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 $***  ***CREATE OR REPLACE FUNCTION count\_students\_under\_80(p\_course\_id int(3))***  ***RETURNS int(5)***  ***BEGIN***  ***DECLARE v\_count int(5);***  ***SET v\_count = (select count(\*)***  ***from students\_courses***  ***where grade < 80 and course\_id = p\_course\_id);***  ***RETURN v\_count;***  ***END$***  ***DELIMITER ;*** |
| *2* | ***Create stored procedure to display the names of the absence students of a certain courses.(Absent means has no grades)*** |
|  | ***delimiter $***  ***create or replace procedure absent\_students(p\_course\_id int(3))***  ***BEGIN***  ***select \* from students***  ***where student\_id in (***  ***select student\_id from students\_courses***  ***where grade is null and course\_id = p\_course\_id***  ***);***  ***end$***  ***delimiter ;*** |
| *3* | ***Create stored procedure to calculate the average grades for certain course.*** |
|  | ***delimiter $***  ***create or replace procedure course\_avg(p\_course\_id int(3))***  ***BEGIN***  ***select floor(avg(grade))***  ***from students\_courses***  ***where course\_id = p\_course\_id;***  ***end$***  ***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 ,**  **user varchar(30),**  **action varchar(40),**  **old\_grade int,**  **new\_grade int,**  **change\_date date**  **);**  **CREATE TRIGGER students\_courses\_changes BEFORE UPDATE ON students\_courses**  **FOR EACH ROW**  **INSERT INTO changes**  **(id, user, action, old\_grade, new\_grade, change\_date)**  **VALUES**  **(old.student\_id, current\_user(), "update student\_course", old.grade, new.grade, current\_date());**  **UPDATE students\_courses**  **SET grade = 95**  **WHERE student\_id = 2 and course\_id = 1;**  **Select \* from changes;**  ***//? on reupdate same student -> error wether before or after update*** |
| *5* | ***Create event to delete the changes tables every 5 minute*** |
|  | **CREATE EVENT delete\_changes**  **ON SCHEDULE EVERY 5 MINUTE**  **DO**  **DELETE FROM changes;** |