Ahsanullah University of Science and Technology

Department of Computer Science & Engineering

Course No: CSE4126

Course Title : Distributed Database Systems Lab

Assignment No : 06

Semester: Spring 2022

Date of Submission : 25/07/23

Submitted To-

Ms.Zarin Tasnim Shejuti, Ms. Sanzana Karim Lora

Submitted By-

190204093 :MD Fardin Jaman Aranyak
Lab Group: B2
Year: 4th
Semester: 1st
Department: CSE

Answer:

```
SET SERVEROUTPUT ON;
SET VERIFY OFF;

DROP TABLE worksOn;
DROP TABLE EMPLOYEE;
DROP TABLE PROJECT;
DROP TABLE DEPARTMENT;




CREATE TABLE PROJECT(
   pId INTEGER,
   title VARCHAR(20),
   estimatedDuration INTEGER,
            primary key (pId)
);

CREATE OR REPLACE TRIGGER projectTrigg
After INSERT
ON PROJECT
DECLARE
BEGIN
            DBMS_OUTPUT.PUT_LINE('PROJECT Table is created with values');
END;
/
INSERT INTO project VALUES (1, 'AB', 2);
INSERT INTO project VALUES (2, 'CD', 5);
INSERT INTO project VALUES (3, 'EF', 3);
INSERT INTO project VALUES (4, 'GH', 1);
INSERT INTO project VALUES (5, 'IJ', 4);


CREATE TABLE DEPARTMENT (
  dID INTEGER,
  dName VARCHAR(20),
  supID INTEGER,
  primary key (dID)
);

CREATE OR REPLACE TRIGGER departmentTrigg
After INSERT
ON DEPARTMENT
DECLARE
BEGIN
        DBMS_OUTPUT.PUT_LINE('DEPARTMENT Table is created with values');
END;
/
INSERT INTO department VALUES (1, 'HR', 5);
INSERT INTO department VALUES (2, 'Customer Service', 3);
INSERT INTO department VALUES (3, 'IT', 6);
INSERT INTO department VALUES (4, 'FINANCE', 4);


CREATE TABLE EMPLOYEE(
   eID INTEGER,
   eName VARCHAR(20),
   age INTEGER,
   gender VARCHAR(20),
   salary INTEGER,
   dID INTEGER,
        primary key (eID),
        FOREIGN KEY (dID) REFERENCES department(dID)
);
CREATE OR REPLACE TRIGGER employeeTrigg
After INSERT
ON EMPLOYEE
DECLARE
BEGIN
        DBMS_OUTPUT.PUT_LINE('EMPLOYEE Table is created with values');
END;
/
INSERT INTO EMPLOYEE VALUES (1, 'A', 32, 'M', 32000, 3);
INSERT INTO EMPLOYEE VALUES (2, 'B', 45, 'F', 39000, 2);
INSERT INTO EMPLOYEE VALUES (3, 'C', 48, 'M', 40000, 2);
INSERT INTO EMPLOYEE VALUES (4, 'D', 25, 'F', 25000, 4);
INSERT INTO EMPLOYEE VALUES (5, 'E', 56, 'M', 60000, 1);
INSERT INTO EMPLOYEE VALUES (6, 'F', 49, 'F', 42000, 3);



CREATE TABLE WorksOn(
        pId INTEGER,
        eId INTEGER,
        FOREIGN KEY (pId) REFERENCES PROJECT(pId),
        FOREIGN KEY (eId) REFERENCES EMPLOYEE(eId)
);

CREATE OR REPLACE TRIGGER WorkOn
After INSERT
ON WorksOn
```

```
DECLARE

BEGIN

        DBMS_OUTPUT.PUT_LINE('WorksOn Table created with values');

END;

/

INSERT INTO WorksOn VALUES (1, 2);

INSERT INTO WorksOn VALUES (2, 3);

INSERT INTO WorksOn VALUES (3, 4);

INSERT INTO WorksOn VALUES (3, 5);

INSERT INTO WorksOn VALUES (3, 6);

INSERT INTO WorksOn VALUES (4, 3);

INSERT INTO WorksOn VALUES (5, 4);

INSERT INTO WorksOn VALUES (1, 5);



commit;


Accept x number prompt "Enter the dId:  "



CREATE OR REPLACE PACKAGE myPack AS

FUNCTION OldestAge(i IN DEPARTMENT.dId%TYPE)

RETURN EMPLOYEE.age%TYPE;

PROCEDURE EmployeeSearch(s IN EMPLOYEE.age%TYPE);

end myPack;

/


CREATE OR REPLACE PACKAGE Body myPack AS


FUNCTION OldestAge(i IN DEPARTMENT.dId%TYPE)

RETURN EMPLOYEE.age%TYPE
```

```
IS
  X EMPLOYEE.age%TYPE;

  CURSOR age
  IS
  SELECT MAX(EMPLOYEE.age)  FROM
  EMPLOYEE INNER JOIN WorksOn ON EMPLOYEE.eID = WorksOn.eID
  WHERE EMPLOYEE.dID = i;

BEGIN
           OPEN age;
     FETCH age INTO X;
           CLOSE age;
           RETURN X;
end OldestAge;



PROCEDURE EmployeeSearch(s IN EMPLOYEE.age%TYPE)
IS
BEGIN
           FOR R IN (select EMPLOYEE.eName,EMPLOYEE.age,EMPLOYEE.gender,EMPLOYEE.salary,DEPARTMENT.dName FROM EMPLOYEE INNER JOIN DEPARTMENT ON EMPLOYEE.dID = DEPARTMENT.dID WHERE EMPLOYEE.age = s) LOOP
                      DBMS_OUTPUT.PUT_LINE('Name: ' || R.eName);
                      DBMS_OUTPUT.PUT_LINE('Age: ' || R.age);
                      DBMS_OUTPUT.PUT_LINE('Gender: ' || R.gender);
                      DBMS_OUTPUT.PUT_LINE('Salary: ' || R.salary);
                      DBMS_OUTPUT.PUT_LINE('Department: ' || R.dName);
           END LOOP;
END EmployeeSearch;
END myPack;
/

DECLARE
        dID NUMBER;
   ex EXCEPTION;
        oldest employee.age%TYPE;

BEGIN
   dID := &x;
   oldest := myPack.OldestAge(dID);

        IF oldest IS NULL THEN
                RAISE ex;
        END IF;
        myPack.EmployeeSearch(oldest);
EXCEPTION
        WHEN ex THEN
                DBMS_OUTPUT.PUT_LINE('Department does not exist');
END;
/
```