

CMPT 354 –Database Systems I (Section D100)

Assignment #7

Instructor: Richard Frank (rfrank@sfu.ca)

TA: Ankit Gupta (aga53@sfu.ca)

Total Marks: 50 (5% of the Individual Assignments)

Due Date: Nov 4, 14:30

For the below questions consider the relation below, and assume that firstname/lastname are unique.

Employees(EmpID integer, firstname varchar(50), lastname varchar(50), dateofbirth date)

Wherever you are required to *implement*, ensure it works in SQL Server 2008 R2.

Question 1) [20 marks]

My business, CMPT'R'Us, wants to hire you to help with the management of the employee database. We need to create a way (i.e. database procedure) to insert employee data into this table. According to our practice and our current database setup, we have the following restrictions:

- Although EmpID is the primary key, it does not autogenerate.
- If the employee already exists, we need an error message.
- If multiple instances of the employee exist, we must assume that John Doe mucked around in the database, so we discipline him, but must delete all instances of the employee and replace it with the values that are incoming from the user.
- If an employee does not exist in the database, we must store them.

We need a database procedure that will always keep the database in a consistent state, even after a failure somewhere in the system. Please provide instructions on how to call the procedure.

```
CREATE PROCEDURE AddEmployee
    @FirstName VARCHAR(50),
    @LastName VARCHAR(50),
    @dateofbirth DATE
AS
BEGIN TRANSACTION;
DECLARE @CountOfEmployees AS INTEGER;
DECLARE @MaxID AS INTEGER;

--find how many instances of this person are there
SELECT @CountOfEmployees = COUNT(*)
FROM Employees
WHERE FirstName = @FirstName AND LastName = @LastName;

IF @CountOfEmployees <> 1
BEGIN
    IF @CountOfEmployees > 1
    BEGIN
        --delete all instances
        DELETE FROM Employees WHERE FirstName = @FirstName AND LastName =
@LastName;
        IF @@ERROR <> 0 GOTO ErrorExit;
        SET @MaxID = 0;
    END
END
```

CMPT 354 –Database Systems I (Section D100)

```
        END
        --get the MAX ssn
    ELSE
        SELECT @MaxID = MAX(empid) FROM Employees;

        INSERT INTO Employees VALUES(@MaxID + 1, @FirstName, @LastName,
@dateofbirth);
        IF @@ERROR <> 0 GOTO ErrorExit;
        --straight insert
    END
    ELSE
        --print an error message
        Print 'Employee exists already';

    COMMIT
ErrorExit:
    ROLLBACK TRANSACTION;

GO
```

Question 2) [10 marks]

Implement a database function to calculate the Manhattan Distance, given two XY coordinate pairs. Please provide instructions on how to use the function.

```
CREATE FUNCTION ManhattanDistance(@X1 FLOAT, @Y1 FLOAT, @X2 FLOAT, @Y2 FLOAT)
RETURNS FLOAT
AS
BEGIN
    DECLARE @ManDist AS FLOAT;
    SET @ManDist = @X2-@X1 + @Y2-@Y1;
    RETURN(@ManDist);
END;
GO

SELECT dbo.ManhattanDistance(1,1,2,2) FROM Employees
```

Question 3) [5 marks]

In Question 1, John Doe was able to muck around in the database and cause duplicate employees (i.e. duplicate firstname/lastname combinations). Assuming SQL Server 2008 supported Tuple-Based Checks (which it does not, nor does it support assertions), how would you stop 2 people with the same name to be inserted? Write the SQL code to do this (again, note that SQL Server 2008 doesn't support this).

```
CREATE TABLE Employees(
    EmpID integer,
    firstname varchar(50),
    lastname varchar(50),
    dateofbirth date
```

```
CHECK
(2 < (SELECT COUNT(*)
      FROM Employees E
      WHERE E.FirstName = FirstName
            AND E.LastName = LastName));
```

Question 4) [5 marks]

Given that SQL Server 2008 does not support Tuple-Based Checks, nor Assertions, but you still wanted to implement this restriction at the database level. What could you do? Do not actually implement anything, just list the strategies you could try.

- Use a trigger
- Create a function to count the number of employees of that name. Then use the function in an attribute-based check.
- Define firstname/lastname to be unique, i.e.: have *ADD UNIQUE (firstname, lastname)* as part of the table-definition.

Question 5) [5 marks]

Create a trigger which copies employees who are above the age of 50 into another table *ExperienceEmployees*.

```
CREATE TRIGGER CopyExperiencedEmployees
ON Employees
AFTER INSERT
AS BEGIN

    INSERT INTO ExperiencedEmployees
    SELECT * FROM Inserted
    WHERE dateofbirth < GETDATE() - 50;

END
```

Question 6) [5 marks]

We talked about transactions quite a bit. Can transactions be nested? That is, have a *transaction* within a transaction? If so, how do COMMIT and ROLLBACK work?

Yes, it's possible.

COMMIT only commits the last (innermost) transaction that was declared.

ROLLBACK rolls back everything, regardless of how many transactions were started.