MySQL Labs

MySQL (Day2):

1	***Update students courses table, set the registration date value to "Today";***
	update students_courses set reg_date = '2024-1-23';
2	***Display the registration date in the following format:*** ***Day, month/ year***
	select date_format(reg_date,'%d, %M/%y')As formated_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(first_name ,' ',last_name) as full_name ,Case When grade >85 then 'excellent' When grade between 75 and 85 then 'very good' When grade between 65 and 75 then 'good' When grade between 55 and 65 then 'pass' Else 'failed' End as grade_status From students as s , students_courses as c 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 UPPER(last_name)as capitalized_last_name,IFNULL(CAST(grade as CHAR),'absent')as grade_or_absent From students as s left join students_courses as c on s.student_id = c.student_id;
5	***Display students' names, course name along with their grades.***
	SELECT s.first_name AS 'Student First Name', s.last_name AS 'Student Last Name', c.course_name AS 'Course Name',sc.grade AS 'Grade'FROM students s ***JOIN students_courses sc ON s.student_id = sc.student_id*** ***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 c.course_name, min(sc.grade)as min_grade, max(sc.grade)as max_grade, avg(sc.grade)as avg_grade,count(sc.grade)as num_attend_students from courses c left join students_courses sc on c.course_id=sc.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 first_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 course_id, course_name,credit_hour ***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 ***student_id, first_name,last_name, tel,email,gender,birth_date*** ***FROM students*** ***WHERE gender = 'female';*** ***SELECT * FROM female_students_vu;***
11	***Try to insert a male student through your view***
	INSERT INTO female_students_vu (first_name, last_name, tel, email, gender, birth_date) ***VALUES ('John', 'Doe', '123456789', 'john.doe@example.com', 'male', '1995-05-15');***
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 ***student_id, first_name,last_name, tel,email,gender,birth_date*** ***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_NAME AS 'Table Name',TABLE_SCHEMA AS 'Schema', ***IS_UPDATABLE AS 'Updatability',*** ***FROM information_schema.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 TABLE_NAME AS 'Table Name', CREATE_TIME AS 'Create Time',TABLE_ROWS AS 'Table Rows', AUTO_INCREMENT AS 'Auto Increment',TABLE_COMMENT AS 'Table Comment' ***FROM information_schema.TABLES*** ***WHERE TABLE_SCHEMA = 'grades'AND TABLE_NAME = 'students';***
16	***Create a nonunique index on the foreign key column (COURSE_ID) in the students_courses table.***
	CREATE INDEX idx_course_id ON students_courses (course_id);