**COMP 3005  
Assignment #4   
Due: March 11**

**Instruction**

1. You should do the assignment 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 and contain everything required.
3. 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 to mark.
5. For Part 1, you need to use [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 SQLPLUS interface to Oracle DBMS, test each program carefully and submit the final version of the program together with several representative screenshots of the execution of the program. If there is no screenshot, you will get 0 for the question.

**Part 1 PL/SQL (50 Marks)**

This part is based on the Bank-Customer database that has three tables shown below. Note that the database is slightly different from the one in previous assignments.

**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 |

**Bank**

|  |  |  |
| --- | --- | --- |
| **B#** | **Name** | **City** |
| B1 | England | London |
| B2 | America | Chicago |
| B3 | Royal | Toronto |
| B4 | France | Paris |

**Customer**

|  |  |  |  |
| --- | --- | --- | --- |
| **C#** | **Name** | **Age** | **City** |
| C1 | Adams | 20 | London |
| C2 | Blake | 30 | Paris |
| C3 | Clark | 25 | Chicago |
| C4 | Lastname | 20 | Ottawa |
| C5 | Smith | 30 | Toronto |

1. Delete all three tables created before and then write a PL/SQL program that uses **execute immediate** statements to create and populate the three tables (20 marks)

BEGIN

EXECUTE IMMEDIATE 'CREATE TABLE Bank (B# CHAR(4), Name CHAR(10), City CHAR(10))';

EXECUTE IMMEDIATE 'CREATE TABLE Customer (C# CHAR(4), Name CHAR(10), Age INT, City CHAR(10))';

EXECUTE IMMEDIATE 'CREATE TABLE Account (C# CHAR(4), B# CHAR(4), Balance INT)';

EXECUTE IMMEDIATE 'INSERT INTO Customer VALUES (''C2'', ''Blake'', 30, ''Paris'')';

EXECUTE IMMEDIATE 'INSERT INTO Customer VALUES (''C3'', ''Clark'', 25, ''Chicago'')';

EXECUTE IMMEDIATE 'INSERT INTO Customer VALUES (''C4'', ''Beg'', 21, ''Ottawa'')';

EXECUTE IMMEDIATE 'INSERT INTO Customer VALUES (''C5'', ''Smith'', 30, ''Toronto'')';

EXECUTE IMMEDIATE 'INSERT INTO Bank VALUES (''B1'', ''England'', ''London'')';

EXECUTE IMMEDIATE 'INSERT INTO Bank VALUES (''B2'', ''America'', ''Chicago'')';

EXECUTE IMMEDIATE 'INSERT INTO Bank VALUES (''B3'', ''Royal'', ''Toronto'')';

EXECUTE IMMEDIATE 'INSERT INTO Bank VALUES (''B4'', ''France'', ''Paris'')';

EXECUTE IMMEDIATE 'INSERT INTO Customer VALUES (''C1'', ''Adams'', 20, ''London'')';

EXECUTE IMMEDIATE 'INSERT INTO Account VALUES (''C1'', ''B1'', 1000)';

EXECUTE IMMEDIATE 'INSERT INTO Account VALUES (''C1'', ''B2'', 2000)';

EXECUTE IMMEDIATE 'INSERT INTO Account VALUES (''C1'', ''B3'', 3000)';

EXECUTE IMMEDIATE 'INSERT INTO Account VALUES (''C1'', ''B4'', 4000)';

EXECUTE IMMEDIATE 'INSERT INTO Account VALUES (''C2'', ''B1'', 2000)';

EXECUTE IMMEDIATE 'INSERT INTO Account VALUES (''C2'', ''B2'', 3000)';

EXECUTE IMMEDIATE 'INSERT INTO Account VALUES (''C2'', ''B2'', 4000)';

EXECUTE IMMEDIATE 'INSERT INTO Account VALUES (''C3'', ''B1'', 3000)';

EXECUTE IMMEDIATE 'INSERT INTO Account VALUES (''C3'', ''B2'', 4000)';

EXECUTE IMMEDIATE 'INSERT INTO Account VALUES (''C4'', ''B1'', 4000)';

EXECUTE IMMEDIATE 'INSERT INTO Account VALUES (''C4'', ''B2'', 5000)';

END;

/

Text

Description automatically generated

1. Write a PL/SQL program to list all customer rows, in customer number order so that each customer row is immediately followed in the listing by all bank rows for banks that the customer has account in, in bank number order. Customers who do not bank should still be listed. (15 marks)

DECLARE

CURSOR C1 IS

SELECT C.C#, C.Name, B.B#, B.Name, A.Balance

FROM Customer C LEFT OUTER JOIN Account A

ON C.C# = A.C#

LEFT OUTER JOIN Bank B

ON A.B# = B.B#

ORDER BY C.C#, B.B#;

V\_C# CHAR(5) := NULL;

BEGIN

DBMS\_OUTPUT.ENABLE;

FOR R1 IN C1 LOOP

IF V\_C# != R1.C# THEN

V\_C# := R1.C#;

DBMS\_OUTPUT.PUT\_LINE('Customer: ' || R1.C# || ', ' || R1.Name);

END IF;

IF R1.B# IS NOT NULL THEN

DBMS\_OUTPUT.PUT\_LINE(' Bank: ' || R1.B# || ', ' || R1.Name || ', Balance: ' || R1.Balance);

END IF;

END LOOP;

END;

(no screenshots because my vm keeps crashing)

1. Redo question 2 using parameterized cursor that takes a customer name. It should first prompt the user to enter a customer name and then display the same information as in 3 just for the given customer. Use your Lastname to test this program. (15 marks)

DECLARE

-- declare variables

V\_CustomerName VARCHAR2(15);

-- declare cursor

CURSOR C1 (P\_CustomerName VARCHAR2) IS

SELECT C.C#, C.Name, B.B#, B.Name, A.Balance

FROM Customer C LEFT OUTER JOIN Account A

ON C.C# = A.C#

LEFT OUTER JOIN Bank B

ON A.B# = B.B#

WHERE C.Name = P\_CustomerName

ORDER BY C.C#, B.B#;

V\_C# CHAR(5) := NULL;

BEGIN

-- get customer name from user

V\_CustomerName := '&Enter\_Customer\_Name';

DBMS\_OUTPUT.ENABLE;

-- loop through cursor

FOR R1 IN C1(V\_CustomerName) LOOP

IF V\_C# != R1.C# THEN

V\_C# := R1.C#;

DBMS\_OUTPUT.PUT\_LINE('Customer: ' || R1.C# || ', ' || R1.Name);

END IF;

IF R1.B# IS NOT NULL THEN

DBMS\_OUTPUT.PUT\_LINE(' Bank: ' || R1.B# || ', ' || R1.Name || ', Balance: ' || R1.Balance);

END IF;

END LOOP;

END;

/

(no screenshots because my vm keeps crashing)

**Part 2 ER Model (60 marks)**

A university information system involves buildings, classrooms, offices, department, courses, sections, chairs, instructors, and students.

1. A building has a unique building number such as HP, a unique name, and a number of classrooms and offices.
2. A classroom has a room number such as 5125 that is unique in the building, the number of seats, and is either empty or used by a number of sections at different day and time.
3. An office has a room number that is unique in the building, the size in square feet, and is either empty, or occupied by a chair or up to 4 instructors.
4. A department has a unique dept code such as COMP, a unique name, 0 or 1 chair, 0 to 10 instructors, 0 to 100 students, 0 to 10 courses, 0 to 5 offices in same or different buildings and no offices are shared by different departments.
5. A course has a unique course number such as 3005 and name such as Databases that are unique in the department that offers the course, credit hours and a number of prerequisite courses. Courses are offered as sections and not all courses are offered.
6. A section has a unique section code such as A and B within the course, semester, year, classroom, day and time such as MW 11:55-12:55, TR 10:05-11:55, textbooks, and is related to one course, one instructor, and 5 to 20 students. Just consider current sections only.
7. A chair or an instructor has a unique employee number, a name, an office, 0 to 3 phone numbers, and can only work in one department. Note that a chair is not an instructor, vice versa. An instructor teaches 0-3 sections.
8. A student has a unique student number, a name, majors in one department and takes 0 to 5 sections. (and have a grade for each section).

Draw the ER diagram for this information system that can represent the constraints specified above. You can use free draw.io to do this part. (50 marks)

Diagram

Description automatically generated