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 num\_student\_less80(c\_id int)*  *returns int(100)*  *begin*  *return (select count(sc.student\_id) from students\_courses sc*  *where c\_id=sc.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(c\_name varchar(30))*  *begin*  *Select concat\_ws(first\_name , " " ,last\_name ) as full\_name, c.course\_name, IFNULL(e.grade, "Absent")*  *From (courses c, students s) left join (students\_courses e)*  *on e.course\_id = c.course\_id*  *AND e.student\_id = s.student\_id and grade=null*  *having course\_name=c\_name;*  *end #*  *delimiter ;*  *call absent\_students('Network');* |
| *3* | ***Create stored procedure to calculate the average grades for certain course.*** |
|  | *delimiter #*  *create procedure average\_grade()*  *begin*  *select avg(grade),course\_name*  *from students\_courses sc,courses c*  *where sc.course\_id=c.course\_id*  *group by course\_name;*  *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 trigger grade\_update after update on students\_courses**  **for each row**  **insert into change\_table values(1,CURRENT\_USER(),'update',old.grade,new.grade,now());**   * **-- For test:**   **update students\_courses set grade=50**  **where student\_id=1 and course\_id=1;** |
| *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 change\_table;** |