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)*** |
|  | DROP FUNCTION IF EXISTS StudentGradeLess80;  -> DELIMITER $  -> CREATE FUNCTION StudentGradeLess80(cour\_id integer)  -> RETURNS INT  -> BEGIN  -> RETURN (SELECT COUNT(\*)  -> FROM students\_courses  -> WHERE course\_id = cour\_id and grade < 80);  -> END$  -> DELIMITER ; |
| *2* | ***Create stored procedure to display the names of the absence students of a certain courses.(Absent means has no grades)*** |
|  | DROP FUNCTION IF EXISTS DisplayAbsent;  -> DELIMITER $  -> CREATE PROCEDURE DisplayAbsent(p\_couseID integer)  -> BEGIN  -> SELECT CONCAT(first\_name , ' ' , last\_name) as student\_name, grade  -> FROM students s LEFT JOIN students\_courses sc  -> ON s.student\_id = sc.student\_id  -> WHERE sc.course\_id = p\_couseID and sc.grade IS NULL;  -> END$  -> DELIMITER ; |
| *3* | ***Create stored procedure to calculate the average grades for certain course.*** |
|  | DROP FUNCTION IF EXISTS courseAVG  -> DELIMITER $  -> CREATE PROCEDURE courseAVG(p\_couseID integer)  -> BEGIN  -> SELECT AVG(grade)  -> FROM students\_courses  -> WHERE course\_id = p\_couseID;  -> 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\_table(  -> id int primary key ,  -> user varchar(30),  -> action varchar(40),  -> old\_grade int,  -> new\_grade int,  -> change\_date date);  DELIMITER $  CREATE TRIGGER updateChanges  -> AFTER UPDATE ON students\_courses  -> FOR EACH ROW  -> BEGIN  -> INSERT INTO changes\_table  -> VALUES (old.student\_id, CURRENT\_USER(), "update", old.grade, NEW.grade, CURRENT\_DATE());  -> END$  -> DELIMITER ;  UPDATE students\_courses set grade = 100  -> WHERE student\_id = 1 and course\_id = 1; |
| *5* | ***Create event to delete the changes tables every 5 minute*** |
|  | CREATE EVENT delete\_changes  -> ON SCHEDULE EVERY 5 MINUTE  -> DO  -> DELETE FROM changes\_table; |