East West University
Department of Computer Science and Engineering

CSE 464: Assignment 01
Course Instructor: Dr. Mohammad Rezwanul Huq

Problem Solving using PL/SQL

General Instructions:

- All the procedures used in the code should be named according to the following format.
  Proc_<question-number>_<student-id>
  Suppose, you are answering question 1 and your student ID is 2022-1-60-001 then the procedure name should be – Proc_1_2022-1-60-001
  Consider the following Procedure.

```
-- Question 1
CREATE OR REPLACE PROCEDURE Proc_1_2022_1_60_001 IS
BEGIN
    dbms_output.put_line('Student ID: 2022-1-60-001');
    dbms_output.put_line('The rest of the computation will
follow.');

END;
/

EXECUTE Proc_1_2022_1_60_001;
```

- Inside the body of the Procedure it must display your student id first as shown in the above procedure.
- The function and the trigeers should be also named in the same way.
- You must create the related tables and put sufficient rows into the table, if required.
- You must submit two files.
  - **<Student-ID>_<Section-Number>.SQL** – this SQL file must include all codes. Use appropriate commenting to indicate the start of the answer of a question. See the above code.
  - **<Student-ID>_<Section-Number>.DOCX** – this DOC file must contain the screen shot of the execution of your code. Look at the following screenshot for the above code.
    Answer: Question 1

```
Statement processed.
Student ID: 2022-1-60-001
The rest of the computation will follow.
```

Problem Sets

1. (Basic Table Manipulation) Write a PL/SQL Procedure that declares an integer variable called num, assigns a value to it, and computes and inserts into the tempp table the value of the variable itself, its square, and its cube. Consider the following DDL statements to create tempp table.

```
CREATE TABLE tempp ( item number, square number, CUBE number );
```

2. (Conditional Statements) Write a PL/SQL Procedure that asks for three positive integers representing the sides of a triangle, and determine whether they form a valid triangle. Hint: In a triangle, the sum of any two sides must always be greater than the third side. Display the output on the screen using dbms_output.put_line.

3. (Conditional Statements) Suppose the grade obtained by a student depends upon his scores and the grading rule is as follows. :-

Scores	Grades
95-100	A
70-84	B
70-84	C
60-69	D
0-59	F

   Write a PL/SQL Procedure that accepts a student's marks and output his grade. Display the output on the screen using dbms_output.put_line.

4. (Conditional Statements) A company manufactures three products:- computer stationery, fixed disks and computers. The following codes are used to indicate them:-

Product	Code
Computer Stationery	1
Fixed Disks	2
Computers	3

   The company has a discount policy as follows:

Product Order	amount	Discount rate
Computer stationery	5000 and above	12%
Computer stationery	3000 and above	8%
Computer stationery	Below 3000	2%
Fixed disks	20000 and above	10%
Fixed disks	15000 and above	5%
Computers	50000 and above	10%
Computers	25000 and above	5%

   Write a PL/SQL Procedure to accept the order details i.e. product code and order amounts for the products, calculate the discount amounts as per this policy and output the net order amount (after subtracting discounted amount). Display the output on the screen using dbms_output.put_line.

5. (Iteration) Write a PL/SQL Procedure that examines all the numbers from 1 to 999, displaying all those for which the sum of the cubes of the digits equal the number itself. Display the output on the screen using dbms_output.put_line.

6. (Iteration) A palindrome is a word that is spelled the same forward and backward, such as level, radar, etc. Write a PL/SQL Procedure that reads in a five letter word from the user and determine whether it is a palindrome. Display the results on the screen using dbms_output.put_line.

7. (Procedure) The CUSTOMER table of a state electricity board consists of the following fields:

```
Meter_Number varchar2(4)
Meter_Type char(1)
Previous_Reading Number
Current_Reading Number
Customer_Type char(1)
```

```
Last_Bill_payment char(1) (values could be 'Y' or 'N')
```

There are two types of meters.

- 3- phase coded as 'T'
- 1-phase coded as 'S'

There are 4 types of customers.

- Agricultural coded as 'A'
- Industrial coded as 'I'
- Commercial coded as 'C'
- Residential coded as 'R'

Formulae used:

Units used = Current Reading - Previous Reading
Rate = 1/ 1.25/ 1.50/ 1.30 for A/I/C/R respectively.
Amount = rate*units used
Surcharge = 5% for single phase
10% for 3 phase
Excise = 30% of (amount +Surcharge)
Net = Amount +Surcharge + Excise

Write a procedure calculate_bill to calculate the bill for each customer. Consider the following customer table to read input data.

```
CREATE TABLE CUSTOMER (
     Meter_Number varchar2(4),
     Meter_Type char(1),
     Previous_Reading number,
     Current_Reading number,
     Customer_Type char(1),
     Last_Bill_payment char(1),
     check(Last_Bill_payment = 'Y' OR Last_Bill_payment = 'N')
);
```

--insert dummy data into table customer

Meter_number	Meter_Type	Previous_Reading	Current_Reading	Customer_Type	Last_Bill_payment
1000	S	3000	5000	A	Y
1001	T	3000	5000	R	Y
1002	S	400	2000	R	Y

After calculating the bill, the procedure should insert the total Amount, Surcharge, Excise and Net into the Bill table as shown below.

```
CREATE TABLE BILL (
     Meter_Number varchar2(4) PRIMARY KEY,
     units number,
     rate number,
     amount number,
     surcharge number,
     Excise number,
     Net number
);
```

8. (Function) Write a PL/SQL function to take three parameters, the sides of a triangle. The function should return a Boolean value - true if the triangle is valid, false otherwise. A triangle is valid if the length of each side is less than the sum of the lengths of the other two sides. Check if the dimensions entered by the user can form a valid triangle by writing an appropriate PL/SQL Procedure. Display the results on the screen using dbms_output.put_line.

9. (Trigger) Consider the following relations.

   Account relation consists of three attributes - Account_no, Account_Name, Account_Balance.

   Let us assume that the Account table initially contains the records as shown below.

Account_no	Account_Name	Account_Balance
A1	C1	5000
A2	C2	15000

   Transactions relation consists of five attributes - Transaction_no, Src_Acc_no, Dst_Acc_No, Transaction_Type, Amount. Transaction_no is unique. Src_Acc_no is the account number that initiated the transaction. There are three different types of transactions – Deposit (D), Withdraw (W) and Transfer (T). In case of Deposit and Withdraw transaction, the Dst_Acc_No (destination account number) is NULL. However, when an account holder transfers an amount from his/her account to another, Dst_Acc_No holds the other account number. In that case, the balance of the Src_Acc_No will be decreased and the balance of the Dst_Acc_No will be increased.

   Your job is to write an appropriate trigger that works for all trnasaction types and updates account balance correctly.

   As for example, consider the following records in Transaction relations.

Transaction_no	Src_Acc_no	Dst_Acc_No	Transaction_Type	Amount
1	A1	NULL	W	1000
2	A2	NULL	D	3000
3	A2	A1	T	5000

   If your trigger is executed correctly for each row in Transactions table while being inserted, the Account table should look like the following at the end of these transactions.

Account_no	Account_Name	Account_Balance
A1	C1	9000
A2	C2	13000