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 exam\_grades(cid int)*  *returns int(11)*  *begin*  *declare num INT;*  *select count(student\_id) into num*  *from students\_courses*  *where grade < 80 && course\_id = cid;*  *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\_names(cid int)*  *begin*  *select s.first\_name, s.last\_name*  *from students s, students\_courses sc*  *where s.student\_id = sc.student\_id*  *&& sc.course\_id = cid && sc.grade is null;*  *end $*  *delimiter ;* |
| *3* | ***Create stored procedure to calculate the average grades for certain course.*** |
|  | *delimiter $*  *create procedure avg\_grades(cid int)*  *begin*  *select avg(grade)*  *from students\_courses*  *where course\_id = cid;*  *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 grades\_update**  **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,CURRENT\_TIMESTAMP());** |
| *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; |