MySQL Labs

MySQL (Day3):
Mariam khaled Saad Ibrahim (open source)

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 count_student;
	delimiter $
	CREATE FUNCTION count_student(courseId integer)
	RETURNS int(11)
	BEGIN
	DECLARE count int(11);
	set count= (select count(*) from students_courses where course_id=courseId and grade<80 group by course_id);
	RETURN count;
	END$
	delimiter ;**

2	***Create stored procedure to display the names of the absence students of a certain courses.(Absent means has no grades)***
	drop procedure if exists absence_students; delimiter $ CREATE procedure absence_students (courseId integer) BEGIN select first_name as absence_students from students s,students_courses sc where s.student_id=sc.student_id and sc.grade is null and sc.course_id=courseId; END$ delimiter ;
3	***Create stored procedure to calculate the average grades for certain course.***

	**drop procedure if exists avrage_grade_for_course;
	delimiter $
	CREATE procedure avrage_grade_for_course (courseId integer)
	BEGIN
	select avg(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);

delimiter $
 DROP TRIGGER if exists change_grades;
CREATE TRIGGER change_grades
AFTER update ON students_courses
for EACH ROW
begin
if (new.grade != old.grade)
then
INSERT INTO changes(user,action,old_grade,new_grade,change_date)
VALUES (current_user(),"update",OLD.grade,new.grade,current_time());
end if;
end;$
delimiter ;
```

5 *Create event to delete the changes tables every 5 minute*

```
select @@global.event_scheduler;
set @@global.event_scheduler=1;
CREATE EVENT delete_changes ON SCHEDULE EVERY 5 minute DO
DELETE FROM changes;
```