Exercises on Triggers and Cursors

Batch 1: Roll Nos. COE 1 – COE 30

1)a) Write the following as triggers(*Any four*) on the following EMPLOYEE Schema. In each case, disallow if it does not satisfy the stated constraint. You may assume that the desired condition holds before any change to the database is attempted. Also, prefer to modify the database, even if it means inserting tuples with NULL or default values, rather than rejecting the attempted modification.

EMPLOYEE(Fname:String, Minit: string, Lname: string, Ssn: String, Bdate: DATE, Address: String, Sex: String, Salary: decimal, Super_ssn:string, Dno: Integer)
DEPARTMENT(Dname: String, Dnumber: Integer, Mgr_ssn: String, Mgr_start_date: DATE, **UNIQUE** (Dname))
DEPT_LOCATIONS(Dnumber: Integer, Dlocation: String)
PROJECT(Pname: String, Pnumber: Integer, Plocation: String, Dnum: Integer), **UNIQUE** (Pname))
WORKS_ON(Essn: String, Pno: Integer, Hours: DECIMAL)
DEPENDENT(Essn: String, Dependent_name: String, Sex: String, Bdate: DATE, Relationship: String)

1. Assure that deleting details of an employee deletes his dependent records also.
2. Whenever a department with exactly one project is shifted to a new location, ensure that the project is also shifted to the new location.
3. Assure at all times that there are no departments with more than 3 projects.
4. Assure that no employees work for more than one department.
5. Whenever a project is dropped, dissociate all the employees from the particular project.
6. When a new department is inaugurated, ensure that it is not co-located with any other departments.
7. For every employee, ensure that his dependent Bdate is less than his Bdate.
8. Increment 1000 rs to the salary for those employees if any of his/her dependent expire.

b) Write the following as Cursors on the EMPLOYEE Schema.

1. Develop a stored procedure to insert a new attribute ‘address’ in DEPENDENT and update the same as that of the employee’s address.

2. Develop a stored procedure to display the fname, ssn and salary, grade of an employee. Handle the condition such that if salary of an employee is 1 - 10000, assign grade3, grade2 if salary in between 10000 and 50000 and grade1 if salary > 50000. Handle exception with an error message when an invalid case occurs.

3. Create a stored procedure to display deptno, avgsalary and #employees in each department. Handle exceptions with an error message when invalid deptno is given.

Batch 2: Roll Nos. COE31 – CED 13 + Branch Transfers

2) a) Write the following as triggers(*Any four*) on the following FLIGHT Schema. In each case, disallow if it does not satisfy the stated constraint. You may assume that the desired condition holds before any change to the database is attempted. Also, prefer to modify the database, even if it means inserting tuples with NULL or default values, rather than rejecting the attempted modification.

Flights(flno: integer, from: string, to: string, distance: integer, Departs:time, arrives:time, price:integer)
Aircraft(aid: integer, aname: string, cruisingrange: integer)
Certified(eid: integer, aid: integer)
Employees(eid: integer, ename: string, salary: integer)

1) Create a trigger that handles an update command to find the total salary of all pilots. Check the condition such that the new tuples inserted should not be null and salary should be more than 50,000.

2) Create a trigger to set salary as 30,000 if there is a NULL present in it. Also check whether a salary of a pilot is greater than the salary of a non pilot.

3) Create a trigger to foil any attempt to lower the salary of an employee.

4) When inserting a new certification for an employee, check that the aircraft id exists in the Aircraft.
5) When making any modifications to the Aircraft table, check that the cruisingrange is greater than or equal to distance of flights.

6) When a new certification is inserted into Certified, also insert an employee with the id of that employee and a NULL salary.

7) Terminate pilots and their certification when the pilot retires.

10) Write a trigger for the condition mentioned: Suppose we want to prevent the average salary of an employee from dropping below Rs. 50,000. This constraint could be violated by an insertion, a deletion, or an update to the salary column of Employee Table.

b) Write the following as Cursors on the FLIGHT Schema.

1) Develop a stored procedure to update an employee record given the employee id. Print a message after the update is successfully done with an exception handling of a invalid employee id.

2) Develop a stored procedure to display the name, salary of each employee from employee table. Handle the condition such that if salary of an employee is above 50,000 rank them as Grade ‘A’ else as Grade ‘B’.

3) Develop a stored procedure that builds a name list of all employees who are certified for a Boeing aircraft and handle an exception with an error message.

Batch 3: Roll Nos. CED 14 – CED 41

3)a) Write the following as triggers(*Any four*) on the following SUPPLIERS Schema. In each case, disallow if it does not satisfy the stated constraint. You may assume that the desired condition holds before any change to the database is attempted. Also, prefer to modify the database, even if it means inserting tuples with NULL or default values, rather than rejecting the attempted modification.

Supplier(sno: integer, sname:string, status:integer, city: string)
Part(pno:integer, pname:string, color: string, weight:integer, city:string)
Project(jno:integer, jname:string, city:string)
Spj(sno, pno, jno:integer, qty:number)

1. Assure that deleting details of a Part deletes its SPJ records also.

2. Write a trigger for the condition mentioned: Suppose we want to prevent the average quantity of a SPJ table from dropping below Rs. 50. This constraint could be violated by an insertion, a deletion, or an update to the quantity column of SPJ Table.

3. When inserting a new SPJ for an order, check that the supplier id exists in the Supplier.

4. Create a trigger to foil any attempt to lower the status of a supplier.

5. Assure that no Project exists in more than 2 cities.

6. Create a trigger to set quantity in SPJ as 30 if there is a NULL present in it. Also ensure that a quantity of all parts with red color is greater than the quantity of all green parts.

7. Whenever a project is dropped (deleted), decrement the status of those suppliers supplying at least one part to that project.

8. When a new part is inserted with a weight greater than 1kg, allow the insertion, but change the weight to 1kg.

b) Write the following as Cursors on the SUPPLIERS Schema.

1) Develop a stored procedure to update an Part record given the Part id. Print a message after the update is successfully done with an exception handling of a invalid Part id.

2) Develop a stored procedure to display the projectno, project name, city and Rank of a Project. Handle the condition such that a project is of Rank 1 if the total quantity of the parts for that project is 1-100, Rank 2 if the total quantity is in between 100 – 200 and Rank 3 if it is above 200. Handle exception with an error message when an invalid case occurs.

3) Develop a stored procedure that builds a name list of all parts which are assigned to Projectid 5 and handle an exception with an error message.
