

output

view-Top students

Student-ID

Name

Marks

101

Alice

91

105

Rahul

88

108

Sheha

85

Task-7

Task-7 i - Triggers, views and Exceptions

Objective:-

To understand and implement Triggers, views and Exception Handling for performing and managing CRUD (Create, Read, Update, Delete) operation in an Oracle database.

Part:- Implement Triggers

1. prevent Insertion of underage students

CREATE TABLE Students (

Students ID NUMBER PRIMARY KEY,

NAME VARCHAR (50)

Age NUMBER

Department VARCHAR (50),

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;

Department

Total-Students

Average-
marks

CSG

3

87.33

IT

2

85.00

ECG

90.00

CREATE a Log Table

CREATE TABLE Student Log(

logID NUMBER PRIMARY KEY,

Student ID Number,

ACTION TYPE VARCHAR(20),

ACTION DATE TIME STAMP DEFAULT SYSTEM
STAMP

Part 2: - Creating views

1. view for Top students

CREATE OR REPLACE VIEW view-Top Student AS

SELECT Student ID, Name, Marks

FROM Students

WHERE Marks > 80

2. View for Department Summary

CREATE OR REPLACE VIEW view-Department
Summary AS

SELECT

Department

Count(*) AS Total Students,

ROUND(AVG(Mark), 2) AS Average Marks

FROM STUDENTS

GROUP BY Department

output :-

Insert - log Table

LogID - Student_ID - Operation - Times of

1 110
2 105

'Insert'

'Insert'

13-Sep-2023 18:08:00
13-Sep-2025 18:06:00

Part-3:- Exception Handling

1. stored procedure with Exception Handling for Inserting student Records

```
CREATE OR REPLACE PROCEDURE Insert Student (  
  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_output.put_line ('Error: ' || SQLERRM);  
  END;
```

2. Function to Fetch student Details with Error Handling

```
CREATE OR REPLACE FUNCTION Get Student  
Details (P_StudentID IN NUMBER)
```

```
RETURN VARCHAR2
```

```
IS
```

```
  Student_into VARCHAR2(255);
```

```
  BEGIN
```

```
    SELECT 'NAME: ' || Name || 'Age: ' || Age || 'Department  
  || Department
```

Marks: '11 Marks

INTO student_info

FROM student

WHERE Student ID = p_student ID;

RETURN student_info;

Exception

WHEN NO_DATA_FOUND THEN

RETURN 'STUDENT NOT FOUND';

WHEN OTHERS THEN

RETURN 'Error: ' || SQLERRM;

END;

VEL TECH - CSE	
EX NO.	
PERFORMANCE (5)	
RESULT AND ANALYSIS (3)	
VIVA VOCE (3)	
RECORD (4)	
TOTAL (15)	
SIGN WITH DATE	

Result:- Thus the task is executed and verified successfully