```
COMMIT;
DBMS_output.PUT_LINE('Discount Applied :'||discount_percent
||'%');
END;
```

TASK 7:-

Triggers, Views, and Exceptions

Objective:-

To understand and implement Triggers, views and Exceptions Handiling for performing and managing CRUD operations in an oracle database.

Part 1:- Implementing Triggers.

1. Prevent Insertion of underage Students.

```
 CREATE TABLE students (Student ID NUMBER PRIMARY KEY,
Name VARCHAR(50), Age Number, Department VARCHAR (20),
 MARKS Number);
CREATE OR REPLACE TRIGGER Prevent_underage_students
 BEFORE INSERT ON Students FOR EACH ROW
 BEGIN
     IF :NEW.Age<18 THEN
          RAISE_Application_ERROR (-20001, 'Age must be 18
 or above');
   END IF;
  END;
```

2. CREATE a Log Table:-

```
 CREATE TABLE StudentLog (
  LogID NUMBER PRIMARY KEY, StudentID NUMBER,
 ActionType VARCHAR2(20), ActionDate TIMESTAMP
 DEFAULT SYSYTIMESTAMP);
```

PART 2:- Creating views

1. View for Top Students

```
  CREATE OR REPLACE VIEW View_TopStudents 45
  SELECT student_ID, Name, Marks FROM Students
```

```
WHERE MARKS > 80;
```

2. View for Department Summary:

```
CREATE OR REPLACE VIEW view_DepartmentSummary AS
SELECT Department, COUNT(*) AS TotalStudents, ROUND(AVG
(Marks), 2) AS AverageMarks FROM Students GROUP BY
Department;
```

PART-3 :- Exception Handling

1. Stored Procedure with Exceptions Handling for Inserting Student Records.

```
CREATE OR REPLACE PROCEDORE InsertStudent (P_StudentID IN Number,
P_Name IN VARCHAR2,
P_Age IN Number,
P_Department IN VARCHAR2,
P_MARKS IN Number)
IS
  BEGIN
      INSERT INTO Students (StudentID, Name, Age, Department, MARKS)
      VALUES (P_StudentID , P_Name, P_Age, P_Department,
P_MARKS);
    DBMS_OutPut .PUT_Line('Record Inserted successfully');
Exception
    WHEN OTHERS THEN
                DBMS_OUTPot.PUT_LINE ('Error:' || SQLERRM);
  END;
```

2. FUNCTIONS to Fetch Student Details with Error Handiling:-

```
CREATE OR REPLACE FUNCTION GetStudentDetails (P_StudentID IN NUMBER) RETURN VARCHAR2
IS
  Student_info VARCHAR2(225);
BEGIN
  SELECT 'Name:' || Name || ', Age :' || ', Department :', || Department || ', Marks:' || Marks
INTO Student_ info
FROM Students
WHERE StudentID = P_Student_ID;
 RETURN Student_info;
Exception
   WHEN NO_DATE_FOOND THEN
   RETURN 'Student NOT Found';
  WHEN others THEN
    RETURN 'Error:' || SQLERRM;
 END;
```

Result:- Hence, Implementing Triggers, views and Exceptions has been done successfully.

22/7/25