Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

EXERCISE 11: TRIGGERS & CURSORS

Faculty Name: Dr. Balasundaram A
Slot: L43+L44
Class Number: CH2021221000708
Date: 20/10/2021

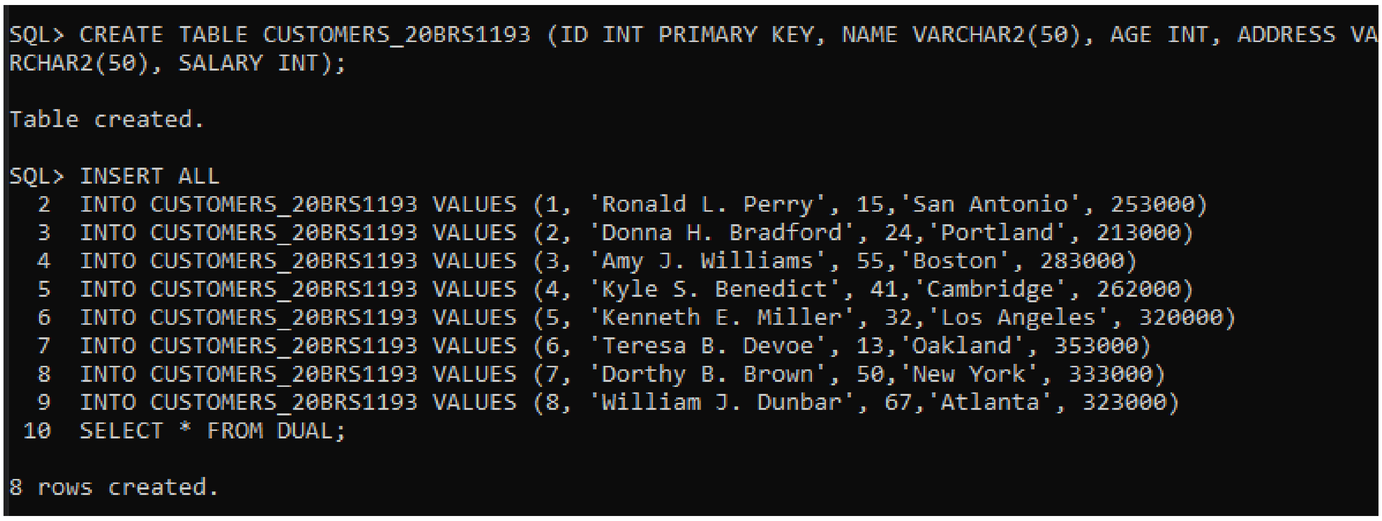
DBMS LAB ASSIGNMENT 10 MEHER SHRISHTI NIGAM 20BRS1193

PLSQL PROCEDURES AND FUNCTIONS

1. Create a trigger named display_salary_changes. The trigger should fire whenever there is a delete or insert or update on customers table. The difference in salary should be computed and displayed. Assume that the table customers contains the fields id, name, age, address, salary.

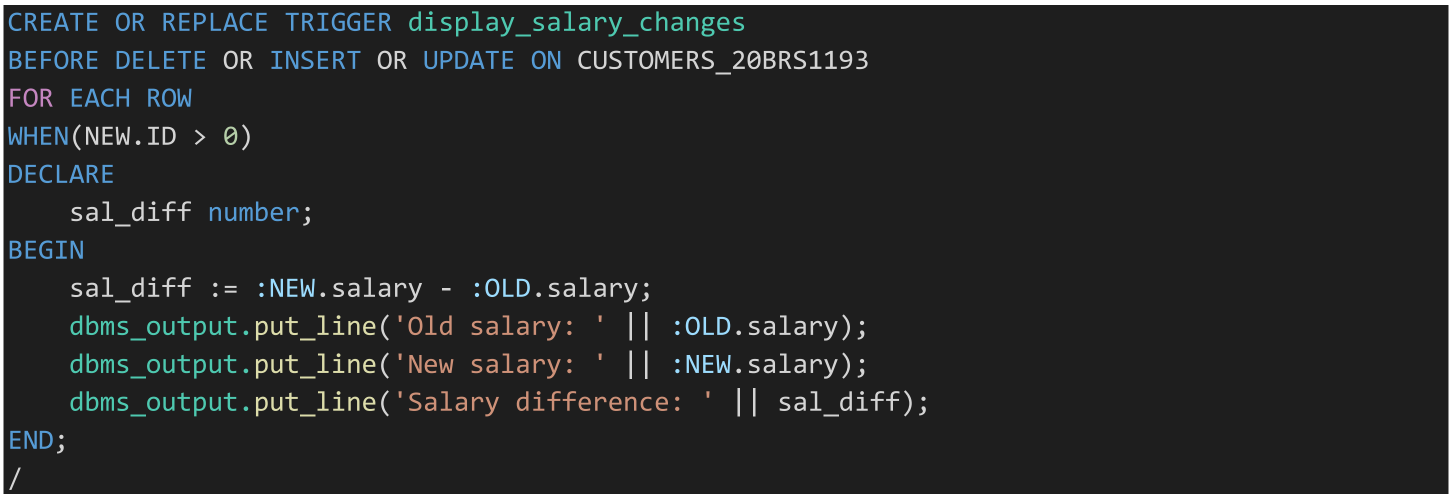
CUSTOMERS_20BRS1193 TABLE

```sql
CREATE TABLE CUSTOMERS_20BRS1193 (ID INT PRIMARY KEY, NAME VARCHAR2(50), AGE INT, ADDRESS
VARCHAR2(50), SALARY INT);
INSERT ALL
    INTO CUSTOMERS_20BRS1193 VALUES (1, 'Ronald L. Perry', 15,'San Antonio', 253000)
    INTO CUSTOMERS_20BRS1193 VALUES (2, 'Donna H. Bradford', 24,'Portland', 213000)
    INTO CUSTOMERS_20BRS1193 VALUES (3, 'Amy J. Williams', 55,'Boston', 283000)
    INTO CUSTOMERS_20BRS1193 VALUES (4, 'Kyle S. Benedict', 41,'Cambridge', 262000)
    INTO CUSTOMERS_20BRS1193 VALUES (5, 'Kenneth E. Miller', 32,'Los Angeles', 320000)
    INTO CUSTOMERS_20BRS1193 VALUES (6, 'Teresa B. Devoe', 13,'Oakland', 353000)
    INTO CUSTOMERS_20BRS1193 VALUES (7, 'Dorthy B. Brown', 50,'New York', 333000)
    INTO CUSTOMERS_20BRS1193 VALUES (8, 'William J. Dunbar', 67,'Atlanta', 323000)
SELECT * FROM DUAL;
```



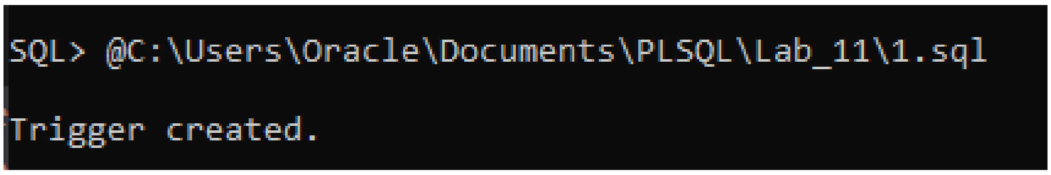
```
SQL> CREATE TABLE CUSTOMERS_20BRS1193 (ID INT PRIMARY KEY, NAME VARCHAR2(50), AGE INT, ADDRESS VA
RCHAR2(50), SALARY INT);

Table created.

SQL> INSERT ALL
  2  INTO CUSTOMERS_20BRS1193 VALUES (1, 'Ronald L. Perry', 15,'San Antonio', 253000)
  3  INTO CUSTOMERS_20BRS1193 VALUES (2, 'Donna H. Bradford', 24,'Portland', 213000)
  4  INTO CUSTOMERS_20BRS1193 VALUES (3, 'Amy J. Williams', 55,'Boston', 283000)
  5  INTO CUSTOMERS_20BRS1193 VALUES (4, 'Kyle S. Benedict', 41,'Cambridge', 262000)
  6  INTO CUSTOMERS_20BRS1193 VALUES (5, 'Kenneth E. Miller', 32,'Los Angeles', 320000)
  7  INTO CUSTOMERS_20BRS1193 VALUES (6, 'Teresa B. Devoe', 13,'Oakland', 353000)
  8  INTO CUSTOMERS_20BRS1193 VALUES (7, 'Dorthy B. Brown', 50,'New York', 333000)
  9  INTO CUSTOMERS_20BRS1193 VALUES (8, 'William J. Dunbar', 67,'Atlanta', 323000)
 10  SELECT * FROM DUAL;

8 rows created.
```

TRIGGER CODE



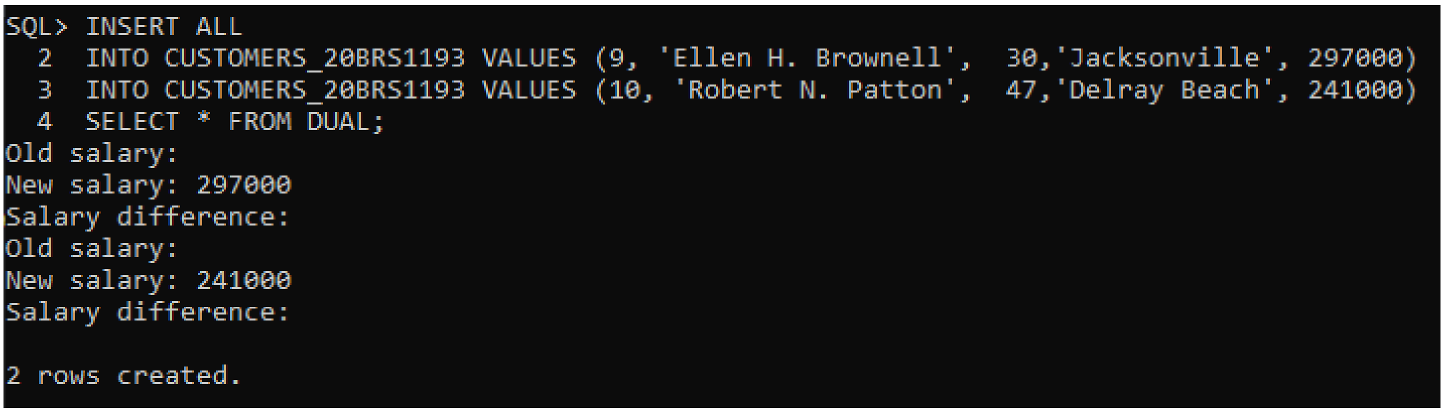
```
CREATE OR REPLACE TRIGGER display_salary_changes
BEFORE DELETE OR INSERT OR UPDATE ON CUSTOMERS_20BRS1193
FOR EACH ROW
WHEN(NEW.ID > 0)
DECLARE
    sal_diff number;
BEGIN
    sal_diff := :NEW.salary - :OLD.salary;
    dbms_output.put_line('Old salary: ' || :OLD.salary);
    dbms_output.put_line('New salary: ' || :NEW.salary);
    dbms_output.put_line('Salary difference: ' || sal_diff);
END;
/
```

TRIGGER CREATED

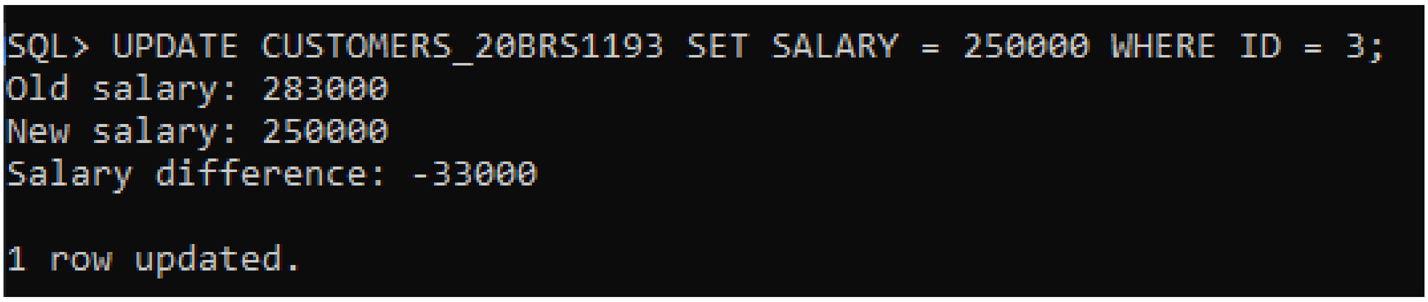


```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_11\1.sql

Trigger created.
```

USING TRIGGER



```
SQL> INSERT ALL
  2  INTO CUSTOMERS_20BRS1193 VALUES (9, 'Ellen H. Brownell',  30,'Jacksonville', 297000)
  3  INTO CUSTOMERS_20BRS1193 VALUES (10, 'Robert N. Patton',  47,'Delray Beach', 241000)
  4  SELECT * FROM DUAL;
Old salary:
New salary: 297000
Salary difference:
Old salary:
New salary: 241000
Salary difference:

2 rows created.
```

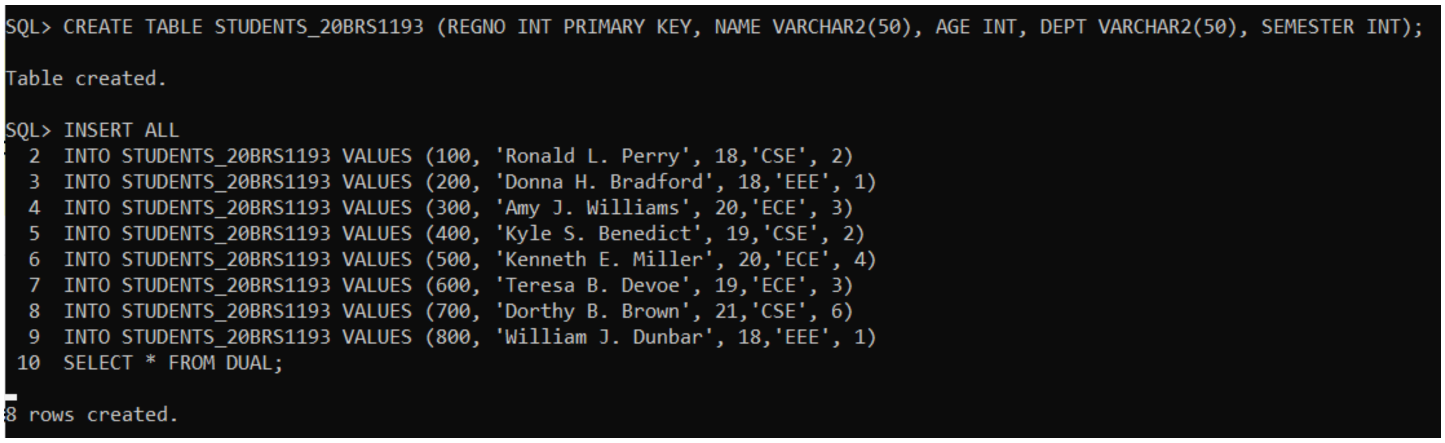


```
SQL> UPDATE CUSTOMERS_20BRS1193 SET SALARY = 250000 WHERE ID = 3;
Old salary: 283000
New salary: 250000
Salary difference: -33000

1 row updated.
```

2. Create a trigger named display_semester_changes. The trigger should fire whenever a student semester value is changed in student table. Assume that the student table contains the fields regno, name, age, dept, semester. Display the old and new value in the command line.

STUDENTS_20BRS1193 TABLE

```
CREATE TABLE STUDENTS_20BRS1193 (REGNO INT PRIMARY KEY, NAME VARCHAR2(50), AGE INT, DEPT
VARCHAR2(50), SEMESTER INT);
INSERT ALL
    INTO STUDENTS_20BRS1193 VALUES (100, 'Ronald L. Perry', 18,'CSE', 2)
    INTO STUDENTS_20BRS1193 VALUES (200, 'Donna H. Bradford', 18,'EEE', 1)
    INTO STUDENTS_20BRS1193 VALUES (300, 'Amy J. Williams', 20,'ECE', 3)
    INTO STUDENTS_20BRS1193 VALUES (400, 'Kyle S. Benedict', 19,'CSE', 2)
    INTO STUDENTS_20BRS1193 VALUES (500, 'Kenneth E. Miller', 20,'ECE', 4)
    INTO STUDENTS_20BRS1193 VALUES (600, 'Teresa B. Devoe', 19,'ECE', 3)
    INTO STUDENTS_20BRS1193 VALUES (700, 'Dorthy B. Brown', 21,'CSE', 6)
    INTO STUDENTS_20BRS1193 VALUES (800, 'William J. Dunbar', 18,'EEE', 1)
SELECT * FROM DUAL;
```

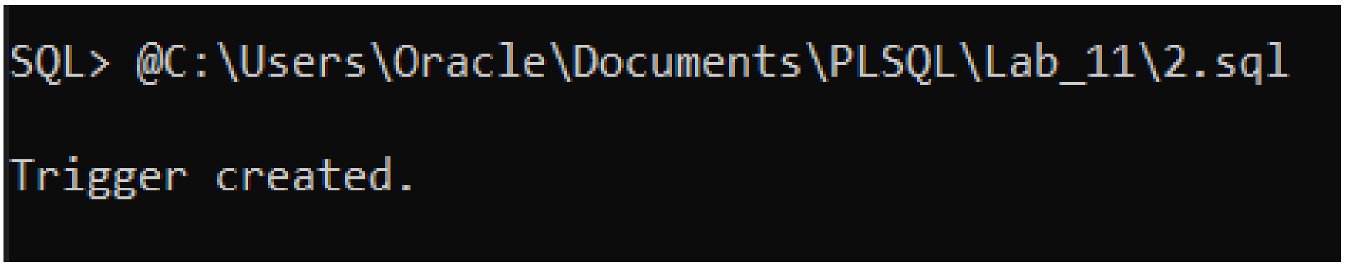


```
SQL> CREATE TABLE STUDENTS_20BRS1193 (REGNO INT PRIMARY KEY, NAME VARCHAR2(50), AGE INT, DEPT VARCHAR2(50), SEMESTER INT);

Table created.

SQL> INSERT ALL
     INTO STUDENTS_20BRS1193 VALUES (100, 'Ronald L. Perry', 18,'CSE', 2)
     INTO STUDENTS_20BRS1193 VALUES (200, 'Donna H. Bradford', 18,'EEE', 1)
     INTO STUDENTS_20BRS1193 VALUES (300, 'Amy J. Williams', 20,'ECE', 3)
     INTO STUDENTS_20BRS1193 VALUES (400, 'Kyle S. Benedict', 19,'CSE', 2)
     INTO STUDENTS_20BRS1193 VALUES (500, 'Kenneth E. Miller', 20,'ECE', 4)
     INTO STUDENTS_20BRS1193 VALUES (600, 'Teresa B. Devoe', 19,'ECE', 3)
     INTO STUDENTS_20BRS1193 VALUES (700, 'Dorthy B. Brown', 21,'CSE', 6)
     INTO STUDENTS_20BRS1193 VALUES (800, 'William J. Dunbar', 18,'EEE', 1)
     SELECT * FROM DUAL;

8 rows created.
```

TRIGGER CODE

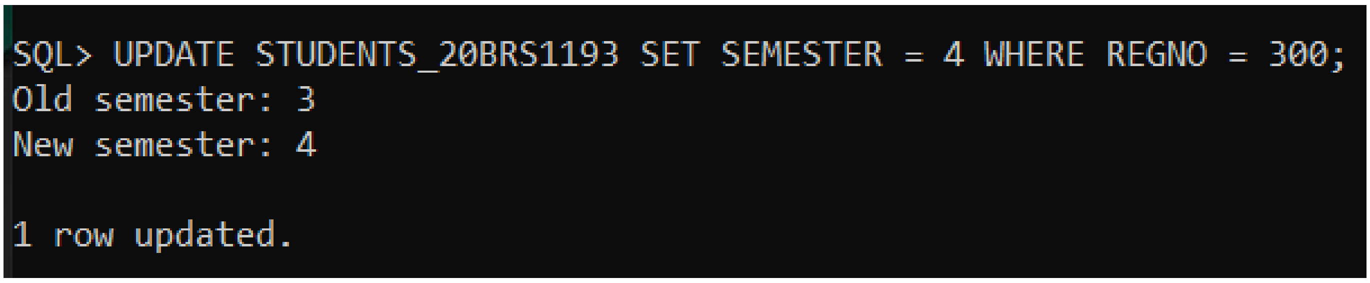
```
CREATE OR REPLACE TRIGGER display_semester_changes
BEFORE UPDATE OF SEMESTER ON STUDENTS_20BRS1193
FOR EACH ROW
WHEN(NEW.REGNO > 0)
BEGIN
    dbms_output.put_line('Old semester: ' || :OLD.semester);
    dbms_output.put_line('New semester: ' || :NEW.semester);
END;
/
```

TRIGGER CREATED



```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_11\2.sql

Trigger created.
```

USING TRIGGER



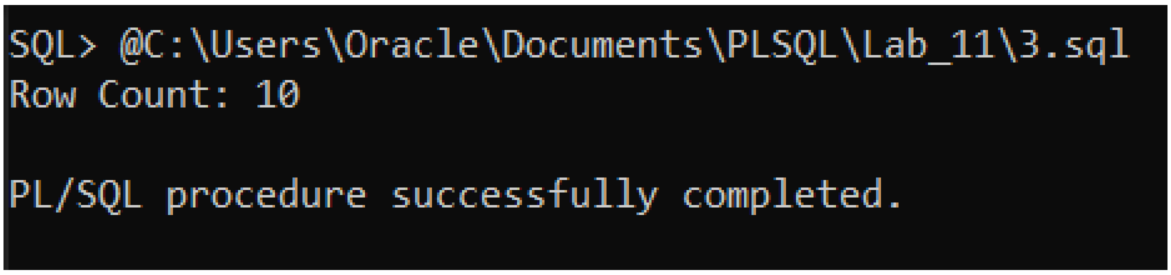
```
SQL> UPDATE STUDENTS_20BRS1193 SET SEMESTER = 4 WHERE REGNO = 300;
Old semester: 3
New semester: 4

1 row updated.
```

3. Demonstrate an example for implicit cursor – ROWCOUNT.

CURSOR CODE

```
set serverout on;
DECLARE
    TOTAL_ROWS INT;
BEGIN
    UPDATE CUSTOMERS_20BRS1193
    SET SALARY = SALARY + 0;
    TOTAL_ROWS := SQL%ROWCOUNT;
    dbms_output.put_line('Row Count: ' || TOTAL_ROWS);
END;
/
```

USING CURSOR



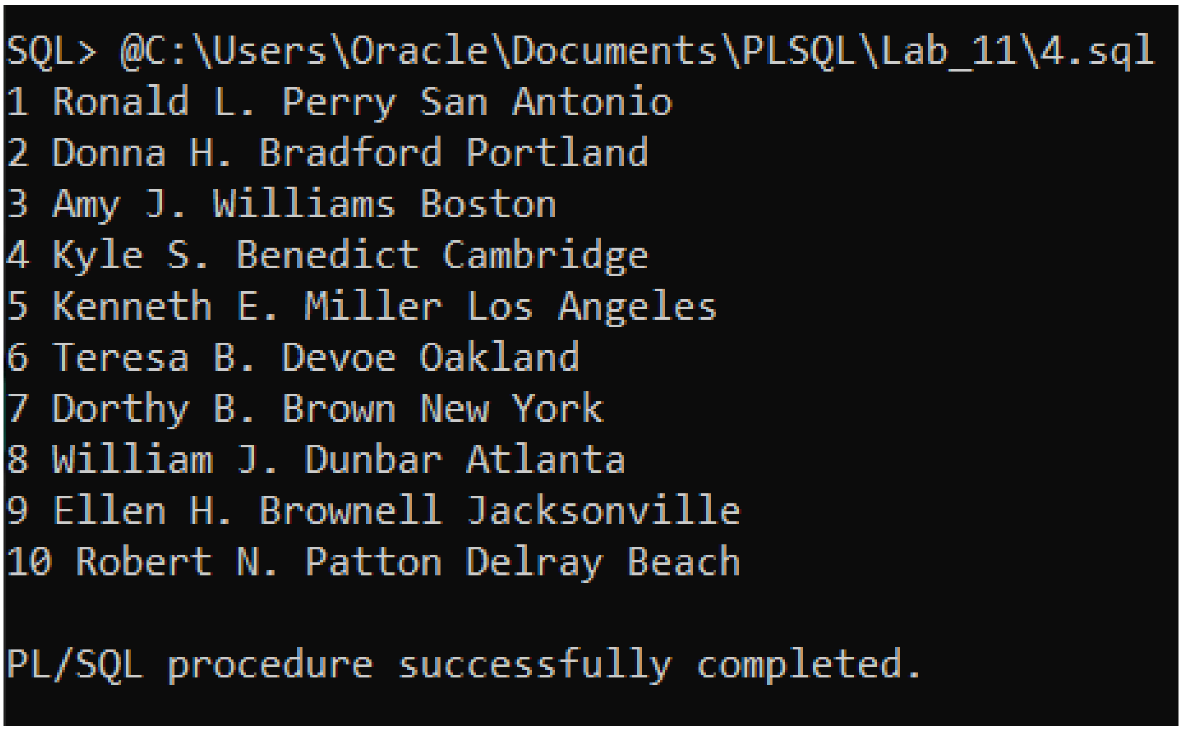
```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_11\3.sql
Row Count: 10

PL/SQL procedure successfully completed.
```

4. Create an explicit cursor named c_customers and fetch the id, name and address of all customers in the customer table using the cursor.

CURSOR CODE

```
set serverout on;
DECLARE
    CURSOR C1 IS SELECT ID, NAME, ADDRESS FROM CUSTOMERS_20BRS1193;
    CUST_ID CUSTOMERS_20BRS1193.ID%TYPE;
    CUST_NAME CUSTOMERS_20BRS1193.NAME%TYPE;
    CUST_ADDRESS CUSTOMERS_20BRS1193.ADDRESS%TYPE;
BEGIN
    OPEN C1;
        LOOP
            FETCH C1 INTO CUST_ID, CUST_NAME, CUST_ADDRESS;
            EXIT WHEN C1%NOTFOUND;
            dbms_output.put_line(CUST_ID || ' ' || CUST_NAME || ' ' || CUST_ADDRESS);
        END LOOP;
    CLOSE C1;
END;
/
```

USING CURSOR



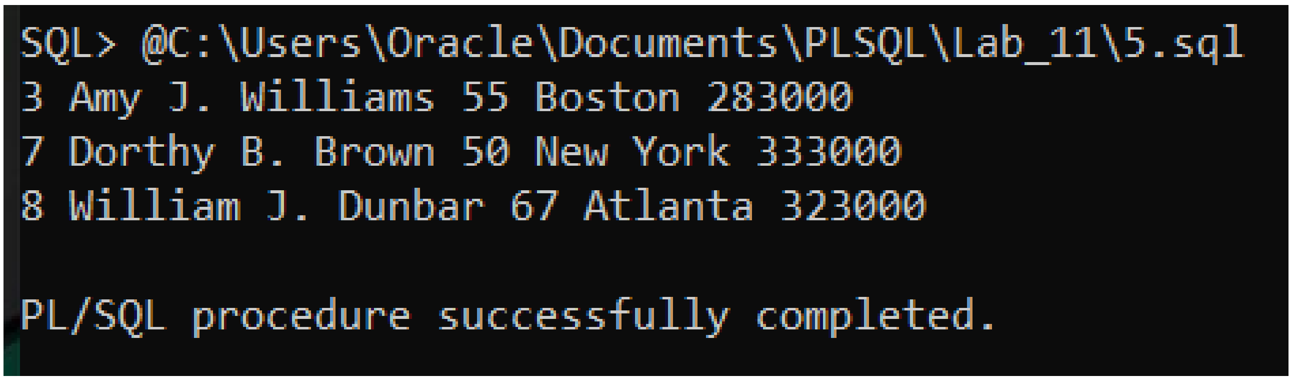
```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_11\4.sql
1 Ronald L. Perry San Antonio
2 Donna H. Bradford Portland
3 Amy J. Williams Boston
4 Kyle S. Benedict Cambridge
5 Kenneth E. Miller Los Angeles
6 Teresa B. Devoe Oakland
7 Dorthy B. Brown New York
8 William J. Dunbar Atlanta
9 Ellen H. Brownell Jacksonville
10 Robert N. Patton Delray Beach

PL/SQL procedure successfully completed.
```

5. Create an explicit cursor named c_customers and fetch the details of all customers in the customer table whose age is greater than 50 using the cursor.

CURSOR CODE

```
set serverout on;
DECLARE
    CURSOR C2 IS SELECT * FROM CUSTOMERS_20BRS1193 WHERE AGE >= 50;
    CUST_REC CUSTOMERS_20BRS1193%ROWTYPE;
BEGIN
    OPEN C2;
        LOOP
            FETCH C2 INTO CUST_REC;
            EXIT WHEN C2%NOTFOUND;
                dbms_output.put_line(CUST_REC.ID || ' ' || CUST_REC.NAME || ' ' ||
CUST_REC.AGE || ' ' || CUST_REC.ADDRESS || ' ' || CUST_REC.SALARY);
        END LOOP;
    CLOSE C2;
END;
/
```

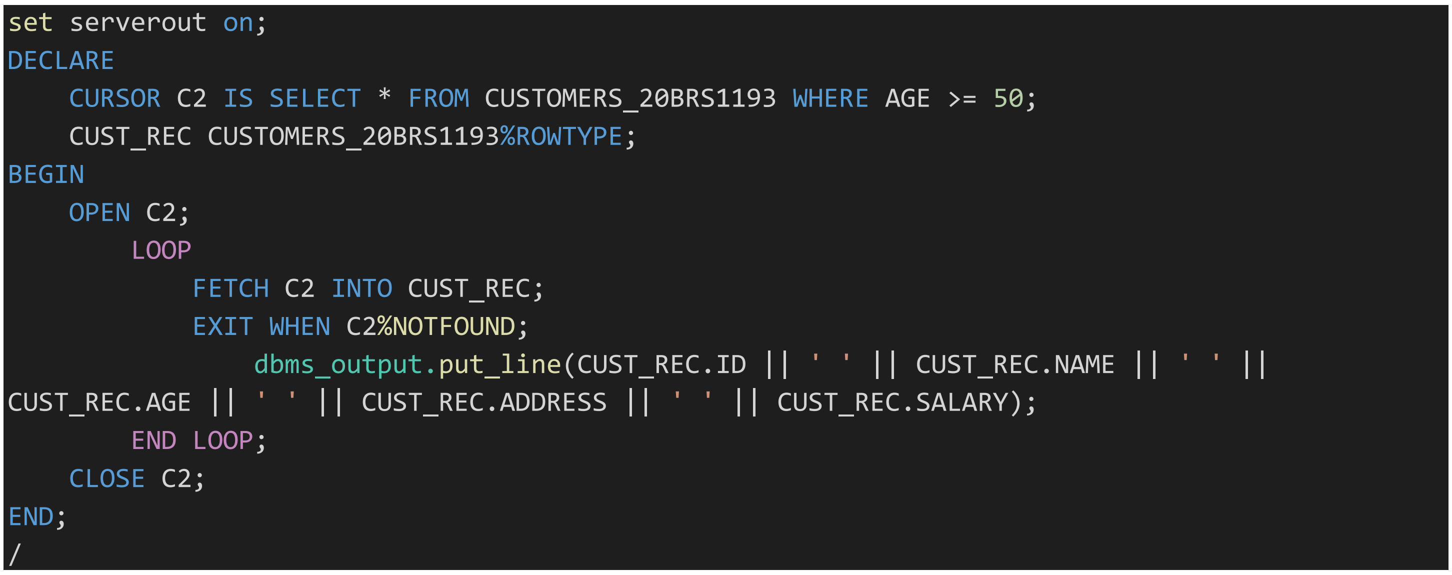
USING CURSOR



```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_11\5.sql
3 Amy J. Williams 55 Boston 283000
7 Dorthy B. Brown 50 New York 333000
8 William J. Dunbar 67 Atlanta 323000

PL/SQL procedure successfully completed.
```

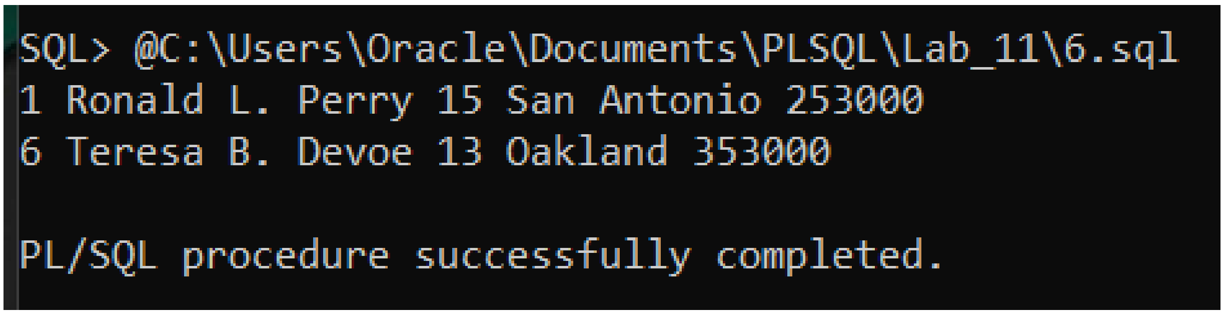
6. Create an explicit cursor named c_customers and fetch the details of all customers who are minors.

CURSOR CODE



```
set serverout on;
DECLARE
    CURSOR C2 IS SELECT * FROM CUSTOMERS_20BRS1193 WHERE AGE >= 50;
    CUST_REC CUSTOMERS_20BRS1193%ROWTYPE;
BEGIN
    OPEN C2;
        LOOP
            FETCH C2 INTO CUST_REC;
            EXIT WHEN C2%NOTFOUND;
                dbms_output.put_line(CUST_REC.ID || ' ' || CUST_REC.NAME || ' ' ||
CUST_REC.AGE || ' ' || CUST_REC.ADDRESS || ' ' || CUST_REC.SALARY);
        END LOOP;
    CLOSE C2;
END;
/
```

USING CURSOR



```
SQL> @C:\Users\Oracle\Documents\PLSQL\Lab_11\6.sql
1 Ronald L. Perry 15 San Antonio 253000
6 Teresa B. Devoe 13 Oakland 353000

PL/SQL procedure successfully completed.
```