|  |  |
| --- | --- |
| Marie Kelling | |
|  | Your Answers |
| **1** | Before INSERT trigger |
| Screen shot of the result grid from SELECT statement shown in the homework |
| SQL that creates your trigger, including the UPDATE statement  -- 14.1 Marie Kelling  -- Insert trigger that ensures the Email of people added to the facutly table is in the proper format  USE College;  DROP TRIGGER IF EXISTS Faculty\_Before\_Insert ;  DELIMITER $$  CREATE TRIGGER Faculty\_Before\_Insert  BEFORE INSERT ON Faculty  FOR EACH ROW  BEGIN    SET NEW.Email = CONCAT(NEW.FirstName, NEW.LastName, '@college.edu');    END $$  DELIMITER ;  INSERT INTO Faculty (LastName, FirstName, Email, HireDate, Salary, DepartmentID)  VALUES ('Kelling', 'Marie', 'bad@bad.bad', '2017-01-01', 60000.00, 1) ; |
| **2** | Before UPDATE trigger |
| Screen shot of the Output Window showing the output both UPDATE statements |
| SQL that creates your trigger  -- 14.2 Marie Kelling  -- Create trigger that will prevent an update to a student table if EnrolledDate is in the future  USE College ;  DROP TRIGGER IF EXISTS Student\_Before\_Update;  DELIMITER $$  CREATE TRIGGER Student\_Before\_Update  BEFORE UPDATE ON Student  FOR EACH ROW  BEGIN  IF NEW.EnrolledDate > CURDATE()  THEN  SIGNAL SQLSTATE VALUE '45000'  SET MESSAGE\_TEXT= 'Enrolled date may not be in the future';  END IF;  END $$  DELIMITER ; |

|  |  |
| --- | --- |
| **3** | Add another rule of your choosing |
| Screen shot of the Output Window showing the output with the error. |
| SQL that creates your trigger  -- 14.3 Marie Kelling  -- Adds rule to the Student\_Before\_Update Trigger that prevents students born before the year 2000 from enrolling  USE College ;  DROP TRIGGER IF EXISTS Student\_Before\_Update;  DELIMITER $$  CREATE TRIGGER Student\_Before\_Update  BEFORE UPDATE ON Student  FOR EACH ROW  BEGIN  IF NEW.EnrolledDate > CURDATE()  THEN  SIGNAL SQLSTATE VALUE '45000'  SET MESSAGE\_TEXT= 'Enrolled date may not be in the future';  END IF;    IF NEW.DateOfBirth > '2000-01-01'  THEN  SIGNAL SQLSTATE VALUE '45000'  SET MESSAGE\_TEXT= 'DateOfBirth not valid';  END IF;    END $$  DELIMITER ;  UPDATE Student  SET DateOfBirth = DATE(NOW())  WHERE ID = 1; |

|  |  |
| --- | --- |
| **4** | After UPDATE trigger |
| Screen shot of the Output Window showing the output both UPDATE statements |
| SQL that creates your trigger  USE College ;  DROP TRIGGER IF EXISTS Student\_After\_Update;  DELIMITER $$  CREATE TRIGGER Student\_After\_Update  AFTER UPDATE ON Student  FOR EACH ROW  BEGIN  INSERT INTO SysLog (TableName, Message, Created)  VALUES ('Student', CONCAT('Updated ID = ', New.ID), NOW());  END $$  DELIMITER ; |

|  |  |
| --- | --- |
| **5** | Before Insert Trigger on Registration |
| Paste the text of your INSERT statement and a screen shot of the Output Window showing the output from two runs of the statement.  INSERT INTO Registration(StudentID, SectionID, Grade)  VALUES(1, 2, 3.9);  INSERT INTO Registration(StudentID, SectionID, Grade)  VALUES(1, 2, 3.9); |
| SQL that creates your trigger  -- 14.5 Marie Kelling  -- Create Trigger to prevent a Student from registering for a Section that the student is already registered for  USE college;  DROP TRIGGER IF EXISTS Registration\_Before\_Insert;  DELIMITER $$  CREATE TRIGGER Registration\_Before\_Insert  BEFORE INSERT ON Registration  FOR EACH ROW  BEGIN    DECLARE SectionCount INT;  SELECT COUNT(\*) INTO SectionCount  FROM Registration  WHERE StudentID = NEW.StudentID AND NEW.SectionID = SectionID;  IF SectionCount > 0 THEN  SIGNAL SQLSTATE VALUE '45000'  SET MESSAGE\_TEXT= 'Student is already registered for this section';  END IF ;  END $$  DELIMITER ; |