**Lab Tasks:**

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)**

-- Without subquery

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

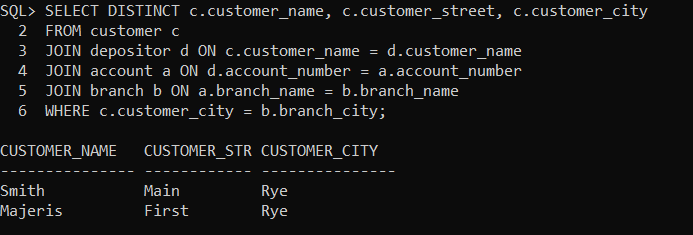
FROM customer c

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;



-- With subquery

SELECT customer\_name, customer\_street, customer\_city

FROM customer

WHERE customer\_name IN (

SELECT d.customer\_name

FROM depositor d

JOIN account a ON d.account\_number = a.account\_number

JOIN branch b ON a.branch\_name = b.branch\_name

WHERE b.branch\_city = (

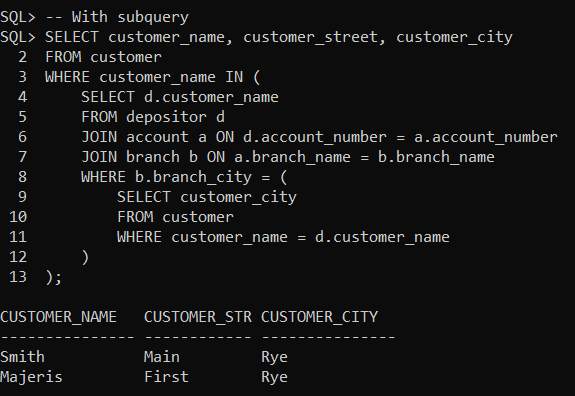
SELECT customer\_city

FROM customer

WHERE customer\_name = d.customer\_name

)

);



1. **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)**

-- Without subquery

SELECT DISTINCT C.\*

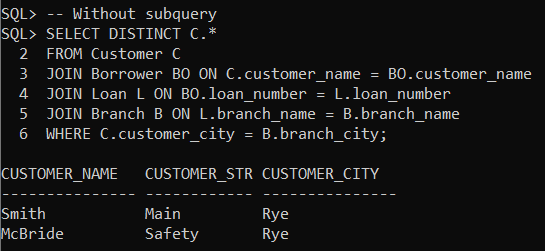
FROM Customer C

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;



-- With subquery

SELECT \*

FROM Customer C

WHERE C.customer\_name IN (

    SELECT BO.customer\_name

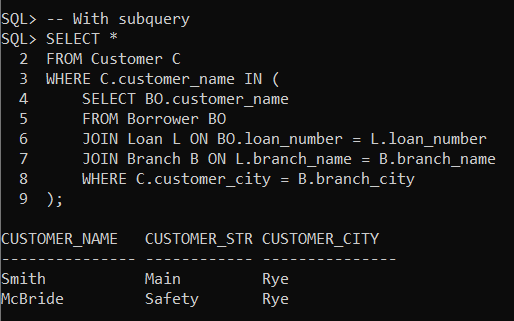
    FROM Borrower BO

    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

);



1. **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)**

-- Without HAVING clause

SELECT B.branch\_city, AVG(A.balance) AS avg\_balance

FROM Account A

JOIN Branch B ON A.branch\_name = B.branch\_name

WHERE B.branch\_city NOT IN (

SELECT B2.branch\_city

FROM Account A2

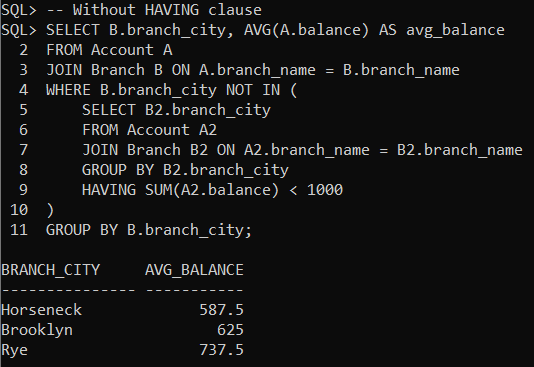
JOIN Branch B2 ON A2.branch\_name = B2.branch\_name

GROUP BY B2.branch\_city

HAVING SUM(A2.balance) < 1000

)

GROUP BY B.branch\_city;



-- With HAVING clause

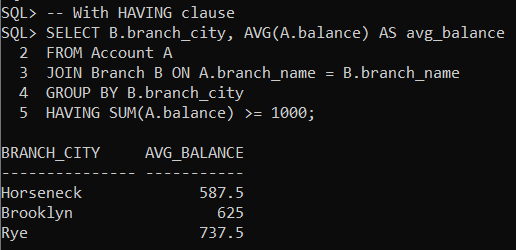
SELECT B.branch\_city, AVG(A.balance) AS avg\_balance

FROM Account A

JOIN Branch B ON A.branch\_name = B.branch\_name

GROUP BY B.branch\_city

HAVING SUM(A.balance) >= 1000;



1. **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)**

-- Without HAVING clause

SELECT B.branch\_city, AVG(L.amount) AS avg\_loan

FROM Loan L

JOIN Branch B ON L.branch\_name = B.branch\_name

WHERE B.branch\_city NOT IN (

SELECT B2.branch\_city

FROM Loan L2

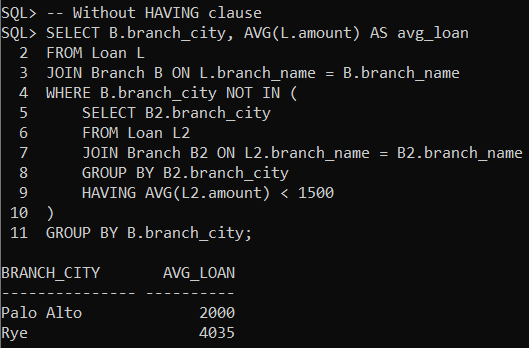
JOIN Branch B2 ON L2.branch\_name = B2.branch\_name

GROUP BY B2.branch\_city

HAVING AVG(L2.amount) < 1500

)

GROUP BY B.branch\_city;



-- With HAVING clause

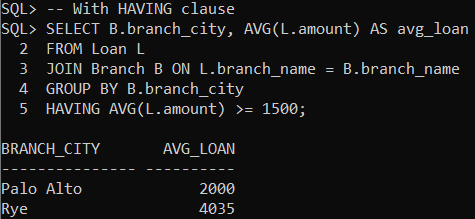
SELECT B.branch\_city, AVG(L.amount) AS avg\_loan

FROM Loan L

JOIN Branch B ON L.branch\_name = B.branch\_name

GROUP BY B.branch\_city

HAVING AVG(L.amount) >= 1500;



1. **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)**

-- Without using ALL keyword:

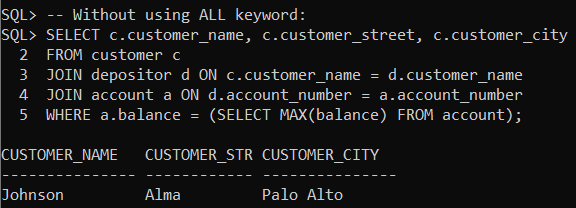
SELECT c.customer\_name, c.customer\_street, c.customer\_city

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);



-- Using ALL keyword:

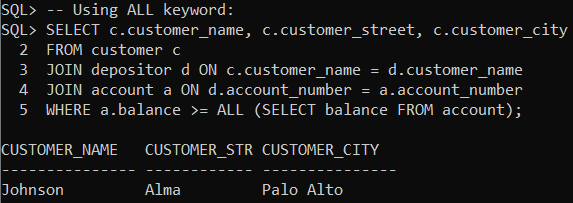
SELECT c.customer\_name, c.customer\_street, c.customer\_city

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);



1. **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)**

-- With ALL:

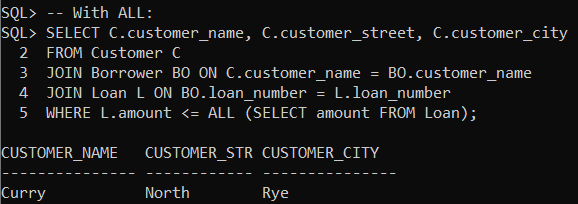
SELECT C.customer\_name, C.customer\_street, C.customer\_city

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);



-- Without ALL:

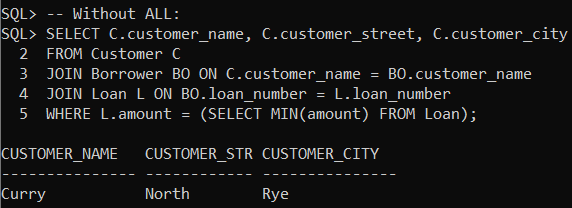
SELECT C.customer\_name, C.customer\_street, C.customer\_city

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);



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

-- Using Exists

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

FROM Branch B

WHERE EXISTS (

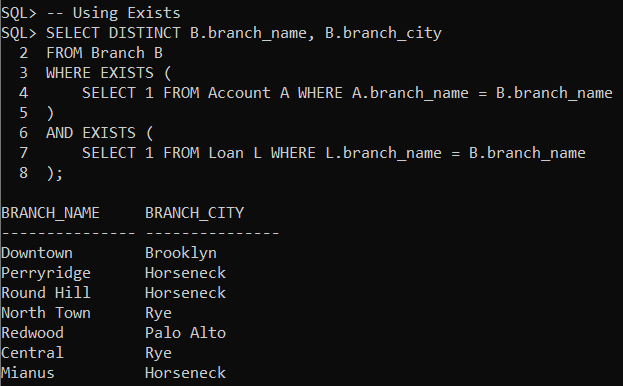
SELECT 1 FROM Account A WHERE A.branch\_name = B.branch\_name

)

AND EXISTS (

SELECT 1 FROM Loan L WHERE L.branch\_name = B.branch\_name

);



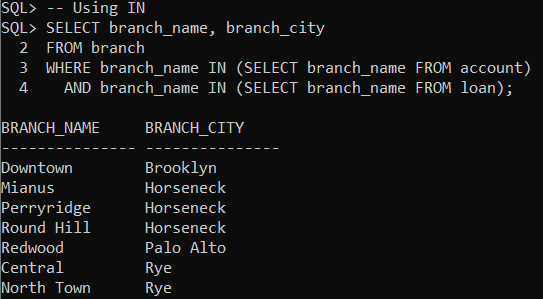
-- Using IN

SELECT branch\_name, branch\_city

FROM branch

WHERE branch\_name IN (SELECT branch\_name FROM account)

  AND branch\_name IN (SELECT branch\_name FROM loan);



1. **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)**

-- Using NOT IN keyword:

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

FROM customer c

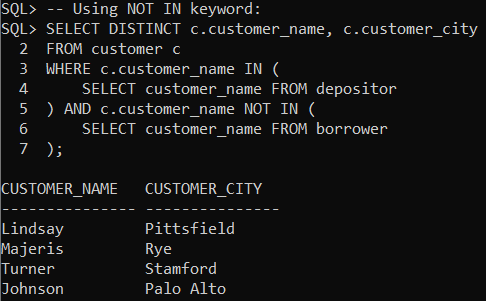
WHERE c.customer\_name IN (

SELECT customer\_name FROM depositor

) AND c.customer\_name NOT IN (

SELECT customer\_name FROM borrower

);



-- Using NOT EXISTS keyword:

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

FROM customer c

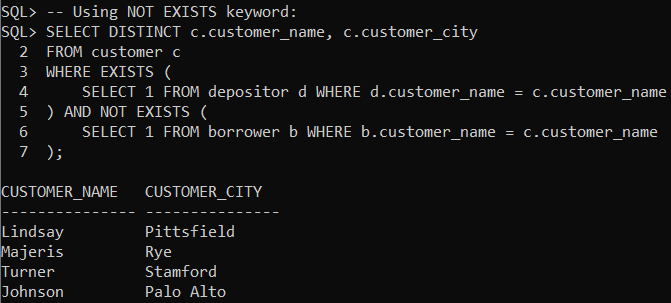
WHERE EXISTS (

    SELECT 1 FROM depositor d WHERE d.customer\_name = c.customer\_name

) AND NOT EXISTS (

    SELECT 1 FROM borrower b WHERE b.customer\_name = c.customer\_name

);



1. **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)**

-- With clause

WITH Avg\_Balance AS (

SELECT AVG(balance) AS overall\_avg\_balance

FROM Account

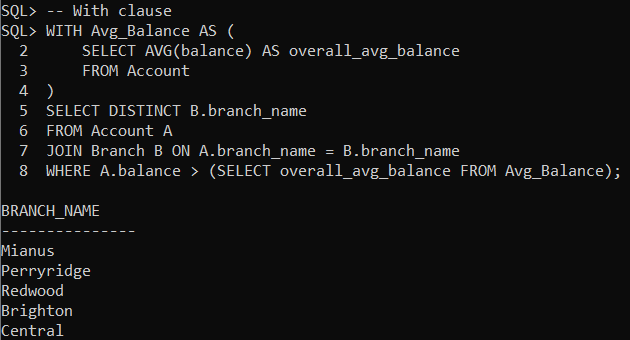
)

SELECT DISTINCT B.branch\_name

FROM Account A

JOIN Branch B ON A.branch\_name = B.branch\_name

WHERE A.balance > (SELECT overall\_avg\_balance FROM Avg\_Balance);



-- Without clause

SELECT DISTINCT B.branch\_name

FROM Account A

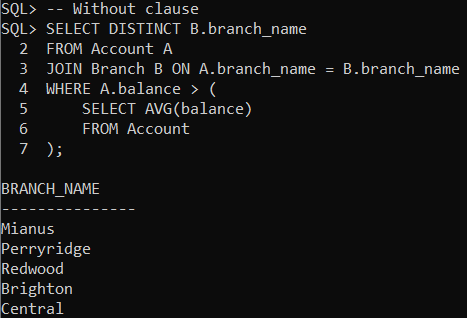
JOIN Branch B ON A.branch\_name = B.branch\_name

WHERE A.balance > (

    SELECT AVG(balance)

    FROM Account

);



1. **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)**

-- With clause

WITH Avg\_Loan AS (

SELECT AVG(amount) AS overall\_avg\_loan

FROM Loan

)

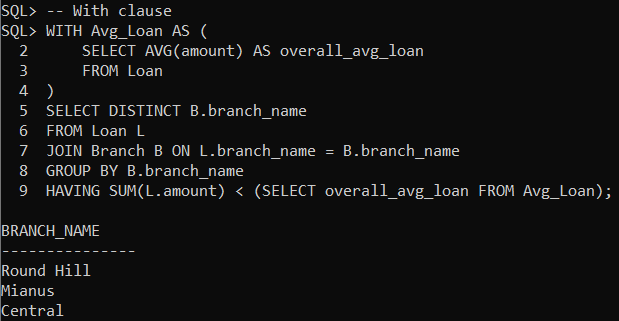
SELECT DISTINCT B.branch\_name

FROM Loan L

JOIN Branch B ON L.branch\_name = B.branch\_name

GROUP BY B.branch\_name

HAVING SUM(L.amount) < (SELECT overall\_avg\_loan FROM Avg\_Loan);



-- Without clause

SELECT DISTINCT B.branch\_name

FROM Loan L

JOIN Branch B ON L.branch\_name = B.branch\_name

GROUP BY B.branch\_name

HAVING SUM(L.amount) < (

    SELECT AVG(amount)

    FROM Loan

);

