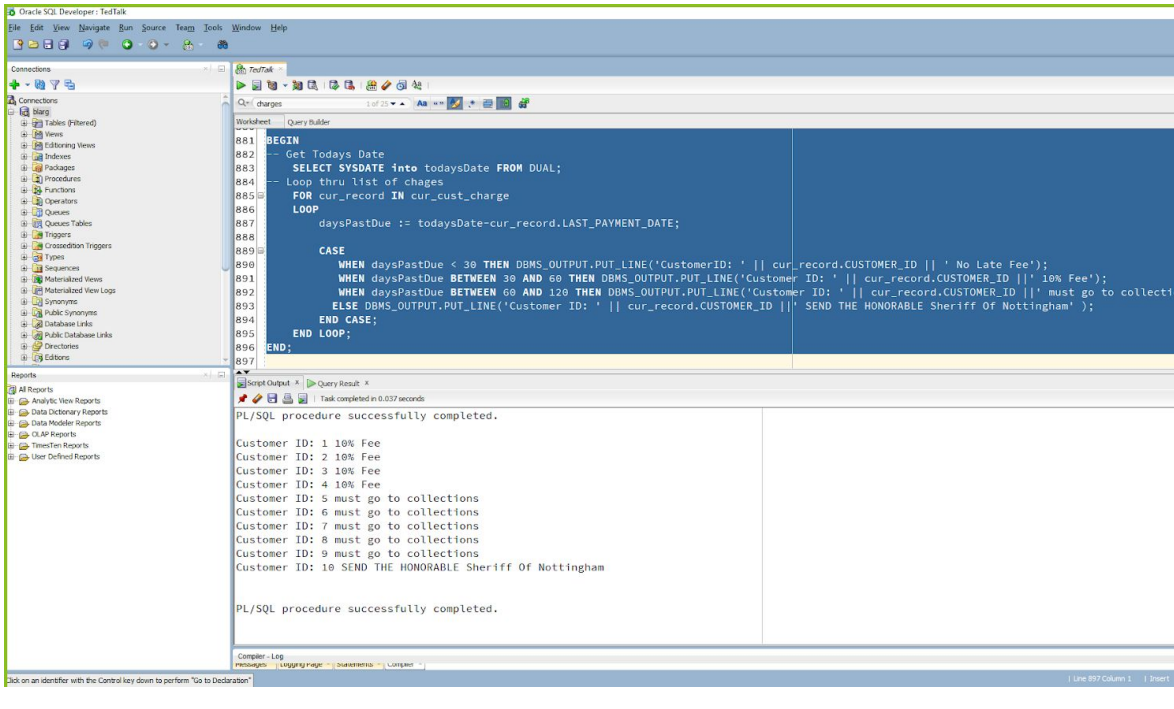
Robert Moore
& Greg Swaim
Lab 3
276 SQL
4/16/2019

Looping Lab ☺

1) Write an anonymous block that will:
   a. Create a cursor (or a cursor loop, your choice) that will select the customer ID, order ID, last payment date and current balance from each record in the CHARGES table.

   b. Loop through the following for each record

      i. If the last payment date is < 30 days no late payments will apply
      ii. If the last payment date is between 30 and 60 days a 10% late payment will apply
      iii. If the last payment date is between 60 and 120 days a message should output saying “Customer ID” (customer number) “must go to collections”
      iv. Close the cursor



insert into charges values ('1','1','1',(SELECT SYSDATE-30 FROM DUAL), 50.00);
insert into charges values ('2','2','2',(SELECT SYSDATE-40 FROM DUAL), 60.00);

```
insert into charges values ('3','3','3',(SELECT SYSDATE-50 FROM DUAL),
70.00 );
insert into charges values ('4','4','4',(SELECT SYSDATE-60 FROM DUAL),
80.00 );
insert into charges values ('5','5','5',(SELECT SYSDATE-70 FROM DUAL),
90.00 );
insert into charges values ('6','6','6',(SELECT SYSDATE-80 FROM DUAL),
100.00 );
insert into charges values ('7','7','7',(SELECT SYSDATE-90 FROM DUAL),
120.00 );
insert into charges values ('8','8','8',(SELECT SYSDATE-100 FROM DUAL),
130.00 );
insert into charges values ('9','9','9',(SELECT SYSDATE-120 FROM DUAL),
140.00 );
insert into charges values ('10','10','10',(SELECT SYSDATE-130 FROM
DUAL), 150.00 );


SET SERVEROUTPUT ON;
DECLARE
   cursor cur_cust_charge IS SELECT
CUSTOMER_ID,ORDER_ID,LAST_PAYMENT_DATE,CURRENT_BALANCE
FROM CHARGES;
    daysPastDue number(10) := NULL;
    todaysDate date := NULL;

BEGIN
-- Get Todays Date
   SELECT SYSDATE into todaysDate FROM DUAL;
-- Loop thru list of chages
   FOR cur_record IN cur_cust_charge
   LOOP
      daysPastDue := todaysDate-cur_record.LAST_PAYMENT_DATE;

      CASE
        WHEN daysPastDue < 30 THEN
DBMS_OUTPUT.PUT_LINE('CustomerID: ' || cur_record.CUSTOMER_ID || '
No Late Fee');
        WHEN daysPastDue BETWEEN 30 AND 60 THEN
DBMS_OUTPUT.PUT_LINE('Customer ID: ' || cur_record.CUSTOMER_ID ||'
10% Fee');
        WHEN daysPastDue BETWEEN 60 AND 120 THEN
DBMS_OUTPUT.PUT_LINE('Customer ID: ' || cur_record.CUSTOMER_ID ||'
must go to collections');
```

```
        ELSE DBMS_OUTPUT.PUT_LINE('Customer ID: ' ||
cur_record.CUSTOMER_ID ||' SEND THE HONORABLE Sheriff Of
Nottingham' );
      END CASE;
   END LOOP;
END;
```

2) Is the above cursor an explicit or implicit cursor?

Explicit - we declared a cursor.

3) I have two tables: A STUDENT table, and an ENROLLMENT table (sound familiar?). I need to calculate each student's GPA for **this term**.

The calculation for GPA (for this lab) is as follows:

If the grade is an A, the weight is 4 points.

If the grade is a B, the weight is 3 points.

If the grade is a C, the weight is 2 points.

If the grade is a D, the weight is 1 point.

An F receives 0 points.

The calculation for someone who received 2 A's, 1 B, and 1 C would be as follows:

2 A's = 8 points

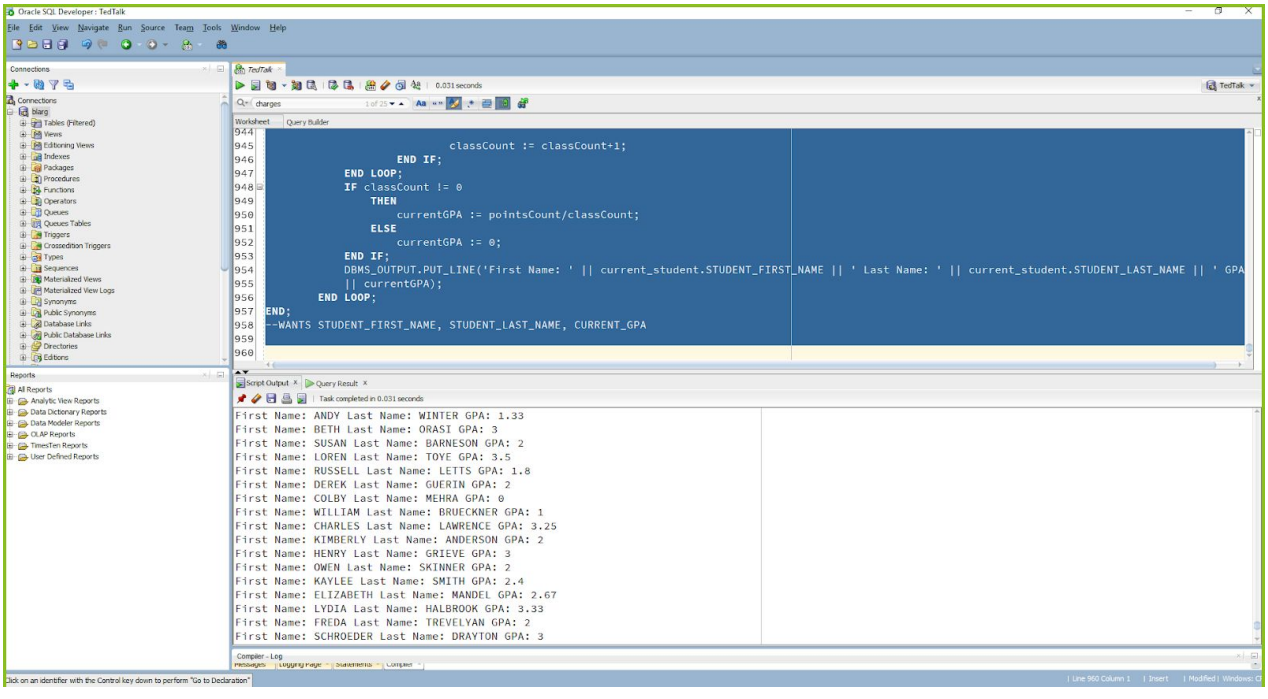
1 B = 3 points

1 C = 2 points

The total GPA would then be 13/4 = 3.25

Write an anonymous block that will:

a) Read in the student's information as well as their enrollment information through a cursor (or cursors)
b) Calculate the current term GPA into a scalar variable
c) Print out the following for each student:
   i) Student First Name
   ii) Student Last Name
   iii) Current Term GPA




```
DECLARE
   cursor cur_student IS SELECT STUDENT_FIRST_NAME,STUDENT_LAST_NAME,STUDENT_ID
FROM STUDENT;
   cursor cur_studentdata IS SELECT GRADE,FK_STUDENT_ID FROM ENROLLMENT;

   classCount Number(5) := NULL;
   pointsCount Number(5) := NULL;
   currentGPA Decimal(5,2) := NULL;
   BEGIN

   -- Begin Nested Loop thru students and course data
      FOR current_student IN cur_student
      LOOP
      -- Setup Counters
```

```
classCount := 0;
pointsCount := 0;
currentGPA := 0;
FOR current_data IN cur_studentdata
   LOOP
      IF current_student.STUDENT_ID = current_data.FK_STUDENT_ID
         THEN
            CASE current_data.GRADE
            WHEN 'A' THEN pointsCount := pointsCount + 4;
            WHEN 'A-' THEN pointsCount := pointsCount + 4;
            WHEN 'A+' THEN pointsCount := pointsCount + 4;
            WHEN 'B' THEN pointsCount := pointsCount + 3;
            WHEN 'B-' THEN pointsCount := pointsCount + 3;
            WHEN 'B+' THEN pointsCount := pointsCount + 3;
            WHEN 'C' THEN pointsCount := pointsCount + 2;
            WHEN 'C-' THEN pointsCount := pointsCount + 2;
            WHEN 'C+' THEN pointsCount := pointsCount + 2;
            WHEN 'D' THEN pointsCount := pointsCount + 1;
            WHEN 'D-' THEN pointsCount := pointsCount + 1;
            WHEN 'D+' THEN pointsCount := pointsCount + 1;
            ELSE
               NULL;
            END CASE;

            classCount := classCount+1;
      END IF;
END LOOP;
IF classCount != 0
```

```
            THEN
               currentGPA := pointsCount/classCount;
            ELSE
               currentGPA := 0;
         END IF;
         DBMS_OUTPUT.PUT_LINE('First Name: ' || current_student.STUDENT_FIRST_NAME || '
Last Name: ' || current_student.STUDENT_LAST_NAME || ' GPA: '
         || currentGPA);
      END LOOP;
END;
```