

בס"ד

כותבי הדוח:

אריאל פאליק - 065974289

רועי בוק - 208560102

שלב 1)

הארגון שלנו – אוניברסיטה / מכללה / סמינר.

הארגון שלנו כולל כמה וכמה תת אגפים כמו מרצים, סטודנטים, מעונות, מבחנים וציונים (שזה האגף שאנחנו בחרנו- ציונים).

א. סטודנטים

ב. מרצים

ג. מבחנים

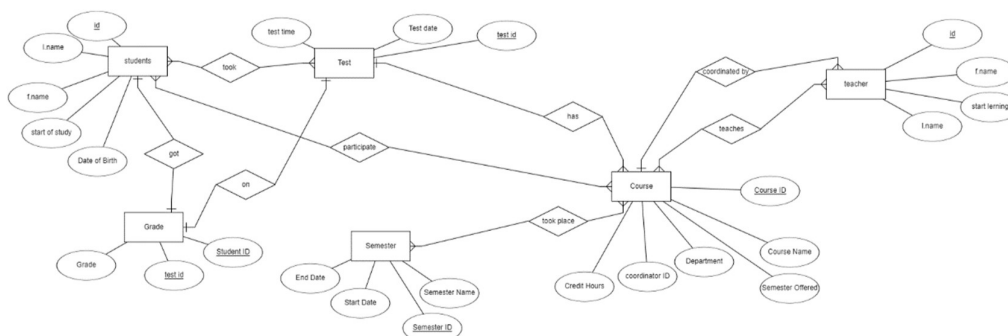
ד. ציונים

ה. קורסים

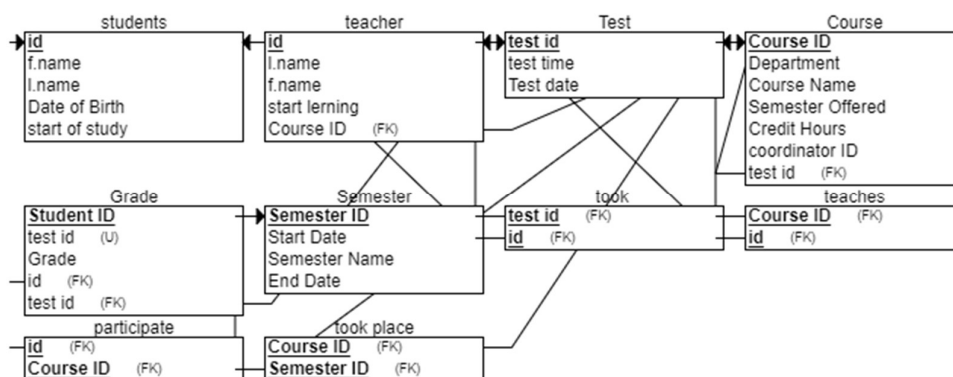
ו. סמסטר

מטרת המערכת היא לקבל נתונים על הציונים של הסטודנטים ולנתח אותם בשיטות סטטיסטיות על מנת להעלות את ממוצע הציונים.

תרשים ERD:



תרשים DSD:



נרמול הסכימה:

הסכימה נמצאת ברמת נרמול NF - 3, מכיוון שלא קיימות תכונות מורכבות (1-NF), אין תכונות התלויות רק בחלק מהמפתח (2-NF), ואין תכונות התלויות בתכונות אחרות שאינן חלק המפתח (3-NF).

```
CREATE TABLE students)
  id INT PRIMARY KEY,
  first_name VARCHAR,(50)
  last_name VARCHAR,(50)
  date_of_birth DATE,
  start_of_study INT
; (
```

```
CREATE TABLE teacher)
  id INT PRIMARY KEY,
  fname VARCHAR,(50)
  lname VARCHAR,(50)
  start_learning INT
; (
```

```
CREATE TABLE course)
  course_id INT PRIMARY KEY,
  course_name VARCHAR,(100)
  department VARCHAR,(100)
  credit_hours INT,
  coordinator_id INT,
  FOREIGN KEY (coordinator_id) REFERENCES teacher(id)
; (
```

```
CREATE TABLE test)
  test_id INT PRIMARY KEY,
  test_time TIME,
  test_date DATE,
```

```
course_id INT,  
FOREIGN KEY (course_id) REFERENCES course(course_id)  
;(  
  

```

```
CREATE TABLE grade)  
  
student_id INT,  
test_id INT,  
grade INT,  
PRIMARY KEY (student_id, test_id),  
FOREIGN KEY (student_id) REFERENCES students(id),  
FOREIGN KEY (test_id) REFERENCES test(test_id)  
;(  
  

```

```
CREATE TABLE semester)  
  
semester_id INT PRIMARY KEY,  
semester_name VARCHAR,(50)  
start_date DATE,  
end_date DATE  
;(  
  

```

```
CREATE TABLE course_teacher)  
  
course_id INT,  
teacher_id INT,  
PRIMARY KEY (course_id, teacher_id),  
FOREIGN KEY (course_id) REFERENCES course(course_id),  
FOREIGN KEY (teacher_id) REFERENCES teacher(id)  
;(  
  

```

```
CREATE TABLE student_test_participation)  
  
student_id INT,  
test_id INT,  
PRIMARY KEY (student_id, test_id),  
  

```

```

FOREIGN KEY (student_id) REFERENCES students(id),

FOREIGN KEY (test_id) REFERENCES test(test_id)

;

CREATE TABLE course_semester)

course_id INT,

semester_id INT,

PRIMARY KEY (course_id, semester_id),

FOREIGN KEY (course_id) REFERENCES course(course_id),

FOREIGN KEY (semester_id) REFERENCES semester(semester_id)

;

```

Show query box

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available. ⓘ

Your SQL query has been executed successfully.

```
desc course_semester;
```

phpMyAdmin

Server: 127.0.0.1 » Database: levdb

Structure SQL Search Query Export Import Operations Privileges Routines Events

Importing into the database "levdb"

File to import:

File may be compressed (gzip, bzip2) or uncompressed.
A compressed file's name must end in `.[format].[compression]`. Example: `.sql.zip`

Browse your computer: (Max: 40MiB)

בחירת קובץ levdb.sql

You may also drag and drop a file on any page.

Character set of the file:

utf-8

phpMyAdmin

Server: 127.0.0.1 » Database: levdb

Structure SQL Search Query Export Import Operations Privileges Routines Events More

Exporting tables from "levdb" database

Export templates:

New template: Existing templates:

Template name: Template name: Create Update Delete

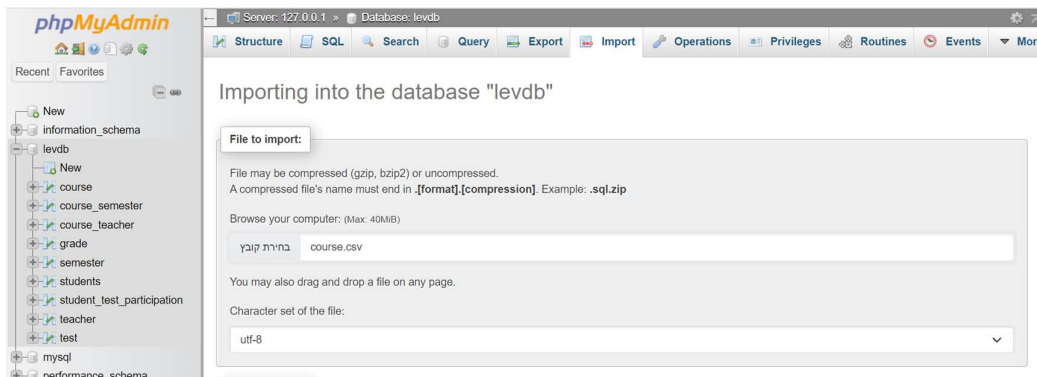
Export method:

☒ Quick - display only the minimal options
☐ Custom - display all possible options

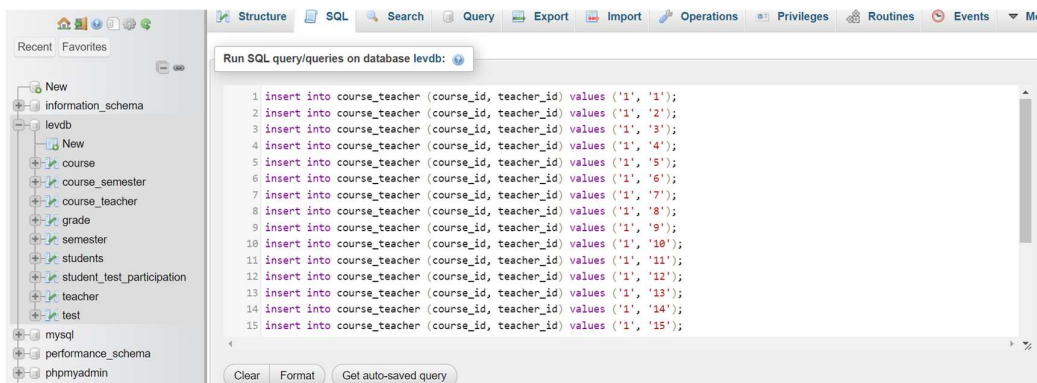
Format:

SQL

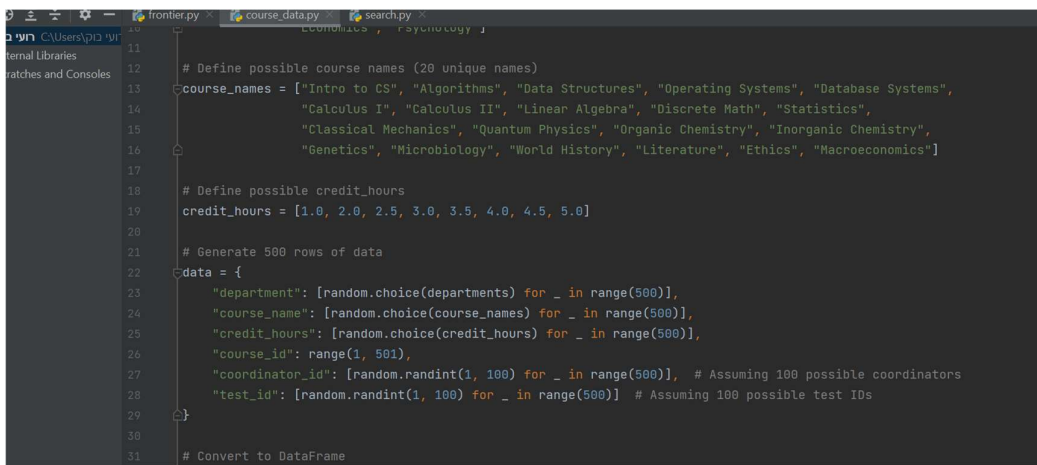
ולהלן שיטת ההכנסה ע"פ פשט:



וכן עוד שיטה של הכנסה ע"פ עוד פשט:



וכן עוד שיטה ע"פ רבני הפייתון:



וכן המקום להודיע להודיע (שכל קווין וכו') שלא השתמשנו בפלט של הקבצי פייתון מכיוון שלא ראינו צורך בטרחה מרובה (שאסורה בימים אלו כמובא בספרים הק'). ולכן תהנו מיצירת המופת שלנו (סליחה בעצם של GPT)

גלוי וידוע לפנינו שאמורה לצאת טבלה אך מפני ענוותנותה של המערכת היא רק דיווחה כי "כל כבודה בת מלך פנימה"

מכיוון שמרצנו הק' ציווה עלינו לעשות את הפרוייקט בmysql phpmyadmini שלא כשאר המרצים שליט"א והמרצות תליט"א כך יוצא מכל הנ"ל שההוראות במודל הם לפי הרוב כי "יחיד ורבים הלכה כרבים"

לכן דוח זה נראה כאילו הוא לאחר תעניות רבות וסיגופים אך אל לכם להביט בקנקן אלא לשתות את היין ההונגרי שנאמר "טעמו וראו כי טוב"

ז"י גזענט ובשורות טובות ישועות ונחמות ובבניין ירושלים ננוחם.

נ.ב. גם אם הציון לא משו לפחות נא להוסיף ציון על הפקטור...

שלב ב

להלן נציג את השאילתות שלנו ומה רצינו לעשות וכו' וכו'

השאילתה הראשונה שלנו היא: המטרה שלנו היא למצוא את 10% של הסטודנטים בעלי הציון הגבוה בשנה מסוימת (בעולם האמיתי האלה שיזכו בהצטיינות בסוף שנה..)

```
# get the top 10% of students with the highest average grades, of a given year
#PREPARE stmt FROM
WITH StudentAverages AS (
    SELECT
        students.id,
        students.first_name,
        students.last_name,
        AVG(grade.grade) AS average_grade,
        YEAR(test.test_date) AS year
    FROM
        students
    JOIN grade ON students.id = grade.student_id
    JOIN test ON grade.test_id = test.test_id
    WHERE
        YEAR(test.test_date) = '2023'
    GROUP BY
        students.id
),
RankedStudents AS (
    SELECT
        *,
        ROW_NUMBER() OVER (ORDER BY average_grade DESC) AS rank
    FROM
        StudentAverages
),
TotalStudents AS (
    SELECT COUNT(*) AS total
    FROM students
)
SELECT
    id,
    first_name,
    last_name,
    average_grade,
    year
FROM
    RankedStudents,
    TotalStudents
WHERE
    rank <= total * 0.10
ORDER BY
    average_grade DESC;
```

תוצאות השאילתה:

id	first_name	last_name	average_grade	year
830	Benn	Dottridge	100.0000	2023
807	Alanna	Edmonstone	100.0000	2023
268	Byrann	Iffe	100.0000	2023
849	Bendicty	Gristwood	100.0000	2023
653	Alasteir	Allberry	100.0000	2023
472	Donia	Jovis	100.0000	2023
496	Roth	Elsegood	99.0000	2023
845	Sully	Betser	99.0000	2023
193	Nicoli	Quibell	99.0000	2023
137	Torre	Pedican	99.0000	2023
989	Raye	Bassingden	98.0000	2023
965	Tanya	Rodear	98.0000	2023
746	Barty	Welbrock	98.0000	2023
405	Dyanne	De Bischof	97.0000	2023
647	Billi	Esslemont	97.0000	2023
820	Linnea	MacConneely	97.0000	2023

השאלתה השנייה שלנו:

באה לגלות מי המרצה שתלמידיו הגיעו להישגים הכי טובים (אולי יש יותר מ1..)

```

# Check which lecturer has the best average
WITH TestAverages AS (
    SELECT
        t.test_id,
        AVG(g.grade) AS average_grade
    FROM
        Test t
    JOIN Grade g ON t.test_id = g.test_id
    GROUP BY
        t.test_id
),
CourseAverages AS (
    SELECT
        c.course_id,
        c.coordinator_ID AS lecturer_id,
        AVG(ta.average_grade) AS average_grade
    FROM
        Course c
    JOIN Test t ON c.course_id = t.course_id
    JOIN TestAverages ta ON t.test_id = ta.test_id
    GROUP BY
        c.course_id,
        c.coordinator_ID
),
LecturerAverages AS (
    SELECT
        l.id AS lecturer_id,
        l.fname,
        l.lname,
        AVG(ca.average_grade) AS average_grade
    FROM
        teacher l
    JOIN CourseAverages ca ON l.id = ca.lecturer_id
    GROUP BY
        l.id,
        l.fname,
        l.lname
),
MaxAverage AS (
    SELECT
        MAX(average_grade) AS max_average_grade
    FROM
        LecturerAverages
)
SELECT
    la.lecturer_id,
    la.fname,
    la.lname,
    la.average_grade
FROM
    LecturerAverages la
JOIN
    MaxAverage ma ON la.average_grade = ma.max_average_grade
ORDER BY
    la.lecturer_id;

```

תוצאות השאלתה:

lecturer_id	fname	lname	average_grade
266	Baily	Revey	100.00000000000000
810	Delcina	Boar	100.00000000000000
988	Philly	Willbourne	100.00000000000000

השאלתה השלישית שלנו: באה לבדוק איזה קורסים היו בעלי ציונים הכי גבוהים מבין הקורסים בשנה מסוימת (שימוש בdate)

```
-- This is a parameterized query where is the parameter.
WITH TestAverages AS (
  SELECT
    t.test_id,
    t.course_id,
    AVG(g.grade) AS average_grade,
    YEAR(t.test_date) AS year
  FROM
    Test t
  JOIN Grade g ON t.test_id = g.test_id
  WHERE
    EXTRACT(YEAR FROM t.test_date) = '2023' -- Use EXTRACT(YEAR FROM ...) to get the year from test_date
  GROUP BY
    t.test_id,
    t.course_id
),
CourseAverages AS (
  SELECT
    course_id, year,
    AVG(average_grade) AS average_grade
  FROM
    TestAverages
  GROUP BY
    course_id,
    year
),
MaxCourseAverage AS (
  SELECT
    MAX(average_grade) AS max_average_grade
  FROM
    CourseAverages
)
SELECT
  ca.course_id,
  c.course_name,
  ca.average_grade,
  year,
  #s.semester_name
FROM
  CourseAverages ca
JOIN
  MaxCourseAverage mca ON ca.average_grade = mca.max_average_grade
JOIN
  Course c ON ca.course_id = c.course_id
ORDER BY
  ca.course_id;
```

והנה התוצאות של השאלתה:

course_id	course_name	average_grade	year
135	Microbiology	100.00000000	2023
475	Inorganic Chemistry	100.00000000	2023
506	Linear Algebra	100.00000000	2023
900	Inorganic Chemistry	100.00000000	2023

בשאלתה הרביעית


```

1 SELECT
2     t.start_learning AS seniority,
3     AVG(g.grade) AS average_grade
4 FROM
5     grade g
6 JOIN
7     test te ON g.test_id = te.test_id
8 JOIN
9     course c ON te.course_id = c.course_id
10 JOIN
11     course_teacher ct ON c.course_id = ct.course_id
12 JOIN
13     teacher t ON ct.teacher_id = t.id
14 GROUP BY
15     t.start_learning;

```

באה לבדוק השפעת הוותק של המרצה על הציונים אך לאחר הרצות גילינו שקיים טעות בבסיס נתונים (שהמוק דאטה ייצר לנו כל המרצים לימדו את אותו קורס...) ולכן שינינו את הבסיס נתונים ולהן השאילתה המתוקנת:

```

SELECT
    t.start_learning AS seniority,
    AVG(g.grade) AS average_grade
FROM
    grade g
JOIN
    test te ON g.test_id = te.test_id
JOIN
    course c ON te.course_id = c.course_id
JOIN
    course_teacher ct ON c.course_id = ct.course_id
JOIN
    teacher t ON ct.teacher_id = t.id
GROUP BY
    t.start_learning
ORDER BY average_grade DESC;

```

והנה התוצאות של השאילתה:

seniority	average_grade ▾ 1
1986	80.0000
1988	71.6667
2013	70.8571
1970	69.3333
2010	66.6316
1999	65.2273
1992	64.6000
2003	64.4231
1989	64.2000

Table	Action	Rows	Type
<input type="checkbox"/> course	★ Browse Structure Search Insert Empty Drop	960	Inn
<input type="checkbox"/> course_semester	★ Browse Structure Search Insert Empty Drop	1,000	Inn
<input type="checkbox"/> course_teacher	★ Browse Structure Search Insert Empty Drop	532	Inn
<input type="checkbox"/> grade	★ Browse Structure Search Insert Empty Drop	962	Inn
<input type="checkbox"/> semester	★ Browse Structure Search Insert Empty Drop	1,000	Inn
<input type="checkbox"/> students	★ Browse Structure Search Insert Empty Drop	1,000	Inn
<input type="checkbox"/> student_test_participation	★ Browse Structure Search Insert Empty Drop	988	Inn
<input type="checkbox"/> teacher	★ Browse Structure Search Insert Empty Drop	1,000	Inn
<input type="checkbox"/> test	★ Browse Structure Search Insert Empty Drop	958	Inn
9 tables	Sum	8,400	Inn

להלן שאלות המחיקה שלנו: השאלתה הראשונה בעצם באה למחוק נתונים על מרצים שיצאו לפנסיה

Server: 127.0.0.1 Database: leldb

Structure SQL Search Query Export Import Operat

Show query box

✓ 94 rows deleted. (Query took 0.0013 seconds.)

```
DELETE FROM teacher WHERE teacher.start_learning < year(CURRENT_DATE())-35;
```

[Edit inline] [Edit] [Create PHP code]

Server: 127.0.0.1 Database: leldb

Structure SQL Search Query Export Import Operations Privileges

Show query box

✓ 0 rows deleted. (Query took 0.0043 seconds.)

```
DELETE from teacher WHERE teacher.id not in (SELECT teacher.id FROM course_teacher);
```

[Edit inline] [Edit] [Create PHP code]

השאלתה השניה באה למחוק את כל המרצים שבעצם לא לימדו אף קורס (הבסיס נתונים שלנו היה טוב ולכן לא היו כאלה..)

שאלות עדכון:

Server: 127.0.0.1 » Database: levdb » Table: grade

Browse Structure SQL Search Insert Export Import

Show query box

Showing rows 0 - 3 (4 total, Query took 0.0021 seconds)

```
SELECT grade.grade FROM grade WHERE grade.test_id = 135;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key:

Extra options

	grade
<input type="checkbox"/> Edit Copy Delete	51
<input type="checkbox"/> Edit Copy Delete	73
<input type="checkbox"/> Edit Copy Delete	21
<input type="checkbox"/> Edit Copy Delete	46

Check all With selected: Edit Copy Delete Export

לפני:

Server: 127.0.0.1 » Database: levdb » Table: grade

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Show query box

MySQL returned an empty result set (i.e. zero rows) (Query took 0.0050 seconds)

```
CREATE PROCEDURE updategrades1 (IN factor DECIMAL(5,2), IN test_to_update INT, IN min_grade INT) BEGIN UPDATE grade SET grade = LEAST(grade * (1 + factor / 100), 100) WHERE test_id = test_to_update AND grade > min_grade; END;
```

MySQL returned an empty result set (i.e. zero rows) (Query took 0.0039 seconds)

```
CALL updategrades1(140, 135, 50);
```

הקוד:

אחרי:

Server: 127.0.0.1 » Database: levdb » Table: grade

Browse Structure SQL Search Insert Export Ex

Show query box

Showing rows 0 - 3 (4 total, Query took 0.0004 seconds)

```
SELECT grade.grade FROM grade WHERE grade.test_id = 135;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table

Extra options

	grade
<input type="checkbox"/> Edit Copy Delete	56
<input type="checkbox"/> Edit Copy Delete	80
<input type="checkbox"/> Edit Copy Delete	21
<input type="checkbox"/> Edit Copy Delete	46

Check all With selected: Edit Copy Delete

השאלתה מדמה פקטור של 10 אחוז החל מציון מסויים (דבר שישנו במקור..)

עוד שאילתת עדכון שעשינו היא באה לדמות ערעור של סטודנט על המבחן (יש יכולת שהציון יעלה אך גם שירד כמו בדוגמא שלנו וכבר היו דוגמאות במכון...) ניתן לראות את הטבלה לפני, את הקוד ואת הטבלה לאחר השינוי...

שאלתה עם פרמטרים:

בעצם בדקנו ממוצע של פרק זמן מסויים (יעיל לבדוק האם בזמן קורונה או מלחמה הממוצע נמוך מאשר סתם זמן מקביל וכו...)

עוד שאילתה עם פרמטרים זה בא להקל על מזכירות ציונים (בעצם כל ציון שתכניס נקבל רשימה של הסטודנטים שקיבלו אותו)


```

1 -- ParamsQueries query 2
2 -- Prepare the statement with placeholders
3 PREPARE stmt FROM 'SELECT *
4     FROM students s, grade g
5     WHERE g.grade = ?';
6
7 -- Set the parameters
8 SET @random_grade = '67';
9
10 -- Execute the prepared statement with the parameters
11 EXECUTE stmt USING @random_grade;
12
13 -- Deallocate the prepared statement (optional)
14 DEALLOCATE PREPARE stmt;

```

ולהלן האילוצים שעשינו ותיאור וכו' אמממה בעקבות ניסיונותינו לעשות אילוצים ועקב שגיאות חוזרות ונשנות הרינו מודיעים קבל עם ועדה (איזה עדה שתבחרו) שיש בעייה לעשות זאת והנה ההוכחה מ.ש.ל.

<https://mariadb.com/kb/en/problem-with-alter-table-and-add-constraint/>



[PRODUCTS](#)
[SERVICES](#)
[PRICING](#)
[RESOURCES](#)

[Knowledge Base](#) » [The Community](#) » [Problem with ALTER TABLE and ADD CONSTRAINT](#)

[Home](#)
[Open Questions](#)
[MariaDB Server](#)
[MariaDB MaxScale](#)
[MariaDB ColumnStore](#)
[Connectors](#)
[History](#)
[Source](#)
[Flag as Spam / Inappropriate](#)

Problem with ALTER TABLE and ADD CONSTRAINT

My MariaDB version id 10.5.9, running on Manjaro Nibia (20.2.1).

I am trying to add a CHECK constraint to an existing field. This is my statement, modelled on the syntax at <https://mariadb.com/kb/en/constraint/>

```
ALTER TABLE sources ADD CONSTRAINT chk_under1000 CHECK (source_dances < 1000);
```

I get the following error: "A new statement was found, but no delimiter between it and the previous one. (near CHECK)". I get this error irrespective of whether I use backtick field delimiters or not. phpMyAdmin also flags up CHECK as an error.

What am I doing wrong, please?

Michael Barraclough

Answer

Answered by [Ian Gilfillan](#) in [this comment](#).

This appears to be a phpMyAdmin error (and the issue has been an [open bug since 2017](#)), so I suggest using an alternative client.

Created
3 years, 3 months ago

Modified
3 years, 3 months ago


אילוח שכן עבד לנו זה
alter table students modify first_name varchar(50) NOT NULL;
האילוח אומר לנו שחייב שיהיה שם פרטי לסטודנט

Table	Create Table
students	<pre>CREATE TABLE `students` (`id` int(11) NOT NULL, `first_name` varchar(50) NOT NULL, `last_name` varchar(50) DEFAULT NULL, `date_of_birth` date DEFAULT NULL, `start_of_study` int(11) DEFAULT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci</pre>

Error

SQL query: [Copy](#)

```
insert into students (id, first_name) values (1111, null);
```

MySQL said: 

#1048 - Column 'first_name' cannot be null

✓ 1 row inserted. (Query took 0.0008 seconds.)

```
insert into students (id) values (1111);
```

[[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)]

 Warning: #1364 Field 'first_name' doesn't have a default value

עוד אילון ;alter table grade modify grade int(11) default 0
בעצם כל ציון שאין לו ציון יקבל ציון 0

✓ 1 row inserted. (Query took 0.0010 seconds.)

```
INSERT INTO grade (student_id, test_id) VALUES (1,1);
```

[[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)]

✓ Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

```
SELECT * FROM grade WHERE student_id=1 AND test_id=1;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 ▾ | Filter rows:

Extra options

	student_id	test_id	grade
<input type="checkbox"/> Edit Copy Delete	1	1	0

עוד אילון ששם הקורס יהיה ייחודי `unique` `varchar(100)` `course_name` `alter table course modify`;

Error

SQL query: [Copy](#)

```
alter table course modify course_name varchar(100) unique;
```

MySQL said: ?

#1062 - Duplicate entry 'Statistics' for key 'course_name'

~~~שלב ג~~~

2 פונקציות:

פונקציה של קורסים:

פונקציה זו מחזירה את הקורסים שהסטודנט נבחן בהם, והרי הקוד שלה (קודה):

DELIMITER //

```

CREATE FUNCTION get_student_courses(p_student_id INT) RETURNS
VARCHAR(1000)
BEGIN
    DECLARE course_list VARCHAR(1000) DEFAULT '';
    DECLARE course_name VARCHAR(100);
    DECLARE done INT DEFAULT 0;
    DECLARE course_cursor CURSOR FOR
        SELECT c.course_name
        FROM course c
        JOIN student_test_participation stp ON c.course_id =
stp.test_id
        WHERE stp.student_id = p_student_id;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

    OPEN course_cursor;

    course_loop: LOOP
        FETCH course_cursor INTO course_name;
        IF done THEN
            LEAVE course_loop;
        END IF;
        SET course_list = CONCAT(course_list, course_name, ', ');
    END LOOP;

    CLOSE course_cursor;

    -- Remove the trailing comma and space
    IF CHAR_LENGTH(course_list) > 2 THEN
        SET course_list = LEFT(course_list, CHAR_LENGTH(course_list) -
2);
    END IF;

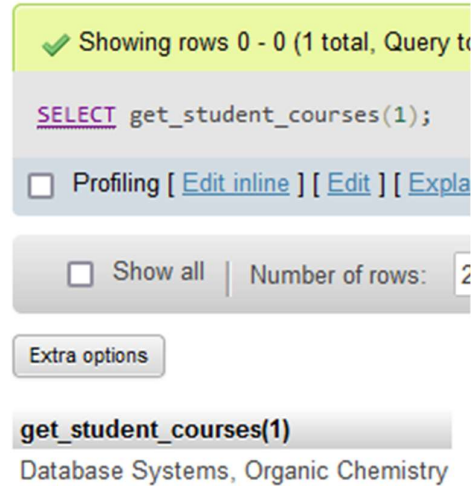
    RETURN course_list;
END;
//

DELIMITER ;

```

הא לכם הוכחה שהפונקציה עובדת:





פונקציה של ציונים:

זוהי פונקציה המחזירה את ממוצע ציוניו של סטודנט, והרי קודה:

```
DELIMITER //
```

```
CREATE FUNCTION calculate_average_grade(p_student_id INT) RETURNS FLOAT
BEGIN
    DECLARE avg_grade FLOAT;
    DECLARE total_grades INT;
    DECLARE grade_count INT;

    -- Handle exceptions
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        -- Return -1 in case of error
        SET avg_grade = -1;
    END;

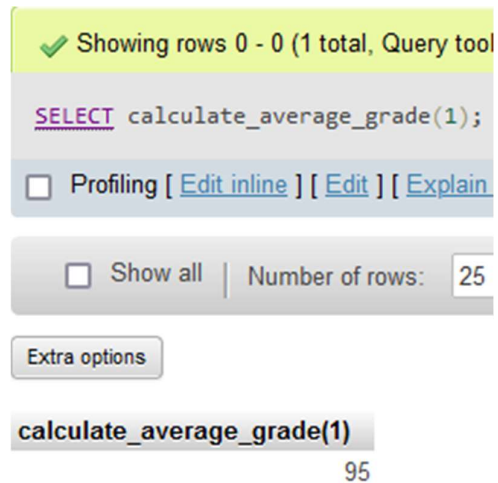
    -- Calculate total grades and count
    SELECT SUM(grade), COUNT(*) INTO total_grades, grade_count
    FROM grade
    WHERE student_id = p_student_id;

    -- Calculate average
    IF grade_count > 0 THEN
        SET avg_grade = total_grades / grade_count;
    ELSE
        SET avg_grade = 0;
    END IF;

    RETURN avg_grade;
END;
//
```

DELIMITER ;

הא לכם הוכחה שהפונקציה עושה את המוטל עליה:



## 2 פרוצדורות:

פרוצדורה של קורסים:

תפקידה של פרוצדורה זו הוא לשייך מרצה לקורס, והנה קודה:

```
DELIMITER //
CREATE PROCEDURE assign_teacher_to_course(IN p_course_id INT, IN
p_teacher_id INT)
BEGIN
    DECLARE course_exists INT;
    DECLARE teacher_exists INT;
    DECLARE error_message VARCHAR(255) DEFAULT '';

    -- Check if the course exists
    SELECT COUNT(*) INTO course_exists FROM course WHERE course_id =
p_course_id;
    IF course_exists = 0 THEN
        SET error_message = 'Course does not exist';
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = error_message;
    END IF;

    -- Check if the teacher exists
    SELECT COUNT(*) INTO teacher_exists FROM teacher WHERE id =
p_teacher_id;
    IF teacher_exists = 0 THEN
        SET error_message = 'Teacher does not exist';
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = error_message;
    END IF;

    -- Check if the teacher is already assigned to the course
```

```

SELECT COUNT(*) INTO teacher_exists FROM course_teacher WHERE
course_id = p_course_id AND teacher_id = p_teacher_id;
IF teacher_exists > 0 THEN
    SET error_message = 'Teacher is already assigned to the
course';
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = error_message;
END IF;

-- Attempt to assign teacher to course
BEGIN
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION
    BEGIN
        GET DIAGNOSTICS CONDITION 1
        error_message = MESSAGE_TEXT;
        SELECT CONCAT('Error assigning teacher to course: ',
error_message) AS detailed_error;
    END;
    INSERT INTO course_teacher (course_id, teacher_id) VALUES
(p_course_id, p_teacher_id);
END;
END;
//
DELIMITER ;

```

חוצים לראות שהיא עובדת? קבלו:

נבדוק, ברשותכם, האם המרצה מס' 1 מלמד את הקורס מס' 1:

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 second)

```
SELECT * FROM course_teacher WHERE course_id = 1 AND teacher_id = 1;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

| course_id | teacher_id |
|-----------|------------|
|-----------|------------|

חזיז ורעם! הוא לא מלמד את הקורס! נפנה מיד לתיקון העוול ההיסטורי (בתקווה שבג"ץ לא יפסול):

✓ MySQL returned an empty result set (i.e. :)

```
CALL assign_teacher_to_course(1, 1);
```

בחיל וברעדה נבדוק האם המרצה שובץ:

✓ Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

`SELECT * FROM course_teacher WHERE course_id = 1 AND teacher_id = 1;`

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows: 25 | Filter rows:

Extra options

|                                                                                           | course_id | teacher_id |
|-------------------------------------------------------------------------------------------|-----------|------------|
| <input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> | 1         | 1          |

חי זקני! הלם ותדהמה! הפרוצדורה עבדה!

פרוצדורה של ציונים:

פרוצדורה זו מעדכנת את ציוניו של סטודנט, וזה לך הקוד:

DELIMITER //

```
CREATE PROCEDURE update_student_grade(IN p_student_id INT, IN p_test_id
INT, IN p_new_grade INT)
BEGIN
    DECLARE old_grade INT;

    -- Handle exceptions
    DECLARE EXIT HANDLER FOR SQLWARNING
    BEGIN
        -- Output error message
        SELECT 'Error updating grade';
    END;

    -- Update grade
    SELECT grade INTO old_grade FROM grade WHERE student_id =
p_student_id AND test_id = p_test_id;

    IF old_grade IS NOT NULL THEN
        UPDATE grade SET grade = p_new_grade WHERE student_id =
p_student_id AND test_id = p_test_id;
    ELSE
        INSERT INTO grade (student_id, test_id, grade) VALUES
(p_student_id, p_test_id, p_new_grade);
    END IF;
END;
//
```

DELIMITER ;

עובדת? יצאנו לבדוק:

זהו ציונו של הסטודנט מס' 1 בקורס מס' 1:

Showing rows 0 - 0 (1 total, Query took 0.0026 seconds.)

```
SELECT * FROM grade WHERE student_id = 1 AND test_id = 1;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Export](#) ]

☐ Show all | Number of rows: 25 | Filter rows:

Extra options

|                                                                                           | student_id | test_id | grade |
|-------------------------------------------------------------------------------------------|------------|---------|-------|
| <input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> | 1          | 1       | 95    |

אכן כן, 95, אתם לא טועים. מצטיין נשיא.

הסטודנט שלנו לא פראייר ורוצה להתקבל למשרה הנחשקת של בודק תרגילים במכון לב, ולכן הוא מגיש ערעור, לאחר דין ודברים וחילופי אש המרצה נכנע, והמזת"ל ניגשים לעדכן את הציון:

MySQL returned an empty result set (i.e. :)

```
CALL update_student_grade(1, 1, 100);
```

המתח גואה, תופים בבקשה:

Showing rows 0 - 0 (1 total, Query took 0.0006 seconds.)

```
SELECT * FROM grade WHERE student_id = 1 AND test_id = 1;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Export](#) ]

☐ Show all | Number of rows: 25 | Filter rows:

Extra options

|                                                                                           | student_id | test_id | grade |
|-------------------------------------------------------------------------------------------|------------|---------|-------|
| <input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> | 1          | 1       | 100   |

שיאו! עכשיו הוא יוכל לבדוק תרגילים בשכר מינימום!

## 2 תוכניות ראשיות:

תוכנית ראשית של קורסים:

תוכנית זו, נו איך לומר, קוראת לפונקציה ולפרוצדורה של הקורסים. וכך היא נראית:

DELIMITER //

```

CREATE PROCEDURE main_for_courses()
BEGIN
    DECLARE student_id INT DEFAULT 1;
    DECLARE course_id INT DEFAULT 1;
    DECLARE teacher_id INT DEFAULT 1;
    DECLARE courses VARCHAR(1000);

    -- Call procedure to assign teacher to course
    CALL assign_teacher_to_course(course_id, teacher_id);

    -- Call function to get student's courses
    SET courses = get_student_courses(student_id);

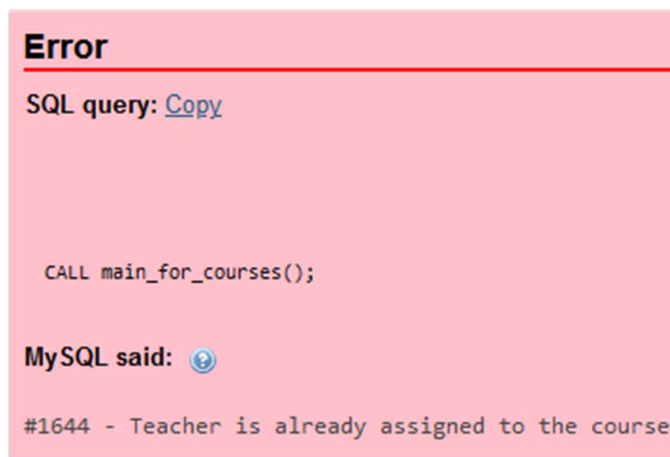
    -- Output the list of courses
    SELECT courses;

    -- the DECLARE and the SET are just to put in more programmatic
stuff
END;
//

DELIMITER ;

```

התוכנית אמורה לזרוק הפעם חריגה, כי כבר שיבצנו את המרצה מס' 1 לקורס מס' 1, כזכור לקורא, ועל הדרך להדגים את הדפסת החריגות:



מהמם.

תוכנית ראשית של ציונים:

תכנית זאת מה עושה? קוראת לפונקציה ולפרוצדורה של הציונים, זה מה.

וזוה הקוד:

```
DELIMITER //
```

```
CREATE PROCEDURE main_for_grades()
```

```

BEGIN
  DECLARE student_id INT DEFAULT 1;
  DECLARE test_id INT DEFAULT 1;
  DECLARE new_grade INT DEFAULT 95;
  DECLARE avg_grade FLOAT;

  -- Call procedure to update grade
  CALL update_student_grade(student_id, test_id, new_grade);

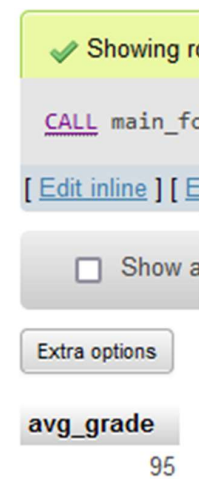
  -- Call function to calculate average grade
  SET avg_grade = calculate_average_grade(student_id);

  -- Output the average grade
  SELECT avg_grade;
END;
//

DELIMITER ;

```

נריץ:



וואו.

ובזאת ידיו, תמה לה הצגת שלב 3 המושלם, שבח לאל בורא עולם