SSN COLLEGE OF ENGINEERING, KALAVAKKAM
(An Autonomous Institution, Affiliated to Anna University, Chennai)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**UCS1412 – DBMS Lab**
**Assignment – 6**

*Name: Jayannthan P T* *Dept: CSE 'A'* *Roll No.: 205001049*

**Title:** Stored Procedures and Functions

**Spool File Output:**

```
SQL>
-- Write a PL/SQL stored procedure for the following:

-- 1. For the given receipt number, calculate the Discount as follows:
-- For total amount > $10 and total amount < $25: Discount=5%
-- For total amount > $25 and total amount < $50: Discount=10%
-- For total amount > $50: Discount=20%
-- Calculate the amount (after the discount) and update the same in Receipts table.

SQL> CREATE OR REPLACE PROCEDURE PRINTRECEIPT
  2  (ORDER_NUMBER IN ITEM_LIST.RNO%TYPE)
  3  IS
  4      CFNAME CUSTOMERS.FNAME%TYPE;
  5      CLNAME CUSTOMERS.LNAME%TYPE;
  6   RECDATE RECEIPTS.RDATE%TYPE;
  7      -- TOTAL_ITEM NUMBER:=0;
  8      TOTAL_PRICE NUMBER:=0;
  9      DISCOUNT NUMBER:=0;
 10      CURSOR DETAILS IS SELECT FOOD,FLAVOR,PRICE,COUNT(*) AS QUANTITY FROM
ITEM_LIST,PRODUCTS WHERE RNO=ORDER_NUMBER AND PID=ITEM GROUP BY FOOD,FLAVOR,PRICE;
 11      DETAILROW DETAILS%ROWTYPE;
 12  BEGIN
 13      SELECT FNAME,LNAME,RDATE INTO CFNAME,CLNAME,RECDATE FROM CUSTOMERS JOIN RECEIPTS
USING(CID) WHERE RNO=ORDER_NUMBER;
 14      DBMS_OUTPUT.PUT_LINE('RECEIPT NUMBER: '||ORDER_NUMBER);
 15      DBMS_OUTPUT.PUT_LINE('CUSTOMER NAME: '||CFNAME||' '||CLNAME);
 16      DBMS_OUTPUT.PUT_LINE('RECEIPT DATE: '||RECDATE);
 17      OPEN DETAILS;
 18      DBMS_OUTPUT.PUT_LINE('FOOD            FLAVOR            QUANTITY');
 19      LOOP
 20          FETCH DETAILS INTO DETAILROW;
 21          IF DETAILS%NOTFOUND THEN
 22              EXIT;
 23          ELSE
 24     DBMS_OUTPUT.PUT_LINE(DETAILROW.FOOD||'            '||DETAILROW.FLAVOR||'
    '||DETAILROW.QUANTITY);
 25              -- TOTAL_ITEM:=TOTAL_ITEM+DETAILROW.QUANTITY;
```

```
 26              TOTAL_PRICE:=TOTAL_PRICE+DETAILROW.QUANTITY*DETAILROW.PRICE;
 27          END IF;
 28      END LOOP;
 29      DBMS_OUTPUT.PUT_LINE('TOTAL AMOUNT :$ '||TOTAL_PRICE);
 30   IF TOTAL_PRICE<=10 THEN
 31       DBMS_OUTPUT.PUT_LINE('NO DISCOUNT');
 32   END IF;
 33   IF TOTAL_PRICE>10 AND TOTAL_PRICE<25 THEN
 34    DISCOUNT:=0.05*TOTAL_PRICE;
 35       DBMS_OUTPUT.PUT_LINE('DISCOUNT (5%) :$ '||DISCOUNT);
 36   END IF;
 37   IF TOTAL_PRICE<50 THEN
 38    DISCOUNT:=0.1*TOTAL_PRICE;
 39    DBMS_OUTPUT.PUT_LINE('DISCOUNT (10%) :$ '||DISCOUNT);
 40   END IF;
 41   IF TOTAL_PRICE>=50 THEN
 42    DISCOUNT:=0.2*TOTAL_PRICE;
 43    DBMS_OUTPUT.PUT_LINE('DISCOUNT (20%) :$ '||DISCOUNT);
 44   END IF;
 45      CLOSE DETAILS;
 46   TOTAL_PRICE:=TOTAL_PRICE-DISCOUNT;
 47   DBMS_OUTPUT.PUT_LINE('AMOUNT TO BE PAID $ '||TOTAL_PRICE);
 48   EXCEPTION
 49      WHEN NO_DATA_FOUND THEN
 50          DBMS_OUTPUT.PUT_LINE('INVALID ORDER NUMBER!');
 51  END;
 52  /

Procedure created.

SQL>
SQL> CALL PRINTRECEIPT(13355);
RECEIPT NUMBER: 13355
CUSTOMER NAME: TOUSSAND SHARRON
RECEIPT DATE: 19-OCT-07
FOOD            FLAVOR             QUANTITY
Cake             Napoleon             1
Cookie             Lemon              1
Cake             Opera              1
TOTAL AMOUNT :$ 30.23
DISCOUNT (10%) :$ 3.023
AMOUNT TO BE PAID $ 27.207

Call completed.

-- 2. Ask the user for the budget and his/her preferred food type. You recommend the best
-- item(s) within the planned budget for the given food type. The best item is
-- determined by the maximum ordered product among many customers for the given
-- food type.
SQL>
SQL> CREATE OR REPLACE PROCEDURE BUDGETFOOD
  2  (BUDGET IN NUMBER,FOODTYPE IN PRODUCTS.FOOD%TYPE)
  3  IS
```

```
  4      CURSOR ITEMLIST_CURSOR IS SELECT PID,FOOD,FLAVOR,PRICE FROM PRODUCTS WHERE
FOOD=FOODTYPE AND PRICE<BUDGET;
  5      PRODUCTROW ITEMLIST_CURSOR%ROWTYPE;
  6      CNT NUMBER:=0;
  7      FLAG NUMBER:=0;
  8      PIID ITEM_LIST.ITEM%TYPE;
  9      FLAVOR PRODUCTS.FLAVOR%TYPE;
 10      productprice PRODUCTS.PRICE%TYPE;
 11      ITEMSINBAG NUMBER;
 12
 13  BEGIN
 14      DBMS_OUTPUT.PUT_LINE('BUDGET:'||BUDGET||'    '||'FOOD:'||FOODTYPE);
 15      DBMS_OUTPUT.PUT_LINE('');
 16      DBMS_OUTPUT.PUT_LINE('ITEM ID    FOOD    FLAVOR    PRICE');
 17      OPEN ITEMLIST_CURSOR;
 18      LOOP
 19          FETCH ITEMLIST_CURSOR INTO PRODUCTROW;
 20          IF ITEMLIST_CURSOR%NOTFOUND THEN
 21              EXIT;
 22          END IF;
 23          SELECT COUNT(*) INTO FLAG FROM ITEM_LIST WHERE ITEM=PRODUCTROW.PID;
 24          IF FLAG>CNT THEN
 25              CNT:=FLAG;
 26              PIID:=PRODUCTROW.PID;
 27
 28          END IF;
 29          DBMS_OUTPUT.PUT_LINE(PRODUCTROW.PID||'    '||PRODUCTROW.FOOD||'    '||PRODUCT
ROW.FLAVOR||'    '||PRODUCTROW.PRICE);
 30      END LOOP;
 31      CLOSE ITEMLIST_CURSOR;
 32      SELECT FLAVOR INTO FLAVOR FROM PRODUCTS WHERE PID=PIID;
 33      SELECT PRICE INTO productprice FROM PRODUCTS WHERE PID=PIID;
 34      DBMS_OUTPUT.PUT_LINE('');
 35      DBMS_OUTPUT.PUT_LINE('THE ITEM WITH ID '||PIID||' AND '||FLAVOR||' IS THE BEST
CHOICE FOR YOU');
 36      ITEMSINBAG:=BUDGET/productprice;
 37      DBMS_OUTPUT.PUT_LINE('YOU CAN BUY A MAXIMUM OF '||ITEMSINBAG||' ITEM OF THIS
PRODUCT');
 38      DBMS_OUTPUT.PUT_LINE('');
 39
 40  END;
 41  /

Procedure created.

SQL>
SQL> call BUDGETFOOD(10,'Meringue');
BUDGET:10    FOOD:Meringue
ITEM ID    FOOD    FLAVOR    PRICE
70-M-CH-DZ    Meringue    Chocolate    1.25
70-M-VA-SM-DZ    Meringue    Vanilla    1.15
THE ITEM WITH ID 70-M-CH-DZ AND Chocolate IS THE BEST CHOICE FOR YOU
YOU CAN BUY A MAXIMUM OF 8 ITEM OF THIS PRODUCT
```

```
Call completed.


-- 3. Take a receipt number and item as arguments, and insert this information into the
-- Item list. However, if there is already a receipt with that receipt number, then keep
-- adding 1 to the maximum ordinal number. Else before inserting into the Item list
-- with ordinal as 1, ask the user to give the customer name who placed the order and
-- insert this information into the Receipts.
SQL>
SQL>
SQL> --PROCEDURE TO INSERT INTO ITEM_LIST
SQL> CREATE OR REPLACE PROCEDURE INSERT_INTO_ITEM_LIST
  2  (RECEIPTNO_TO_INS IN RECEIPTS.RNO%TYPE, ORDINAL_TO_INS IN ITEM_LIST.ORDINAL%TYPE,
PRODID_TO_INS IN PRODUCTS.PID%TYPE) IS
  3
  4  BEGIN
  5   INSERT INTO ITEM_LIST VALUES(RECEIPTNO_TO_INS,ORDINAL_TO_INS,PRODID_TO_INS);
  6  END;
  7  /

Procedure created.

SQL>
SQL> --PROCEDURE TO UPDATE INTO ITEM_LIST
SQL> CREATE OR REPLACE PROCEDURE UPDATE_INTO_ITEM_LIST
  2  (RECEIPTNO_TO_INS IN ITEM_LIST.RNO%TYPE, PRODID_TO_INS IN ITEM_LIST.ITEM%TYPE) IS
  3
  4  BEGIN
  5   UPDATE ITEM_LIST
  6      SET ORDINAL=ORDINAL+1
  7    WHERE RNO=RECEIPTNO_TO_INS;
  8  END;
  9  /

Procedure created.

SQL>
SQL> --PROCEDURE TO INSERT INTO RECEIPTS
SQL> CREATE OR REPLACE PROCEDURE INSERT_INTO_RECEIPT
  2  (RECEIPTNO_TO_INS IN RECEIPTS.RNO%TYPE, DATE_TO_INS IN
RECEIPTS.RDATE%TYPE,CUSTID_TO_INS IN RECEIPTS.CID%TYPE) IS
  3  BEGIN
  4   INSERT INTO RECEIPTS VALUES(RECEIPTNO_TO_INS,DATE_TO_INS,CUSTID_TO_INS);
  5  END;
  6  /

Procedure created.

SQL>
SQL> --PROCEDURE TO FIND CUSTOMER WITH GIVEN NAME AND RETURN CUSTOMER ID
SQL> CREATE OR REPLACE PROCEDURE SEARCH_CUSTOMER
  2  (CUST_FNAME IN CUSTOMERS.FNAME%TYPE, CUST_LNAME IN CUSTOMERS.LNAME%TYPE,CUSTID_SAVE
OUT CUSTOMERS.CID%TYPE) IS
  3  BEGIN
```

```
  4   BEGIN
  5    SELECT CID INTO CUSTID_SAVE FROM CUSTOMERS
  6    WHERE FNAME=CUST_FNAME AND LNAME=CUST_LNAME;
  7   EXCEPTION
  8    WHEN NO_DATA_FOUND THEN
  9    DBMS_OUTPUT.PUT_LINE('CUSTOMER ID NOT IN CUSTOMER TABLE');
 10    CUSTID_SAVE:=0;
 11   END;
 12  END;
 13  /

Procedure created.

SQL>  select * from item_list where rno=11548;

       RNO    ORDINAL ITEM
---------- ---------- ---------------
     11548          2 45-CO
     11548          3 90-APIE-10
SQL>
SQL> DECLARE
  2   CUSTFNAME CUSTOMERS.FNAME%TYPE;
  3   CUSTLNAME CUSTOMERS.LNAME%TYPE;
  4   CUSTID CUSTOMERS.CID%TYPE;
  5   RECEIPTNO_TO_INS RECEIPTS.RNO%TYPE;
  6   ORDINAL_TO_INS ITEM_LIST.ORDINAL%TYPE;
  7   PRODID_TO_INS PRODUCTS.PID%TYPE;
  8   DATE_TO_INS RECEIPTS.RDATE%TYPE;
  9   ITEM_ROW ITEM_LIST%ROWTYPE;
 10   CURSOR ITEMLIST_CURSOR IS
 11   SELECT * FROM ITEM_LIST WHERE RNO=RECEIPTNO_TO_INS AND ITEM=PRODID_TO_INS;
 12   MAXORDI ITEM_LIST.ORDINAL%TYPE;
 13
 14  BEGIN
 15
 16   RECEIPTNO_TO_INS:='&RECEIPT_NO';
 17   PRODID_TO_INS:='&PRODUCT_ID';
 18   OPEN ITEMLIST_CURSOR;
 19   FETCH ITEMLIST_CURSOR INTO item_row;
 20   IF ITEMLIST_CURSOR%ROWCOUNT>0 THEN
 21    BEGIN
 22    ORDINAL_TO_INS:=item_row.ORDINAL+1;
 23    UPDATE_INTO_ITEM_LIST(RECEIPTNO_TO_INS,PRODID_TO_INS);
 24    RETURN;
 25    END;
 26   ELSE
 27    BEGIN
 28    DBMS_OUTPUT.PUT_LINE('RECEIPT NOT FOUND');
 29    DBMS_OUTPUT.PUT_LINE('CREATING NEW RECEIPT');
 30    CUSTFNAME:='&FIRST_NAME';
 31    CUSTLNAME:='&LAST_NAME';
 32    DATE_TO_INS:='&DATE';
 33    SEARCH_CUSTOMER(CUSTFNAME,CUSTLNAME,CUSTID);
 34    INSERT_INTO_RECEIPT(RECEIPTNO_TO_INS,DATE_TO_INS,CUSTID);
```

```
 35    ORDINAL_TO_INS:=1;
 36    INSERT_INTO_ITEM_LIST(RECEIPTNO_TO_INS,ORDINAL_TO_INS,PRODID_TO_INS);
 37    RETURN;
 38    END;
 39   END IF;
 40  END;
 41  /
Enter value for receipt_no: 11548
old  16:  RECEIPTNO_TO_INS:='&RECEIPT_NO';
new  16:  RECEIPTNO_TO_INS:='11548';
Enter value for product_id: 45-CO
old  17:  PRODID_TO_INS:='&PRODUCT_ID';
new  17:  PRODID_TO_INS:='45-CO';
Enter value for first_name: nil
old  30:   CUSTFNAME:='&FIRST_NAME';
new  30:   CUSTFNAME:='nil';
Enter value for last_name: nil
old  31:   CUSTLNAME:='&LAST_NAME';
new  31:   CUSTLNAME:='nil';
Enter value for date: nil
old  32:   DATE_TO_INS:='&DATE';
new  32:   DATE_TO_INS:='nil';

PL/SQL procedure successfully completed.

SQL>  select * from item_list where rno=11548;

       RNO    ORDINAL ITEM
---------- ---------- ---------------
     11548          3 45-CO
     11548          4 90-APIE-10

SQL> rollback;

Rollback complete.

-- 4. Write a stored function to display the customer name who ordered maximum for the
-- given food and flavor.
SQL> CREATE OR REPLACE FUNCTION MOST_CUST
  2  (GIVENFOOD PRODUCTS.FOOD%TYPE, GIVENFLAVOR PRODUCTS.FLAVOR%TYPE)
  3  RETURN CUSTOMER.CNAME%TYPE
  4   AS
  5
  6      CUST_NAME CUSTOMER.CNAME%TYPE;
  7
  8  BEGIN
  9      SELECT FNAME ||' '|| LNAME
 10      INTO CUST_NAME
 11      FROM CUSTOMERS
 12      WHERE CID = (
 13          SELECT CID FROM (
 14           SELECT A.CID, COUNT(*)
 15           FROM CUSTOMERS A, PRODUCTS B, ITEM_LIST C, RECEIPTS D
```

```
 16           WHERE A.CID = D.CID AND C.RNO = D.RNO AND C.ITEM = B.PID AND LOWER(B.FOOD) =
LOWER(GIVENFOOD) AND LOWER(FLAVOR) = LOWER(GIVENFLAVOR)
 17           GROUP BY A.CID ORDER BY 2 DESC)
 18          WHERE ROWNUM = 1);
 19
 20  RETURN CUST_NAME;
 21  END;
 22  /

Function created.

SQL>
SQL> ---CALLING FUNCTION FROM ANONYMOUS BLOCK
SQL>
SQL> DECLARE
  2
  3   CUSTNAME varchar(50);
  4   prodid_to_ins PRODUCTS.PID%TYPE;
  5   GIVENFOOD PRODUCTS.FOOD%TYPE;
  6   GIVENFLAVOR PRODUCTS.FLAVOR%TYPE;
  7
  8  BEGIN
  9
 10   GIVENFOOD:='&FOOD';
 11   GIVENFLAVOR:='&FLAVOR';
 12   CUSTNAME:=MOST_CUST(GIVENFOOD,GIVENFLAVOR);
 13   DBMS_OUTPUT.PUT_LINE(CUSTNAME||' BOUGHT MAXIMUM OF '||UPPER(GIVENFOOD)||' IN
'||UPPER(GIVENFLAVOR)||' FLAVOR');
 14  END;
 15  /
Enter value for food: cake
old  10:  GIVENFOOD:='&FOOD';
new  10:  GIVENFOOD:='cake';
Enter value for flavor: lemon
old  11:  GIVENFLAVOR:='&FLAVOR';
new  11:  GIVENFLAVOR:='lemon';
ESPOSITA TRAVIS BOUGHT MAXIMUM OF CAKE IN LEMON FLAVOR

PL/SQL procedure successfully completed.

-- 5. Implement Question (1) using stored function to return the amount to be paid and
-- update the same, for the given receipt number.
SQL>
SQL> ---CREATING FUNCTION
SQL> CREATE OR REPLACE FUNCTION CALC_DISCOUNT_FUNC
  2  (PRODUCT_PRICE IN PRODUCTS.PRICE%TYPE,DISC_PER OUT PRODUCTS.PRICE%TYPE,DISCOUNT_PRICE
OUT PRODUCTS.PRICE%TYPE)
  3  RETURN PRODUCTS.PRICE%TYPE IS
  4
  5  FINAL_PRICE PRODUCTS.PRICE%TYPE;
  6
  7  BEGIN
  8   IF PRODUCT_PRICE>10 AND PRODUCT_PRICE<25 THEN
  9    DISCOUNT_PRICE:=0.05*PRODUCT_PRICE;
```

```
 10    DISC_PER:=5;
 11   END IF;
 12   IF PRODUCT_PRICE>25 AND PRODUCT_PRICE<50 THEN
 13    DISCOUNT_PRICE:=0.1*PRODUCT_PRICE;
 14    DISC_PER:=10;
 15   END IF;
 16   IF PRODUCT_PRICE>50 THEN
 17    DISCOUNT_PRICE:=.2*PRODUCT_PRICE;
 18    DISC_PER:=20;
 19   END IF;
 20   FINAL_PRICE:=PRODUCT_PRICE-DISCOUNT_PRICE;
 21   RETURN FINAL_PRICE;
 22  END;
 23  /

Function created.

SQL>
SQL> DECLARE
  2      TOTAL PRODUCTS.PRICE%TYPE;
  3      PERCENT INT;
  4      DISC PRODUCTS.PRICE%TYPE;
  5      DOP DATE;
  6      SNO INT := 0;
  7      CUST_FNAME CUSTOMERS.FNAME%TYPE;
  8      CUST_LNAME CUSTOMERS.LNAME%TYPE;
  9      RECEIPT_NO_SEARCH RECEIPTS.RNO%TYPE;
 10      CURSOR ITEM_CURSOR IS
 11      SELECT FLAVOR, FOOD, PRICE FROM PRODUCTS, ITEM_LIST
 12      WHERE PRODUCTS.PID = ITEM_LIST.ITEM AND ITEM_LIST.RNO = RECEIPT_NO_SEARCH;
 13      TEMP_ITEM ITEM_CURSOR%ROWTYPE;
 14      FINAL_PRICE PRODUCTS.PRICE%TYPE;
 15
 16
 17  BEGIN
 18      RECEIPT_NO_SEARCH:='&RECEIPT_NUM';
 19      SELECT SUM(PRICE) INTO TOTAL FROM ITEM_LIST, PRODUCTS
 20   WHERE ITEM_LIST.ITEM = PRODUCTS.PID and ITEM_LIST.RNO = RECEIPT_NO_SEARCH;
 21   DBMS_OUTPUT.PUT_LINE('RECEIPT NUMBER : ' || RECEIPT_NO_SEARCH);
 22
 23   SELECT RDATE INTO DOP FROM RECEIPTS
 24   WHERE RNO = RECEIPT_NO_SEARCH;
 25
 26   DBMS_OUTPUT.PUT_LINE('RECEIPT DATE : ' || DOP);
 27
 28   SELECT FNAME, LNAME INTO CUST_FNAME, CUST_LNAME FROM CUSTOMERS, RECEIPTS
 29   WHERE CUSTOMERS.CID = RECEIPTS.CID AND RECEIPTS.RNO = RECEIPT_NO_SEARCH;
 30
 31   DBMS_OUTPUT.PUT_LINE('CUSTOMER NAME : ' || CUST_FNAME || ' ' || CUST_LNAME);
 32   DBMS_OUTPUT.PUT_LINE('FLAVOR     FOOD     PRICE');
 33      OPEN ITEM_CURSOR;
 34   LOOP
 35    FETCH ITEM_CURSOR INTO TEMP_ITEM;
 36    IF ITEM_CURSOR%NOTFOUND THEN
```

```
 37     EXIT;
 38    END IF;
 39    DBMS_OUTPUT.PUT_LINE(TEMP_ITEM.FLAVOR || '     ' || TEMP_ITEM.FOOD || '     ' ||
TEMP_ITEM.PRICE);
 40   END LOOP;
 41      CLOSE ITEM_CURSOR;
 42
 43      FINAL_PRICE:=CALC_DISCOUNT_FUNC(TOTAL,PERCENT,DISC);
 44   DBMS_OUTPUT.PUT_LINE('TOTAL AMOUNT : $' || TOTAL);
 45   DBMS_OUTPUT.PUT_LINE('DISCOUNT ' || PERCENT ||'% : $' || DISC);
 46   DBMS_OUTPUT.PUT_LINE('AMOUNT TO BE PAID : $' ||FINAL_PRICE );
 47  END;
 48  /
Enter value for receipt_num: 13355
old  18:     RECEIPT_NO_SEARCH:='&RECEIPT_NUM';
new  18:     RECEIPT_NO_SEARCH:='13355';
RECEIPT NUMBER : 13355
RECEIPT DATE : 19-OCT-07
CUSTOMER NAME : TOUSSAND SHARRON
FLAVOR     FOOD     PRICE
Opera     Cake     15.95
Lemon     Cookie     .79
Napoleon     Cake     13.49
TOTAL AMOUNT : $30.23
DISCOUNT 10% : $3.023
AMOUNT TO BE PAID : $27.207

PL/SQL procedure successfully completed.

SQL> spool off;
```