Tase-6:- Procedures, functions and loops. 12/9/25

Case study:- Online food ordering

objective:-

To objective of this mash is no design, implement and execute pt/SQL procedures functions and loops no handle need-word (Queries) Stenarios near ecle from online food ordering system. This will keep in automoding transactions improving database efficiency and enforcing business soles in a structured monner.

Step-1:- Ensure the neccessary tables, exist Before, runing the procedure and functions, create the required table in your oracle Data base.

```
Drop Table  Order table PURGE;
DROP TABLE  Delivery PURGE;
DROP TABLE  menu_Item  PURGE;
CREATE, Table order Table (order_ID number primary-key,
Cust_iD. Number, order_Data_Date, order.total number (10,2)
Payment_status varchar(20)
CREATE Table. Delivery (order_ID Number. primary key,
Delivery_status varchar2(20), foreign. key (order_ID)
References order table (order_ID));
CREATE Table menu_item (Item_ID number primary key,
Item_name. varchar2(100). Price number (10,2));
Insert into order table values. (1,101, To_date ('2024_02_01'.
'YYYY_MM_DD'), 250, 50, 'pending');
Insert into. order table values. (2, 102, To_Date. ('2024_02_02'
'YYYY_MM_DD'). 400 - 75. 'paid');
Insert. In to order table values (3, To Date. ('2024_02_03',
'YYYY_MM_DD'), 15000, 'pending');
```

```
Order ID: 1, Date: 01-FEB-24, Total:
250.5, Status: Paid
Statement processed.
```

```
Discount Applied: 10%
Statement processed.
```

```
Payment status updated successfully
for order ID: 1
Statement processed.
```

Insert into Delivery values (1, 'pending');
Insert into Delivery values (2' 'Delivered');
Insert into Delivery values (3' pending');
Insert into menu_Item values (1, 'pizza', 500);
Insert into menu_Item values (2, 'Burger', 300);
Insert into menu_Item values (3, 'pasta', 450);

1) Procedure to update Payment status:-

Step:-1 Create a procedure:-

```
Create OR Replace procedure update_payment_status.(P_order
ID. In number, P_new_status. IN varchar 2).
As Begin update order table set payment_status = P_
new_status. where order_ID = P_order_ID;
Commit;
DBMS_output_line.('payment status updated successfully
for order ID: '||P_order_ID);
END;
```

Expected Output:-

Procedure Created.

Step 2: Execution

```
Begin
    update_payment_status.(1, paid);
End;
```

Expected Output:-

Payment_status updated successfully for order ID: 1
Statement processed.

Query 2:- function to calculate total Revenue.

Step-1:- Create a function:-

```
Create OR Replace function Get_total_Revenue
Return number as v_total_Revenue number;
  Begin
Select sum (order_table) Into v_total_Renue. From order
  table;
```

```
GET_TOTAL_REVENUE()
--------------------
            801.25
```

```
Return v_total_Revenue;
End;
/
```

Expected Output :-

Function Created.

Step-2:- Execution:-

GET_TOTAL_Revenue()

801.25

Query 3:- loop : mark ALL undelivered orders as "Delayed"

```
Declare r_order_ID order table order_ID %Type;
Cursor cur ID Select order ID from Delivery where
Delivery_status = "pending";
  Begin
      open cur;
        loop fetch cur into v_order_ID;
          exit when cur% Not found;
  Update delivery
    Set Delivery_status = Delayed"
  Where order_ID = v_order_ID;
DBMS_output_line ("order"||r_order||"marked as Delayed");
End loop;
Close cur;
Commit;
END;
/
```

Expected Output:-

/row (s) update

Query-4:- Procedure to get order Details by Customers ID

Step-1:- Create a procedure :-

```
Create of Replace procedure net_customer_oders
P_cust_ID In number
 IAs
Begin
```

order_ID	Cust_ID	order_Date	order_total	payment_status
1	101	2024-02-01	250.50	Paid
2	102	2024-02-02	400.75	Paid
3	103	2024-02-03	150.00	Pending

order_ID	Delivery_status
1	Delayed
2	Delivered
3	Delayed

for order_rec in (select order_ID, order Date, order_total, payment_status. from order table
where cust_ID = p_cust_ID) loop.

DBMS_Output_line ('Order_ID: '||order_rec.order_ID);
'|, Date: * ||order_rec.order_Date.||
'|, Total: ' || order_rec.order_Total||
'|, Status' : '|| order_rec.order_payment status.');
End loop;
End;
/

Expected Output
Procedure Created.

Step-2:- Execution:-
Begin.
Get_order_Details_By_customer (1);
End;
Expected output:-
Order ID : 1, Date : 2024-02-01, Total : 2505
Payment : paid
Statement processed.

Query 5:- Procedure to Apply Discount on menu_Items.
Step-1:- Create a procedure.

Create or Replace procedure Apply_Discount (
discount_payment In number.
)
is
Begin
Update menu_item
Set price = price -(price * discount_payment /100);
Commit;

Item_ID	Item_name	price
1	Pizza	450.00
2	Burger	450.00
3	Pasta	405.00

Result:- Thus the
loop's on number the
was successfully compe

VEL T
EX NO.
PERFORMANCE
RESULT AND ANA
VIVA VOCE (5)
RECORD (5)
TOTAL (20)
SIGN WITH DATE

```
DBMS_output.put_line('Discount Applied :'||discount_percent||'%');
End;
/
```

Expected Output:-

Procedure Created.

Step-2 :- Execution:-

```
Begin
  Apply_Discount(10);
End;
/
```

Expected output:-

Discount Applied : 10 %

Statement processed.

Result:- Thus the PL/SQL procedures, functions and loops on number theory bussiness scenarious experiment was successfully completed and results are verified.

VEL TECH - CSE	
EX NO.	6
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	20
SIGN WITH DATE	

12/9