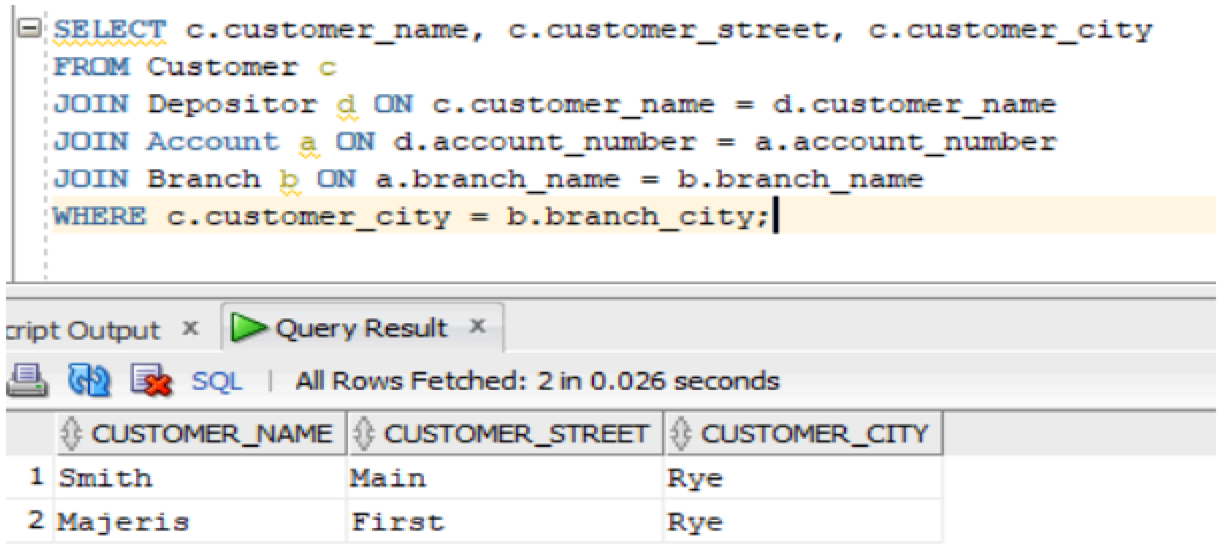
LAB-5
Assignment

Prepared by:

Md. Asif Hossain [2022-3-60-007]
Section: 07

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



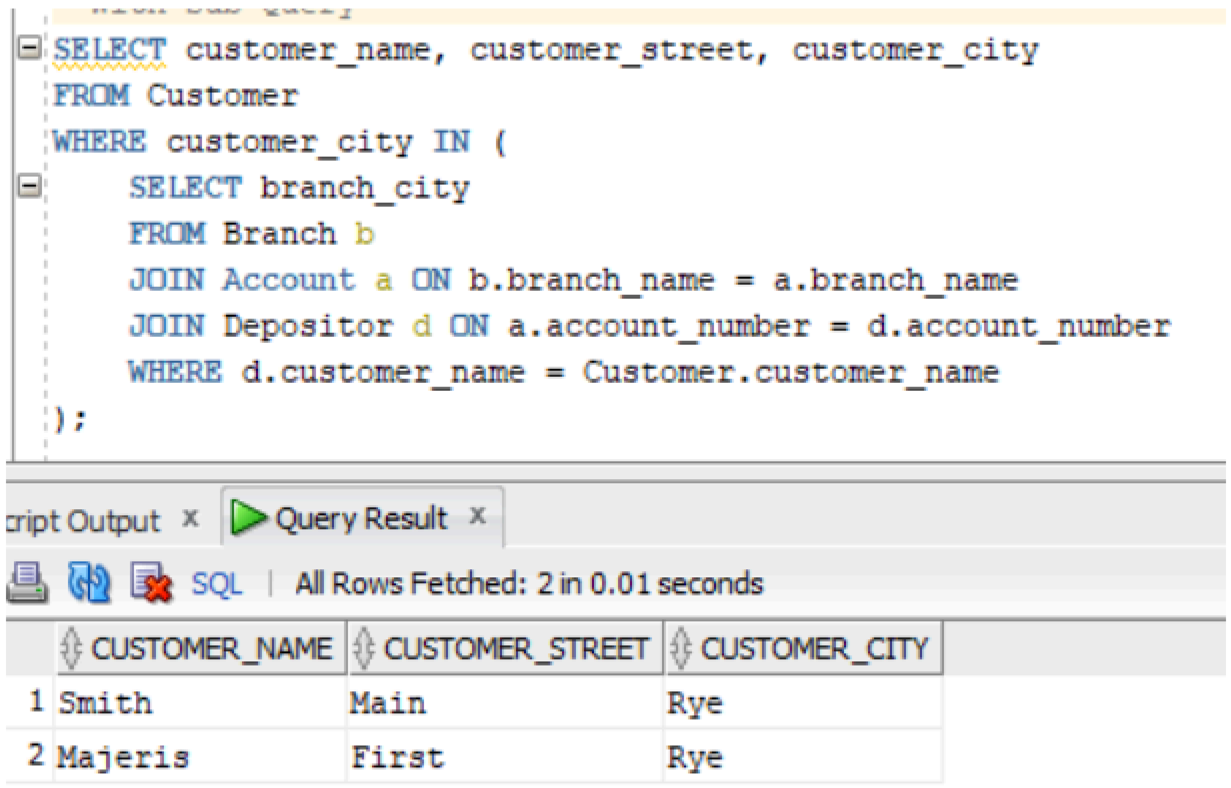
```sql
SELECT c.customer_name, c.customer_street, c.customer_city
FROM Customer c
JOIN Depositor d ON c.customer_name = d.customer_name
JOIN Account a ON d.account_number = a.account_number
JOIN Branch b ON a.branch_name = b.branch_name
WHERE c.customer_city = b.branch_city;
```

Script Output × | Query Result ×

SQL | All Rows Fetched: 2 in 0.026 seconds

	CUSTOMER_NAME	CUSTOMER_STREET	CUSTOMER_CITY
1	Smith	Main	Rye
2	Majeris	First	Rye

With using subqueries:



```sql
SELECT customer_name, customer_street, customer_city
FROM Customer
WHERE customer_city IN (
    SELECT branch_city
    FROM Branch b
    JOIN Account a ON b.branch_name = a.branch_name
    JOIN Depositor d ON a.account_number = d.account_number
    WHERE d.customer_name = Customer.customer_name
);
```

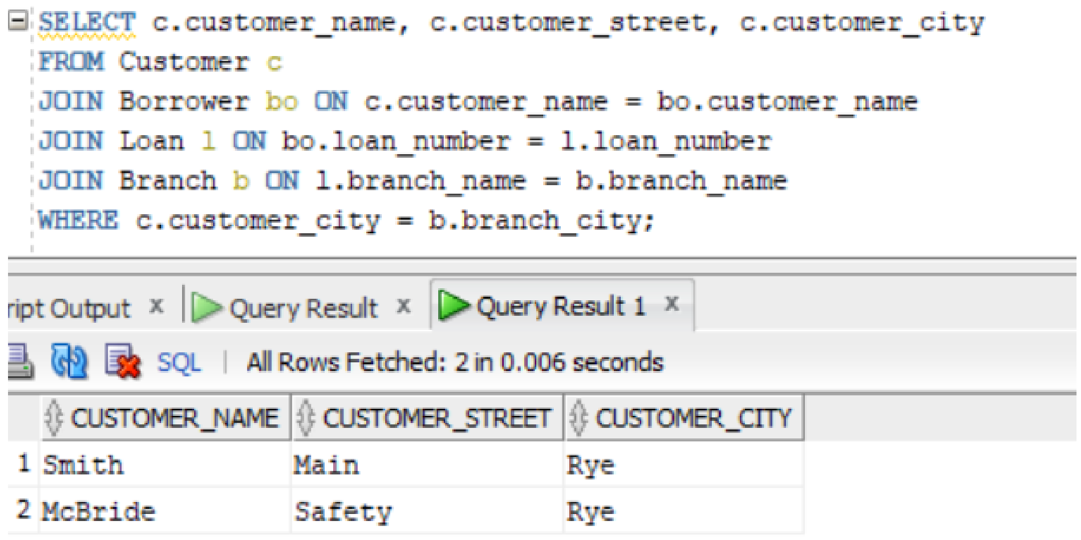
Script Output × | Query Result ×

SQL | All Rows Fetched: 2 in 0.01 seconds

	CUSTOMER_NAME	CUSTOMER_STREET	CUSTOMER_CITY
1	Smith	Main	Rye
2	Majeris	First	Rye

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

Without using subqueries:

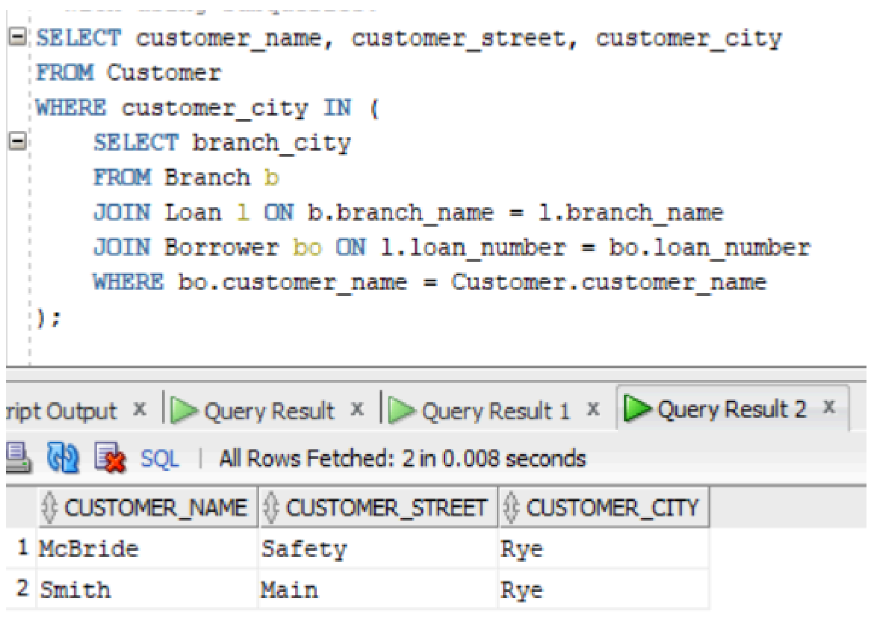


```
SELECT c.customer_name, c.customer_street, c.customer_city
FROM Customer c
JOIN Borrower bo ON c.customer_name = bo.customer_name
JOIN Loan l ON bo.loan_number = l.loan_number
JOIN Branch b ON l.branch_name = b.branch_name
WHERE c.customer_city = b.branch_city;
```

All Rows Fetched: 2 in 0.006 seconds

	CUSTOMER_NAME	CUSTOMER_STREET	CUSTOMER_CITY
1	Smith	Main	Rye
2	McBride	Safety	Rye

With using subqueries:



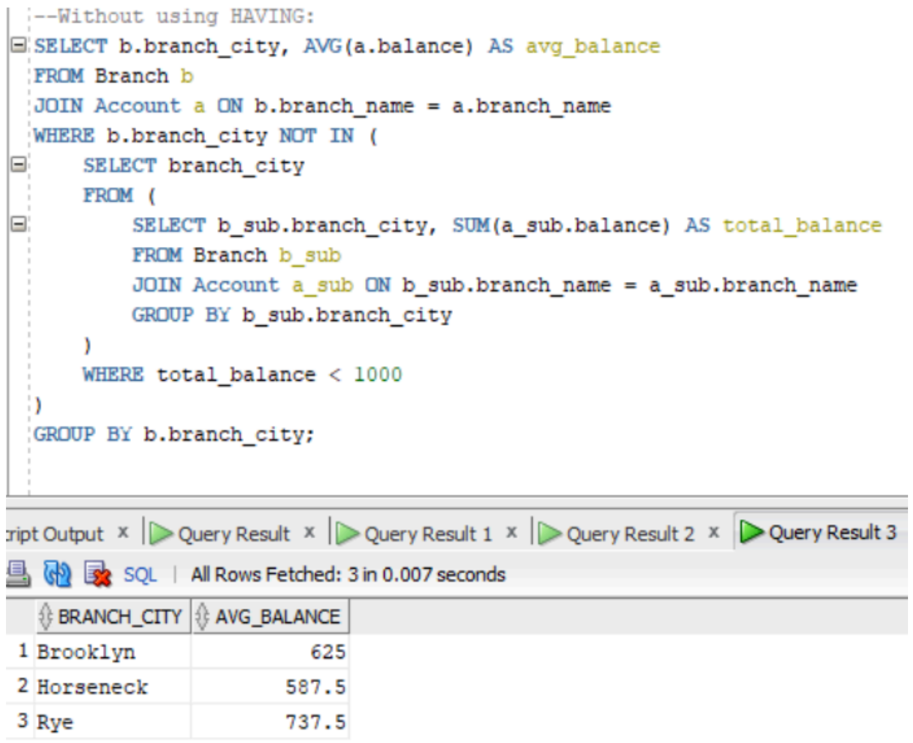
```
SELECT customer_name, customer_street, customer_city
FROM Customer
WHERE customer_city IN (
    SELECT branch_city
    FROM Branch b
    JOIN Loan l ON b.branch_name = l.branch_name
    JOIN Borrower bo ON l.loan_number = bo.loan_number
    WHERE bo.customer_name = Customer.customer_name
);
```

All Rows Fetched: 2 in 0.008 seconds

	CUSTOMER_NAME	CUSTOMER_STREET	CUSTOMER_CITY
1	McBride	Safety	Rye
2	Smith	Main	Rye

3. For each branch city, find the average balance of all accounts in that city. Exclude branch cities where the total balance is less than 1000.

Without using `HAVING`:



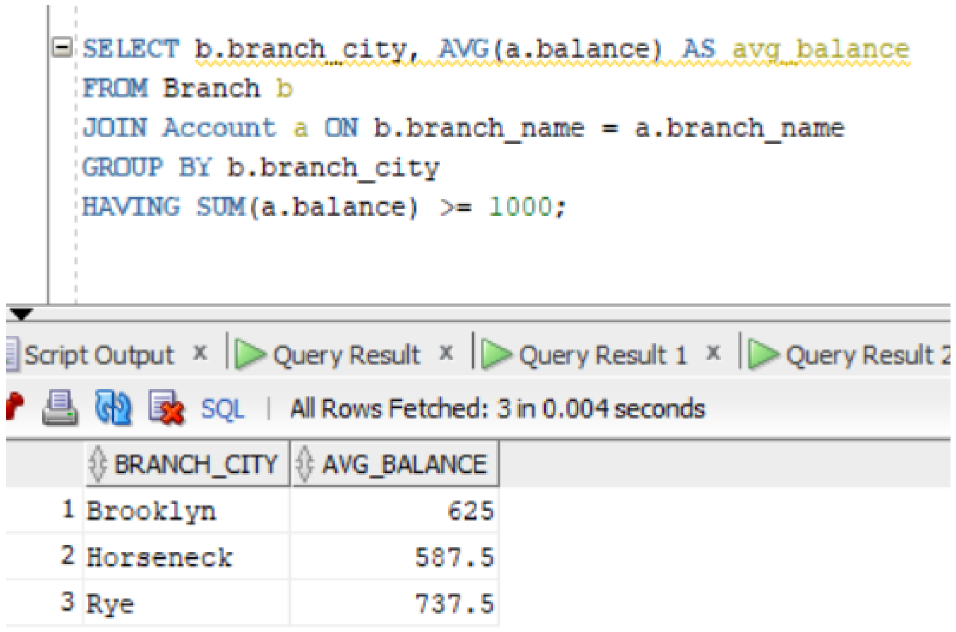
```
--Without using HAVING:
SELECT b.branch_city, AVG(a.balance) AS avg_balance
FROM Branch b
JOIN Account a ON b.branch_name = a.branch_name
WHERE b.branch_city NOT IN (
    SELECT branch_city
    FROM (
        SELECT b_sub.branch_city, SUM(a_sub.balance) AS total_balance
        FROM Branch b_sub
        JOIN Account a_sub ON b_sub.branch_name = a_sub.branch_name
        GROUP BY b_sub.branch_city
    )
    WHERE total_balance < 1000
)
GROUP BY b.branch_city;
```

ript Output × | Query Result × | Query Result 1 × | Query Result 2 × | Query Result 3

SQL | All Rows Fetched: 3 in 0.007 seconds

	BRANCH_CITY	AVG_BALANCE
1	Brooklyn	625
2	Horseneck	587.5
3	Rye	737.5

With using `HAVING`:



```
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;
```

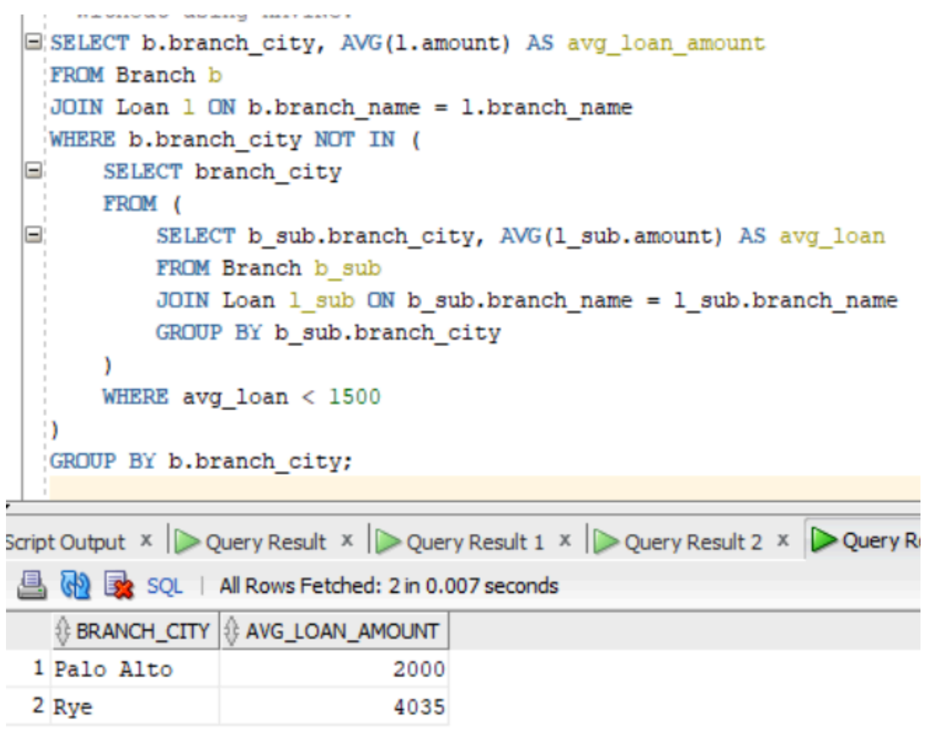
Script Output × | Query Result × | Query Result 1 × | Query Result 2

SQL | All Rows Fetched: 3 in 0.004 seconds

	BRANCH_CITY	AVG_BALANCE
1	Brooklyn	625
2	Horseneck	587.5
3	Rye	737.5

4. For each branch city, find the average loan amount. Exclude branch cities where the average loan amount is less than 1500.

Without using `HAVING`:

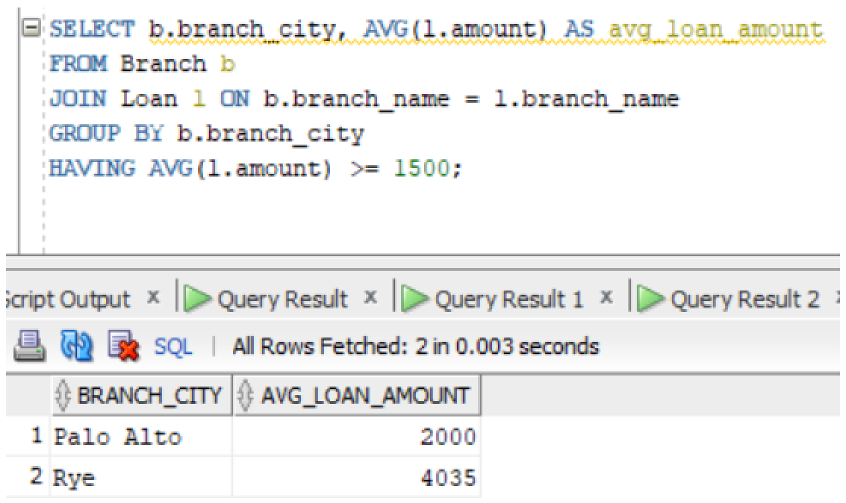


```
SELECT b.branch_city, AVG(l.amount) AS avg_loan_amount
FROM Branch b
JOIN Loan l ON b.branch_name = l.branch_name
WHERE b.branch_city NOT IN (
    SELECT branch_city
    FROM (
        SELECT b_sub.branch_city, AVG(l_sub.amount) AS avg_loan
        FROM Branch b_sub
        JOIN Loan l_sub ON b_sub.branch_name = l_sub.branch_name
        GROUP BY b_sub.branch_city
    )
    WHERE avg_loan < 1500
)
GROUP BY b.branch_city;
```

```
All Rows Fetched: 2 in 0.007 seconds

   BRANCH_CITY   AVG_LOAN_AMOUNT
1  Palo Alto                2000
2  Rye                      4035
```

Using `HAVING`:



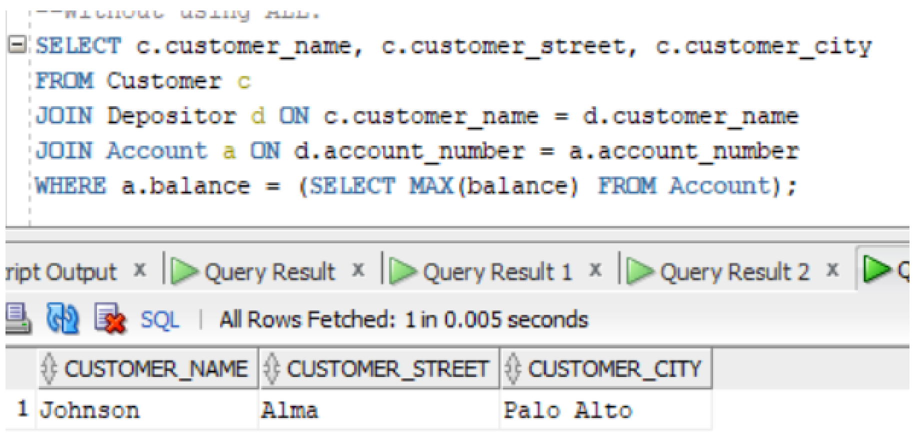
```
SELECT b.branch_city, AVG(l.amount) AS avg_loan_amount
FROM Branch b
JOIN Loan l ON b.branch_name = l.branch_name
GROUP BY b.branch_city
HAVING AVG(l.amount) >= 1500;
```

```
All Rows Fetched: 2 in 0.003 seconds

   BRANCH_CITY   AVG_LOAN_AMOUNT
1  Palo Alto                2000
2  Rye                      4035
```

5. Find the customer with the account having the highest balance.

Without using ALL:

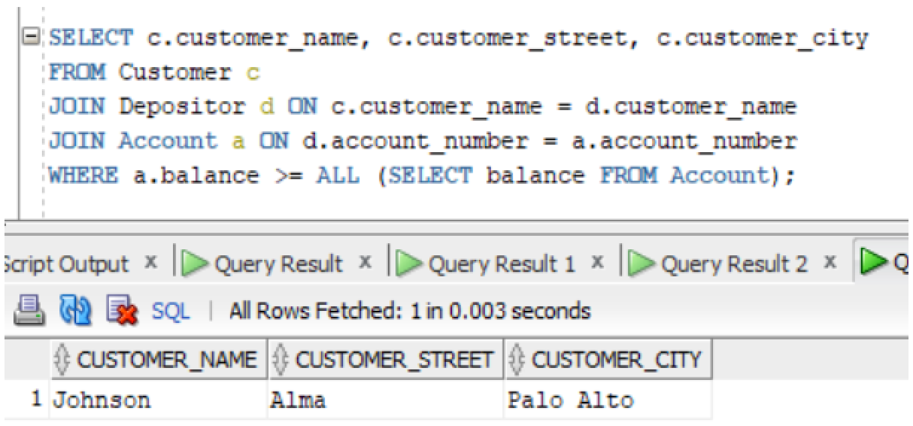


```
--Without using ALL:
SELECT c.customer_name, c.customer_street, c.customer_city
FROM Customer c
JOIN Depositor d ON c.customer_name = d.customer_name
JOIN Account a ON d.account_number = a.account_number
WHERE a.balance = (SELECT MAX(balance) FROM Account);
```

```
All Rows Fetched: 1 in 0.005 seconds

  CUSTOMER_NAME  CUSTOMER_STREET  CUSTOMER_CITY
1 Johnson        Alma             Palo Alto
```

Using ALL:



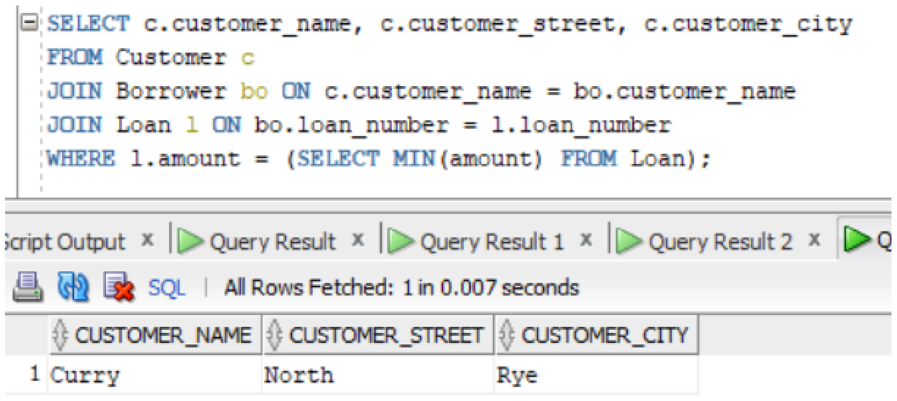
```
SELECT c.customer_name, c.customer_street, c.customer_city
FROM Customer c
JOIN Depositor d ON c.customer_name = d.customer_name
JOIN Account a ON d.account_number = a.account_number
WHERE a.balance >= ALL (SELECT balance FROM Account);
```

```
All Rows Fetched: 1 in 0.003 seconds

  CUSTOMER_NAME  CUSTOMER_STREET  CUSTOMER_CITY
1 Johnson        Alma             Palo Alto
```

6. Find the customer with the loan having the lowest amount.

Without using ALL:

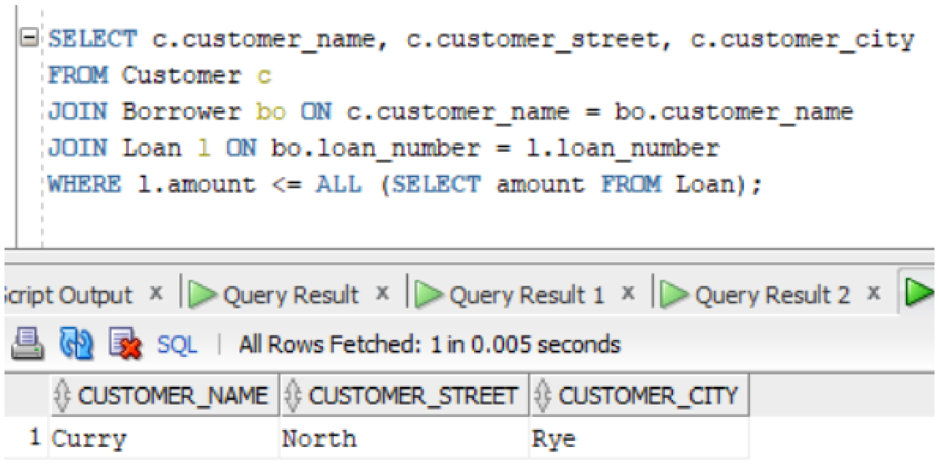


```
SELECT c.customer_name, c.customer_street, c.customer_city
FROM Customer c
JOIN Borrower bo ON c.customer_name = bo.customer_name
JOIN Loan l ON bo.loan_number = l.loan_number
WHERE l.amount = (SELECT MIN(amount) FROM Loan);
```

```
All Rows Fetched: 1 in 0.007 seconds

  CUSTOMER_NAME  CUSTOMER_STREET  CUSTOMER_CITY
1 Curry          North            Rye
```

Using ALL:



```
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);
```

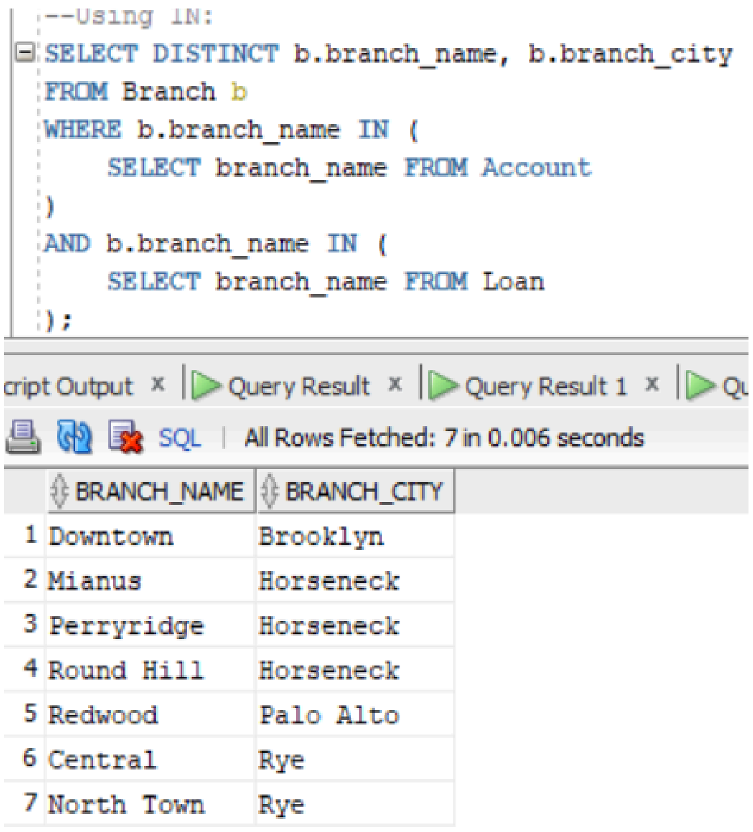
cript Output × | Query Result × | Query Result 1 × | Query Result 2 ×

SQL | All Rows Fetched: 1 in 0.005 seconds

	CUSTOMER_NAME	CUSTOMER_STREET	CUSTOMER_CITY
1	Curry	North	Rye

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

Using IN:



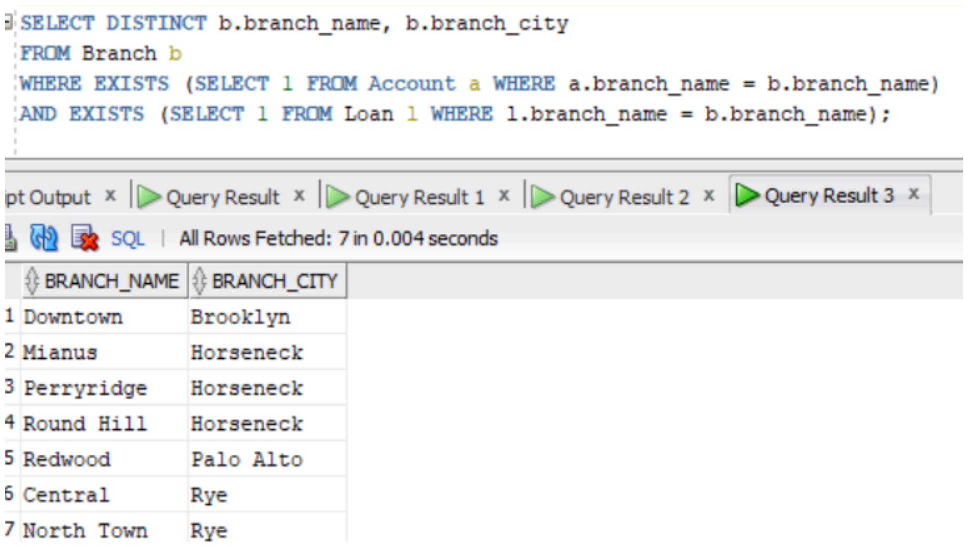
```
--Using IN:
SELECT DISTINCT b.branch_name, b.branch_city
FROM Branch b
WHERE b.branch_name IN (
    SELECT branch_name FROM Account
)
AND b.branch_name IN (
    SELECT branch_name FROM Loan
);
```

cript Output × | Query Result × | Query Result 1 × | Qu

SQL | All Rows Fetched: 7 in 0.006 seconds

	BRANCH_NAME	BRANCH_CITY
1	Downtown	Brooklyn
2	Mianus	Horseneck
3	Perryridge	Horseneck
4	Round Hill	Horseneck
5	Redwood	Palo Alto
6	Central	Rye
7	North Town	Rye

Using EXISTS:



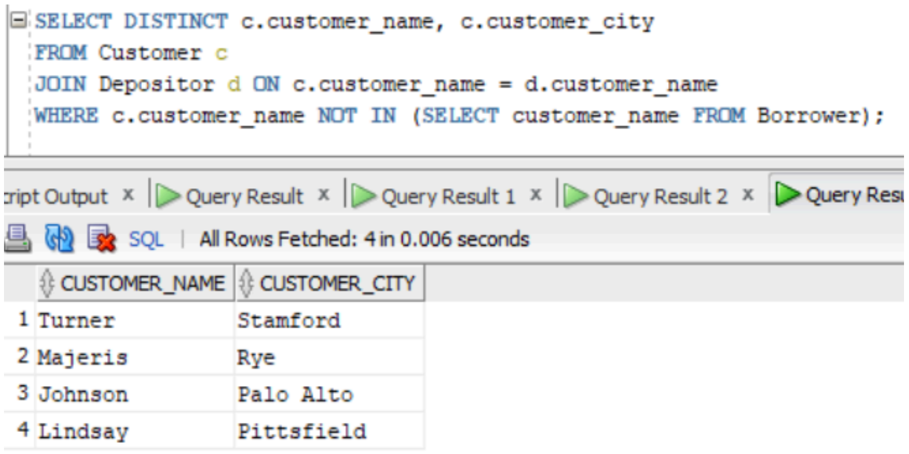
```
SELECT DISTINCT b.branch_name, b.branch_city
FROM Branch b
WHERE EXISTS (SELECT 1 FROM Account a WHERE a.branch_name = b.branch_name)
AND EXISTS (SELECT 1 FROM Loan l WHERE l.branch_name = b.branch_name);
```

```
All Rows Fetched: 7 in 0.004 seconds

  BRANCH_NAME  BRANCH_CITY
1 Downtown     Brooklyn
2 Mianus       Horseneck
3 Perryridge   Horseneck
4 Round Hill   Horseneck
5 Redwood      Palo Alto
6 Central      Rye
7 North Town   Rye
```

8. Find distinct customers who do not have loans but have accounts.

Using IN:

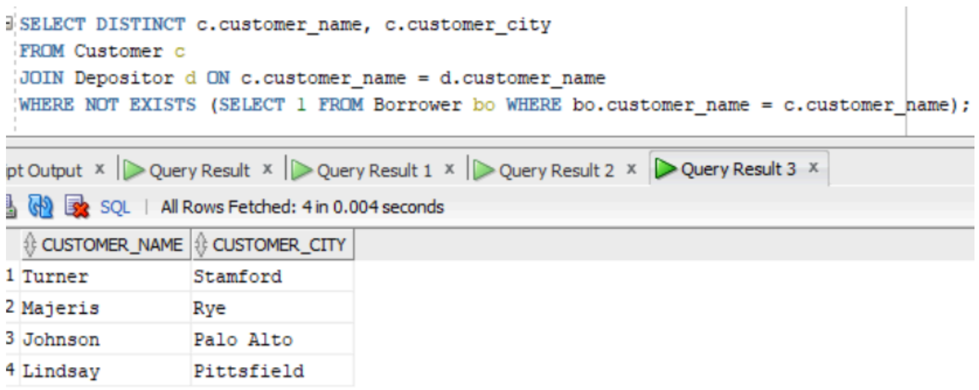


```
SELECT DISTINCT c.customer_name, c.customer_city
FROM Customer c
JOIN Depositor d ON c.customer_name = d.customer_name
WHERE c.customer_name NOT IN (SELECT customer_name FROM Borrower);
```

```
All Rows Fetched: 4 in 0.006 seconds

  CUSTOMER_NAME  CUSTOMER_CITY
1 Turner         Stamford
2 Majeris        Rye
3 Johnson        Palo Alto
4 Lindsay        Pittsfield
```

Using EXISTS:



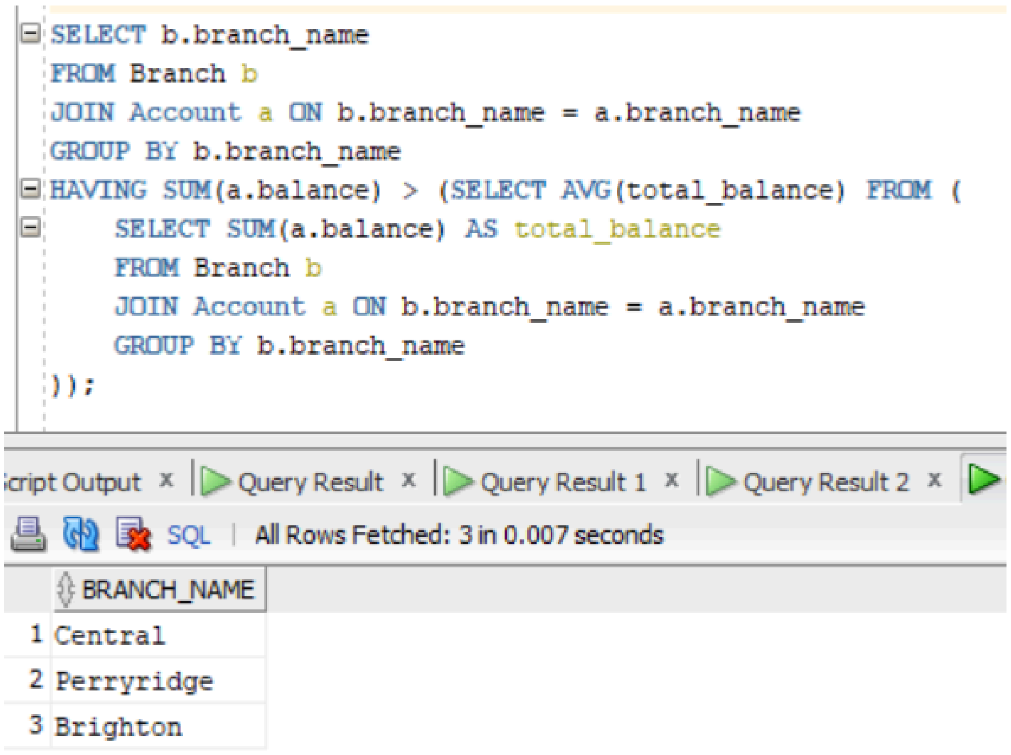
```
SELECT DISTINCT c.customer_name, c.customer_city
FROM Customer c
JOIN Depositor d ON c.customer_name = d.customer_name
WHERE NOT EXISTS (SELECT 1 FROM Borrower bo WHERE bo.customer_name = c.customer_name);
```

```
All Rows Fetched: 4 in 0.004 seconds

  CUSTOMER_NAME  CUSTOMER_CITY
1 Turner         Stamford
2 Majeris        Rye
3 Johnson        Palo Alto
4 Lindsay        Pittsfield
```

9. Find branches with a total account balance greater than the average total balance across all branches.

Without `WITH`:

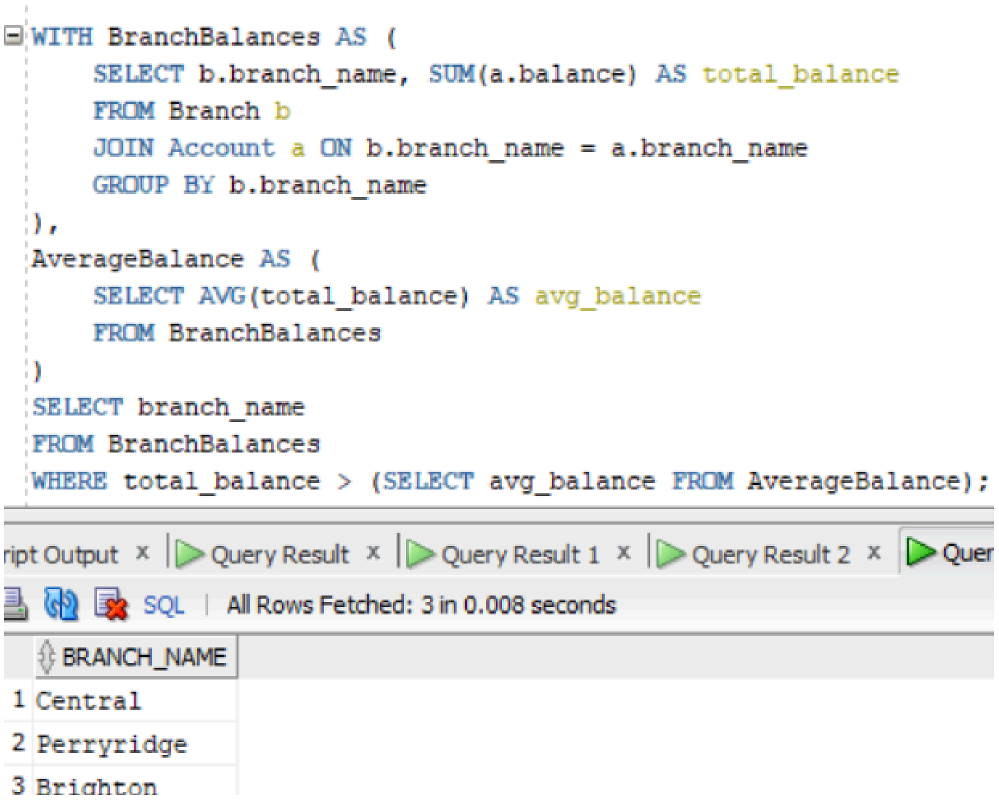


```sql
SELECT b.branch_name
FROM Branch b
JOIN Account a ON b.branch_name = a.branch_name
GROUP BY b.branch_name
HAVING SUM(a.balance) > (SELECT AVG(total_balance) FROM (
    SELECT SUM(a.balance) AS total_balance
    FROM Branch b
    JOIN Account a ON b.branch_name = a.branch_name
    GROUP BY b.branch_name
));
```

All Rows Fetched: 3 in 0.007 seconds

```
  BRANCH_NAME
1 Central
2 Perryridge
3 Brighton
```

Using `WITH`:



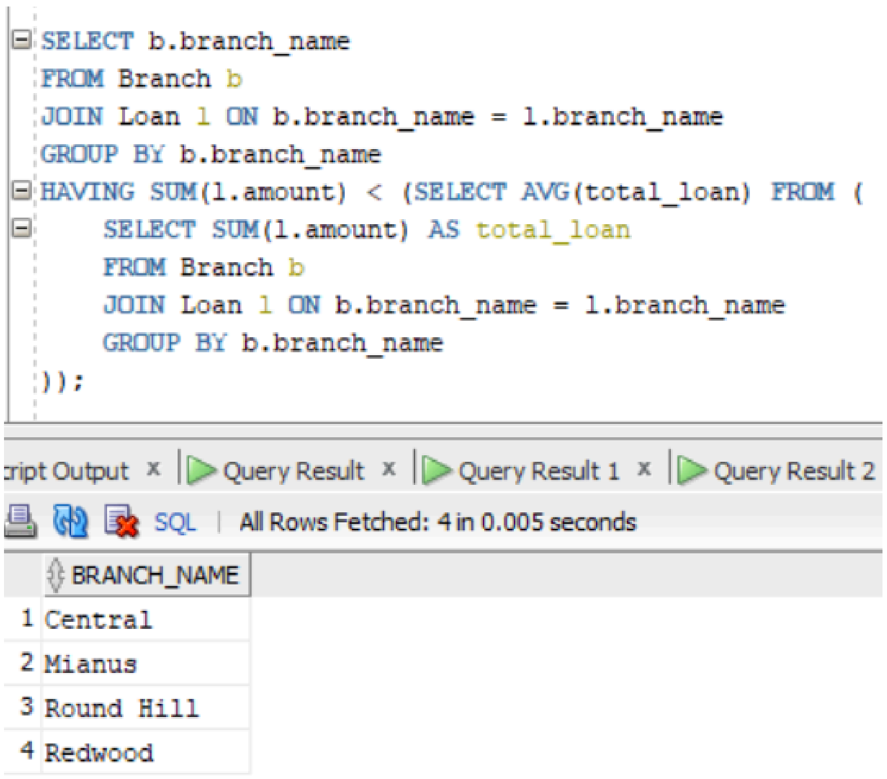
```sql
WITH BranchBalances AS (
    SELECT b.branch_name, SUM(a.balance) AS total_balance
    FROM Branch b
    JOIN Account a ON b.branch_name = a.branch_name
    GROUP BY b.branch_name
),
AverageBalance AS (
    SELECT AVG(total_balance) AS avg_balance
    FROM BranchBalances
)
SELECT branch_name
FROM BranchBalances
WHERE total_balance > (SELECT avg_balance FROM AverageBalance);
```

All Rows Fetched: 3 in 0.008 seconds

```
  BRANCH_NAME
1 Central
2 Perryridge
3 Brighton
```

10. Find branches with a total loan amount less than the average total loan amount across all branches.

Without WITH:

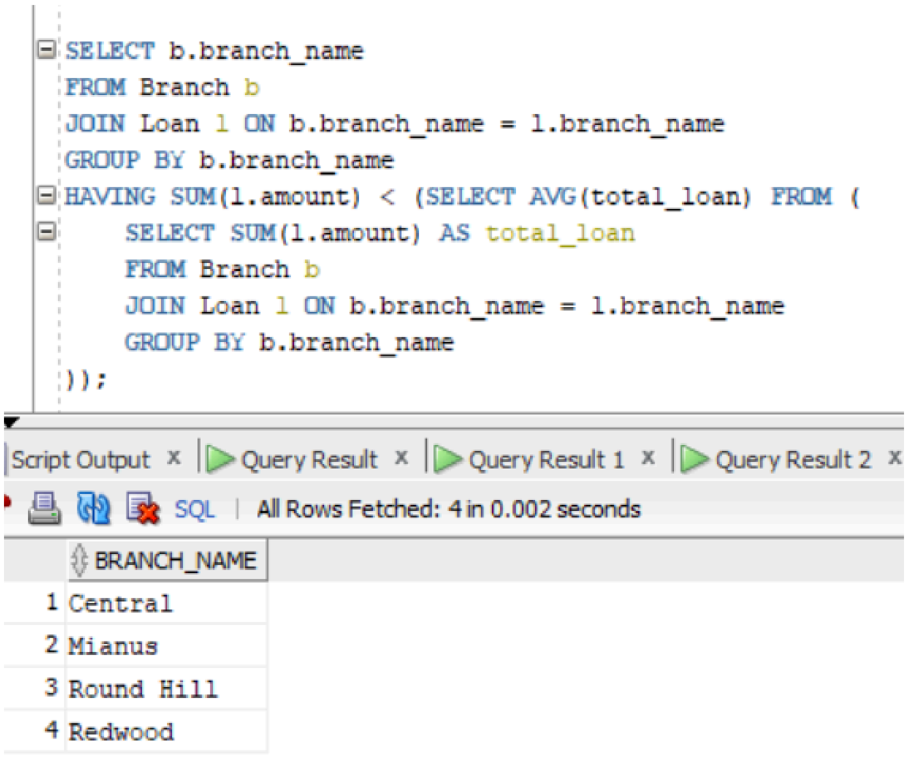


```
SELECT b.branch_name
FROM Branch b
JOIN Loan l ON b.branch_name = l.branch_name
GROUP BY b.branch_name
HAVING SUM(l.amount) < (SELECT AVG(total_loan) FROM (
    SELECT SUM(l.amount) AS total_loan
    FROM Branch b
    JOIN Loan l ON b.branch_name = l.branch_name
    GROUP BY b.branch_name
));
```

```
All Rows Fetched: 4 in 0.005 seconds

  BRANCH_NAME
1 Central
2 Mianus
3 Round Hill
4 Redwood
```

Using WITH:



```
SELECT b.branch_name
FROM Branch b
JOIN Loan l ON b.branch_name = l.branch_name
GROUP BY b.branch_name
HAVING SUM(l.amount) < (SELECT AVG(total_loan) FROM (
    SELECT SUM(l.amount) AS total_loan
    FROM Branch b
    JOIN Loan l ON b.branch_name = l.branch_name
    GROUP BY b.branch_name
));
```

```
All Rows Fetched: 4 in 0.002 seconds

  BRANCH_NAME
1 Central
2 Mianus
3 Round Hill
4 Redwood
```