Assignment 2 (15% of total marks)

Due date: Sunday, 14 November 2021

Scope:

The tasks of this assignment cover topics on **PLSQL**.

Assessment criteria:

Marks will be awarded for:

- Correct,
- Comprehensive, and
- Appropriate

application of the materials covered in this subject.

Please read carefully information listed below.

This assignment contributes to 15% of the total assessment mark for the subject CSCI235.

A submission procedure is explained at the end of specification.

This assignment consists of 4 tasks and specification of each task starts from a new page.

A policy regarding late submissions is included in the subject outline.

For all the implemented tasks, your report or output must include a listing of all PL/SQL statements processed. To achieve that put the following SQL*Plus commands in all your scripts:

```
SPOOL file-name
SET ECHO ON
SET FEEDBACK ON
SET LINESIZE 100
SET PAGESIZE 200
SET SERVEROUTPUT ON
```

at the beginning of SQL script and

```
SPOOL OFF
```

at the end of SQL script.

Assignment Specification:

Task 1 (5.0 marks)

Stored PL/SQL procedure

Implement a **stored** PL/SQL procedure SUPPLIERACCBAL that lists information about the account balances of suppliers in nations within a specified region. The procedure first computes the average account balance of a nation. The procedure then extracts the supplier's information within a nation and determine if the supplier's account balance is above or below the nation's average. The information to be displayed include the supplier name, supplier phone number, the account balance, and a comment indicating whether the account balance is above or below the nation's average. An example of a segment of the output for suppliers from various nation within the region of ASIA is as follow:

Nation name: CHINA

Supplier name	Supplier phone	Account balance	Comment
Supplier#000001221	28-332-756-9313	$4,533.75	The account balance is above the nation average of $4,369.32.
Supplier#000002325	28-167-932-2440	$4,875.37	The account balance is above the nation average of $4,369.32.
Supplier#000000255	28-629-327-4139	$4,663.08	The account balance is above the nation average of $4,369.32.
Supplier#000000302	28-734-845-8630	$4,422.77	The account balance is above the nation average of $4,369.32.

Nation name: INDIA

Supplier name	Supplier phone	Account balance	Comment
Supplier#000001037	18-415-126-3978	$4,176.78	The account balance is below the nation average of $4,554.48.
Supplier#000000136	18-175-739-8397	$4,623.48	The account balance is above the nation average of $4,554.48.
Supplier#000000060	18-550-360-2464	$4,515.80	The account balance is below the nation average of $4,554.48.
Supplier#000001516	18-431-532-9957	$4,755.71	The account balance is above the nation average of $4,554.48.
Supplier#000001906	18-738-147-3630	$4,978.10	The account balance is above the nation average of $4,554.48.

…

It is up to you to decide if you want to handle exception in your procedure.

Deliverables

Hand in an SQL script and the report from execution of the script. The report must have no errors related to the implementation of your task and it must list all PL/SQL and SQL statements processed.

Remember to set ECHO option of SQL*Plus to ON!

Sample solution:

```
create or replace procedure supplierAccBalSP(regionName
REGION.r_name%type,
                    acctBal1 SUPPLIER.s_acctbal%type,
                    acctBal2 SUPPLIER.s_acctbal%type) IS
-- Declaration
prevNation  NATION.n_name%type := 'NotAvail';
balStatus    varchar2(100);
averageAcctBal    number(9,2);
state         varchar2(5);
--
begin -- Begin procedure block
  -- Create an implicit cursor to retrieve the required information
  for QRow in ( select s_name, n_name, s_phone, s_acctbal, s_nationkey
               from supplier, nation, region
               where s_nationkey = n_nationkey
               and n_regionkey = r_regionkey
               and r_name = upper(regionName)
               and s_acctbal between acctBal1 and acctBal2
               order by n_name )
  loop -- Begin implicit cursor loop
    --
    if (prevNation != QRow.n_name) then -- New nation or change nation
      -- Print header
      dbms_output.put_line( chr(10)      || 'Nation name: ' || QRow.n_name
|| chr(10) );
      dbms_output.put_line( chr(9)       || rpad('Supplier name',20) || ' '
                                  || rpad('Supplier phone',15) || ' '
                                  || rpad('Account balance',15) || ' '
                                  || 'Comment' || chr(10) );
      prevNation := QRow.n_name;
      --
      -- Compute the average account balance for a nation
      select avg(s_acctbal) into averageAcctBal
      from supplier
      where s_nationkey = QRow.s_nationkey;
      --
    end if;
    --
    -- Determine supplier's account balance status
    if (QRow.s_acctbal < averageAcctBal) then
      state := 'below';
    elsif (QRow.s_acctbal = averageAcctBal) then
      state := 'equal';
    else
```

```
      state := 'above';
    end if;
    --
    balStatus := 'The account balance is ' || state ||
                 ' the nation average of ' ||
                 trim(to_char(averageAcctBal,'$999G999G999D99')) || '.';
    -- print the supplier information
    dbms_output.put_line( chr(9)  || rpad(QRow.s_name, 20) || ' '
                                  || rpad(QRow.s_phone, 15) || ' '
                                  ||
lpad(trim(to_char(QRow.s_acctbal,'$999G999G999D99')),15) || ' '
                                  || trim(balStatus) );
    --
  end loop; -- End implicit cursor loop
  dbms_output.put_line(null);
  --
end; -- End procedure block
--
/
show error
```

Sample output:

```
SQL> execute supplieraccbalsp('ASIA',4000,5000);

Nation name: CHINA

          Supplier name        Supplier phone  Account balance Comment

          Supplier#000001221   28-332-756-9313      $4,533.75 The account balance is above the nation average
of $4,369.32.
          Supplier#000002325   28-167-932-2440      $4,875.37 The account balance is above the nation average
of $4,369.32.
          Supplier#000000724   28-471-255-1476      $4,696.62 The account balance is above the nation average
of $4,369.32.
          Supplier#000002402   28-116-664-4294      $4,724.36 The account balance is above the nation average
of $4,369.32.
          Supplier#000002631   28-153-887-4167      $4,658.15 The account balance is above the nation average
of $4,369.32.
          Supplier#000002263   28-438-817-3756      $4,903.47 The account balance is above the nation average
of $4,369.32.
          Supplier#000002184   28-665-517-6175      $4,397.36 The account balance is above the nation average
of $4,369.32.
          Supplier#000002742   28-884-467-8595      $4,571.04 The account balance is above the nation average
of $4,369.32.
          Supplier#000000117   28-470-879-3141      $4,589.18 The account balance is above the nation average
of $4,369.32.
          Supplier#000001976   28-244-422-3128      $4,911.27 The account balance is above the nation average
of $4,369.32.
          Supplier#000000196   28-430-406-1127      $4,710.62 The account balance is above the nation average
of $4,369.32.
          Supplier#000000255   28-629-327-4139      $4,663.08 The account balance is above the nation average
of $4,369.32.
          Supplier#000000302   28-734-845-8630      $4,422.77 The account balance is above the nation average
of $4,369.32.

Nation name: INDIA
```

Supplier name	Supplier phone	Account balance	Comment
Supplier#000001037	18-415-126-3978	$4,176.78	The account balance is below the nation average of $4,554.48.
Supplier#000000931	18-174-741-5563	$4,398.36	The account balance is below the nation average of $4,554.48.
Supplier#000000878	18-462-213-5795	$4,140.02	The account balance is below the nation average of $4,554.48.
Supplier#000000136	18-175-739-8397	$4,623.48	The account balance is above the nation average of $4,554.48.
Supplier#000000060	18-550-360-2464	$4,515.80	The account balance is below the nation average of $4,554.48.
Supplier#000000303	18-932-912-3102	$4,500.90	The account balance is below the nation average of $4,554.48.
Supplier#000002945	18-946-203-4742	$4,164.16	The account balance is below the nation average of $4,554.48.
Supplier#000001598	18-847-850-8215	$4,145.32	The account balance is below the nation average of $4,554.48.
Supplier#000001516	18-431-532-9957	$4,755.71	The account balance is above the nation average of $4,554.48.
Supplier#000001464	18-984-442-6908	$4,336.46	The account balance is below the nation average of $4,554.48.
Supplier#000001906	18-738-147-3630	$4,978.10	The account balance is above the nation average of $4,554.48.
Supplier#000001799	18-642-860-9158	$4,648.72	The account balance is above the nation average of $4,554.48.
Supplier#000002246	18-257-298-6662	$4,801.02	The account balance is above the nation average of $4,554.48.
Supplier#000002156	18-105-354-2253	$4,617.70	The account balance is above the nation average of $4,554.48.
Supplier#000001320	18-732-433-1443	$4,925.36	The account balance is above the nation average of $4,554.48.

Nation name: INDONESIA

Supplier name	Supplier phone	Account balance	Comment
Supplier#000001542	19-766-450-1320	$4,668.72	The account balance is above the nation average of $4,539.48.
Supplier#000000050	19-561-560-7437	$4,515.87	The account balance is below the nation average of $4,539.48.
Supplier#000001827	19-477-545-3938	$4,919.94	The account balance is above the nation average of $4,539.48.
Supplier#000002313	19-738-303-4653	$4,325.58	The account balance is below the nation average of $4,539.48.
Supplier#000002801	19-733-439-2229	$4,772.52	The account balance is above the nation average of $4,539.48.
Supplier#000001048	19-493-938-3406	$4,694.02	The account balance is above the nation average of $4,539.48.
Supplier#000000313	19-648-945-5128	$4,114.68	The account balance is below the nation average of $4,539.48.

Nation name: JAPAN

Supplier name	Supplier phone	Account balance	Comment
Supplier#000001987	22-196-523-5808	$4,101.27	The account balance is below the nation average of $4,705.99.
Supplier#000002168	22-144-605-9504	$4,616.71	The account balance is below the nation average of $4,705.99.
Supplier#000001303	22-688-457-2776	$4,368.88	The account balance is below the nation average of $4,705.99.
Supplier#000000685	22-599-473-1489	$4,297.36	The account balance is below the nation average of $4,705.99.
Supplier#000000277	22-758-939-2357	$4,300.15	The account balance is below the nation average of $4,705.99.

```
        Supplier#000000112   22-617-876-1402      $4,332.95 The account balance is below the nation average
of $4,705.99.
        Supplier#000000691   22-930-512-3497      $4,239.95 The account balance is below the nation average
of $4,705.99.

Nation name: VIETNAM

        Supplier name        Supplier phone  Account balance Comment

        Supplier#000001318   31-779-241-5392      $4,839.36 The account balance is above the nation average
of $4,010.78.
        Supplier#000003000   31-910-209-2002      $4,223.03 The account balance is above the nation average
of $4,010.78.
        Supplier#000001351   31-971-224-5549      $4,647.09 The account balance is above the nation average
of $4,010.78.
        Supplier#000001090   31-523-106-8117      $4,126.63 The account balance is above the nation average
of $4,010.78.
        Supplier#000001016   31-756-833-2500      $4,168.02 The account balance is above the nation average
of $4,010.78.
        Supplier#000002995   31-224-848-8404      $4,614.07 The account balance is above the nation average
of $4,010.78.
        Supplier#000001481   31-207-618-9010      $4,436.86 The account balance is above the nation average
of $4,010.78.
        Supplier#000002677   31-596-125-8311      $4,980.70 The account balance is above the nation average
of $4,010.78.
        Supplier#000002655   31-276-987-1638      $4,856.74 The account balance is above the nation average
of $4,010.78.
        Supplier#000002520   31-826-707-8432      $4,965.26 The account balance is above the nation average
of $4,010.78.
        Supplier#000000400   31-514-285-7013      $4,624.87 The account balance is above the nation average
of $4,010.78.
        Supplier#000000035   31-720-790-5245      $4,381.41 The account balance is above the nation average
of $4,010.78.
        Supplier#000001742   31-253-408-5060      $4,627.46 The account balance is above the nation average
of $4,010.78.

PL/SQL procedure successfully completed.

SQL>
```

Task 2 (5.0 marks)

Stored PL/SQL Function

Implement a **stored** PL/SQL function that finds a supplier who supplies the cheapest (lowest supply cost) or the dearest (highest supply cost) for a part specified by a user. The function should obtain the part key as its formal parameter, and it should return a string of values consisting of the supplier key, supplier name, and the cost (cheapest and dearest). Example of the string output are as follow:

```
Supplier with cheapest cost: 1344, Supplier#000001344, $172.09.
Supplier with dearest cost: 2882, Supplier#000002882, $791.07.

Supplier with cheapest cost: 770, Supplier#000000770, $74.55.
Supplier with dearest cost: 2308, Supplier#000002308, $826.35.
```

Next use the function to list the string output as specified above for part numbers 3753, 43064, 57574 and 60000 using a select statement.

Deliverables

Hand in an SQL script and the report from execution of the script. The report must have no errors related to the implementation of your task and it must list all PL/SQL and SQL statements processed.

Remember to set ECHO option of SQL*Plus to ON!

```
create or replace function findSupplierByPartSF( partKey
PART.p_partkey%type )
return varchar2
is
partsuppRow             PARTSUPP%rowtype;
cheapestSupplier SUPPLIER.s_name%type;
dearestSupplier         SUPPLIER.s_name%type;
res                 varchar2(1024) := '';
begin
  -- Find the supplier that supplies the part with cheapest cost
  for QRow in (     select *
                from PARTSUPP
                where ps_partkey = partkey
                and ps_supplycost = (
                        select min(ps_supplycost)
                        from partsupp
                        where ps_partkey = partkey ) )
  --
  loop
    select s_name into cheapestSupplier
    from SUPPLIER
    where s_suppkey = QRow.ps_suppkey;
    --
    res := trim(res) || 'Supplier supplying part '
        || QRow.ps_partkey
        || ' with cheapest cost is/are: ' || chr(10)
        || chr(9) || QRow.ps_suppkey || ', '
        || trim(cheapestSupplier) || ', '
        || trim(to_char(QRow.ps_supplycost,'$999G999G999D99'))
        || '.' || chr(10);
  --
  end loop;
  res := trim(res) || chr(10);
  --
  -- Find the supplier that supplies the part with the dearest cost
  for QRow in (     select *
```

```
                from PARTSUPP
                where ps_partkey = partkey
                and ps_supplycost = (
                        select max(ps_supplycost)
                        from partsupp
                        where ps_partkey = partkey ) )
  --
  loop
    select s_name into dearestSupplier
    from SUPPLIER
    where s_suppkey = QRow.ps_suppkey;
    --
    res := trim(res) || 'Supplier supplying part '
        || QRow.ps_partkey
        || ' with dearest cost is/are: ' || chr(10)
        || chr(9) || QRow.ps_suppkey || ', '
        || trim(dearestSupplier) || ', '
        || trim(to_char(QRow.ps_supplycost,'$999G999G999D99'))
        || '.' || chr(10);
  --
  end loop;
  return trim(res);
end;
/
show errors;

sample output:
SQL> select findSupplierByPartSF(p_partkey)
from part
where p_partkey in (3753, 43064, 57574, 60000);

FINDSUPPLIERBYPARTSF(P_PARTKEY)
--------------------------------------------------------------------------------
-------------
Supplier supplying part 3753 with cheapest cost is/are:
        754, Supplier#000000754, $305.95.
Supplier supplying part 3753 with cheapest cost is/are:
        2256, Supplier#000002256, $305.95.

Supplier supplying part 3753 with dearest cost is/are:
        1505, Supplier#000001505, $457.10.

Supplier supplying part 43064 with cheapest cost is/are:
        1065, Supplier#000001065, $667.30.
Supplier supplying part 43064 with cheapest cost is/are:
```

1829, Supplier#000001829, $667.30.

Supplier supplying part 43064 with dearest cost is/are:
357, Supplier#000000357, $848.13.

Supplier supplying part 57574 with cheapest cost is/are:
1344, Supplier#000001344, $172.09.

Supplier supplying part 57574 with dearest cost is/are:
2882, Supplier#000002882, $791.07.

Supplier supplying part 60000 with cheapest cost is/are:
770, Supplier#000000770, $74.55.

Supplier supplying part 60000 with dearest cost is/are:
2308, Supplier#000002308, $826.35.

SQL>

Task 3 (5.0 marks)

Stored trigger

Implement **a row trigger** that enforces the following consistency constraint.

The column c_comment in the relational table CUSTOMER of the TPCHR benchmark database is defined as 'NOT NULL'. Create a row trigger that automatically updates the values in the column (c_comment) to 'New customer was created on <the system date>' if the comment of the newly inserted record is left as NULL when a new customer is inserted into the relational table CUSTOMER. Your trigger, once activated, will enforce the consistency constraint described.
When ready, process the SQL script solution2.sql and record the results of processing in a file solution2.lst.

Deliverables

Hand in the SQL script and the report from execution of scripts.

Remember to set ECHO option of SQL*Plus to ON!

Sample solution:

```
CREATE OR REPLACE TRIGGER insertCustomerComment
BEFORE INSERT OR UPDATE ON customer
FOR EACH ROW
WHEN (NEW.c_custkey > 0)
DECLARE
  CustComment   CUSTOMER.c_comment%type;
BEGIN
  if (:NEW.c_comment is null) then
       :NEW.C_COMMENT := 'New customer was created on ' || sysdate;
       dbms_output.put_line('Customer key: ' || :NEW.c_custkey
                    || ', ' || :new.c_name);
       dbms_output.put_line('Comment: ' || :NEW.c_comment);
  end if;
END;
/
show error
```

Sample output:

```
SQL> insert into customertest
values(999999,'Test customer 999999','Test
address',10,'1234567890',1000,'AUTOMOBILE',null);  2
```

```
Customer key: 999999, Test customer 999999
Comment: New customer was created on 31-OCT-21
Customer key: 999999, Test customer 999999
Old Account Balance:
New Account Balance: 1000
Account balance difference:

1 row created.

SQL> insert into customertest
values(99999,'Test customer 999999','Test
address',10,'1234567890',1000,'AUTOMOBILE','Test Comment');  2
Customer key: 99999, Test customer 999999
Old Account Balance:
New Account Balance: 1000
Account balance difference:

1 row created.

SQL>  update customertest
 set c_comment = null
where c_custkey = '99999';
Customer key: 99999, Test customer 999999
Comment: New customer was created on 31-OCT-21
Customer key: 99999, Test customer 999999
Old Account Balance: 1000
New Account Balance: 1000
Account balance difference: 2000

1 row updated.

SQL>
```

Suggested Solution:

```
CREATE OR REPLACE TRIGGER verifyccard
BEFORE INSERT OR UPDATE ON Pbasket
FOR EACH ROW
DECLARE
   lv_c# pbasket.c#%type := null;
   loop_pbasket pbasket%rowtype;
BEGIN
   for loop_pbasket IN ( SELECT *
                         FROM   Pbasket)
```

```
   loop
    IF (loop_pbasket.ccard = :NEW.ccard AND
        loop_pbasket.c# != :NEW.c#) then
        RAISE_APPLICATION_ERROR ( -20001, 'Credit card ' || to_char(:new.ccard) || '
has been used by a customer with customer number ' || to_char(lv_c#));
    END IF;
   end loop loop_pbasket;
      exception
        when no_data_found then /* do nothing */
        null;
END;
/

SQL> select * from pbasket;

WHENCREAT WHENFINAL      CCARD        C#
--------- --------- ---------- ----------
02-APR-19 02-APR-19 4594567829       100
12-APR-19 12-APR-19 1294522229       101
08-MAY-19 08-MAY-19 1294522229       101
16-JAN-19 16-JAN-19 6781394229       102
05-MAY-19 05-MAY-19 9981364599       100

SQL> INSERT INTO Pbasket VALUES
  2  ( TO_DATE('29-MAY-2019:09:00', 'DD-MON-YYYY:HH:MI'), TO_DATE('10-JUN-
2019:09:15', 'DD-MON-YYYY:HH:MI'), 9981364599, 100 );

1 row created.

SQL> INSERT INTO Pbasket VALUES
  2  ( TO_DATE('20-MAY-2019:09:00', 'DD-MON-YYYY:HH:MI'), TO_DATE('21-MAY-
2019:09:15', 'DD-MON-YYYY:HH:MI'), 9981364599, 101 );
INSERT INTO Pbasket VALUES
            *
ERROR at line 1:
ORA-20001: Credit card 9981364599 has been used by a customer with customer
number
ORA-06512: at "JAPIT235.VERIFYCCARD", line 10
ORA-06512: at "JAPIT235.VERIFYCCARD", line 10
ORA-04088: error during execution of trigger 'JAPIT235.VERIFYCCARD'


SQL> INSERT INTO Pbasket VALUES
  2  ( TO_DATE('30-MAY-2019:09:00', 'DD-MON-YYYY:HH:MI'), TO_DATE('11-JUN-
2019:09:15', 'DD-MON-YYYY:HH:MI'), 9981364608, 100 );

1 row created.
```

```
SQL> select * from pbasket;

WHENCREAT WHENFINAL      CCARD        C#
--------- --------- ---------- ----------
02-APR-19 02-APR-19 4594567829        100
12-APR-19 12-APR-19 1294522229        101
08-MAY-19 08-MAY-19 1294522229        101
16-JAN-19 16-JAN-19 6781394229        102
05-MAY-19 05-MAY-19 9981364599        100
29-MAY-19 10-JUN-19 9981364599        100
30-MAY-19 11-JUN-19 9981364608        100

7 rows selected.

SQL>
```

Submissions

This assignment is due by 9:00 pm (21:00 hours) 20 May 2021, **Singapore time**.

Submit the files **solution1.pdf**, **solution2.pdf**, and **solutions3.pdf** through Moodle in the following way:

1) Zip all the files (Solution1.pdf, solution2.pdf, and solution3.pdf into one zipped folder.)
2) Access Moodle at **http://moodle.uowplatform.edu.au/**
3) To login use a Login link located in the right upper corner the Web page or in the middle of the bottom of the Web page
4) When successfully logged in, select a site CSCI235 (SP221) Database Systems
5) Scroll down to a section Submissions of Assignments
6) Click at Submit your Assignment 2 here link.
7) Click at a button Add Submission
8) Move the zipped file created in Step 1 above into an area provided in Moodle. You can drag and drop files here to add them. You can also use a link *Add…*
9) Click at a button Save changes,
10) Click at check box to confirm authorship of a submission,
11) When you are satisfied, remember to click at a button Submit assignment.

A policy regarding late submissions is included in the subject outline.

Only one submission per student is accepted.

Assignment 2 is an individual assignment and it is expected that all its tasks will be solved individually without any cooperation with the other students. Plagiarism is treated seriously. Students involved will likely receive zero. If you have any doubts, questions, etc. please consult your lecturer or tutor during lab classes or over e-mail.

End of specification