

Experiment-8

Student Name: Gauri Prabhakar

UID: 18BCS6201

Branch: 18AITAIML-2

Section/Group: B

Semester: 7

Date of Performance: 10th November, 2021

Subject Name: Advanced Database Management Lab

Subject Code: CSP - 434

1. Aim/Overview of the practical:

To implement various operations on Packages and Triggers.

2. Task to be done:

To implement various operations on Packages and Triggers.

3. Steps to be followed:

PACKAGE:

Creating a table STUDENTS and then returning it:

1. CREATE TABLE STUDENTS(Roll_No int, Name varchar(50), Age int);
INSERT INTO STUDENTS VALUES(1,'Gauri',21);
INSERT INTO STUDENTS VALUES(2,'Lilly',22);
INSERT INTO STUDENTS VALUES(3,'Rose',18);
SELECT * FROM STUDENTS;

```
Select Run SQL Command Line
SQL> connect
Enter user-name: system
Enter password:
Connected.
SQL> CREATE TABLE STUDENTS(Roll_No int, Name varchar(50), Age int);

Table created.

SQL> INSERT INTO STUDENTS VALUES(1,'Gauri',21);

1 row created.

SQL> INSERT INTO STUDENTS VALUES(2,'Lilly',22);

1 row created.

SQL> INSERT INTO STUDENTS VALUES(3,'Rose',18);

1 row created.

SQL> SELECT * FROM STUDENTS;

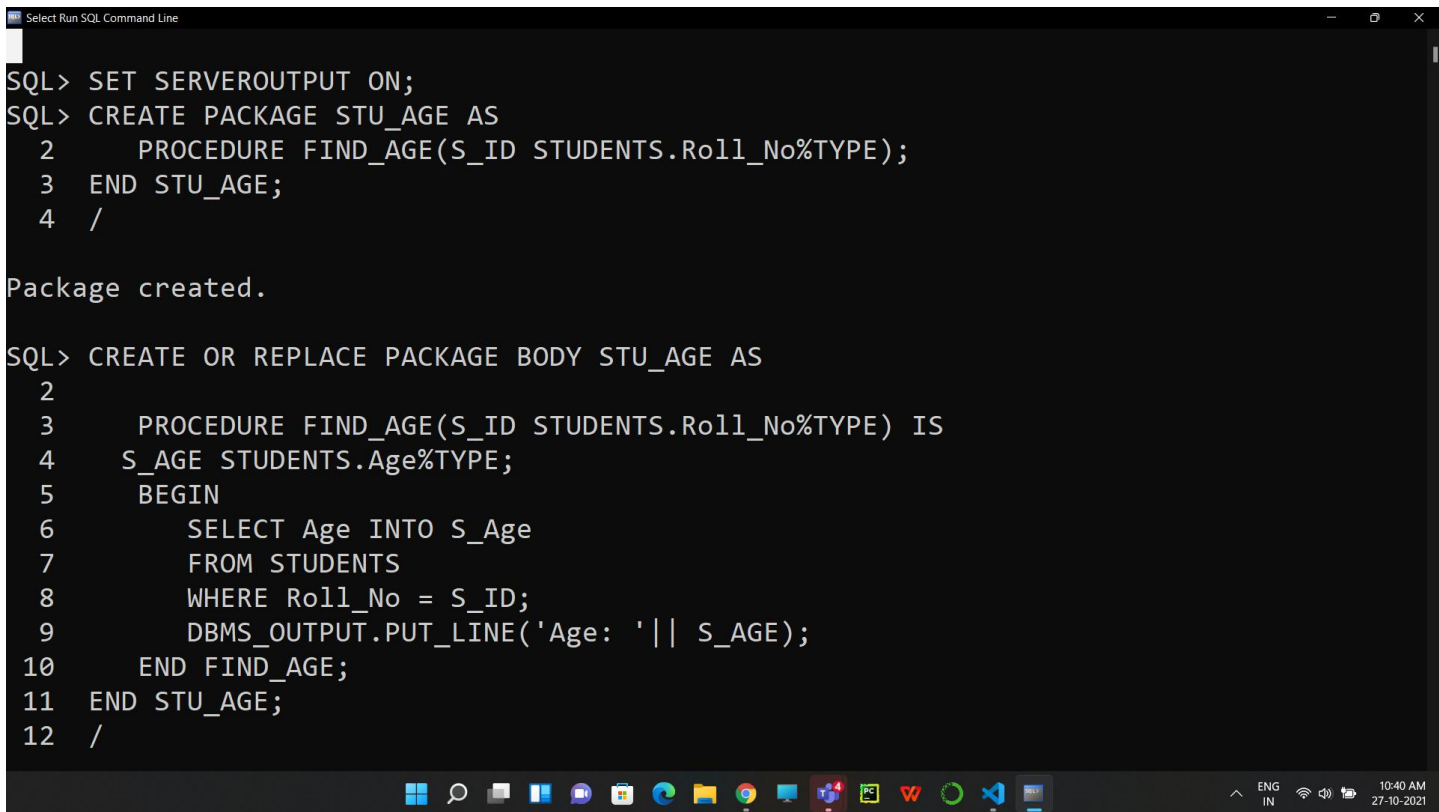
ROLL_NO NAME AGE
-----
1 Gauri 21
2 Lilly 22
3 Rose 18
```

Creating the package body and then defining it:

2. SET SERVEROUTPUT ON;

```
CREATE PACKAGE STU_AGE AS
    PROCEDURE FIND_AGE(S_ID STUDENTS.Roll_No%TYPE);
END STU_AGE;
/

CREATE OR REPLACE PACKAGE BODY STU_AGE AS
    PROCEDURE FIND_AGE(S_ID STUDENTS.Roll_No%TYPE) IS
        S_AGE STUDENTS.Age%TYPE;
    BEGIN
        SELECT Age INTO S_Age
        FROM STUDENTS
        WHERE Roll_No = S_ID;
        DBMS_OUTPUT.PUT_LINE('Age: ' || S_AGE);
    END FIND_AGE;
END STU_AGE;
/
```



The screenshot shows a SQL Command Line window with the following text:

```
Select Run SQL Command Line

SQL> SET SERVEROUTPUT ON;
SQL> CREATE PACKAGE STU_AGE AS
2     PROCEDURE FIND_AGE(S_ID STUDENTS.Roll_No%TYPE);
3 END STU_AGE;
4 /

Package created.

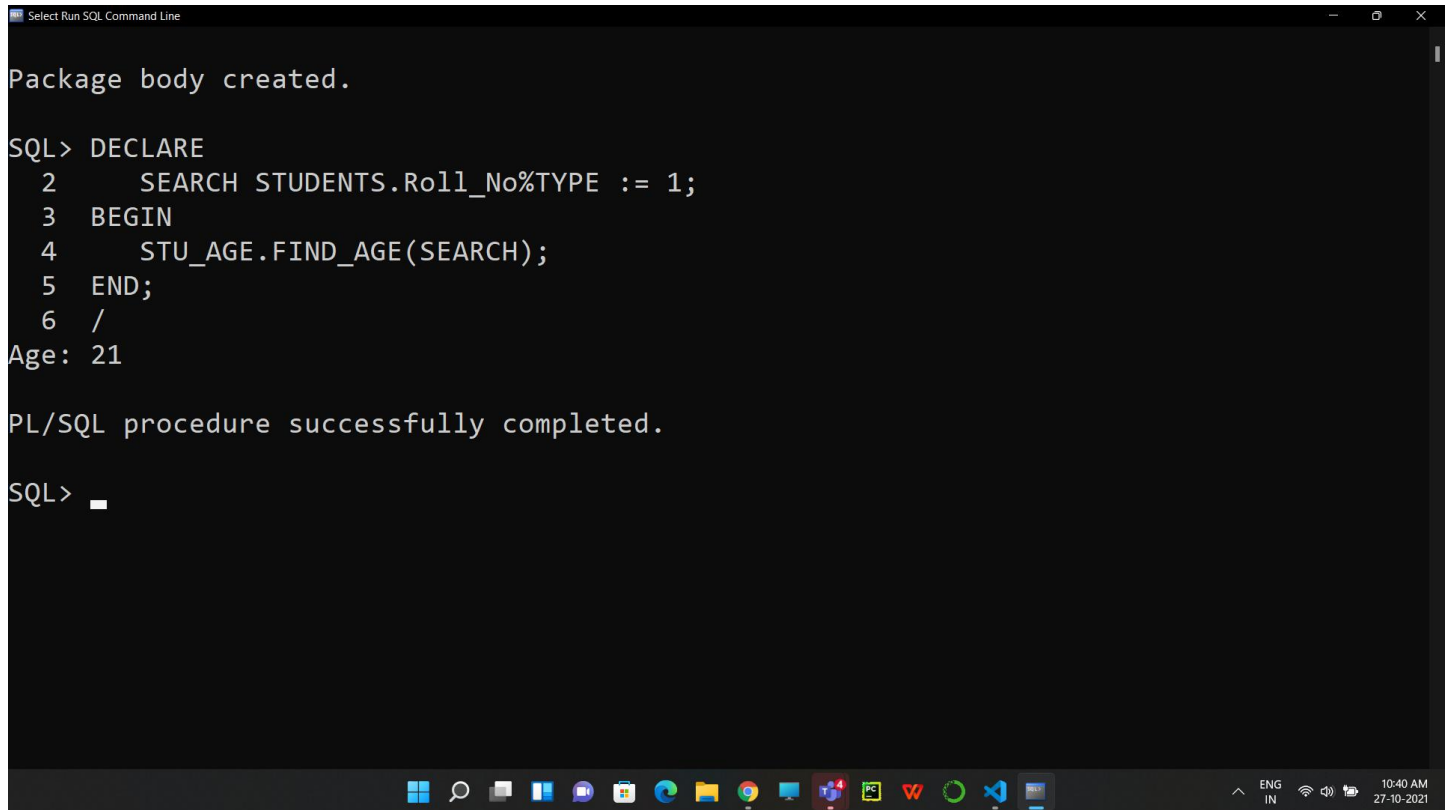
SQL> CREATE OR REPLACE PACKAGE BODY STU_AGE AS
2
3     PROCEDURE FIND_AGE(S_ID STUDENTS.Roll_No%TYPE) IS
4 S_AGE STUDENTS.Age%TYPE;
5 BEGIN
6     SELECT Age INTO S_Age
7     FROM STUDENTS
8     WHERE Roll_No = S_ID;
9     DBMS_OUTPUT.PUT_LINE('Age: ' || S_AGE);
10 END FIND_AGE;
11 END STU_AGE;
12 /
```

The window also shows a Windows taskbar at the bottom with various application icons and a system tray on the right indicating the language is English (IN), and the time is 10:40 AM on 27-10-2021.

Implementing the Package:

3. DECLARE

```
SEARCH STUDENTS.Roll_No%TYPE := 1;  
BEGIN  
STU_AGE.FIND_AGE(SEARCH);  
END;  
/
```



The screenshot shows a Windows command prompt window titled "Select Run SQL Command Line". The output of the SQL execution is as follows:

```
Package body created.  
  
SQL> DECLARE  
2     SEARCH STUDENTS.Roll_No%TYPE := 1;  
3 BEGIN  
4     STU_AGE.FIND_AGE(SEARCH);  
5 END;  
6 /  
Age: 21  
  
PL/SQL procedure successfully completed.  
  
SQL> █
```

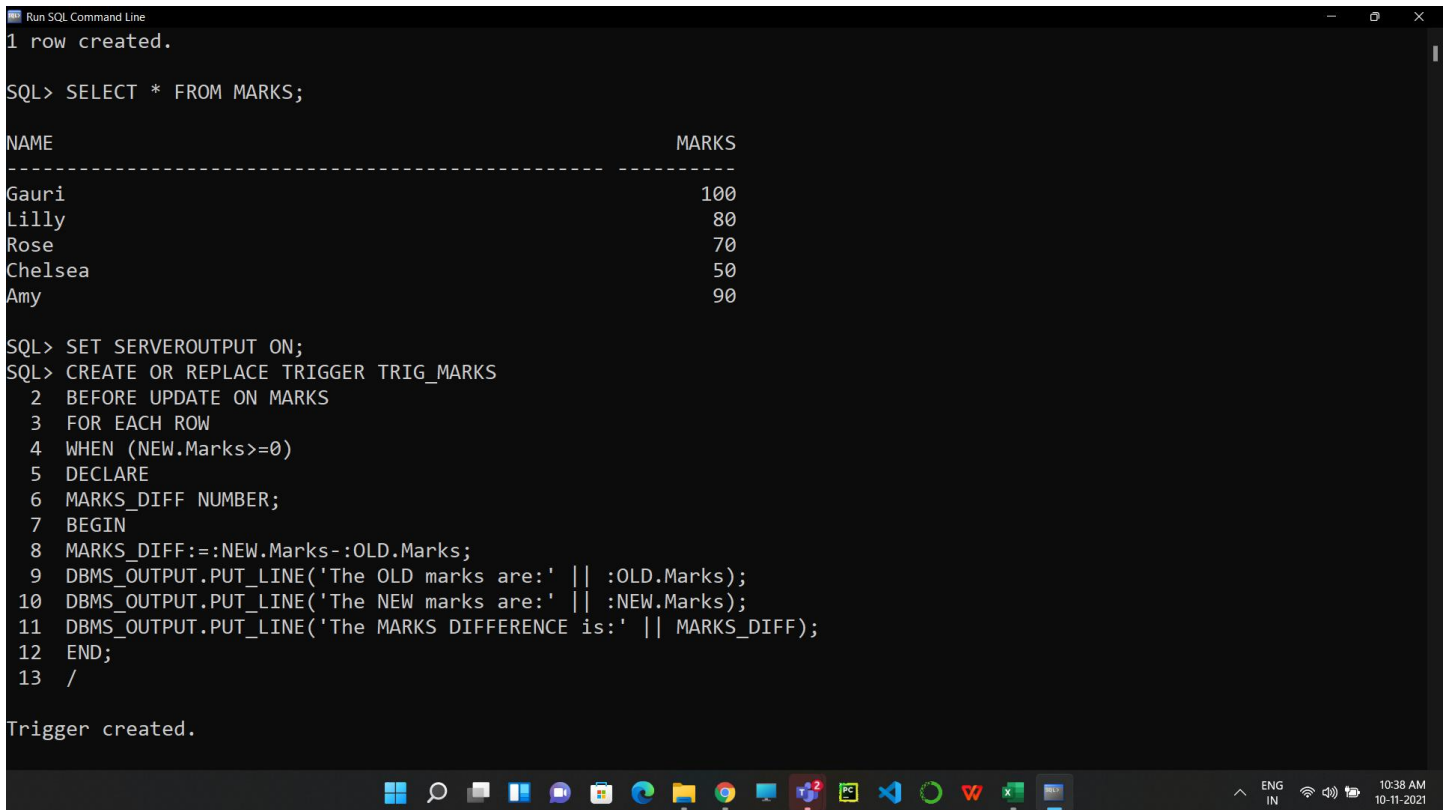
The Windows taskbar at the bottom shows various application icons and the system clock indicating 10:40 AM on 27-10-2021.

TRIGGERS:

Returning the table MARKS and declaring the Trigger:

4. SELECT * FROM MARKS;

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE TRIGGER TRIG_MARKS
BEFORE UPDATE ON MARKS
FOR EACH ROW
WHEN (NEW.Marks>=0)
DECLARE
MARKS_DIFF NUMBER;
BEGIN
MARKS_DIFF:=:NEW.Marks-:OLD.Marks;
DBMS_OUTPUT.PUT_LINE('The OLD marks are:' || :OLD.Marks);
DBMS_OUTPUT.PUT_LINE('The NEW marks are:' || :NEW.Marks);
DBMS_OUTPUT.PUT_LINE('The MARKS DIFFERENCE is:' || MARKS_DIFF);
END;
/
```



The screenshot shows a 'Run SQL Command Line' window with the following content:

```
1 row created.

SQL> SELECT * FROM MARKS;

NAME                                MARKS
-----
Gauri                                100
Lilly                                80
Rose                                 70
Chelsea                              50
Amy                                  90

SQL> SET SERVEROUTPUT ON;
SQL> CREATE OR REPLACE TRIGGER TRIG_MARKS
2 BEFORE UPDATE ON MARKS
3 FOR EACH ROW
4 WHEN (NEW.Marks>=0)
5 DECLARE
6 MARKS_DIFF NUMBER;
7 BEGIN
8 MARKS_DIFF:=:NEW.Marks-:OLD.Marks;
9 DBMS_OUTPUT.PUT_LINE('The OLD marks are:' || :OLD.Marks);
10 DBMS_OUTPUT.PUT_LINE('The NEW marks are:' || :NEW.Marks);
11 DBMS_OUTPUT.PUT_LINE('The MARKS DIFFERENCE is:' || MARKS_DIFF);
12 END;
13 /

Trigger created.
```

The Windows taskbar at the bottom shows the time as 10:38 AM on 10-11-2021, with system language set to ENG IN.

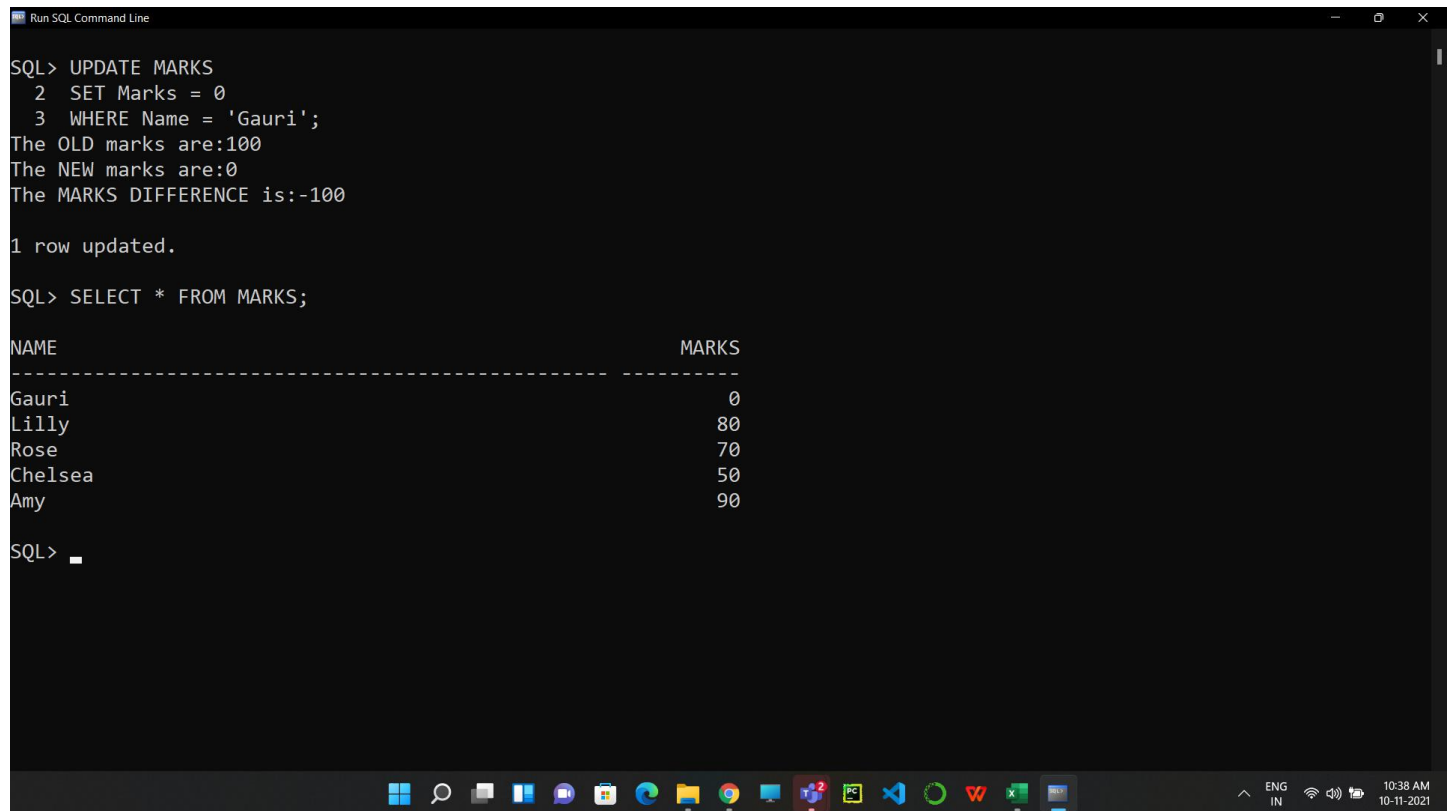
Implementing the Trigger:

5. UPDATE MARKS

SET Marks = 0

WHERE Name = 'Gauri';

SELECT * FROM MARKS;



The screenshot shows a 'Run SQL Command Line' window with a black background and white text. The SQL commands and their outputs are as follows:

```
SQL> UPDATE MARKS
  2  SET Marks = 0
  3  WHERE Name = 'Gauri';
The OLD marks are:100
The NEW marks are:0
The MARKS DIFFERENCE is:-100

1 row updated.

SQL> SELECT * FROM MARKS;
```

NAME	MARKS
Gauri	0
Lilly	80
Rose	70
Chelsea	50
Amy	90

The window also shows a Windows taskbar at the bottom with various application icons and a system tray on the right displaying 'ENG IN', signal icons, and the date/time '10:38 AM 10-11-2021'.

4. Result/Output/Writing Summary:

- Successfully implemented PACKAGES.
- Successfully implemented TRIGGERS.
- Successfully implemented operations on both PACKAGES and TRIGGERS.
- Successfully understood the functioning and importance of the above mentioned.

5. Learning outcomes (What I have learnt):

- How to implement PACKAGES on SQL Command Line.
- How to implement TRIGGERS on SQL Command Line.
- How to implement TRIGGERS on a table.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			