COMP 3005
Assignment #4
Krystian Wojcicki

Part 1. SQL Queries (90 marks)

Your Lastname Use Oracle-VM SQL Data Definition Language to create the following Bank-and-Customer database. You should properly define primary keys and foreign keys and then use SQL Query Language to express the following queries. In the document that you submit, you should have the question with its number, SQL query, and query result generated by Oracle-VM for each question. Each query is 3 marks and the result is 2 marks.

Bank

B#	Name	City
B1	England	London
B2	America	New York
B3	Royal	Toronto
B4	France	Paris

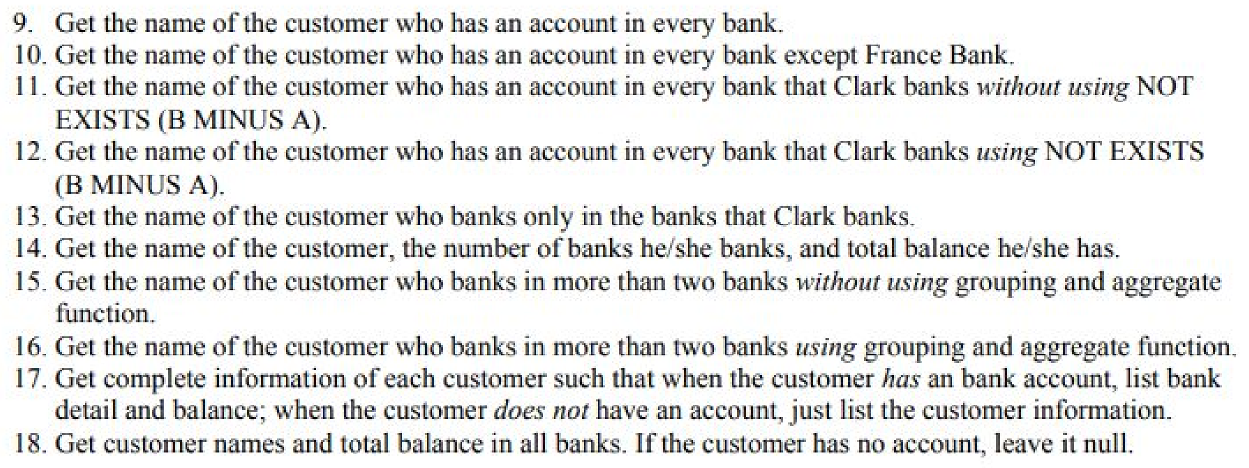
Customer

C#	Name	Age	City
C1	Adams	20	London
C2	Blake	30	Paris
C3	Clark	25	Paris
C4	Your Lastname	20	London
C5	Smith	30	Toronto

Account

C#	B#	Balance
C1	B1	1000
C1	B2	2000
C1	B3	3000
C1	B4	4000
C2	B1	2000
C2	B2	3000
C3	B2	3000
C3	B3	4000
C4	B2	4000
C4	B3	5000

1. Get the name of the bank that "Your Lastname" banks.
2. Get the name of the customer who banks in Royal bank *using EXISTS*.
3. Get the name of the customer who has an account with balance less than 3000.
4. Get the name of the customer who banks in Royal or America bank.
5. Get the customer name/bank name pairs such that the indicated customer has an account in the indicated bank.
6. Get the customer name/bank name pairs such that the indicated customer does not an account in the indicated bank.
7. Get names of the banks in which Blake or Clark has accounts *using UNION without duplicates*.
8. Get the name of the customer who does not have any bank account.

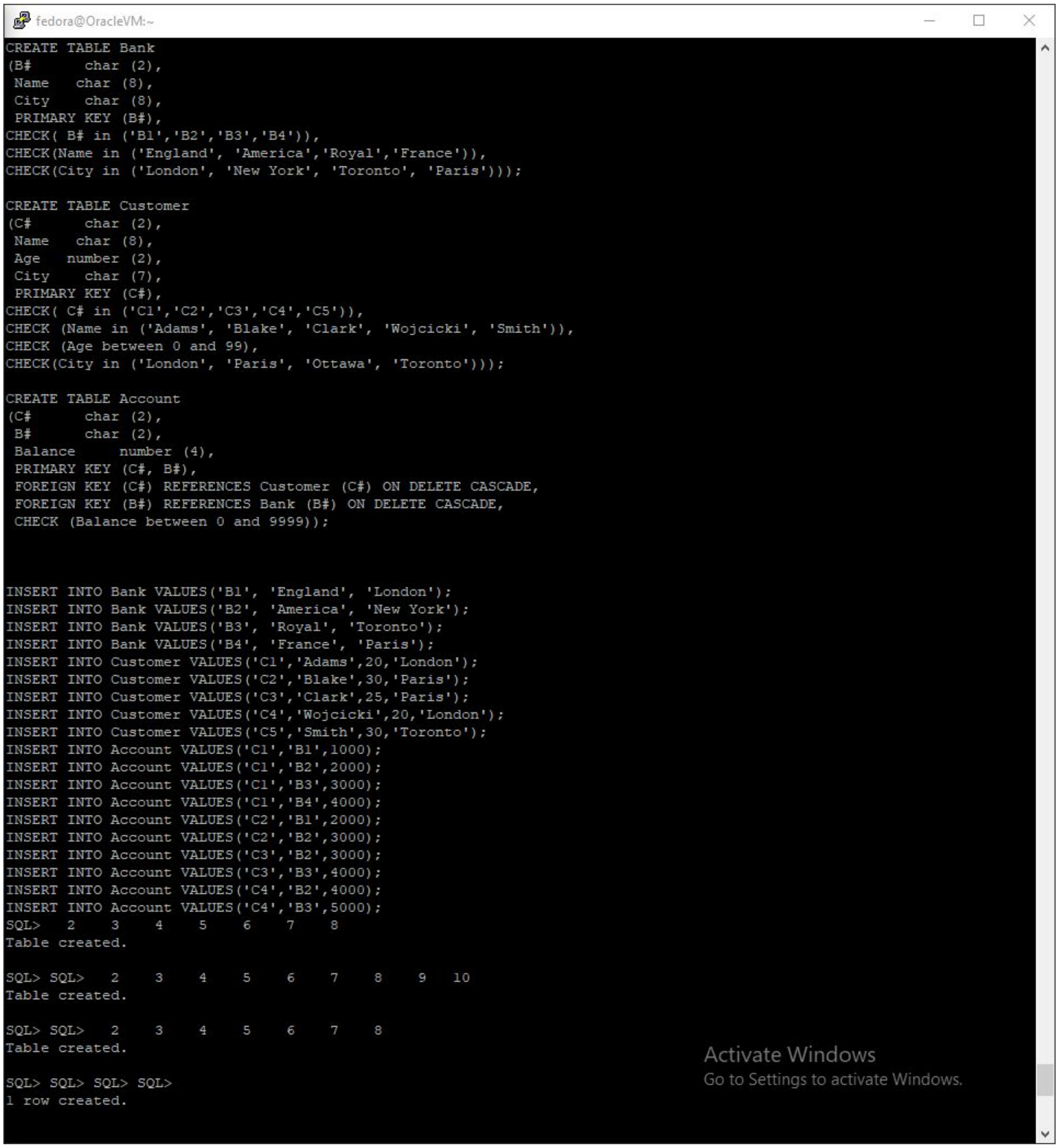


9. Get the name of the customer who has an account in every bank.
10. Get the name of the customer who has an account in every bank except France Bank.
11. Get the name of the customer who has an account in every bank that Clark banks *without using* NOT EXISTS (B MINUS A).
12. Get the name of the customer who has an account in every bank that Clark banks *using* NOT EXISTS (B MINUS A).
13. Get the name of the customer who banks only in the banks that Clark banks.
14. Get the name of the customer, the number of banks he/she banks, and total balance he/she has.
15. Get the name of the customer who banks in more than two banks *without using* grouping and aggregate function.
16. Get the name of the customer who banks in more than two banks *using* grouping and aggregate function.
17. Get complete information of each customer such that when the customer *has* an bank account, list bank detail and balance; when the customer *does not* have an account, just list the customer information.
18. Get customer names and total balance in all banks. If the customer has no account, leave it null.

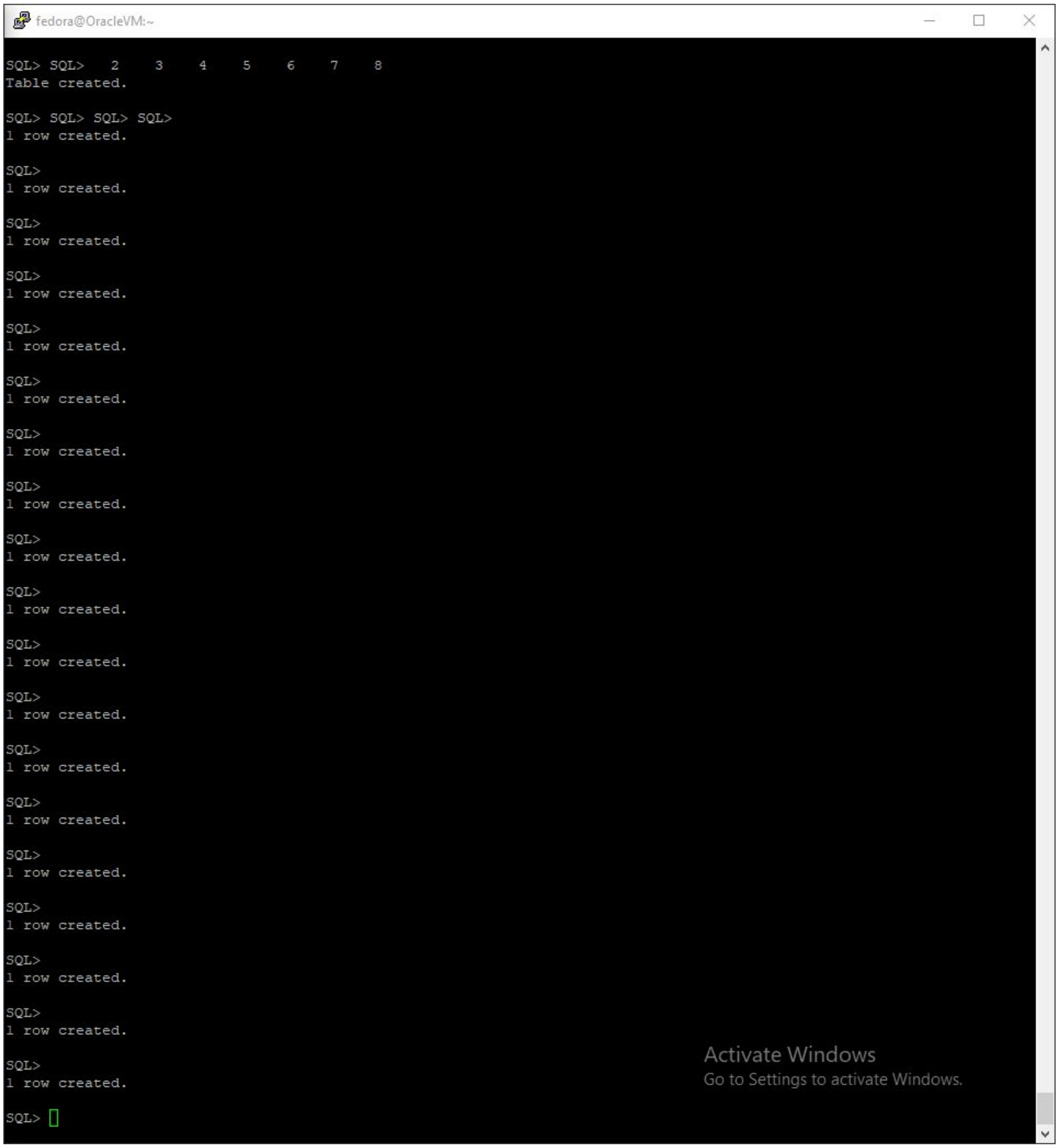
Part 1)

```
CREATE TABLE Bank
(B#      char (2),
 Name   char (8),
 City    char (8),
 PRIMARY KEY (B#),
CHECK( B# in ('B1','B2','B3','B4')),
CHECK(Name in ('England', 'America','Royal','France')),
CHECK(City in ('London', 'New York', 'Toronto', 'Paris')));

CREATE TABLE Customer
(C#      char (2),
 Name   char (8),
 Age   number (2),
 City    char (7),
 PRIMARY KEY (C#),
CHECK( C# in ('C1','C2','C3','C4','C5')),
CHECK (Name in ('Adams', 'Blake', 'Clark', 'Wojcicki', 'Smith')),
CHECK (Age between 0 and 99),
CHECK(City in ('London', 'Paris', 'Ottawa', 'Toronto')));

CREATE TABLE Account
(C#      char (2),
 B#      char (2),
 Balance     number (4),
 PRIMARY KEY (C#, B#),
 FOREIGN KEY (C#) REFERENCES Customer (C#) ON DELETE CASCADE,
 FOREIGN KEY (B#) REFERENCES Bank (B#) ON DELETE CASCADE,
 CHECK (Balance between 0 and 9999));
```

```
INSERT INTO Bank VALUES('B1', 'England', 'London');
INSERT INTO Bank VALUES('B2', 'America', 'New York');
INSERT INTO Bank VALUES('B3', 'Royal', 'Toronto');
INSERT INTO Bank VALUES('B4', 'France', 'Paris');
INSERT INTO Customer VALUES('C1','Adams',20,'London');
INSERT INTO Customer VALUES('C2','Blake',30,'Paris');
INSERT INTO Customer VALUES('C3','Clark',25,'Paris');
INSERT INTO Customer VALUES('C4','Wojcicki',20,'London');
INSERT INTO Customer VALUES('C5','Smith',30,'Toronto');
INSERT INTO Account VALUES('C1','B1',1000);
INSERT INTO Account VALUES('C1','B2',2000);
INSERT INTO Account VALUES('C1','B3',3000);
INSERT INTO Account VALUES('C1','B4',4000);
INSERT INTO Account VALUES('C2','B1',2000);
INSERT INTO Account VALUES('C2','B2',3000);
INSERT INTO Account VALUES('C3','B2',3000);
INSERT INTO Account VALUES('C3','B3',4000);
INSERT INTO Account VALUES('C4','B2',4000);
INSERT INTO Account VALUES('C4','B3',5000);
```

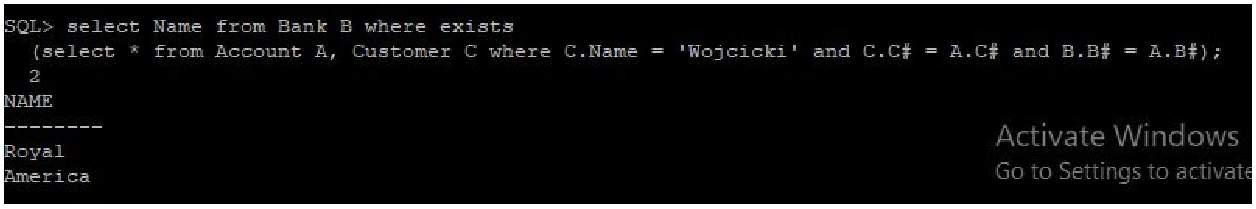


```
CREATE TABLE Bank
(B#      char (2),
 Name   char (8),
 City    char (8),
 PRIMARY KEY (B#),
CHECK( B# in ('B1','B2','B3','B4')),
CHECK(Name in ('England', 'America','Royal','France')),
CHECK(City in ('London', 'New York', 'Toronto', 'Paris')));

CREATE TABLE Customer
(C#      char (2),
 Name   char (8),
 Age   number (2),
 City    char (7),
 PRIMARY KEY (C#),
CHECK( C# in ('C1','C2','C3','C4','C5')),
CHECK (Name in ('Adams', 'Blake', 'Clark', 'Wojcicki', 'Smith')),
CHECK (Age between 0 and 99),
CHECK(City in ('London', 'Paris', 'Ottawa', 'Toronto')));

CREATE TABLE Account
(C#      char (2),
 B#      char (2),
 Balance     number (4),
 PRIMARY KEY (C#, B#),
 FOREIGN KEY (C#) REFERENCES Customer (C#) ON DELETE CASCADE,
 FOREIGN KEY (B#) REFERENCES Bank (B#) ON DELETE CASCADE,
 CHECK (Balance between 0 and 9999));



INSERT INTO Bank VALUES('B1', 'England', 'London');
INSERT INTO Bank VALUES('B2', 'America', 'New York');
INSERT INTO Bank VALUES('B3', 'Royal', 'Toronto');
INSERT INTO Bank VALUES('B4', 'France', 'Paris');
INSERT INTO Customer VALUES('C1','Adams',20,'London');
INSERT INTO Customer VALUES('C2','Blake',30,'Paris');
INSERT INTO Customer VALUES('C3','Clark',25,'Paris');
INSERT INTO Customer VALUES('C4','Wojcicki',20,'London');
INSERT INTO Customer VALUES('C5','Smith',30,'Toronto');
INSERT INTO Account VALUES('C1','B1',1000);
INSERT INTO Account VALUES('C1','B2',2000);
INSERT INTO Account VALUES('C1','B3',3000);
INSERT INTO Account VALUES('C1','B4',4000);
INSERT INTO Account VALUES('C2','B1',2000);
INSERT INTO Account VALUES('C2','B2',3000);
INSERT INTO Account VALUES('C3','B2',3000);
INSERT INTO Account VALUES('C3','B3',4000);
INSERT INTO Account VALUES('C4','B2',4000);
INSERT INTO Account VALUES('C4','B3',5000);
SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5    6    7    8    9   10
Table created.

SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL> SQL> SQL>
1 row created.
```



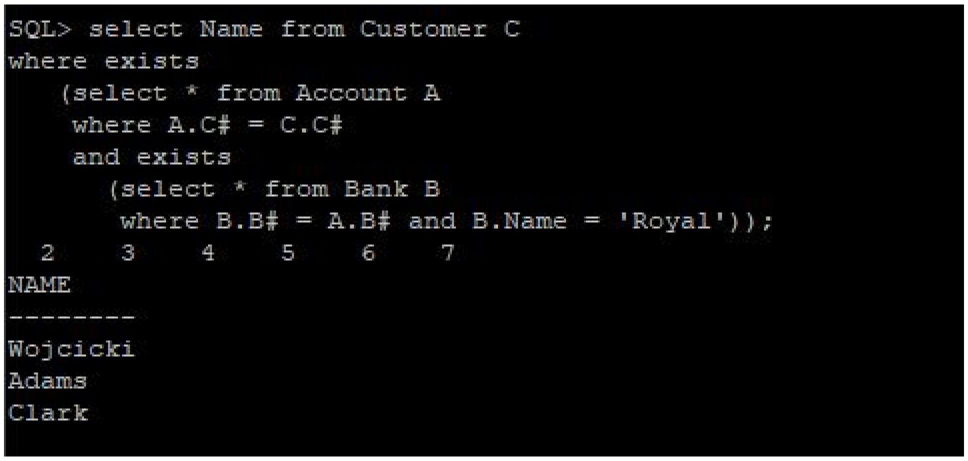
```
SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL> SQL> SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
```

1) select Name from Bank B where exists
  (select * from Account A, Customer C where C.Name = 'Wojcicki' and C.C# = A.C# and B.B# = A.B#);



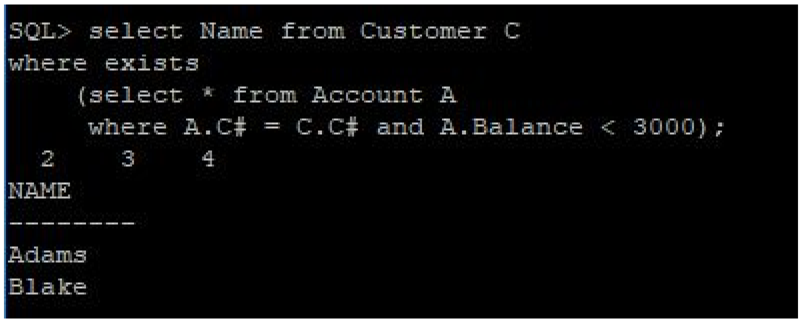
```
SQL> select Name from Bank B where exists
  (select * from Account A, Customer C where C.Name = 'Wojcicki' and C.C# = A.C# and B.B# = A.B#);
  2
NAME
--------
Royal
America
```

2) select Name from Customer C
where exists
  (select * from Account A
   where A.C# = C.C#
   and exists
     (select * from Bank B
      where B.B# = A.B# and B.Name = 'Royal'));



```
SQL> select Name from Customer C
where exists
   (select * from Account A
    where A.C# = C.C#
    and exists
      (select * from Bank B
       where B.B# = A.B# and B.Name = 'Royal'));
  2    3    4    5    6    7
NAME
--------
Wojcicki
Adams
Clark
```

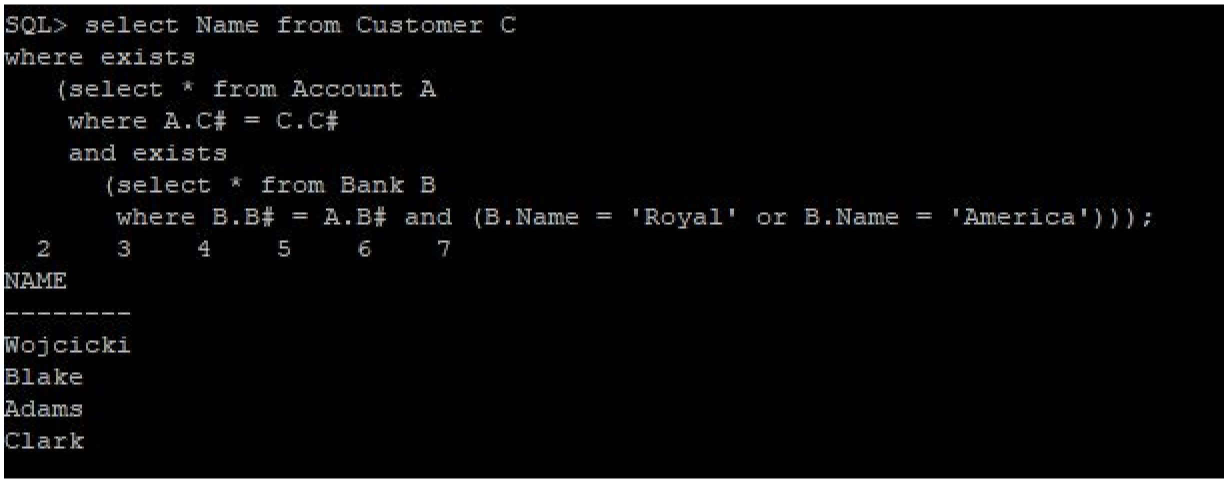
3) select Name from Customer C
where exists
   (select * from Account A
    where A.C# = C.C# and A.Balance < 3000);



```
SQL> select Name from Customer C
where exists
     (select * from Account A
      where A.C# = C.C# and A.Balance < 3000);
  2    3    4
NAME
--------
Adams
Blake
```

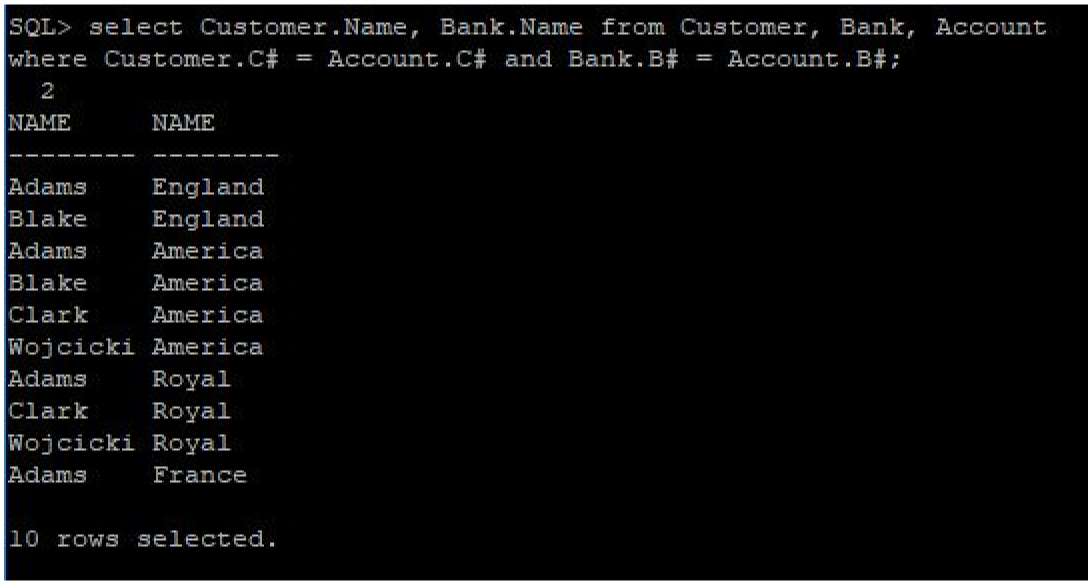
4) select Name from Customer C
where exists
  (select * from Account A
   where A.C# = C.C#
   and exists
     (select * from Bank B

        where B.B# = A.B# and (B.Name = 'Royal' or B.Name = 'America')));



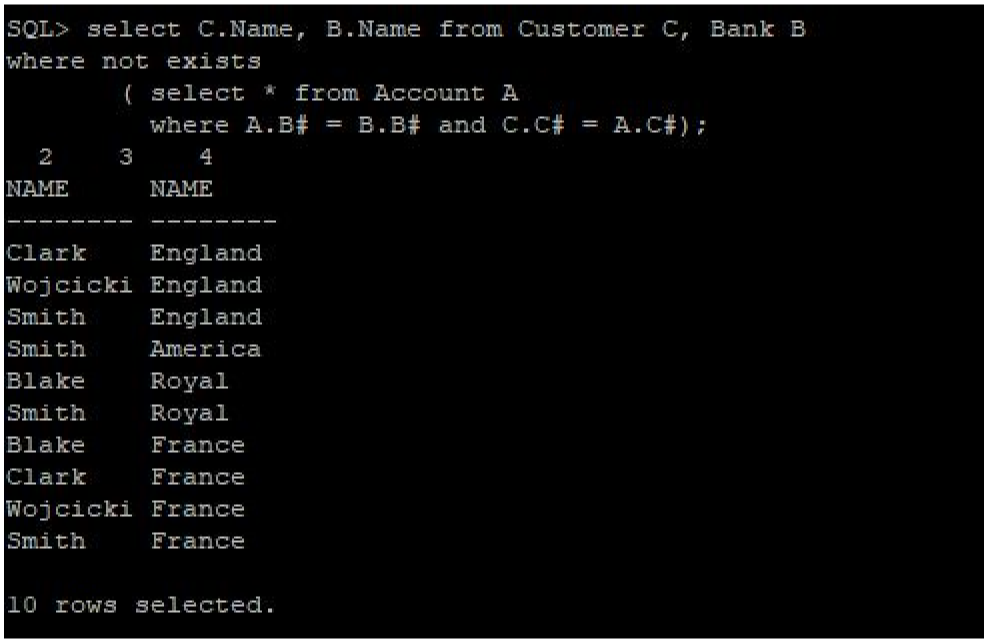
```
SQL> select Name from Customer C
where exists
   (select * from Account A
    where A.C# = C.C#
    and exists
      (select * from Bank B
       where B.B# = A.B# and (B.Name = 'Royal' or B.Name = 'America')));
  2    3    4    5    6    7
NAME
--------
Wojcicki
Blake
Adams
Clark
```

5) select Customer.Name, Bank.Name from Customer, Bank, Account
where Customer.C# = Account.C# and Bank.B# = Account.B#;



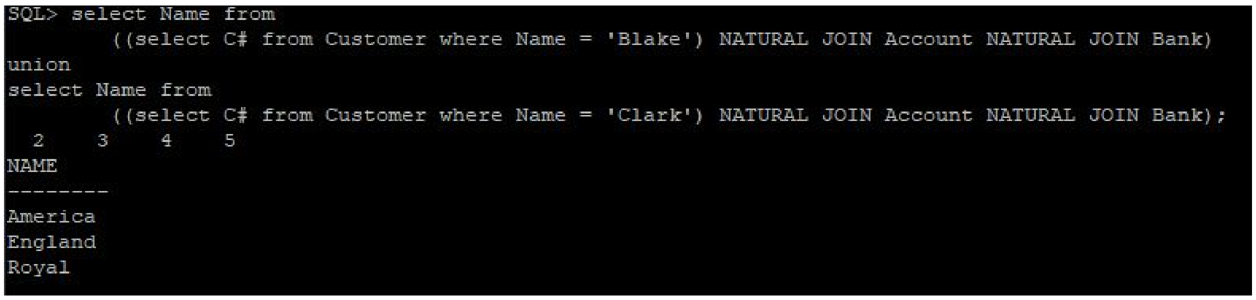
```
SQL> select Customer.Name, Bank.Name from Customer, Bank, Account
where Customer.C# = Account.C# and Bank.B# = Account.B#;
  2
NAME     NAME
-------- --------
Adams    England
Blake    England
Adams    America
Blake    America
Clark    America
Wojcicki America
Adams    Royal
Clark    Royal
Wojcicki Royal
Adams    France

10 rows selected.
```

6) select C.Name, B.Name from Customer C, Bank B
where not exists
        (select * from Account A
          where A.B# = B.B# and C.C# = A.C#);



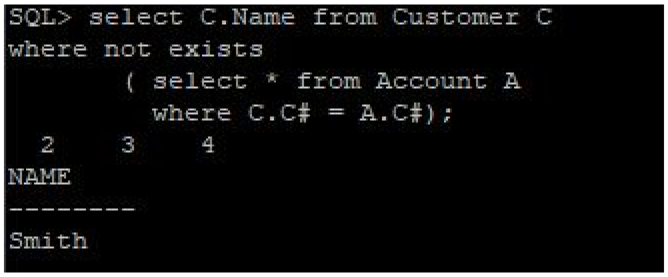
```
SQL> select C.Name, B.Name from Customer C, Bank B
where not exists
       ( select * from Account A
         where A.B# = B.B# and C.C# = A.C#);
  2    3    4
NAME     NAME
-------- --------
Clark    England
Wojcicki England
Smith    England
Smith    America
Blake    Royal
Smith    Royal
Blake    France
Clark    France
Wojcicki France
Smith    France

10 rows selected.
```

7) select Name from
((select C# from Customer where Name = 'Blake') NATURAL JOIN Account NATURAL JOIN Bank)
union
select Name from
((select C# from Customer where Name = 'Clark') NATURAL JOIN Account NATURAL JOIN Bank);



```
SQL> select Name from
        ((select C# from Customer where Name = 'Blake') NATURAL JOIN Account NATURAL JOIN Bank)
union
select Name from
        ((select C# from Customer where Name = 'Clark') NATURAL JOIN Account NATURAL JOIN Bank);
  2    3    4    5
NAME
--------
America
England
Royal
```

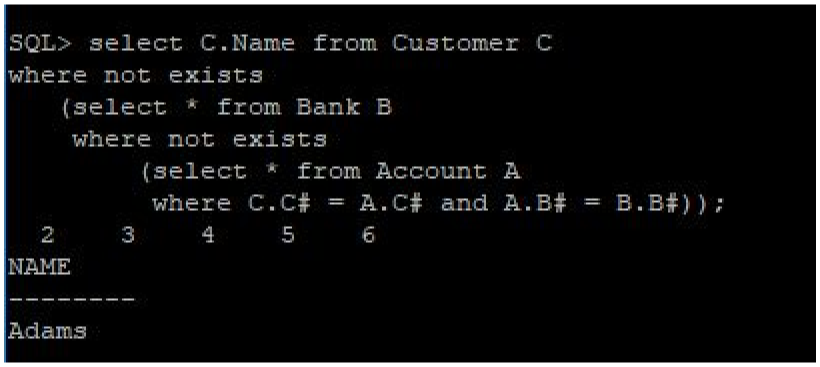
8) select C.Name from Customer C
where not exists
(select * from Account A
where C.C# = A.C#);



```
SQL> select C.Name from Customer C
where not exists
       ( select * from Account A
         where C.C# = A.C#);
  2    3    4
NAME
--------
Smith
```

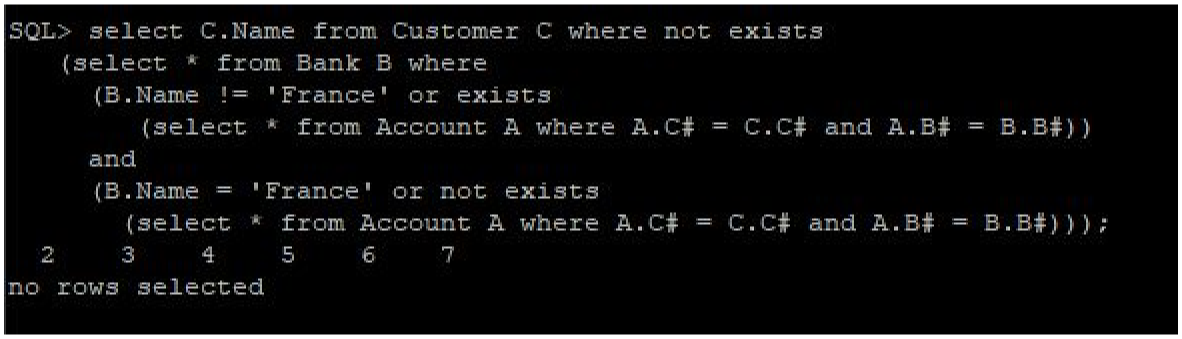
9) select C.Name from Customer C

```
where not exists
  (select * from Bank B
   where not exists
       (select * from Account A
        where C.C# = A.C# and A.B# = B.B#));
```



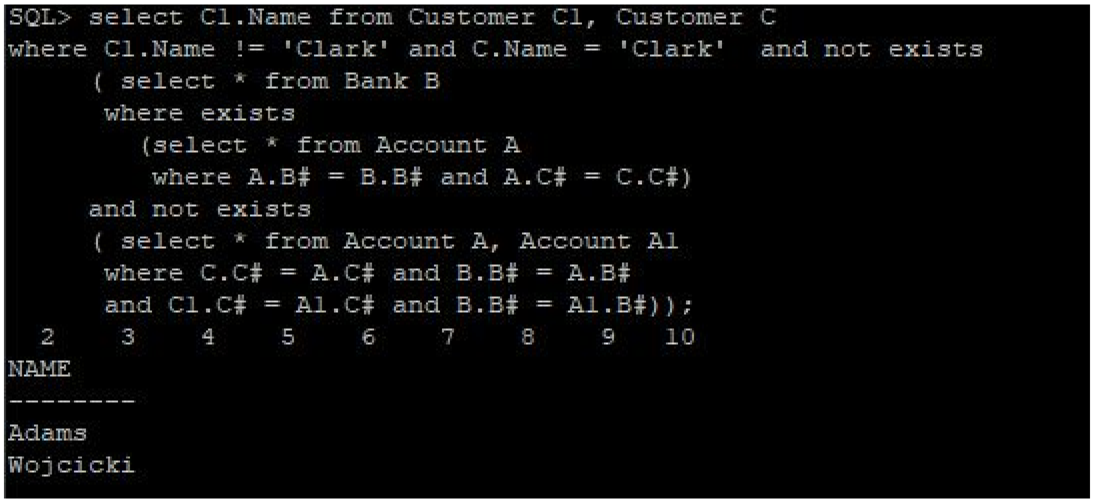
```
SQL> select C.Name from Customer C
where not exists
   (select * from Bank B
    where not exists
        (select * from Account A
         where C.C# = A.C# and A.B# = B.B#));
  2    3    4    5    6
NAME
--------
Adams
```

```
10) select C.Name from Customer C where not exists
  (select * from Bank B where
    (B.Name != 'France' or exists
       (select * from Account A where A.C# = C.C# and A.B# = B.B#))
    and
    (B.Name = 'France' or not exists
      (select * from Account A where A.C# = C.C# and A.B# = B.B#)));
```



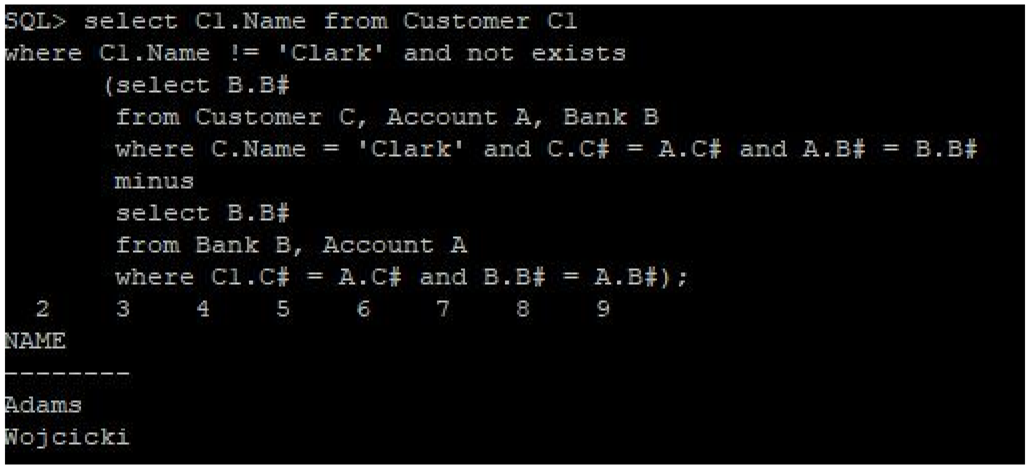
```
SQL> select C.Name from Customer C where not exists
   (select * from Bank B where
     (B.Name != 'France' or exists
        (select * from Account A where A.C# = C.C# and A.B# = B.B#))
     and
     (B.Name = 'France' or not exists
       (select * from Account A where A.C# = C.C# and A.B# = B.B#)));
  2    3    4    5    6    7
no rows selected
```

```
11) select C1.Name from Customer C1, Customer C
where C1.Name != 'Clark' and C.Name = 'Clark'  and not exists
     ( select * from Bank B
      where exists
        (select * from Account A
         where A.B# = B.B# and A.C# = C.C#)
     and not exists
     ( select * from Account A, Account A1
      where C.C# = A.C# and B.B# = A.B#
      and C1.C# = A1.C# and B.B# = A1.B#));
```



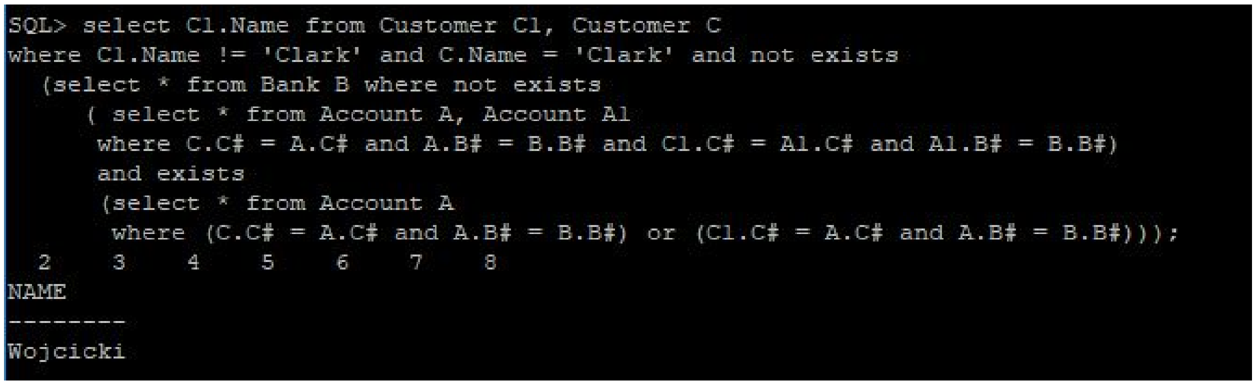
```
SQL> select C1.Name from Customer C1, Customer C
where C1.Name != 'Clark' and C.Name = 'Clark'  and not exists
     ( select * from Bank B
      where exists
        (select * from Account A
         where A.B# = B.B# and A.C# = C.C#)
     and not exists
     ( select * from Account A, Account A1
      where C.C# = A.C# and B.B# = A.B#
      and C1.C# = A1.C# and B.B# = A1.B#));
  2    3    4    5    6    7    8    9   10
NAME
--------
Adams
Wojcicki
```

12) select C1.Name from Customer C1
where C1.Name != 'Clark' and not exists
      (select B.B#
       from Customer C, Account A, Bank B
       where C.Name = 'Clark' and C.C# = A.C# and A.B# = B.B#
       minus
       select B.B#
       from Bank B, Account A
       where C1.C# = A.C# and B.B# = A.B#);



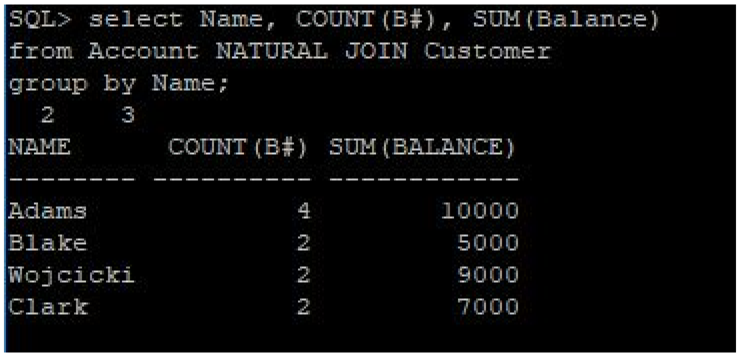
```
SQL> select C1.Name from Customer C1
where C1.Name != 'Clark' and not exists
      (select B.B#
       from Customer C, Account A, Bank B
       where C.Name = 'Clark' and C.C# = A.C# and A.B# = B.B#
       minus
       select B.B#
       from Bank B, Account A
       where C1.C# = A.C# and B.B# = A.B#);
  2    3    4    5    6    7    8    9
NAME
--------
Adams
Wojcicki
```

13) select C1.Name from Customer C1, Customer C
where C1.Name != 'Clark' and C.Name = 'Clark' and not exists
  (select * from Bank B where not exists
    (select * from Account A, Account A1
     where C.C# = A.C# and A.B# = B.B# and C1.C# = A1.C# and A1.B# = B.B#)
     and exists
     (select * from Account A
      where (C.C# = A.C# and A.B# = B.B#) or (C1.C# = A.C# and A.B# = B.B#)));



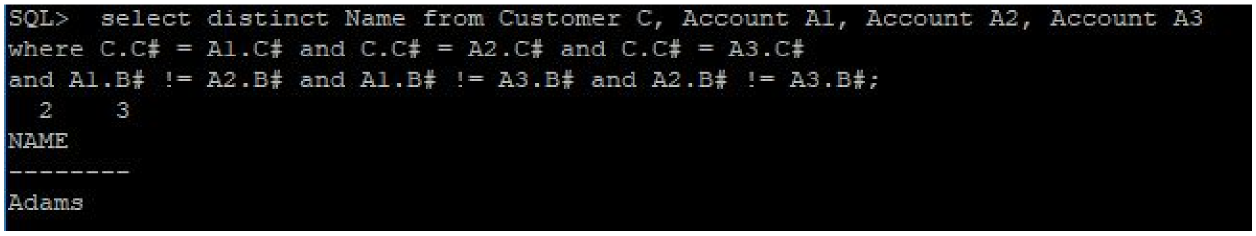
```
SQL> select C1.Name from Customer C1, Customer C
where C1.Name != 'Clark' and C.Name = 'Clark' and not exists
  (select * from Bank B where not exists
     ( select * from Account A, Account A1
      where C.C# = A.C# and A.B# = B.B# and C1.C# = A1.C# and A1.B# = B.B#)
      and exists
      (select * from Account A
       where (C.C# = A.C# and A.B# = B.B#) or (C1.C# = A.C# and A.B# = B.B#)));
  2    3    4    5    6    7    8
NAME
--------
Wojcicki
```

14) select Name, COUNT(B#), SUM(Balance)
from Account NATURAL JOIN Customer
group by Name;



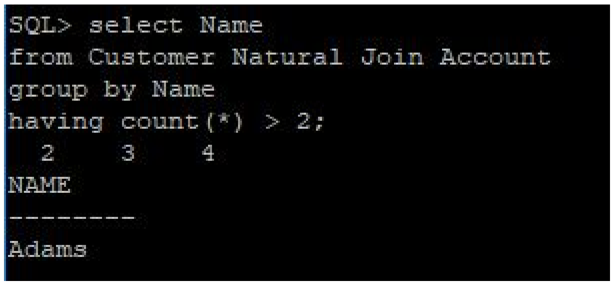
```
SQL> select Name, COUNT(B#), SUM(Balance)
from Account NATURAL JOIN Customer
group by Name;
  2    3
NAME      COUNT(B#) SUM(BALANCE)
-------- ---------- ------------
Adams             4        10000
Blake             2         5000
Wojcicki          2         9000
Clark             2         7000
```

15) select distinct Name from Customer C, Account A1, Account A2, Account A3
where C.C# = A1.C# and C.C# = A2.C# and C.C# = A3.C#
and A1.B# != A2.B# and A1.B# != A3.B# and A2.B# != A3.B#;



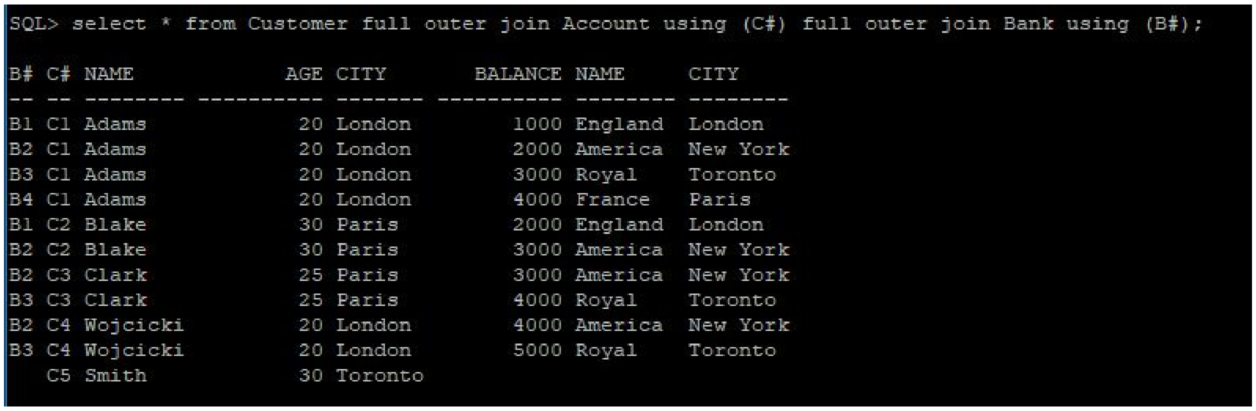
```
SQL>  select distinct Name from Customer C, Account A1, Account A2, Account A3
where C.C# = A1.C# and C.C# = A2.C# and C.C# = A3.C#
and A1.B# != A2.B# and A1.B# != A3.B# and A2.B# != A3.B#;
  2    3
NAME
--------
Adams
```

16) select Name
from Customer Natural Join Account
group by Name
having count(*) > 2;



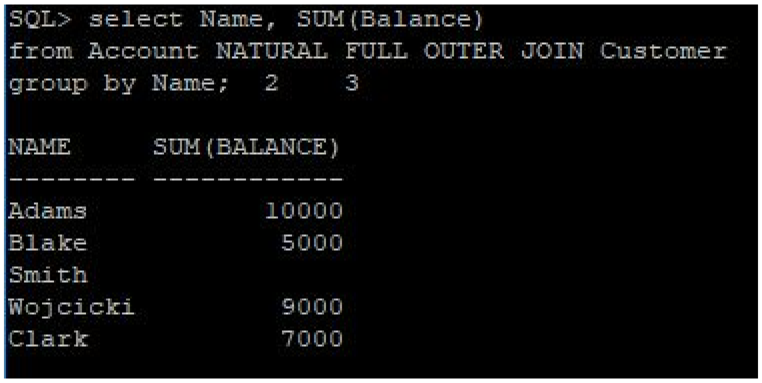
```
SQL> select Name
from Customer Natural Join Account
group by Name
having count(*) > 2;
  2    3    4
NAME
--------
Adams
```

17) select * from Customer full outer join Account using (C#) full outer join Bank using (B#);



```
SQL> select * from Customer full outer join Account using (C#) full outer join Bank using (B#);

B# C# NAME            AGE CITY       BALANCE NAME     CITY
-- -- -------- ---------- ------- ---------- -------- --------
B1 C1 Adams            20 London        1000 England  London
B2 C1 Adams            20 London        2000 America  New York
B3 C1 Adams            20 London        3000 Royal    Toronto
B4 C1 Adams            20 London        4000 France   Paris
B1 C2 Blake            30 Paris         2000 England  London
B2 C2 Blake            30 Paris         3000 America  New York
B2 C3 Clark            25 Paris         3000 America  New York
B3 C3 Clark            25 Paris         4000 Royal    Toronto
B2 C4 Wojcicki         20 London        4000 America  New York
B3 C4 Wojcicki         20 London        5000 Royal    Toronto
   C5 Smith            30 Toronto
```

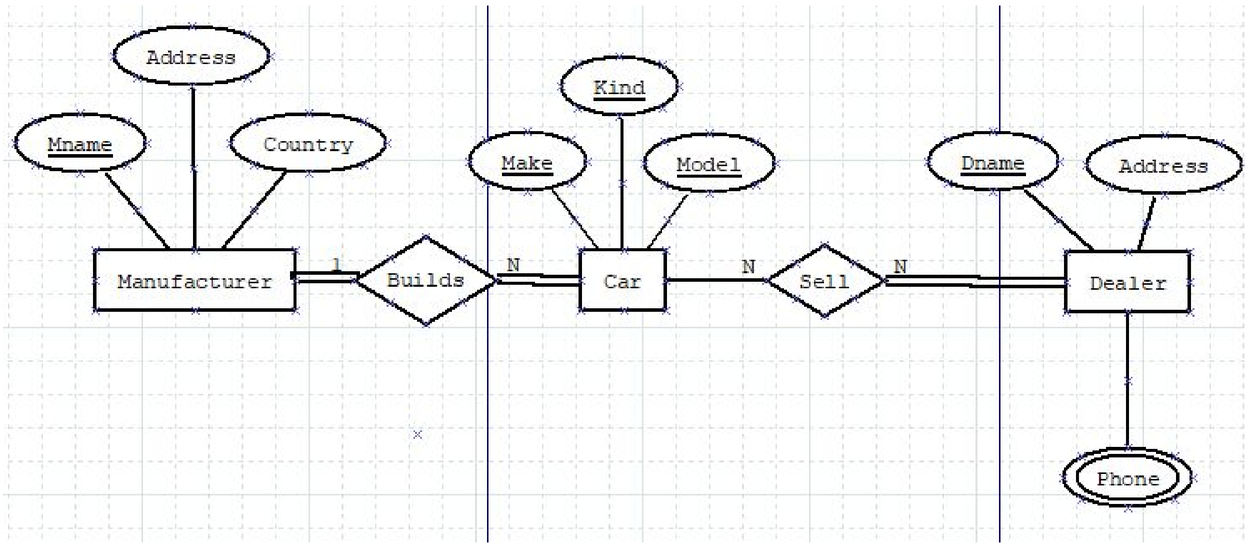
18) select Name, SUM(Balance)
from Account NATURAL FULL OUTER JOIN Customer
group by Name;



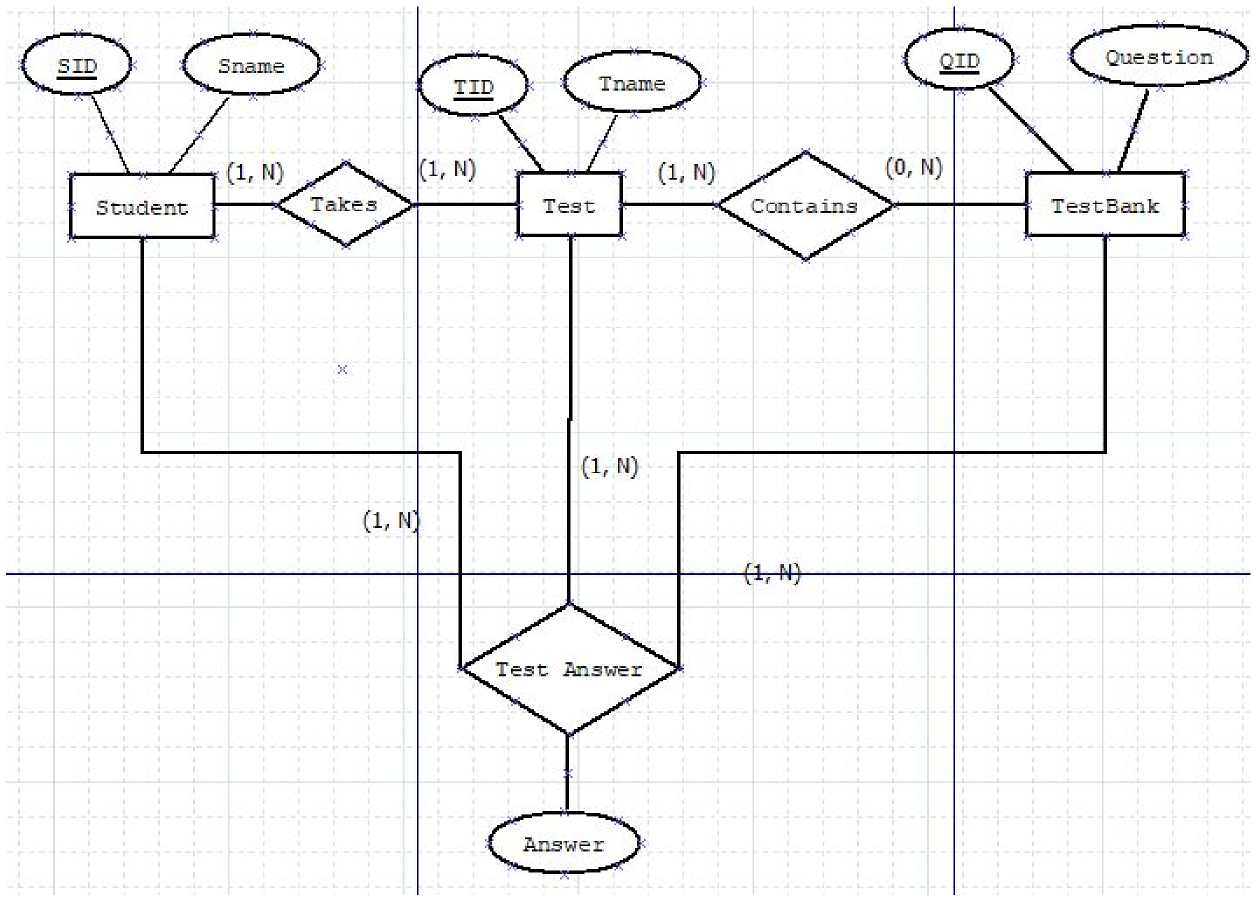
```
SQL> select Name, SUM(Balance)
from Account NATURAL FULL OUTER JOIN Customer
group by Name;  2    3

NAME     SUM(BALANCE)
-------- ------------
Adams           10000
Blake            5000
Smith
Wojcicki         9000
Clark            7000
```

Part 2)
1) Automotive enterprises involve cars, car manufacturers, and car dealerships. Car Manufacturers are companies that build cars and they have attributes name, headquarters address, country of incorporation. Cars have attributes such as make (e.g., GM, Mercedes, Chrysler), model, kind of car (e.g., sedan, SUV, wagon). A dealership sells cars. It has a name, address, and telephone numbers (typically more than one.) A manufacturer may make several different kinds and models of cars (as, for example, the manufacturer Mercedes/Chrysler does).

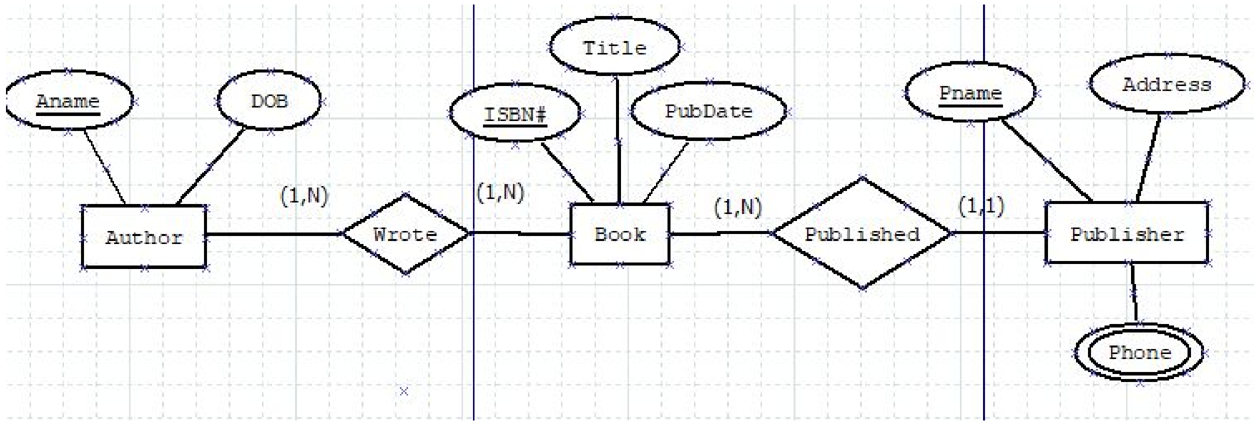
A car is made by a single manufacturer. A dealer can sell cars from several different manufacturers but does not have to sell all the cars from a single manufacturer. For example, Dilawri Auto Group in Ottawa sells Pontiacs from manufacturer GM and Hondas from manufacturer Honda, but it doesn't sell Chevrolets that are also made by GM. A manufacturer can make a car (such as a racing car) that is not sold through dealerships. Design an ER diagram that describes this database application with all relevant constraints represented using Method 1. (10 marks)



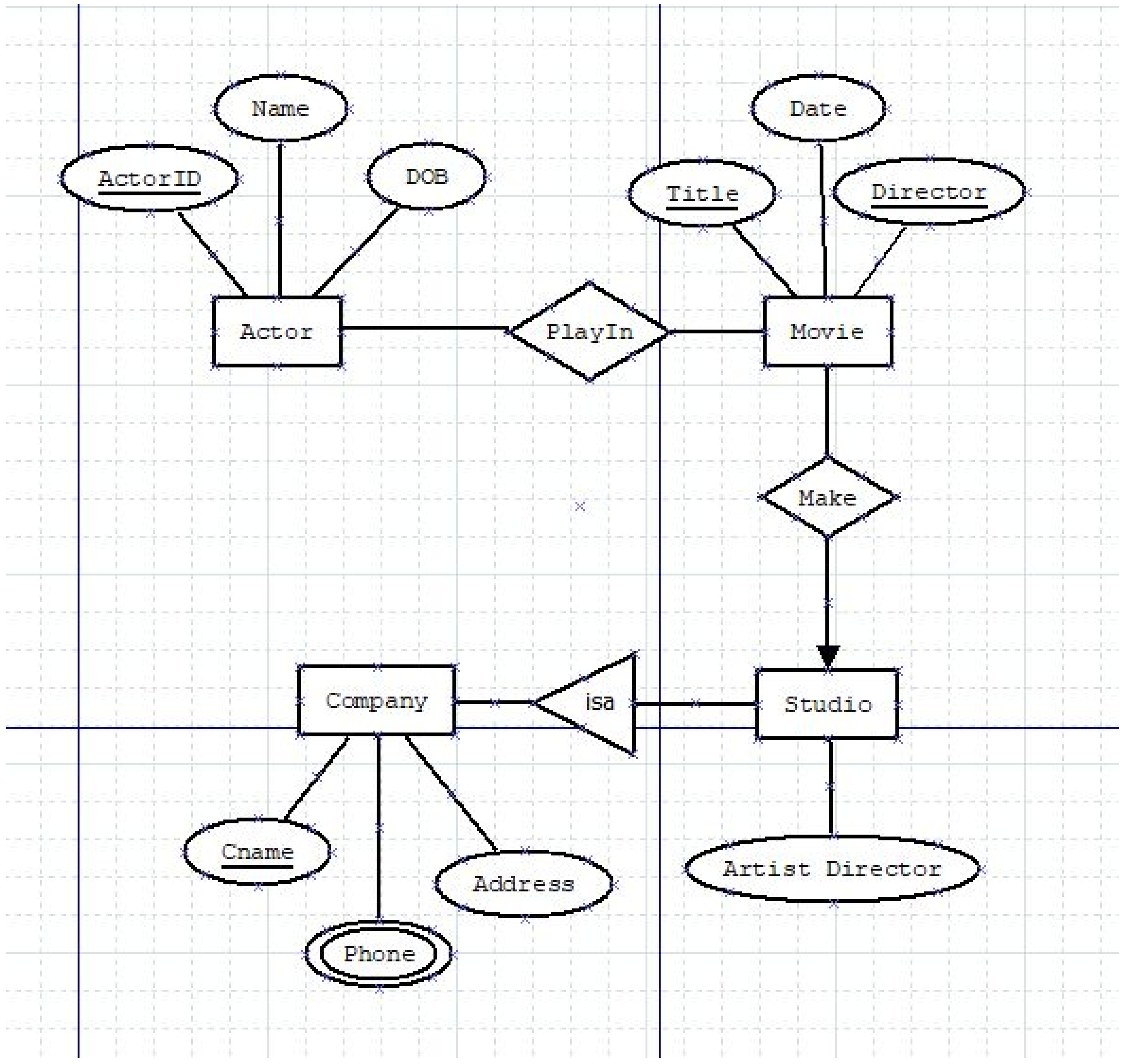
2) Test centers involve students, tests and test banks. Students have attributes student id and student name. Tests have test id and test name. Test banks have question id and questions. Students can take one or more tests and have answers to the questions in the tests. Design an ER diagram that describes this database application using Method 2. (10 marks)



3) Publication enterprises include books, authors and publishers. Authors are people with name and date of birth, but in addition they wrote one or more books. A book has title, ISBN, publication date. Publishers are companies that publish books. They have an address, phone numbers (typically more than one), and name. A book can be written by more than one author, but it can be published by only one publisher. An author can write more than one book and to be called an author one, of course, has to write at least one book. Design an ER diagram that describes this database application with all relevant constraints represented using Method 2. (10 marks)



4) Movie enterprises include movies, actors, and studios that produce movies. Actors are people with normal attributes, like Id, name, date of birth. Actors play in movies. A movie has the usual attributes: title, release date, director. Studios are companies. A company has an address, phone numbers (typically more than one), name. Studios have additional attributes, such as the artistic director. A movie has at least one actor, and exactly one studio makes each particular movie. Every actor played in at least one movie. Some studios may be brand new and had no time to make any movies yet. Design an EER diagram that describes this database application with all relevant constraints represented using Method 3. (10 marks)


Name
ActorID
DOB
Actor
PlayIn
Date
Title
Director
Movie
Make
Company
isa
Studio
Cname
Phone
Address
Artist Director