****

USMAN INSTITUTE OF TECHNOLOGY

Department of Computer Science

**CS311 Introduction to Database Systems**

**Fall 2019**

**Lab # 13** Open ended Lab

**Objective:**

.

SHOAIB AHMED

**Name of Student:**

17B-033-SE

A

**Roll No: Section:**

17-JAN-2020

**Date of Experiment:**

24-JAN-2020

**Date Submitted:**

Marks Obtained/Remarks: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Open Ended Experiment Rubrics:**

Following rubrics will be followed while marking the experiment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Marks**  **Criteria** | **1** | **2** | **3** | **4** | **5** |
| Performance | The student has put no effort in the experiment and shows no progress | The student has put some effort in the experiment but shows no progress | There is a reflection of effort in the experiment, but the solution is inappropriate | Student has completed the major req. of the experiment but unreasonable in some aspects | Student has completed the experiment along with all the requirements |
| Viva | Student is unable to explain the method / procedure and has lacking in the basic concepts | Student is unable to explain the method / procedure but has clear basic concepts | Student is able to explain the procedure but cannot defend why was that particular procedure followed | Student is able to explain and defend the procedure followed but can Not provide any alternative method | Student is able to explain and defend the procedure followed and is able to provide an alternative method |

Experiment will be marked “zero” if found copied or cheated.

**Scenario:**

You have been hired by some Super Store to maintain database for their products using SQL Developer environment. Unfortunately, no other software is available for building a GUI and you have to build up manual queries to add new entries or show data available in the database.

You should be able to do the following tasks:

1. Create a table having name as **your full name.** 
   1. Table should have the following columns:
      1. Item ID Integer (primary key, not null & generated with sequence)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| It\_ID | It\_Code | It\_Na | It\_Pri | It\_qty |
| 1 | NK\_42 | USB | 500 | 15 |
| …… | …… | …… | …… | …… |

* + 1. Item Code Varchar2
    2. Item Name Varchar2
    3. Item Price Number
    4. Item Qty Number

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Or\_ID | IT\_ID | Or\_Qty | Or\_Pri | Cus\_ID |
| 1 | 36 | 2 | 200 | 5 |
| 2 | 43 | 7 | 3500 | 5 |
| 3 | 67 | 5 | 2500 | 11 |
| … | …. | … | …. | …. |

1. Create another table having name as **your first name \_SE**
   1. Table should have the following columns:
      1. Order ID Integer
      2. Item ID Integer (Foreign Key)
      3. Order Qty Integer
      4. Order Price Integer
      5. Customer ID Integer
2. Make some query to insert 10 rows in both tables as order by any customer of a particular item.
3. During the insertion in Second table if order qty is not available or less, a message should be generated telling the user “Item not available!” (**hint:** create a trigger on insert in second table to check the order quantity should be less then item quantity in first table)

Scenario # 2

**Tables:**

**Highschooler(ID int, name text, grade int);**

**Friend(ID1 int, ID2 int);**

**Likes(ID1 int, ID2 int);**

1 - Write one or more triggers to maintain symmetry in friend relationships. Specifically, if (A,B) is deleted from Friend, then (B,A) should be deleted too. If (A,B) is inserted into Friend then (B,A) should be inserted too. Don't worry about updates to the Friend table

2 - Write a trigger that automatically deletes students when they graduate, i.e., when their grade is updated to exceed 12. In addition, write a trigger so when a student is moved ahead one grade, then so are all of his or her friends.

3 - Write a trigger to enforce the following behavior: If A liked B but is updated to A liking C instead, and B and C were friends, make B and C no longer friends. Don't forget to delete the friendship in both directions, and make sure the trigger only runs when the "liked" (ID2) person is changed but the "liking" (ID1) person is not changed.