Schema 1: Consider the following schemas for 'bank' database relations, where the primary keys are underlined

branch(*branch_name*, *branch_city, assets*)

customer(*customer_name*, *customer_street, customer_city*)

loan(*loan_number*, *branch_name, amount*)

borrower(*customer_name*, *loan_number*)

account(*account_number*, *branch_name, balance*)

depositor(*customer_name*, *account_number*)

Problems:

i) Find all customers who have both a loan and an account at the bank

ii) Find the average account balance at the 'perryridge' branch

iii) Insert record into the account relation with the values of account number is "AC-101" at "Dhanmondi" branch and balance is tk 30000

Create a database using sql query.

SQL:

```
CREATE DATABASE bank;
```

Create a table and insert value using sql query.

SQL:

```
CREATE TABLE branch (
        branch_name VARCHAR(55) NOT NULL,
        branch_city VARCHAR(55),
        assets INT(11),
        PRIMARY KEY (branch_name)
);

INSERT INTO branch (branch_name, branch_city, assets)
 VALUES
        ('Dhanmondi', 'Dhaka', 5000000),
        ('New London', 'London', 20000000),
        ('Perryridge', 'Dhake', 10000000),
        ('San Francisco', 'New York', 15000000);
```

+ Options

	branch_name	branch_city	assets
Edit Copy Delete	Dhanmondi	Dhaka	5000000
Edit Copy Delete	New London	London	20000000
Edit Copy Delete	Perryridge	Dhake	10000000
Edit Copy Delete	San Francisco	New York	15000000

Create a table and insert value using sql query.

SQL:

```
CREATE TABLE customer(
        customer_name VARCHAR(255) NOT NULL,
        customer_street VARCHAR(55),
        customer_city VARCHAR(55),
        PRIMARY KEY (customer_name)
);

INSERT INTO customer(customer_name, customer_street, customer_city)
VALUES
        ('John Doe', '123 Main Street', 'New York'),
        ('Jane Doe', '456 Elm Street', 'Dhaka'),
        ('Michael Jones', '456 Main Street', 'London'),
        ('Susan Smith', '123 Elm Street', 'San Francisco');
```

+ Options

	customer_name	customer_street	customer_city
Edit Copy Delete	Jane Doe	456 Elm Street	Dhaka
Edit Copy Delete	John Doe	123 Main Street	New York
Edit Copy Delete	Michael Jones	456 Main Street	London
Edit Copy Delete	Susan Smith	123 Elm Street	San Francisco

Create a table and insert value using sql query.

SQL:

```
CREATE TABLE loan (
        loan_number INT(11) NOT NULL,
        branch_name VARCHAR(255),
        amount INT(11),
        PRIMARY KEY (loan_number)
);
```

```
INSERT INTO loan(loan_number, branch_name, amount) VALUES
        ('123456', 'Perryidge', 10000),
        ('234567', 'San Francisco', 20000),
        ('789101', 'Dhanmondi', 5000),
        ('890123', 'London', 15000);
```

+ Options

	loan_number	branch_name	amount
Edit Copy Delete	123456	Perryidge	10000
Edit Copy Delete	234567	San Francisco	20000
Edit Copy Delete	789101	Dhanmondi	5000
Edit Copy Delete	890123	London	15000

Create a table and insert value using sql query.

SQL:

```
CREATE TABLE borrower (
        customer_name VARCHAR(255) NOT NULL,
        loan_number INT(11) NOT NULL,
        FOREIGN KEY (customer_name) REFERENCES customer
        (customer_name)
                on update cascade on delete cascade,
        FOREIGN KEY (loan_number) REFERENCES loan(loan_number)
                on update cascade on delete cascade
);

INSERT INTO borrower(customer_name, loan_number)
VALUES
        ('Susan Smith', '234567'),
        ('Michael Jones', '890123'),
        ('John Doe', '123456'),
        ('Jane Doe', '789101');
```

+ Options

customer_name	loan_number
Susan Smith	234567
Michael Jones	890123
John Doe	123456
Jane Doe	789101

Create a table and insert value using sql query.

SQL:

```
CREATE TABLE account (
        account_number VARCHAR(50) NOT NUll,
        branch_name VARCHAR(255) NOT NULL,
        balance INT(11),
        PRIMARY KEY(account_number)
);

INSERT INTO
account(account_number, branch_name, balance)
VALUES
        ('123456789', "Perryridge", 10000),
        ('987654321', 'Dhanmondi', 5000),
        ('321098765', 'San Francisco', 20000),
        ('654321098', 'London', 15000);
```

+ Options

	account_number	branch_name	balance
Edit Copy Delete	123456789	Perryidge	10000
Edit Copy Delete	321098765	San Francisco	20000
Edit Copy Delete	654321098	London	15000
Edit Copy Delete	987654321	Dhanmondi	5000

Create a table and insert value using sql query.

SQL:

```
CREATE TABLE depositor(
        customer_name VARCHAR(255) NOT NULL,
        account_number VARCHAR(50) NOT NULL,
        FOREIGN KEY(customer_name) REFERENCES
        customer(customer_name)
                on update cascade on delete cascade,
        FOREIGN KEY(account_number) REFERENCES account
        (account_number)
                on update cascade on delete cascade
);

INSERT INTO depositor(customer_name, account_number)
VALUES
        ('Susan Smith', '321098765'),
        ('Michael Jones', '654321098'),
        ('John Doe', '123456789'),
        ('Jane Doe', '987654321');
```

+ Options

customer_name	account_number
Susan Smith	321098765
Michael Jones	654321098
John Doe	123456789
Jane Doe	987654321

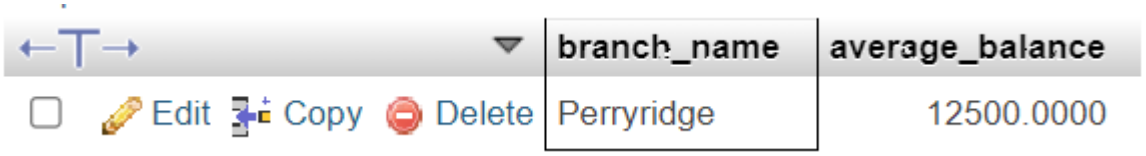
i) Find all customers who have both a loan and an account at the bank

```
SELECT customer_name FROM borrower
    WHERE customer_name IN (
        SELECT customer_name FROM depositor
    );
```

customer_name
Jane Doe
John Doe
Michael Jones
Susan Smith

ii) Find the average account balance at the ‘perryridge’ branch

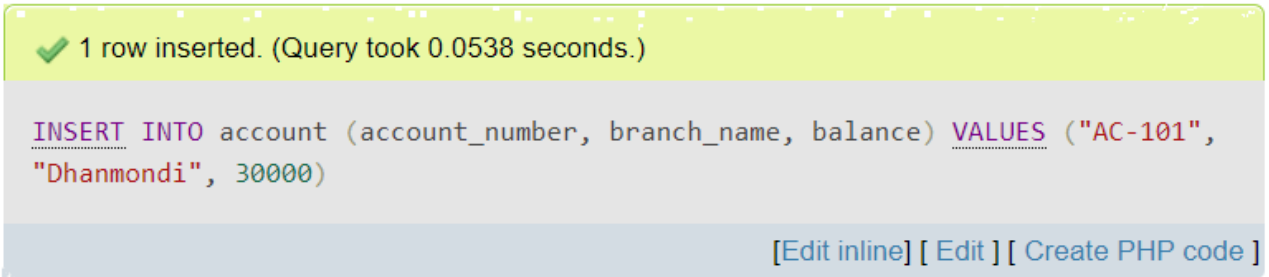
```
SELECT branch_name, AVG(balance) AS
average_balance FROM account

    WHERE branch_name = "Perryridge";
```



	branch_name	average_balance
Edit Copy Delete	Perryridge	12500.0000

iii) Insert record into the account relation with the values of account number is “AC-101” at “Dhanmondi” branch and balance is tk 30000

```
INSERT INTO account (account_number,
branch_name, balance)

VALUE("AC-101", "Dhanmondi", 30000);
```



1 row inserted. (Query took 0.0538 seconds.)

```
INSERT INTO account (account_number, branch_name, balance) VALUES ("AC-101",
"Dhanmondi", 30000)
```

[Edit inline] [Edit] [Create PHP code]

Schema 2: Consider the following schemas for 'bank' database relations, where the primary keys are underlined

branch(*branch_name*, *branch_city, assets*)

customer(*customer_name*, *customer_street, customer_city*)

loan(*loan_number*, *branch_name, amount*)

borrower(*customer_name*, *loan_number*)

account(*account_number*, *branch_name, balance*)

depositor(*customer_name*, *account_number*)

Problems:

i) find the number of depositor at each branch

ii) list in alphabetic order all customers who have a loan the “perryridge” branch

iii) Update database to change karim’s street to new one

i) Find the number of depositor at each branch

```
SELECT branch_name, COUNT(DISTINCT customer_name) AS num_depositors
FROM depositor, account
    WHERE
        account.account_number = depositor.account_number
    and
        account.branch_name IN (SELECT branch_name FROM branch) GROUP BY account.branch_name
```

branch_name	num_depositors
Dhanmondi	1
perryridge	2
San Francisco	1

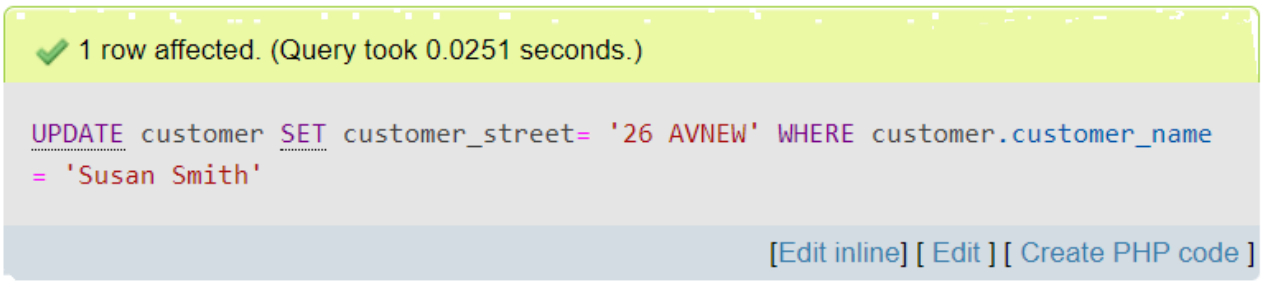
ii) List in alphabetical order all customers who have a loan the "perryridge" branch

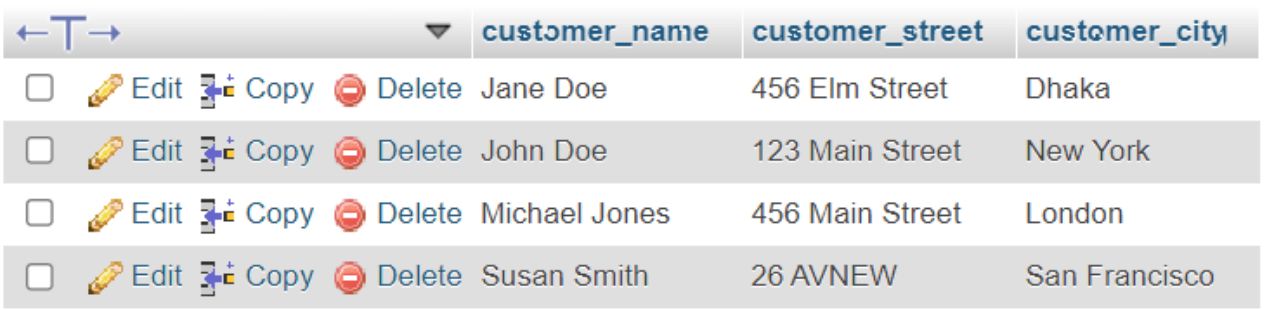
```
SELECT customer_name, branch_name

FROM borrower, loan

    WHERE

        borrower.loan_number =
    loan.loan_number

    AND

        loan.branch_name = 'perryridge' ORDER
    BY customer_name
```

customer_name	branch_name
John Doe	Perryridge

iii) Update database to change karim's street to new one

```
UPDATE customer
    SET customer_street= '26 AVNEW'
    WHERE customer.customer_name = 'Susan Smith'
```





	customer_name	customer_street	customer_city
Edit Copy Delete	Jane Doe	456 Elm Street	Dhaka
Edit Copy Delete	John Doe	123 Main Street	New York
Edit Copy Delete	Michael Jones	456 Main Street	London
Edit Copy Delete	Susan Smith	26 AVNEW	San Francisco

Schema 3: Consider the following schemas for company database relations where the primary keys are underlined

employee (*employee_name, street, city*)

works(*employee_name, company_name, salary*)

company(*company_name, city*)

managers(*employee_name, manager_name*)

Problems:

i) Find tha total salary of each company

ii) Find all employees in the database who do not work for ACI Ltd

iii) Insert record into the employee table with proper values

Create a database using sql query.

SQL: CREATE DATABASE company;

Create a table and insert value using sql query.

SQL:

```
CREATE TABLE employee (
        employee_name VARCHAR(255) NOT NULL,
        street VARCHAR(255),
        city VARCHAR(255),
        PRIMARY KEY (employee_name)
);

INSERT INTO employee (employee_name, street, city)
VALUES
        ('John Smith', '123 Main Street', 'New York'),
        ('Jane Doe', '456 Elm Street', 'Los Angeles'),
        ('Bill Jones', '789 Oak Street', 'Chicago');
```

	employee_name	street	city
Edit Copy Delete	Bill Jones	789 Oak Street	Chicago
Edit Copy Delete	Jane Doe	456 Elm Street	Los Angeles
Edit Copy Delete	John Smith	123 Main Street	New York

Create a table and insert value using sql query.

SQL:

```
CREATE TABLE company (
        company_name VARCHAR(255) NOT NULL,
        city VARCHAR(255),
        PRIMARY KEY (company_name)
);

INSERT INTO company (company_name, city)
VALUES
        ('Acme Corporation', 'New York'),
        ('Widgets Inc.', 'Los Angeles'),
        ('The Boring Company', 'Chicago');
```

	company_name	city
Edit Copy Delete	Acme Corporation	New York
Edit Copy Delete	The Boring Company	Chicago
Edit Copy Delete	Widgets Inc.	Los Angeles

Create a table and insert value using sql query.

SQL:

```
CREATE TABLE works (
        employee_name VARCHAR(255) NOT NULL,
        company_name VARCHAR(255) NOT NULL,
        salary INT(11),
        FOREIGN KEY (employee_name) REFERENCES employee
        (employee_name)
                on update cascade on delete cascade,
        FOREIGN KEY (company_name) REFERENCES company
        (company_name)
                on update cascade on delete cascade
);
```

```
INSERT INTO works (employee_name, company_name, salary)
VALUES
    ('John Smith', 'Acme Corporation', 100000),
    ('Jane Doe', 'Widgets Inc.', 50000),
    ('Bill Jones', 'The Boring Company', 25000);
```

employee_name	company_name	salary
John Smith	Acme Corporation	100000
Jane Doe	Widgets Inc.	50000
Bill Jones	The Boring Company	25000

Create a table and insert value using sql query.

SQL:

```
CREATE TABLE manages (
    employee_name VARCHAR(255) NOT NULL,
    manager_name VARCHAR(255) NOT NULL,
    FOREIGN KEY (employee_name) REFERENCES employee
    (employee_name)
);

INSERT INTO manages (employee_name, manager_name) VALUES
('John Smith', 'Jane Doe'),
('Jane Doe', 'Bill Jones');
```

employee_name	manager_name
John Smith	Jane Doe
Jane Doe	Bill Jones

i) Find tha total salary of each company

```
SELECT company_name, SUM(salary) AS total_salary
    FROM works GROUP BY company_name
```

company_name	total_salary
Acme Corporation	100000
The Boring Company	25000
Widgets Inc.	50000

i) Find all employees in the database who do not work for ACI Ltd

```
SELECT employee_name FROM employee
    WHERE employee_name NOT IN (
        SELECT employee_name FROM works WHERE
        company_name="ACI Ltd");
```

←T→	employee_name
Edit Copy Delete	Bill Jones
Edit Copy Delete	Jane Doe
Edit Copy Delete	John Smith
Edit Copy Delete	Jones basa

iii) Insert record into the employee table with proper values

INSERT INTO employee (employee_name, street, city)

VALUES ('Bill Jones', 'Widgets Inc.', 'Los Angeles');

1 row inserted. (Query took 0.0412 seconds.)

```
INSERT INTO employee (employee_name, street, city) VALUES ('Jones basa',
'Widgets Inc.', 'Los Angeles')
```

[Edit inline] [Edit] [Create PHP code]