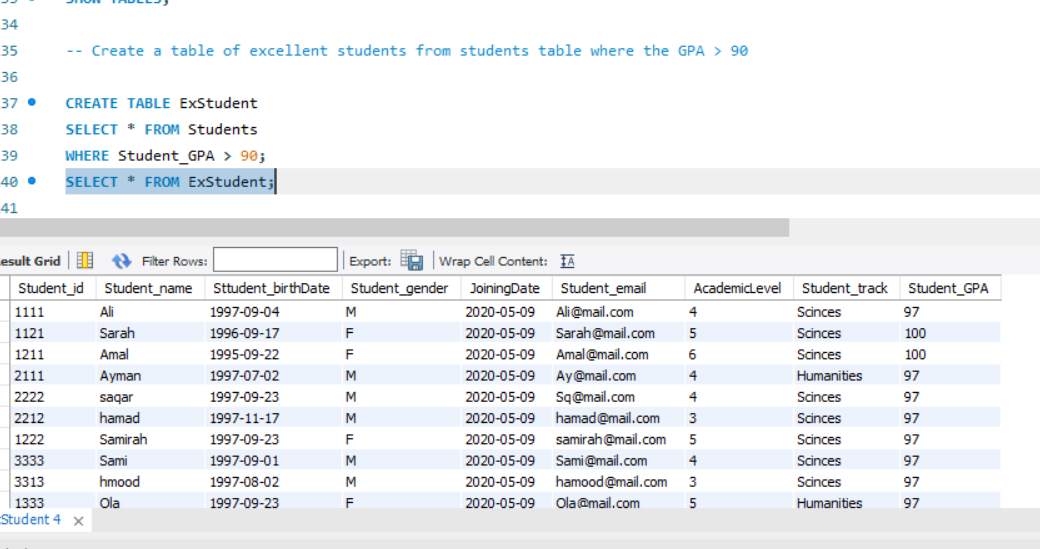
SQL 2 Project

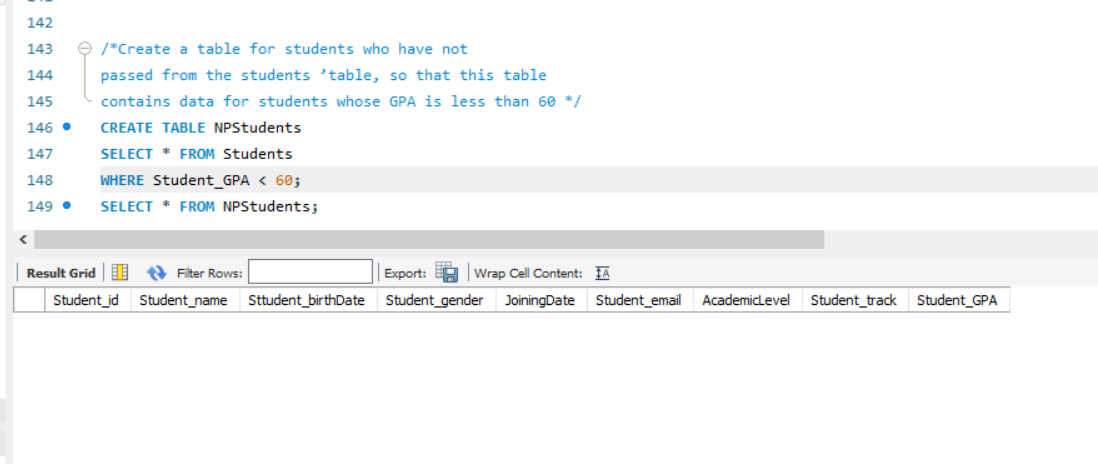
Samirah Abed Alhusayni

مشروع ٢

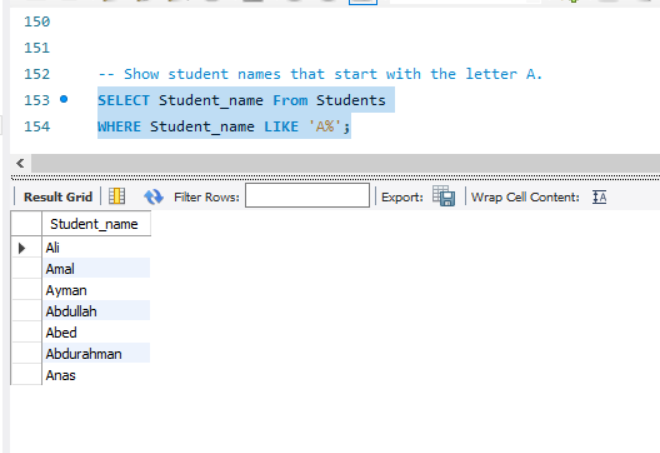
باستخدام ما تعلمته خلال هذهِ الدورة قم بتطبيق المتطلبات التالية، حيث أن هذا المشروع مكمّل لمشروع SQL المستوى الأول.

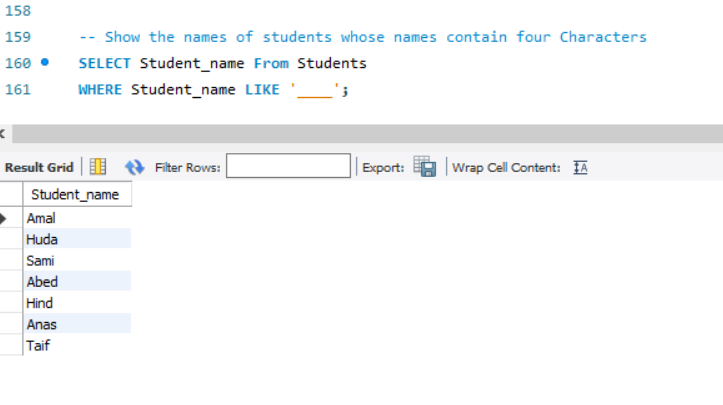
المتطلبات:

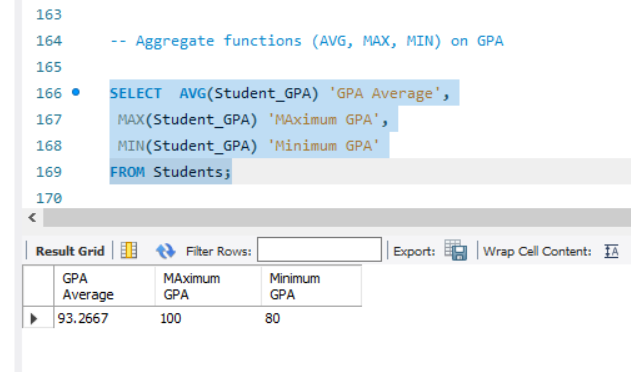
* إنشاء جدول للطلاب المتفوقين من جدول الطلاب، بحيث يحتوي هذا الجدول على بيانات الطلاب الذي يكون معدلهم التراكمي أعلى من ٩٠.
* إنشاء جدول للطلاب الغير مجتازين من جدول الطلاب، بحيث يحتوي هذا الجدول على بيانات الطلاب الذي يكون معدلهم التراكمي أقل من ٦٠.

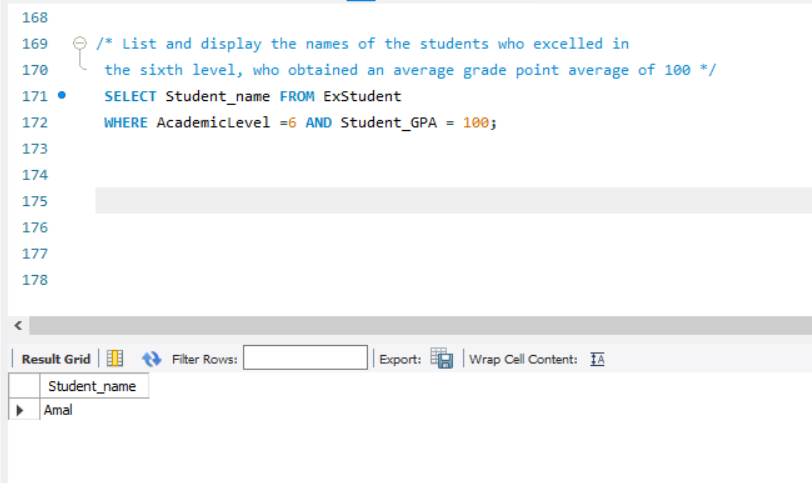
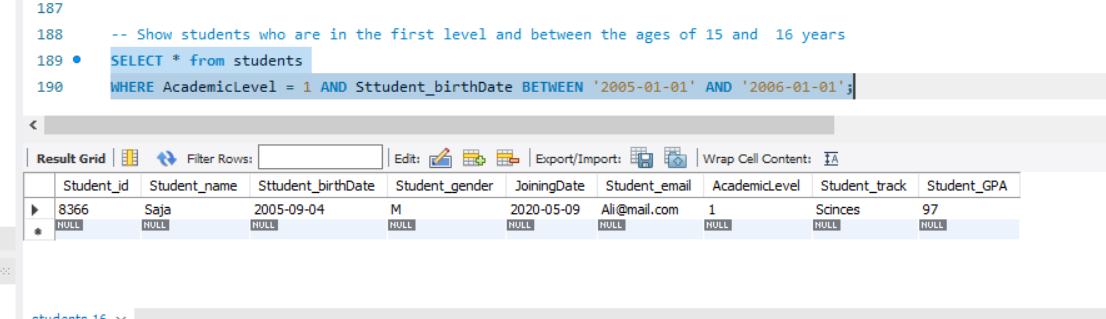
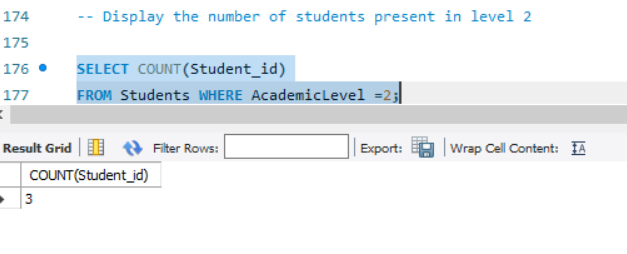
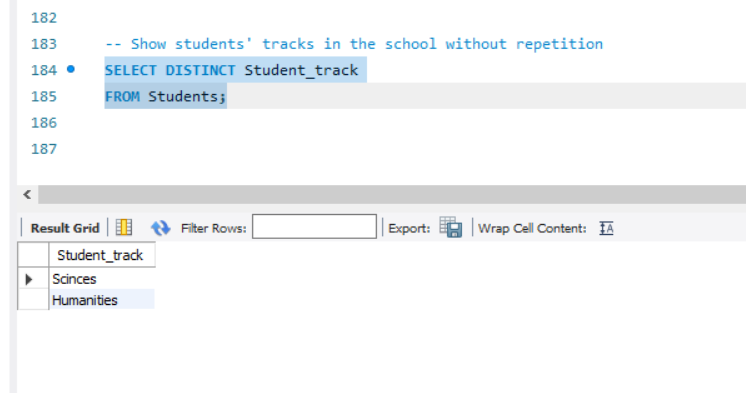
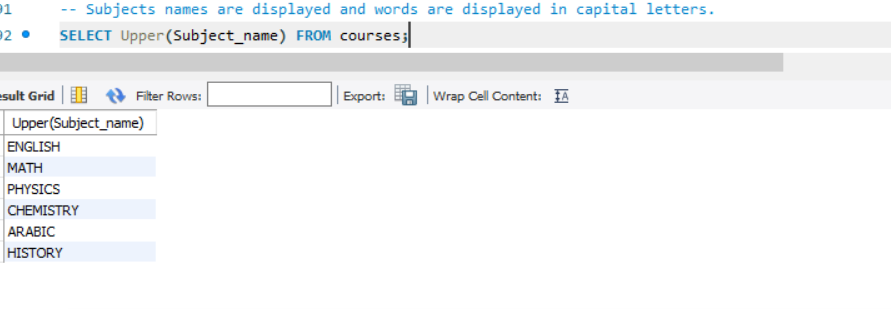
لانه لا يوجد في قاعدة البيانات الحالية طلاب أقل من 60 درجة

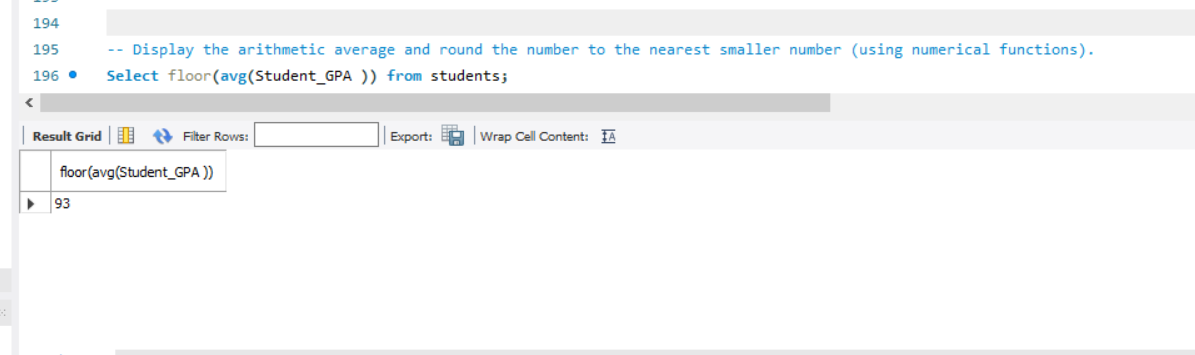
* عرض أسماء الطلاب التي تبدأ بحرف A.



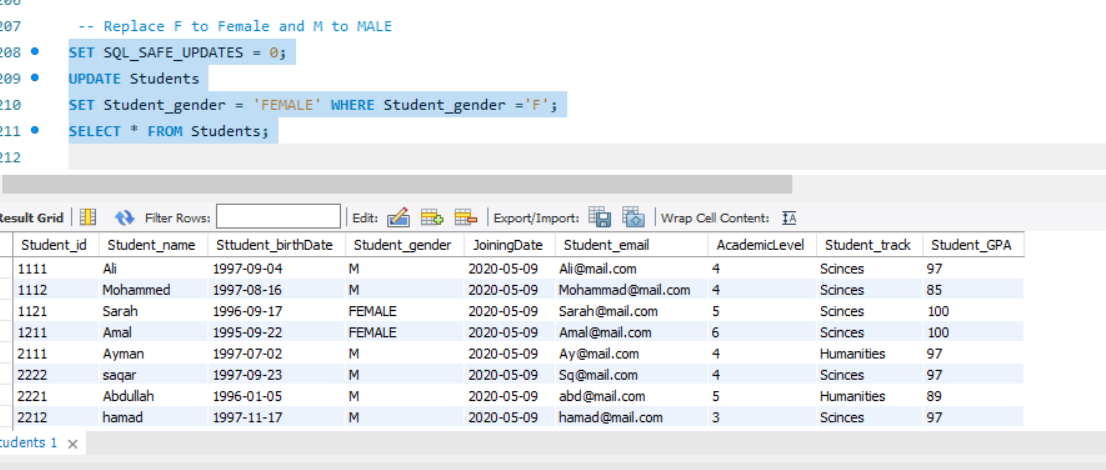
* عرض أسماء الطلاب التي تحتوي أسمائهم على أربع خانات.
* تطبيق AVG, MAX, MIN) Aggregate functions) على المعدل التراكمي للطالب مع إضافة تسمية واضحة للناتج.

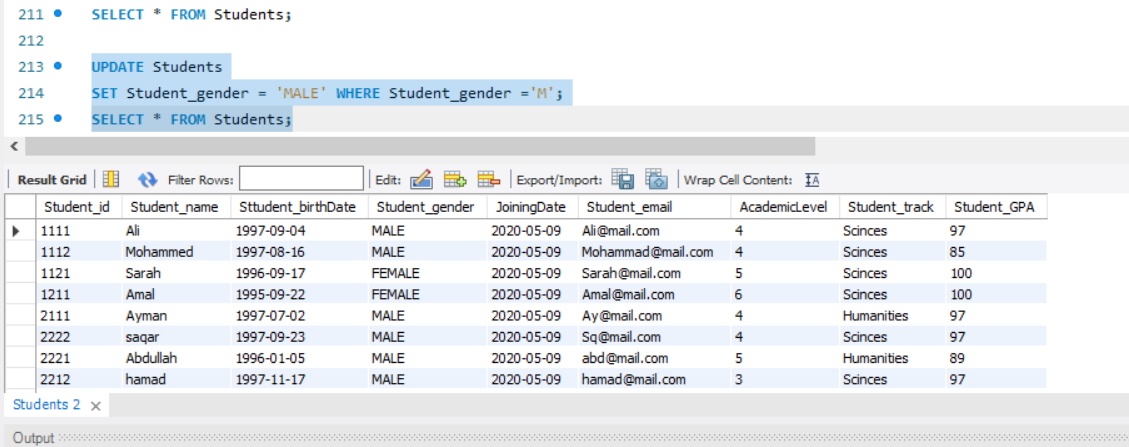


* حصر وعرض أسماء الطلاب المتفوقين في المستوى السادس الحاصلين على معدل تراكمي يساوي 100.
* عرض الطلاب اللذين في المستوى الأول وأعمارهم بين ١٥ و ١٦ سنة.
* عرض عدد الطلاب الموجودين بالمستوى ٢.
* استعراض مسارات الطلاب في المدرسة بدون تكرار.
* عرض أسماء المواد ويتم عرض الكلمات بالأحرف الكبيرة.
* عرض المتوسط الحسابي للمعدل التراكمي وقرب الرقم لأقرب أصغر عدد (باستخدام numeric functions)

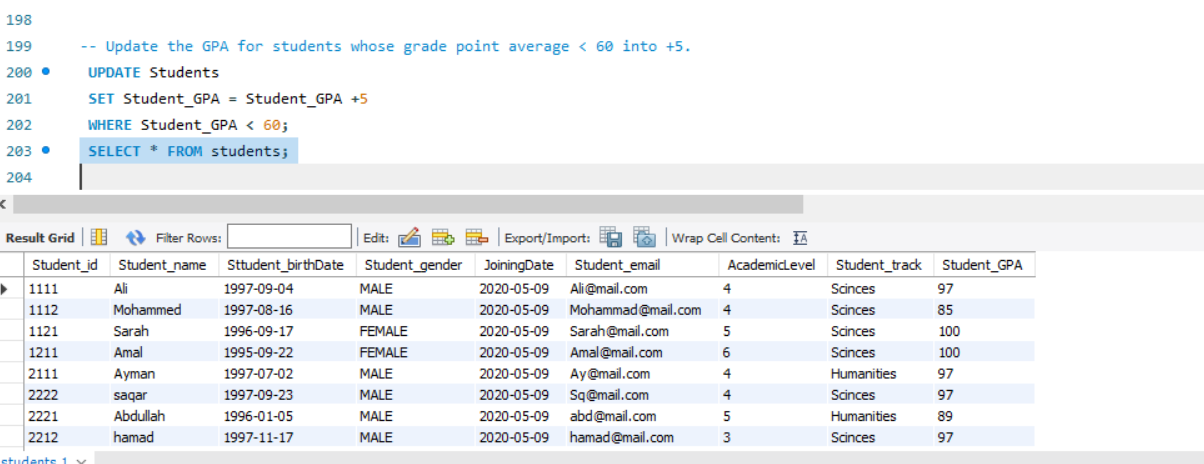
.

* تبديل جميع الصفوف من جدول الطلاب التي تحتوي على الجنس F إلى Female، و M إلى Male (باستخدام string functions).





* تحديث المعدل التراكمي للطلاب الذي معدلهم التراكمي أقل من ٦٠ وزيادة معدلهم بخمس درجات.



* قم بأخذ لقطة شاشة لكل مما سبق مع النتائج.

Source Code

-- SQL 2

-- Create a table of excellent students from students table where the GPA > 90

CREATE TABLE ExStudent

SELECT \* FROM Students

WHERE Student\_GPA > 90;

/\*Create a table for students who have not

passed from the students ’table, so that this table

contains data for students whose GPA is less than 60 \*/

CREATE TABLE NPStudents

SELECT \* FROM Students

WHERE Student\_GPA < 60;

-- Show student names that start with the letter A.

SELECT Student\_name From Students

WHERE Student\_name LIKE 'A%';

-- Show the names of students whose names contain four Characters

SELECT Student\_name From Students

WHERE Student\_name LIKE '\_\_\_\_';

-- Aggregate functions (AVG, MAX, MIN) on GPA

SELECT AVG (Student\_GPA) 'GPA Average',

max (Student\_GPA) 'MAximum GPA',

min (Student\_GPA) 'Minimum GPA'

FROM Students;

/\* List and display the names of the students who excelled in

 the sixth level, who obtained an average grade point average of 100 \*/

 SELECT Student\_name FROM ExStudent

 WHERE AcademicLevel =6 AND Student\_GPA = 100;

 -- Show students who are in the first level and between the ages of 15 and  16 years

SELECT \* from students

WHERE  AcademicLevel =1 AND  Sttudent\_birthDate BETWEEN '2005-01-01' AND '2005-01-01';

-- Display the number of students present in level 2

SELECT COUNT(Student\_id)

FROM Students WHERE AcademicLevel =2;

-- Show students' tracks in the school without repetition

SELECT DISTTNCT Student\_track

FROM Students;

-- Subjects names are displayed and words are displayed in capital letters.

SELECT Upper(Subject\_name) FROM courses;

-- Display the arithmetic average and round the number to the nearest smaller number (using numerical functions).

Select floor(avg(Student\_GPA )) from students;

 -- Replace F to Female and M to MALE

 SET SQL\_SAFE\_UPDATES = 0;

UPDATE Students

SET Student\_gender = 'FEMALE' WHERE Student\_gender ='F';

SELECT \* FROM Students;

UPDATE Students

SET Student\_gender = 'MALE' WHERE Student\_gender ='M';

SELECT \* FROM Students;

-- Update the GPA for student's GPA into GPA+5 if GPA < 60

UPDATE Students

SET Student\_GPA = Student\_GPA + 5

WHERE Student\_GPA < 60;