MySQL Labs

**MySQL (Day3):**

**insert into students\_courses**

**values**

**(1,4,60,NULL),**

**(2,1,NULL,NULL),**

**(2,4,75,NULL),**

**(3,1,NULL,NULL),**

**(3,2,NULL,NULL),**

**(3,3,75,NULL);**

|  |  |
| --- | --- |
| *1* | ***Create function to calculate the number of students who get grade less than 80 in a certain exam (course id will be sent as a parameter)*** |
|  | **DELIMITER $**  **CREATE FUNCTION calc\_num\_students\_lt\_80(courseId integer)**  **RETURNS integer**  **BEGIN**  **DECLARE num integer;**  **SET num = (SELECT COUNT(student\_id) FROM students\_courses WHERE grade < 80 AND course\_id = courseId);**  **RETURN num;**  **END$**  **DELIMITER ;** |
| *2* | ***Create stored procedure to display the names of the absence students of a certain courses.(Absent means has no grades)*** |
|  | **DELIMITER $**  **CREATE PROCEDURE display\_absent\_students(courseId integer)**  **BEGIN**  **SELECT CONCAT(first\_name, " ", last\_name) AS Absent\_Students FROM students\_courses INNER JOIN students ON students.student\_id = students\_courses.student\_id WHERE grade IS NULL AND course\_id = courseId GROUP BY students\_courses.student\_id;**  **END$**  **DELIMITER ;** |
| *3* | ***Create stored procedure to calculate the average grades for certain course.*** |
|  | **DELIMITER $**  **CREATE PROCEDURE display\_course\_average(courseId integer)**  **BEGIN**  **SELECT IFNULL(AVG(grade), 0) AS Average\_Grade FROM students\_courses WHERE course\_id = courseId;**  **END$**  **DELIMITER ;** |
| *4* | ***Create trigger to keep track the changes(updates) of the grades in the studnets\_courses table***  ***( create changes table with the following fields:***  ***id int primary key ,***  ***user varchar(30),***  ***action varchar(40),***  ***old\_grade int,***  ***new\_grade int,***  ***change\_date date).***  ***Test the trigger by updating grade int the “Students\_courses” table***  ***Confirm that the row is added in the” change\_table”*** |
|  | **CREATE TABLE changes (**  **id INT PRIMARY KEY AUTO\_INCREMENT,**  **user VARCHAR(30),**  **action VARCHAR(40),**  **old\_grade INT,**  **new\_grade INT,**  **change\_date DATE**  **);**  **CREATE TRIGGER track\_grades\_changes**  **AFTER UPDATE on students\_courses**  **FOR EACH ROW**  **INSERT INTO changes(user, action, old\_grade, new\_grade, change\_date) VALUES(CURRENT\_USER, “update”, Old.grade, New.grade, NOW());** |
| *5* | ***Create event to delete the changes tables every 5 minute*** |
|  | **SET @@global.event\_scheduler=1;**  **CREATE EVENT delete\_changes**  **ON SCHEDULE EVERY 5 MINUTE**  **DO**  **DELETE FROM changes;** |