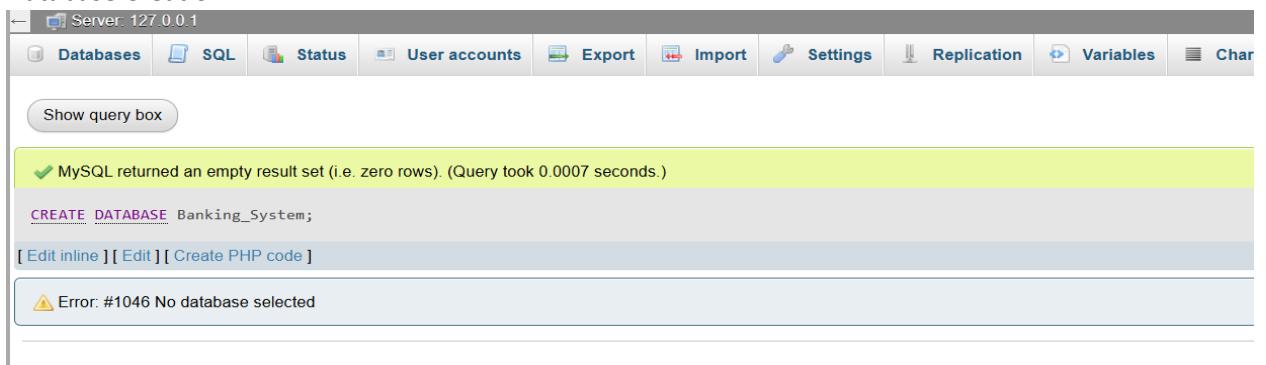
**9. ATM Transactions:** This entity records transactions conducted at ATMs, including withdrawals and balance inquiries.

- atm\_transaction\_id: A unique identifier for each ATM transaction.
- atm\_id: The ATM involved in the transaction.
- card\_id: The card used for the transaction.
- transaction\_type: The type of transaction (e.g., Withdrawal, Balance Inquiry).
- amount: The amount involved in the ATM transaction.
- transaction\_date: The date and time of the transaction.

#### 10. Services: This entity tracks various banking services offered to customers.

- service\_id: A unique identifier for each service.
- service\_name: The name of the service (e.g., Loan, Credit Card).
- availability\_status: The current status of the service (e.g., Active, Inactive).

### 1. Database Creation



The screenshot shows the MySQL Workbench interface with the following details:

- Server:** 127.0.0.1
- Tab Bar:** Databases, SQL, Status, User accounts, Export, Import, Settings, Replication, Variables, Char
- Query Box:** Show query box
- Status Bar:** MySQL returned an empty result set (i.e. zero rows). (Query took 0.0007 seconds.)
- Query:** CREATE DATABASE Banking\_System;
- Buttons:** [Edit inline] [Edit] [Create PHP code]
- Message:** Error: #1046 No database selected

### 2. Create Branches Table



The screenshot shows the MySQL Workbench interface with the following details:

- Left Panel:** Recent, Favorites, New, banking\_system, branches, cse, d1, information\_schema
- Server:** 127.0.0.1
- Tab Bar:** Recent, Favorites
- Query Box:** Show query box
- Status Bar:** MySQL returned an empty result set (i.e. zero rows). (Query took 0.0055 seconds.)
- Query:** CREATE TABLE Branches ( branch\_id INT PRIMARY KEY, branch\_name VARCHAR(100), address VARCHAR(100), contact\_number VARCHAR(15), manager\_id INT );
- Buttons:** [Edit inline] [Edit] [Create PHP code]

### 3. Create Customers Table



The screenshot shows the MySQL Workbench interface with the following details:

- Left Panel:** Recent, Favorites, New, banking\_system, customers, cse, d1, information\_schema
- Server:** 127.0.0.1
- Tab Bar:** Recent, Favorites
- Query Box:** Show query box
- Status Bar:** MySQL returned an empty result set (i.e. zero rows). (Query took 0.0202 seconds.)
- Query:** CREATE TABLE Customers( customer\_id INT PRIMARY KEY, first\_name VARCHAR(50), last\_name VARCHAR(50), date\_of\_birth DATE, gender VARCHAR(10), contact\_number VARCHAR(15), email VARCHAR(100), address VARCHAR(255), branch\_id INT, FOREIGN KEY (branch\_id) REFERENCES Branches(branch\_id) );
- Buttons:** [Edit inline] [Edit] [Create PHP code]

#### 4. Create Accounts Table

The screenshot shows the MySQL Workbench interface. On the left, there's a tree view of the schema named 'banking\_system' containing tables like 'accounts', 'branches', and 'customers'. The main panel displays a query results window with a green success message: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0212 seconds.)'. Below it is the SQL query:

```
CREATE TABLE Accounts( account_id int PRIMARY KEY, customer_id int, branch_id int, account_type varchar(20), balance double(15, 2), created_date DATE, status varchar(20), FOREIGN KEY (customer_id) REFERENCES Customers(customer_id), FOREIGN KEY (branch_id) REFERENCES Branches(branch_id) );
```

Buttons at the bottom include 'Edit inline', 'Edit', and 'Create PHP code'.

#### 5. Create Cards Table

The screenshot shows the MySQL Workbench interface. The schema 'banking\_system' is selected. The main panel displays a query results window with a green success message: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0200 seconds.)'. Below it is the SQL query:

```
CREATE TABLE Cards( card_id int PRIMARY KEY, account_id int, card_number varchar(16), expiration_date DATE, is_blocked boolean, card_type varchar(10), FOREIGN KEY (account_id) REFERENCES Accounts(account_id) );
```

Buttons at the bottom include 'Edit inline', 'Edit', and 'Create PHP code'.

#### 6. Create Loans Table

The screenshot shows the MySQL Workbench interface. The schema 'banking\_system' is selected. The main panel displays a query results window with a green success message: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0189 seconds.)'. Below it is the SQL query:

```
CREATE TABLE Loans( loan_id int PRIMARY KEY, account_id int, loan_type varchar(20), amount double(15, 2), amount_paid double(15, 2), interest_rate double(5, 2), start_date DATE, due_date DATE, FOREIGN KEY (account_id) REFERENCES Accounts(account_id) );
```

Buttons at the bottom include 'Edit inline', 'Edit', and 'Create PHP code'.

#### 7. Create Transactions Table

The screenshot shows the MySQL Workbench interface. The schema 'banking\_system' is selected. The main panel displays a query results window with a green success message: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0146 seconds.)'. Below it is the SQL query:

```
CREATE TABLE Transactions( transaction_id int PRIMARY KEY, account_id int, transaction_type varchar(20), amount double(15, 2), transaction_date DATE, FOREIGN KEY (account_id) REFERENCES Accounts(account_id) );
```

Buttons at the bottom include 'Edit inline', 'Edit', and 'Create PHP code'.

#### 8. Create Employees Table

The screenshot shows the MySQL Workbench interface. The schema 'banking\_system' is selected. The main panel displays a query results window with a green success message: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0190 seconds.)'. Below it is the SQL query:

```
CREATE TABLE Employees( employee_id int PRIMARY KEY, branch_id int, name varchar(100), position varchar(50), salary decimal(15, 2), contact_number varchar(15), FOREIGN KEY (branch_id) REFERENCES Branches(branch_id) );
```

Buttons at the bottom include 'Edit inline', 'Edit', and 'Create PHP code'.

#### 9. Create ATM\_Locations Table

The screenshot shows the MySQL Workbench interface. The schema 'banking\_system' is selected. The main panel displays a query results window with a green success message: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0196 seconds.)'. Below it is the SQL query:

```
CREATE TABLE ATM_Locations( atm_id int PRIMARY KEY, location varchar(100), branch_id int, status varchar(20), FOREIGN KEY (branch_id) REFERENCES Branches(branch_id) );
```

Buttons at the bottom include 'Edit inline', 'Edit', and 'Create PHP code'.

## 10. Create ATM\_Transactions Table

The screenshot shows the MySQL Workbench interface. On the left, there is a tree view of the schema named 'banking\_system'. Under 'banking\_system', there are several tables: New, accounts, atm\_locations, atm\_transactions, branches, cards, customers, employees, loans, and transactions. The 'atm\_transactions' table is selected. The main pane displays the SQL code for creating the table:

```
CREATE TABLE ATM_Transactions( atm_transaction_id int PRIMARY KEY, atm_id int, card_id int, transaction_type varchar(20), amount double(15, 2), transaction_date datetime, FOREIGN KEY (atm_id) REFERENCES ATM_Locations(atm_id), FOREIGN KEY (card_id) REFERENCES Cards(card_id) );
```

Below the code, there are three buttons: [Edit inline], [Edit], and [Create PHP code]. A status message at the top right says: "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0206 seconds.)".

## 11. Create Services Table

The screenshot shows the MySQL Workbench interface. The schema tree on the left is identical to the previous one, with 'atm\_transactions' selected. The main pane displays the SQL code for creating the 'Services' table:

```
CREATE TABLE Services( service_id int PRIMARY KEY, service_name varchar(50), availability_status varchar(20) );
```

Below the code, there are three buttons: [Edit inline], [Edit], and [Create PHP code]. A status message at the top right says: "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0110 seconds.)".

# Data Insertion:

## 1. Branches Table

The screenshot shows the MySQL Workbench interface for the 'branches' table. The main pane displays the results of a SELECT query:

```
Showing rows 0 - 9 (10 total, Query took 0.0002 seconds.)
```

Below the results, there is a SQL editor with the query:

```
SELECT * FROM `branches`
```

At the bottom of the editor, there are buttons for Profiling, Edit inline, Edit, Explain SQL, Create PHP code, and Refresh.

Below the editor, there are filtering options: Show all (unchecked), Number of rows: 25, Filter rows: Search this table, and Sort by key: None.

At the bottom left, there is an 'Extra options' button.

The main data area is a table with the following columns: branch\_id, branch\_name, address, contact\_number, and manager\_id. The data is as follows:

	branch_id	branch_name	address	contact_number	manager_id
<input type="checkbox"/>	1	Dhaka Main Branch	Mirpur-1, Dhaka	01512YYYYYY	101
<input type="checkbox"/>	2	Faridpur Branch	New Market, Faridpur	01513YYYYYY	102
<input type="checkbox"/>	3	Rajshahi Branch	Bashundhara, Rajshahi	01514YYYYYY	103
<input type="checkbox"/>	4	Sylhet Branch	Madar Bazar, Sylhet	01515YYYYYY	104
<input type="checkbox"/>	5	Uttara Branch	Uttara, Dhaka	01515YYYYYY	105
<input type="checkbox"/>	6	Barisal Branch	Cantonment, Barisal	01516YYYYYY	106
<input type="checkbox"/>	7	Chittagong Branch	Station Road, Chittagong	01518YYYYYY	107
<input type="checkbox"/>	8	Gulshan Branch	House #35, Road #3, Gulshan	01519YYYYYY	108
<input type="checkbox"/>	9	Dhanmondi Branch	Shukrabad, Dhanmondi	01521YYYYYY	109
<input type="checkbox"/>	10	Rajbari Branch	Rajbari Sadar, Rajbari	01522YYYYYY	110

At the bottom, there are buttons for Check all, With selected:, Edit, Copy, Delete, and Export.

## 2. Customers Table

Showing rows 0 - 9 (10 total, Query took 0.0002 seconds.)

```
SELECT * FROM `customers`
```

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_id	first_name	last_name	date_of_birth	gender	contact_number	email	address	branch_id
<input type="checkbox"/>	1	Ayesha	Sultana	1990-06-15	Female	018XXXXXXX	ayesha@gmail.com	Mirpur-10, Dhaka	1
<input type="checkbox"/>	2	Mohammad	Hossain	1985-04-22	Male	019XXXXXXX	mohammad@gmail.com	Agrabad, Chittagong	7
<input type="checkbox"/>	3	Shakil	Rahman	1992-08-10	Male	017XXXXXXX	shakil@gmail.com	Kazla, Rajshahi	3
<input type="checkbox"/>	4	Tania	Khan	1987-11-30	Female	016XXXXXXX	tania@gmail.com	VIP Road, Sylhet	4
<input type="checkbox"/>	5	Imran	Hossain	1995-03-05	Male	015XXXXXXX	imran@gmail.com	Uttara, Dhaka	5
<input type="checkbox"/>	6	Nabila	Chowdhury	1993-06-12	Female	014XXXXXXX	nabila@gmail.com	Barisal City, Barisal	6
<input type="checkbox"/>	7	Sohan	Ahmed	1994-07-25	Male	013XXXXXXX	sohan@gmail.com	Faridpur Sadar, Faridpur	2
<input type="checkbox"/>	8	Saima	Akter	1988-09-15	Female	012XXXXXXX	saima@gmail.com	Uttara, Dhaka	8
<input type="checkbox"/>	9	Sabbir	Rahman	1991-05-10	Male	011XXXXXXX	sabbir@gmail.com	Gulshan-1, Gulshan	9
<input type="checkbox"/>	10	Niher	Ronjon	1992-12-20	Male	010XXXXXXX	niher@gmail.com	Rajbari Sadar, Rajbari	10

Check all    With selected:  Edit  Copy  Delete  Export

## 3. Accounts Table

Showing rows 0 - 9 (10 total, Query took 0.0002 seconds.)

```
SELECT * FROM `accounts`
```

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

	account_id	customer_id	branch_id	account_type	balance	created_date	status
<input type="checkbox"/>	101	2	1	Current Account	50000.00	2023-01-15	Active
<input type="checkbox"/>	102	7	2	Student Account	15000.00	2022-08-10	Active
<input type="checkbox"/>	103	4	3	Business Account	250000.00	2024-03-20	Active
<input type="checkbox"/>	104	9	4	Current Account	100000.00	2023-02-25	Inactive
<input type="checkbox"/>	105	5	5	Fixed Deposit Accoun	45000.00	2021-12-10	Active
<input type="checkbox"/>	106	10	6	Savings Account	70000.00	2023-06-15	Active
<input type="checkbox"/>	107	6	7	Business Account	500000.00	2022-11-05	Active
<input type="checkbox"/>	108	1	8	Student Account	12000.00	2024-01-10	Active
<input type="checkbox"/>	109	3	9	Fixed Deposit Accoun	35000.00	2020-09-15	Closed
<input type="checkbox"/>	110	8	10	Savings Account	60000.00	2024-03-01	Active

Check all    With selected:  Edit  Copy  Delete  Export

#### 4. Cards Table

Showing rows 0 - 9 (10 total, Query took 0.0002 seconds.)

`SELECT * FROM `cards``

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

		card_id	account_id	card_number	expiration_date	is_blocked	card_type
<input type="checkbox"/>	Edit  Copy  Delete	11	101	4567123456789012	2027-12-31	0	Credit
<input type="checkbox"/>	Edit  Copy  Delete	22	102	9876123456789012	2026-11-30	0	Debit
<input type="checkbox"/>	Edit  Copy  Delete	33	103	1234567890123456	2025-10-31	1	Credit
<input type="checkbox"/>	Edit  Copy  Delete	44	104	2345678901234567	2025-01-01	0	Credit
<input type="checkbox"/>	Edit  Copy  Delete	55	105	3456789012345678	2026-11-30	0	Debit
<input type="checkbox"/>	Edit  Copy  Delete	66	106	4567890123456789	2027-05-15	0	Debit
<input type="checkbox"/>	Edit  Copy  Delete	77	107	5678901234567890	2026-09-22	1	Debit
<input type="checkbox"/>	Edit  Copy  Delete	88	108	6789012345678901	2028-02-20	0	Credit
<input type="checkbox"/>	Edit  Copy  Delete	99	109	7890123456789012	2025-03-30	0	Debit
<input type="checkbox"/>	Edit  Copy  Delete	111	110	8901234567890123	2027-06-11	1	Credit

Check all With selected: Edit Copy Delete Export

#### 5. Loans Table

Showing rows 0 - 9 (10 total, Query took 0.0002 seconds.)

`SELECT * FROM `loans``

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

		loan_id	account_id	loan_type	amount	amount_paid	interest_rate	start_date	due_date
<input type="checkbox"/>	Edit  Copy  Delete	111	101	Personal Loan	100000.00	25000.00	10.50	2023-02-10	2025-02-10
<input type="checkbox"/>	Edit  Copy  Delete	120	110	Business Loan	1500000.00	500000.00	9.75	2023-12-20	2033-12-20
<input type="checkbox"/>	Edit  Copy  Delete	222	102	Student Loan	50000.00	100000.00	8.75	2022-05-12	2032-05-12
<input type="checkbox"/>	Edit  Copy  Delete	333	103	Business Loan	200000.00	50000.00	7.25	2023-06-15	2026-06-15
<input type="checkbox"/>	Edit  Copy  Delete	444	104	Personal Loan	150000.00	30000.00	12.00	2024-01-01	2026-01-01
<input type="checkbox"/>	Edit  Copy  Delete	555	105	Business Loan	800000.00	200000.00	9.50	2021-07-20	2031-07-20
<input type="checkbox"/>	Edit  Copy  Delete	666	106	Home Loan	100000.00	150000.00	8.00	2022-03-25	2032-03-25
<input type="checkbox"/>	Edit  Copy  Delete	777	107	Personal Loan	350000.00	50000.00	11.00	2023-09-05	2025-09-05
<input type="checkbox"/>	Edit  Copy  Delete	888	108	Student Loan	60000.00	100000.00	8.75	2024-02-18	2034-02-18
<input type="checkbox"/>	Edit  Copy  Delete	999	109	Home Loan	400000.00	120000.00	10.25	2020-10-10	2025-10-10

Check all With selected: Edit Copy Delete Export

## 6. Transactions

The screenshot shows the phpMyAdmin interface for the 'accounts' table in the 'banking\_system' database. The table has columns: account\_id, customer\_id, branch\_id, account\_type, balance, created\_date, and status. There are 10 rows of data.

account_id	customer_id	branch_id	account_type	balance	created_date	status
101	2	1	Current Account	50000.00	2023-01-15	Active
102	7	2	Student Account	15000.00	2022-08-10	Active
103	4	3	Business Account	250000.00	2024-03-20	Active
104	9	4	Current Account	100000.00	2023-02-25	Inactive
105	5	5	Fixed Deposit Account	45000.00	2021-12-10	Active
106	10	6	Savings Account	70000.00	2023-06-15	Active
107	6	7	Business Account	500000.00	2022-11-05	Active
108	1	8	Student Account	12000.00	2024-01-10	Active
109	3	9	Fixed Deposit Account	35000.00	2020-09-15	Closed
110	8	10	Savings Account	60000.00	2024-03-01	Active

## 7. Employees

The screenshot shows the phpMyAdmin interface for the 'employees' table in the 'banking\_system' database. The table has columns: employee\_id, branch\_id, name, position, salary, and contact\_number. There are 12 rows of data.

employee_id	branch_id	name	position	salary	contact_number
111	1	Md. Rahim	Branch Manager	50000.00	01511122334
112	2	Nasrin Akter	Assistant Manager	40000.00	01712233445
113	3	Shamim Hossain	Cashier	30000.00	01813344556
114	4	Farzana Sultana	Customer Service Officer	35000.00	01914455678
115	1	Abdul Karim	Security Officer	25000.00	01615566778
116	2	Sultana Begum	Accountant	37000.00	01516677889
117	3	Hasan Ali	Loan Officer	42000.00	01717788990
118	4	Mita Chowdhury	IT Officer	38000.00	01618899001
119	1	Tarvir Ahmed	Clerk	22000.00	01919900112
120	2	Natisha Islam	Marketing Officer	36000.00	01620011223

## 8. ATM\_Locations

The screenshot shows the phpMyAdmin interface for the 'atm\_locations' table in the 'banking\_system' database. The table has columns: atm\_id, location, branch\_id, and status. There are 10 rows of data, all marked as Active.

atm_id	location	branch_id	status
221	Mirpur-1, Dhaka	1	Active
222	New Market, Fardpur	2	Active
223	Bashundhara, Rajshahi	3	Active
224	Madar Bazar, Sylhet	4	Active
225	Uttara, Dhaka	5	Active
226	Cantonment, Barisal	6	Active
227	Station Road, Chittagong	7	Active
228	House #35, Road #3, Gulshan	8	Active
229	Shukrabad, Dhanmondi	9	Inactive
230	Rajbari Sadar, Rajbari	10	Inactive

## 9. ATM\_Transactions

The screenshot shows the phpMyAdmin interface for the 'atm\_transactions' table in the 'banking\_system' database. The table has columns: atm\_transaction\_id, atm\_id, card\_id, transaction\_type, amount, and transaction\_date. There are 10 rows of data.

atm_transaction_id	atm_id	card_id	transaction_type	amount	transaction_date
331	221	11	Withdrawal	5000.00	2024-12-01 00:00:00
332	222	22	Deposit	15000.00	2024-12-02 00:00:00
333	223	33	Withdrawal	7000.00	2024-12-03 00:00:00
334	224	44	Balance Inquiry	0.00	2024-12-04 00:00:00
335	225	55	Withdrawal	3000.00	2024-12-05 00:00:00
336	226	66	Deposit	20000.00	2024-12-06 00:00:00
337	227	77	Withdrawal	4000.00	2024-12-07 00:00:00
338	228	88	Deposit	10000.00	2024-12-08 00:00:00
339	229	99	Withdrawal	6000.00	2024-12-09 00:00:00
340	230	111	Balance Inquiry	0.00	2024-12-10 00:00:00

## 10. Services

The screenshot shows the phpMyAdmin interface for the 'banking\_system' database. The left sidebar lists various databases and tables, including 'assignment\_library', 'banking\_system' (which is expanded to show 'accounts', 'atm\_locations', 'atm\_transactions', 'branches', 'cards', 'customers', 'employees', 'loans', 'services', and 'transactions'), and others like 'companydb', 'cse', 'd1', etc. The main panel displays the 'services' table with the following data:

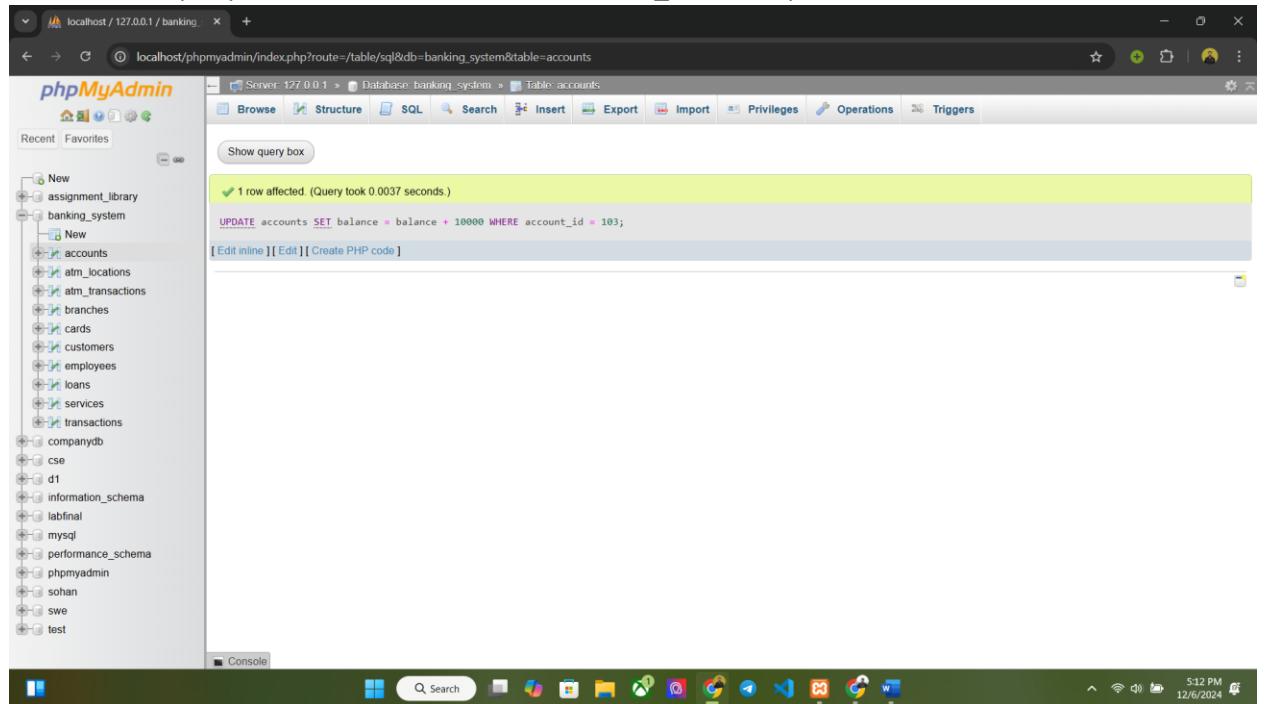
	service_id	service_name	availability_status
<input type="checkbox"/>	551	Account Opening	Available
<input type="checkbox"/>	552	Loan Application	Available
<input type="checkbox"/>	553	Credit Card Services	Unavailable
<input type="checkbox"/>	554	ATM Cash Withdrawal	Available
<input type="checkbox"/>	555	Online Banking	Available
<input type="checkbox"/>	556	Mobile Banking	Available
<input type="checkbox"/>	557	Fixed Deposit	Unavailable
<input type="checkbox"/>	558	Locker Facility	Available
<input type="checkbox"/>	559	Cheque Book Request	Available
<input type="checkbox"/>	560	Foreign Exchange	Unavailable

Below the table, there are buttons for 'Check all', 'With selected', 'Edit', 'Copy', 'Delete', and 'Export'. At the bottom of the page, there are links for 'Open results operations' and 'Console'.

# Question List Using Accounts Table:

## QS Using Update Query:

1. Write an SQL query to increase the balance of account\_id = 103 by 10,000.

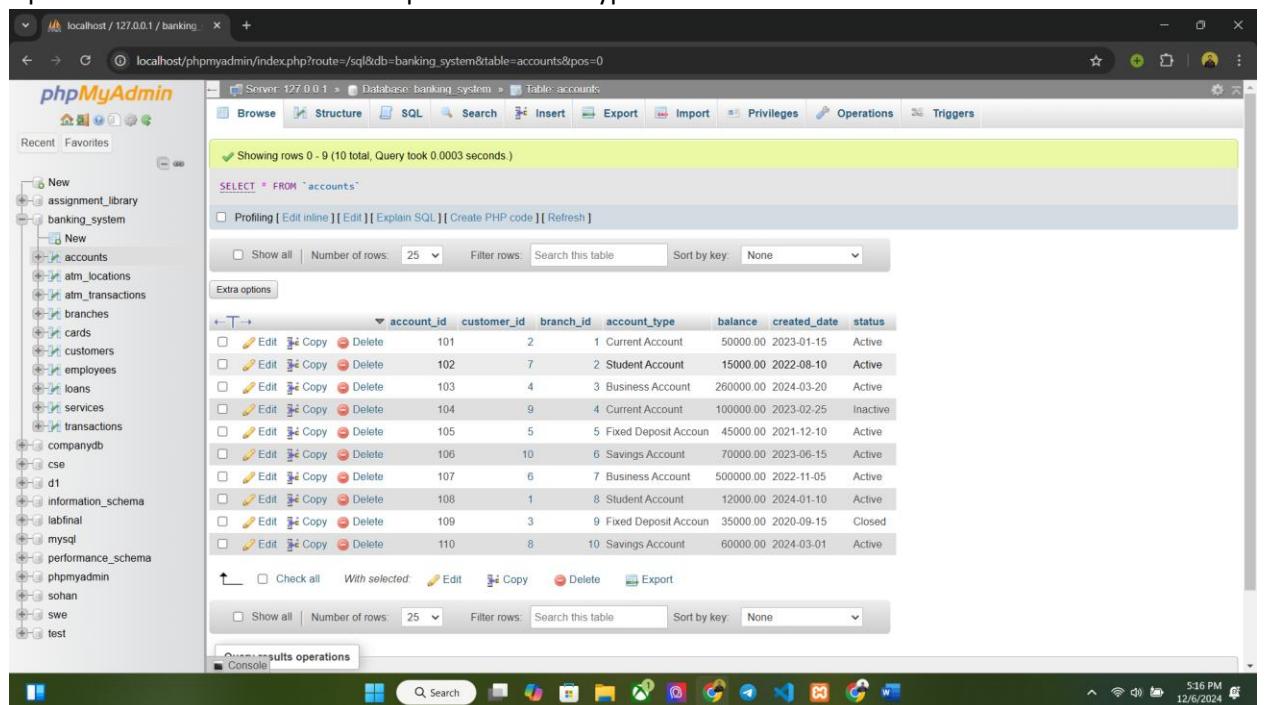


The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The 'accounts' table is selected. In the SQL tab, the following query is run:

```
UPDATE accounts SET balance = balance + 10000 WHERE account_id = 103;
```

The result message indicates 1 row affected. The status bar at the bottom right shows the date and time as 5:16 PM 12/6/2024.

2. Update the status of all "Fixed Deposit Account" types to "Matured".



The screenshot shows the phpMyAdmin interface for the same database and table. In the SQL tab, the following query is run:

```
SELECT * FROM `accounts`
```

The results show a list of accounts with columns: account\_id, customer\_id, branch\_id, account\_type, balance, created\_date, and status. The status column for the last two rows ('Fixed Deposit Account' type) is updated to 'Matured'.

account_id	customer_id	branch_id	account_type	balance	created_date	status
101	2	1	Current Account	50000.00	2023-01-15	Active
102	7	2	Student Account	15000.00	2022-08-10	Active
103	4	3	Business Account	260000.00	2024-03-20	Active
104	9	4	Current Account	100000.00	2023-02-25	Inactive
105	5	5	Fixed Deposit Accoun	45000.00	2021-12-10	Active
106	10	6	Savings Account	70000.00	2023-06-15	Active
107	6	7	Business Account	500000.00	2022-11-05	Active
108	1	8	Student Account	12000.00	2024-01-10	Active
109	3	9	Fixed Deposit Account	35000.00	2020-09-15	Closed
110	8	10	Savings Account	60000.00	2024-03-01	Active

## QS Using Alter Query:

1. Add a new column interest\_rate to store the interest rate for accounts.

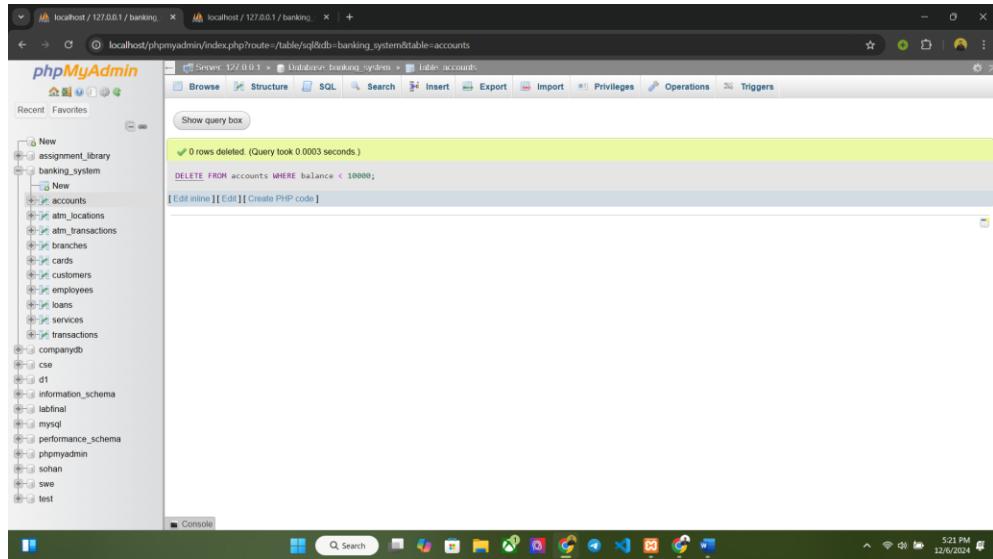
The screenshot shows the phpMyAdmin interface for a MySQL database named 'banking\_system'. The left sidebar lists various databases and tables, including 'accounts'. The main panel shows the 'Structure' tab for the 'accounts' table. A SQL query has been run: `ALTER TABLE accounts ADD COLUMN interest_rate DECIMAL(5, 2);`. The results show a green success message: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0096 seconds.)'. Below the message, there are 'Edit inline', 'Edit', and 'Create PHP code' buttons.

2. Modify the status column to allow a maximum of 20 characters instead of its current size.

The screenshot shows the phpMyAdmin interface for the same MySQL database. The 'Structure' tab for the 'accounts' table is displayed. A SQL query has been run: `ALTER TABLE accounts MODIFY COLUMN status VARCHAR(20);`. The results show a green success message: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0026 seconds.)'. Below the message, there are 'Edit inline', 'Edit', and 'Create PHP code' buttons.

## QS Using Delete Query:

1. Delete accounts with balance less than 10,000



The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The left sidebar lists various tables: assignment\_library, banking\_system, accounts, atm\_locations, atm\_transactions, branches, cards, customers, employees, loans, services, transactions, companydb, cse, d1, information\_schema, labfinal, mysql, performance\_schema, phpmyadmin, sohan, swe, and test. The 'accounts' table is selected. In the main query editor, the following SQL code is entered:

```
DELETE FROM accounts WHERE balance < 10000;
```

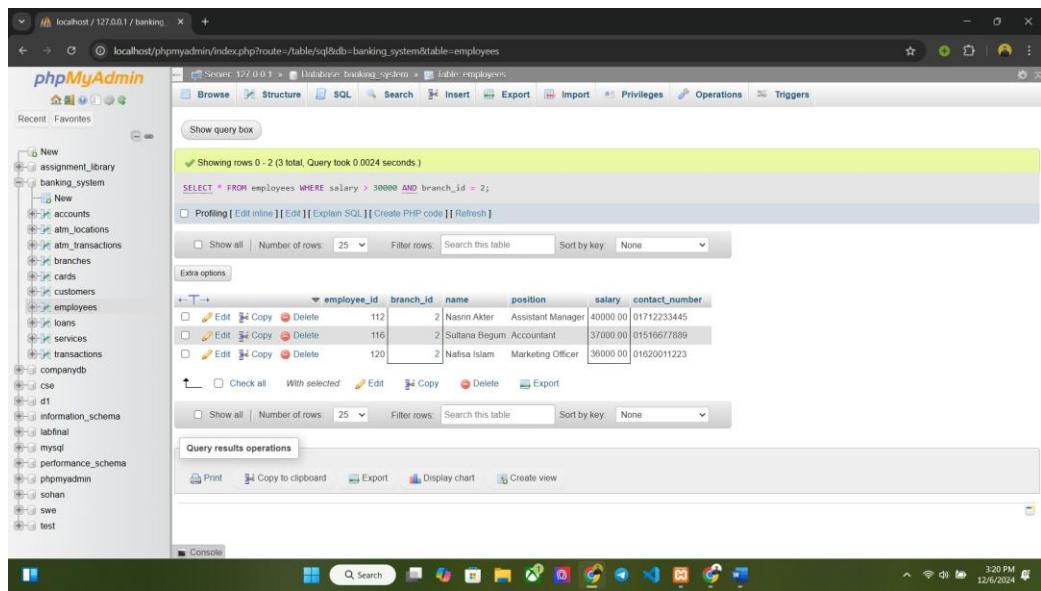
The status bar at the bottom right indicates the query took 0.0003 seconds and deleted 0 rows.

## Question List Using Employees Table:

### Basic SELECT Operations Questions

#### Using AND

1. Write an SQL query to select employees with a salary greater than 30,000 AND who work in branch\_id = 2.



The screenshot shows the phpMyAdmin interface for the same 'banking\_system' database. The 'employees' table is selected. The query editor contains the following SQL code:

```
SELECT * FROM employees WHERE salary > 30000 AND branch_id = 2;
```

The results table displays three rows of employee data:

employee_id	branch_id	name	position	salary	contact_number
112	2	Nasrin Akter	Assistant Manager	40000.00	01712233445
116	2	Sultana Begum	Accountant	37000.00	01516677899
120	2	Nafisa Islam	Marketing Officer	36000.00	01620011223

2. Write an SQL query to find employees with the position "Cashier" AND a salary less than 40,000.

The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The left sidebar lists various tables and databases. The main area displays the results of a query:

```
SELECT * FROM employees WHERE position = 'Cashier' AND salary < 40000;
```

The results table shows one row:

employee_id	branch_id	name	position	salary	contact_number
113	3	Shamim Hossain	Cashier	30000.00	01813344556

## Using OR

1. Write an SQL query to find employees who are either "Branch Manager" OR "Accountant".

The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The left sidebar lists various tables and databases. The main area displays the results of a query:

```
SELECT * FROM employees WHERE position = 'Branch Manager' OR position = 'Accountant';
```

The results table shows two rows:

employee_id	branch_id	name	position	salary	contact_number
111	1	Md. Rahim	Branch Manager	50000.00	01511122334
116	2	Sultana Begum	Accountant	37000.00	01516677889

2. Write an SQL query to fetch employees who have a salary greater than 40,000 OR belong to branch\_id = 3.

The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The left sidebar lists various tables: assignment\_library, banking\_system (with sub-tables accounts, atm\_locations, atm\_transactions, branches, cards, customers, employees, loans, services, transactions), companydb, cse, d1, information\_schema, labfinal, mysql, performance\_schema, phpmyadmin, sohan, swe, and test. The 'employees' table is selected in the main area. A query has been run in the SQL tab:

```
SELECT * FROM employees WHERE salary > 40000 OR branch_id = 3;
```

The results show three employees:

employee_id	branch_id	name	position	salary	contact_number
111	1	Md. Rahim	Branch Manager	50000.00	01511122334
113	3	Shamim Hossain	Cashier	30000.00	01813344556
117	3	Hasan Ali	Loan Officer	42000.00	01717788990

## Using NOT

1. Write an SQL query to select employees who are NOT working in branch\_id = 1.

The screenshot shows the phpMyAdmin interface for the same database. The 'employees' table is selected. A query has been run in the SQL tab:

```
SELECT * FROM employees WHERE NOT branch_id = 1;
```

The results show all employees except those from branch\_id 1:

employee_id	branch_id	name	position	salary	contact_number
112	2	Nasrin Akter	Assistant Manager	40000.00	01712233445
113	3	Shamim Hossain	Cashier	30000.00	01813344556
114	4	Farzana Sultana	Customer Service Officer	35000.00	01914455667
116	2	Sultana Begum	Accountant	37000.00	01516677889
117	3	Hasan Ali	Loan Officer	42000.00	01717788990
118	4	Mita Chowdhury	IT Officer	38000.00	01818899001
120	2	Nafisa Islam	Marketing Officer	36000.00	01620011223

2. Write an SQL query to find employees who are NOT earning a salary greater than 35,000.

The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The 'employees' table is selected. A query is run: `SELECT * FROM employees WHERE NOT salary > 35000;`. The results show four rows of employees whose salaries are 30000, 35000, 25000, and 22000 respectively. The 'contact\_number' column contains unique identifiers for each employee.

employee_id	branch_id	name	position	salary	contact_number
113	3	Shamim Hossain	Cashier	30000.00	01813344556
114	4	Farzana Sultana	Customer Service Officer	35000.00	01914455667
115	1	Abdul Karim	Security Officer	25000.00	01615566778
119	1	Tanvir Ahmed	Clerk	22000.00	01919900112

## Using BETWEEN

1. Write an SQL query to select employees whose salary is BETWEEN 25,000 and 40,000.

The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The 'employees' table is selected. A query is run: `SELECT * FROM employees WHERE salary BETWEEN 25000 AND 40000;`. The results show seven rows of employees whose salaries fall within the specified range. The 'contact\_number' column contains unique identifiers for each employee.

employee_id	branch_id	name	position	salary	contact_number
112	2	Nasrin Akter	Assistant Manager	40000.00	01712233445
113	3	Shamim Hossain	Cashier	30000.00	01813344556
114	4	Farzana Sultana	Customer Service Officer	35000.00	01914455667
115	1	Abdul Karim	Security Officer	25000.00	01615566778
116	2	Sultana Begum	Accountant	37000.00	01516677889
118	4	Mita Chowdhury	IT Officer	38000.00	01818899001
120	2	Nafisa Islam	Marketing Officer	36000.00	01620011223

2. Write an SQL query to find employees with employee\_id BETWEEN 115 and 119.

The screenshot shows the phpMyAdmin interface for the 'employees' table in the 'banking\_system' database. The query executed was:

```
SELECT * FROM employees WHERE employee_id BETWEEN 115 AND 119;
```

The results are:

employee_id	branch_id	name	position	salary	contact_number
115	1	Abdul Karim	Security Officer	25000.00	01615566778
116	2	Sultana Begum	Accountant	37000.00	01516677889
117	3	Hasan Ali	Loan Officer	42000.00	01717788990
118	4	Mita Chowdhury	IT Officer	38000.00	01818899001
119	1	Tanvir Ahmed	Clerk	22000.00	01919900112

## Using ORDER BY

1. Write an SQL query to display all employees ordered by salary in ascending order.

The screenshot shows the phpMyAdmin interface for the 'employees' table in the 'banking\_system' database. The query executed was:

```
SELECT * FROM employees ORDER BY salary ASC;
```

The results are:

employee_id	branch_id	name	position	salary	contact_number
119	1	Tanvir Ahmed	Clerk	22000.00	01919900112
115	1	Abdul Karim	Security Officer	25000.00	01615566778
113	3	Shamim Hossain	Cashier	30000.00	01813344556
114	4	Farzana Sultana	Customer Service Officer	35000.00	01914455667
120	2	Nafisa Islam	Marketing Officer	36000.00	01620011223
116	2	Sultana Begum	Accountant	37000.00	01516677889
118	4	Mita Chowdhury	IT Officer	38000.00	01818899001
112	2	Nasrin Akter	Assistant Manager	40000.00	01712233445
117	3	Hasan Ali	Loan Officer	42000.00	01717788990
111	1	Md. Rahim	Branch Manager	50000.00	01511122334

2. Write an SQL query to display employees ordered by branch\_id in descending order.

The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The left sidebar lists various tables and databases. The main area displays the results of a query: 'SELECT \* FROM employees ORDER BY branch\_id DESC;'. The results table shows 10 rows of employee data, sorted by branch\_id in descending order. The columns are: employee\_id, branch\_id, name, position, salary, and contact\_number. The data includes employees like Mita Chowdhury (branch\_id 4), Farzana Sultan (branch\_id 4), Shamim Hossain (branch\_id 3), Hasan Ali (branch\_id 2), Nafisa Islam (branch\_id 2), Sultana Begum (branch\_id 2), Nasrin Akter (branch\_id 2), Abdul Karim (branch\_id 1), Tanvir Ahmed (branch\_id 1), and Md. Rahim (branch\_id 1).

employee_id	branch_id	name	position	salary	contact_number
118	4	Mita Chowdhury	IT Officer	38000.00	01818899001
114	4	Farzana Sultan	Customer Service Officer	35000.00	01914455667
113	3	Shamim Hossain	Cashier	30000.00	01813344556
117	3	Hasan Ali	Loan Officer	42000.00	01717788990
120	2	Nafisa Islam	Marketing Officer	36000.00	01620011223
116	2	Sultana Begum	Accountant	37000.00	01516677889
112	2	Nasrin Akter	Assistant Manager	40000.00	01712233445
115	1	Abdul Karim	Security Officer	25000.00	01615566778
119	1	Tanvir Ahmed	Clerk	22000.00	01919900112
111	1	Md. Rahim	Branch Manager	50000.00	01511122334

## Using LIMIT

1. Write an SQL query to fetch the top 3 employees with the highest salary.

The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The left sidebar lists various tables and databases. The main area displays the results of a query: 'SELECT \* FROM employees ORDER BY salary DESC LIMIT 3;'. The results table shows 3 rows of employee data, sorted by salary in descending order. The columns are: employee\_id, branch\_id, name, position, salary, and contact\_number. The data includes employees like Md. Rahim (salary 50000.00), Hasan Ali (salary 42000.00), and Nasrin Akter (salary 40000.00).

employee_id	branch_id	name	position	salary	contact_number
111	1	Md. Rahim	Branch Manager	50000.00	01511122334
117	3	Hasan Ali	Loan Officer	42000.00	01717788990
112	2	Nasrin Akter	Assistant Manager	40000.00	01712233445

2. Write an SQL query to retrieve the first 5 employees from the employees table.

The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The 'employees' table is selected. A query has been run: `SELECT * FROM employees LIMIT 5;`. The results show five rows of employee data:

employee_id	branch_id	name	position	salary	contact_number
111	1	Md. Rahim	Branch Manager	50000.00	01511122334
112	2	Nasrin Akter	Assistant Manager	40000.00	01712233445
113	3	Shamim Hossain	Cashier	30000.00	01813344556
114	4	Farzana Sultana	Customer Service Officer	35000.00	01914455667
115	1	Abdul Karim	Security Officer	25000.00	01615566778

## Using IN

1. Write an SQL query to find employees whose position is IN ("Loan Officer", "IT Officer").

The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The 'employees' table is selected. A query has been run: `SELECT * FROM employees WHERE position IN ('Loan Officer', 'IT Officer');`. The results show two rows of employee data:

employee_id	branch_id	name	position	salary	contact_number
117	3	Hasan Ali	Loan Officer	42000.00	01717788990
118	4	Mita Chowdhury	IT Officer	38000.00	01818899001

2. Write an SQL query to fetch employees with branch\_id IN (1, 4, 2).

The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The 'employees' table is selected. A query has been run: `SELECT * FROM employees WHERE branch_id IN (1, 4, 2);`. The results show 8 rows of employee data:

employee_id	branch_id	name	position	salary	contact_number
111	1	Md. Rahim	Branch Manager	50000.00	01511122334
112	2	Nasrin Akter	Assistant Manager	40000.00	01712233445
114	4	Farzana Sultana	Customer Service Officer	35000.00	01914455667
115	1	Abdul Karim	Security Officer	25000.00	01615566778
116	2	Sultana Begum	Accountant	37000.00	01516677889
118	4	Mita Chowdhury	IT Officer	38000.00	01818899001
119	1	Tanvir Ahmed	Clerk	22000.00	01919900112
120	2	Nafisa Islam	Marketing Officer	36000.00	01620011223

## Aggregate Functions with Questions

### Using SUM ()

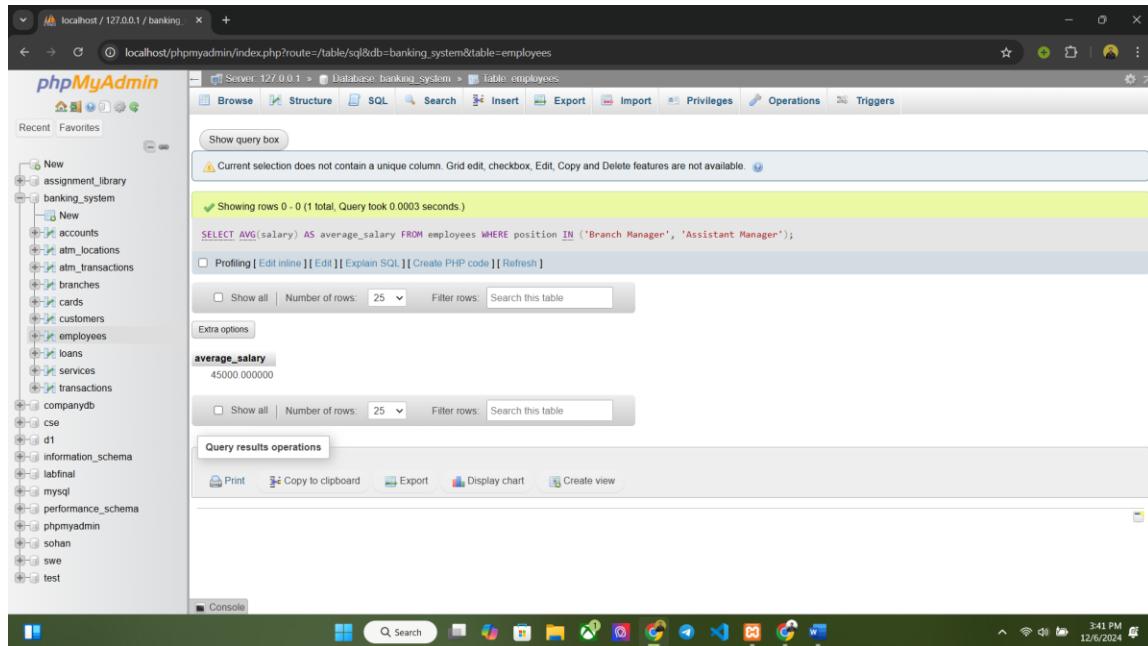
1. Write an SQL query to calculate the total salary of all employees in the employees table.

The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The 'employees' table is selected. A query has been run: `SELECT SUM(salary) AS total_salary FROM employees;`. The results show a single row with the total salary:

total_salary
355000.00

## Using AVG ()

1. Write an SQL query to find the average salary of employees with the position **IN** ("Branch Manager", "Assistant Manager").



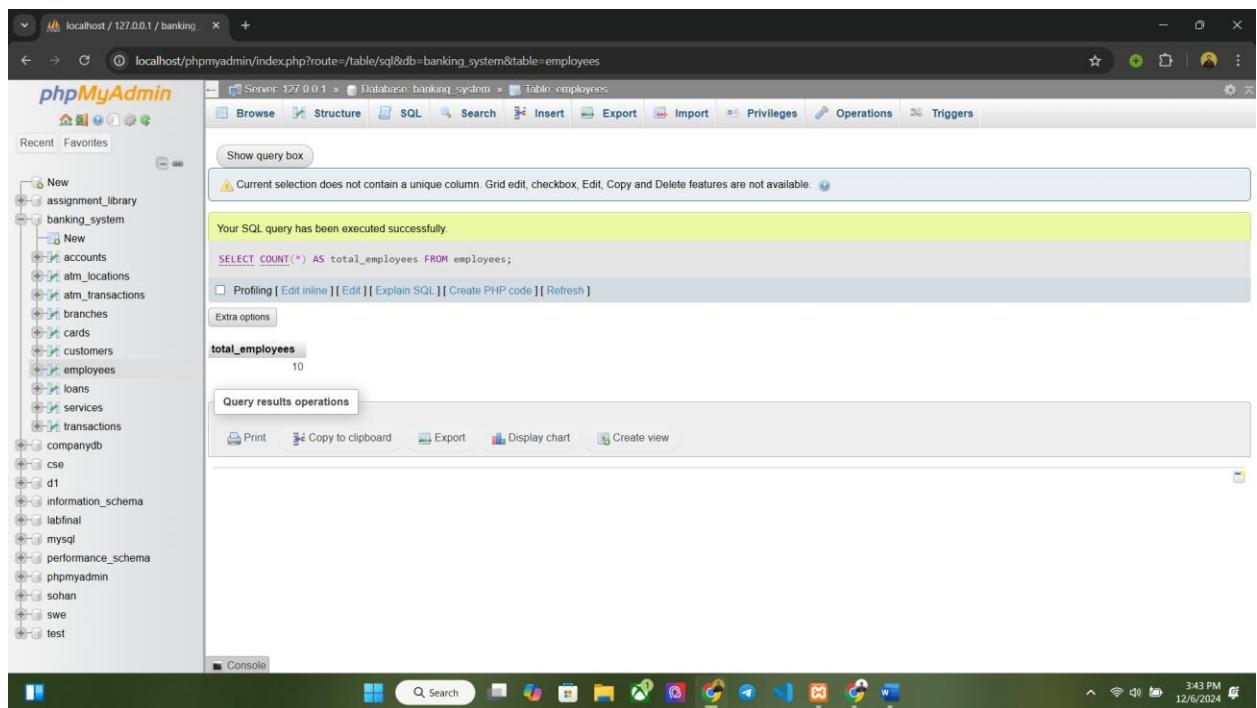
The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The 'employees' table is selected. In the SQL tab, the following query is run:

```
SELECT AVG(salary) AS average_salary FROM employees WHERE position IN ('Branch Manager', 'Assistant Manager');
```

The results show a single row with the value '45000 00000' under the 'average\_salary' column.

## Using COUNT ()

1. Write an SQL query to count the total number of employees in the employees table.



The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The 'employees' table is selected. In the SQL tab, the following query is run:

```
SELECT COUNT(*) AS total_employees FROM employees;
```

The results show a single row with the value '10' under the 'total\_employees' column.

## Using MIN ()

1. Write an SQL query to find the minimum salary of employees with the position "Cashier".

The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The left sidebar lists various tables: assignment\_library, banking\_system, accounts, atm\_locations, atm\_transactions, branches, cards, customers, employees, loans, services, transactions, companydb, cse, d1, information\_schema, labfinal, mysql, performance\_schema, phpmyadmin, sohan, swe, and test. The 'employees' table is selected. The main area displays the results of the query: `SELECT MIN(salary) AS minimum_salary FROM employees WHERE position = 'Cashier';`. The result is a single row with the value 30000.00 under the 'minimum\_salary' column. The status bar at the bottom right shows the date and time as 12/6/2024 3:44 PM.

## Using MAX ()

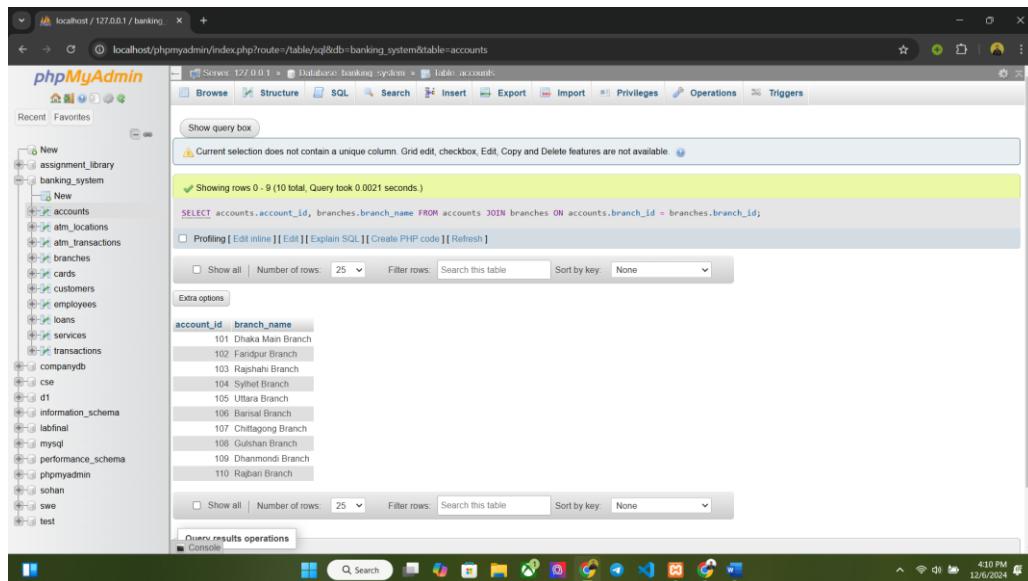
1. Write an SQL query to find the maximum salary of employees working in branch\_id = 2.

The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The left sidebar lists various tables: assignment\_library, banking\_system, accounts, atm\_locations, atm\_transactions, branches, cards, customers, employees, loans, services, transactions, companydb, cse, d1, information\_schema, labfinal, mysql, performance\_schema, phpmyadmin, sohan, swe, and test. The 'employees' table is selected. The main area displays the results of the query: `SELECT MAX(salary) AS maximum_salary FROM employees WHERE branch_id = 2;`. The result is a single row with the value 40000.00 under the 'maximum\_salary' column. The status bar at the bottom right shows the date and time as 12/6/2024 4:08 PM.

# Questions About SQL Join & Join With Condition

## General Join

1. Write an SQL query to combine the accounts and branches tables to get account\_id and branch\_name.



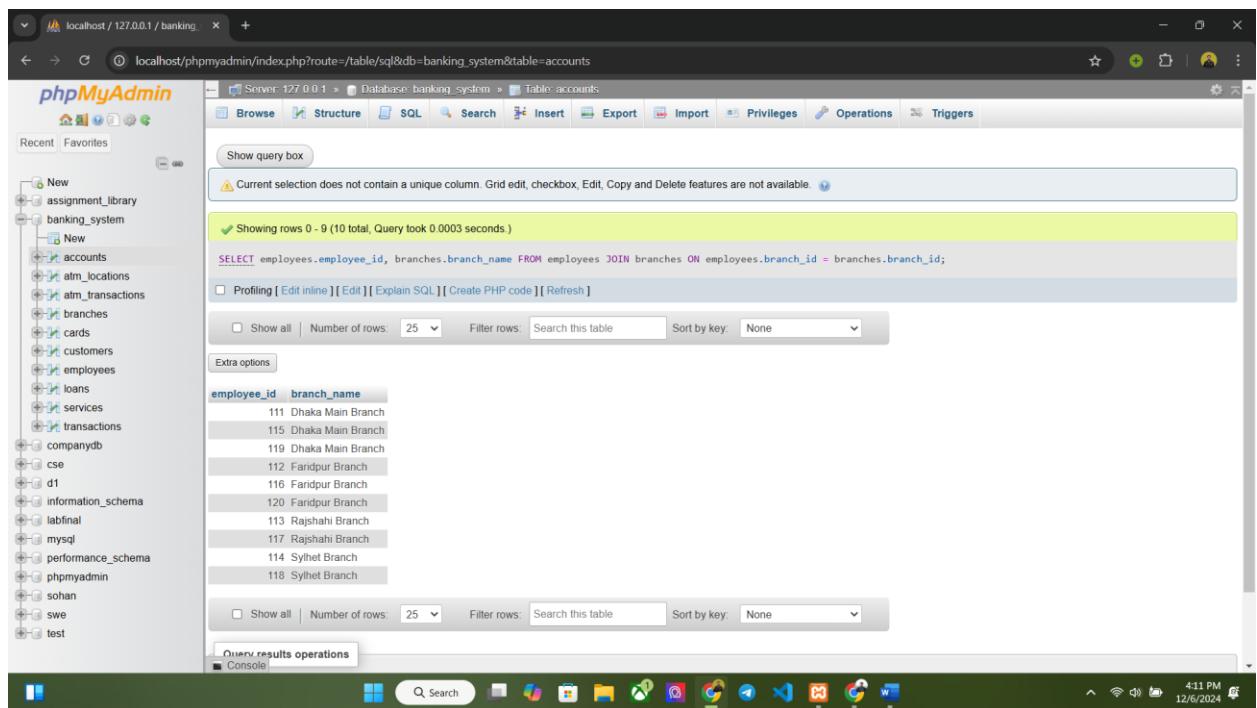
The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The 'accounts' table is selected. The results of the following SQL query are displayed:

```
SELECT accounts.account_id, branches.branch_name FROM accounts JOIN branches ON accounts.branch_id = branches.branch_id;
```

The results show a list of account IDs and their corresponding branch names:

account_id	branch_name
101	Dhaka Main Branch
102	Fardpur Branch
103	Rajshahi Branch
104	Sylhet Branch
105	Uttara Branch
106	Barisal Branch
107	Chittagong Branch
108	Gulshan Branch
109	Dhamondi Branch
110	Rajbari Branch

2. Write an SQL query to join the employees and branches tables to fetch employee\_id and branch\_name.



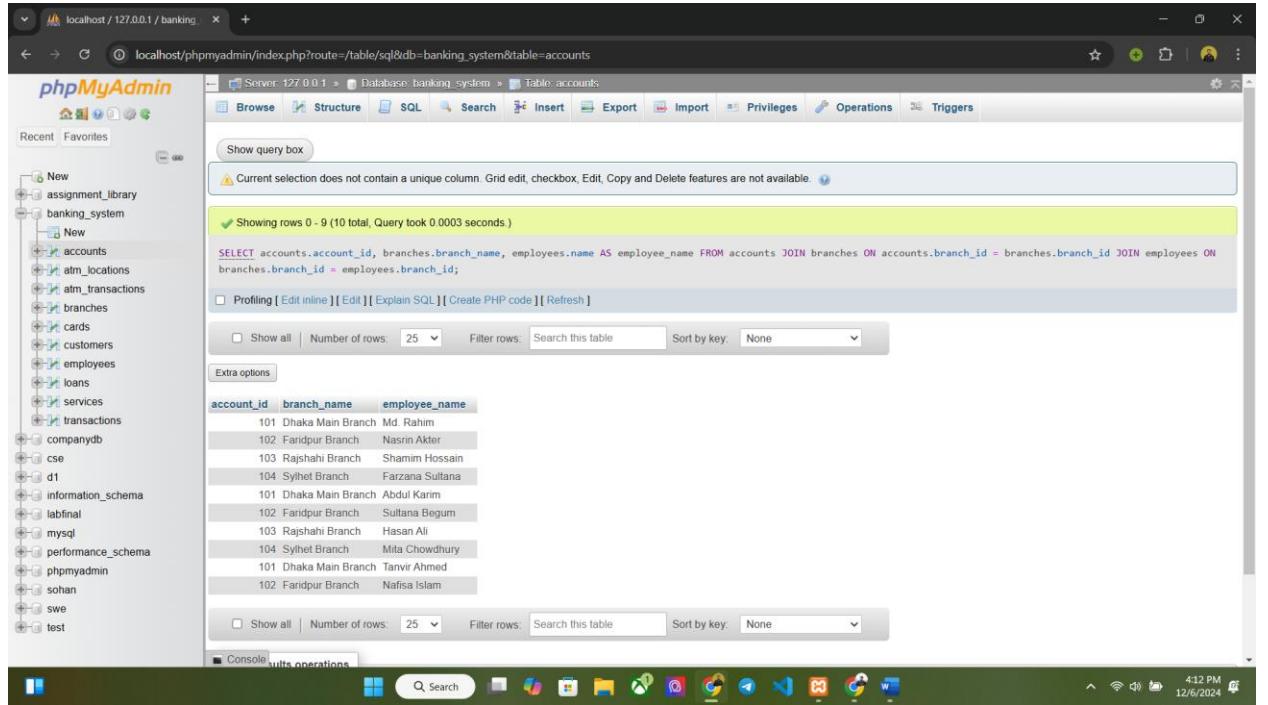
The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The 'accounts' table is selected. The results of the following SQL query are displayed:

```
SELECT employees.employee_id, branches.branch_name FROM employees JOIN branches ON employees.branch_id = branches.branch_id;
```

The results show a list of employee IDs and their corresponding branch names:

employee_id	branch_name
111	Dhaka Main Branch
115	Dhaka Main Branch
119	Dhaka Main Branch
112	Fardpur Branch
116	Fardpur Branch
120	Fardpur Branch
113	Rajshahi Branch
117	Rajshahi Branch
114	Sylhet Branch
118	Sylhet Branch

3. Write an SQL query to combine the accounts, branches, and employees tables to retrieve account\_id, branch\_name, and employee name.



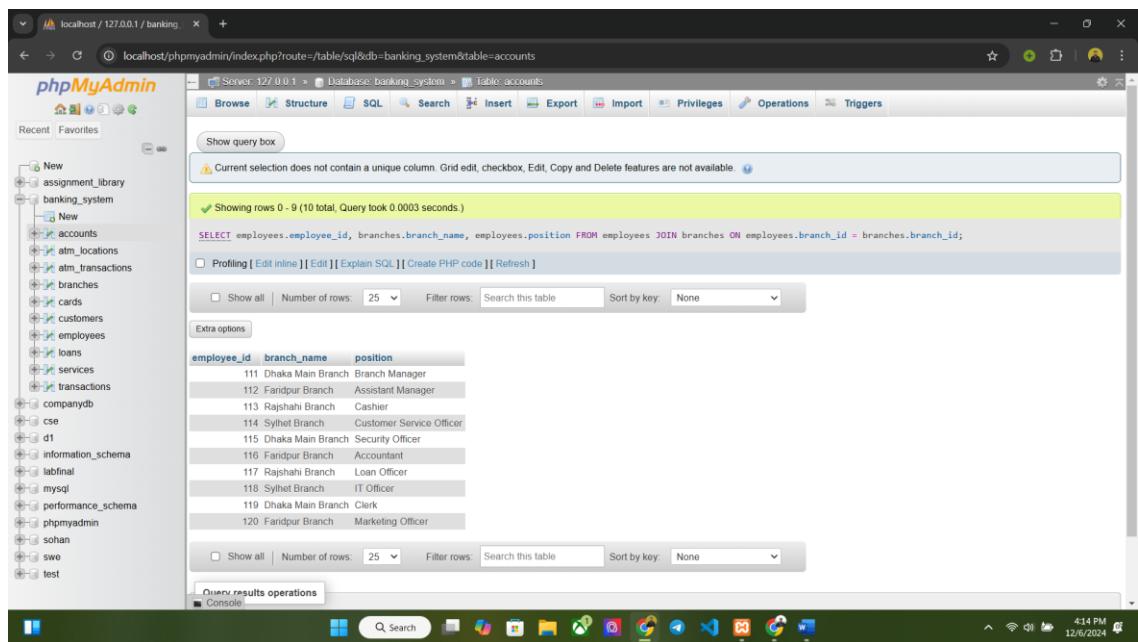
The screenshot shows the phpMyAdmin interface for the 'banking\_system' database. The 'accounts' table is selected. The results of the following SQL query are displayed:

```
SELECT accounts.account_id, branches.branch_name, employees.name AS employee_name FROM accounts JOIN branches ON accounts.branch_id = branches.branch_id JOIN employees ON branches.branch_id = employees.branch_id;
```

The results table contains 10 rows:

account_id	branch_name	employee_name
101	Dhaka Main Branch	Md. Rahim
102	Fardipur Branch	Nasrin Akter
103	Rajshahi Branch	Shamim Hossain
104	Sylhet Branch	Fazhana Sultana
101	Dhaka Main Branch	Abdul Karim
102	Fardipur Branch	Sultana Begum
103	Rajshahi Branch	Hasan Ali
104	Sylhet Branch	Mita Chowdhury
101	Dhaka Main Branch	Tanvir Ahmed
102	Fardipur Branch	Nafisa Islam

4. Write an SQL query to join the employees and branches tables on branch\_id and include their position.



The screenshot shows the phpMyAdmin interface for the 'banking\_system' database. The 'accounts' table is selected. The results of the following SQL query are displayed:

```
SELECT employees.employee_id, branches.branch_name, employees.position FROM employees JOIN branches ON employees.branch_id = branches.branch_id;
```

The results table contains 12 rows:

employee_id	branch_name	position
111	Dhaka Main Branch	Branch Manager
112	Fardipur Branch	Assistant Manager
113	Rajshahi Branch	Cashier
114	Sylhet Branch	Customer Service Officer
115	Dhaka Main Branch	Security Officer
116	Fardipur Branch	Accountant
117	Rajshahi Branch	Loan Officer
118	Sylhet Branch	IT Officer
119	Dhaka Main Branch	Clerk
120	Fardipur Branch	Marketing Officer

5. Write an SQL query to combine the accounts and employees tables to fetch account\_id and employee\_id.

The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The left sidebar lists tables such as accounts, atm\_locations, atm\_transactions, branches, cards, customers, employees, loans, services, and transactions. The 'accounts' table is selected. The main area displays a grid of data with columns 'account\_id' and 'employee\_id'. A SQL query at the top of the results page is:

```
SELECT accounts.account_id, employees.employee_id FROM accounts JOIN employees ON accounts.branch_id = employees.branch_id;
```

The results show 10 rows of data:

account_id	employee_id
101	111
101	115
101	119
102	112
102	116
102	120
103	113
103	117
104	114
104	118

## Inner Join

1. Write an SQL query to fetch the account\_id, branch\_name, and employee name for all accounts by joining accounts, branches, and employees.

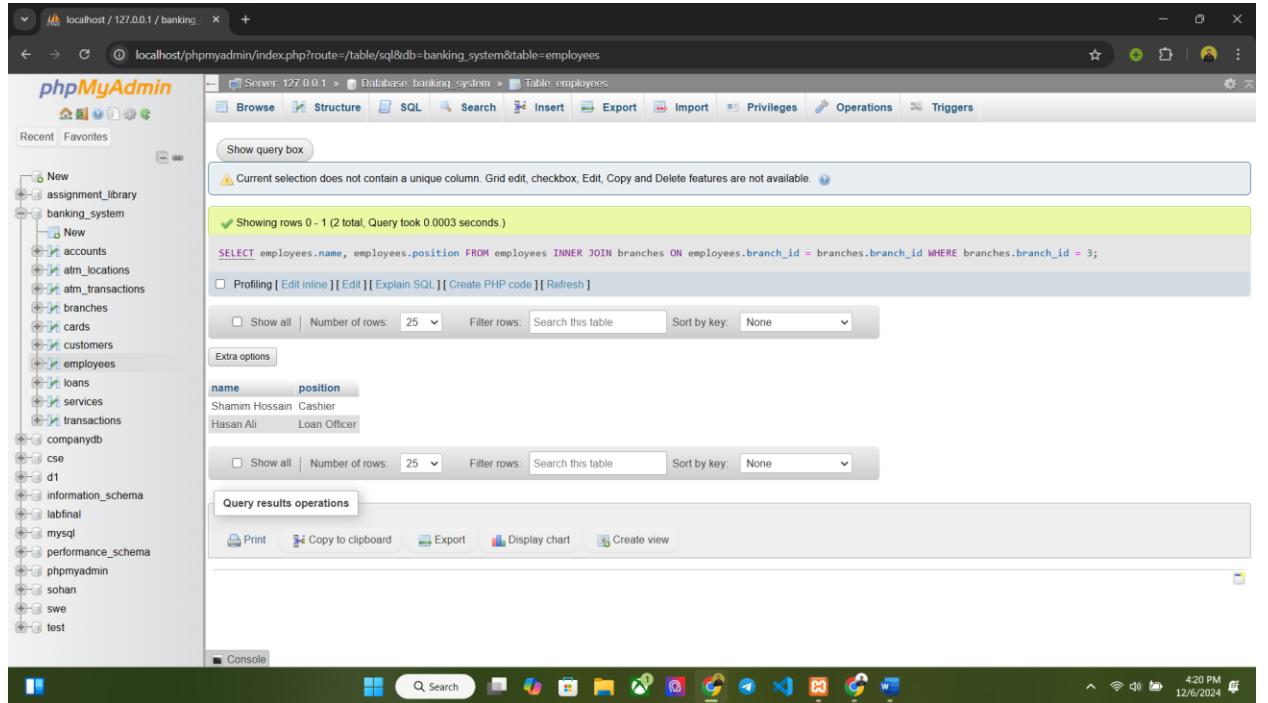
The screenshot shows the phpMyAdmin interface for the same database. The 'accounts' table is selected. The main area displays a grid of data with columns 'account\_id', 'branch\_name', and 'employee\_name'. A SQL query at the top of the results page is:

```
SELECT accounts.account_id, branches.branch_name, employees.name AS employee_name FROM accounts INNER JOIN branches ON accounts.branch_id = branches.branch_id INNER JOIN employees ON branches.branch_id = employees.branch_id;
```

The results show 10 rows of data:

account_id	branch_name	employee_name
101	Dhaka Main Branch	Md. Rahim
102	Fardpur Branch	Nasrin Akter
103	Rajshahi Branch	Shamim Hossain
104	Sylhet Branch	Farzana Sultana
101	Dhaka Main Branch	Abdul Karim
102	Fardpur Branch	Sultana Begum
103	Rajshahi Branch	Hasan Ali
104	Sylhet Branch	Mita Chowdhury
101	Dhaka Main Branch	Tanvir Ahmed
102	Fardpur Branch	Nafisa Islam

2. Write an SQL query to retrieve the name and position of employees who belong to branches with a branch\_id = 3.



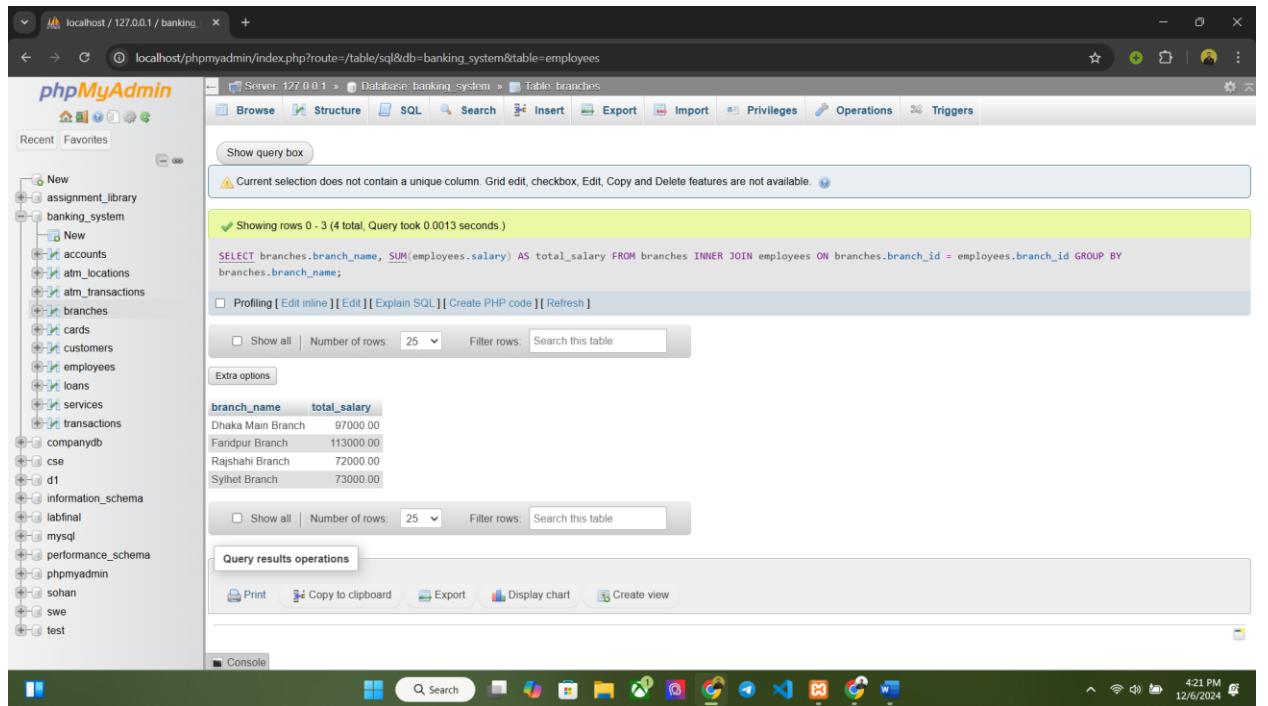
The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The left sidebar lists various tables: assignment\_library, banking\_system, accounts, atm\_locations, atm\_transactions, branches, cards, customers, employees, loans, services, transactions, companydb, cse, d1, information\_schema, labfinal, mysql, performance\_schema, phpmadmin, sohan, swe, test. The 'employees' table is selected. The main panel displays the results of the following SQL query:

```
SELECT employees.name, employees.position FROM employees INNER JOIN branches ON employees.branch_id = branches.branch_id WHERE branches.branch_id = 3;
```

The results show two rows:

name	position
Shamim Hossain	Cashier
Hasan Ali	Loan Officer

3. Write an SQL query to fetch branch\_name and the total salary of employees in each branch.



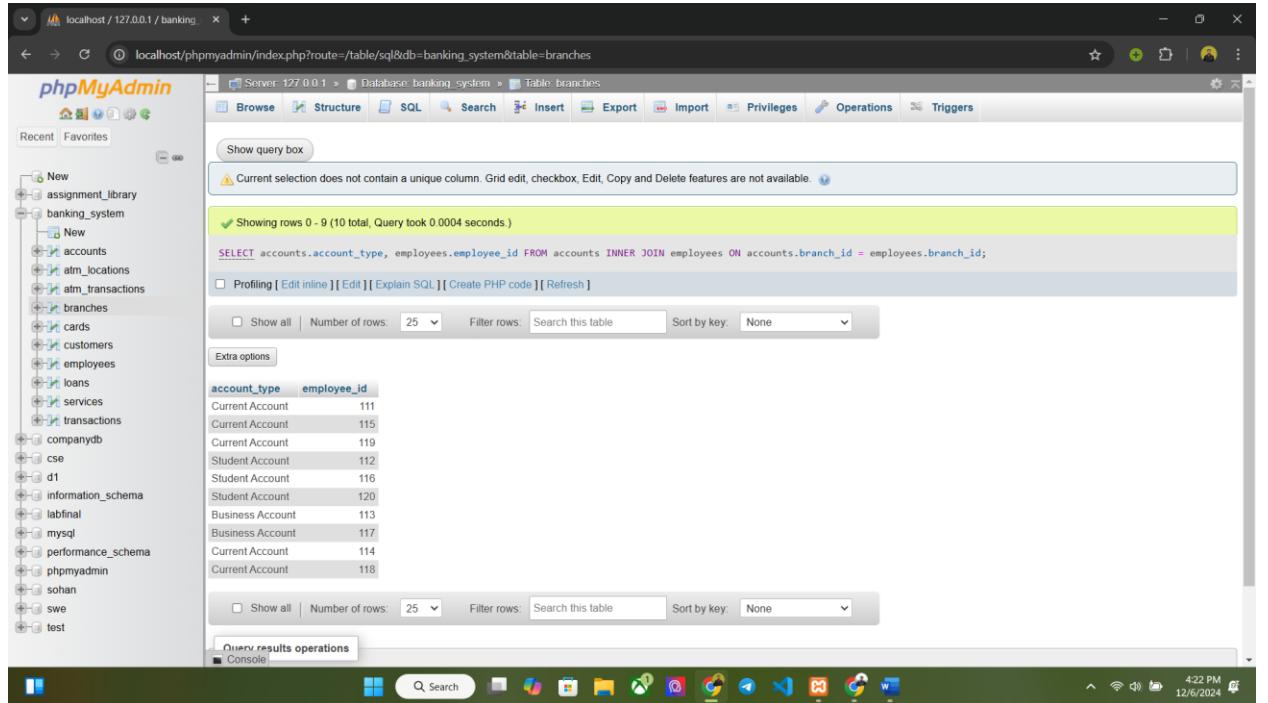
The screenshot shows the phpMyAdmin interface for the same database. The 'branches' table is selected. The main panel displays the results of the following SQL query:

```
SELECT branches.branch_name, SUM(employees.salary) AS total_salary FROM branches INNER JOIN employees ON branches.branch_id = employees.branch_id GROUP BY branches.branch_name;
```

The results show four branches with their respective total salaries:

branch_name	total_salary
Dhaka Main Branch	97000.00
Fardpur Branch	113000.00
Rajshahi Branch	72000.00
Sylhet Branch	73000.00

4. Write an SQL query to find the account\_type and employee\_id for employees managing accounts in the same branch.



The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The left sidebar lists various tables: assignment\_library, banking\_system (with sub-tables accounts, atm\_locations, atm\_transactions, branches, cards, customers, employees, loans, services, transactions), companydb, cse, d1, information\_schema, labfinal, mysql, performance\_schema, phpmadmin, sohan, swe, and test. The 'branches' table is selected in the main area. A SQL query is displayed:

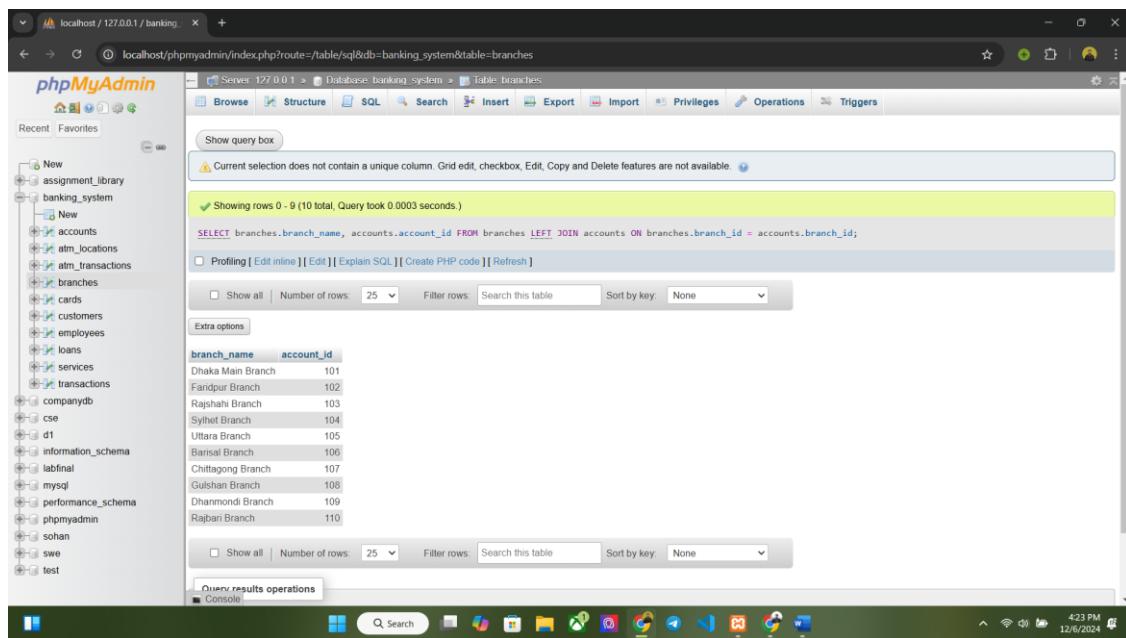
```
SELECT accounts.account_type, employees.employee_id FROM accounts INNER JOIN employees ON accounts.branch_id = employees.branch_id;
```

The results show a list of account types and their corresponding employee IDs:

account_type	employee_id
Current Account	111
Current Account	115
Current Account	119
Student Account	112
Student Account	116
Student Account	120
Business Account	113
Business Account	117
Current Account	114
Current Account	118

## Left Join

1. Write an SQL query to fetch all branch\_name from branches along with their respective account\_id, even if no accounts exist in some branches.



The screenshot shows the phpMyAdmin interface for the same database. The 'branches' table is selected. A SQL query is displayed:

```
SELECT branches.branch_name, accounts.account_id FROM branches LEFT JOIN accounts ON branches.branch_id = accounts.branch_id;
```

The results show the branch names and their corresponding account IDs, including branches without accounts:

branch_name	account_id
Dhaka Main Branch	101
Fardpur Branch	102
Rajshahi Branch	103
Sylhet Branch	104
Uttara Branch	105
Bansal Branch	106
Chittagong Branch	107
Gulshan Branch	108
Dharmodi Branch	109
Rajbari Branch	110

2. Write an SQL query to retrieve all employees and their branch details, including employees who do not belong to any branch.

The screenshot shows the phpMyAdmin interface with the 'branches' table selected. The results of the following query are displayed:

```
SELECT employees.*, branches.branch_name FROM employees LEFT JOIN branches ON employees.branch_id = branches.branch_id;
```

The results show 16 rows of data, including employees from various branches like Dhaka Main Branch, Fardpur Branch, Rajshahi Branch, and Sylhet Branch, along with some employees who do not have a branch assigned (branch\_id is NULL).

employee_id	branch_id	name	position	salary	contact_number	branch_name
111	1	Md. Rahim	Branch Manager	50000.00	01511122334	Dhaka Main Branch
112	2	Nasrin Akter	Assistant Manager	40000.00	01712233445	Fardpur Branch
113	3	Shamim Hossain	Cashier	30000.00	01813344556	Rajshahi Branch
114	4	Farzana Sultan	Customer Service Officer	35000.00	01914455667	Sylhet Branch
115	1	Abdul Karim	Security Officer	25000.00	01615566778	Dhaka Main Branch
116	2	Sultana Begum	Accountant	37000.00	01516677889	Fardpur Branch
117	3	Hasan Ali	Loan Officer	42000.00	01717788990	Rajshahi Branch
118	4	Mita Chowdhury	IT Officer	38000.00	01818899001	Sylhet Branch
119	1	Tanvir Ahmed	Clerk	22000.00	01919900112	Dhaka Main Branch
120	2	Nafisa Islam	Marketing Officer	36000.00	01620011223	Fardpur Branch

3. Write an SQL query to display all account\_id, branch\_name, and employee\_id, including accounts without employees.

The screenshot shows the phpMyAdmin interface with the 'branches' table selected. The results of the following query are displayed:

```
SELECT accounts.account_id, branches.branch_name, employees.employee_id FROM accounts LEFT JOIN branches ON accounts.branch_id = branches.branch_id LEFT JOIN employees ON accounts.branch_id = employees.branch_id;
```

The results show 16 rows of data, including accounts from various branches like Dhaka Main Branch, Fardpur Branch, Rajshahi Branch, and Sylhet Branch, along with some accounts that do not have an employee assigned (employee\_id is NULL).

account_id	branch_name	employee_id
101	Dhaka Main Branch	111
101	Dhaka Main Branch	115
101	Dhaka Main Branch	119
102	Fardpur Branch	112
102	Fardpur Branch	116
102	Fardpur Branch	120
103	Rajshahi Branch	113
103	Rajshahi Branch	117
104	Sylhet Branch	114
104	Sylhet Branch	118
105	Uttara Branch	NULL
106	Barisal Branch	NULL
107	Chittagong Branch	NULL
108	Gulshan Branch	NULL

# Right Join

1. Write an SQL query to display all account\_id and their respective branch\_name, even if no branches are linked to accounts.

The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The left sidebar lists various tables: assignment\_library, banking\_system, accounts, atm\_locations, atm\_transactions, branches, cards, customers, employees, loans, services, transactions, companydb, cse, d1, information\_schema, labfinal, mysql, performance\_schema, phpmysqladmin, sohan, swe, and test. The 'branches' table is selected. The main area displays the results of a SQL query: `SELECT accounts.account_id, branches.branch_name FROM accounts RIGHT JOIN branches ON accounts.branch_id = branches.branch_id;`. The results show 9 rows, each mapping an account ID to a branch name. Some account IDs have NULL values in the branch\_name column, indicating no link to a branch.

account_id	branch_name
101	Dhaka Main Branch
102	Faridpur Branch
103	Rajshahi Branch
104	Sylhet Branch
105	Uttara Branch
106	Barisal Branch
107	Chittagong Branch
108	Gulshan Branch
109	Dhanmondi Branch
110	Rajbari Branch

2. Write an SQL query to fetch employee\_id and their respective branch\_name, even if no branches are assigned to employees.

The screenshot shows the phpMyAdmin interface for the same database. The 'branches' table is selected. The main area displays the results of a SQL query: `SELECT employees.employee_id, branches.branch_name FROM employees RIGHT JOIN branches ON employees.branch_id = branches.branch_id;`. The results show 16 rows, each mapping an employee ID to a branch name. Many employee IDs have NULL values in the branch\_name column, indicating no link to a branch.

employee_id	branch_name
111	Dhaka Main Branch
115	Dhaka Main Branch
119	Dhaka Main Branch
112	Faridpur Branch
116	Faridpur Branch
120	Faridpur Branch
113	Rajshahi Branch
117	Rajshahi Branch
114	Sylhet Branch
118	Sylhet Branch
NULL	Uttara Branch
NULL	Barisal Branch
NULL	Chittagong Branch
NULL	Gulshan Branch
NULL	Dhanmondi Branch

3. Write an SQL query to display all branch\_name and account\_id, including branches that do not have any accounts.

The screenshot shows the phpMyAdmin interface for the 'banking\_system' database. The 'branches' table is selected. The results of the following query are displayed:

```
SELECT branches.branch_name, accounts.account_id FROM branches RIGHT JOIN accounts ON branches.branch_id = accounts.branch_id;
```

The results show 10 rows of data:

branch_name	account_id
Dhaka Main Branch	101
Fardpur Branch	102
Rajshahi Branch	103
Sylhet Branch	104
Uttara Branch	105
Barisal Branch	106
Chittagong Branch	107
Gulshan Branch	108
Dhamondi Branch	109
Rajbari Branch	110

## Full Outer Join

1. Write an SQL query to display all branch\_name and account\_id, including rows that are unmatched in either branches or accounts.

The screenshot shows the phpMyAdmin interface for the 'banking\_system' database. The 'branches' table is selected. The results of the following query are displayed:

```
SELECT branches.branch_name, accounts.account_id FROM branches LEFT JOIN accounts ON branches.branch_id = accounts.branch_id UNION SELECT branches.branch_name, accounts.account_id FROM accounts RIGHT JOIN branches ON accounts.branch_id = branches.branch_id;
```

The results show 10 rows of data, matching the previous screenshot:

branch_name	account_id
Dhaka Main Branch	101
Fardpur Branch	102
Rajshahi Branch	103
Sylhet Branch	104
Uttara Branch	105
Barisal Branch	106
Chittagong Branch	107
Gulshan Branch	108
Dhamondi Branch	109
Rajbari Branch	110

2. Write an SQL query to display all branch\_id and employee\_id, even if rows do not have matches in the other table.

The screenshot shows the phpMyAdmin interface for the 'banking\_system' database. The 'branches' table is selected. The results of the following query are displayed:

```
SELECT branches.branch_id, employees.employee_id FROM branches LEFT JOIN employees ON branches.branch_id = employees.branch_id UNION SELECT branches.branch_id, employees.employee_id FROM employees RIGHT JOIN branches ON employees.branch_id = branches.branch_id;
```

The results table shows the following data:

branch_id	employee_id
1	111
1	115
1	119
2	112
2	116
2	120
3	113
3	117
4	114
4	118
5	NULL
6	NULL
7	NULL
8	NULL

## Subquery Questions

1. Write an SQL query to find the name and position of employees who work in branches where the branch\_id exists in the accounts table.

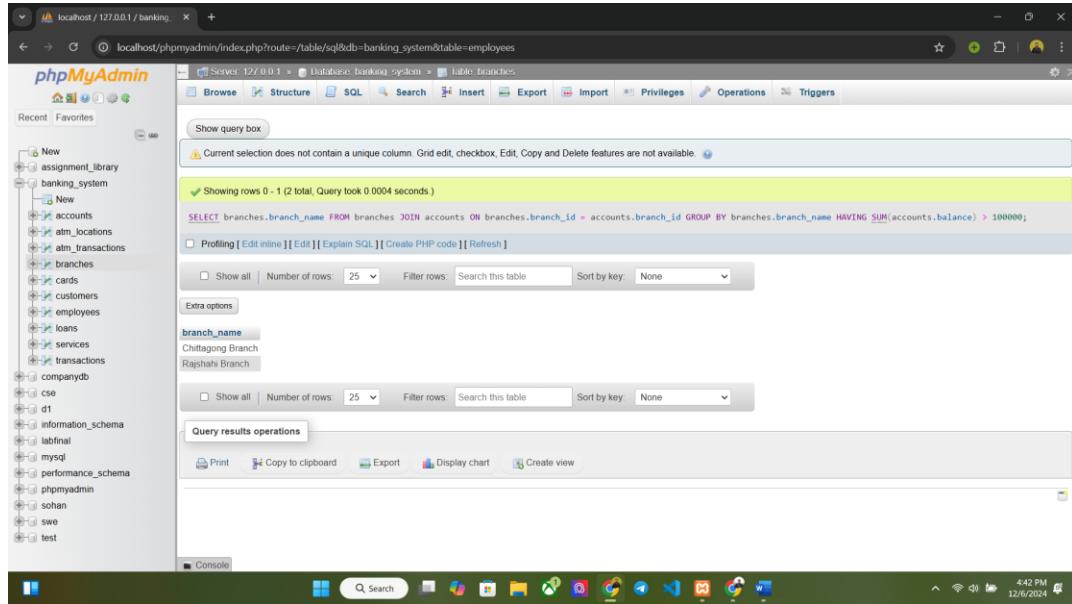
The screenshot shows the phpMyAdmin interface for the 'banking\_system' database. The 'employees' table is selected. The results of the following query are displayed:

```
SELECT employees.name, employees.position FROM employees WHERE employees.branch_id IN (SELECT DISTINCT branch_id FROM accounts);
```

The results table shows the following data:

	name	position
0	Md. Rahim	Branch Manager
1	Nasrin Akter	Assistant Manager
2	Shamim Hossain	Cashier
3	Farzana Sultana	Customer Service Officer
4	Abdul Karim	Security Officer
5	Sultana Begum	Accountant
6	Hasan Ali	Loan Officer
7	Mita Chowdhury	IT Officer
8	Tanvir Ahmed	Clerk
9	Nafisa Islam	Marketing Officer

2. Write an SQL query to fetch the branch\_name of branches where the total balance of accounts in that branch is greater than 100,000.



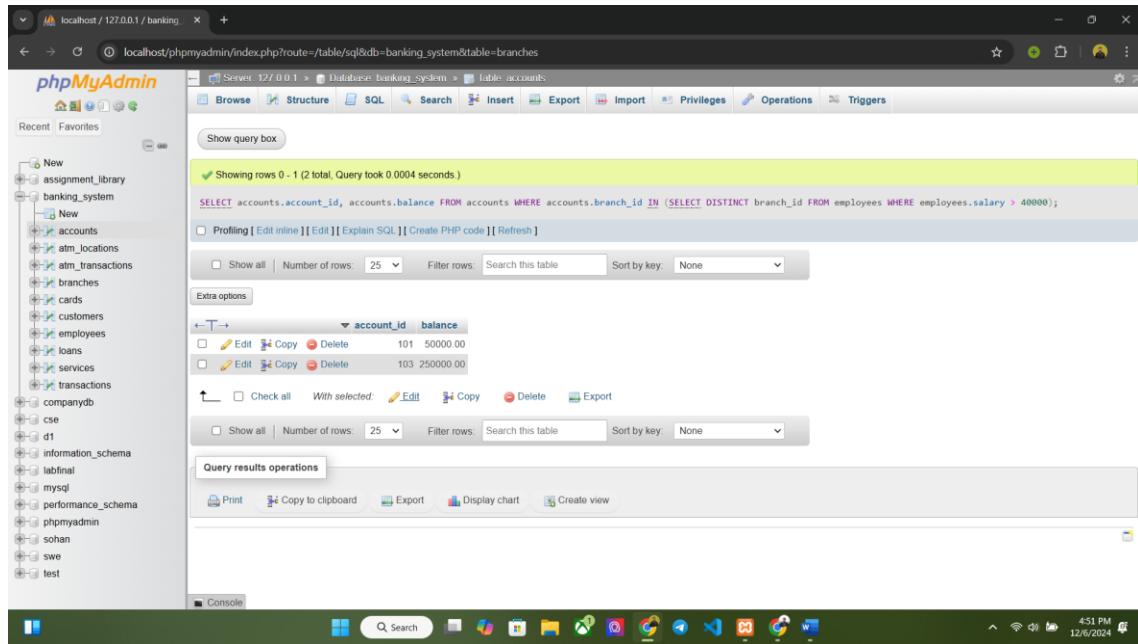
The screenshot shows the phpMyAdmin interface for a database named 'banking\_system'. The left sidebar lists various tables: assignment\_library, banking\_system (accounts, atm\_locations, atm\_transactions, branches, cards, customers, employees, loans, services, transactions), companydb, cse, d1, information\_schema, labfinal, mysql, performance\_schema, phpmyadmin, sohan, swe, and test. The 'branches' table is selected. The main area displays the results of the following SQL query:

```
SELECT branches.branch_name FROM branches JOIN accounts ON branches.branch_id = accounts.branch_id GROUP BY branches.branch_name HAVING SUM(accounts.balance) > 100000;
```

The results show two rows:

branch_name
Chittagong Branch
Rajshahi Branch

3. Write an SQL query to display the account\_id and balance of accounts that have a branch\_id with employees earning more than 40,000.



The screenshot shows the phpMyAdmin interface for the same database. The 'accounts' table is selected. The main area displays the results of the following SQL query:

```
SELECT accounts.account_id, accounts.balance FROM accounts WHERE accounts.branch_id IN (SELECT DISTINCT branch_id FROM employees WHERE employees.salary > 40000);
```

The results show two rows:

account_id	balance
101	50000.00
103	250000.00

Use Case Diagram

Draw.io Link: [Use Case](#)

ER Diagram

Draw.io Link: [ERD](#)