DATE: 15/9/25

TASK: 6

TRIGGERS VIEWS AND EXCEPTIONS

AIM: To implement PL/SQL procedures, Functions and loops on Number Theory and business Scenarios..

procedure:

- PL/SQL is a combinatination customer table
- Insert the values
- perform trigger operation
- perform view operation
- perform Exception Handling

1. Implementing Triggers.

```
CREATE TABLE Custome(
    customer ID  INT  PRIMARY KEY,
     Name  VARCHAR(50)
     Age  INT
     phone  VARCHAR(15)
     Email.  VARCHAR(50)
     );
```

```
CREATE OR REPLACE TRIGGER prevent,
          underage. customers
 BEFORE  INSERT  ON  customer
 FOR EACH  ROW
  BEGIN
     IF : NEW-Ag <<18THEN
       RAISE- APPLICATION. ERROR (-2000', customer
       age  must  be  18 orabove')
     END IF;
   END;
```

2. Create a log table for customer Actions

```
   CREATE TABLE  customerlog (
     LogID  NUMBER  PRIMARY KEY,
     customerID INT,
      Action type   VARCHAR(20)
       Action Date  TIMESTAMP DEFAULT
       SYSTIMESTAMP
     );
```

2. Creating veiws.

view for customer contact information

```
   CREATE OR REPLACE VIEW view, customer
   contacts AS
   SELECT   customerID, Name, phone, Email
```

```
FROM customer;

CREATE OR REPLACE VIEW _customer Age
  categahy AS
 SELECT
      customer ID,
      Name,
      CASE
       WHEN Age BETWEEN 18 and 25 THEN 'YOUNG'
       WHEN Age BETWEEN 26 and 45 THEN 'adult'
       WHEN Age BETWEEN 46 and 65 THEN 'middle
                                        Age'
        ELSE Senior.
       END AS Age category
   FROM customer;
```

3. Exception Handling

Sorted procedure with Exception Handling for Inserting customer Records.

```
 CREATE OR REPLACE PROCEDURE Insertcusto
                                    mer(
      P_customer ID IN INIT,
      P-Name IN VARCHAR,
      P-Age IN INT,
      P-phone IN VARCHAR.
      P-EMAIL IN VARCHR
   )
   IS
   BEGIN
```

```
INSERT INTO customer (customer ID, Name,
                      Age, phone, Email)
VALUES (P_customer ID, P_name, P_Age, P_Phone,
        P_Email);
   DBMS_OUTPUT.PUT_LINE ('customer Record
        Inserted succesfully');
EXCEPTION
   WHEN OTHERS THEN
        DBMS-OUTPUT. PUT_LINE ('ERROR:'||
   END;                              SALERRM);
```

Function to Fetch customer Details with Error Handling

```
CREATE OR REPLACE FUNCTION Get
 customer Details (P_Customer ID IN NumB)
 RETURN VARCHAR
 IS
   customer_info VARCHAR(200);
 BEGIN
   SELECT 'Name"|| Name": Age: '||Agel', Phone:'
 || phone || :Email:'|| Email
 INTO custmer - Info
 FROM customer
```

```
WHERE customer ID = p_customer.ID;
RETURN customer-into;

EXCEPTION
    WHEN NO-DATA-FOUND THEN
    RETURN 'customer not Found';
    WHEN OTHERS THEN
        RETURN 'Error:'

END;
```

| TECH | |
|---|---|
| EX No | 6 |
| PERFORMANCE | 5 |
| RESULT AND | 5 |
| VIVA VOCE (5) | 5 |
| RECORD | |
| TOTAL | 15 |
| SIGN | 15/9/23 |

Result:

Thus, The implementation of triggers, view and exceptions in SQL are executed successfully