CSE302 (Section 2)

[Spring 2022]

Lab Assignment Submission Report

Assignment Title: LAB 05

Submitted by:

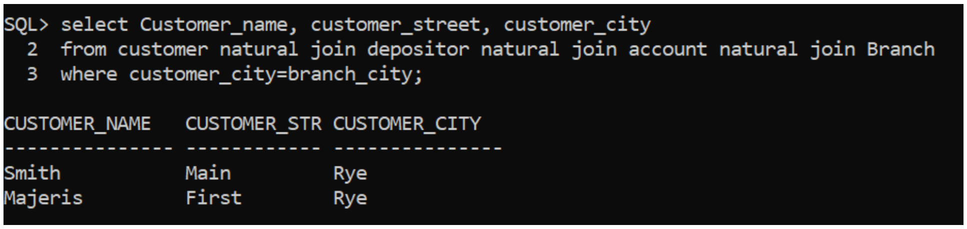
Rizvee Hassan Prito

2019-3-60-041

1. Screenshots

Problem 1:

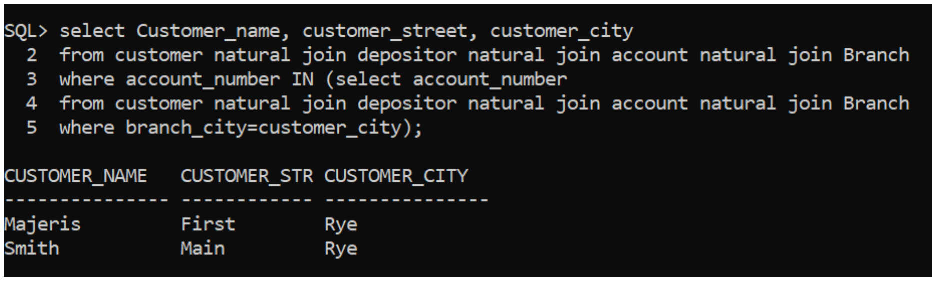
Without subqueries:



```
SQL> select Customer_name, customer_street, customer_city
  2  from customer natural join depositor natural join account natural join Branch
  3  where customer_city=branch_city;

CUSTOMER_NAME   CUSTOMER_STR CUSTOMER_CITY
--------------- ------------ ---------------
Smith           Main         Rye
Majeris         First        Rye
```

With subqueries:

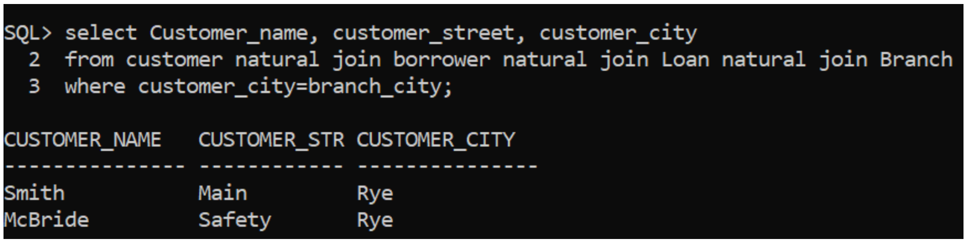


```
SQL> select Customer_name, customer_street, customer_city
  2  from customer natural join depositor natural join account natural join Branch
  3  where account_number IN (select account_number
  4  from customer natural join depositor natural join account natural join Branch
  5  where branch_city=customer_city);

CUSTOMER_NAME   CUSTOMER_STR CUSTOMER_CITY
--------------- ------------ ---------------
Majeris         First        Rye
Smith           Main         Rye
```

Problem 2:

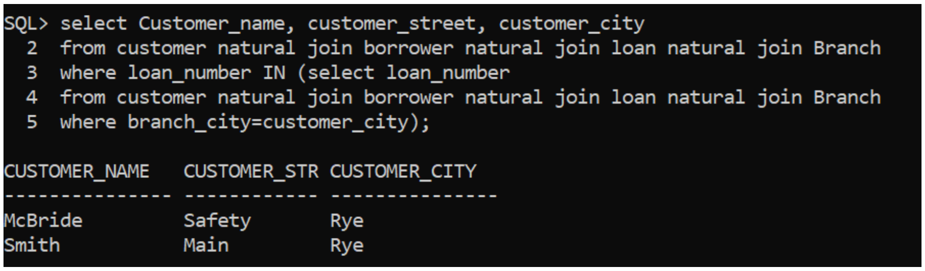
Without subqueries:



```
SQL> select Customer_name, customer_street, customer_city
  2  from customer natural join borrower natural join Loan natural join Branch
  3  where customer_city=branch_city;

CUSTOMER_NAME   CUSTOMER_STR CUSTOMER_CITY
--------------- ------------ ---------------
Smith           Main         Rye
McBride         Safety       Rye
```

With subqueries:

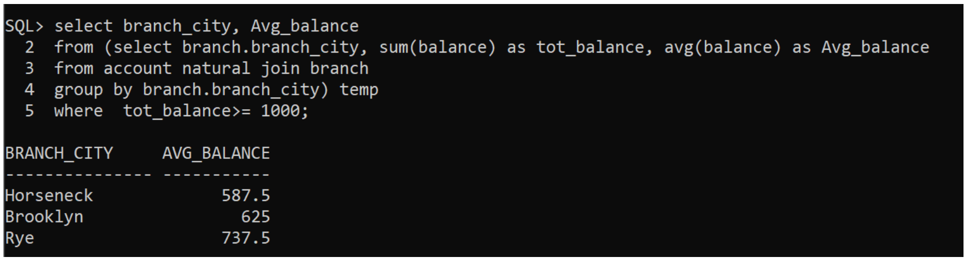


```
SQL> select Customer_name, customer_street, customer_city
  2  from customer natural join borrower natural join loan natural join Branch
  3  where loan_number IN (select loan_number
  4  from customer natural join borrower natural join loan natural join Branch
  5  where branch_city=customer_city);

CUSTOMER_NAME   CUSTOMER_STR CUSTOMER_CITY
--------------- ------------ ---------------
McBride         Safety       Rye
Smith           Main         Rye
```

Problem 3:

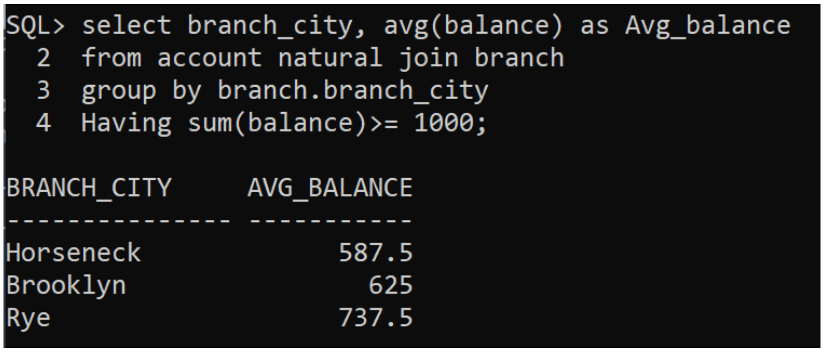
Without ‘Having’ clause:



```
SQL> select branch_city, Avg_balance
     from (select branch.branch_city, sum(balance) as tot_balance, avg(balance) as Avg_balance
     from account natural join branch
     group by branch.branch_city) temp
     where  tot_balance>= 1000;

BRANCH_CITY     AVG_BALANCE
--------------- -----------
Horseneck             587.5
Brooklyn                625
Rye                   737.5
```

With ‘Having’ clause:

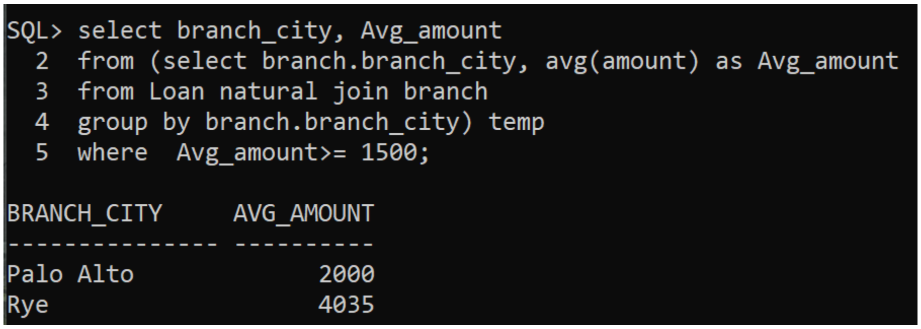


```
SQL> select branch_city, avg(balance) as Avg_balance
     from account natural join branch
     group by branch.branch_city
     Having sum(balance)>= 1000;

BRANCH_CITY     AVG_BALANCE
--------------- -----------
Horseneck             587.5
Brooklyn                625
Rye                   737.5
```

Problem 4:

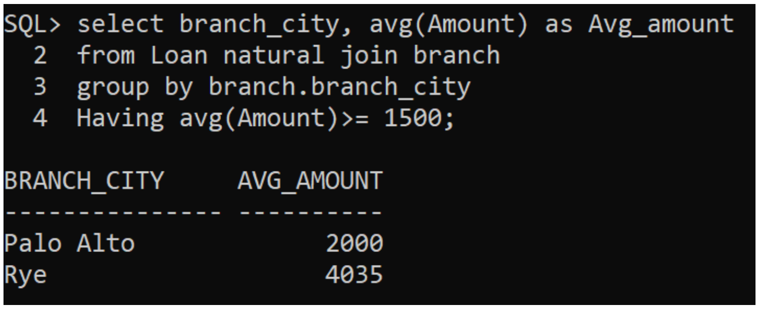
Without ‘Having’ clause:



```
SQL> select branch_city, Avg_amount
     from (select branch.branch_city, avg(amount) as Avg_amount
     from Loan natural join branch
     group by branch.branch_city) temp
     where  Avg_amount>= 1500;

BRANCH_CITY     AVG_AMOUNT
--------------- ----------
Palo Alto             2000
Rye                   4035
```

With ‘Having’ clause:

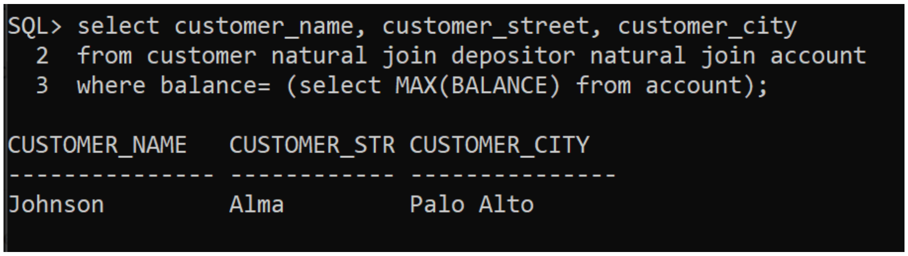


```
SQL> select branch_city, avg(Amount) as Avg_amount
     from Loan natural join branch
     group by branch.branch_city
     Having avg(Amount)>= 1500;

BRANCH_CITY     AVG_AMOUNT
--------------- ----------
Palo Alto             2000
Rye                   4035
```

Problem 5:

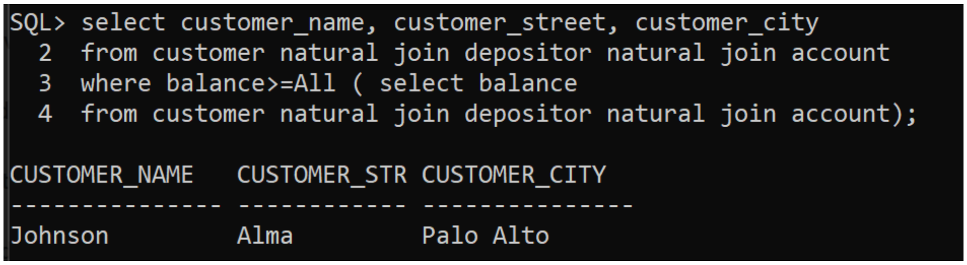
Without ‘All’ keyword:



```
SQL> select customer_name, customer_street, customer_city
     from customer natural join depositor natural join account
     where balance= (select MAX(BALANCE) from account);

CUSTOMER_NAME   CUSTOMER_STR CUSTOMER_CITY
--------------- ------------ ---------------
Johnson         Alma         Palo Alto
```

With ‘All’ keyword:

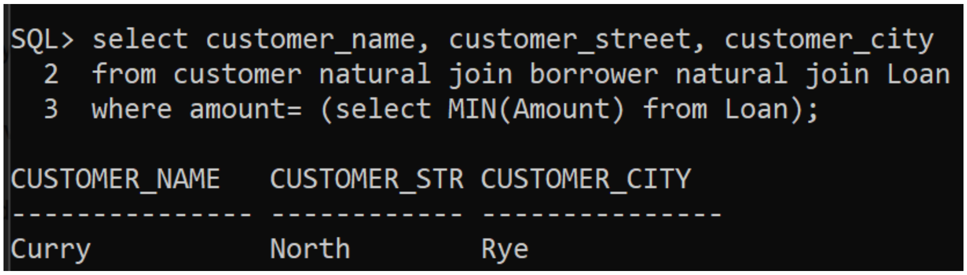


```
SQL> select customer_name, customer_street, customer_city
     from customer natural join depositor natural join account
     where balance>=All ( select balance
     from customer natural join depositor natural join account);

CUSTOMER_NAME   CUSTOMER_STR CUSTOMER_CITY
--------------- ------------ ---------------
Johnson         Alma         Palo Alto
```

Problem 6:

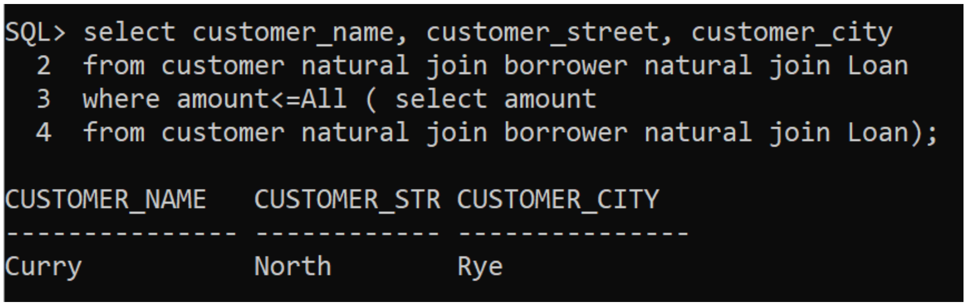
Without ‘All’ keyword:



```
SQL> select customer_name, customer_street, customer_city
     from customer natural join borrower natural join Loan
     where amount= (select MIN(Amount) from Loan);

CUSTOMER_NAME   CUSTOMER_STR CUSTOMER_CITY
--------------- ------------ ---------------
Curry           North        Rye
```

With ‘All’ keyword:

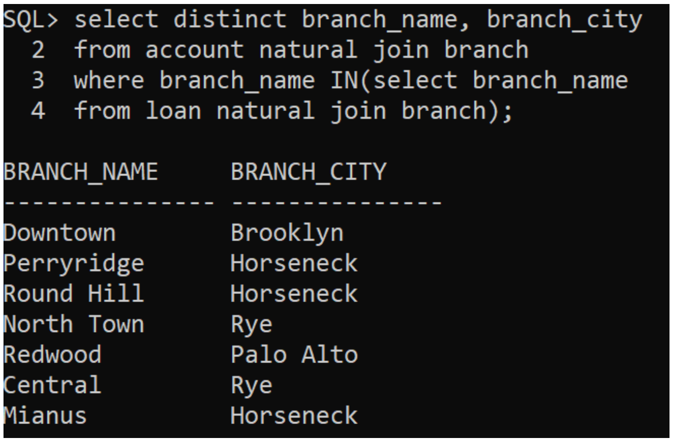


```
SQL> select customer_name, customer_street, customer_city
     from customer natural join borrower natural join Loan
     where amount<=All ( select amount
     from customer natural join borrower natural join Loan);

CUSTOMER_NAME   CUSTOMER_STR CUSTOMER_CITY
--------------- ------------ ---------------
Curry           North        Rye
```

Problem 7:

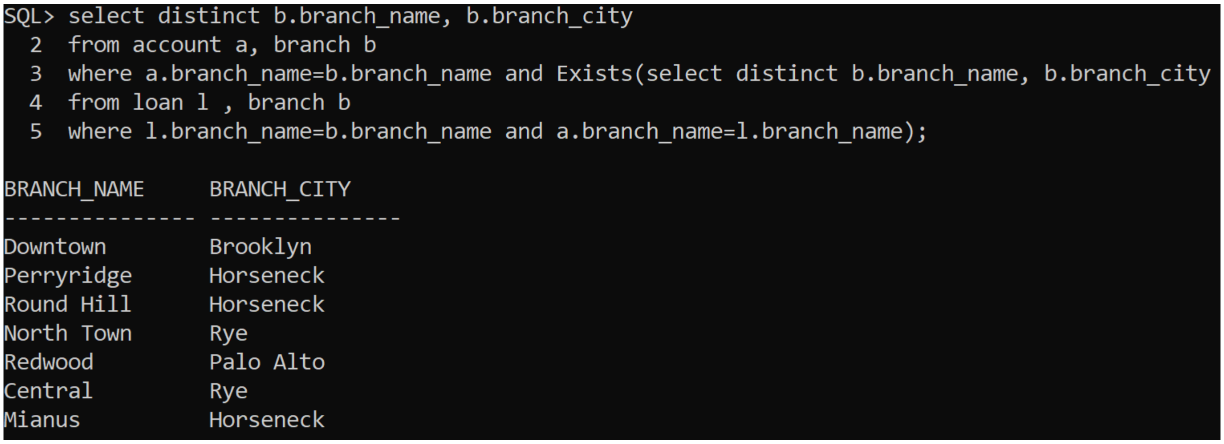
With ‘In’ keyword:



```
SQL> select distinct branch_name, branch_city
     from account natural join branch
     where branch_name IN(select branch_name
     from loan natural join branch);

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

<u>With 'Exists' keyword:</u>

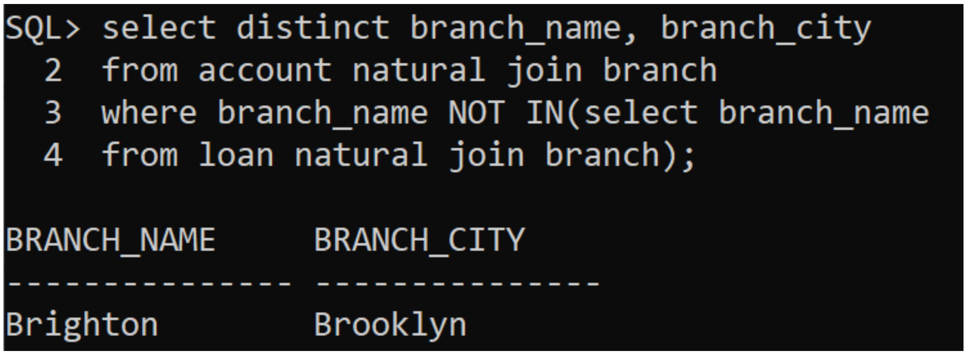


```
SQL> select distinct b.branch_name, b.branch_city
     from account a, branch b
     where a.branch_name=b.branch_name and Exists(select distinct b.branch_name, b.branch_city
     from loan l , branch b
     where l.branch_name=b.branch_name and a.branch_name=l.branch_name);

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

<u>Problem 8:</u>

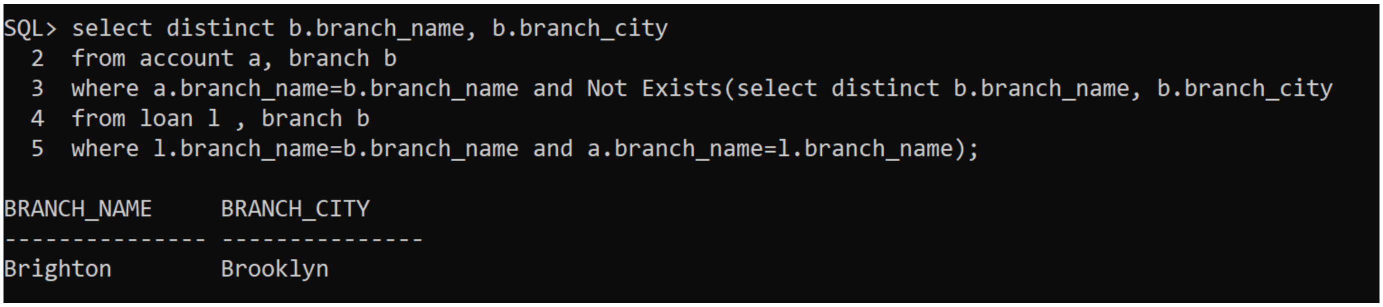
<u>With 'Not In' keyword:</u>



```
SQL> select distinct branch_name, branch_city
     from account natural join branch
     where branch_name NOT IN(select branch_name
     from loan natural join branch);

BRANCH_NAME     BRANCH_CITY
--------------- ---------------
Brighton        Brooklyn
```

<u>With 'Not Exists' keyword:</u>

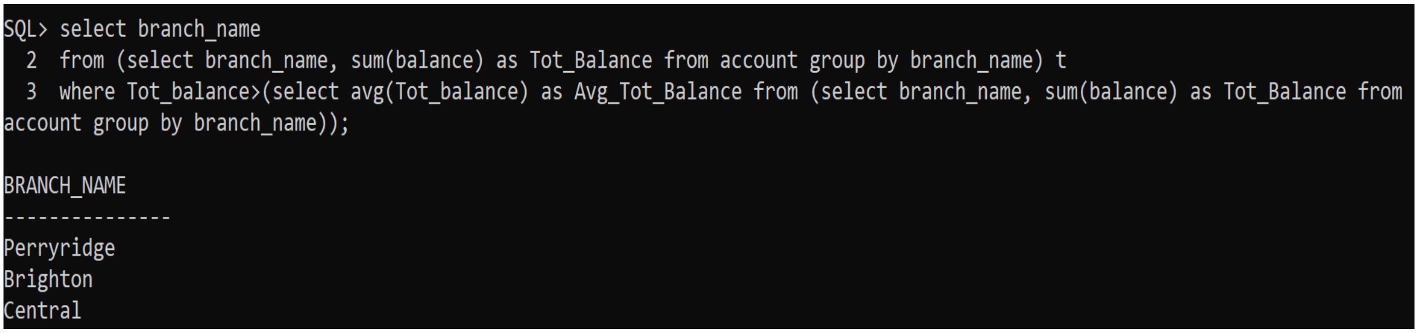


```
SQL> select distinct b.branch_name, b.branch_city
     from account a, branch b
     where a.branch_name=b.branch_name and Not Exists(select distinct b.branch_name, b.branch_city
     from loan l , branch b
     where l.branch_name=b.branch_name and a.branch_name=l.branch_name);

BRANCH_NAME     BRANCH_CITY
--------------- ---------------
Brighton        Brooklyn
```

Problem 9:

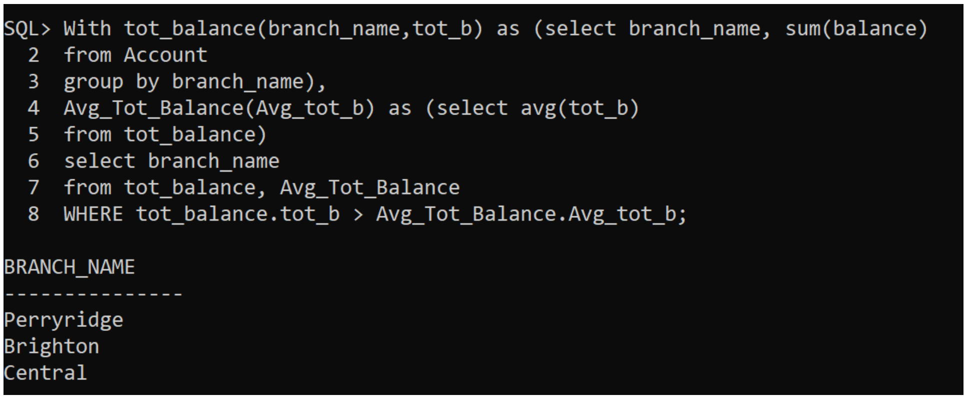
Without ‘With’ clause:



```
SQL> select branch_name
     from (select branch_name, sum(balance) as Tot_Balance from account group by branch_name) t
     where Tot_balance>(select avg(Tot_balance) as Avg_Tot_Balance from (select branch_name, sum(balance) as Tot_Balance from
account group by branch_name));

BRANCH_NAME
---------------
Perryridge
Brighton
Central
```

With ‘With’ clause:

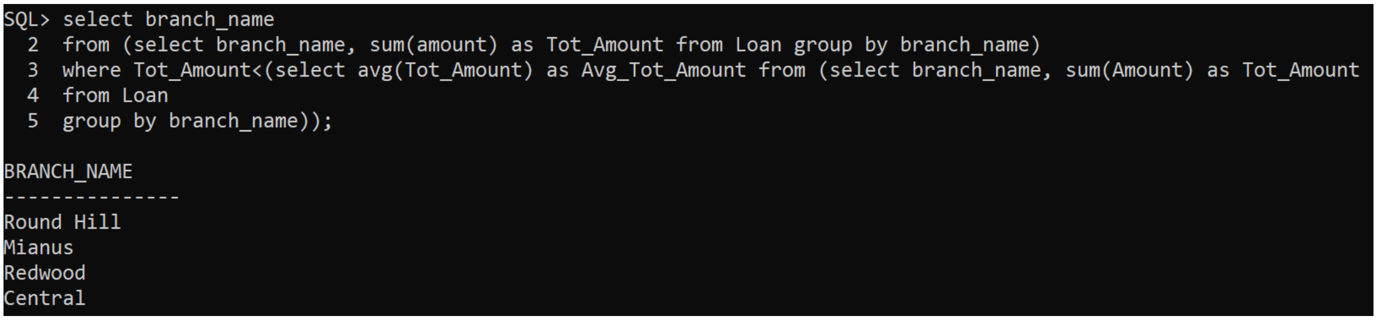


```
SQL> With tot_balance(branch_name,tot_b) as (select branch_name, sum(balance)
     from Account
     group by branch_name),
     Avg_Tot_Balance(Avg_tot_b) as (select avg(tot_b)
     from tot_balance)
     select branch_name
     from tot_balance, Avg_Tot_Balance
     WHERE tot_balance.tot_b > Avg_Tot_Balance.Avg_tot_b;

BRANCH_NAME
---------------
Perryridge
Brighton
Central
```

Problem 10:

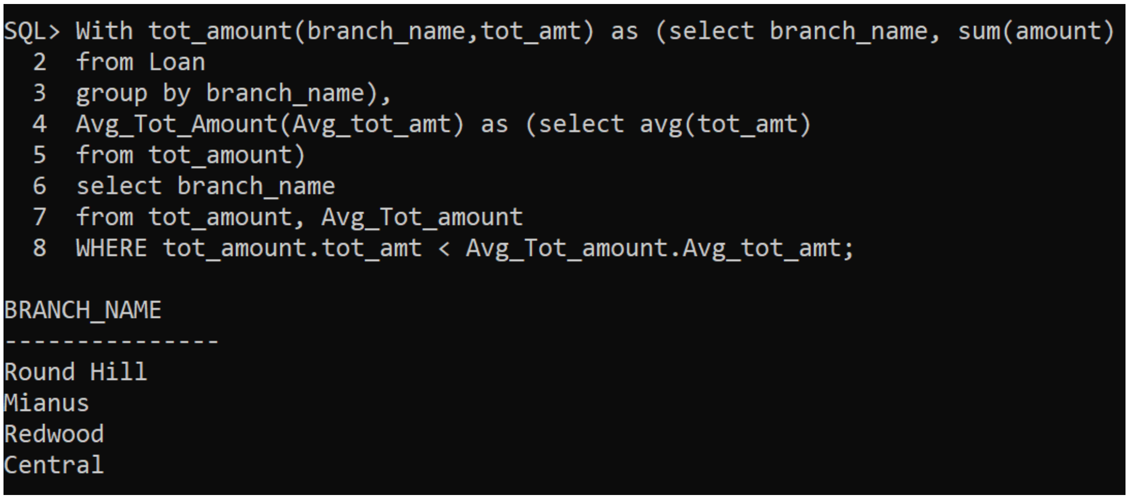
Without ‘With’ clause:



```
SQL> select branch_name
  2  from (select branch_name, sum(amount) as Tot_Amount from Loan group by branch_name)
  3  where Tot_Amount<(select avg(Tot_Amount) as Avg_Tot_Amount from (select branch_name, sum(Amount) as Tot_Amount
  4  from Loan
  5  group by branch_name));

BRANCH_NAME
---------------
Round Hill
Mianus
Redwood
Central
```

<u>With ‘With’ clause:</u>



```
SQL> With tot_amount(branch_name,tot_amt) as (select branch_name, sum(amount)
  2  from Loan
  3  group by branch_name),
  4  Avg_Tot_Amount(Avg_tot_amt) as (select avg(tot_amt)
  5  from tot_amount)
  6  select branch_name
  7  from tot_amount, Avg_Tot_amount
  8  WHERE tot_amount.tot_amt < Avg_Tot_amount.Avg_tot_amt;

BRANCH_NAME
---------------
Round Hill
Mianus
Redwood
Central
```

2. Learning outcomes:

From this lab I have learned how to write subquery commands. There are three types of method for doing subqueries. The clauses of three types of methods are ‘Select’, ‘From’, ‘Where’. ‘ALL\Not All’, ‘In\Not In’, ‘Exists\Not Exists’ are the keywords for writing ‘Where’ clause subquery command. I have learned the using of ‘From’ clause in doing subqueries. I have learned to do complex queries by using ‘With’ clause which really helps us to do complex queries much easily.