MySQL Labs

MySQL (Day2):

Mariam khaled Saad Ibrahim   (open source)

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" ) 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),s.student_id,grade ,`
	`case`
	`when grade >=85 then "Excellent"`
	`when grade < 85 and grade >=75 then "Very good"`
	`when grade <75 and grade >= 65 then "good"`
	`when grade <65 and grade>=55 then "pass"`
	`else "failed"`
	`end as student_grade`
	`from students s,students_courses sc where s.student_id=sc.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),ifnull(grade,"absent")  from students s  left JOIN` `students_courses sc on s.student_id = sc.student_id;`
5	**Display _students' names_, _course name_ along with their grades.**

	`select concat(first_name,last_name),course_name,grade from courses c,students s`
	`,students_courses sc`
	`where s.student_id=sc.student_id and c.course_id=sc.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(student_id)`
	`from courses c , students_courses sc where  c.course_id = sc.course_id  group by`
	`course_name ;`

7	**Use subquery to display the <u>names of the students</u> who were born before student no 1.**	
	select concat(first_name,last_name) from students where birth_date < (select birth_date from students where student_id=1);	
8	**Use subquery to display the <u>data of all the courses</u> 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 like "female";	
11	**Try to <u>insert a male student</u> through your view**	
	insert into female_students_vu(student_id,first_name,last_name,tel,email,gender,birth_date) values(20 , "mohamed" , "zain", "01118726857","mohamed@gmail.com", "male" , "1999-09-02");	
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 like "female"WITH CHECK OPTION;	
14	**Use the information schema to display the <u>table name</u> , <u>schema</u> and the <u>updatability</u> 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 <u>create time</u>, <u>table_rows</u>, <u>auto_increment</u>, and the <u>comments</u> on the students table.**	
	select create_time , table_rows , auto_increment , table_comment from information_schema.tables where table_name = "students";	
16	**Create a nonunique index on the foreign key column (COURSE_ID) in the students_courses table.**	
	create index courses_id_index on students_courses (course_id);	
17	**Create a user with your name and give him the privilege to access the grades database**	
	create user "mariam" IDENTIFIED BY "111115"; GRANT ALL ON grades.* TO 'mariam';	

18	*Connect to mysql using the user you created and try to insert one record in the courses table.*
	insert into courses(course_id,course_name,credit_hour)values(5,"reactjs",50);
19	*Change your password.*
	alter USER 'mariam' identified by '12345';
20	*Show your privileges.*
	SHOW GRANTS FOR 'mariam';