**MySQL Labs**

**MySQL (Day2):**

|  |  |
| --- | --- |
| ***1*** | ***Update students courses table, set the registration date value to “Today”;*** |
|  | **UPDATE students\_courses SET reg\_date = current\_date;** |
| ***2*** | ***Display the registration date in the following format:***  ***Day, month/ year*** |
|  | **SELECT DATE\_FORMAT(reg\_date, "%d , %M / %Y") as Date\_Modified 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**  **FROM students S INNER JOIN students\_courses C**  **ON 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 if NULL function]*** |
|  | **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**  **FROM students S INNER JOIN students\_courses C**  **ON S.student\_id = C.student\_id** |
| ***5*** | ***Display students' names, course name along with their grades.*** |
|  | **SELECT CONCAT(first\_name, ' ', last\_name) AS full\_name, course\_name, grade**  **FROM (**  **(students S INNER JOIN students\_courses SC ON S.student\_id = SC.student\_id)**  **INNER JOIN courses C ON SC.course\_id=C.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(grade)**  **FROM courses C INNER JOIN students\_courses SC**  **ON C.course\_id = SC.course\_id**  **GROUP BY course\_name;** |
| ***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*** |
|  | **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 ('10', 'Hossam', 'Adel', 10, 'H@gmail.com', 'male', '1995-8-25');**  ***\*\*Student was added to the table but cannot be seen in the view.*** |
| ***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\_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*** |
|  | **USE information\_schema;**  **SELECT TABLE\_NAME, TABLE\_SCHEMA, IS\_UPDATABLE**  **FROM VIEW**  **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 C\_ID**  **ON students\_courses(course\_id);** |
| ***17*** | ***Create a user with your name and give him the privilege to access the grades database*** |
|  | **CREATE USER Hossam IDENTIFIED BY '12345678';**  **GRANT ALL ON grades TO Hossam;** |
| ***18*** | ***Connect to mysql using the user you created and try to insert one record in the courses table.*** |
|  | **mysql -u Hossam -p**  **12345678**  **INSERT INTO courses**  **VALUES (10, 'NEW\_COURSE', 13);** |
| ***19*** | ***Change your password.*** |
|  | **ALTER USER 'Hossam' IDENTIFIED by '87654321';** |
| ***20*** | ***Show your privileges.*** |
|  | **SHOW GRANTS;** |