

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	<p>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)</p>
	<p><i>drop function if exists num_of_students;</i> <i>delimiter \$</i> <i>CREATE FUNCTION num_of_students(p_id integer)</i> <i>RETURNS int(11)</i> <i>BEGIN</i> <i>Declare v_count integer;</i> <i>SET v_count =</i> <i>(select count(student_id) from students_courses where course_id = p_id and grade < 80) ;</i> <i>RETURN v_count;</i> <i>END\$</i> <i>delimiter ;</i></p> <pre>mysql> CREATE FUNCTION num_of_students(p_id integer) -> RETURNS int(11) -> BEGIN -> Declare v_count integer; -> SET v_count = -> (select count(student_id) from students_courses where course_id = p_id and grade < 80) ; -> RETURN v_count; -> END\$ Query OK, 0 rows affected, 1 warning (0.03 sec) mysql> delimiter ; mysql> select num_of_students(4); +-----+ num_of_students(4) +-----+ 3 +-----+ 1 row in set (0.02 sec)</pre>
2	<p>Create stored procedure to display the names of the absence students of a certain courses.(Absent means has no grades)</p>
	<p><i>drop procedure if exists display_absents;</i> <i>delimiter \$</i> <i>CREATE PROCEDURE display_absents ()</i> <i>BEGIN</i></p>

```

Select CONCAT(s.first_name, " ", s.last_name) as Name,
c.course_name, c.course_id, IFNULL(e.grade, "Absent")
From (courses c, students s) left join (students_courses e)
ON e.course_id = c.course_id
AND e.student_id = s.student_id
AND e.grade = "Absent";

```

END\$

delimiter ;

```
mysql> CALL display_absents();
```

Name	course_name	course_id	IFNULL(e.grade, "Absent")
Ahmed Aly	Database	1	Absent
Ahmed Aly	C	2	Absent
Ahmed Aly	Network	3	Absent
Ahmed Aly	OS	4	Absent
Ahmed Aly	MySQL	5	Absent
Ahmed Aly	Java	6	Absent
Ahmed Ibrahim	Database	1	Absent
Ahmed Ibrahim	C	2	Absent
Ahmed Ibrahim	Network	3	Absent
Ahmed Ibrahim	OS	4	Absent
Ahmed Ibrahim	MySQL	5	Absent
Ahmed Ibrahim	Java	6	Absent
Ahmed Ossama	Database	1	Absent
Ahmed Ossama	C	2	Absent
Ahmed Ossama	Network	3	Absent
Ahmed Ossama	OS	4	Absent
Ahmed Ossama	MySQL	5	Absent
Ahmed Ossama	Java	6	Absent
Hoda Khaled	Database	1	Absent
Hoda Khaled	C	2	Absent
Hoda Khaled	Network	3	Absent
Hoda Khaled	OS	4	Absent
Hoda Khaled	MySQL	5	Absent
Hoda Khaled	Java	6	Absent
Mona Khalil	Database	1	Absent
Mona Khalil	C	2	Absent
Mona Khalil	Network	3	Absent
Mona Khalil	OS	4	Absent
Mona Khalil	MySQL	5	Absent
Mona Khalil	Java	6	Absent
Mahmoud Kamal	Database	1	Absent
Mahmoud Kamal	C	2	Absent
Mahmoud Kamal	Network	3	Absent
Mahmoud Kamal	OS	4	Absent
Mahmoud Kamal	MySQL	5	Absent
Mahmoud Kamal	Java	6	Absent

36 rows in set (0.01 sec)

Or with parameter:

drop procedure if exists display_absents2;

delimiter \$

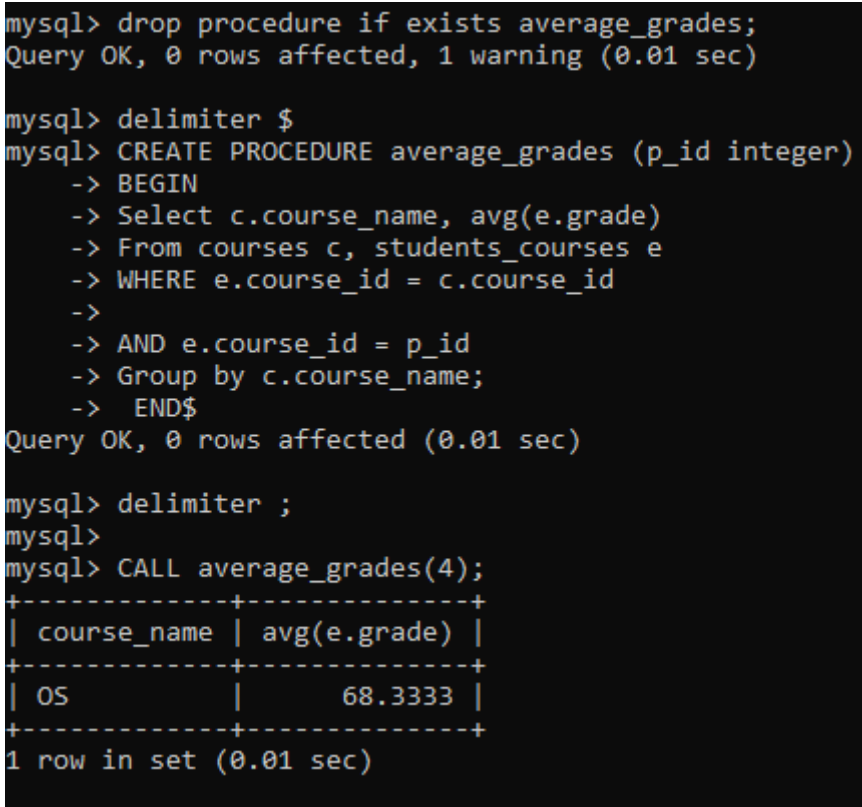
CREATE PROCEDURE display_absents2 (p_id integer)

BEGIN

```

Select CONCAT(s.first_name, " ", s.last_name) as Name,
c.course_name, c.course_id, IFNULL(e.grade, "Absent")
From (courses c, students s) left join (students_courses e)
ON e.course_id = c.course_id

```

	<pre> AND c.course_id = p_id AND e.student_id = s.student_id AND e.grade = "Absent"; END\$ delimiter ; </pre>
3	<p><i>Create stored procedure to calculate the average grades for certain course.</i></p> <pre> drop procedure if exists average_grades; delimiter \$ CREATE PROCEDURE average_grades (p_id integer) BEGIN Select c.course_name, avg(e.grade) From courses c, students_courses e WHERE e.course_id = c.course_id AND e.course_id = p_id Group by c.course_name; END\$ delimiter ; </pre>  <pre> mysql> drop procedure if exists average_grades; Query OK, 0 rows affected, 1 warning (0.01 sec) mysql> delimiter \$ mysql> CREATE PROCEDURE average_grades (p_id integer) -> BEGIN -> Select c.course_name, avg(e.grade) -> From courses c, students_courses e -> WHERE e.course_id = c.course_id -> -> AND e.course_id = p_id -> Group by c.course_name; -> END\$ Query OK, 0 rows affected (0.01 sec) mysql> delimiter ; mysql> mysql> CALL average_grades(4); +-----+-----+ course_name avg(e.grade) +-----+-----+ OS 68.3333 +-----+-----+ 1 row in set (0.01 sec) </pre>
4	<p><i>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).</i></p>

	<p><i>Test the trigger by updating grade int the “Students_courses” table</i></p> <p><i>Confirm that the row is added in the” change_table”</i></p>
	<pre>CREATE TABLE IF NOT EXISTS <i>changes</i>(id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY, user VARCHAR(30), action VARCHAR(40), old_grade int, new_grade int, change_date date) ENGINE = INNODB; DELIMITER // CREATE TRIGGER grades_change AFTER UPDATE ON students_courses FOR EACH ROW BEGIN IF !(NEW.grade <=> OLD.grade) THEN INSERT INTO <i>changes</i> (user, action, old_grade, new_grade, change_date) VALUES (current_user(), “Change grade”, OLD.grade, NEW.grade, Current_date()); END IF; END;// DELIMITER ; UPDATE <i>students_courses</i> SET <i>grade</i> = 85 WHERE student_id = 1 AND course_id = 1;</pre>

```
mysql> select * from students_courses;
+-----+-----+-----+-----+
| student_id | course_id | grade | reg_date |
+-----+-----+-----+-----+
| 1 | 1 | 80 | 2022-01-04 |
| 1 | 2 | 90 | 2022-01-04 |
| 1 | 3 | 100 | 2022-01-04 |
| 1 | 4 | 60 | NULL |
| 2 | 1 | NULL | NULL |
| 2 | 2 | 99 | 2022-01-04 |
| 2 | 3 | 80 | 2022-01-04 |
| 2 | 4 | 75 | NULL |
| 3 | 1 | NULL | NULL |
| 3 | 2 | NULL | NULL |
| 3 | 3 | 75 | NULL |
| 3 | 4 | 70 | 2022-01-04 |
+-----+-----+-----+-----+
12 rows in set (0.01 sec)

mysql> UPDATE students_courses
-> SET grade = 85
-> WHERE student_id = 1 AND course_id = 1;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from students_courses;
+-----+-----+-----+-----+
| student_id | course_id | grade | reg_date |
+-----+-----+-----+-----+
| 1 | 1 | 85 | 2022-01-04 |
| 1 | 2 | 90 | 2022-01-04 |
| 1 | 3 | 100 | 2022-01-04 |
| 1 | 4 | 60 | NULL |
| 2 | 1 | NULL | NULL |
| 2 | 2 | 99 | 2022-01-04 |
| 2 | 3 | 80 | 2022-01-04 |
| 2 | 4 | 75 | NULL |
| 3 | 1 | NULL | NULL |
| 3 | 2 | NULL | NULL |
| 3 | 3 | 75 | NULL |
| 3 | 4 | 70 | 2022-01-04 |
+-----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql> select * from changes;
+-----+-----+-----+-----+-----+-----+
| id | user | action | old_grade | new_grade | change_date |
+-----+-----+-----+-----+-----+-----+
| 1 | ROOT | Change grade | 80 | 85 | 2022-01-05 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

UPDATE *students_courses*
SET *grade* = 95
WHERE *student_id* = 1 AND *course_id* = 2;

```
mysql> UPDATE students_courses
-> SET grade = 95
-> WHERE student_id = 1 AND course_id = 2;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from students_courses;
+-----+-----+-----+-----+
| student_id | course_id | grade | reg_date |
+-----+-----+-----+-----+
| 1 | 1 | 85 | 2022-01-04 |
| 1 | 2 | 95 | 2022-01-04 |
| 1 | 3 | 100 | 2022-01-04 |
| 1 | 4 | 60 | NULL |
| 2 | 1 | NULL | NULL |
| 2 | 2 | 99 | 2022-01-04 |
| 2 | 3 | 80 | 2022-01-04 |
| 2 | 4 | 75 | NULL |
| 3 | 1 | NULL | NULL |
| 3 | 2 | NULL | NULL |
| 3 | 3 | 75 | NULL |
| 3 | 4 | 70 | 2022-01-04 |
+-----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql> select * from changes;
+----+-----+-----+-----+-----+-----+
| id | user | action | old_grade | new_grade | change_date |
+----+-----+-----+-----+-----+-----+
| 1 | ROOT | Change grade | 80 | 85 | 2022-01-05 |
| 2 | ROOT | Change grade | 90 | 95 | 2022-01-05 |
+----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

5 *Create event to delete the changes tables every 5 minute*

```
select @@global.event_scheduler;
set @@global.event_scheduler=1;
.
.
.
.
CREATE EVENT delete_changes
ON SCHEDULE EVERY 5 MINUTE
DO
DELETE FROM changes;
```

```
mysql> select @@global.event_scheduler;
```

```
+-----+
| @@global.event_scheduler |
+-----+
| ON                        |
+-----+
```

```
1 row in set (0.01 sec)
```

```
mysql> CREATE EVENT delete_changes
```

```
-> ON SCHEDULE EVERY 5 MINUTE
```

```
-> DO
```

```
-> DELETE FROM changes;
```

```
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> select * from changes;
```

```
Empty set (0.00 sec)
```