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 $*  *create function exam\_emp\_count(p\_course\_id int)*  *returns int*  *deterministic*  *begin*  *declare student\_count int;*  *set student\_count = (*  *select count(student\_id)*  *from students\_courses*  *where course\_id = p\_course\_id*  *and grade < 80*  *);*  *return student\_count;*  *end$*  *delimiter ;*  *or*  *delimiter $*  *create function exam\_emp\_count(p\_course\_id int)*  *returns int*  *deterministic*  *begin*  *return(*  *select count(student\_id)*  *from students\_courses*  *where course\_id = p\_course\_id*  *and grade < 80*  *);*  *end$*  *delimiter ;* |
| *2* | ***Create stored procedure to display the names of the absence students of a certain courses.(Absent means has no grades)*** |
|  | *delimiter $*  *create procedure absent\_students(p\_course\_id int)*  *begin*  *select concat(first\_name, ' ', last\_name) as student\_name*  *from students st, students\_courses sc*  *where st.student\_id = sc.student\_id and isnull(grade) and sc.course\_id = p\_course\_id;*  *end$*  *delimiter ;*  *call absent\_students(1);* |
| *3* | ***Create stored procedure to calculate the average grades for certain course.*** |
|  | *delimiter $*  *create procedure average\_grades(p\_course\_id int)*  *begin*  *select avg(grade) as average\_grades*  *from students\_courses*  *where course\_id = p\_course\_id;*  *end$*  *delimiter ;*  *call average\_grades(1);* |
| *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);*  *create trigger track\_changes*  *after update*  *on students\_courses*  *for each row*  *insert into changes\_table values(OLD.course\_id, USER(), 'Grade Changed' ,OLD.grade, NEW.grade, current\_date());* |
| *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;* |