**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](https://carleton.ca/scs/tech-support/scs-open-stack/openstack-technical-support/openstack-step-by-step-guide/) or [Oracle VM](https://git.scs.carleton.ca/downloads/CourseVirtualMachines/2022F-2023W/COMP3005-W23.ova) 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 |

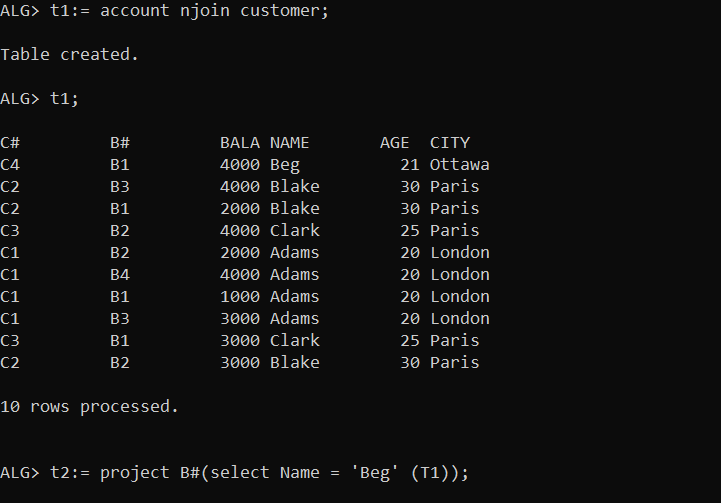
1. Get the name of the bank that Lastna1me banks.

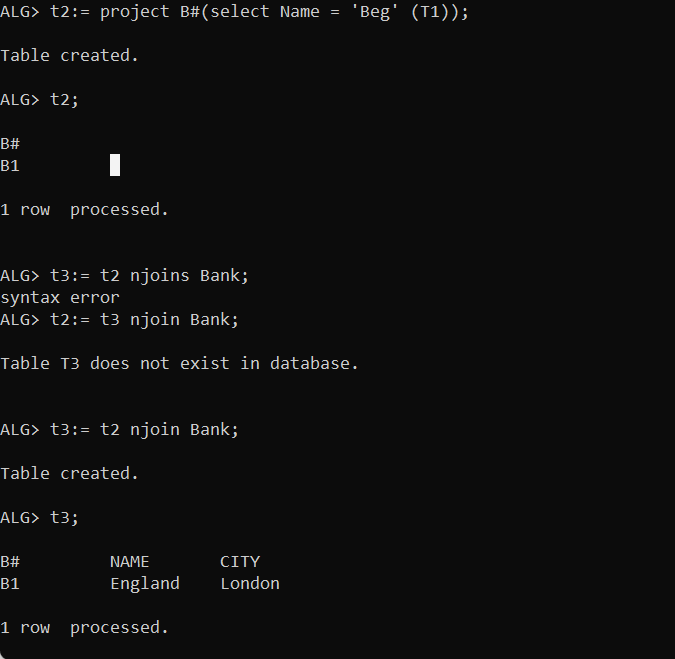
T1 := Account njoin Customer;

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

T3 := T2 njoin Bank;

T3;



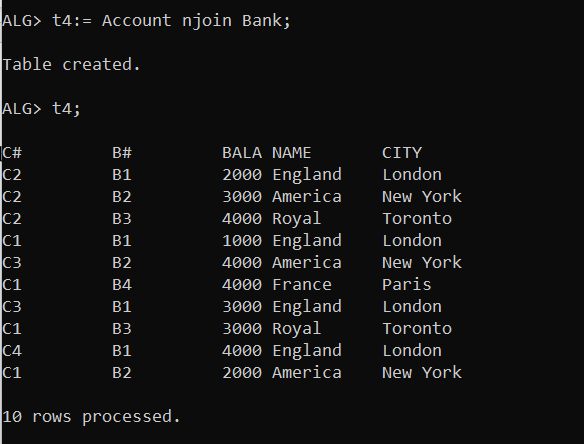


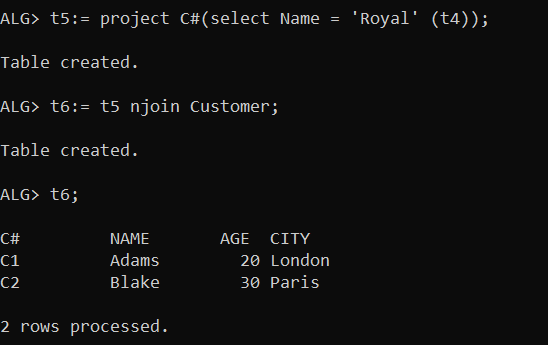
1. Get the name of the customer who banks in Royal bank.

t4 := Account njoin Bank;

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

t6:= t5 njoin Customer;



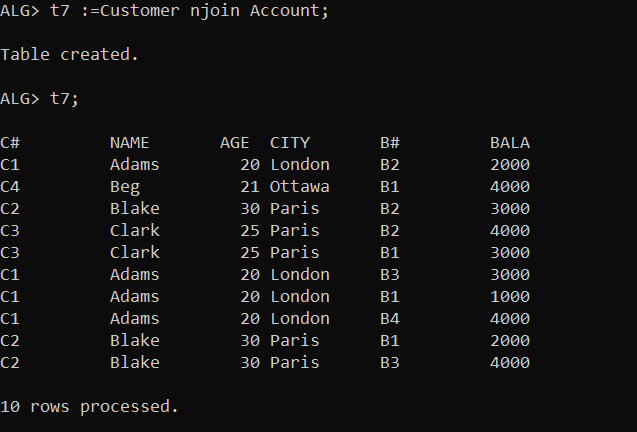


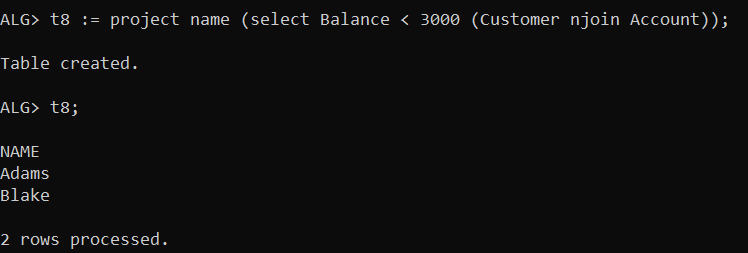
1. Get the name of the customer who has an account with balance less than 3000.

t7 :=Customer njoin Account;

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

t8;

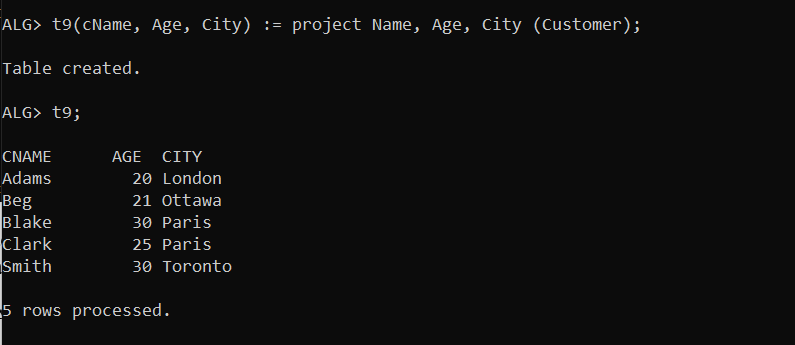


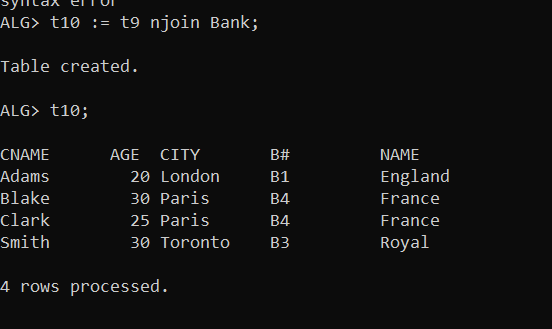


1. Get the customer name/bank name pairs such that the indicated customer has an account in the indicated bank.

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

t10 := t9 njoin Bank;





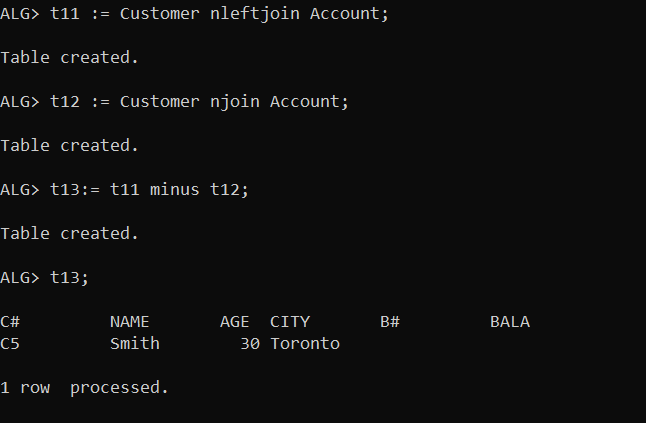
1. Get the name of the customer who does not have any bank account.

t11 := Customer nleftjoin Account;

t12 := Customer njoin Account;

t13:= t11 minus t12;

t13;



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

Bc acc

B bank

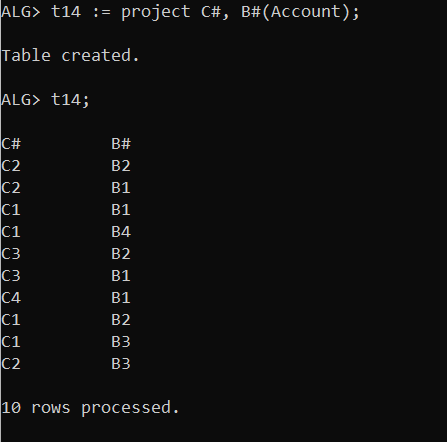
t14 := project C#, B#(Account);

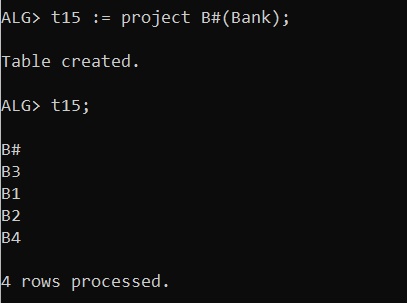
t15 := project B#(Bank);

t16 := t14 divideby t15;

t17 := t16 njoin Customer;

t17;







1. 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));

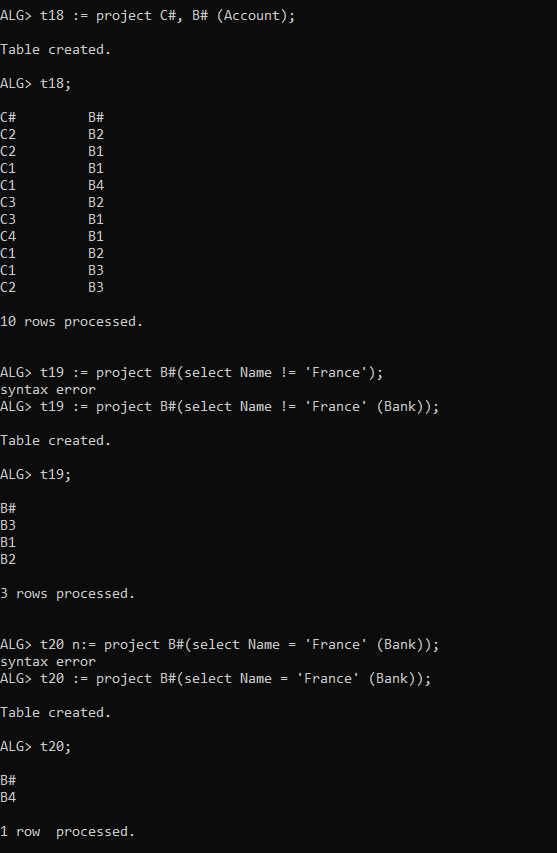
t21 := t18 divideby t19;

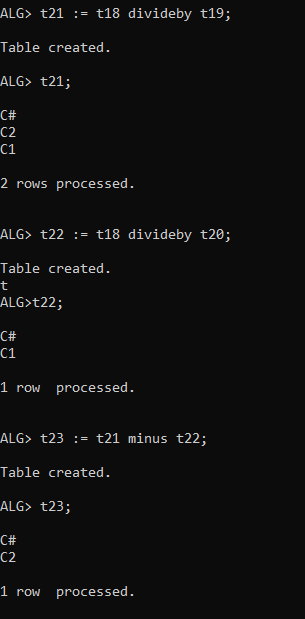
t22 := t18 divideby t20;

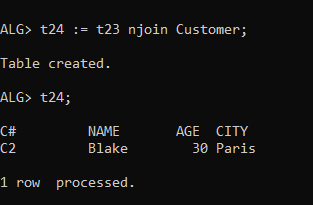
t23 := t21 minus t22;

t24 := t23 njoin Customer;

t24;







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

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

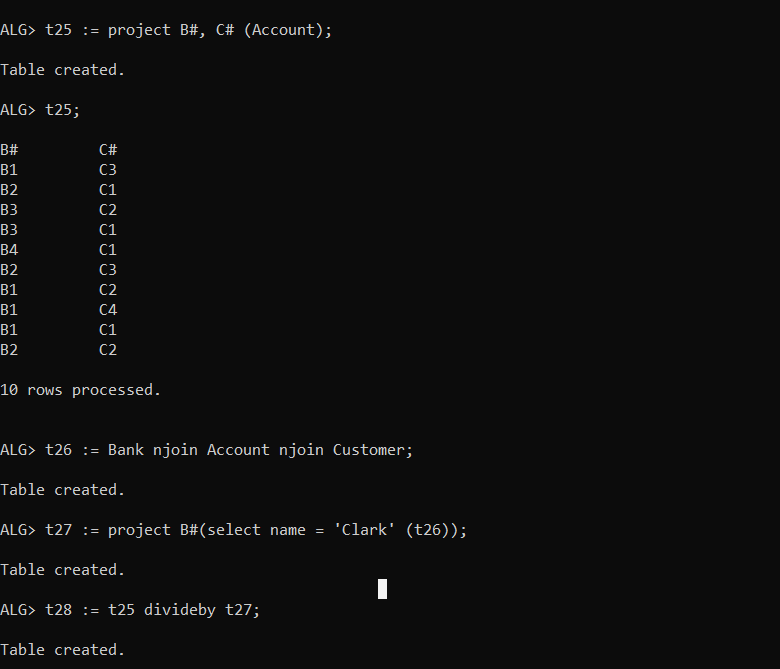
t26 := Bank njoin Account njoin Customer;

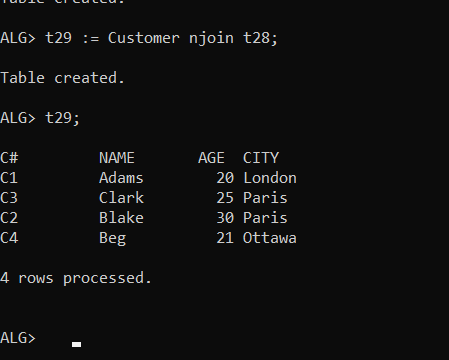
t27 := project B#( select Name = ‘Clark’ (t26));

t28 := t25 divideby t27;

t29 := Customer njoin t28;

t29;





1. Get the name of the customer who banks only in the banks that Clark banks.
2. 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;

