Consider a banking database and perform actions from the relation below and display the output along with the SQL query.

```
Customer (customer_id, customer_name, address, phone, email)
Borrows (borrow_id, customer_id, loan_number)
Loan (loan_number, loan_type, amount)
```

- a. Display the names of all customers who live in "Lalitpur" in ascending order of name.
- b. Count the total number of customers having loan at the bank.
- c. Find the name of those customers who have loan amount greater than or equal to 500000.
- d. Find the average loan amount of each type.

### **Creation of Tables:**

# SQL Query:

```
Create table customer (
    customer_id INT PRIMARY KEY,
    customer_name VARCHAR(40),
    address VARCHAR(40),
    phone VARCHAR(40),
    email VARCHAR(100) UNIQUE
);
Create table loan (
    loan_number INT PRIMARY KEY,
    loan_type VARCHAR(50),
    amount INT
);
Create table borrows (
    borrow id INT PRIMARY KEY,
```

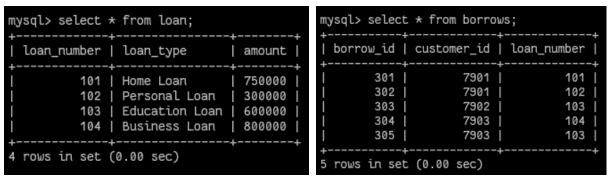
```
customer id INT,
      loan number INT,
      FOREIGN KEY (customer id) REFERENCES customer (customer id)
      FOREIGN KEY (loan number) REFERENCES loan (loan number)
);
Insertion of Data:
SQL Query:
Insert into customer (customer id, customer name, address, phone, email)
Values
      (7901, "Hari Bahadur", "Lalitpur", "9876543210", "hari123@gmail.com"),
      (7902, "Madan Shrestha", "Kathmandu", "9812345678", "sthamadan@gmail.com"),
      (7903, "Rajesh Hamal", "Lalitpur", "9845612370", "rajesh123@gmail.com");
Insert into loan (loan number, loan type, amount)
Values
      (101, "Home Loan", 750000),
      (102, "Personal Loan", 300000),
      (103, "Education Loan", 600000),
      (104, "Business Loan", 800000);
Insert into borrows (borrow id, customer id, loan number)
Values
      (301, 7901, 101),
      (302, 7901, 102),
      (303, 7902, 103);
      (304, 7903, 104),
```

(305, 7903, 103);

## Table after Insertion of Data:

customer_id   customer_name	address	phone	
+	Kathmandu	9812345678	sthamadan@gmail.com

### Customer Table



Loan Table Borrow Table

a. Display the names of all customers who live in "Lalitpur" in ascending order of name.

### SQL Query:

Select customer name from customer where address = "Lalitpur"

Order by customer name asc;

#### Table:

Here, the table shows the name of the customer (customer\_name) by ascending order who are residing in "Lalitpur".

```
mysql> select customer_name from customer where address = "Lalitpur"
    → order by customer_name asc;
+-----+
| customer_name |
+-----+
| Hari Bahadur |
| Rajesh Hamal |
+-----+
2 rows in set (0.00 sec)
```

## b. Count the total number of customers having loan at the bank.

# SQL Query:

Select count (distinct customer\_id) as Total\_Customers\_with\_Loan From borrows;

#### Table:

Here, the table shows the number of customers who have taken at least one loan. It ensures that each customer is counted only once, even if they have multiple loans.

c. Find the name of those customers who have loan amount greater than or equal to 500000.

## SQL Query:

Select distinct c.customer name

From customer as c

Join borrows as b on c.customer id = b.customer id

Join loan as I on I.loan number = b.loan number

Where 1.amount  $\geq$  500000;

#### Table:

Here, the table shows the names of customers who have at least one loan with an amount of 5,00,000 or more.

# d. Find the average loan amount of each type

# SQL Query:

Select loan type, avg(amount) as Average Loan

From loan

Group by loan type;

#### Table:

Here, the table shows the average loan amount for each type of loan.