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 c.\*

FROM Customer c, Account a, Branch b, depositor d

WHERE c.customer\_city = b.branch\_city

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

AND c.customer\_name = d.customer\_name;

(with subquery)

SELECT \*

FROM Customer c

WHERE customer\_city IN (

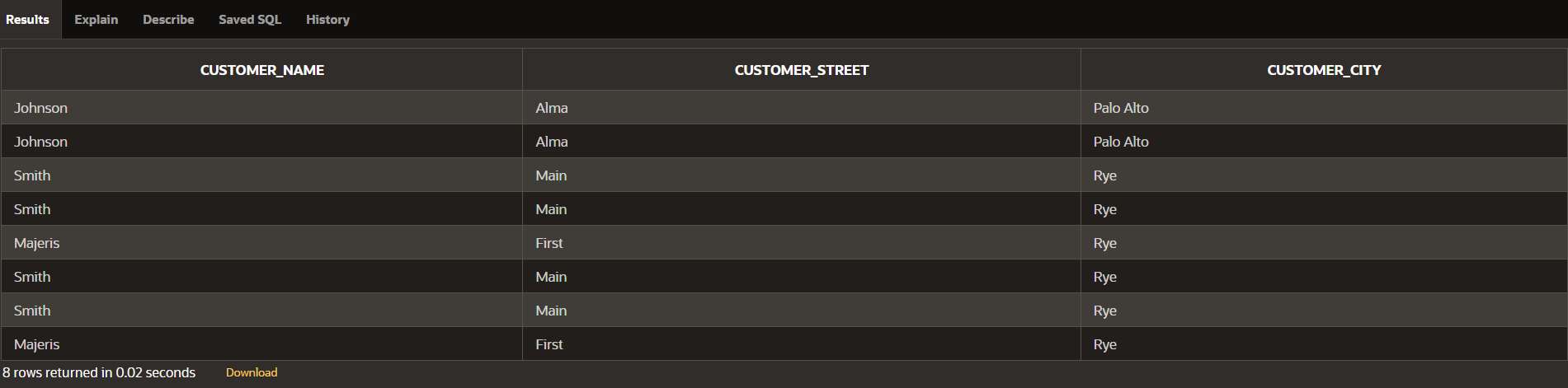
SELECT branch\_city

FROM Branch b, Account a, depositor d

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

AND d.customer\_name = c.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 subqueries)

SELECT c.\*

FROM Customer c, Loan l, Branch b, depositor d

WHERE c.customer\_city = b.branch\_city

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

AND c.customer\_name = d.customer\_name;

(with subqueries)

SELECT \*

FROM Customer c

WHERE customer\_city IN (

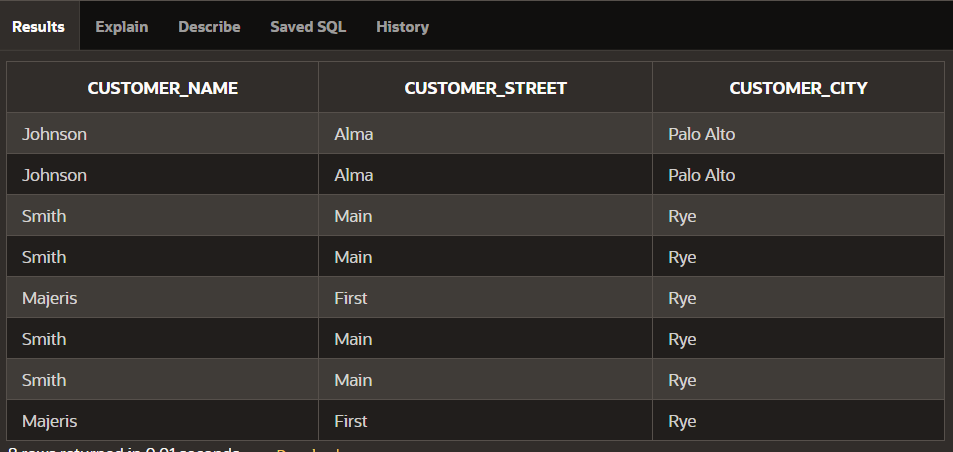
SELECT branch\_city

FROM Branch b, Loan l, depositor d

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

AND d.customer\_name = c.customer\_name

);



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)

i.

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

FROM Branch b, Account a

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

GROUP BY branch\_city

HAVING SUM(balance) >= 1000;

ii.

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 AVG(a.balance) \* COUNT(a.account\_number) >= 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)

i.

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;

ii.

SELECT branch\_city, avg\_loan\_amount

FROM (

SELECT b.branch\_city,

AVG(l.amount) AS avg\_loan\_amount

FROM

Branch b

JOIN

Loan l ON b.branch\_name = l.branch\_name

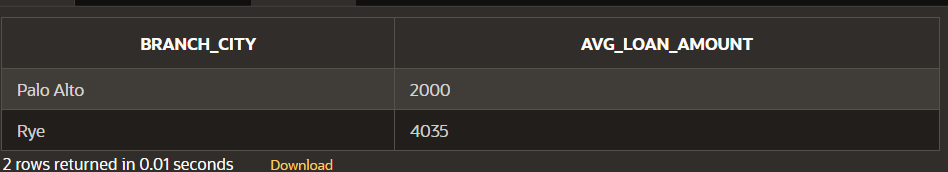
GROUP BY

b.branch\_city

) AS sub

WHERE

avg\_loan\_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)

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

FROM Customer c, Account a

WHERE (a.balance) >= ALL (SELECT balance FROM Account)

AND c.customer\_name IN (SELECT customer\_name FROM Depositor WHERE account\_number = a.account\_number);

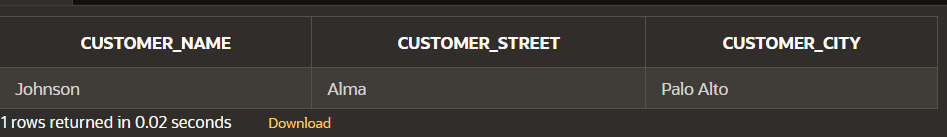
ii.SELECT c.customer\_name, c.customer\_street, c.customer\_city

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);



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)

i.

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

FROM Customer c, Loan l

WHERE (l.amount) <= ALL (SELECT amount FROM Loan)

AND c.customer\_name IN (SELECT customer\_name FROM Borrower WHERE loan\_number = l.loan\_number);

ii.

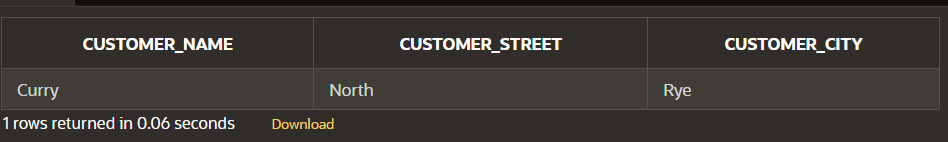
SELECT c.customer\_name, c.customer\_street, c.customer\_city

FROM Customer c

JOIN Borrower b ON c.customer\_name = b.customer\_name

JOIN Loan l ON b.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)

i.

SELECT DISTINCT branch\_name, branch\_city

FROM Branch

WHERE branch\_name IN (

SELECT branch\_name

FROM Account

) AND branch\_name IN (

SELECT branch\_name

FROM Loan

);

ii.

SELECT DISTINCT branch\_name, branch\_city

FROM Branch b

WHERE EXISTS (

SELECT 1

FROM Account a

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

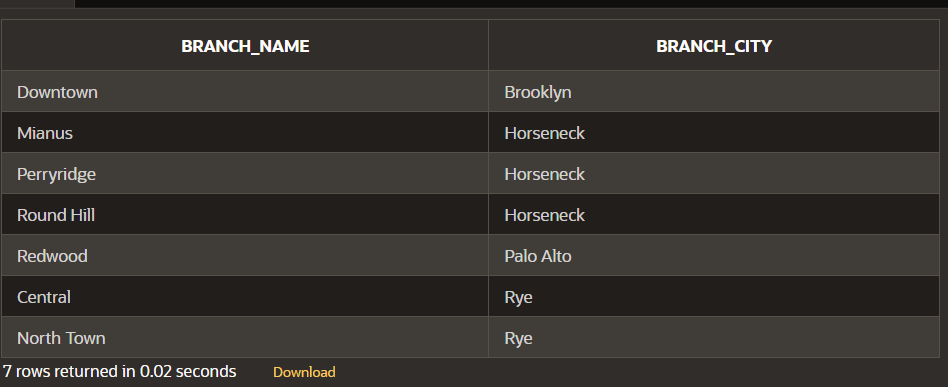
) AND EXISTS (

SELECT 1

FROM Loan l

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

);



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)

i.SELECT DISTINCT customer\_name, customer\_city

FROM Customer

WHERE customer\_name NOT IN (

SELECT DISTINCT c.customer\_name

FROM Customer c

JOIN Borrower b ON c.customer\_name = b.customer\_name

);

ii.SELECT DISTINCT customer\_name, customer\_city

FROM Customer c

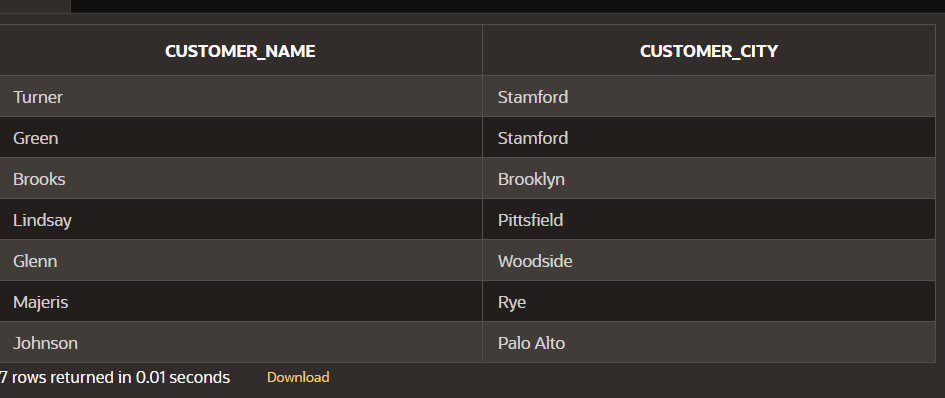
WHERE NOT EXISTS (

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)

i.WITH branch\_balances AS (

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

FROM Account

GROUP BY branch\_name

)

SELECT branch\_name

FROM branch\_balances

WHERE total\_balance > (

SELECT AVG(total\_balance)

FROM branch\_balances

);

ii.SELECT branch\_name

FROM Account

GROUP BY branch\_name

HAVING SUM(balance) > (

SELECT AVG(total\_balance)

FROM (

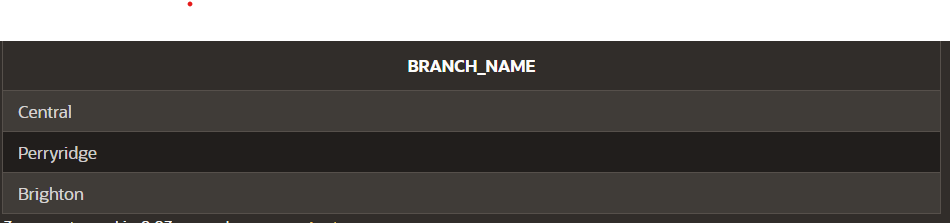
SELECT SUM(balance) AS total\_balance

FROM Account

GROUP BY branch\_name)

AS branch\_balances

);



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)

i. WITH branch\_loan\_amounts AS (

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

FROM Loan

GROUP BY branch\_name

)

SELECT branch\_name

FROM branch\_loan\_amounts

WHERE total\_amount < (

SELECT AVG(total\_amount)

FROM branch\_loan\_amounts

);

ii.SELECT branch\_name

FROM Loan

GROUP BY branch\_name

HAVING SUM(amount) < (

SELECT AVG(total\_amount)

FROM (

SELECT SUM(amount) AS total\_amount

FROM Loan

GROUP BY branch\_name

) AS branch\_loan\_amounts

);

