COMP 3005
Assignment #2
Due: February 5@11:59PM

Instruction

1. You should do the assignments independently. Copying is not allowed.
2. The assignment must be typed, completed on an individual basis, and submitted as a single Word/PDF file with your name as the filename to **brightspace**. Scanned handwritten documents *won't* be accepted. Make sure your uploaded file can be opened.
3. It is based on the database you create in the first assignment where Lastname in Customer table is your last name. If your information is not shown correctly in the result, you will get 0 mark for the assignment.
4. You should directly do your assignment on this document and name the document with your last name followed by your first name so that it is easy for TAs.
5. You need to use either Openstack or Oracle VM and ALG interface to Oracle DBMS for the ALG part of this assignment by entering the ALG query expressions, generating query results and putting the screenshots of the query together with the generated results into the assignment document. For TRC queries, we don't have the corresponding interface to Oracle DBMS so you just type the TRC queries in the assignment document in the same way as shown in the class.

Queries (100 marks)

Use both Relational Algebra (ALG) and Tuple Relational Calculus (TRC) to express the following queries based on the given Bank-Customer database. Submit your ALG and TRC query expressions for these queries as well as the final query results. Each ALG and TRC query is 4 marks and the result is 2 marks. Note that for queries 8 and 9, the TRC results are not the same as ALG you have to exclude Clark as specified in the class. Therefore, you need to provide not only ALG result but also TRC result for them.

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	Lastname	20	Ottawa
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
C2	B3	4000
C3	B1	3000
C3	B2	4000
C4	B1	4000
C4	B2	5000

1. Get the name of the bank that Lastna1me banks.

   T1 := Account njoin Customer;

   T2 := project B#(select Name = 'Beg ' (T1);

   T3 := T2 njoin Bank;

   T3;

```
ALG> t1:= account njoin customer;

Table created.

ALG> t1;

C#          B#          BALA NAME       AGE  CITY
C4          B1          4000 Beg          21 Ottawa
C2          B3          4000 Blake        30 Paris
C2          B1          2000 Blake        30 Paris
C3          B2          4000 Clark        25 Paris
C1          B2          2000 Adams        20 London
C1          B4          4000 Adams        20 London
C1          B1          1000 Adams        20 London
C1          B3          3000 Adams        20 London
C3          B1          3000 Clark        25 Paris
C2          B2          3000 Blake        30 Paris

10 rows processed.


ALG> t2:= project B#(select Name = 'Beg' (T1));
```

```
ALG> t2:= project B#(select Name = 'Beg' (T1));

Table created.

ALG> t2;

B#
B1

1 row  processed.


ALG> t3:= t2 njoins Bank;
syntax error
ALG> t2:= t3 njoin Bank;

Table T3 does not exist in database.


ALG> t3:= t2 njoin Bank;

Table created.

ALG> t3;

B#          NAME        CITY
B1          England     London

1 row  processed.
```

2. Get the name of the customer who banks in Royal bank.

   t4 := Account njoin Bank;

   t5:= project B#(select Name = 'Royal' (T4));

   t6:= t5 njoin Customer;

```
ALG> t4:= Account njoin Bank;

Table created.

ALG> t4;

C#         B#         BALA NAME       CITY
C2         B1         2000 England    London
C2         B2         3000 America    New York
C2         B3         4000 Royal      Toronto
C1         B1         1000 England    London
C3         B2         4000 America    New York
C1         B4         4000 France     Paris
C3         B1         3000 England    London
C1         B3         3000 Royal      Toronto
C4         B1         4000 England    London
C1         B2         2000 America    New York

10 rows processed.
```

```
ALG> t5:= project C#(select Name = 'Royal' (t4));

Table created.

ALG> t6:= t5 njoin Customer;

Table created.

ALG> t6;

C#         NAME       AGE  CITY
C1         Adams        20 London
C2         Blake        30 Paris

2 rows processed.
```

3. Get the name of the customer who has an account with balance less than 3000.

   t7 :=Customer njoin Account;

   t8 := project name (select Balance < 3000 (Customer njoin Account));

   t8;

```
ALG> t7 :=Customer njoin Account;

Table created.

ALG> t7;

C#         NAME       AGE  CITY       B#         BALA
C1         Adams        20 London     B2         2000
C4         Beg          21 Ottawa     B1         4000
C2         Blake        30 Paris      B2         3000
C3         Clark        25 Paris      B2         4000
C3         Clark        25 Paris      B1         3000
C1         Adams        20 London     B3         3000
C1         Adams        20 London     B1         1000
C1         Adams        20 London     B4         4000
C2         Blake        30 Paris      B1         2000
C2         Blake        30 Paris      B3         4000

10 rows processed.
```

```
ALG> t8 := project name (select Balance < 3000 (Customer njoin Account));

Table created.

ALG> t8;

NAME
Adams
Blake

2 rows processed.
```

4. Get the customer name/bank name pairs such that the indicated customer has an account in the indicated bank.

   t9(cName, Age, City) := project Name, Age, City (Customer);

   t10 := t9 njoin Bank;

```
ALG> t9(cName, Age, City) := project Name, Age, City (Customer);

Table created.

ALG> t9;

CNAME      AGE  CITY
Adams        20 London
Beg          21 Ottawa
Blake        30 Paris
Clark        25 Paris
Smith        30 Toronto

5 rows processed.
```

```
syntax error
ALG> t10 := t9 njoin Bank;

Table created.

ALG> t10;

CNAME      AGE  CITY       B#         NAME
Adams        20 London     B1         England
Blake        30 Paris      B4         France
Clark        25 Paris      B4         France
Smith        30 Toronto    B3         Royal

4 rows processed.
```

5. Get the name of the customer who does not have any bank account.

   t11 := Customer nleftjoin Account;

   t12 := Customer njoin Account;

   t13:= t11 minus t12;

   t13;

```
ALG> t11 := Customer nleftjoin Account;

Table created.

ALG> t12 := Customer njoin Account;

Table created.

ALG> t13:= t11 minus t12;

Table created.

ALG> t13;

C#         NAME       AGE  CITY       B#         BALA
C5         Smith        30 Toronto

1 row  processed.
```

6. Get the name of the customer who has an account in every bank.

   Bc acc

   B bank

   t14 := project C#, B#(Account);

   t15 := project B#(Bank);

   t16 := t14 divideby t15;

   t17 := t16 njoin Customer;

   t17;

```
ALG> t14 := project C#, B#(Account);

Table created.

ALG> t14;

C#          B#
C2          B2
C2          B1
C1          B1
C1          B4
C3          B2
C3          B1
C4          B1
C1          B2
C1          B3
C2          B3

10 rows processed.
```

```
ALG> t15 := project B#(Bank);

Table created.

ALG> t15;

B#
B3
B1
B2
B4

4 rows processed.
```

```
ALG> t16 := t14 divideby t15;

Table created.

ALG> t16;

C#
C1

1 row  processed.


ALG> t17 := t16 njoin Customer;

Table created.

ALG> t17;

C#         NAME       AGE  CITY
C1         Adams        20 London

1 row  processed.
```

7. Get the name of the customer who has an account in every bank except France Bank.

   t18 := project C#, B# (Account);

   t19 := project B#(select Name != 'France' (Bank));

   t20 := project B#(select Name = 'France' (Bank));

   t21 := t18 divideby t19;

   t22 := t18 divideby t20;

   t23 := t21 minus t22;

   t24 := t23 njoin Customer;

   t24;

```
ALG> t18 := project C#, B# (Account);

Table created.

ALG> t18;

C#         B#
C2         B2
C2         B1
C1         B1
C1         B4
C3         B2
C3         B1
C4         B1
C1         B2
C1         B3
C2         B3

10 rows processed.


ALG> t19 := project B#(select Name != 'France');
syntax error
ALG> t19 := project B#(select Name != 'France' (Bank));

Table created.

ALG> t19;

B#
B3
B1
B2

3 rows processed.


ALG> t20 n:= project B#(select Name = 'France' (Bank));
syntax error
ALG> t20 := project B#(select Name = 'France' (Bank));

Table created.

ALG> t20;

B#
B4

1 row  processed.
```

```
ALG> t21 := t18 divideby t19;

Table created.

ALG> t21;

C#
C2
C1

2 rows processed.


ALG> t22 := t18 divideby t20;

Table created.
t
ALG>t22;

C#
C1

1 row  processed.


ALG> t23 := t21 minus t22;

Table created.

ALG> t23;

C#
C2

1 row  processed.
```

```
ALG> t24 := t23 njoin Customer;

Table created.

ALG> t24;

C#         NAME       AGE  CITY
C2         Blake        30 Paris

1 row  processed.
```

8. Get the name of the customer who has an account in every bank that Clark banks.

   t25 := project B#, C#(Account);

   t26 := Bank njoin Account njoin Customer;

   t27 := project B#(select Name = ‘Clark’ (t26));

   t28 := t25 divideby t27;

   t29 := Customer njoin t28;

t29;

```
ALG> t25 := project B#, C# (Account);

Table created.

ALG> t25;

B#         C#
B1         C3
B2         C1
B3         C2
B3         C1
B4         C1
B2         C3
B1         C2
B1         C4
B1         C1
B2         C2

10 rows processed.


ALG> t26 := Bank njoin Account njoin Customer;

Table created.

ALG> t27 := project B#(select name = 'Clark' (t26));

Table created.

ALG> t28 := t25 divideby t27;

Table created.
```

```
ALG> t29 := Customer njoin t28;

Table created.

ALG> t29;

C#         NAME       AGE  CITY
C1         Adams        20 London
C3         Clark        25 Paris
C2         Blake        30 Paris
C4         Beg          21 Ottawa

4 rows processed.


ALG>
```

9. Get the name of the customer who banks only in the banks that Clark banks.

10. Get the name of the customer who banks in more than two banks.

    t30(c#, Bank) := aggregate c#, count(*)(Account);

    t31 := select Bank>2(t30);

    t32 := t31 njoin Customer;

    t33 := project name (t32);

    t33;

```
ALG> t30(c#, Bank) := aggregate c#, count(*)(Account);

Table created.

ALG> t30;

C#         BANK
C1            4
C3            2
C2            3
C4            1

4 rows processed.


ALG> t31 := select Bank>2(t30);

Table created.

ALG> t31;

C#         BANK
C1            4
C2            3

2 rows processed.

ALG> t32 := t31 njoin Customer;

Table created.

ALG> t32;

C#         BANK NAME       AGE  CITY
C2            3 Blake        30 Paris
C1            4 Adams        20 London

2 rows processed.
```

```
ALG> t33 := project name (t32);

Table created.

ALG> t33;

NAME
Adams
Blake

2 rows processed.


ALG>
```