**MySQL Labs**

**MySQL (Day2):**

|  |  |
| --- | --- |
| ***1*** | ***Update students courses table, set the registration date value to “Today”;*** |
|  | ***update students\_courses set reg\_date=current\_date()***  ***where student\_id in (1,2,3);*** |
| ***2*** | ***Display the registration date in the following format:***  ***Day, month/ year*** |
|  | ***select DATE\_FORMAT(reg\_date,'%d,%m/%Y') from students\_courses;*** |
| ***3*** | ***Display the full name (first, last) of the student with his grade.***  ***if his garde is greater than 85% Excellent, from 75% to 85% Very good, from 65% to 75% Good and from 55% to 65% pass otherwise will be graded as failed.*** |
|  | **SELECT concat\_ws(" ", first\_name, last\_name) as name, grade,**  **CASE**  **WHEN grade> 85 THEN "excellent"**  **WHEN grade <85 and grade >75 THEN "very good"**  **WHEN grade <75 and grade >65 THEN " good"**  **WHEN grade <65 and grade >55 THEN "pass"**  **ELSE "fail"**  **END as eval**  **From students\_courses as c, students as s**  **Where s.student\_id = c. student\_id;** |
| ***4*** | ***Display the capitalized last name , and the grade , if he has no grade display the keyword absent. [using ifNULL function]*** |
|  | ***Select ucase(s.last\_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***  ***Order by s.last\_name;*** |
| ***5*** | ***Display students' names, course name along with their grades.*** |
|  | ***select concat\_ws(" ",first\_name,last\_name) as student\_name,course\_name,grade***  ***from students s,students\_courses sc,courses c***  ***where s.student\_id=sc.student\_id and c.course\_id=sc.course\_id;*** |
| ***6*** | ***For each course, display the course name, min grade, max grade, average grade, number of attended students.*** |
|  | Select c.course\_name, min(e.grade), max(e.grade), avg(e.grade), count(e.student\_id) From courses c, students\_courses e Where e.course\_id = c.course\_id Group by c.course\_name; |
| ***7*** | ***Use subquery to display the names of the students who were born before student no 1.*** |
|  | Select concat(s.first\_name,” “,s.last\_name)as student\_nameFrom students sWhere birth\_date < (select birth\_date From students where student\_id =1); |
| ***8*** | ***Use subquery to display the data of all the courses with a credit hour similar to MySQL's credit hours*** |
|  | Select \* From coursesWhere credit\_hour < (select credit\_hour from courses wherecourse\_name like (“MySQL”)); |
| ***10*** | ***Create a view called female\_students\_vu to display all the female students*** |
|  | CREATE VIEW female\_students\_vuASSELECT \* FROM students where gender = “female”; |
| ***11*** | ***Try to insert a male student through your view*** |
|  | Insert into female\_students\_vu  Values(6, "Ali", "Kamal", NULL, NULL, "male", "1991-09-01"); |
| ***12*** | ***Select all the data from your view and then from the students table*** |
|  | ***Select \* from female\_students\_vu***  ***Select \* from students*** |
| ***13*** | ***Prevent the ability to insert another male student through you view*** |
|  | alter view female\_students\_vuASSELECT \* FROM students where gender = “female”WITH CHECK OPTION; |
| ***14*** | ***Use the information schema to display the table name , schema and the updatability of the female\_students\_vu view*** |
|  | USE INFORMATION\_SCHEMA;SELECT \* FROM INFORMATION\_SCHEMA.VIEWSWHERE TABLE\_NAME = “female\_students\_vu”; |
| ***15*** | ***Use the information schema to display the create time, table\_rows, auto\_increment, and the comments on the students table.*** |
|  | ***SELECT TABLE\_ROWS,TABLE\_COMMENT,CREATE\_TIME,AUTO\_INCREMENT FROM***  ***INFORMATION\_SCHEMA.Tables WHERE TABLE\_NAME = "students";*** |
| ***16*** | ***Create a nonunique index on the foreign key column (COURSE\_ID) in the students\_courses table.*** |
|  | ALTER TABLE students\_courses ADD INDEX (COURSE\_ID); |
| ***17*** | ***Create a user with your name and give him the privilege to access the grades database*** |
|  | CREATE USER “OS”@”localhost”IDENTIFIED BY “os123”;GRANT ALL PRIVILEGES ON \*.\*TO “open\_source”@”localhost”WITH GRANT OPTION; |
| ***18*** | ***Connect to mysql using the user you created and try to insert one record in the courses table.*** |
|  | ***S****ELECT CURRENT\_USER();*  *OS@localhost* |
| ***19*** | ***Change your password.*** |
|  | SET PASSWORD FOR“OS”@”localhost” = “123os”; |
| ***20*** | ***Show your privileges.*** |
|  | SHOW GRANTS FOR CURRENT\_USER(); |