Indian Institute of Information Technology Surat

Lab Report on
Advanced Database Management (CS 604) Practical

Submitted by

[RAHUL KUMAR SINGH] (UI21CS44)

Course Faculty

Mr. Rishi Sharma

Department of Computer Science and Engineering
Indian Institute of Information Technology Surat
Gujarat-394190, India

Jan-2024

Lab No: 4

Aim: Design and develop a suitable Student Database application. One of the attributes to me maintained is the attendance of a student in each subject for which he/she has enrolled.

Description: Using TRIGGERS, we write active rules to do the following:

a) Whenever attendance is updated, check if the attendance is less than 85%; if so notify the Head of Department concerned.

b) Whenever the marks in the Internal Assessment Test are entered, check if the marks are less than 40%; if so, notify the Head of the Department concerned.

Source Code:

Student_Attendance Table:

```
CREATE TABLE Student_Attendance (
        t_no INT PRIMARY KEY,
        Day1 TINYINT,
        Day2 TINYINT,
        Day3 TINYINT,
        Day4 TINYINT,
        Day5 TINYINT,
        Day6 TINYINT,
        Day7 TINYINT,
        Day8 TINYINT,
        Day9 TINYINT,
        Day10 TINYINT
);
```

Student_Marks Table:

```
CREATE TABLE Student_Marks (
        t_no INT PRIMARY KEY,
        Sub1 DECIMAL(5,2),
        Sub2 DECIMAL(5,2),
        Sub3 DECIMAL(5,2),
        Sub4 DECIMAL(5,2),
        Sub5 DECIMAL(5,2),
        Sub6 DECIMAL(5,2)
);
```

Insertion:

```
INSERT INTO `Student_Attendance` VALUES (5,0,0,1,0,1,0,1,0,1,1);
INSERT INTO `Student_Attendance` VALUES (6,0,1,1,1,1,1,1,1,1,1);

INSERT INTO `Student_Marks` VALUES (5,20.0,20.0,30.0,50.0,60.0,20.0);
INSERT INTO `Student_Marks` VALUES (6,50.0,40.0,30.0,50.0,60.0,30.0);
```

Task 1:

```
DELIMITER //
CREATE TRIGGER after_insert_attendance_student
AFTER INSERT ON Student_Attendance
FOR EACH ROW
BEGIN
  IF
((NEW.Day1+NEW.Day2+NEW.Day3+NEW.Day4+NEW.Day5+NEW.Day6+NEW.Day7+NEW.Day8+NEW
.Day9+NEW.Day10)*(100/10) < 85) THEN
        -- CALL NotifyHeadOfDepartment(NEW.t_no, 'Low internal assessment marks');
        SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = "Notice: Attendance for the entered student
is less than 85%";
  END IF;
END;
//
DELIMITER ;
```

Task 2:

```
DELIMITER //
CREATE TRIGGER after_insert_marks_student
AFTER INSERT ON Student_Marks
FOR EACH ROW
BEGIN
  IF ((NEW.Sub1+NEW.Sub2+NEW.Sub3+NEW.Sub4+NEW.Sub5+NEW.Sub6)/6 < 40) THEN
        -- CALL NotifyHeadOfDepartment(NEW.t_no, 'Low internal assessment marks');
        SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = "Notice: Marks for Internal Assessment are
less than 40%";
  END IF;
END;
//
DELIMITER ;
```

Test:

```
CREATE TABLE Students (
   StudentID INT PRIMARY KEY,
   Name VARCHAR(100),
   DepartmentID INT
);

CREATE TABLE Subjects (
   SubjectID INT PRIMARY KEY,
   Name VARCHAR(100)
);

CREATE TABLE Enrollments (
   StudentID INT,
   SubjectID INT,
   Attendance FLOAT,
   InternalAssessmentMarks FLOAT,
   FOREIGN KEY (StudentID) REFERENCES Students(StudentID),
   FOREIGN KEY (SubjectID) REFERENCES Subjects(SubjectID)
);

CREATE TRIGGER AttendanceTrigger
```

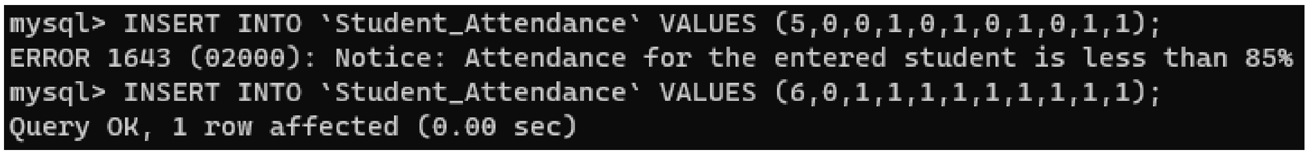
```
AFTER UPDATE ON Enrollments
FOR EACH ROW
BEGIN
  IF NEW.Attendance < 85 THEN
    CALL NotifyHeadOfDepartment(NEW.StudentID, 'Low attendance');
  END IF;
END;

CREATE TRIGGER AssessmentTrigger
AFTER UPDATE ON Enrollments
FOR EACH ROW
BEGIN
  IF NEW.InternalAssessmentMarks < 40 THEN
    CALL NotifyHeadOfDepartment(NEW.StudentID, 'Low internal assessment marks');
  END IF;
END;

DELIMITER //
CREATE PROCEDURE NotifyHeadOfDepartment(IN studentID INT, IN message VARCHAR(255))
BEGIN
  SELECT studentID, ": ", message as Output;
  -- DECLARE departmentHeadEmail VARCHAR(255);
  -- SELECT Email INTO departmentHeadEmail
  -- FROM DepartmentHeads
  -- WHERE DepartmentID = (SELECT DepartmentID FROM Students WHERE StudentID = studentID);
  -- CALL SendEmail(departmentHeadEmail, message);
END //
DELIMITER ;
```

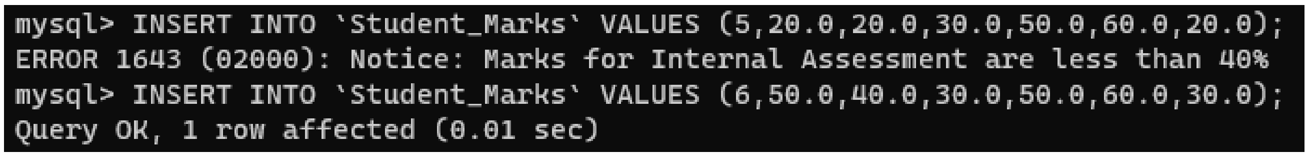
Output:

Task 1:



```
mysql> INSERT INTO `Student_Attendance` VALUES (5,0,0,1,0,1,0,1,0,1,1);
ERROR 1643 (02000): Notice: Attendance for the entered student is less than 85%
mysql> INSERT INTO `Student_Attendance` VALUES (6,0,1,1,1,1,1,1,1,1,1);
Query OK, 1 row affected (0.00 sec)
```

Task 2:



```
mysql> INSERT INTO `Student_Marks` VALUES (5,20.0,20.0,30.0,50.0,60.0,20.0);
ERROR 1643 (02000): Notice: Marks for Internal Assessment are less than 40%
mysql> INSERT INTO `Student_Marks` VALUES (6,50.0,40.0,30.0,50.0,60.0,30.0);
Query OK, 1 row affected (0.01 sec)
```

Conclusion:

- The code is structured in a modular manner using a MySQL Procedure block for better understanding.
- The code is designed for execution in interactive environments.
- Utilized BEGIN sections to define variables and execute trigger logic.
- Created "NotifyHeadOfDepartment" and "SendEmail" Procedure to notify and send mail to the respected authority.
- Applied the DBMS_OUTPUT.PUT_LINE function for displaying output for all the procedures.
  .