

**LAB-05**

**Title: Introducing Nested Subqueries and Outer Joins in SQL**

**Name: Azizul Abedin Azmi**

**ID: 2022-1-60-130**

**Section: 02**

**Course Code: CSE302**

**Course Title: (Database Systems)**

**Date: 04/09/2024**

**Course Instructor:**

**Mahmuda Rawnak Jahan**

**Lecturer**

**Department of Computer Science and Engineering**

**Lab Tasks:**

**1. Find all customer-related information who have an account in a branch located in the same city as where they live.**

**Without Subqueries:**

SELECT C.customer\_name, C.customer\_street, C.customer\_city, D.account\_number

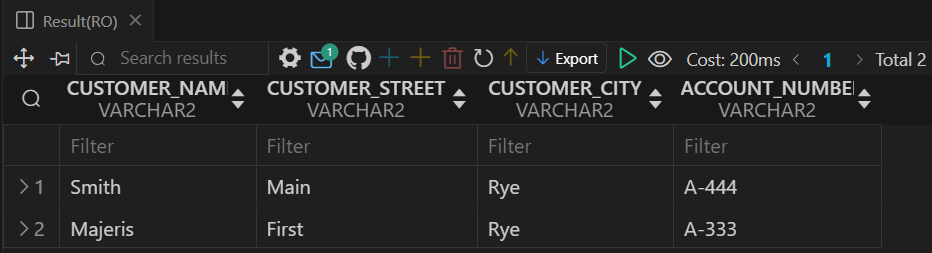
FROM Customer C, Depositor D, Account A, Branch B

WHERE C.customer\_name = D.customer\_name

AND A.account\_number = D.account\_number

AND A.branch\_name = B.branch\_name

AND B.Branch\_city = C.customer\_city;



**With Subquery:**

SELECT c.customer\_name, c.customer\_street, c.customer\_city

FROM Customer c

WHERE c.customer\_name IN (

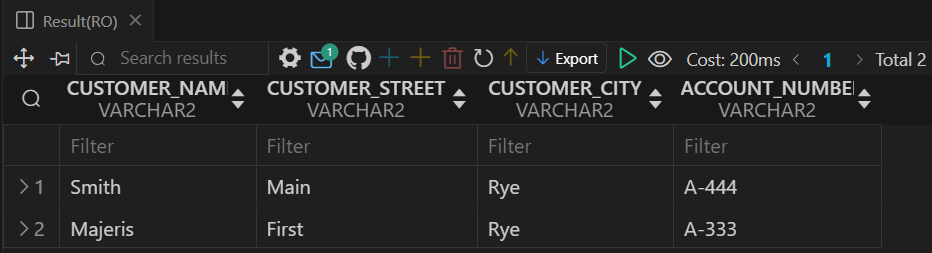
    SELECT d.customer\_name

    FROM Depositor d, Account a, Branch b

    WHERE d.account\_number = a.account\_number

    AND a.branch\_name = b.branch\_name

    AND c.customer\_city = b.branch\_city);



**2. Find all customer-related information who have a loan in a branch located in the same city as where they live.**

**Without Subqueries:**

SELECT DISTINCT c.customer\_name, c.customer\_street, c.customer\_city

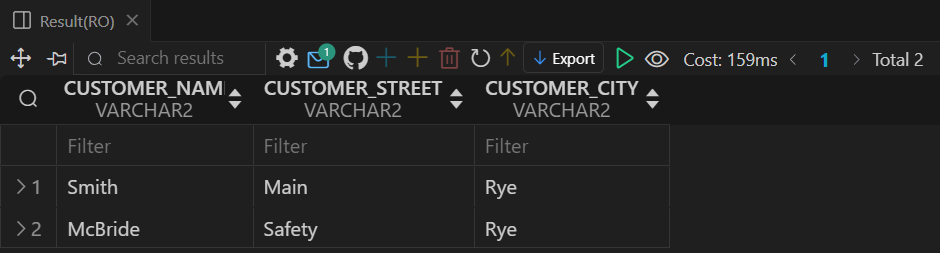
FROM Customer c, Loan l, Branch b, Borrower br

WHERE c.customer\_name = br.customer\_name

AND br.loan\_number = l.loan\_number

AND l.branch\_name = b.branch\_name

AND c.customer\_city = b.branch\_city;



**With Subquery:**

SELECT c.customer\_name, c.customer\_street, c.customer\_city

FROM Customer c

WHERE c.customer\_name IN (

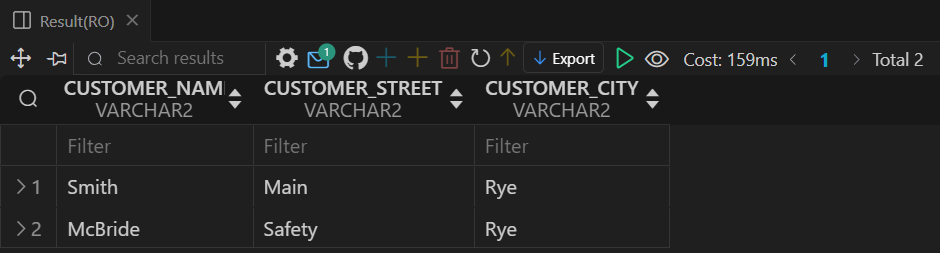
    SELECT br.customer\_name

    FROM Borrower br, Loan l, Branch b

    WHERE br.loan\_number = l.loan\_number

    AND l.branch\_name = b.branch\_name

    AND c.customer\_city = b.branch\_city);



**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.**

**Without HAVING clause:**

WITH temp1(Branch\_city,avg\_balance,sum\_balance) AS(

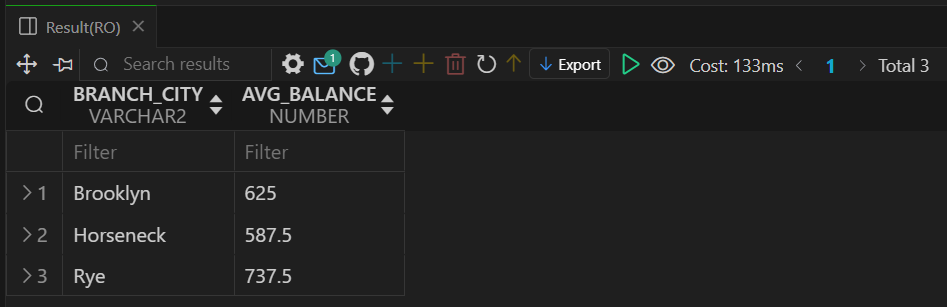
    SELECT B.Branch\_city, AVG(A.balance) AS avg\_balance,SUM(A.balance) AS sum\_balance

    FROM Branch B, Account A

    WHERE B.branch\_name=A.branch\_name

    GROUP BY B.Branch\_city )

SELECT Branch\_city,avg\_balance FROM temp1 WHERE sum\_balance>=1000;



**With HAVING clause:**

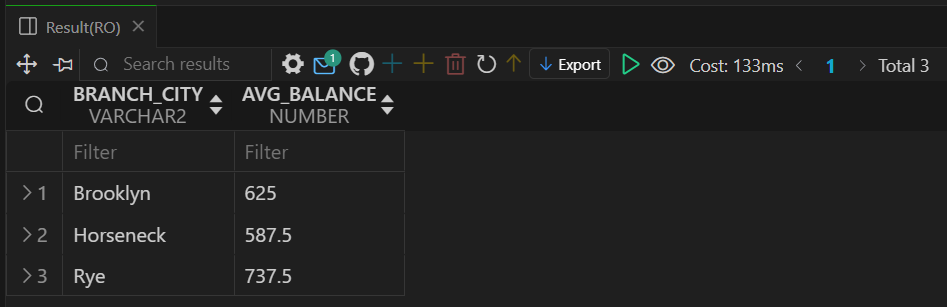
SELECT b.branch\_city, AVG(a.balance) AS avg\_balance

FROM Branch b, Account a

WHERE b.branch\_name = a.branch\_name

GROUP BY b.branch\_city

HAVING SUM(a.balance) >= 1000;



**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.**

**Without HAVING clause:**

WITH temp1(Branch\_city,AVG\_AMOUNT,sum\_amount) AS(

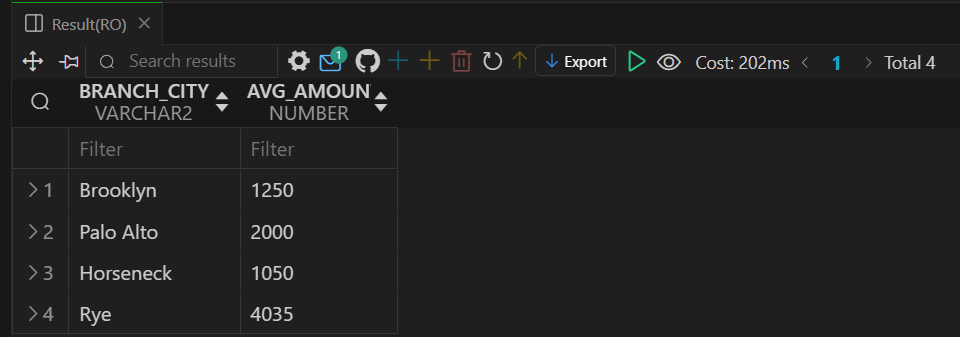
    SELECT B.Branch\_city, AVG(L.amount) AS AVG\_AMOUNT,SUM(L.amount) AS sum\_amount

    FROM Branch B, Loan L

    WHERE B.branch\_name=L.branch\_name

    GROUP BY B.Branch\_city )

SELECT Branch\_city,AVG\_AMOUNT FROM temp1 WHERE sum\_amount>=1000;



**With HAVING clause:**

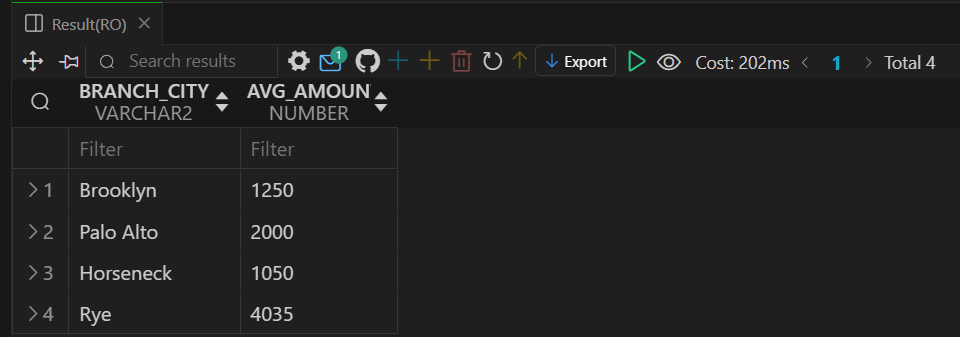
SELECT B.Branch\_city, AVG(L.amount) AS avg\_balance

FROM Branch B, Loan L

WHERE B.branch\_name = L.branch\_name

GROUP BY B.Branch\_city

HAVING SUM(L.amount)>=1500;



**5. Find the customer name, customer street, customer city of the account which has the highest balance among all the accounts.**

**Without ALL keyword:**

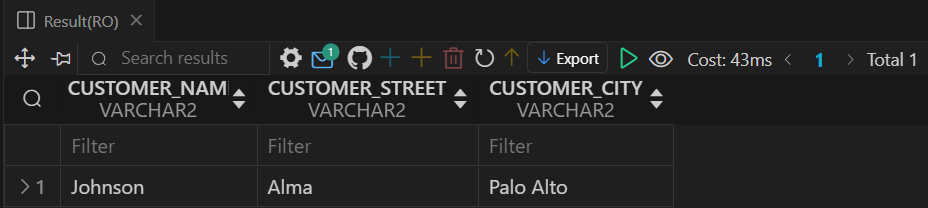
SELECT c.customer\_name, c.customer\_street, c.customer\_city

FROM Customer c, Depositor d, Account a

WHERE c.customer\_name = d.customer\_name

AND d.account\_number = a.account\_number

AND a.balance = (SELECT MAX(balance) FROM Account);



**With ALL keyword:**

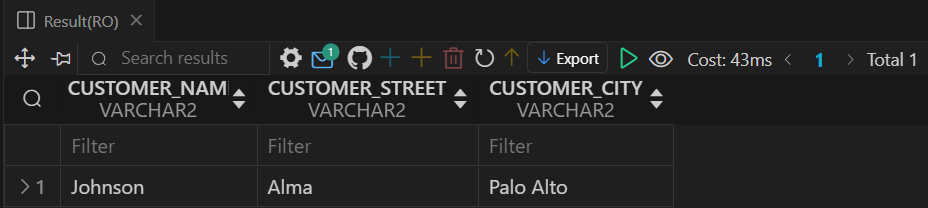
SELECT c.customer\_name, c.customer\_street, c.customer\_city

FROM Customer c, Depositor d, Account a

WHERE c.customer\_name = d.customer\_name

AND d.account\_number = a.account\_number

AND a.balance >= ALL (SELECT balance FROM Account);



**6. Find the customer name, customer street, customer city of the loan which has the lowest amount among all the loans.**

**Without ALL keyword:**

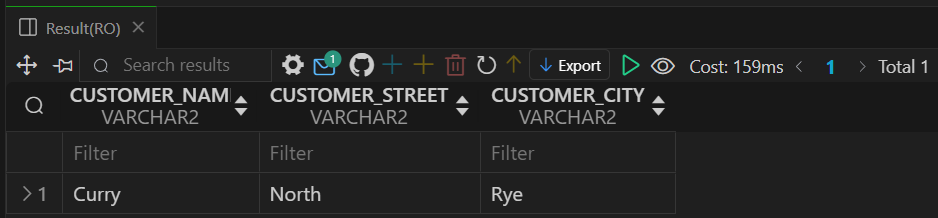
SELECT c.customer\_name, c.customer\_street, c.customer\_city

FROM Customer c, Borrower br, Loan l

WHERE c.customer\_name = br.customer\_name

AND br.loan\_number = l.loan\_number

AND l.amount = (SELECT MIN(amount) FROM Loan);



**With ALL keyword:**

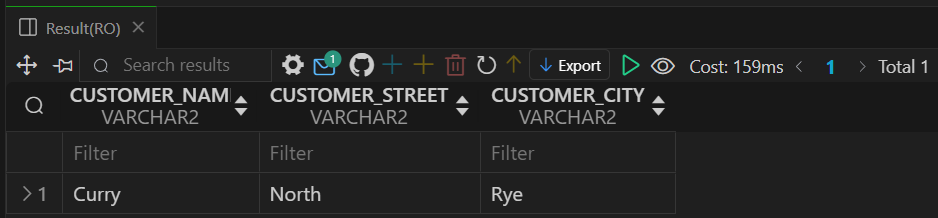
SELECT c.customer\_name, c.customer\_street, c.customer\_city

FROM Customer c, Borrower br, Loan l

WHERE c.customer\_name = br.customer\_name

AND br.loan\_number = l.loan\_number

AND l.amount <= ALL (SELECT amount FROM Loan);



**7. Find the distinct branches (name and city) that have opened both accounts and loans.**

**Using IN keyword:**

SELECT DISTINCT b.branch\_name, b.branch\_city

FROM Branch b

WHERE b.branch\_name IN (

    SELECT branch\_name

    FROM Account

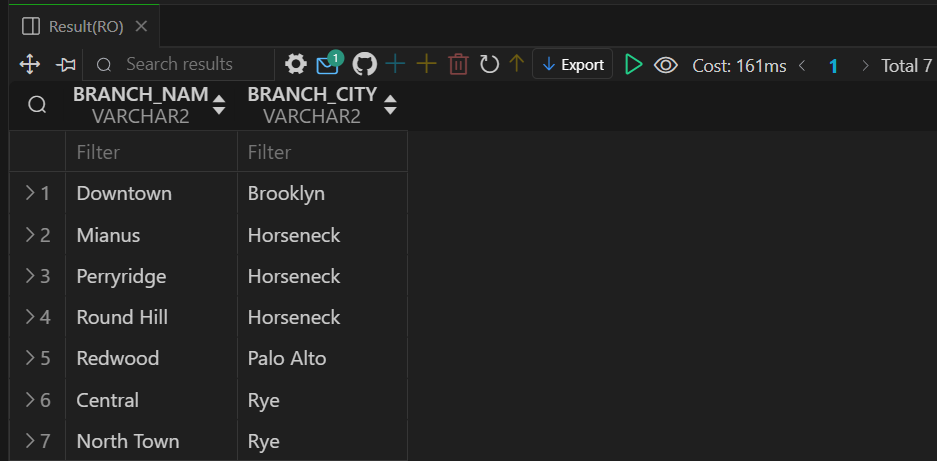
)

AND b.branch\_name IN (

    SELECT branch\_name

    FROM Loan

);



**Using EXISTS keyword:**

SELECT DISTINCT b.branch\_name, b.branch\_city

FROM Branch b

WHERE EXISTS (

    SELECT 1

    FROM Account a

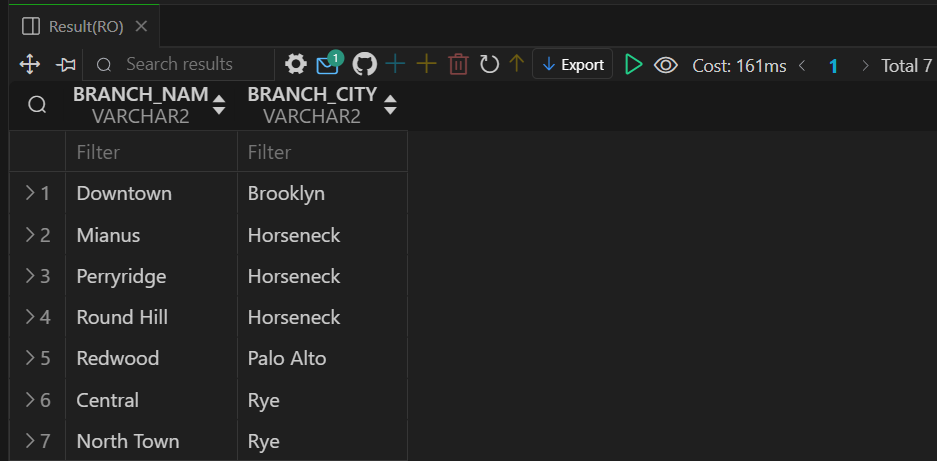
    WHERE a.branch\_name = b.branch\_name )

AND EXISTS (

    SELECT 1

    FROM Loan l

    WHERE l.branch\_name = b.branch\_name );



**8. Find the distinct customers (name and city) who do not have loans but have accounts.**

**Using NOT IN keyword:**

SELECT DISTINCT c.customer\_name, c.customer\_city

FROM Customer c

WHERE c.customer\_name IN (

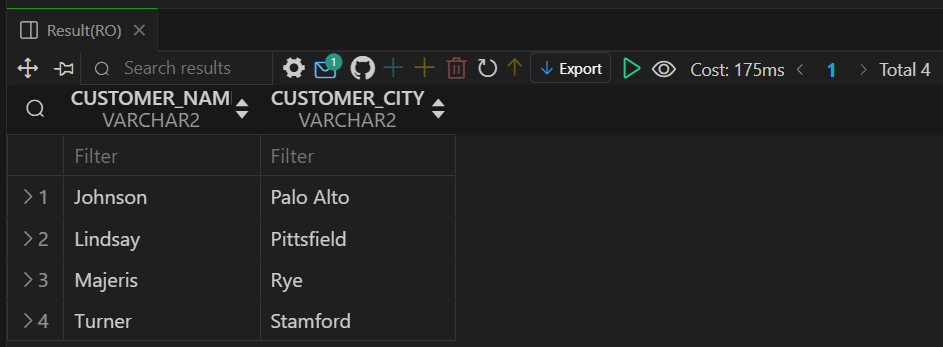
    SELECT d.customer\_name

    FROM Depositor d )

AND c.customer\_name NOT IN (

    SELECT br.customer\_name

    FROM Borrower br );



**Using NOT EXISTS keyword:**

SELECT DISTINCT c.customer\_name, c.customer\_city

FROM Customer c

WHERE EXISTS (

    SELECT 1

    FROM Depositor d

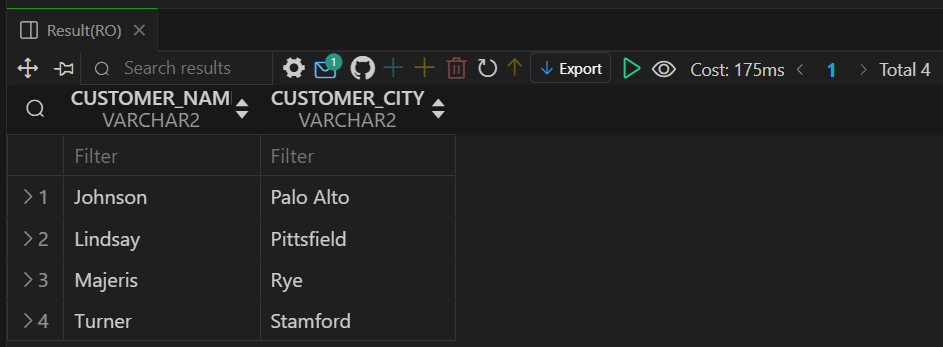
    WHERE d.customer\_name = c.customer\_name )

AND NOT EXISTS (

    SELECT 1

    FROM Borrower br

    WHERE br.customer\_name = c.customer\_name );



**9. Find those branch names which have a total account balance greater than the average of total balance among all the branches.**

**Without WITH clause:**

SELECT branch\_name

FROM (

    SELECT branch\_name, SUM(balance) AS total\_balance

    FROM Account

    GROUP BY branch\_name )

WHERE total\_balance > (

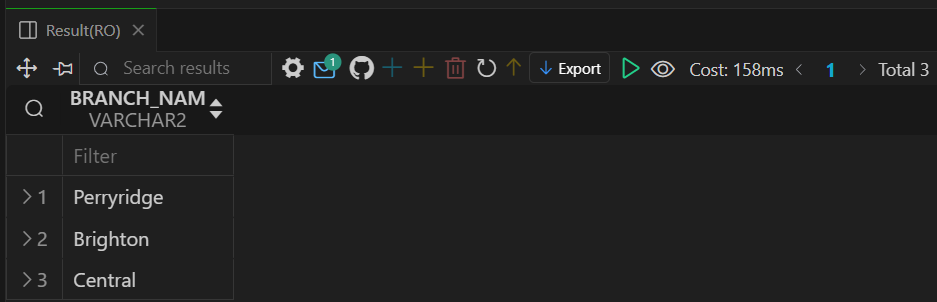
    SELECT AVG(total\_balance)

    FROM (

        SELECT branch\_name, SUM(balance) AS total\_balance

        FROM Account

        GROUP BY branch\_name ) );



**With WITH clause:**

WITH temp1 AS (

    SELECT B.branch\_name, SUM(A.balance) AS total\_balance

    FROM Branch B, Account A

    WHERE B.branch\_name = A.branch\_name

    GROUP BY B.branch\_name )

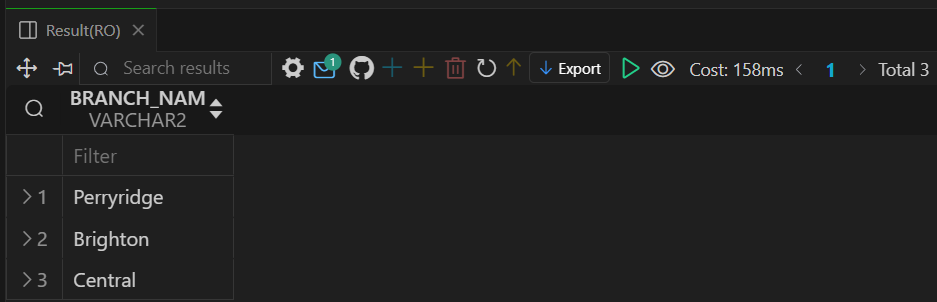
SELECT branch\_name

FROM temp1

WHERE total\_balance > (

    SELECT AVG(total\_balance)

    FROM temp1 );



**10. Find those branch names which have a total loan amount less than the average of the total loan amount among all the branches.**

**Without WITH clause:**

SELECT branch\_name

FROM (

    SELECT branch\_name, SUM(amount) AS total\_amount

    FROM Loan

    GROUP BY branch\_name )

WHERE total\_amount < (

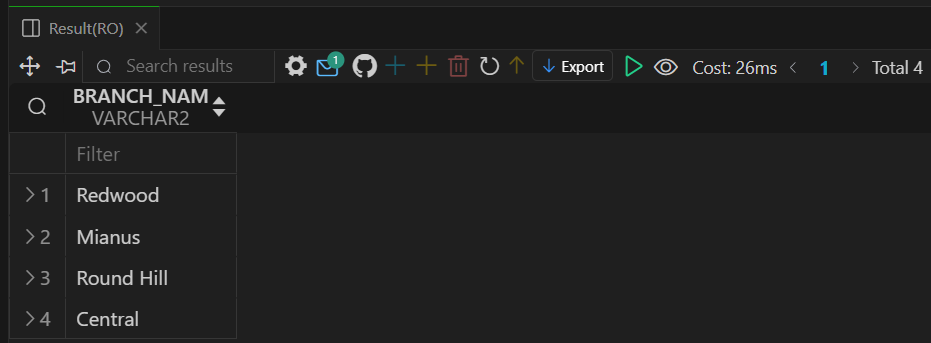
    SELECT AVG(total\_amount)

    FROM (

        SELECT branch\_name, SUM(amount) AS total\_amount

        FROM Loan

        GROUP BY branch\_name ) );



**With WITH clause:**

WITH temp1 AS (

    SELECT B.branch\_name, SUM(L.amount) AS total\_amount

    FROM Branch B, Loan L

    WHERE 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 );

