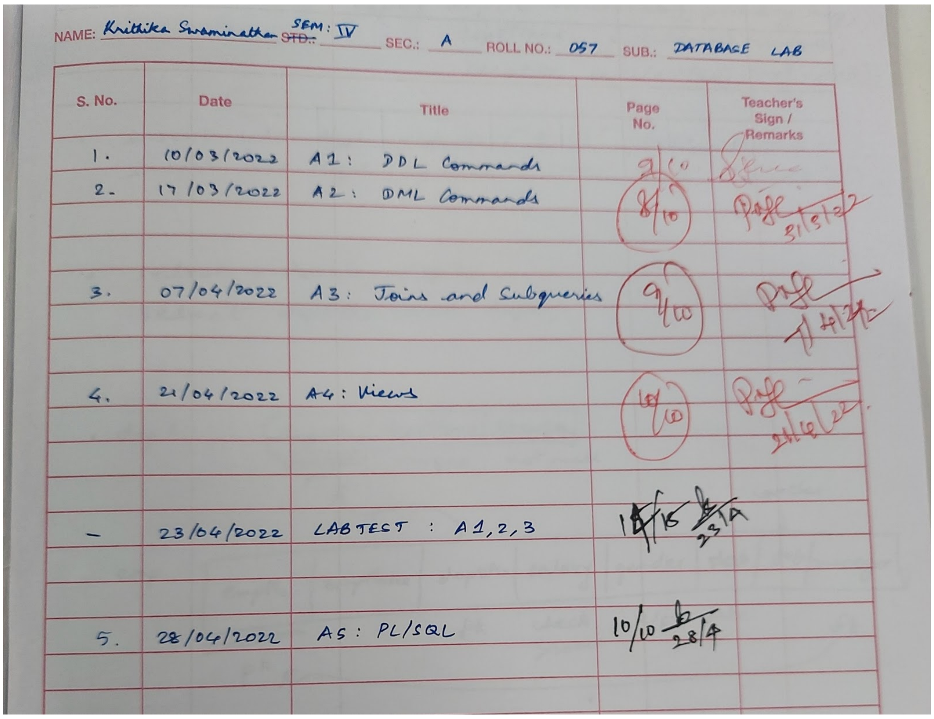
Assignment 5 – PL/SQL: Structures

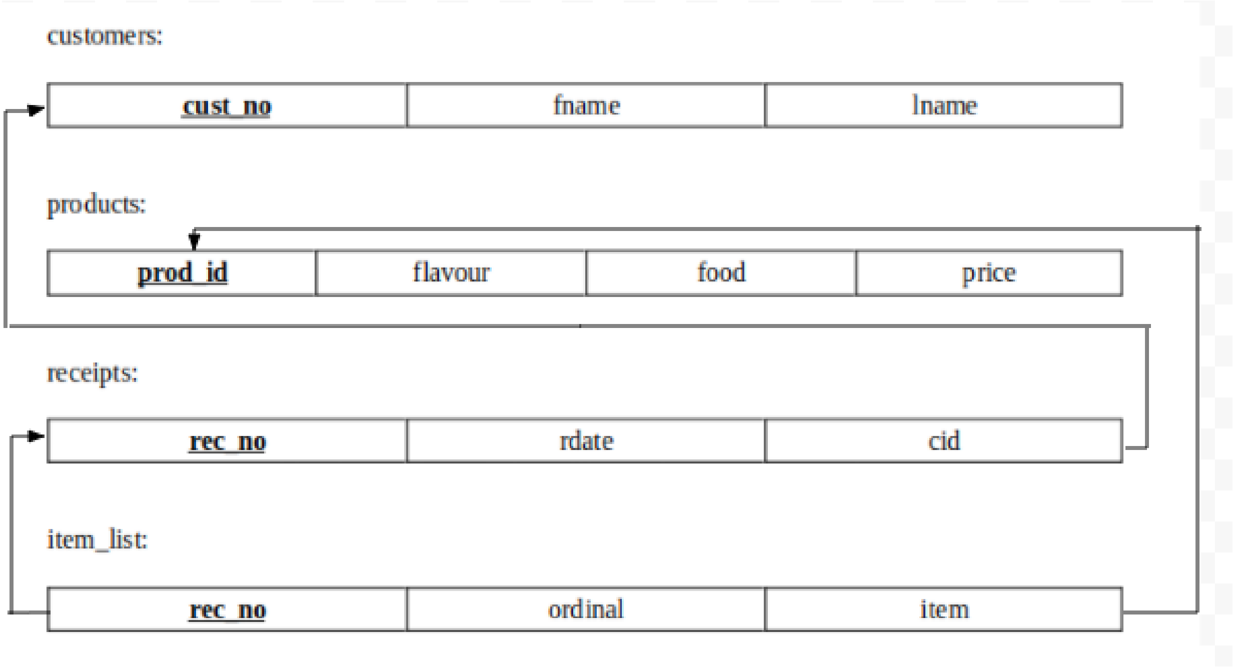
Validation:



NAME: Krithika Swaminathan ~~STD.:~~ SEM: IV SEC.: A ROLL NO.: 057 SUB.: DATABASE LAB

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
1.	10/03/2022	A1: DDL Commands	9/10	
2.	17/03/2022	A2: DML Commands	8/10	31/3/22
3.	07/04/2022	A3: Joins and Subqueries	9/10	7/4/22
4.	21/04/2022	A4: Views	10/10	21/4/22
–	23/04/2022	LAB TEST : A1,2,3	14/15	23/4
5.	28/04/2022	A5: PL/SQL	10/10	28/4

Schema diagram:



customers:

cust_no	fname	lname

products:

prod_id	flavour	food	price

receipts:

rec_no	rdate	cid

item_list:

rec_no	ordinal	item

Data file:

```
SQL> @C:/Krithika/DBL/a5data.sql;
SQL> REM Population of Bakery Database

SQL> drop table item_list;
Table dropped.

SQL> drop table receipts;
Table dropped.

SQL> drop table products;
Table dropped.

SQL> drop table customers;
Table dropped.

SQL>
SQL> create table customers(
  2         cust_no number(2) constraint c_pk primary key,
  3         lname varchar2(20),
  4         fname varchar2(20)
  5         );

Table created.

SQL>
SQL> insert into customers values(1, 'LOGAN', 'JULIET');
1 row created.
.
.
.
SQL> insert into customers values(21, 'JOHN', 'DAVID');
1 row created.

SQL> create table products(
  2         prod_id varchar2(20) constraint prod_pk primary key,
  3         flavour varchar2(20),
  4         food varchar2(20),
  5         price number
  6         );

Table created.

SQL>
SQL> insert into products values('20-BC-C-10','Chocolate','Cake',8.95);
1 row created.
.
.
```

```
.
SQL> insert into products values('51-BLU','Blueberry','Danish',1.15);
1 row created.

SQL> create table receipts(
  2          rec_no number(5) constraint rec_pk primary key,
  3          rdate date,
  4          cid number(2) constraint rec_fk references customers(cust_no)
  5          );

Table created.

SQL>
SQL> INSERT INTO Receipts values(18129, '28-Oct-2007', 15);
1 row created.
.
.
.
SQL> INSERT INTO Receipts values(34378, '23-Oct-2007', 6);
1 row created.

SQL> create table item_list(
  2          rec_no number(5) constraint it_fk1 references receipts(rec_no),
  3          ordinal number(2),
  4          item varchar2(20) constraint it_fk2 references products(prod_id),
  5          constraint item_pk primary key(rec_no,ordinal)
  6          );

Table created.

SQL>
SQL> insert into item_list values(18129, 1,  '70-TU');
1 row created.
.
.
.
SQL> insert into item_list values(34378, 2,  '45-VA');
1 row created.

SQL>
SQL> REM *** End of database population ***

SQL>
SQL> REM ******************** END OF DATA FILE ************************
```

Script file:

```
SQL> @z:/a5plsql.sql;
SQL> REM Assignment 5
SQL>
SQL> REM ----------------------------------------------------------------------
> REM *** ASSIGNMENT QUESTIONS ***
SQL> REM ----------------------------------------------------------------------
> REM Consider the schema used in Assignment 3.
SQL>
SQL>
SQL> REM **_____Write a 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 both are not available, display the relevant message.
SQL>
SQL> create or replace function prod_det(food1 products.food%type, flav1 products.flavour%type) return int is
  2  case1 varchar2(15);
  3  case2 varchar2(15);
  4  case3 varchar2(15);
  5  cursor c1 is select prod_id from products where food=food1 and flavour=flav1;
  6  cursor c2 is select prod_id from products where food=food1;
  7  cursor c3 is select prod_id from products where flavour=flav1;
  8  begin
  9          open c1;
 10          loop
 11                  fetch c1 into case1;
 12                  if c1%found then
 13                          return 1;
 14                  else
 15                          open c2;
 16                          loop
 17                                  fetch c2 into case2;
 18                                  if c2%found then
 19                                          return 2;
 20                                  else
 21                                          return 5;
 22                                  end if;
 23                          end loop;
 24                          close c2;
 25                          open c3;
 26                          loop
 27                                  fetch c3 into case3;
 28                                  if c3%found then
 29                                          return 3;
 30                                  else
 31                                          return 4;
 32                                  end if;
 33                          end loop;
 34                         close c3;
 35                 end if;
 36         end loop;
 37         close c1;
 38  EXCEPTION
 39  when no_data_found then
 40         return 0;
 41  end;
 42  /

Function created.

SQL>
SQL> declare
  2  foods products.food%type;
  3  flavours products.flavour%type;
  4  prod products.prod_id%type;
  5  begin
  6  foods:='&foods';
  7  flavours:='&flavours';
  8  prod:=prod_det(foods,flavours);
  9
 10  if prod = 1 then
 11         dbms_output.put_line('The combination of food '||foods||' and flavour '||flavours||' is available.');
 12  elsif prod = 2 then
 13         dbms_output.put_line('The food '||foods||' is available, but without the flavour '||flavours||'.');
 14  elsif prod = 3 then
 15         dbms_output.put_line('The flavour '||flavours||' is available, but without the food '||foods||'.');
 16  elsif prod = 4 then
 17         dbms_output.put_line('Neither the food '||foods||' nor the flavour '||flavours||' is available.');
 18  elsif prod = 5 then
 19         dbms_output.put_line('Neither the food '||foods||' nor the flavour '||flavours||' is available.');
 20  end if;
 21
 22  end;
 23  /
Enter value for foods: Cake
old   6: foods:='&foods';
new   6: foods:='Cake';
Enter value for flavours: Chocolate
old   7: flavours:='&flavours';
new   7: flavours:='Chocolate';
The combination of food Cake and flavour Chocolate is available.

PL/SQL procedure successfully completed.
```

```
SQL> /
Enter value for foods: Cake
old   6: foods:='&foods';
new   6: foods:='Cake';
Enter value for flavours: Cheese
old   7: flavours:='&flavours';
new   7: flavours:='Cheese';
The food Cake is available, but without the flavour Cheese.

PL/SQL procedure successfully completed.

SQL> /
Enter value for foods: Random
old   6: foods:='&foods';
new   6: foods:='Random';
Enter value for flavours: Variable
old   7: flavours:='&flavours';
new   7: flavours:='Variable';
Neither the food Random nor the flavour Variable is available.

PL/SQL procedure successfully completed.

SQL>
SQL> REM Validations:
SQL>
SQL> select * from products where food='Cake' and flavour='Chocolate';

PROD_ID              FLAVOUR              FOOD                      PRICE
-------------------- -------------------- -------------------- ----------
20-BC-C-10           Chocolate            Cake                       8.95

SQL> select * from products where food='Cake';

PROD_ID              FLAVOUR              FOOD                      PRICE
-------------------- -------------------- -------------------- ----------
20-BC-C-10           Chocolate            Cake                       8.95
20-BC-L-10           Lemon                Cake                       8.95
20-CA-7.5            Casino               Cake                      15.95
24-8x10              Opera                Cake                      15.95
25-STR-9             Strawberry           Cake                      11.95
26-8x10              Truffle              Cake                      15.95
46-11                Napoleon             Cake                      13.49

7 rows selected.

SQL> select * from products where food='Random';

no rows selected
```

```
SQL>
SQL>
SQL> REM 2. On a given date, find the number of items sold (Use Implicit cursor).
SQL>
SQL> create or replace procedure dateitems (dt in date) is
  2  numitems item_list.item%type;
  3  begin
  4          select count(item) into numitems
  5                  from receipts join item_list using(rec_no) where rdate=dt;
  6          if sql%notfound then
  7                  dbms_output.put_line('No items sold');
  8          elsif sql%found then
  9                  dbms_output.put_line('No. of items sold: '||numitems);
 10          end if;
 11  end;
 12  /

Procedure created.

SQL>
SQL> declare
  2  rdate date:=&rdate;
  3  begin
  4          dateitems(rdate);
  5  end;
  6  /
Enter value for rdate: '20-OCT-07'
old   2: rdate date:=&rdate;
new   2: rdate date:='20-OCT-07';
No. of items sold: 25

PL/SQL procedure successfully completed.

SQL> /
Enter value for rdate: 20-MAR-07'
old   2: rdate date:=&rdate;
new   2: rdate date:=20-MAR-07';
ERROR:
ORA-01756: quoted string not properly terminated


SQL>
SQL> REM Validations:
SQL>
SQL> select count(*) from receipts join item_list using(rec_no) where rdate='20-OCT-07';

  COUNT(*)
----------
        25
```

```
SQL> select count(*) from receipts join item_list using(rec_no) where rdate='20-MAR-07';

  COUNT(*)
----------
         0

SQL>
SQL>
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.
SQL> 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>
SQL> create or replace procedure price_det (pr in products.price%type) is
  2  flav1 products.flavour%type;
  3  food1 products.food%type;
  4  price1 products.price%type;
  5  pid1 products.prod_id%type;
  6  cursor c1 is select * from products where price in
  7          (select max(price) from products where price<=pr);
  8  begin
  9          open c1;
 10          dbms_output.put_line('PROD_ID      FOOD          FLAVOUR      QTY');
 11          loop
 12                  fetch c1 into pid1,food1,flav1,price1;
 13                  exit when c1%notfound;
 14                  dbms_output.put_line(pid1||'            '||food1||'      '||flav1||'            '||price1);
 15          end loop;
 16          dbms_output.put_line(c1%rowcount||' product(s) were found equal to or closest to the
given price.');
 17          close c1;
 18  end;
 19  /

Procedure created.

SQL>
SQL> declare
  2  price number:=&price;
  3  begin
  4          price_det(price);
  5  end;
  6  /
Enter value for price: 15.95
old   2: price number:=&price;
new   2: price number:=15.95;
PROD_ID          FOOD         FLAVOUR      QTY
20-CA-7.5          Casino         Cake       15.95
24-8x10          Opera         Cake       15.95
26-8x10          Truffle         Cake       15.95
```

```
3 product(s) were found equal to or closest to the given price.

PL/SQL procedure successfully completed.

SQL> /
Enter value for price: 16
old   2: price number:=&price;
new   2: price number:=16;
PROD_ID         FOOD        FLAVOUR     QTY
20-CA-7.5         Casino        Cake       15.95
24-8x10         Opera        Cake       15.95
26-8x10         Truffle        Cake       15.95
3 product(s) were found equal to or closest to the given price.

PL/SQL procedure successfully completed.

SQL>
SQL> REM Validations:
SQL>
SQL> select * from products where price=15.95;

PROD_ID              FLAVOUR              FOOD                      PRICE
-------------------- -------------------- -------------------- ----------
20-CA-7.5            Casino               Cake                      15.95
24-8x10              Opera                Cake                      15.95
26-8x10              Truffle              Cake                      15.95

SQL>
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.
SQL>
SQL> create or replace procedure ord_det (rnum in number) is
  2  cursor c1 is select food,flavour,count(*) from item_list,products
  3          where rec_no=rnum and item=prod_id
  4          group by food,flavour;
  5  namef customers.fname%type;
  6  namel customers.lname%type;
  7  flav1 products.flavour%type;
  8  food1 products.food%type;
  9  qty1 number;
 10  total number:=0;
 11  begin
 12          select fname,lname into namef,namel from customers join receipts on (cust_no=cid)
where rec_no=rnum;
 13          dbms_output.put_line('Customer name: '||namef||'  '||namel);
 14          dbms_output.put_line('Items: ');
 15          open c1;
 16          dbms_output.put_line('FOOD       FLAVOUR      QTY');
 17          loop
 18                     fetch c1 into food1,flav1,qty1;
 19                     if c1%found then
 20                             total:= total+qty1;
 21                             dbms_output.put_line(food1||'       '||flav1||'           '||qty1);
 22                     else
 23                             exit;
 24                     end if;
 25             end loop;
 26             dbms_output.put_line('------------------------------------');
 27             dbms_output.put_line('Total quantity: '||total);
 28  EXCEPTION
 29  when no_data_found then
 30             dbms_output.put_line('Order number does not exist!');
 31  end;
 32  /

Procedure created.

SQL>
SQL> declare
  2  rec_no number:=&rec_no;
  3  begin
  4             ord_det(rec_no);
  5  end;
  6  /
Enter value for rec_no: 64574
old   2: rec_no number:=&rec_no;
new   2: rec_no number:=64574;
Customer name: JOSETTE  SLINGLAND
Items:
FOOD        FLAVOUR        QTY
Twist       Almond        1
Cookie       Tuile        1
Cookie       Walnut        1
------------------------------------
Total quantity: 3

PL/SQL procedure successfully completed.

SQL> /
Enter value for rec_no: 12345
old   2: rec_no number:=&rec_no;
new   2: rec_no number:=12345;
Order number does not exist!

PL/SQL procedure successfully completed.

SQL>
SQL> REM Validations:
SQL>
SQL> select rec_no from customers join receipts on (cust_no=cid) where rec_no=64574;

   REC_NO
----------
    64574

SQL> select rec_no from receipts where rec_no=12345;

no rows selected

SQL>
SQL>
SQL> REM ************************* END OF FILE *********************************
```