**CENTENNIAL COLLEGE   
Information and Communication**

**Engineering Technology (ICET) DEPARTMENT   
FINAL EXAM EXAMINATION**

**SEMESTER: FALL 2022**

**Version 3**

**SUBJECT NAME:** [**21F --Advanced Database Concepts (SEC. 401)**](https://e.centennialcollege.ca/d2l/home/694870)

**SUBJECT CODE: COMP214 - Lab Hands on**

**EXAMINATION DATE: Dec 15, 2022**

**INSTRUCTOR NAME: ERSAN CAM**

**MARKS ALLOTTED: 30 out of 100 (3 out of 10)**

**WEIGHTING: Overall 15% from both Final Theory Quiz and Lab test totals**

**Instructions:**

1. **Add your name on this Word File when you are submitting.**
2. **Always include copy paste version of your code and as well as Screen shot of code execution.**
3. **\* To detect Plagiarism, instructor has all the rights to use some applications (TurnItIn). If any cloned&copied work among classmates is being detected then students that they involved are subject to Centennial College’s Code of Academic Conduct document located below.**
4. **\* Please refer respected note about Plagiarism and Academic integrity from our Course Outline**

<https://library.centennialcollege.ca/help-services/research-help/academic-integrity/>

**Deepa K**

**Student Number:301250548**

**Question 1: (15 points (1.5 out of 10) )**

**Use Brewbean database for this question.**

**Processing Database Data with IF Statements**

The Brewbean’s application needs a block to determine whether a customer is rated HIGH,

MID, or LOW based on his or her total purchases. The block needs to select the total amount of

orders for a specified customer, determine the rating, and then display the results onscreen. The

code rates the customer HIGH if total purchases are greater than $200, MID if greater than

$100, and LOW if $100 or lower. Use an initialized variable to provide the shopper ID.

Here is the partial code given to you. This is your start point.

*DECLARE*

*lv\_total\_num NUMBER(6,2);*

*……….*

*…..*

*BEGIN*

*SELECT SUM(total)*

*FROM bb\_basket*

*WHERE idShopper =*

*AND orderplaced = 1*

*GROUP BY idshopper;*

*IF lv\_total\_num > 200 THEN*

*END IF;*

*DBMS\_OUTPUT.PUT\_LINE(……………….);*

*END;*

1.Edit the block to perform the required task. Add IF THEN ELSE section

2. Run the block and verify the results. Enter and run the following SQL query to confirm that

the total for this shopper is indeed greater than $200:

**SELECT SUM(total)**

**FROM bb\_basket**

**WHERE idShopper = 22**

**AND orderplaced = 1**

**GROUP BY idshopper;**

DECLARE

v\_shop\_rating varchar2(15);

v\_total\_num number(8,2);

v\_shopr\_ID number(5):=22;

BEGIN

SELECT SUM(total)

INTO v\_total\_num

FROM bb\_basket

WHERE IDSHOPPER = v\_shopr\_ID

AND orderplaced = 1

GROUP BY IDSHOPPER;

IF v\_total\_num > 200 THEN v\_shop\_rating := 'HIGH';

ELSIF v\_total\_num > 100 AND v\_total\_num <= 200 THEN v\_shop\_rating := 'MID';

ELSIF v\_total\_num <= 100 THEN v\_shop\_rating := 'LOW';

END IF;

DBMS\_OUTPUT.PUT\_LINE('Total Purchase is:' ||' '|| v\_total\_num ||' '|| 'The customer Rating is:' ||' '|| v\_shop\_rating);

END;

**Graphical user interface, text, application, email

Description automatically generated**

**Question 2) Work on Human Resources**

Create a stored function to find out number of months of service for a specific employee.

Function name is CALCULATE\_MONTHS\_SERVICE to return the total number of months of service for employee. Function will accept employee\_id as input and return number of month of service. Add error handling to account for an invalid emoployee ID in case this employee doesn’t exists or can not be found.

**Instructions :**

In your function you need to find total number of month in Service. To find that you need to find all the years for that employee in historical jobs (**job\_history** table) and plus add his/her current employment duration to the calculation to find overall years in service (from employees table)

**Step1:**Use Cursor to get all the records from job\_history for that particular employee\_id and then count years .

Step2:Then later on add numbers of years value from Employee table to first calculation

In each case you can use this formula with proper WHERE condition with employee\_id

SELECT MONTHS\_BETWEEN (end\_date, start\_date) INTO v\_historic\_mont FROM job\_history;

+

SELECT MONTHS\_BETWEEN (SYSDATE, hire\_date) INTO v\_current\_mont FROM employees;

Also add EXCEPTION section for no data found situations



**Place your OWN screen shot of test here**

**Above is sample screen shot for employee\_id 106. Place your OWN screen shot of test here**