CSD 4204 Project

Use the provided appbDbcreate.sql SQL script to create the database to be used when working on this assignment.

Question 1. Create Product ID and Product Name Report (2 Marks)

Create a report by creating a cursor named product_cursor that will fetch IDPRODUCT and PRODUCTNAME from BB_PRODUCT table into two variables that you will declare. Use for loop to print all the product id's and product names.

SOLUTION-

```
DECLARE
  CURSOR PRODUCT_CURSOR
  IS
    SELECT
      IDPRODUCT,
      PRODUCTNAME
    FROM
      BB_PRODUCT;
BEGIN
  FOR PC IN PRODUCT_CURSOR
  LOOP
    DBMS_OUTPUT.PUT_LINE(PC.IDPRODUCT || ' ' || PC.PRODUCTNAME );
  END LOOP;
END;
/
```

```
Statement processed.
1 CapressoBar Model #351
2 Capresso Ultima
3 Eileen 4-cup French Press
4 Coffee Grinder
5 Sumatra
6 Guatamala
7 Columbia
8 Brazil
9 Ethiopia
10 Espresso
```

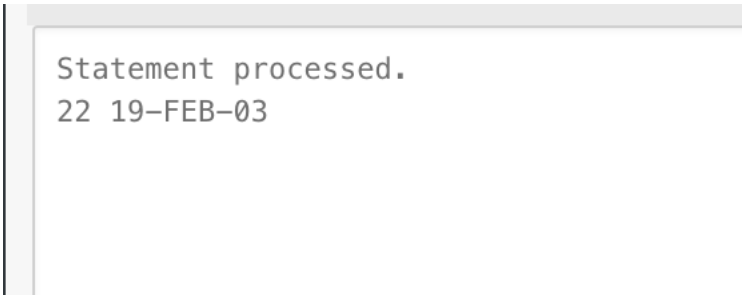
Question 2. Create a report that retrieving Product Name (3 Marks)
Retrieving the names of products from BB_PRODUCT table and print each product on the screen, incorporating associative array. Declare index by table prod_table_type of type bb_product.productname. Declare a variable that will temporarily store the product name. Use loop to print the result.

```
DECLARE
TYPE PRODUCT_NAME_TYPE
IS
  TABLE OF BB_PRODUCT.PRODUCTNAME%TYPE INDEX BY PLS_INTEGER;
  PRODUCT_NAME PRODUCT_NAME_TYPE;
BEGIN
  FOR I IN
  (
    SELECT
      PRODUCTNAME
    FROM
      BB_PRODUCT
  )
  LOOP
    DBMS_OUTPUT.PUT_LINE(I.PRODUCTNAME);
  END LOOP;
END;
```

```
Statement processed.
CapressoBar Model #351
Capresso Ultima
Eileen 4-cup French Press
Coffee Grinder
Sumatra
Guatamala
Columbia
Brazil
Ethiopia
Espresso
```

Question 3. Create a report for Customer basket and the date of creation(2)
Declare PL/SQL records based structure of bb_basket table that will display the values of idshopper and dtcreation of idbasket 5.

```
DECLARE
TYPE BASKET_TABLE_TYPE
IS
  TABLE OF BB_BASKET%ROWTYPE INDEX BY PLS_INTEGER;
  BASKET_TABLE BASKET_TABLE_TYPE;
BEGIN
  SELECT
    *
  INTO
    BASKET_TABLE(1)
  FROM
    BB_BASKET
  WHERE
    IDBASKET = 5;
  DBMS_OUTPUT.PUT_LINE(BASKET_TABLE(1).IDSHOPPER ||' '|| BASKET_TABLE(1)
  .DTCREATED);
END;
/
```

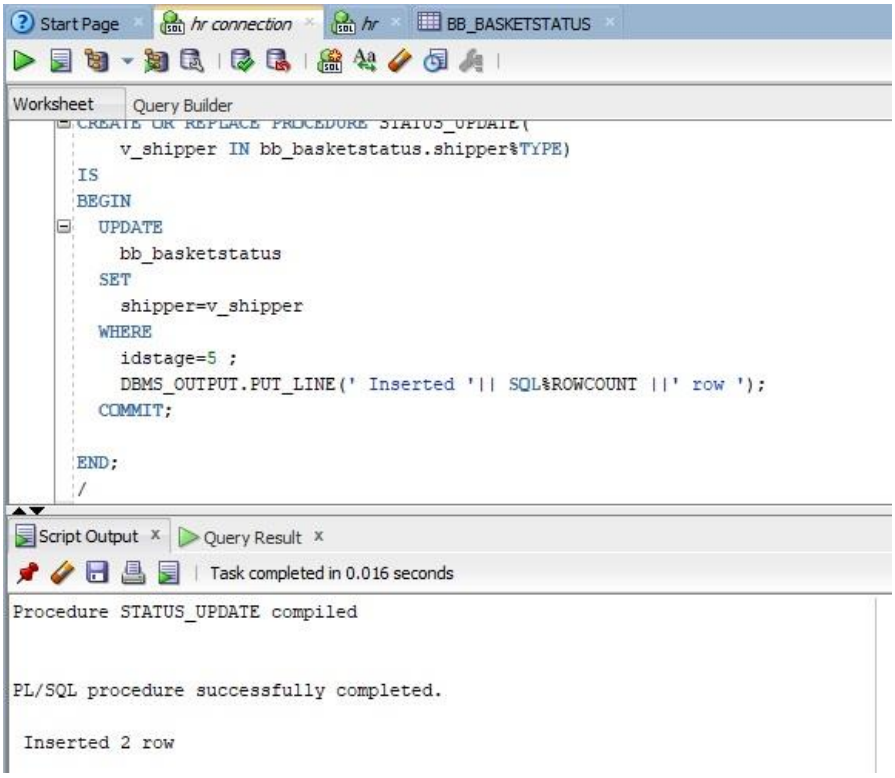


Question 4. Update the Status of an Order(3 Marks)

Create a procedure named STATUS_UPDATE that allows a company to employee in the Shipping Department to update the status of an order to add shipping information.

The procedure should allow the update of a row indicating an IDSTAGE of 5, to have 'DHL' as value of the column SHIPPER.
Invoke the procedure with proper statement

```
CREATE OR REPLACE PROCEDURE STATUS_UPDATE(
          V_SHIPPER IN BB_BASKETSTATUS.SHIPPER%TYPE)
IS
BEGIN
 UPDATE BB_BASKETSTATUS SET SHIPPER=V_SHIPPER WHERE IDSTAGE=5 ;
 COMMIT;
DBMS_OUTPUT.PUT_LINE(' INSERTED '|| SQL%ROWCOUNT ||' ROW ');
END;
/

DECLARE
 V_SHIPPER VARCHAR2(10);
BEGIN
  V_SHIPPER:='DHL';
  STATUS_UPDATE(V_SHIPPER);
END;
```



Question 5. Identify Customers (3 Marks)

The company noticed that they have few orders from customers in the state of 'NY' and they want to offer an incentive of free shipping to those customers .Create a procedure named PROMO_SHIP that determines who these customers are and then updates the BB_PROMOLIST table accordingly. The procedure uses the following information:

1. Date cutoff = Any customers who have not shopped on the site since this date should be included as incentive participants. Use the basket creation date to reflect shopper activity dates.
2. Month = Three-character month (such as APR) that should be added to the promotion table to indicate which month the free shipping is available.
3. Year = Four-digit year indicating the year the promotion is effective.
4. PROMO_FLAG = 1 (representing free shipping).

The BB_PROMOLIST table also has a USED column, which contains a default value of "N" and is updated to a "Y" when the shopper uses the promotion. Test the procedure with a cutoff date of 20-FEB-03. Assign the free shipping for the month of APR and the year 2003.

The BB_shopper table includes customer information.

ANS-

```
CREATE OR REPLACE PROCEDURE PROMO_SHIP(V_DATE
BB_BASKET.DTCREATED%TYPE,
V_MONTH BB_PROMOLIST.MONTH%TYPE,V_YEAR BB_PROMOLIST.YEAR%TYPE) IS
V_SHOPPER_ID BB_SHOPPER.IDSHOPPER%TYPE;
V_DATE_CREATED BB_BASKET.DTCREATED%TYPE;
V_PROMOFLAG INT(20) :=1;
V_USED VARCHAR(20) :='N';
BEGIN
SELECT IDSHOPPER INTO V_SHOPPER_ID FROM BB_SHOPPER WHERE STATE='NY' ;
SELECT IDSHOPPER INTO V_SHOPPER_ID FROM BB_BASKET WHERE
IDSHOPPER=V_SHOPPER_ID AND DTCREATED <V_DATE;

INSERT INTO BB_PROMOLIST(IDSHOPPER, MONTH, YEAR, PROMO_FLAG, USED)
VALUES(V_SHOPPER_ID, V_MONTH,V_YEAR, V_PROMOFLAG, V_USED);
DBMS_OUTPUT.PUT_LINE(' INSERTED '|| SQL%ROWCOUNT
||' ROW ');
END;
/
BEGIN
  PROMO_SHIP('20-FEB-03','APR', 2003);
END;
```

NOTE- No row has state value 'NY' in table.

```
BEGIN
   PROMO_SHIP('20-FEB-03','APR', 2003);
END;
Error report -
ORA-01403: no data found
ORA-06512: at "SYSTEM.PROMO_SHIP", line 8
ORA-06512: at line 2
01403. 00000 -  "no data found"
*Cause:    No data was found from the objects.
*Action:   There was no data from the objects which may be due to end of fetch.
```

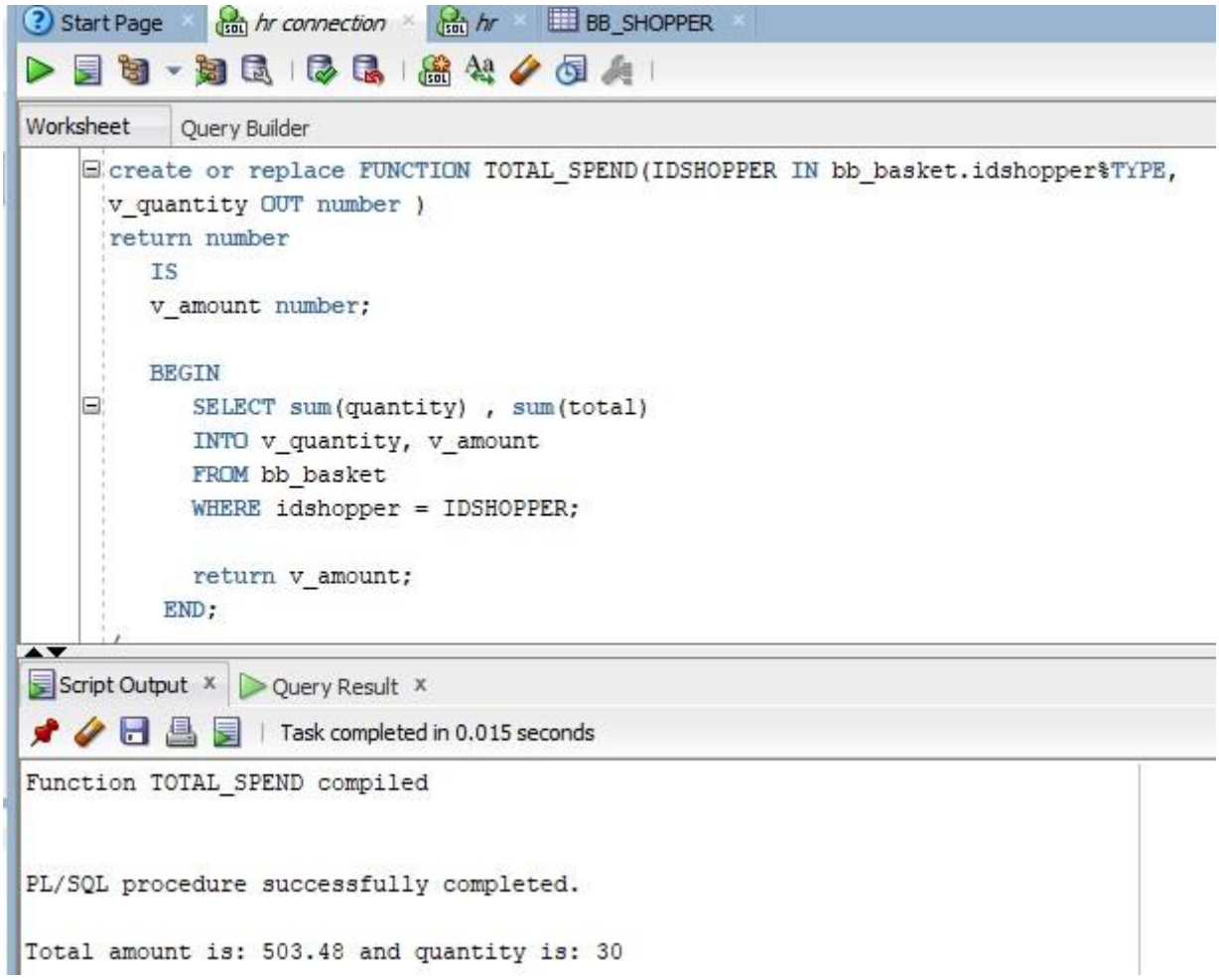
Question 6. Calculate the Total Shopper Spending (3 Marks)

Create a function TOTAL_SPEND that will require to enter IDSHOPPER as an input parameter and return total of his/her quantity ordered and total amount spent with tax amount included.

Run your function with proper IDSHOPPER.
Refer to bb_basket and bb_shopper tables for your function

```
CREATE OR REPLACE FUNCTION TOTAL_SPEND(IDSHOPPER IN
BB_BASKET.IDSHOPPER%TYPE,
V_QUANTITY OUT NUMBER )
RETURN NUMBER
  IS
  V_AMOUNT NUMBER;

  BEGIN
    SELECT SUM(QUANTITY) , SUM(TOTAL)
    INTO V_QUANTITY, V_AMOUNT
    FROM BB_BASKET
    WHERE IDSHOPPER = IDSHOPPER;

    RETURN V_AMOUNT;
   END;
/

DECLARE
V_ID INTEGER;
AMOUNT NUMBER;
QUANTITY NUMBER;
BEGIN
V_ID := 21;

AMOUNT :=TOTAL_SPEND(V_ID, QUANTITY);

DBMS_OUTPUT.PUT_LINE('TOTAL AMOUNT IS: '||AMOUNT ||' AND QUANTITY IS: '||
QUANTITY);
```

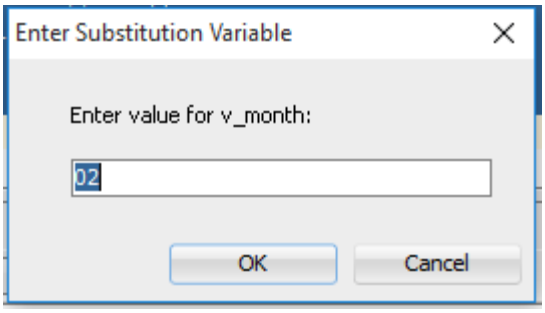
```
END;
/
```


Start Page
hr connection
hr
BB_SHOPPER
Worksheet
Query Builder
create or replace FUNCTION TOTAL_SPEND(IDSHOPPER IN bb_basket.idshopper%TYPE,
v_quantity OUT number)
return number
IS
v_amount number;
BEGIN
SELECT sum(quantity) , sum(total)
INTO v_quantity, v_amount
FROM bb_basket
WHERE idshopper = IDSHOPPER;
return v_amount;
END;
Script Output
Query Result
Task completed in 0.015 seconds
Function TOTAL_SPEND compiled
PL/SQL procedure successfully completed.
Total amount is: 503.48 and quantity is: 30

Question 7. Identify the customer with highest number of order placed (3 Marks)

Create a function ORD_PLACED that will identify the customer with highest number of order placed in each month. Refer to table BB_BASKET that include IDSHOPPER column that represents customer id and ORDERPLACED that show number of the order placed by the customer. Display the idshopper and the total of his order placed

```
CREATE OR REPLACE FUNCTION ORD_PLACED(V_ID OUT
BB_BASKET.IDSHOPPER%TYPE, V_MONTH IN NUMBER)
  RETURN NUMBER IS
V_ORDERPLACED NUMBER;

BEGIN
  SELECT IDSHOPPER, SUM(ORDERPLACED)
  INTO V_ID, V_ORDERPLACED
  FROM BB_BASKET
  GROUP BY IDSHOPPER, TO_CHAR(DTCREATED,'MM')
  HAVING SUM(ORDERPLACED) = ( SELECT MAX(SUM(ORDERPLACED))
  FROM BB_BASKET
  WHERE TO_CHAR(DTCREATED,'MM') = V_MONTH
  GROUP BY IDSHOPPER) AND TO_CHAR(DTCREATED,'MM') = V_MONTH;

  RETURN V_ORDERPLACED;
END;
/

SET SERVEROUTPUT ON;
DECLARE
  V_ORDERPLACED NUMBER;
  V_ID         INT;
  V_MONTH NUMBER;
BEGIN
  V_MONTH := '&V_MONTH';
  V_ORDERPLACED := ORD_PLACED(V_ID, V_MONTH);
   DBMS_OUTPUT.PUT_LINE('SHOPPER ID:'||V_ID);
   DBMS_OUTPUT.PUT_LINE('ORDER PLACED:'|| V_ORDERPLACED);
END;
/
```



```
Function ORD_PLACED compiled

old:DECLARE
  v_orderplaced NUMBER;
  v_id          INT;
  v_month NUMBER;
BEGIN
  v_month := '&v_month';
  v_orderplaced := ord_placed(v_id, v_month);
   dbms_output.put_line('Shopper ID:'||v_id);
   dbms_output.put_line('Order Placed:'|| v_orderplaced);
END;
new:DECLARE
  v_orderplaced NUMBER;
  v_id          INT;
  v_month NUMBER;
BEGIN
  v_month := '02';
  v_orderplaced := ord_placed(v_id, v_month);
   dbms_output.put_line('Shopper ID:'||v_id);
   dbms_output.put_line('Order Placed:'|| v_orderplaced);
END;
Shopper ID:24
Order Placed:2
```

Question 8. Perform Exception Handling with User-Defined Errors (3 Marks)

In order to increase the sales of ID PRODUCT number 8 which is “Brazil” the management requested that to each order of “Brazil” coffee should be more than 2 quantity.
Create an exception that will be raised if any entered order contains IDPRODUCT 8 with less than 2 quantity and display the message “Minimum quantity order of Brazilian Coffee is 2”. Test your exception with insert statement into BB_BASKETITEM table.
Refer to bb_product table in order to find related IDPRODUCT column information

```
CREATE OR REPLACE TRIGGER RESTRICT_QUANTITY
BEFORE INSERT ON BB_BASKETITEM
FOR EACH ROW
BEGIN
IF (:NEW.IDPRODUCT = 8) AND (:NEW.QUANTITY < 2) THEN
RAISE_APPLICATION_ERROR(-20500, 'MINIMUM QUANTITY ORDER OF BRAZILIAN
COFFEE IS 2.');
END IF;
END;


INSERT INTO BB_BASKETITEM VALUES (39, 8, 5.00, 1, 3, 1, 4);
```

```
Trigger RESTRICT_QUANTITY compiled

Error starting at line : 11 in command -
insert into bb_basketItem values (39, 8, 5.00, 1, 3, 1, 4)
Error report -
SQL Error: ORA-20500: Minimum quantity order of Brazilian Coffee is 2.
ORA-06512: at "SYSTEM.RESTRICT_QUANTITY", line 3
ORA-04088: error during execution of trigger 'SYSTEM.RESTRICT_QUANTITY'
```

Question 9. Refraining users to add new product information (3 Marks)
The Management wants to refrain the users from adding new coffee product in bb_product table. Create a triggers PROD_TRG that will stop the users from entering new product and display a message "You are not allowed new Product in BB_PRODUCT table".

```
CREATE OR REPLACE TRIGGER PROD_TRG
BEFORE INSERT ON BB_PRODUCT
BEGIN

RAISE_APPLICATION_ERROR (-20202,
'YOU ARE NOT ALLOWED NEW PRODUCT IN BB_PRODUCT TABLE.');

END;

INSERT INTO BB_PRODUCT(IDPRODUCT, TYPE, PRODUCTNAME, DESCRIPTION,
PRODUCTIMAGE, PRICE, ACTIVE, IDDEPARTMENT)
  VALUES(11,'E','CAPRESSOBAR MODEL #351', 'A FULLY PROGRAMMABLE PUMP
ESPRESSO MACHINE AND 10-CUP COFFEE MAKER COMPLETE WITH GOLDTONE
FILTER', 'CAPRESSO.GIF', 99.99, 1, 2);
```

```
ORA-20202: You are not allowed new Product in BB_PRODUCT table. ORA-06512: at
"SQL_HGQDHXWJEFUFNUKEIVSDARCXT.PROD_TRG", line 3 ORA-06512: at "SYS.DBMS_SQL", line 1721
```