EAST WEST UNIVERSITY

Department of Computer Science and Engineering
Course Title: Database Systems
Course code: CSE302

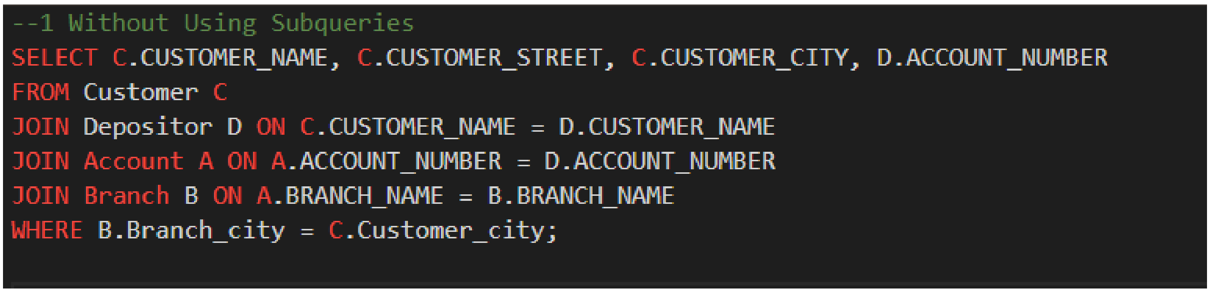
Lab no: 05

Submitted by:
Student name: Suddip Paul Arnab
ID: 2022-1-60-356

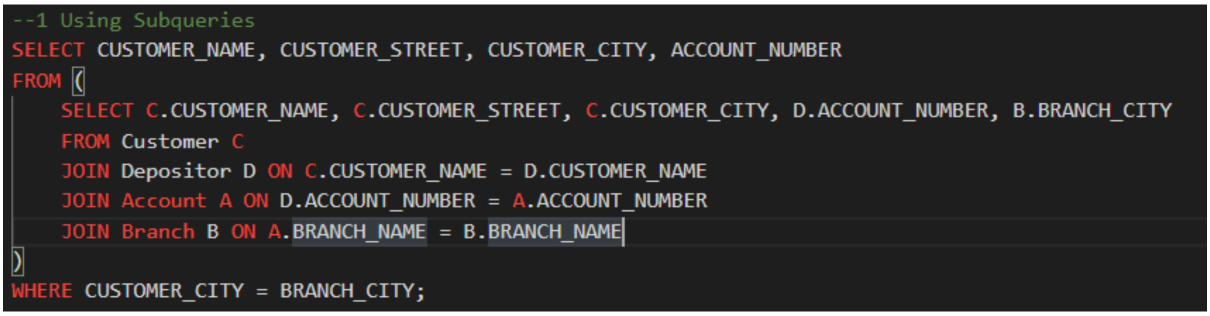
Submitted To:
Dr. Mohammad Rezwanul Huq
Associate Professor
Department of CSE

Date of Submission: 13/03/2024

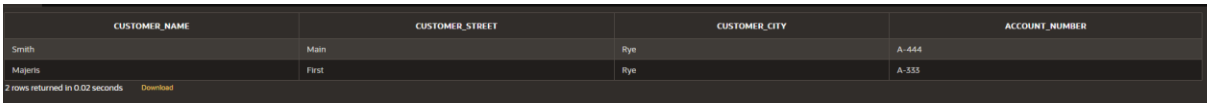
1. Find all customer related information who have an account in a branch, located in the same city as they live. (write this query without using subqueries and then using a subquery)



```
--1 Without Using Subqueries
SELECT C.CUSTOMER_NAME, C.CUSTOMER_STREET, C.CUSTOMER_CITY, D.ACCOUNT_NUMBER
FROM Customer C
JOIN Depositor D ON C.CUSTOMER_NAME = D.CUSTOMER_NAME
JOIN Account A ON A.ACCOUNT_NUMBER = D.ACCOUNT_NUMBER
JOIN Branch B ON A.BRANCH_NAME = B.BRANCH_NAME
WHERE B.Branch_city = C.Customer_city;
```



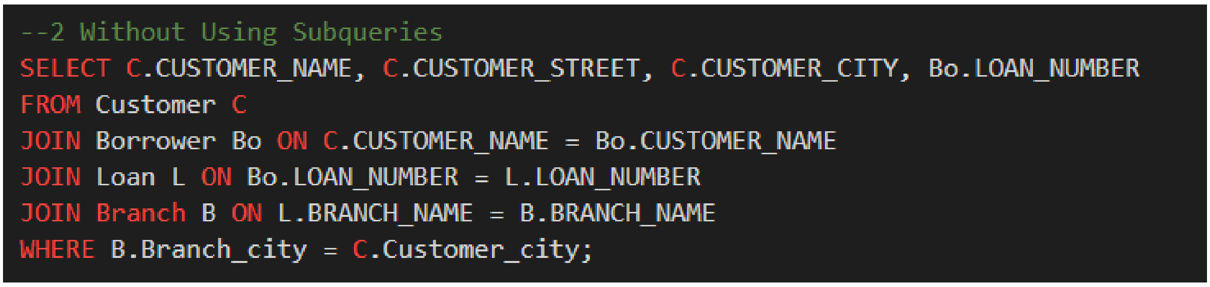
```
--1 Using Subqueries
SELECT CUSTOMER_NAME, CUSTOMER_STREET, CUSTOMER_CITY, ACCOUNT_NUMBER
FROM (
    SELECT C.CUSTOMER_NAME, C.CUSTOMER_STREET, C.CUSTOMER_CITY, D.ACCOUNT_NUMBER, B.BRANCH_CITY
    FROM Customer C
    JOIN Depositor D ON C.CUSTOMER_NAME = D.CUSTOMER_NAME
    JOIN Account A ON D.ACCOUNT_NUMBER = A.ACCOUNT_NUMBER
    JOIN Branch B ON A.BRANCH_NAME = B.BRANCH_NAME
)
WHERE CUSTOMER_CITY = BRANCH_CITY;
```



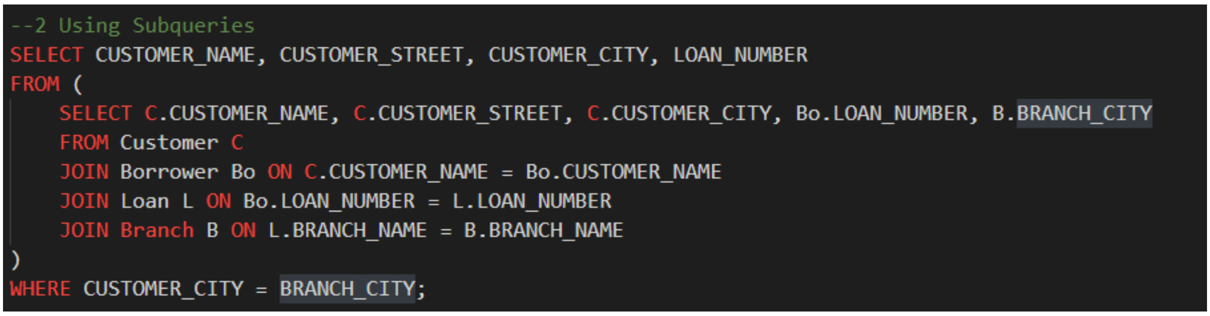
CUSTOMER_NAME	CUSTOMER_STREET	CUSTOMER_CITY	ACCOUNT_NUMBER
Smith	Main	Rye	A-444
Majeris	First	Rye	A-333

2 rows returned in 0.02 seconds Download

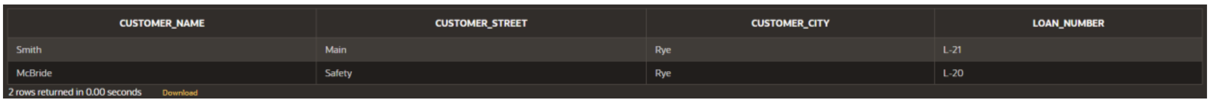
2. Find all customer related information who have a loan in a branch, located in the same city as they live. (write this query without using subqueries and then using a subquery)



```
--2 Without Using Subqueries
SELECT C.CUSTOMER_NAME, C.CUSTOMER_STREET, C.CUSTOMER_CITY, Bo.LOAN_NUMBER
FROM Customer C
JOIN Borrower Bo ON C.CUSTOMER_NAME = Bo.CUSTOMER_NAME
JOIN Loan L ON Bo.LOAN_NUMBER = L.LOAN_NUMBER
JOIN Branch B ON L.BRANCH_NAME = B.BRANCH_NAME
WHERE B.Branch_city = C.Customer_city;
```



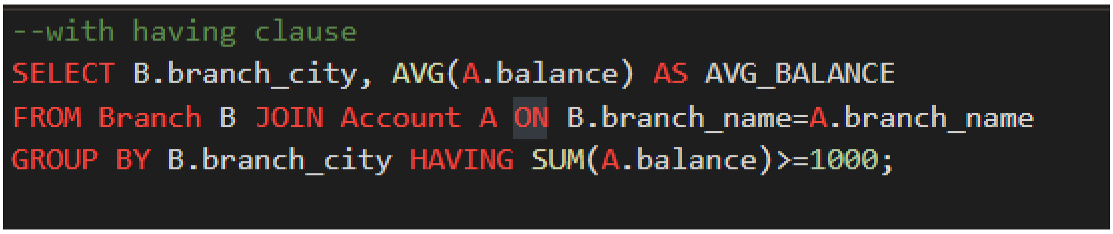
```
--2 Using Subqueries
SELECT CUSTOMER_NAME, CUSTOMER_STREET, CUSTOMER_CITY, LOAN_NUMBER
FROM (
    SELECT C.CUSTOMER_NAME, C.CUSTOMER_STREET, C.CUSTOMER_CITY, Bo.LOAN_NUMBER, B.BRANCH_CITY
    FROM Customer C
    JOIN Borrower Bo ON C.CUSTOMER_NAME = Bo.CUSTOMER_NAME
    JOIN Loan L ON Bo.LOAN_NUMBER = L.LOAN_NUMBER
    JOIN Branch B ON L.BRANCH_NAME = B.BRANCH_NAME
)
WHERE CUSTOMER_CITY = BRANCH_CITY;
```



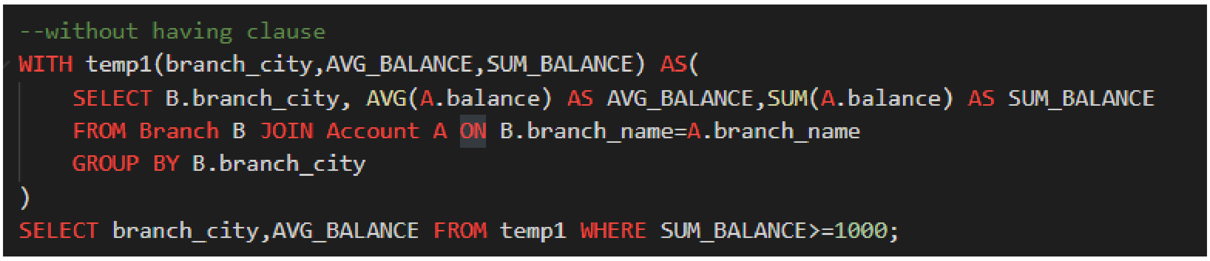
CUSTOMER_NAME	CUSTOMER_STREET	CUSTOMER_CITY	LOAN_NUMBER
Smith	Main	Rye	L-21
McBride	Safety	Rye	L-20

2 rows returned in 0.00 seconds Download

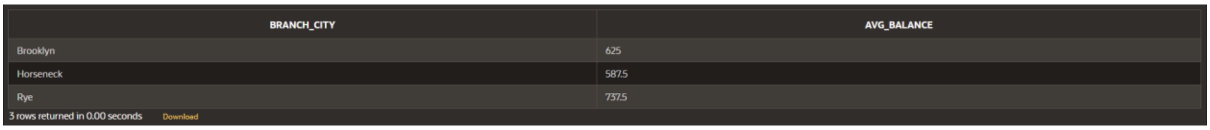
3. For each branch city, find the average balance of all the accounts opened in a branch located in that branch city. Do not include any branch city in the result where the total balance of all accounts opened in a branch located in that city is less than 1000. (Write this query with and without using ‘having’ clause)



```
--with having clause
SELECT B.branch_city, AVG(A.balance) AS AVG_BALANCE
FROM Branch B JOIN Account A ON B.branch_name=A.branch_name
GROUP BY B.branch_city HAVING SUM(A.balance)>=1000;
```



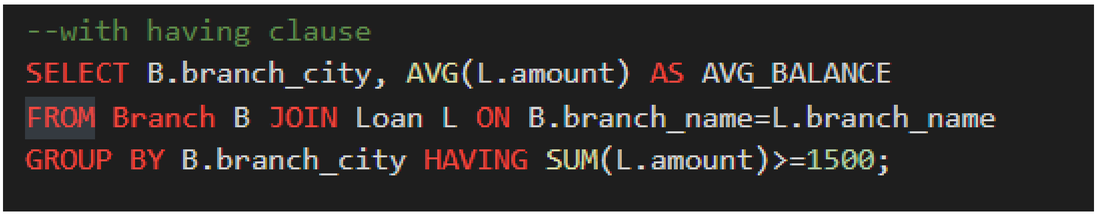
```
--without having clause
WITH temp1(branch_city,AVG_BALANCE,SUM_BALANCE) AS(
    SELECT B.branch_city, AVG(A.balance) AS AVG_BALANCE,SUM(A.balance) AS SUM_BALANCE
    FROM Branch B JOIN Account A ON B.branch_name=A.branch_name
    GROUP BY B.branch_city
)
SELECT branch_city,AVG_BALANCE FROM temp1 WHERE SUM_BALANCE>=1000;
```



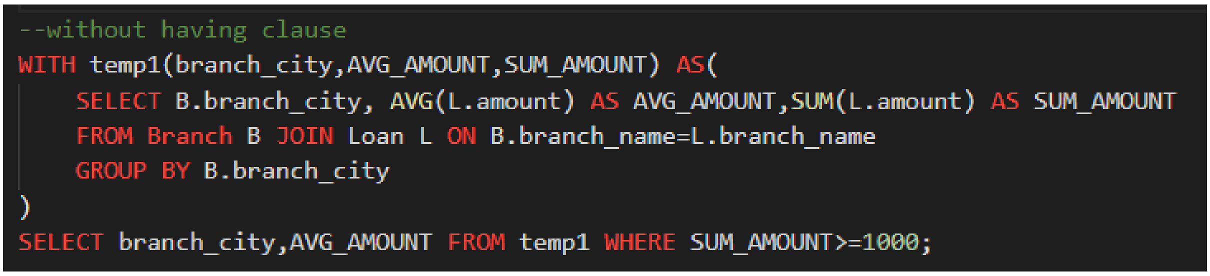
BRANCH_CITY	AVG_BALANCE
Brooklyn	625
Horseneck	587.5
Rye	737.5

3 rows returned in 0.00 seconds Download

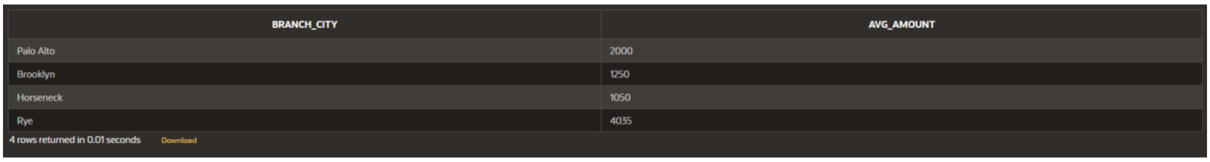
4. For each branch city, find the average amount of all the loans opened in a branch located in that branch city. Do not include any branch city in the result where the average amount of all loans opened in a branch located in that city is less than 1500. (write this query with and without using ‘having’ clause)



```
--with having clause
SELECT B.branch_city, AVG(L.amount) AS AVG_BALANCE
FROM Branch B JOIN Loan L ON B.branch_name=L.branch_name
GROUP BY B.branch_city HAVING SUM(L.amount)>=1500;
```



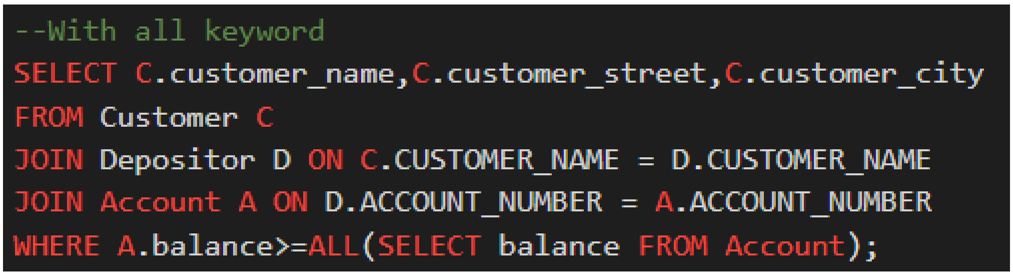
```
--without having clause
WITH temp1(branch_city,AVG_AMOUNT,SUM_AMOUNT) AS(
    SELECT B.branch_city, AVG(L.amount) AS AVG_AMOUNT,SUM(L.amount) AS SUM_AMOUNT
    FROM Branch B JOIN Loan L ON B.branch_name=L.branch_name
    GROUP BY B.branch_city
)
SELECT branch_city,AVG_AMOUNT FROM temp1 WHERE SUM_AMOUNT>=1000;
```



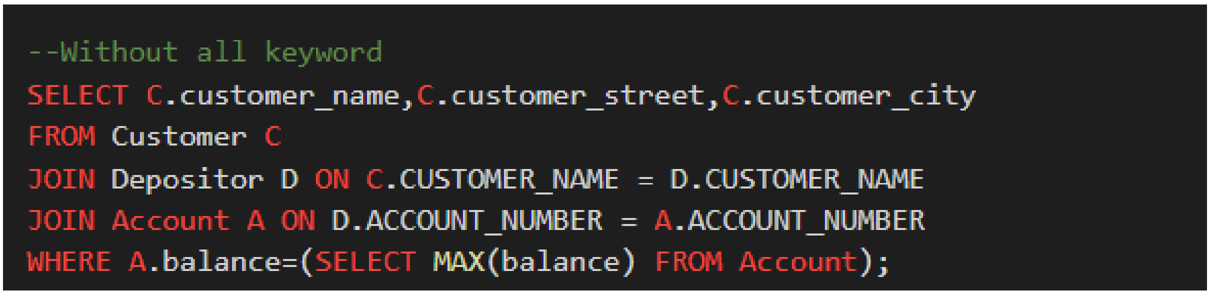
BRANCH_CITY	AVG_AMOUNT
Palo Alto	2000
Brooklyn	1250
Horseneck	1050
Rye	4035

4 rows returned in 0.01 seconds Download

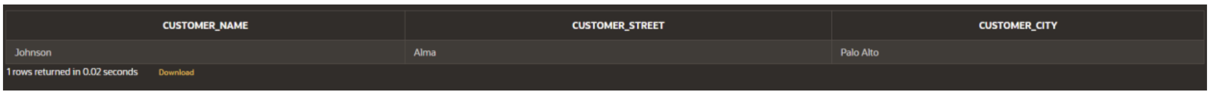
5. Find the customer name, customer street, customer city of the account which has the highest balance among all the accounts. (Write this query with and without using all keyword)



```sql
--With all keyword
SELECT C.customer_name,C.customer_street,C.customer_city
FROM Customer C
JOIN Depositor D ON C.CUSTOMER_NAME = D.CUSTOMER_NAME
JOIN Account A ON D.ACCOUNT_NUMBER = A.ACCOUNT_NUMBER
WHERE A.balance>=ALL(SELECT balance FROM Account);
```



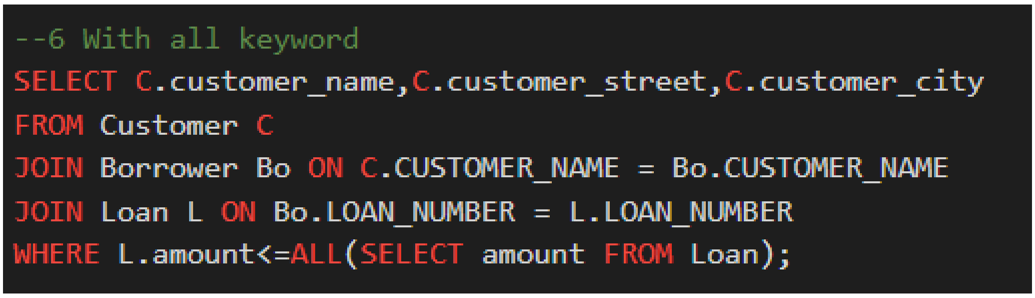
```sql
--Without all keyword
SELECT C.customer_name,C.customer_street,C.customer_city
FROM Customer C
JOIN Depositor D ON C.CUSTOMER_NAME = D.CUSTOMER_NAME
JOIN Account A ON D.ACCOUNT_NUMBER = A.ACCOUNT_NUMBER
WHERE A.balance=(SELECT MAX(balance) FROM Account);
```



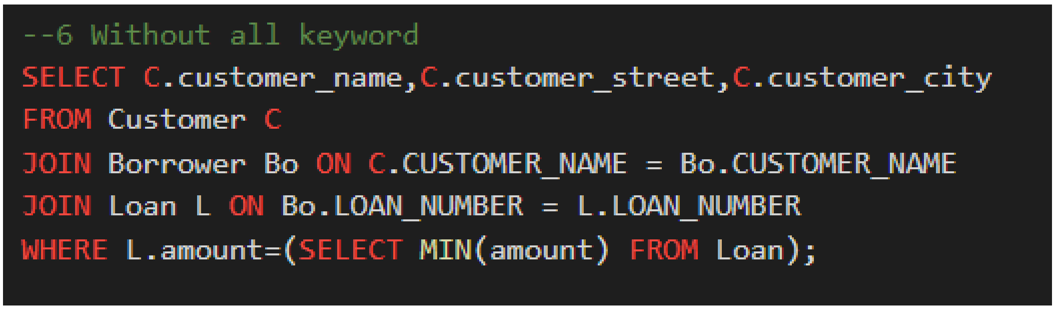
CUSTOMER_NAME	CUSTOMER_STREET	CUSTOMER_CITY
Johnson	Alma	Palo Alto

1 rows returned in 0.02 seconds Download

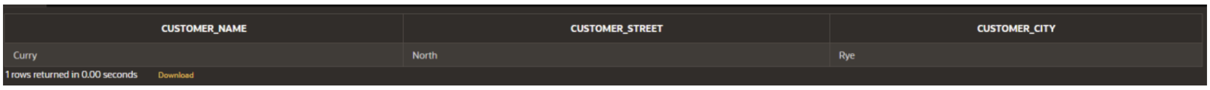
6. Find the customer name, customer street, customer city of the loan which has the lowest amount among all the loans. (write this query with and without using all keyword)



```sql
--6 With all keyword
SELECT C.customer_name,C.customer_street,C.customer_city
FROM Customer C
JOIN Borrower Bo ON C.CUSTOMER_NAME = Bo.CUSTOMER_NAME
JOIN Loan L ON Bo.LOAN_NUMBER = L.LOAN_NUMBER
WHERE L.amount<=ALL(SELECT amount FROM Loan);
```



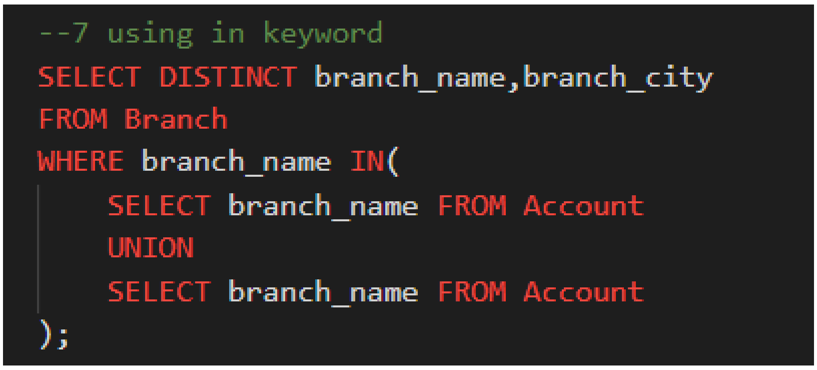
```sql
--6 Without all keyword
SELECT C.customer_name,C.customer_street,C.customer_city
FROM Customer C
JOIN Borrower Bo ON C.CUSTOMER_NAME = Bo.CUSTOMER_NAME
JOIN Loan L ON Bo.LOAN_NUMBER = L.LOAN_NUMBER
WHERE L.amount=(SELECT MIN(amount) FROM Loan);
```



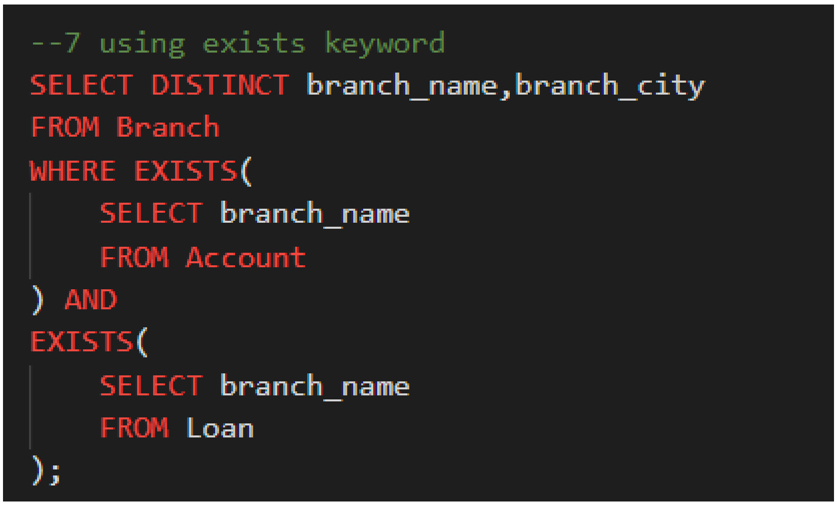
CUSTOMER_NAME	CUSTOMER_STREET	CUSTOMER_CITY
Curry	North	Rye

1 rows returned in 0.00 seconds Download

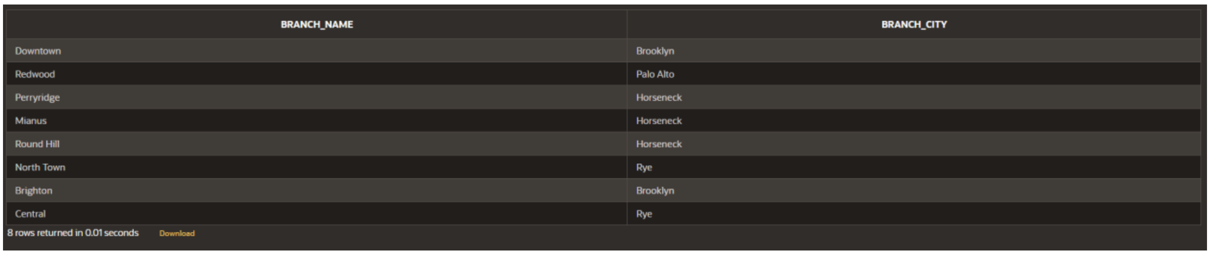
7. Find the distinct branches (name and city) that have opened both accounts and loans. (Write this query using in and exists keyword)



```
--7 using in keyword
SELECT DISTINCT branch_name,branch_city
FROM Branch
WHERE branch_name IN(
    SELECT branch_name FROM Account
    UNION
    SELECT branch_name FROM Account
);
```



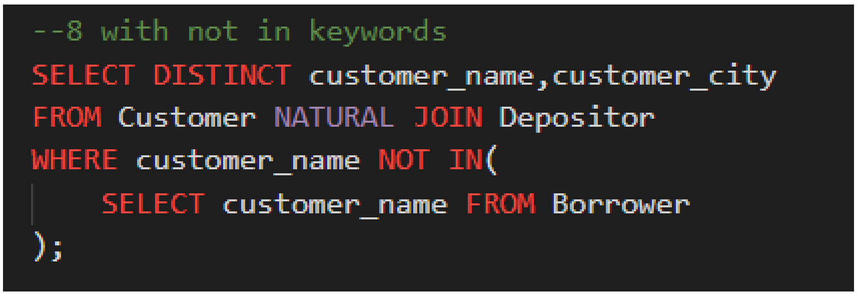
```
--7 using exists keyword
SELECT DISTINCT branch_name,branch_city
FROM Branch
WHERE EXISTS(
    SELECT branch_name
    FROM Account
) AND
EXISTS(
    SELECT branch_name
    FROM Loan
);
```



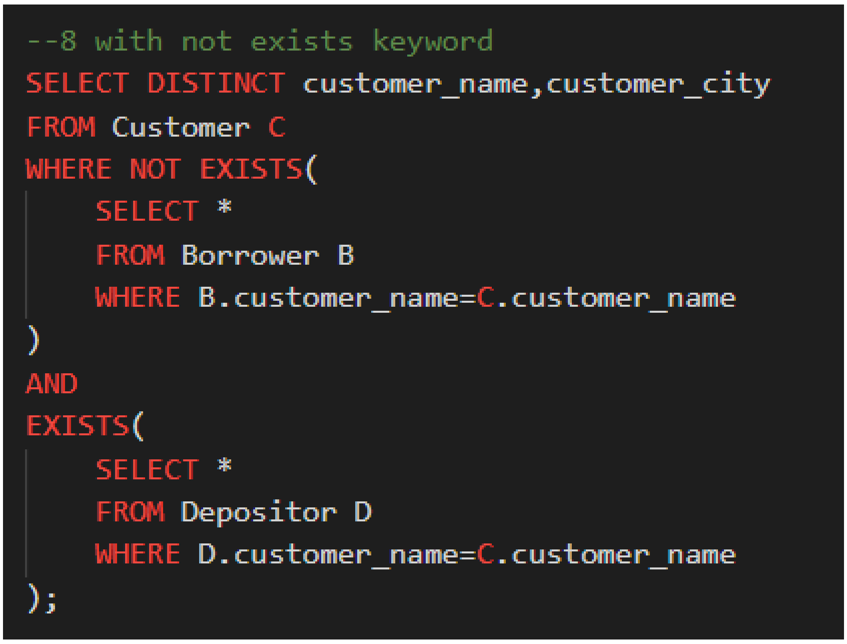
BRANCH_NAME	BRANCH_CITY
Downtown	Brooklyn
Redwood	Palo Alto
Perryridge	Horseneck
Mianus	Horseneck
Round Hill	Horseneck
North Town	Rye
Brighton	Brooklyn
Central	Rye

8 rows returned in 0.01 seconds Download

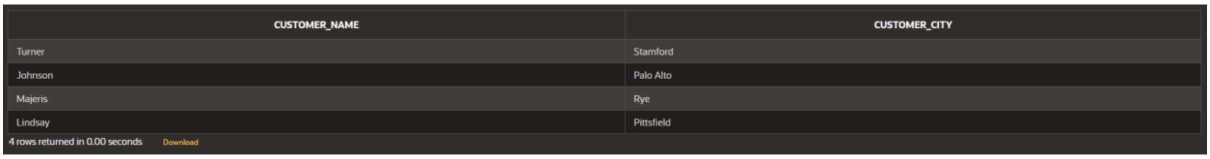
8. Find the distinct customers (name and city) who do not have loans but have accounts. (write this query using not in and not exists keyword)



```
--8 with not in keywords
SELECT DISTINCT customer_name,customer_city
FROM Customer NATURAL JOIN Depositor
WHERE customer_name NOT IN(
    SELECT customer_name FROM Borrower
);
```



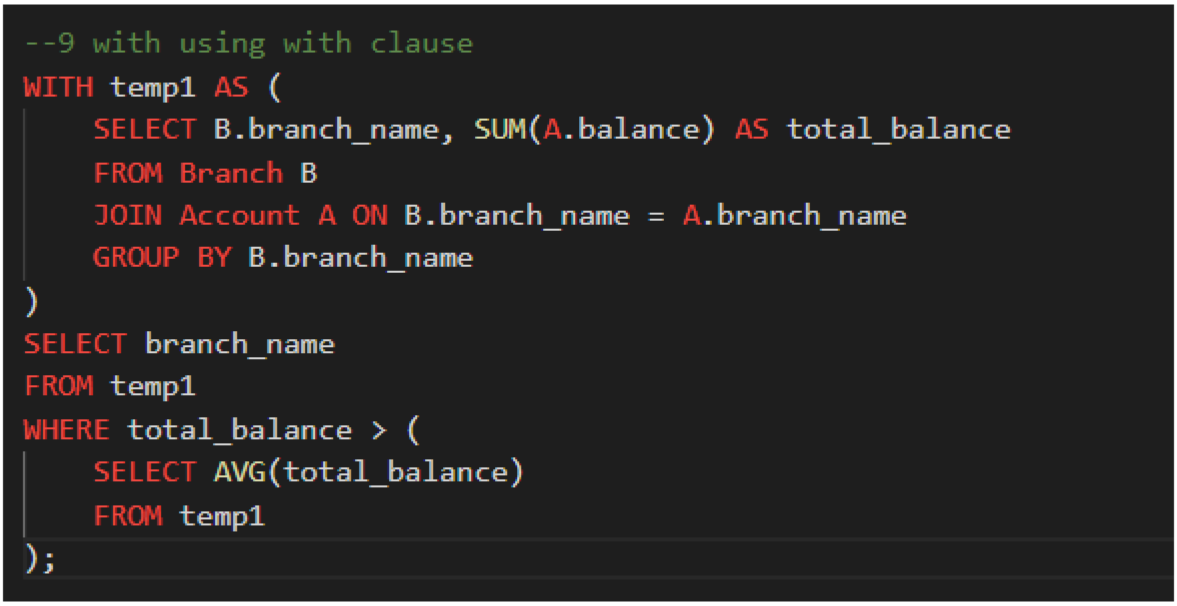
```sql
--8 with not exists keyword
SELECT DISTINCT customer_name,customer_city
FROM Customer C
WHERE NOT EXISTS(
    SELECT *
    FROM Borrower B
    WHERE B.customer_name=C.customer_name
)
AND
EXISTS(
    SELECT *
    FROM Depositor D
    WHERE D.customer_name=C.customer_name
);
```



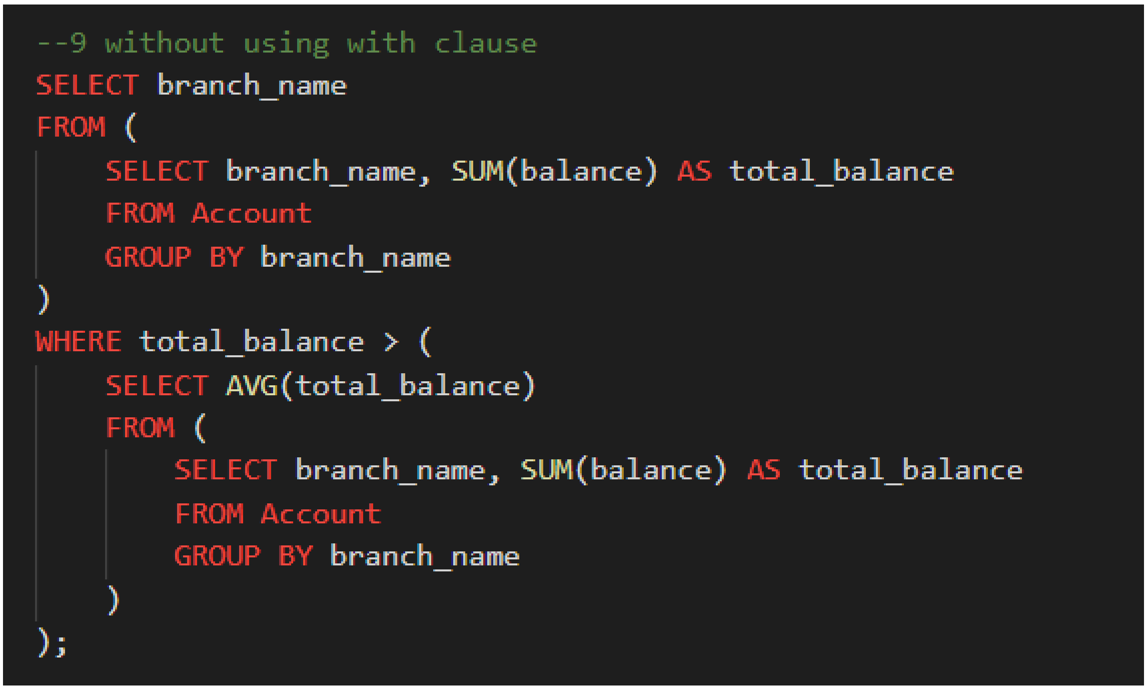
CUSTOMER_NAME	CUSTOMER_CITY
Turner	Stamford
Johnson	Palo Alto
Majeris	Rye
Lindsay	Pittsfield

4 rows returned in 0.00 seconds Download

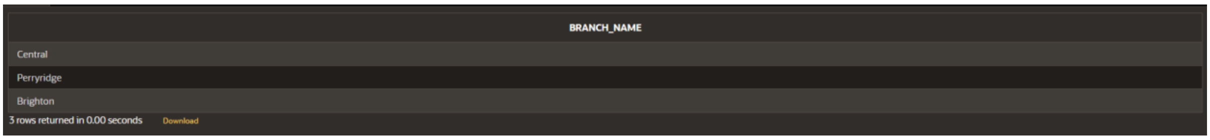
9. Find those branch names which have total account balance greater than the average of total balance among all the branches. (write this query with and without using with clause)



```sql
--9 with using with clause
WITH temp1 AS (
    SELECT B.branch_name, SUM(A.balance) AS total_balance
    FROM Branch B
    JOIN Account A ON B.branch_name = A.branch_name
    GROUP BY B.branch_name
)
SELECT branch_name
FROM temp1
WHERE total_balance > (
    SELECT AVG(total_balance)
    FROM temp1
);
```



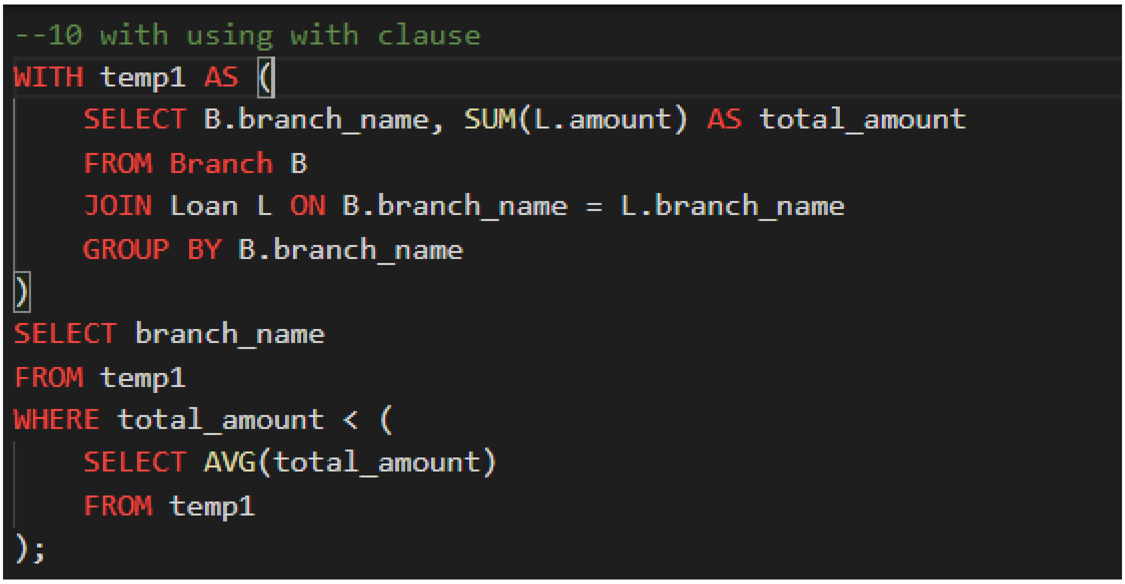
```sql
--9 without using with clause
SELECT branch_name
FROM (
    SELECT branch_name, SUM(balance) AS total_balance
    FROM Account
    GROUP BY branch_name
)
WHERE total_balance > (
    SELECT AVG(total_balance)
    FROM (
        SELECT branch_name, SUM(balance) AS total_balance
        FROM Account
        GROUP BY branch_name
    )
);
```



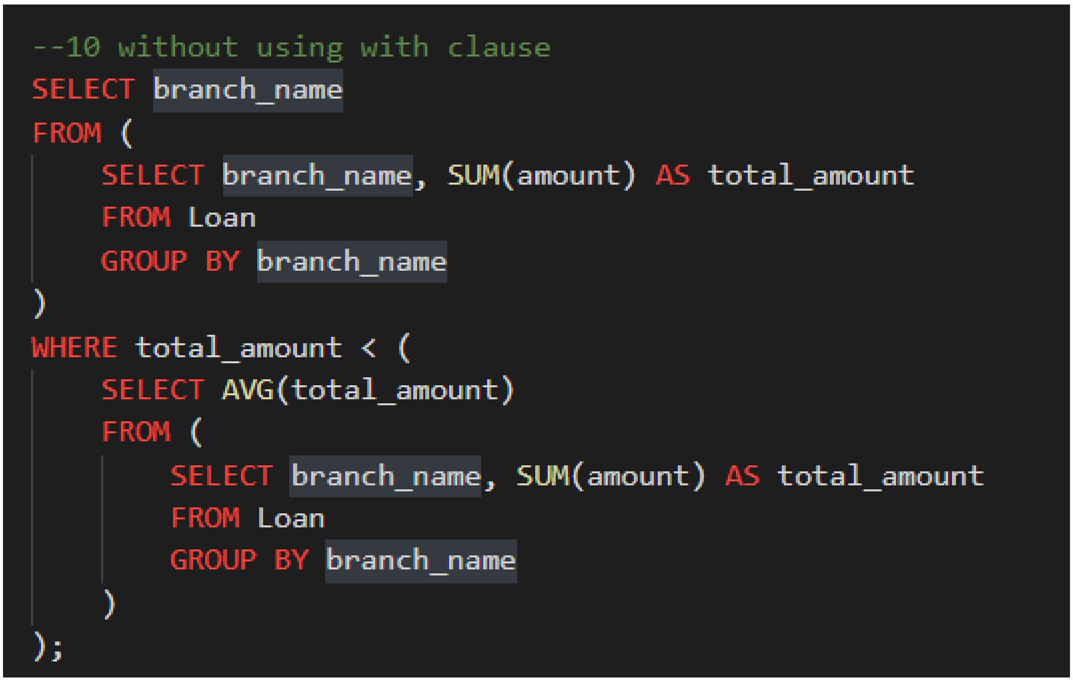
BRANCH_NAME
Central
Perryridge
Brighton

3 rows returned in 0.00 seconds

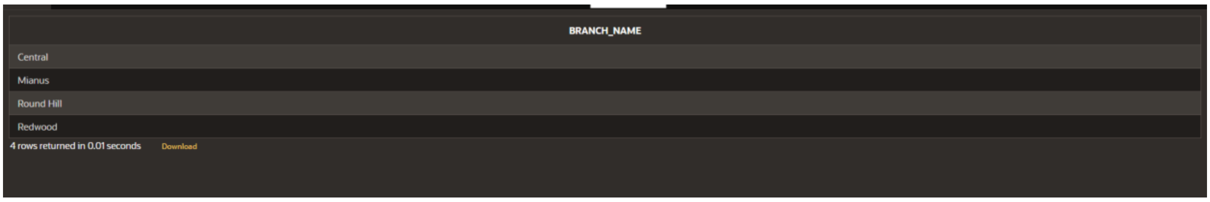
10. Find those branch names which have total loan amount less than the average of total loan amount among all the branches. (write this query with and without using with clause)



```sql
--10 with using with clause
WITH temp1 AS (
    SELECT B.branch_name, SUM(L.amount) AS total_amount
    FROM Branch B
    JOIN Loan L ON B.branch_name = L.branch_name
    GROUP BY B.branch_name
)
SELECT branch_name
FROM temp1
WHERE total_amount < (
    SELECT AVG(total_amount)
    FROM temp1
);
```



```sql
--10 without using with clause
SELECT branch_name
FROM (
    SELECT branch_name, SUM(amount) AS total_amount
    FROM Loan
    GROUP BY branch_name
)
WHERE total_amount < (
    SELECT AVG(total_amount)
    FROM (
        SELECT branch_name, SUM(amount) AS total_amount
        FROM Loan
        GROUP BY branch_name
    )
);
```



BRANCH_NAME
Central
Mianus
Round Hill
Redwood

4 rows returned in 0.01 seconds