**MySQL Labs**

**MySQL (Day2):**

|  |  |
| --- | --- |
| ***1*** | ***Update students courses table, set the registration date value to “Today”;*** |
|  | **UPDATE students\_courses**  **Set reg\_date = “2022-01-04”;**  **Or**  **UPDATE students\_courses**  **Set reg\_date = CURRENT\_DATE();** |
| ***2*** | ***Display the registration date in the following format:***  ***Day, month/ year*** |
|  | **Select dayofmonth(reg\_date), month(reg\_date), year(reg\_date)**  **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(s.first\_name,” “,s.last\_name)as student\_name, c.course\_name, e.grade, case**  **when e.grade >= 85 then “Excellent”**  **when e.grade < 85 and e.grade >= 75 then “Very good”**  **when e.grade < 75 and e.grade >= 65 then “*Good*”**  **when e.grade < 65 and e.grade >= 55 then “Pass”**  **Else “*failed*”**  **End as grade\_**  **From courses c, students\_courses e, students s**  **Where e.course\_id = c.course\_id**  **AND e.student\_id = s.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(s.first\_name,” “,s.last\_name)as student\_name, c.course\_name, e.grade**  **From courses c, students\_courses e, students s**  **Where e.course\_id = c.course\_id**  **AND e.student\_id = s.student\_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\_name**  **From students s**  **Where 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 courses**  **Where credit\_hour < (select credit\_hour from courses where course\_name like (“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, “Mahmoud”, “Kamal”, NULL, NULL, “male”, “1991-09-01”);*** |
| ***12*** | ***Select all the data from your view and then from the students table*** |
|  |  |
| ***13*** | ***Prevent the ability to insert another male student through you view*** |
|  | **alter view *female\_students\_vu***  ***AS***  ***SELECT \* FROM students where gender = “female”***  ***WITH CHECK OPTION;***  ***.***  ***.***  ***.***  ***Insert into female\_students\_vu***  ***Values(7, “Mahmoud”, “Kamal”, NULL, NULL, “male”, “1991-09-01”);*** |
| ***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.VIEWS***  ***WHERE TABLE\_NAME = “female\_students\_vu”***  ***AND TABLE\_SCHEMA = “OS42”\G***  ***Or***  ***SELECT TABLE\_NAME,TABLE\_SCHEMA,VIEW\_DEFINITION FROM INFORMATION\_SCHEMA.VIEWS WHERE TABLE\_SCHEMA = 'OS42' AND 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\_SCHEMA = 'OS42' AND 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);***    ***SHOW INDEX from students\_courses;*** |
| ***17*** | ***Create a user with your name and give him the privilege to access the grades database*** |
|  | ***CREATE USER “open\_source”@”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.*** |
|  | ***“open\_source”@”localhost”***  ***“root”@”localhost”*** |
| ***19*** | ***Change your password.*** |
|  | ***SET PASSWORD FOR***  ***“open\_source”@”localhost” = “iti”;*** |
| ***20*** | ***Show your privileges.*** |
|  | ***SHOW GRANTS FOR CURRENT\_USER();***  ***SHOW GRANTS FOR “open\_source”@”localhost”;*** |