**MySQL Labs**

**MySQL (Day2): Kareem Ibraheem**

|  |  |
| --- | --- |
| ***1*** | ***Update students courses table, set the registration date value to “Today”;*** |
|  | **update students\_courses set reg\_date = 'today';** |
| ***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(first\_name, ' ', last\_name), (CASE WHEN grade > 85 THEN 'Excellent' WHEN grade > 75 THEN 'Very Good' WHEN grade > 65 THEN 'Good' WHEN grade > 55 THEN 'Pass' ELSE 'Failed' END) from students s, students\_courses**  **g where s.student\_id = g.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(last\_name), ifnull(grade, 'Absent') from students s, students\_courses g where s.student\_id = g.student\_id;** |
| ***5*** | ***Display students' names, course name along with their grades.*** |
|  | **select first\_name, course\_name, grade from students s, courses c, students\_courses g where s.student\_id**  **= g.student\_id and c.course\_id = g.course\_id;** |
| ***6*** | ***For each course, display the course name, min grade, max grade, average grade, number of attended students.*** |
|  | **select course\_name, min(grade), max(grade), avg(grade), count(distinct(student\_id)) from courses c, stu**  **dents\_courses g where c.course\_id = g.course\_id group by g.course\_id;** |
| ***7*** | ***Use subquery to display the names of the students who were born before student no 1.*** |
|  | **select first\_name from students where birth\_date > any(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 courses where credit\_hour = (select credit\_hour from courses where course\_name = 'MySQL');** |
| ***10*** | ***Create a view called female\_students\_vu to display all the female students*** |
|  | **create view female\_students\_vu as select \* from students where gender = 'female';** |
| ***11*** | ***Try to insert a male student through your view*** |
|  | **insert into female\_students\_vu values(6, 'mohamed', 'salah', NULL, NULL, 'male', '2001-11-11');** |
| ***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*** |
|  | **create or replace view female\_students\_vu as select \* from students with check option;** |
| ***14*** | ***Use the information schema to display the table name , schema and the updatability of the female\_students\_vu view*** |
|  | **select table\_name, table\_schema, is\_updatable from views where 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 create\_time, table\_rows, auto\_increment, table\_comment from tables where table\_name = 'students';** |
| ***16*** | ***Create a nonunique index on the foreign key column (COURSE\_ID) in the students\_courses table.*** |
|  | **create index ID\_Index on students\_courses(course\_id);** |