Task-6 :- Procedures, functions and loops

Case study:- online food ordering system

objective:- To objective of tws task is to design, implement and execute PL/SQL procedures, functions and loops to handle real-world bussiness scenurious related to am online food ordering system This will help in autowating transctions, ordering system. This will help in autowating transctions, improving database efficiency and enforing business voles in a structured manner.

step1:- Ensure the necesscary tables Exist:-

Before running the procedure and functions, create the required tubles in your oracle Database.

```
DROP TABLE ordertable PURGE;
DROP TABLE Delivery PURGE;
DROP TABLE Menu_Item PURGE;
CREATE TABLE ordertable (order_ID Number primary key, cost_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), foregin 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-data ('2024-20-01', 'YYYY-MM-DD'), 250.50, 'Pending'),
Insert into ordertable values (2, 102, 50-Data ('2024-03-03', 'YYYY-MM-DD') '400.75, 'Paid');
Insert Into order table values (3, To-Date ('2024-02-03', 'YYYY-MM-DD'), 15000, Pending);
Insert into Delivery values (1, 'Pending');
Insert into Delivery values (2, 'Delivered');
Insert into Delivery values (3, 'Pending');
Insert into menn_Item values (1, 'pizza', 500);
Insert Pnbo menn_Item values (2, 'Burger', 300);
Insert into menn_Item values (3, 'pasta', 450);
```

1. Procedure to update payment status

step1:- create a procedure

```
create OR Replace procedure update_payment_status (p_order_ID IN Number, p_new_status IN varchar As Begin update order table set payment_status = p_New_status where order_ID = p_order_ID;
Commit;
DBMS_output put_line ('payment status updated for order ID: ||P_order_ID); successfully)
End;
/
```

Expected output:-

procedure created

step2:- Execution:-

```
Begin
    update_payment_status (1, 'Paid');
    End;
```

```
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.
```

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

Expected output:-
Payment status updated successfully for order ID: 1
Statement processed.

Query2:- function to calculate total Revenue.

step1:- create a function.

```
CREATE OR Replace function Get_total_Renue Return number as
v_total_Revenue number;
Begin
 select sum (order_total) Into v_total_Revenue from
 order table;
Return v_total_Revenue;
END;
```

Expected output:-
Function created

step2:- Execution
GET_TOTAL_REVENUE()
Sol.25

Query3:- loop: mark All underlivered orders as "Delayed".

```
Declare v_order_ID order Table. order ID % Type;
CURSOR CUR Is select order ID from Delivery where
Delivery_status = "pending";
Begin
   open cur;
     loop. fetch cur into r_order_ID;
     Exit when cur%. NOT FOUND;
update Delivery
     set Delivery_status = "Delayed"
     where order_ID = v_order_ID;
     Deny_output.put. line ("order " || v_order || "marked as Delayed");
End loop;
Close curr;
commit;
END;
```

Query5:- Procedure to apply Discount on menes Itens.

step1:- check a procedure

```
create or Replace procedure Apply_Discount (
     discount_percent. In number
     )
is
Begin
update Menn_Item
set price = price_ (price * discount_precent /100);
Commit;
DBMS_output.put_line ('Discount Applied : ' || discount.
Percent + || '%');
End;
```

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

order_ID	Delivery_status
1	Delayed
2	Delivered
3	Delayed

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

Excepected output:-
procedure created
step2:- Execution
Begin
Apply_Discount (10);
End;
/
Expected output:-
Discount Applied: 10 %.
statement processed.

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	25/3/25

Result:- Thus, the PL/SQL procedures, functions and loops on number theory bussiness scenorles experiments was successfully complemented and results are verified.