MySQL Labs

MySQL (Day3):

insert into students_courses
values
(1,4,60,NULL),
(2,1,NULL,NULL),
(2,4,75,NULL),
(3,1,NULL,NULL),
(3,2,NULL,NULL),
(3,3,75,NULL);

1	***Create function to calculate the number of students who get grade less than 80 in a certain exam (course id will be sent as a parameter)***
	DELIMITER $ CREATE FUNCTION calc_no_students(ID INT) -> RETURNS INT -> DETERMINISTIC -> BEGIN -> DECLARE count INT DEFAULT 0; -> SELECT COUNT(*) INTO count -> FROM students_courses -> WHERE course_id = ID -> AND grade < 80; -> RETURN count; -> END$ DELIMITER ; SELECT calc_no_students(4); +----------------------+ | calc_no_students(4) | +----------------------+ | 3 | +----------------------+
2	***Create stored procedure to display the names of the absence students of a certain courses.(Absent means has no grades)***
	DELIMITER $ *CREATE PROCEDURE names_of_abs()* *-> BEGIN* *-> SELECT CONCAT(first_name, ' ', last_name) AS Name, grade* *-> FROM students JOIN students_courses* *-> ON students.student_id = students_courses.student_id* *-> WHERE grade IS NULL;* *-> END$* *DELIMITER ;*

CALL names_of_abs()

Name	grade
Ahmed Ibrahim	NULL
Ahmed Ossama	NULL
Ahmed Ossama	NULL

*SELECT * FROM students join students_courses on students.student_id = students_courses.student_id;*

student_id	first_name	last_name	tel	email	gender	birth_date	student_id	course_id	grade	reg_d
1	Ahmed	Aly	NULL	NULL	male	1991-10-01	1	1	80	2024-
1	Ahmed	Aly	NULL	NULL	male	1991-10-01	1	2	90	2024-
1	Ahmed	Aly	NULL	NULL	male	1991-10-01	1	3	100	2024-
1	Ahmed	Aly	NULL	NULL	male	1991-10-01	1	4	60	NULL
2	Ahmed	Ibrahim	NULL	NULL	male	1991-09-01	2	1	NULL	NULL
2	Ahmed	Ibrahim	NULL	NULL	male	1991-09-01	2	2	99	2024-
2	Ahmed	Ibrahim	NULL	NULL	male	1991-09-01	2	3	80	2024-
2	Ahmed	Ibrahim	NULL	NULL	male	1991-09-01	2	4	75	NULL
3	Ahmed	Ossama	NULL	NULL	male	1992-10-01	3	1	NULL	NULL
3	Ahmed	Ossama	NULL	NULL	male	1992-10-01	3	2	NULL	NULL
3	Ahmed	Ossama	NULL	NULL	male	1992-10-01	3	3	75	NULL
3	Ahmed	Ossama	NULL	NULL	male	1992-10-01	3	4	70	2024-

3 ***Create stored procedure to calculate the average grades for certain course.***

DELIMITER $
CREATE PROCEDURE calc_avg_grades(c_id INT)
-> BEGIN
-> SELECT
-> AVG(grade) AS avg_grade
-> FROM students_courses
-> WHERE course_id = c_id;
-> END$

DELIMITER ;

```
MariaDB [os]> CALL calc_avg_grades(1);
+-----------+
| avg_grade |
+-----------+
|   80.0000 |
```

```
MariaDB [os]> CALL calc_avg_grades(2);
+-----------+
| avg_grade |
+-----------+
|   94.5000 |
```

```
MariaDB [os]> CALL calc_avg_grades(3);
+-----------+
| avg_grade |
+-----------+
|   85.0000 |
+-----------+
```

4

Create trigger to keep track the changes(updates) of the grades in the studnets_courses table (create <u>changes table</u> with the following fields:
id int primary key ,
user varchar(30),
action varchar(40),
old_grade int,
new_grade int,
change_date date).

Test the trigger by updating grade int the "Students_courses" table

Confirm that the row is added in the" change_table"

```
CREATE TABLE changes (
    -> id INT AUTO INCREMENT PRIMARY KEY,
    -> user VARCHAR(30),
    -> action VARCHAR(40),
    -> old_grade INT,
    -> new_grade INT,
    -> change_date date);
```

```
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| id          | int(11)     | NO   | PRI | NULL    |       |
| user        | varchar(30) | YES  |     | NULL    |       |
| action      | varchar(40) | YES  |     | NULL    |       |
| old_grade   | int(11)     | YES  |     | NULL    |       |
| new_grade   | int(11)     | YES  |     | NULL    |       |
| change_date | date        | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
```

```
CREATE TRIGGER grade_update_trigger
    -> AFTER UPDATE ON students_courses
    -> FOR EACH ROW
    -> BEGIN
    -> IF OLD.grade <> NEW.grade THEN
    -> INSERT INTO changes
    -> VALUES (NULL, USER(), 'Grade Update', OLD.grade, NEW.grade,
CURDATE());
    -> END IF;
    -> END$
```

```
Query OK, 1 row affected (0.005 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
MariaDB [os]> SELECT * FROM changes;
+----+----------------+--------------+-----------+-----------+-----------
| id | user           | action       | old_grade | new_grade | change_dat
+----+----------------+--------------+-----------+-----------+-----------
|  1 | root@localhost | Grade Update |        80 |        44 | 2024-07-27
+----+----------------+--------------+-----------+-----------+-----------
```

5

	Create event to delete the changes tables every 5 minute

```
CREATE EVENT delete_changes
    -> ON SCHEDULE EVERY 5 MINUTE
    -> DO
    -> DELETE FROM changes;
```

```
MariaDB [os]> CREATE EVENT delete_changes
    -> ON SCHEDULE EVERY 5 MINUTE
    -> DO
    -> DELETE FROM changes;
Query OK, 0 rows affected, 1 warning (0.003 sec)
```