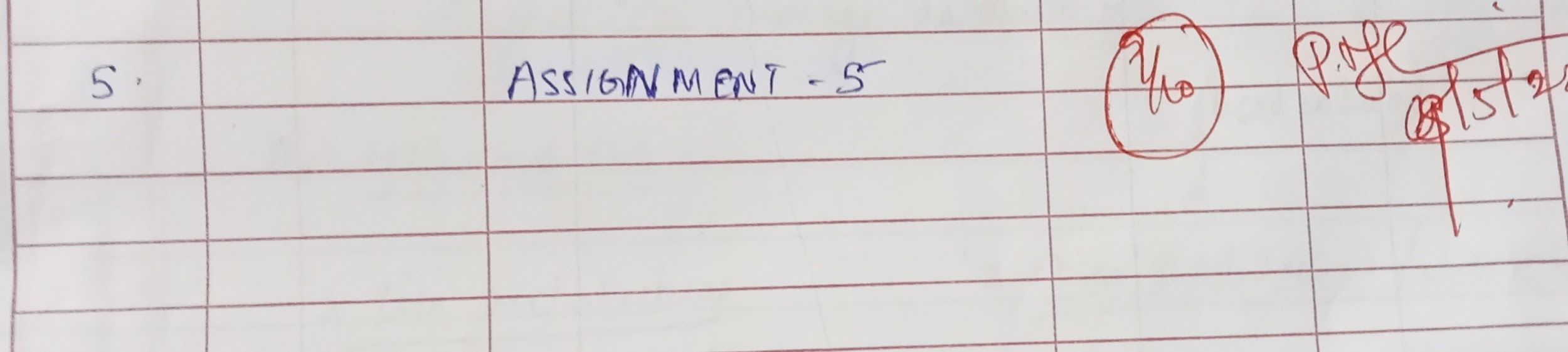
**UCS 1412 - DATABASE LAB**

**Assignment 5 -PL/SQL CONTROL STRUCTURES**

**DIKSHITHA VANI V**

**205001032**

**INDEX PAGE:**

****

**SQL SPOOL FILE:**

SQL> @ Z:\csea\_a\_1032\assig5\assig5.sql

SQL> REM \*\* Write PL/SQL block for the following: \*\*

SQL>

SQL> REM \*\* 1.Check whether the given combination of food and flavor is available. If any one or

SQL> REM both are not available, display the relevant message. \*\*

SQL>

SQL> REM \* Given combination of food and flavor is available in the table \*

SQL>

SQL> declare

2 record products%rowtype;

3 given\_food products.food%type;

4 given\_flavor products.flavor%type;

5 counting int;

6 begin

7 given\_food:='&food\_to\_search';

8 given\_flavor:='&flavor\_to\_search';

9

10 select count(\*) into counting from products where food=given\_food and flavor=given\_flavor;

11 if counting != 0 then

12 dbms\_output.put\_line('Given '|| given\_food||' and '||given\_flavor||' combination is available!!');

13 select \* into record from products where food=given\_food and flavor=given\_flavor;

14 dbms\_output.put\_line('Product id: '||record.pid);

15 else

16 dbms\_output.put\_line('Given combination is not found !');

17 select count(\*) into counting from (select \* from products where given\_food = food);

18 if counting != 0 then

19 dbms\_output.put\_line(given\_food||' found but does not contain '||given\_flavor);

20 else

21 dbms\_output.put\_line(given\_food||' not found');

22 end if;

23 select count(\*) into counting from (select \* from products where given\_flavor = flavor);

24 if counting != 0 then

25 dbms\_output.put\_line(given\_flavor||' found but not containted in '||given\_food);

26 else

27 dbms\_output.put\_line(given\_flavor||' not found');

28 end if;

29 end if;

30 end;

31 /

Enter value for food\_to\_search: Cake

old 7: given\_food:='&food\_to\_search';

new 7: given\_food:='Cake';

Enter value for flavor\_to\_search: Chocolate

old 8: given\_flavor:='&flavor\_to\_search';

new 8: given\_flavor:='Chocolate';

Given Cake and Chocolate combination is available!!

Product id: 20-BC-C-10

PL/SQL procedure successfully completed.

SQL> /

Enter value for food\_to\_search: Chocolate

old 7: given\_food:='&food\_to\_search';

new 7: given\_food:='Chocolate';

Enter value for flavor\_to\_search: Chocolate

old 8: given\_flavor:='&flavor\_to\_search';

new 8: given\_flavor:='Chocolate';

Given combination is not found !

Chocolate not found

Chocolate found but not containted in Chocolate

PL/SQL procedure successfully completed.

SQL> /

Enter value for food\_to\_search: Tart

old 7: given\_food:='&food\_to\_search';

new 7: given\_food:='Tart';

Enter value for flavor\_to\_search: Strawberry

old 8: given\_flavor:='&flavor\_to\_search';

new 8: given\_flavor:='Strawberry';

Given combination is not found !

Tart found but does not contain Strawberry

Strawberry found but not containted in Tart

PL/SQL procedure successfully completed.

SQL> /

Enter value for food\_to\_search: Cake

old 7: given\_food:='&food\_to\_search';

new 7: given\_food:='Cake';

Enter value for flavor\_to\_search: Vanilla

old 8: given\_flavor:='&flavor\_to\_search';

new 8: given\_flavor:='Vanilla';

Given combination is not found !

Cake found but does not contain Vanilla

Vanilla found but not containted in Cake

PL/SQL procedure successfully completed.

SQL> declare

2 counting int;

3 date\_search Receipts.rdate%type;

4 record products%rowtype;

5 begin

6 date\_search:='&date\_search';

7 select count(item) into counting from item\_list i join receipts r on i.rno=r.rno where rdate=date\_search;

8 if SQL%FOUND then

9 dbms\_output.put\_line(chr(9));

10 dbms\_output.put\_line('Number of items sold on '||date\_search||' is '||counting);

11 else

12 dbms\_output.put\_line('Given date has no items bought on the particular date');

13 end if;

14 end;

15 /

Enter value for date\_search: 17-OCT-2007

old 6: date\_search:='&date\_search';

new 6: date\_search:='17-OCT-2007';

Number of items sold on 17-OCT-07 is 11

PL/SQL procedure successfully completed.

SQL> REM \*\* 3 .An user desired to buy the product with the specific price. Ask the user for a price, find the food item(s) that is equal or closest to the desired price.

REM Print the product number, food type, flavor and price. Also print the number of items that is equal or closest to the desired price. \*\*

SQL> declare

2 price\_search products.price%type;

3 record products%rowtype;

4 cursor c\_products is select \* from products where abs(price-price\_search) = (

5 select min(abs(price-price\_search)) from products);

6 begin

7 price\_search:='&price\_search';

8 open c\_products;

9 dbms\_output.put\_line(chr(9));

10 dbms\_output.put\_line('PRODUCT\_ID FOOD FLAVOR PRICE');

11 dbms\_output.put\_line('-------------------------------------------------');

12 loop

13 fetch c\_products into record;

14 if c\_products%NOTFOUND THEN

15 dbms\_output.put\_line('-------------------------------------------------');

16 exit;

17 else

18 dbms\_output.put\_line(record.pid||' '||record.food||' '||record.flavor||' '||record.price);

19 end if;

20 end loop;

21 dbms\_output.put\_line(c\_products%rowcount||' product(s) found EQUAL/CLOSEST to given price.');

22 close c\_products;

23 end;

24 /

Enter value for price\_search: 0.8

old 7: price\_search:='&price\_search';

new 7: price\_search:='0.8';

PRODUCT\_ID FOOD FLAVOR PRICE

-------------------------------------------------

70-LEM Cookie Lemon .79

70-W Cookie Walnut .79

-------------------------------------------------

2 product(s) found EQUAL/CLOSEST to given price.

PL/SQL procedure successfully completed.

SQL>

SQL> REM \*\* 4 Display the customer name along with the details of item and its quantity ordered for the given order number. Also calculate the total quantity ordered

REM as shown below: \*\*

SQL> declare

2 order\_number item\_list.rno%type;

3 cust\_fname customers.fname%type;

4 cust\_lname customers.lname%type;

5 total\_qty number(3):=0;

6 cursor details is select food,flavor,count(\*) as qty from item\_list,products where rno=order\_number and pid=item group by food,flavor;

7 record details%rowtype;

8 begin

9 order\_number:='&order\_number\_to\_search';

10 select fname,lname into cust\_fname,cust\_lname from customers inner join receipts using(cid) where rno=order\_number;

11 dbms\_output.put\_line(chr(9));

12 dbms\_output.put\_line('Customer name: '||cust\_fname||' '||cust\_lname);

13 dbms\_output.put\_line(chr(9));

14 open details;

15 dbms\_output.put\_line('FOOD'||chr(9)||chr(9)||'FLAVOR'||chr(9)||chr(9)||'QUANTITY');

16 dbms\_output.put\_line('-------------------------------------------');

17 loop

18 fetch details into record;

19 if details%NOTFOUND THEN

20 exit;

21 else

22 if length(record.flavor) < 8 then

23 dbms\_output.put\_line(record.food||chr(9)||chr(9)||record.flavor||chr(9)||chr(9)||record.qty);

24 else

25 dbms\_output.put\_line(record.food||chr(9)||chr(9)||record.flavor||chr(9)||record.qty);

26 end if;

27 total\_qty:=total\_qty+record.qty;

28 end if;

29 end loop;

30 dbms\_output.put\_line('-------------------------------------------');

31 dbms\_output.put\_line('TOTAL QUANTITY '||total\_qty);

32 close details;

33 exception

34 when NO\_DATA\_FOUND THEN

35 dbms\_output.put\_line(chr(9));

36 dbms\_output.put\_line('Invalid order number!');

37 end;

38 /

Enter value for order\_number\_to\_search: 51991

old 9: order\_number:='&order\_number\_to\_search';

new 9: order\_number:='51991';

Customer name: SOPKO RAYFORD

FOOD FLAVOR QUANTITY

-------------------------------------------

Pie Apple 1

Cake Truffle 1

Tart Apple 1

Tart Chocolate 1

-------------------------------------------

TOTAL QUANTITY 4

PL/SQL procedure successfully completed.

SQL> spool off;