مشروع ٣ - دورة SQL3

اسم المتدرب: مريم الردادي

اسم المشرف: غادة المطيري

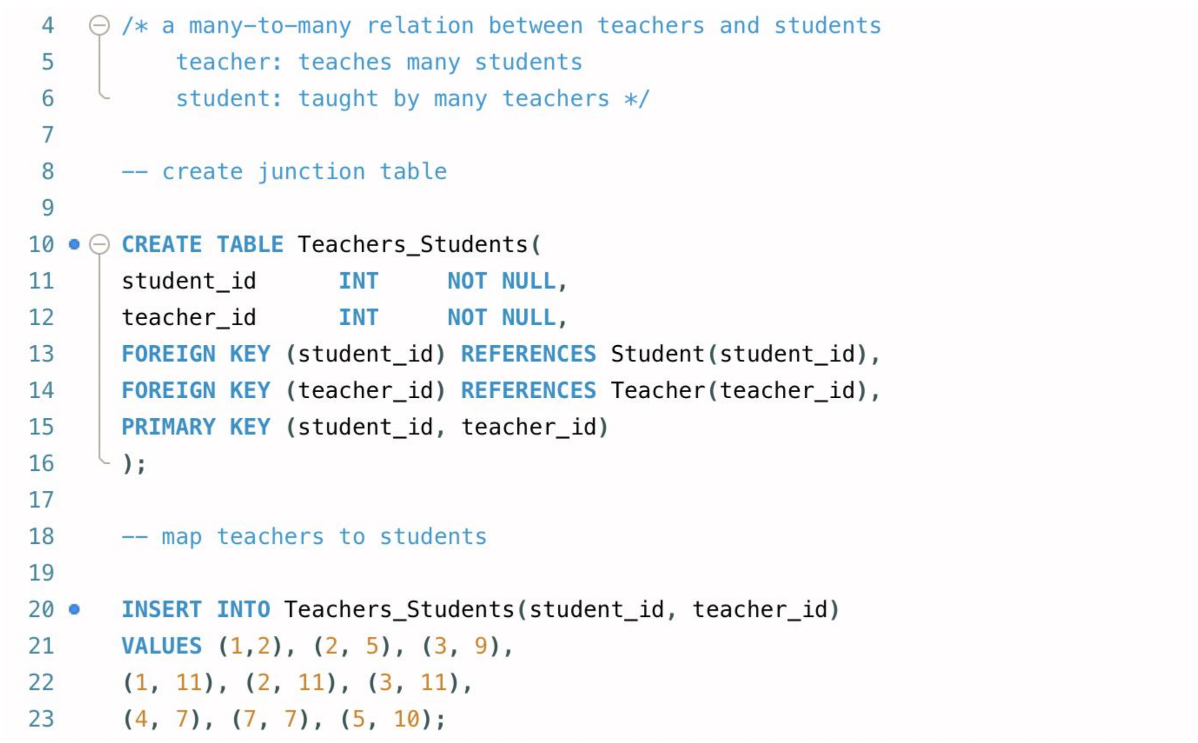
اسم الفريق: السودة

وصف المشروع: مشروع يوظف مفاهيم SQL التالية على قاعدة بيانات مدرسة ثانوية:

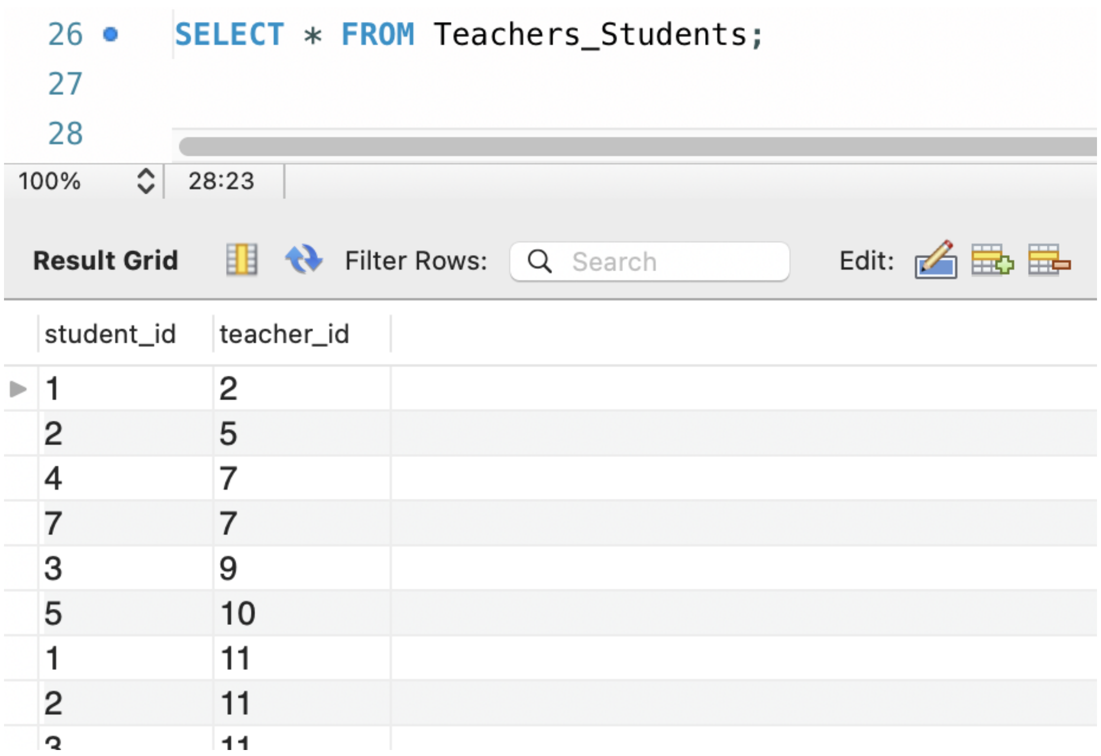
Procedures ,Indexes ,Views ,Joins ,Relationships ,Foreign keys

متطلبات المشروع:

- انشاء علاقة بين جدول المعلمين والطلاب (بحيث أن المعلم يدرّس اكثر من طالب، والطالب يقوم بتدريسه أكثر من معلم)



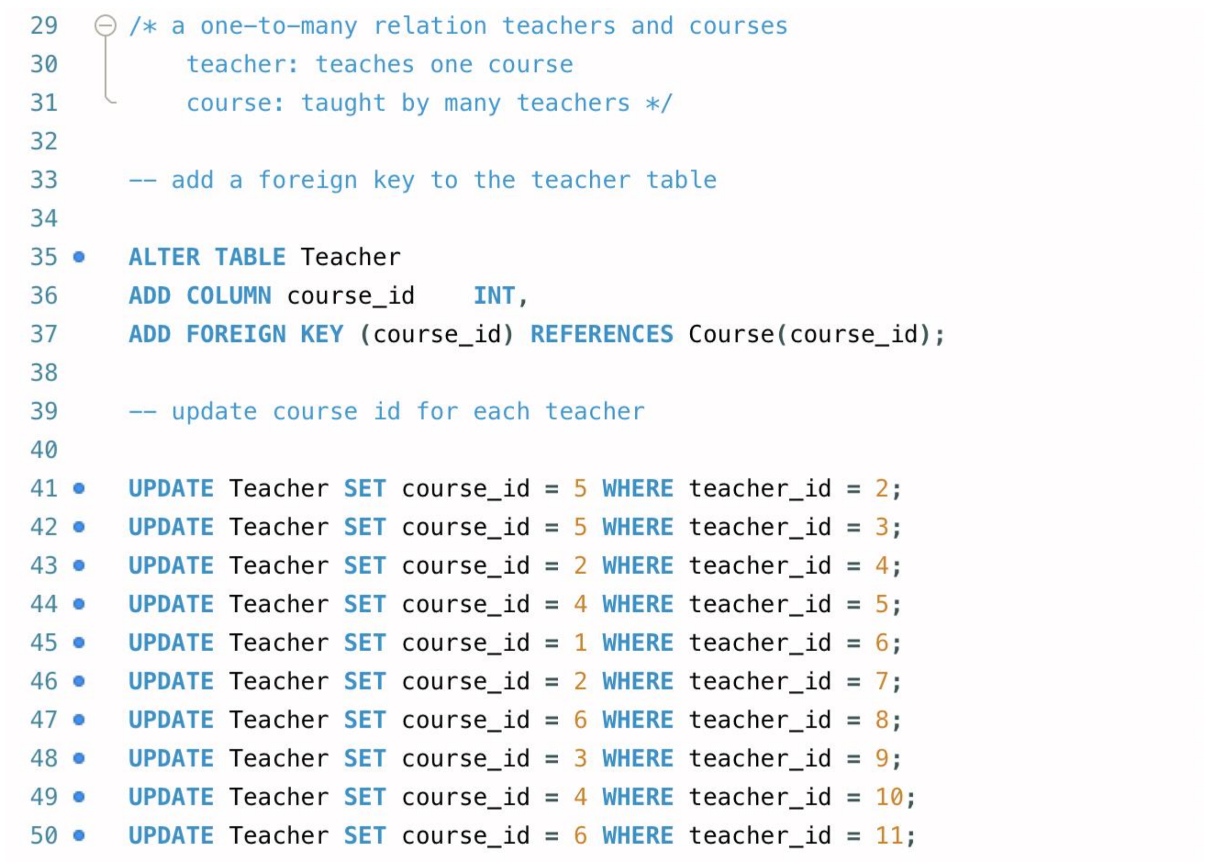
```sql
/* a many-to-many relation between teachers and students
    teacher: teaches many students
    student: taught by many teachers */

-- create junction table

CREATE TABLE Teachers_Students(
student_id      INT     NOT NULL,
teacher_id      INT     NOT NULL,
FOREIGN KEY (student_id) REFERENCES Student(student_id),
FOREIGN KEY (teacher_id) REFERENCES Teacher(teacher_id),
PRIMARY KEY (student_id, teacher_id)
);

-- map teachers to students

INSERT INTO Teachers_Students(student_id, teacher_id)
VALUES (1,2), (2, 5), (3, 9),
(1, 11), (2, 11), (3, 11),
(4, 7), (7, 7), (5, 10);
```



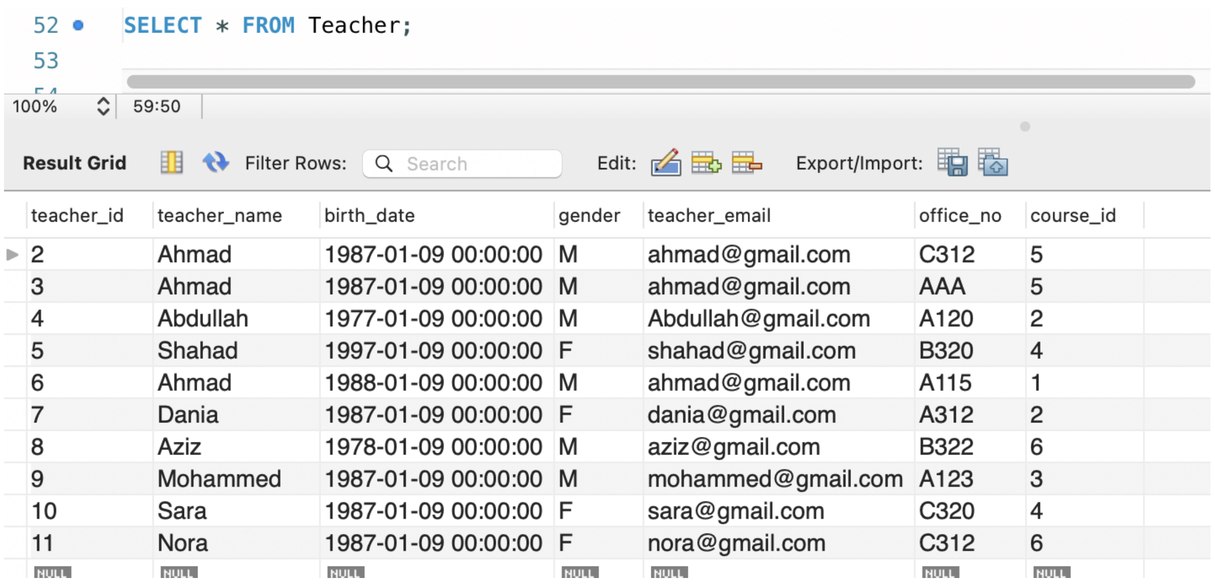
```
SELECT * FROM Teachers_Students;
```

student_id	teacher_id
1	2
2	5
4	7
7	7
3	9
5	10
1	11
2	11
3	11

- انشاء علاقة بين جدول المواد والمعلمين (بحيث أن المعلم يقوم بتدريس مادة واحدة فقط، والمادة يقوم بتدريسها أكثر من معلم).



```sql
/* a one-to-many relation teachers and courses
    teacher: teaches one course
    course: taught by many teachers */

-- add a foreign key to the teacher table

ALTER TABLE Teacher
ADD COLUMN course_id    INT,
ADD FOREIGN KEY (course_id) REFERENCES Course(course_id);

-- update course id for each teacher

UPDATE Teacher SET course_id = 5 WHERE teacher_id = 2;
UPDATE Teacher SET course_id = 5 WHERE teacher_id = 3;
UPDATE Teacher SET course_id = 2 WHERE teacher_id = 4;
UPDATE Teacher SET course_id = 4 WHERE teacher_id = 5;
UPDATE Teacher SET course_id = 1 WHERE teacher_id = 6;
UPDATE Teacher SET course_id = 2 WHERE teacher_id = 7;
UPDATE Teacher SET course_id = 6 WHERE teacher_id = 8;
UPDATE Teacher SET course_id = 3 WHERE teacher_id = 9;
UPDATE Teacher SET course_id = 4 WHERE teacher_id = 10;
UPDATE Teacher SET course_id = 6 WHERE teacher_id = 11;
```



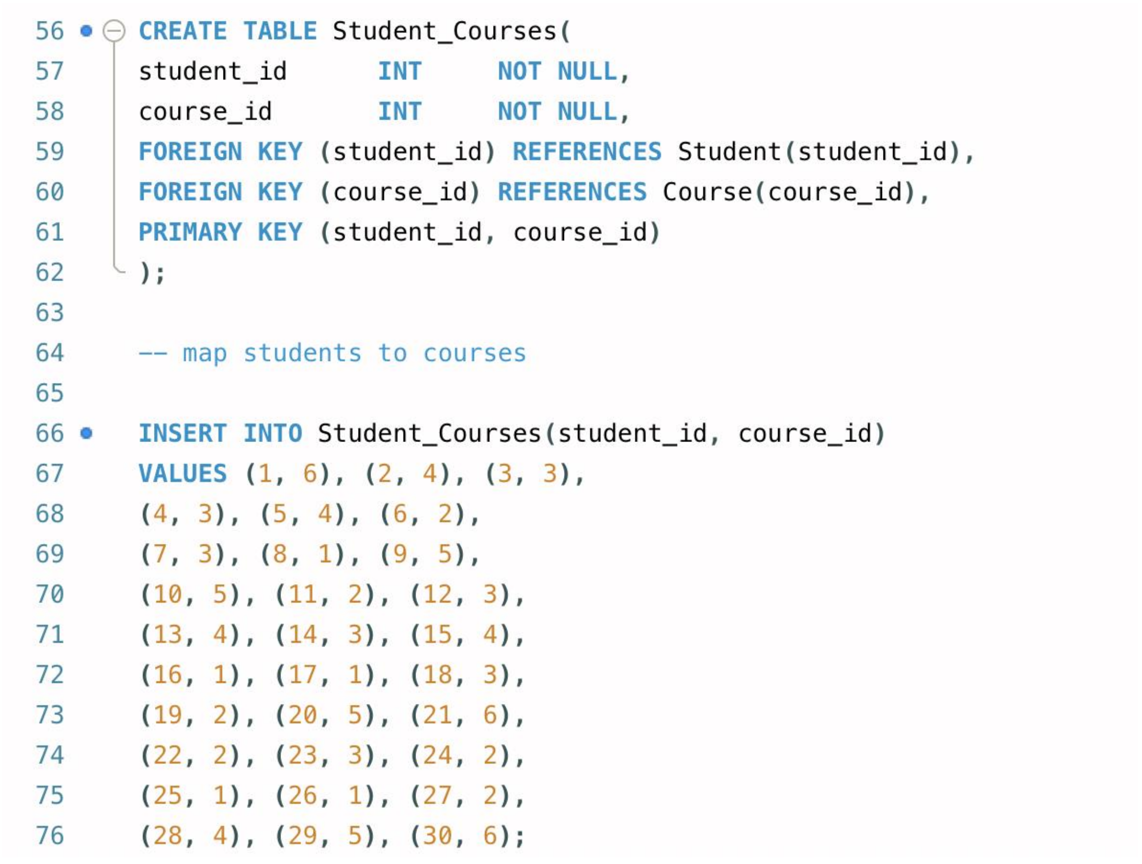
```
SELECT * FROM Teacher;
```

100% 59:50

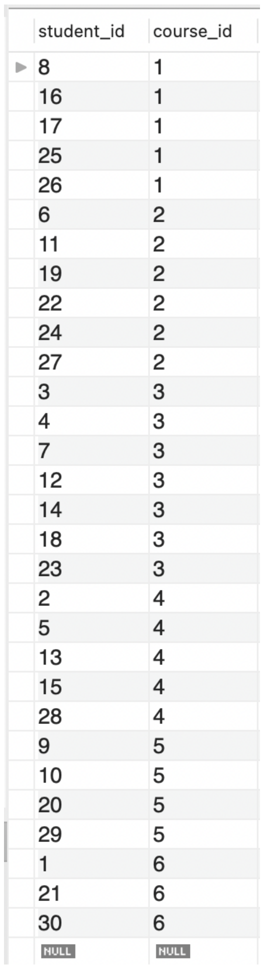
Result Grid Filter Rows: Search Edit: Export/Import:

teacher_id	teacher_name	birth_date	gender	teacher_email	office_no	course_id
2	Ahmad	1987-01-09 00:00:00	M	ahmad@gmail.com	C312	5
3	Ahmad	1987-01-09 00:00:00	M	ahmad@gmail.com	AAA	5
4	Abdullah	1977-01-09 00:00:00	M	Abdullah@gmail.com	A120	2
5	Shahad	1997-01-09 00:00:00	F	shahad@gmail.com	B320	4
6	Ahmad	1988-01-09 00:00:00	M	ahmad@gmail.com	A115	1
7	Dania	1987-01-09 00:00:00	F	dania@gmail.com	A312	2
8	Aziz	1978-01-09 00:00:00	M	aziz@gmail.com	B322	6
9	Mohammed	1987-01-09 00:00:00	M	mohammed@gmail.com	A123	3
10	Sara	1987-01-09 00:00:00	F	sara@gmail.com	C320	4
11	Nora	1987-01-09 00:00:00	F	nora@gmail.com	C312	6
NULL	NULL	NULL	NULL	NULL	NULL	NULL

- انشاء علاقة بين جدول المواد والطلاب (بحيث أن الطالب يدرس أكثر مادة، والمادة يدرسها أكثر من طالب).

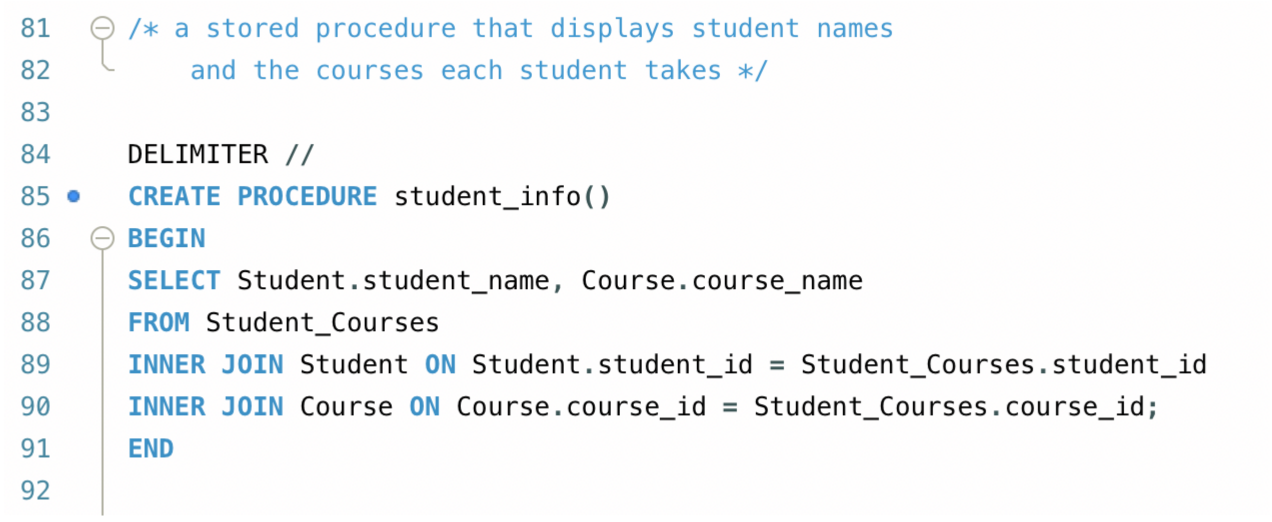


```sql
CREATE TABLE Student_Courses(
student_id      INT     NOT NULL,
course_id       INT     NOT NULL,
FOREIGN KEY (student_id) REFERENCES Student(student_id),
FOREIGN KEY (course_id) REFERENCES Course(course_id),
PRIMARY KEY (student_id, course_id)
);

-- map students to courses

INSERT INTO Student_Courses(student_id, course_id)
VALUES (1, 6), (2, 4), (3, 3),
(4, 3), (5, 4), (6, 2),
(7, 3), (8, 1), (9, 5),
(10, 5), (11, 2), (12, 3),
(13, 4), (14, 3), (15, 4),
(16, 1), (17, 1), (18, 3),
(19, 2), (20, 5), (21, 6),
(22, 2), (23, 3), (24, 2),
(25, 1), (26, 1), (27, 2),
(28, 4), (29, 5), (30, 6);
```



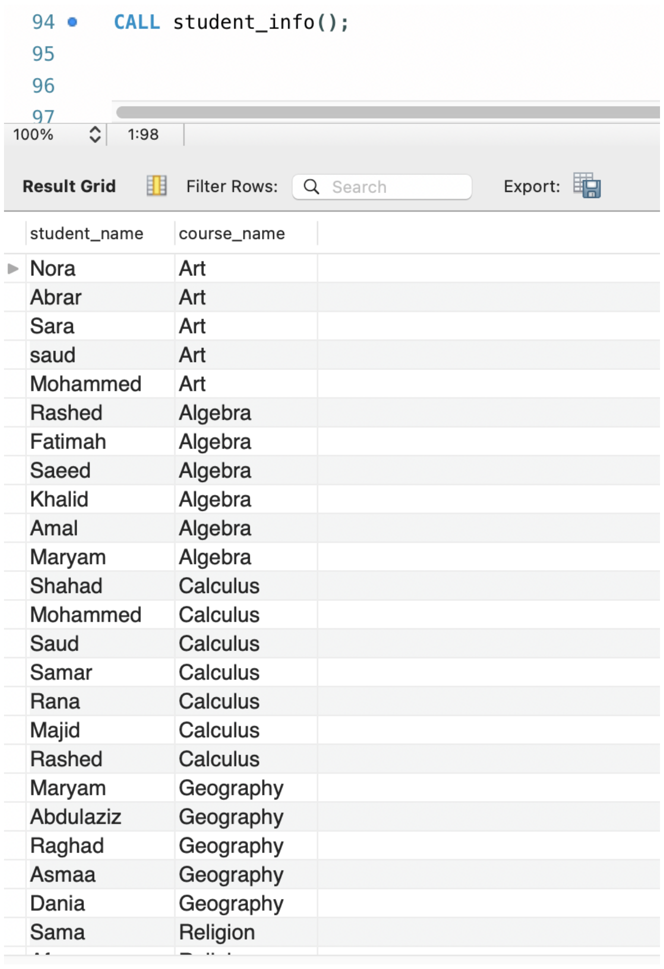
student_id	course_id
8	1
16	1
17	1
25	1
26	1
6	2
11	2
19	2
22	2
24	2
27	2
3	3
4	3
7	3
12	3
14	3
18	3
23	3
2	4
5	4
13	4
15	4
28	4
9	5
10	5
20	5
29	5
1	6
21	6
30	6
NULL	NULL

- قم بانشاء Procedure باسم student_info يعرض اسماء الطلاب و المواد يحتوي على جميع البيانات المشتركه بين جدول المواد و الطلاب



```
/* a stored procedure that displays student names
    and the courses each student takes */

DELIMITER //
CREATE PROCEDURE student_info()
BEGIN
SELECT Student.student_name, Course.course_name
FROM Student_Courses
INNER JOIN Student ON Student.student_id = Student_Courses.student_id
INNER JOIN Course ON Course.course_id = Student_Courses.course_id;
END

```

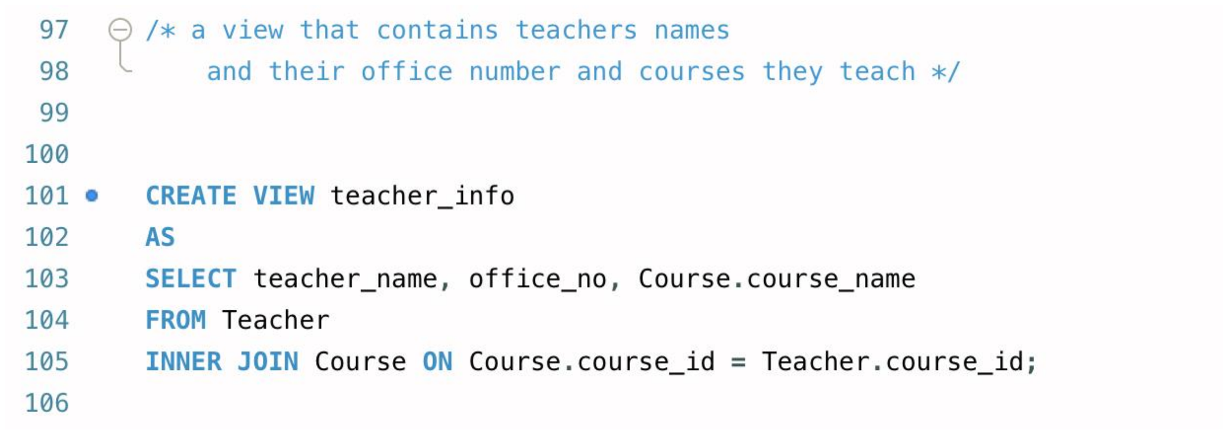
  - قم باستدعائها.



```sql
CALL student_info();
```

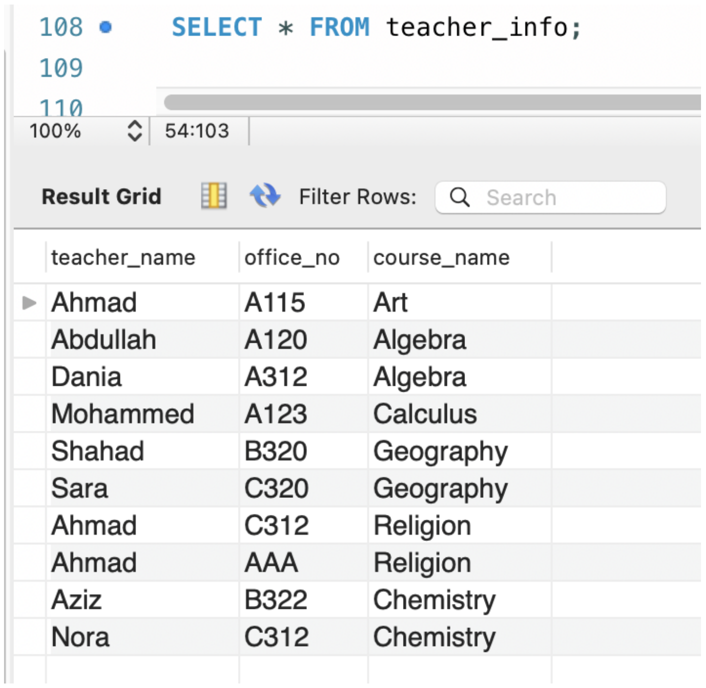
student_name	course_name
Nora	Art
Abrar	Art
Sara	Art
saud	Art
Mohammed	Art
Rashed	Algebra
Fatimah	Algebra
Saeed	Algebra
Khalid	Algebra
Amal	Algebra
Maryam	Algebra
Shahad	Calculus
Mohammed	Calculus
Saud	Calculus
Samar	Calculus
Rana	Calculus
Majid	Calculus
Rashed	Calculus
Maryam	Geography
Abdulaziz	Geography
Raghad	Geography
Asmaa	Geography
Dania	Geography
Sama	Religion

- قم بانشاء view باسم teacher_info يحتوي على اسم المعلم و رقم المكتب و اسم المادة التي يتم تدريسها.



```
/* a view that contains teachers names
    and their office number and courses they teach */


CREATE VIEW teacher_info
AS
SELECT teacher_name, office_no, Course.course_name
FROM Teacher
INNER JOIN Course ON Course.course_id = Teacher.course_id;

```

  - قم بعرض view



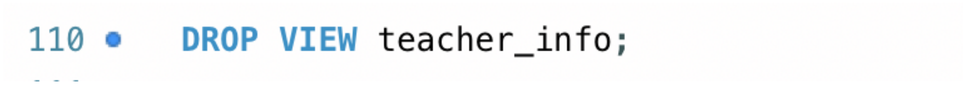
```
SELECT * FROM teacher_info;
```

100% 54:103

Result Grid | Filter Rows: Search

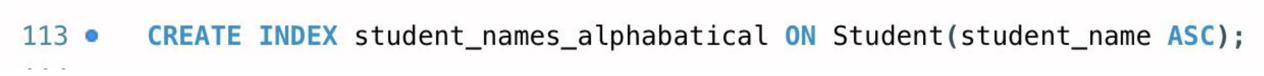
teacher_name	office_no	course_name
Ahmad	A115	Art
Abdullah	A120	Algebra
Dania	A312	Algebra
Mohammed	A123	Calculus
Shahad	B320	Geography
Sara	C320	Geography
Ahmad	C312	Religion
Ahmad	AAA	Religion
Aziz	B322	Chemistry
Nora	C312	Chemistry

  - قم بحذف view



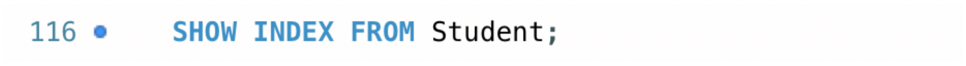
```
DROP VIEW teacher_info;
```

- قم بإنشاء index للبحث باستخدام اسماء الطلاب ابجدياً.



```sql
CREATE INDEX student_names_alphabatical ON Student(student_name ASC);
```

  - قم بعرض index



```sql
SHOW INDEX FROM Student;
```

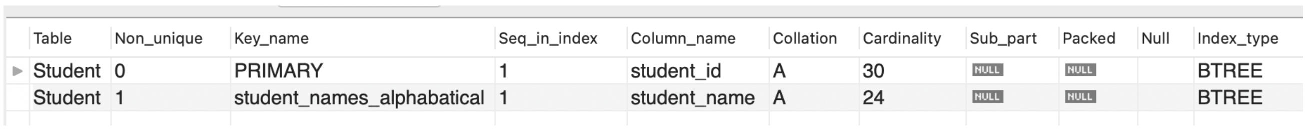
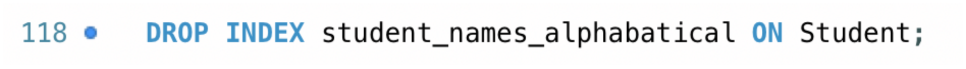


Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
Student	0	PRIMARY	1	student_id	A	30	NULL	NULL		BTREE
Student	1	student_names_alphabatical	1	student_name	A	24	NULL	NULL		BTREE

  - قم بحذف index



```sql
DROP INDEX student_names_alphabatical ON Student;
```

ملحق

Source code المستخدم في المشروع:

```
USE TamayozHighSchool;


/* a many-to-many relation between teachers and students
teacher: teaches many students
    student: taught by many teachers */

-- create junction table

CREATE TABLE Teachers_Students(
student_id        INT         NOT NULL,
teacher_id        INT         NOT NULL,
FOREIGN KEY (student_id) REFERENCES Student(student_id),
FOREIGN KEY (teacher_id) REFERENCES Teacher(teacher_id),
PRIMARY KEY (student_id, teacher_id)
);

-- map teachers to students

INSERT INTO Teachers_Students(student_id, teacher_id)
VALUES (1,2), (2, 5), (3, 9),
(1, 11), (2, 11), (3, 11),
(4, 7), (7, 7), (5, 10);


SELECT * FROM Teachers_Students;


/* a one-to-many relation teachers and courses
teacher: teaches one course
    course: taught by many teachers */

-- add a foreign key to the teacher table

ALTER TABLE Teacher
ADD COLUMN course_id    INT,
ADD FOREIGN KEY (course_id) REFERENCES Course(course_id);

-- update course id for each teacher

UPDATE Teacher SET course_id = 5 WHERE teacher_id = 2;
UPDATE Teacher SET course_id = 5 WHERE teacher_id = 3;
UPDATE Teacher SET course_id = 2 WHERE teacher_id = 4;
UPDATE Teacher SET course_id = 4 WHERE teacher_id = 5;
UPDATE Teacher SET course_id = 1 WHERE teacher_id = 6;
UPDATE Teacher SET course_id = 2 WHERE teacher_id = 7;
UPDATE Teacher SET course_id = 6 WHERE teacher_id = 8;
UPDATE Teacher SET course_id = 3 WHERE teacher_id = 9;
UPDATE Teacher SET course_id = 4 WHERE teacher_id = 10;
UPDATE Teacher SET course_id = 6 WHERE teacher_id = 11;

SELECT * FROM Teacher;


/* a many-to-many relation between students and courses */

-- create junction table

CREATE TABLE Student_Courses(
student_id        INT         NOT NULL,
course_id         INT         NOT NULL,
FOREIGN KEY (student_id) REFERENCES Student(student_id),
FOREIGN KEY (course_id) REFERENCES Course(course_id),
PRIMARY KEY (student_id, course_id)
);

-- map students to courses

INSERT INTO Student_Courses(student_id, course_id)
VALUES (1, 6), (2, 4), (3, 3),
(4, 3), (5, 4), (6, 2),
(7, 3), (8, 1), (9, 5),
(10, 5), (11, 2), (12, 3),
(13, 4), (14, 3), (15, 4),
(16, 1), (17, 1), (18, 3),
(19, 2), (20, 5), (21, 6),
(22, 2), (23, 3), (24, 2),
(25, 1), (26, 1), (27, 2),
(28, 4), (29, 5), (30, 6);


SELECT * FROM Student_Courses;


/* a stored procedure that displays student names
and the courses each student takes */

DELIMITER //
CREATE PROCEDURE student_info()
BEGIN
SELECT Student.student_name, Course.course_name
FROM Student_Courses
INNER JOIN Student ON Student.student_id = Student_Courses.student_id
INNER JOIN Course ON Course.course_id = Student_Courses.course_id;
END;


CALL student_info();
```

```
/* a view that contains teachers names
and their office number and courses they teach */

CREATE VIEW teacher_info
AS
SELECT teacher_name, office_no, Course.course_name
FROM Teacher
INNER JOIN Course ON Course.course_id = Teacher.course_id;

SELECT * FROM teacher_info;

DROP VIEW teacher_info;


/* create an index for student names */

CREATE INDEX student_names_alphabatical ON Student(student_name ASC);

SHOW INDEX FROM Student;

DROP INDEX student_names_alphabatical ON Student;
```