**MySQL Labs**

**MySQL (Day2):**

|  |  |
| --- | --- |
| ***1*** | ***Update students courses table, set the registration date value to “Today”;*** |
|  | UPDATE students\_courses SET reg\_date = NOW(); |
| ***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) AS Full\_Name, grade, 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 AS Grade\_Text FROM students\_courses JOIN students ON students.student\_id = students\_courses.student\_id; |
| ***4*** | ***Display the capitalized last name , and the grade , if he has no grade display the keyword absent. [using ifNULL function]*** |
|  | SELECT CONCAT(UCASE(LEFT(last\_name, 1)), SUBSTRING(last\_name, 2)) AS Last\_Name, IFNULL(grade, "Absent") AS Grade FROM students\_courses RIGHT JOIN students ON students.student\_id = students\_courses.student\_id; |
| ***5*** | ***Display students' names, course name along with their grades.*** |
|  | SELECT CONCAT(first\_name, ' ', last\_name) AS Full\_Name, grade, course\_name FROM students\_courses JOIN students ON students.student\_id = students\_courses.student\_id JOIN courses ON courses.course\_id = students\_courses.course\_id; |
| ***6*** | ***For each course, display the course name, min grade, max grade, average grade, number of attended students.*** |
|  | SELECT Course\_name, Students\_count, Min\_grade, Max\_Grade, Average\_Grade FROM (SELECT COUNT(student\_id) AS students\_count, course\_id FROM students\_courses GROUP BY course\_id) Q1 JOIN courses ON courses.course\_id = Q1.course\_id JOIN (SELECT course\_id, MAX(grade) AS Max\_Grade, AVG(grade) AS Average\_Grade, MIN(grade) AS Min\_Grade FROM students\_courses GROUP BY course\_id) Q2 ON Q2.course\_id = Q1.course\_id ; |
| ***7*** | ***Use subquery to display the names of the students who were born before student no 1.*** |
|  | SELECT CONCAT(first\_name, ' ', last\_name) AS Full\_Name FROM students 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*** |
|  | // including Mysql  SELECT \* FROM courses WHERE credit\_hour = (SELECT credit\_hour FROM courses WHERE course\_name = 'mysql');  // excluding Mysql  SELECT \* FROM courses, (SELECT course\_id, credit\_hour FROM courses WHERE course\_name = 'mysql') Q WHERE courses.course\_id != Q.course\_id AND courses.credit\_hour = Q.credit\_hour; |
| ***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 (first\_name, last\_name, gender, birth\_date) VALUES ('Ahmed', 'Magdy', 'male', '1996-03-21'); |
| ***12*** | ***Select all the data from your view and then from the students table*** |
|  | SELECT \* FROM female\_students\_vu;  SELECT \* FROM students;  SELECT \* FROM students WHERE gender = 'Female'; |
| ***13*** | ***Prevent the ability to insert another male student through you view*** |
|  | CREATE OR REPLACE VIEW female\_students\_vu AS SELECT \* 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*** |
|  | SELECT Table\_schema, Table\_name, 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' AND table\_schema = 'grades'; |
| ***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\_index(course\_id); |
| ***17*** | ***Create a user with your name and give him the privilege to access the grades database*** |
|  | CREATE USER 'ahmed'@'localhost' IDENTIFIED BY '123456';  GRANT ALL PRIVILIGES ON grades.\* TO 'ahmed'; |
| ***18*** | ***Connect to mysql using the user you created and try to insert one record in the courses table.*** |
|  | INSERT INTO courses(course\_name, credit\_hour) VALUES('C++', 3); |
| ***19*** | ***Change your password.*** |
|  | ALTER USER 'ahmed'@'localhost' IDENTIFIED BY '123123';  FLUSH PRIVILEGES; |
| ***20*** | ***Show your privileges.*** |
|  | SHOW GRANTS FOR 'ahmed';  SHOW GRANTS; |