Lab 09: PL/SQL II

Objective

The students should be able to understand the usage of:

- Triggers
- Procedures
- Functions

Submission Requirements

Save your .sql script file and upload it to LMS. Add appropriate comments for clear working.

SQL and PL/SQL

1. Create tables Customer and Sales as follows.

```
CREATE TABLE CUSTOMER
(
  CID                   VARCHAR2(3) PRIMARY KEY,
  CNAME                 VARCHAR2(25),
  CREDIT_LIMIT          NUMBER,
  CREDIT_BALANCE        NUMBER
);

INSERT INTO CUSTOMER VALUES ('C81','Alpha',99,0);
INSERT INTO CUSTOMER VALUES ('C82','Bravo',700,0);
INSERT INTO CUSTOMER VALUES ('C83','Charlie',5000,0);
COMMIT;

CREATE TABLE SALES
(
  SID  NUMBER PRIMARY KEY,
  SDATE  DATE default SYSDATE,
  PCODE  VARCHAR2(3),
  CID    VARCHAR2(3),
  QTY    NUMBER,
  RATE   NUMBER,
  AMOUNT NUMBER,
  FOREIGN KEY (CID) REFERENCES CUSTOMER(CID)
);
```

Create the following Triggers:

a. SALES_Before_insert which updates customer credit balance (in customer table) to credit balance + :new.Amount before inserting amount in the sales table. Test your trigger by inserting a record in the sales table and checking the values in customers table.

   b. SALES_Before_Del which reduces customers credit balance by the amount before deleting each row of the sales table. Delete the record inserted in a to test the trigger.
   c. Drop the 2 triggers created in (a) and (b). Create a trigger, SALES_Insert_Del which combines the functionality of the above 2 triggers into one. Repeat the tests in (a) and (b) to validate your trigger.

2.
```
CREATE TABLE orders (
        order_id number(5) PRIMARY KEY,
        quantity number(4),
        cost_per_item number(6,2),
        total_cost number(8,2),
        discount number(2),
        final_charged number(8,2)
    );
```
   a) Run the DDL commands to create the table with appropriate data types.
   b) Create a trigger that generates the total_cost by multiplying the cost per item and quantity whenever a new record is added. Also calculate the final_charged which will be generated by applying the discount.
   c) Insert the record and view the updated table to validate your trigger.
```
INSERT INTO orders( order_id, quantity, cost_per_item,discount) VALUES (1,10,
200,25);
```
   d) Insert another record of choice

3. Suppose the following schema records the currency rate and fluctuations against the Pakistani Rupee.
```
Currency_con (CID, Currency, Rate)
fluctuations (recDate, Currency, Difference)
Note: We do not require a PK here for fluctuations log table.
```
   a. Write DDL commands to create the table with appropriate data types.
   b. Insert some records in the table for example current US dollar rate and current Pound rate.
   c. Write a trigger to log the currency fluctuations as following:
      i. When new currency is added to the table, then insert a record in the fluctuations table. The date would be the current system date and difference would be set as 0.
      ii. When a currency rate changes, the difference between new and old values would be calculated and stored in the fluctuations table. The positive sign will indicate an increase and negative would indicate a decrease.

4. Suppose we have a Worker table as follows:
```
worker(workerID, lname, gender, salary, commission, deptID)
```
   a. Write DDL commands to create the table with appropriate data types.
   b. Declare a sequence for workerID that begins from 100 and increments by 5.
   c. Write a trigger that automatically inserts the primary key with a sequential number when inserting a record in the worker table.

   d. Insert 2 records to test your trigger, each time providing all attributes except the primary key.

5. Suppose we have the following two tables:

   ```
   OrderHeader(OrderID, Odate, CustID, Total)
   Order_Item(OrderID,ItemID, Qty, Subtotal)
   ```

   a. Write DDL commands to create the tables with suitable datatypes.
   b. Write a statement-level trigger that updates the *Total* in the orderHeader table with the total value of the order_item records for a particular order whenever an insert, update or delete event occurs on the order_item table. For any update error, raise an exception.
   c. Insert the following to test the outcome.

      ```
      INSERT INTO OrderHeader (OrderId, Odate, CustID) VALUES (1,SYSDATE, 1);
      INSERT INTO Order_Item VALUES (1,1, 20, 200);
      INSERT INTO Order_Item VALUES (1,2, 5, 100);
      ```

Do the following using the HR database schema:

6. Create a function named average_dept_salary which takes a department name as input and return the average of the salary. Test the function using the syntax below:

   ```
   select average_dept_salary ('IT')
   from dual;
   ```

7. Create a procedure that prints the region wise maximum salary for all employees in a suitable format. Execute to test the procedure.
8. Create a function that returns the count of employees who have more than one record in the job_history table. Test the function.