

|   |                                       |
|---|---------------------------------------|
| <b>Course Name: Database Systems</b>                              | <b>Course Code: CS363L</b>            |
| <b>Assignment Type: Lab</b>                                       | <b>Dated: 28-03-2022</b>              |
| <b>Semester: 6<sup>th</sup></b>                                   | <b>Session: 2019</b>                  |
| <b>Lab/Project/Assignment #: Lab 9</b>                            | <b>CLOs to be covered: CLO2, CLO3</b> |
| <b>Lab Title: Data Definition Language and Data Modifications</b> | <b>Teacher Name: Ms. Darakhshan</b>   |

### Lab Evaluation:

|                       |   |               |               |               |               |               |
|-----------------------|---|---------------|---------------|---------------|---------------|---------------|
| <b>CLO2</b>           | Construct DDL queries to manage relations, constraints, triggers and indexes. |               |               |               |               |               |
| <b>Levels (Marks)</b> | <b>Level1</b>   | <b>Level2</b> | <b>Level3</b> | <b>Level4</b> | <b>Level5</b> | <b>Level6</b> |
| Cognitive (5)         |   |               |               |               |               |               |
| <b>Total</b>          |   |               |               |               |               | <b>/5</b>     |
| <b>CLO3</b>           | Derive physical model from conceptual design methods                          |               |               |               |               |               |
| <b>Levels (Marks)</b> | <b>Level1</b>   | <b>Level2</b> | <b>Level3</b> | <b>Level4</b> | <b>Level5</b> | <b>Level6</b> |
| Cognitive (5)         |   |               |               |               |               |               |
| <b>Total</b>          |   |               |               |               |               | <b>/5</b>     |

### Rubrics for Current Lab:

| Scale                    | Marks    | Level | Rubric  |
|--------------------------|----------|-------|---|
| <b>Excellent</b>         | <b>5</b> | L1    | Triggers added with DDL + Rubric IV requirements.                                     |
| <b>Very Good</b>         | <b>4</b> | L2    | Constraints added with DDL + Rubric III requirements.                                 |
| <b>Good</b>              | <b>3</b> | L3    | Database Schema Created using DDL + Rubric II requirements                            |
| <b>Basic</b>             | <b>2</b> | L4    | Attributes, Datatypes and Constraints are properly identified + Rubric I requirements |
| <b>Barely Acceptable</b> | <b>1</b> | L5    | All implicit/explicit requirements are clearly identified and written                 |
| <b>Not Acceptable</b>    | <b>0</b> | L6    | Lab missed or solved none of the problems   |

## LAB DETAILS:

### Lab Goals/Objectives:

- Data Definition Language (CREATE, UPDATE, DELETE, INSERT, SET, ALTER)
- Constraints and Triggers

## **Theory/Relevant Material:**

### **Client Requirements:**

- Merchants from India are banned.
- Customer cannot order less than 50 items of same product.
- Email address of any user should be valid
- Valid product statuses: **A:** Available (at least 50 items available), **NA:** (if less than 50 items of same product are available, then product cannot be sold)
- Only continent name in country table can be left empty while data entry.

### **Tutorial:**

#### **Adding Constraints:**

Given below is an example for adding constraints to a table in SQL. Below is a constraint which checks that the user whose data is being inserted into user\_profile table, is not an admin. There will only be one admin.

#### **Users Table:**

| id | Name  | Rank  |
|----|-------|-------|
| 1  | John  | User  |
| 2  | David | User  |
| 3  | Dave  | Admin |

#### **User Profile Table:**

| Id | Employee_code | User_id |
|----|---------------|---------|
| 1  | 200323        | 1       |
| 2  | 200324        | 3       |

```
CREATE FUNCTION dbo.isNotAdmin (@code int)
RETURNS VARCHAR(5)
AS
BEGIN
    IF @code <> (SELECT id FROM users WHERE rank = 'Admin')
        return 'False'
    return 'True'
END;
GO
```

```
CREATE TABLE user_profile (
id int PRIMARY KEY,
employee_code int NOT NULL,
user_id CONSTRAINT isNotAdmin CHECK (dbo.isNotAdmin(user_id) = 'True') int NOT NULL,
);
```

#### **Adding Triggers:**

Given below is an example for adding triggers to a table in SQL.

When an employee's salary is increased beyond 75,000 his rank is upgraded from Associate to Senior.

**University Of Engineering and Technology, Lahore**  
**Computer Engineering Department**

```
CREATE TRIGGER dbo.rankTrigger ON  dbo.employees
AFTER INSERT AS
BEGIN
DECLARE @salary  INT;
DECLARE @_rank  NVARCHAR(3);

DECLARE @salary_cursor as CURSOR;

SET @salary_cursor = CURSOR FOR

SELECT salary,_rank
FROM employees;

OPEN @salary_cursor;
FETCH NEXT FROM @salary_cursor INTO @salary , @_rank;

WHILE @@FETCH_STATUS = 0
BEGIN
    IF(@salary > 75000)
        UPDATE employees
        SET _rank = 'SENIOR'
    ELSE
        UPDATE employees
        SET _rank = 'ASSOCIATE'
        PRINT ('else')
    PRINT cast(@salary as VARCHAR (50)) + ' ' + @_rank;
    FETCH NEXT FROM @salary_cursor INTO @salary , @_rank;
END

CLOSE @salary_cursor;
DEALLOCATE @salary_cursor;
END
```

**Lab Tasks:**

- Create tables using Data Definition Language. You will not use SQL Server Management Studio interface for creating schema scripts. You must write commands.
- Identify all the constraints.

**Homework Questions:**

- Add all constraints using SQL (Primary key, Foreign keys and any other).
- Identify triggers.
- Add triggers to your schema.
- Add at least 20 dummy rows using INSERTs.
- Client has changed its requirement and now merchants from India are allowed. ALTER your constraint accordingly.

**Submission Instructions:**

Name your query files as DBLab9\_2019\_CE\_X.sql, add supporting SQL scripts of your homework and submit on google classroom by Sunday, 6<sup>th</sup> April, 2022 9 P.M