**School management database design**

1)ER diagram analysis:

a)identify all the primary keys and foreign in the schema

\*the primary keys:

|  |  |
| --- | --- |
| entities | Primary key |
| classroom | classroom\_id |
| grade | grade\_id |
| course | course\_id |
| student | student\_id |
| parent | parent\_id |
| teacher | teacher\_id |
| |  | | --- | | classroom\_student |  |  | | --- | |  | | classroom\_id, student\_id |
| attendance | date, student\_id |
| exam\_type | exam\_type\_id |
| exam | exam\_id |
| exam\_result | exam\_id, student\_id, course\_id |

\*The foreign key:

|  |  |  |  |
| --- | --- | --- | --- |
| table | foreign Key | references Table | references Column |
| classroom | grade\_id | grade | grade\_id |
| classroom | teacher\_id | teacher | teacher\_id |
| course | grade\_id | grade | grade\_id |
| student | parent\_id | parent | parent\_id |
| classroom\_student | classroom\_id | classroom | classroom\_id |
| classroom\_student | student\_id | student | student\_id |
| attendance | student\_id | student | student\_id |
| exam | exam\_type\_id | exam\_type | exam\_type\_id |
| exam\_result | |  | | --- | | exam\_id |  |  | | --- | |  | | exam | exam\_id |
| exam\_result | student\_id | student | student\_id |
| exam\_result | course\_id | course | course\_id |

b)review the provided ER digram :

\*table overview:

|  |  |
| --- | --- |
| **table Name** | **Description** |
| classroom | Stores information about classrooms, including grade, section, teacher, and additional remarks. |
| grade | Stores details of grades, such as grade name and description. |
| course | Contains information about courses and their associated grades. |
| student | Holds data about students, including personal details, parent relationships, and status |
| parent | Represents parents or guardians and their contact details. |
| teacher | Stores information about teachers, including their personal and contact details. |
| classroom\_student | Links students to classrooms to manage many-to-many relationships. |
| attendance | Tracks student attendance with dates, statuses, and remarks. |
| exam\_type | Defines types of exams (e.g., quiz, midterm, final). |
| exam | Stores information about exams, including type and start date. |
| exam\_result | |  | | --- | |  |  |  | | --- | | Records exam results for students in specific courses. | |

3) List the relationships (one-to-one, one-to-many, many-to-many) between entities and explain .

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| table 1 | table 2 | relationship Type | |  | | --- | | Foreign Key in Table 1 |  |  | | --- | |  | | |  | | --- | | References Column in Table 2 |  |  | | --- | |  | | | Description | | --- |  |  | | --- | |  | |
| classroom | grade | |  | | --- | | One-to-Many |  |  | | --- | |  | | grade\_id | grade\_id | |  | | --- | | Each classroom belongs to one grade, but one grade can have many classrooms. |  |  | | --- | |  | |
| classroom | teacher | One-to-One | teacher\_id | teacher\_id | |  | | --- | | Each classroom is assigned to one teacher, and each teacher manages one class. |  |  | | --- | |  | |
| course | grade | |  | | --- | | One-to-Many |  |  | | --- | |  | | grade\_id | grade\_id | |  | | --- | | Each course is assigned to one grade, but one grade can have many courses. |  |  | | --- | |  |  |  | | --- | |  | |
| student | |  | | --- | | parent |  |  | | --- | |  | | |  | | --- | | One-to-Many |  |  | | --- | |  | | parent\_id | parent\_id | |  | | --- | | Each student has one parent, but one parent can have multiple students. |  |  | | --- | |  | |
| classroom\_student | classroom | |  | | --- | | Many-to-One |  |  | | --- | |  | | classroom\_id | classroom\_id | |  | | --- | | Links multiple students to a single classroom. |  |  | | --- | |  | |
| classroom\_student | student | |  | | --- | | Many-to-One |  |  | | --- | |  | | student\_id | student\_id | |  | | --- | | Links multiple classrooms to a single student. |  |  | | --- | |  | |
| attendance | student | |  | | --- | | Many-to-One |  |  | | --- | |  | | student\_id | student\_id | |  | | --- | | Tracks attendance for each student in the system. |  |  | | --- | |  | |
| exam | exam\_type | |  | | --- | | One-to-Many |  |  | | --- | |  | | exam\_type\_id | exam\_type\_id | |  | | --- | | Each exam is categorized under a specific exam type. |  |  | | --- | |  | |
| exam\_result | exam | |  | | --- | | Many-to-One |  |  | | --- | |  | | exam\_id | exam\_id | |  | | --- | | Tracks results for multiple students in one exam. |  |  | | --- | |  | |
| exam\_result | student | |  | | --- | | Many-to-One |  |  | | --- | |  | | student\_id | student\_id | |  | | --- | | Tracks results for each student in various exams. |  |  | | --- | |  | |
| exam\_result | course | |  | | --- | | Many-to-One |  |  | | --- | |  | | course\_id | course\_id | |  | | --- | |  |  |  | | --- | |  |   Tracks exam results for specific courses. |

2)sql table relation :

**The file attached, named 'DDL', contains the tables, relationships between the tables, constraints, and notes. The answer is included in it."(في الملف المرفق )**

File extension:DDl.sql :

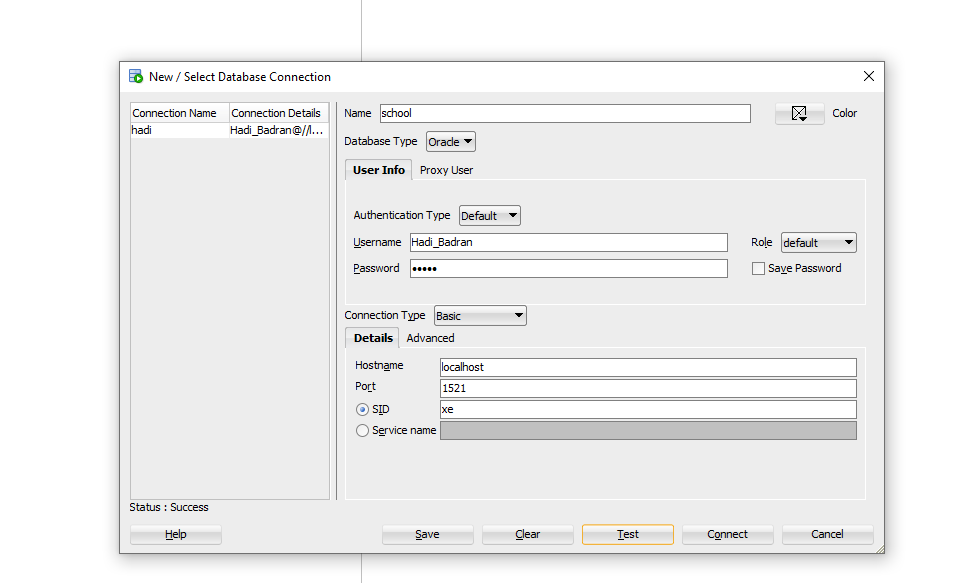
**Important note:**

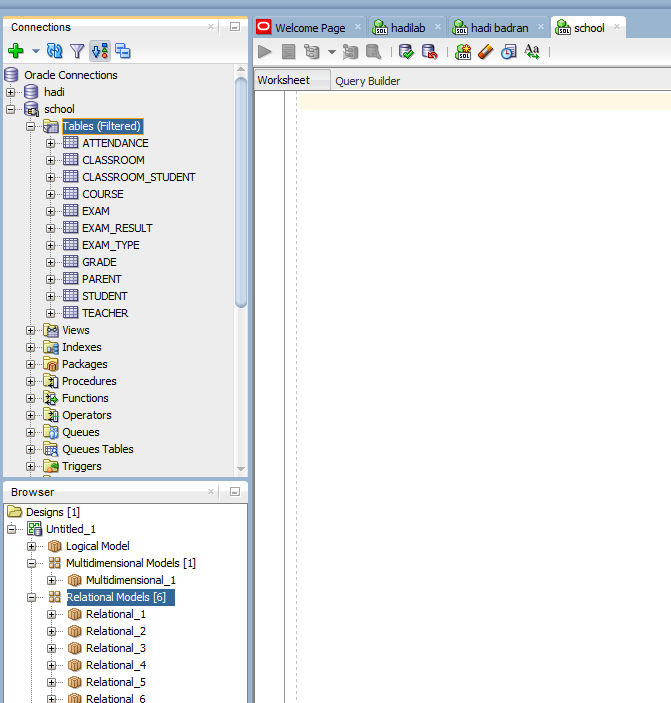
To ensure the accuracy of the relationships and tables, I opened Oracle SQL Developer, then connected it with the Oracle 10g database. I entered the username and password for the database I created, then linked the two programs. After that, I established the relationships between the tables using Oracle SQL, as Oracle 10g does not support displaying the ER diagram. The following images show how I linked the two programs and created the relationships in the database I created.

"From the **View** menu, go to **Data Modeler**, then select **Browsers**, and finally click on **new relational model**:

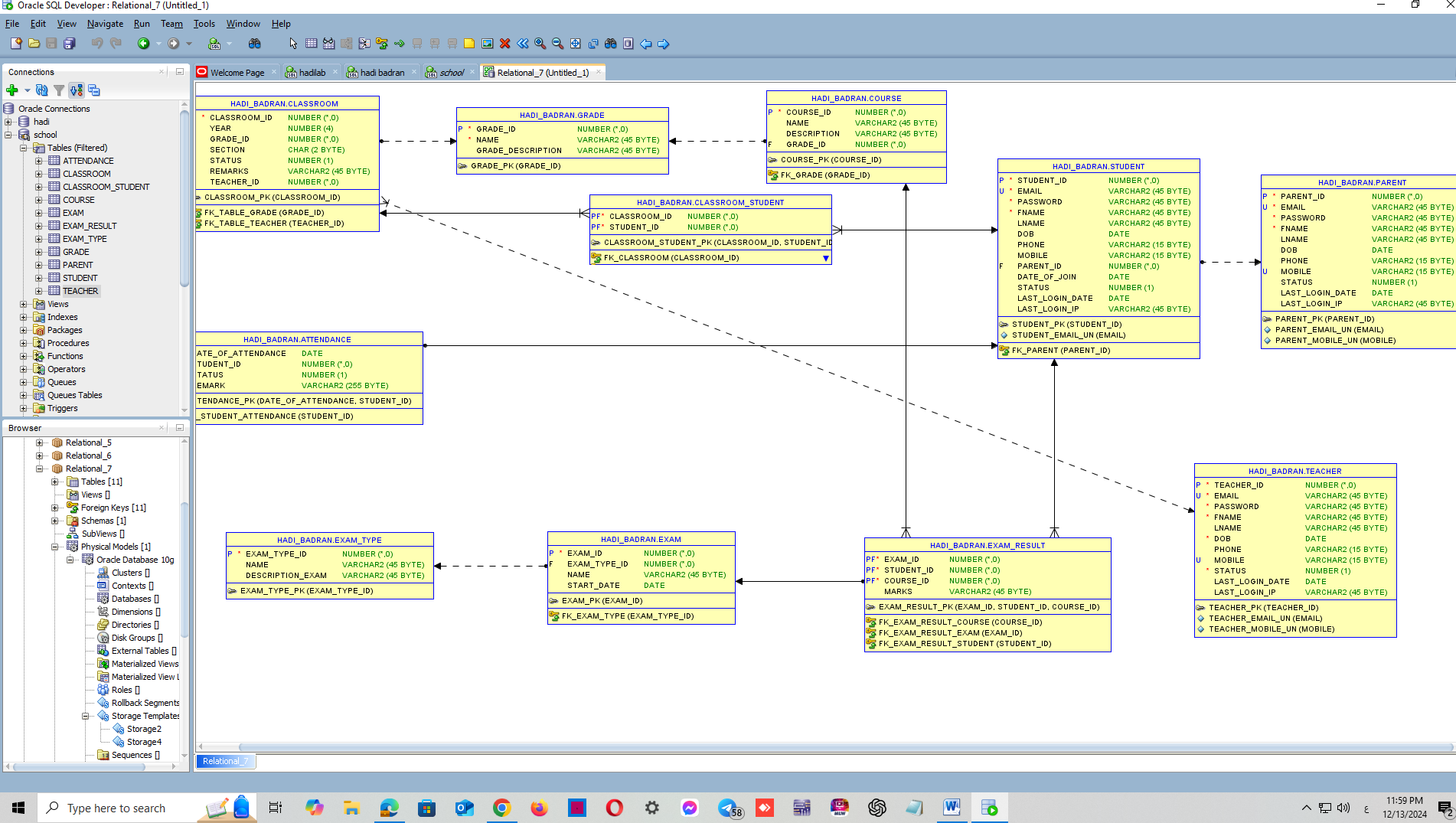
باختصار عملت ربط بين البرنامجين للتاكد من صحه العلاقات والجداول كما في المخطط المعطى

Case connect:





Case ER:-

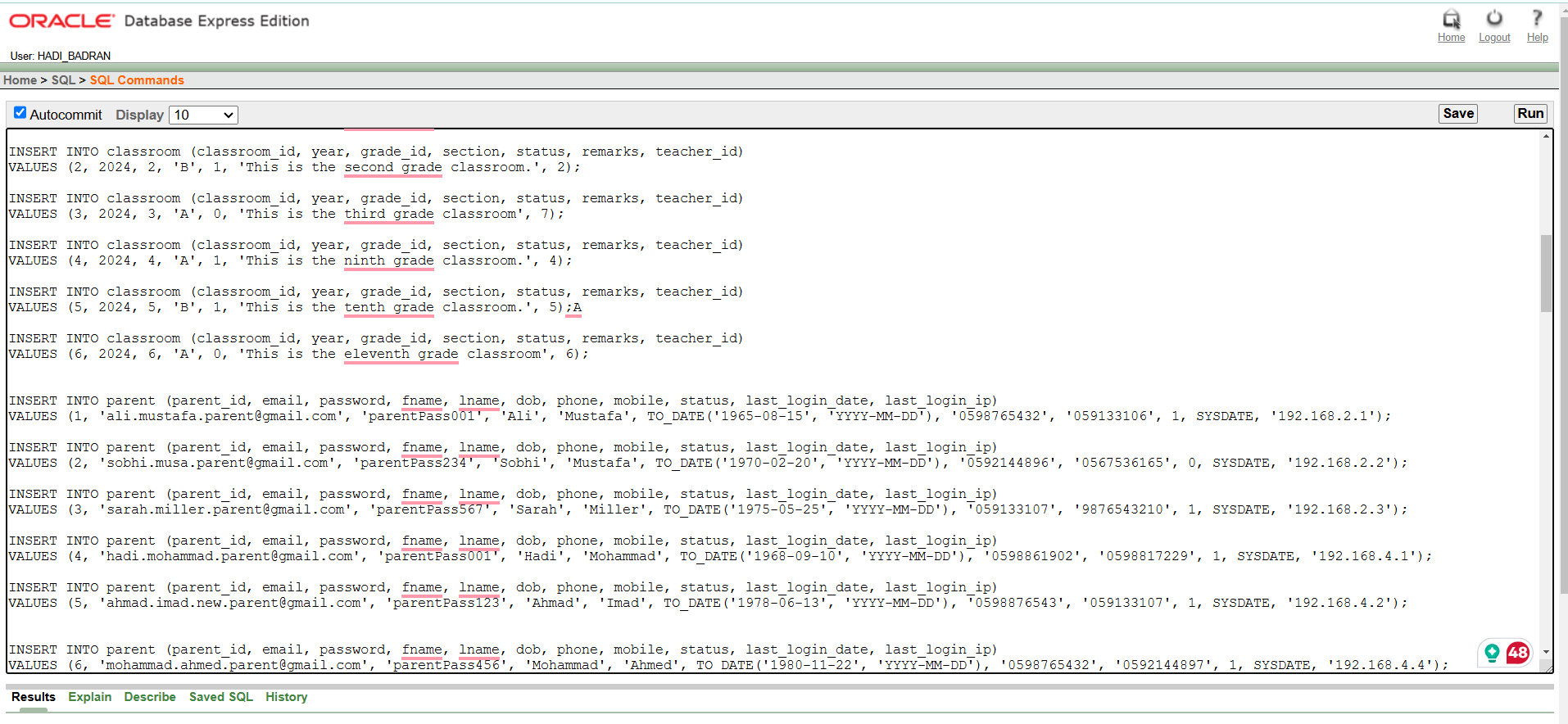


3) Insert sample data for the following tables:

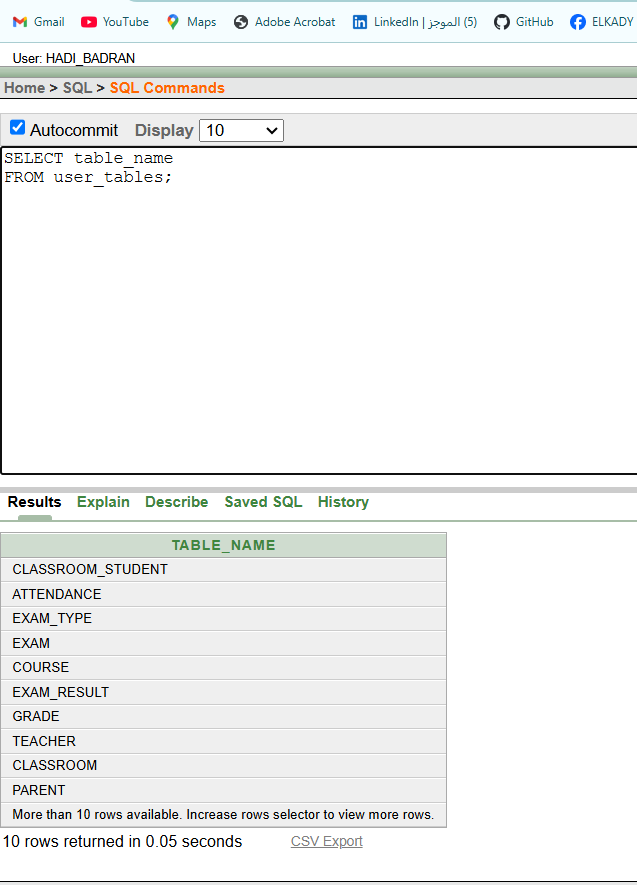
In the attached file:



How to enter data into the program:



Regarding the tables:



4. QUERY WRITING:

the attached file



1. Retrieve a list of all students in a specific classroom along with their grade:

بدو نرجع كل الطلاب الي الهم فصل دراسي معين مع العلامات

Sol:

**SELECT**

**classroom.classroom\_id AS classroom\_id,**

**student.fname || ' ' || student.lname AS student\_name,**

**grade.name AS grade\_name,**

**grade.grade\_description AS grade\_desc**

**FROM**

**student**

**JOIN**

**classroom\_student ON student.student\_id = classroom\_student.student\_id**

**JOIN**

**classroom ON classroom\_student.classroom\_id = classroom.classroom\_id**

**JOIN**

**grade ON classroom.grade\_id = grade.grade\_id**

**WHERE**

**classroom.classroom\_id = 3**

**ORDER BY**

**classroom.classroom\_id, student.lname;**

**ممكن اعمل اختصار لاسماء الجداول :**

**SELECT**

**c.classroom\_id AS classroom\_id,**

**s.fname || ' ' || s.lname AS student\_name,**

**g.name AS grade\_name,**

**g.grade\_description AS grade\_desc**

**FROM**

**student s**

**JOIN**

**classroom\_student cs ON s.student\_id = cs.student\_id**

**JOIN**

**classroom c ON cs.classroom\_id = c.classroom\_id**

**JOIN**

**grade g ON c.grade\_id = g.grade\_id**

**WHERE**

**c.classroom\_id = 3**

**ORDER BY**

**c.classroom\_id, s.lname;**

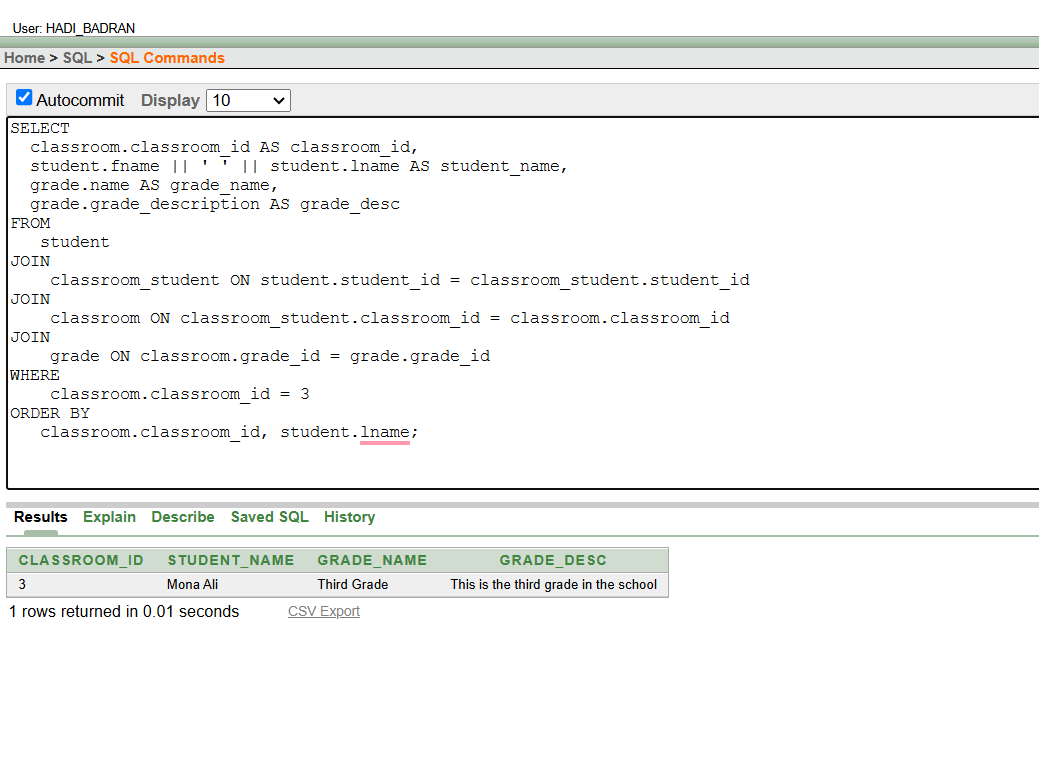
The student's name, grade, and classroom are selected by joining the tables through keys and relationships, starting with the student table and classroom\_student table, then the classroom table, and finally the grade table, with the results ordered by classroom and student name.

يتم اختيار الأعمدة المطلوبة والتي تشمل **اسم الطالب، علامته، والفصل الدراسي:**

1) يتم ربط **جدول الطالب** مع **جدول الفصل الدراسي - الطالب** باستخدام المفتاح الأجنبي student\_id

2) ثم يتم ربط **جدول الفصل الدراسي - الطالب** مع **جدول الفصل الدراسي** باستخدام المفتاحclassroom\_id

3) يتم ربط **جدول الفصل الدراسي** مع **جدول العلامة** من خلال المفتاح grade\_id



**ملاحظه:ممكن اخلي المستخدم يدخل الفصل الدراسي المعين عن طريق user prompot باضافه شرط:**

WHERE classroom.classroom\_id =' &classroom\_id';

**ممكن عمل الاستعلام عن طريق where من دون join:**

**SELECT**

**classroom.classroom\_id AS classroom\_id,**

**student.fname || ' ' || student.lname AS student\_name,**

**grade.name AS grade\_name,**

**grade.grade\_description AS grade\_desc**

**FROM**

**student,**

**classroom\_student,**

**classroom,**

**grade**

**WHERE**

**student.student\_id = classroom\_student.student\_id**

**AND classroom\_student.classroom\_id = classroom.classroom\_id**

**AND classroom.grade\_id = grade.grade\_id**

**AND classroom.classroom\_id = 3**

**ORDER BY**

**classroom.classroom\_id, student.lname;**

2)Fetch the details of all teachers teaching a specific course.يطلب ارجاع بيانات المعلمين في كورس معين

**SELECT**

**t.\*,**

**co.course\_id**

**FROM**

**teacher t**

**JOIN**

**classroom c ON t.teacher\_id = c.teacher\_id**

**JOIN**

**course co ON c.grade\_id = co.grade\_id**

**WHERE**

**co.course\_id = 4**

**ORDER BY**

**co.course\_id, t.teacher\_id;**

****I retrieved the teacher's data along with the course ID and course name to identify which course each teacher is associated with. The tables were joined using foreign keys, linking the teacher table with the classroom table, and then linking the classroom table with the course table using the grade\_id relationship.

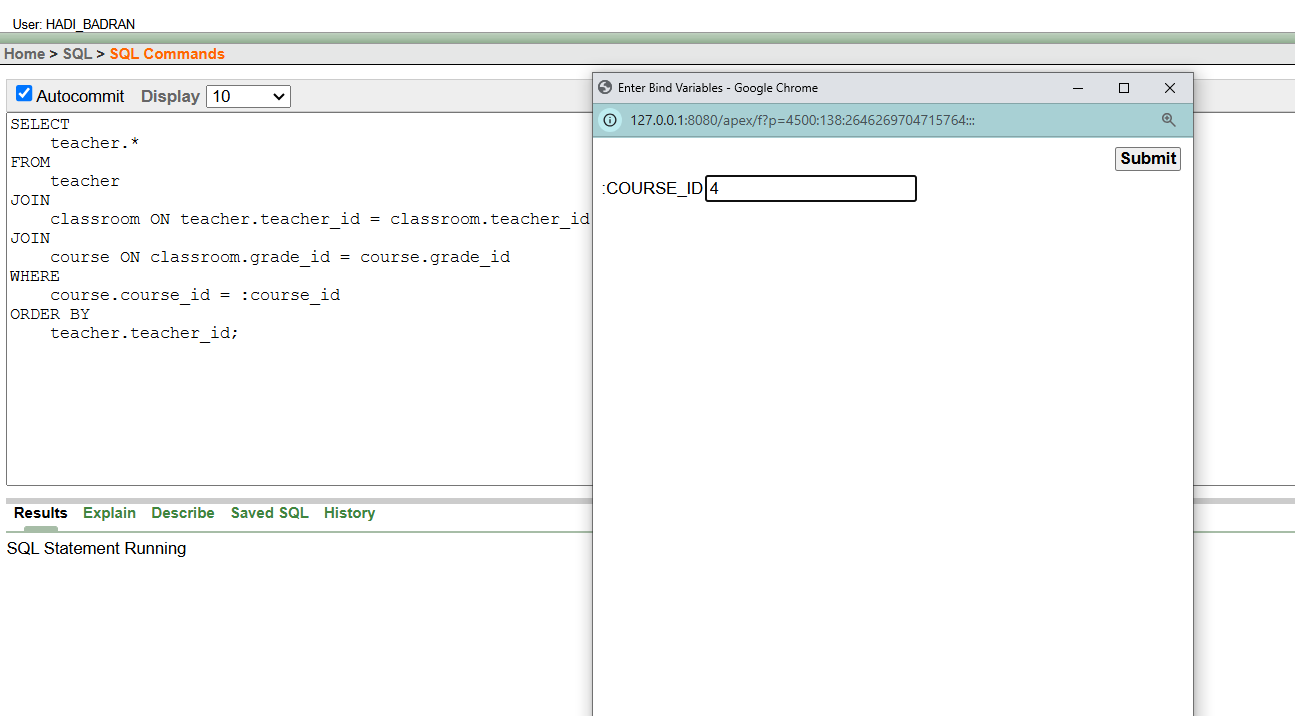
أخذت بيانات المعلم مع معرف الدورة (course ID) واسم الدورة (course name) لمعرفة المعلم المرتبط بكل دورة. تم ربط الجداول باستخدام المفاتيح، حيث تم ربط جدول المعلم مع جدول الفصل الدراسي، ومن ثم ربط جدول الفصل الدراسي مع جدول الدورة باستخدام العلاقة (grade\_id)..

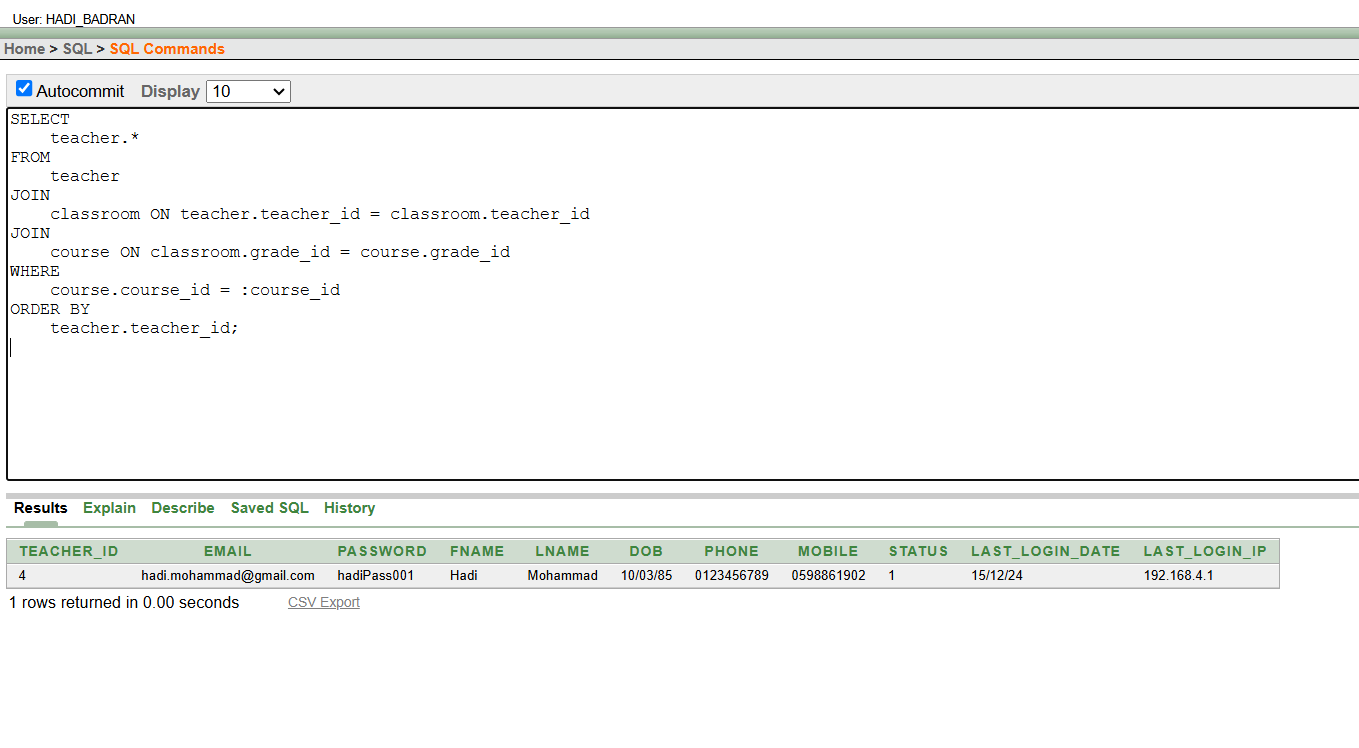
ملاحظه:في حال نريد بيانات المعلم لكورس معين يدخلها المستخدم نعمل شرط:

في اوراكل where course.course\_id = :course\_id

وفي الشاشه السوداء :where course.course\_id = '&course\_id'

في حال بدي بيانات المعلمين لكل الكورسات بعمل على ازاله الشرط..............................................

في حال عن طريق user prompot 



**3)Find all exams taken by a student along with their marks.:** استرجع جميع الامتحانات لكل طالب مع العلامه

**SELECT**

**exam.name AS exam\_name,**

**exam\_type.name AS exam\_type\_name,**

**exam\_result.marks**

**FROM**

**student**

**JOIN**

**exam\_result ON student.student\_id = exam\_result.student\_id**

**JOIN**

**exam ON exam\_result.exam\_id = exam.exam\_id**

**JOIN**

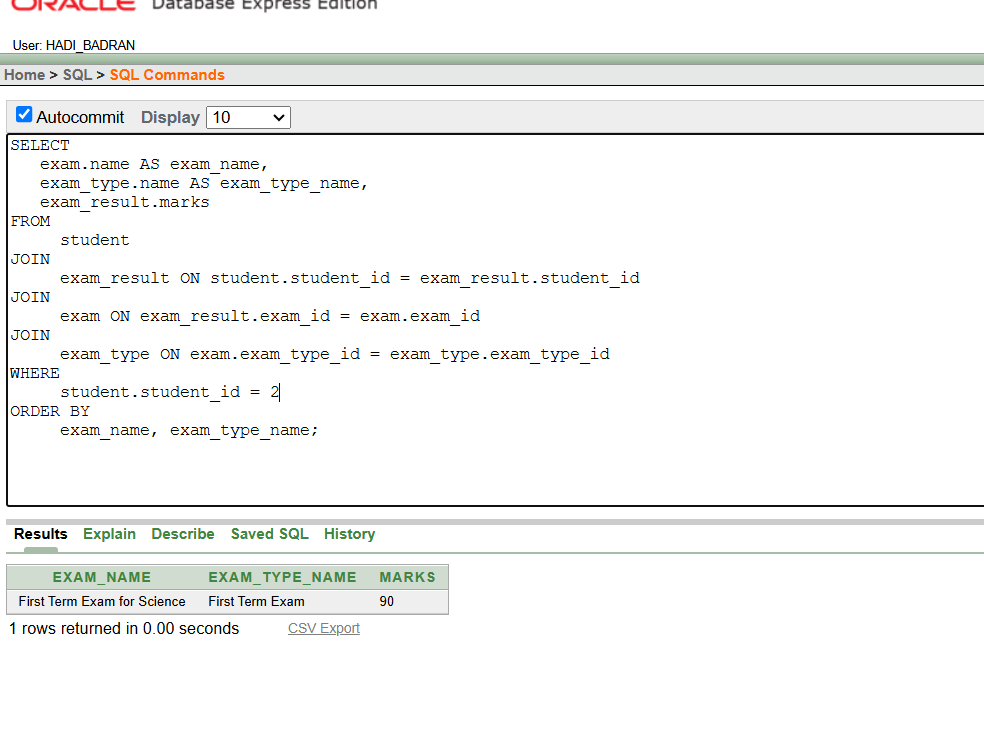
**exam\_type ON exam.exam\_type\_id = exam\_type.exam\_type\_id**

**WHERE**

**student.student\_id = 2**

**ORDER BY**

**exam\_name, exam\_type\_name;**

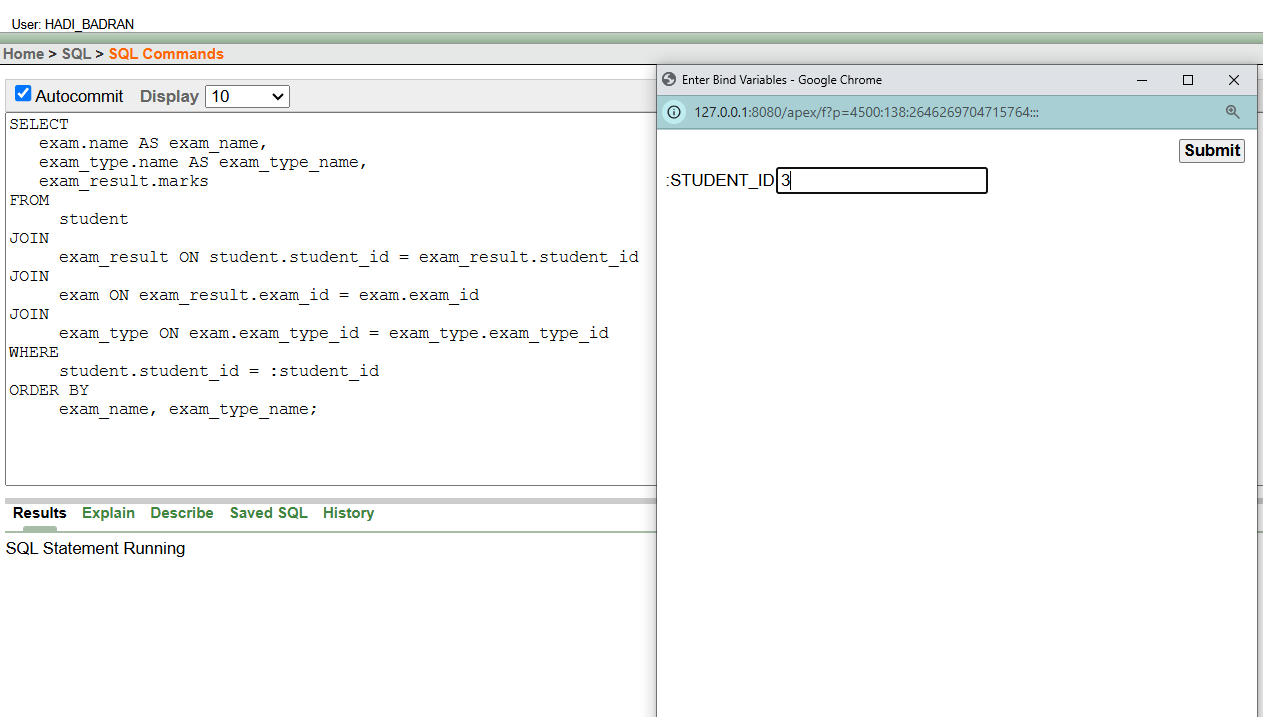
****

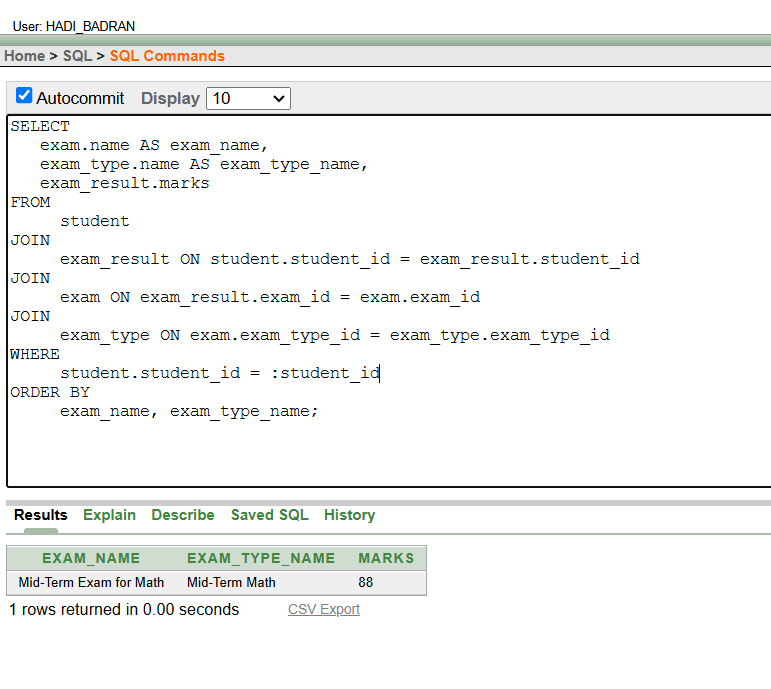
في هذا الحل، قمنا بجلب عمود العلامة مع اسم الطالب واسم الامتحان ونوعه(زياده للفهم ما نوع الامتحان).  
تم الربط بين **جدول الطالب** و**جدول exam\_result** باستخدام المفتاح الأساسي **student\_id** لجلب نتائج الامتحانات، ثم تم الربط مع **جدول exam** باستخدام **exam\_id** لجلب تفاصيل الامتحانات.  
بهذا الشكل، حصلنا على جميع الامتحانات التي أداها الطالب مع العلامات التي حصل عليها.

In this solution, we selected the marks column along with the student's name and the exam ,  
We joined the **student** table with the **exam\_result** table using the primary key **student\_id** to retrieve the exam results, and then joined it with the **exam** table using **exam\_id** to get the exam details.  
This way, we retrieved all the exams taken by the student along with their marks.

\*\*ملاحظه من الممكن ايضا عمل شرط لاسترجاع امتحانات طالب معين فقط : WHERE student.student\_id =' &student\_id'

او من خلال :where student.student\_id =:student\_id





الادخال من قبل المستخدم يكون نفس الشرط (نفس الطريقه) لكل الجمل لكن بتغيرر المطلوب .

يمكن حل السوال ايضا بدون join:

SELECT

exam.name AS exam\_name,

exam\_type.name AS exam\_type\_name,

exam\_result.marks

FROM

student,

exam\_result,

exam,

exam\_type

WHERE

student.student\_id = exam\_result.student\_id

AND exam\_result.exam\_id = exam.exam\_id

AND exam.exam\_type\_id = exam\_type.exam\_type\_id

AND student.student\_id = 2

ORDER BY

exam\_name, exam\_type\_name;

4) . Retrieve the attendance records for a specific student within a given date range.

السوال يطلب سجلات الحضور لطالب معين بناءا على student\_id ضمن نطاق زمني محدد date\_of\_attendance

النظاق الزمني :المعنى تاريخ البدايه والنهايه

الشرح:

ربطنا جدول student مع attendance باستخدام student\_id، حيث يمثل كل سجل في جدول attendance حضورًا لطالب معين ثم اخذنا الاعمده المطلوبه مثل date of attendace :لعرض سجلات الحضور ,status:لعرض حاله الحضور,remark :لعرض الملاحظات المتعلقه بحضور الطالب .

SELECT

attendance.date\_of\_attendance,

attendance.status,

attendance.remark

FROM

attendance

WHERE

attendance.student\_id = 6

AND attendance.date\_of\_attendance

BETWEEN TO\_DATE('2024-1-01', 'YYYY-MM-DD')

AND TO\_DATE('2024-01-5', 'YYYY-MM-DD')

ORDER BY

attendance.date\_of\_attendance;

يمكن استخدام :join

SELECT

attendance.date\_of\_attendance,

attendance.status,

attendance.remark

FROM

attendance

JOIN

student ON attendance.student\_id = student.student\_id

WHERE

student.student\_id = 6

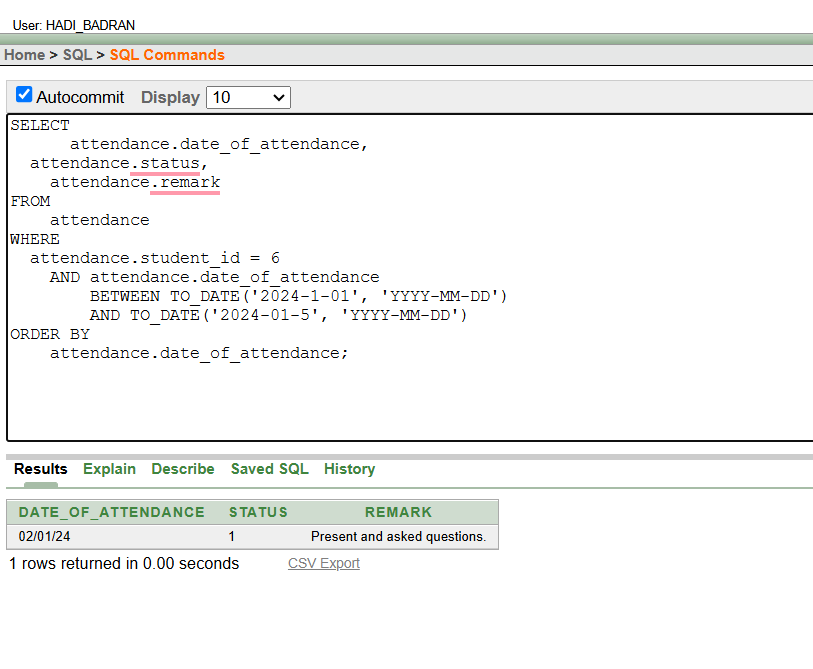
AND attendance.date\_of\_attendance

BETWEEN TO\_DATE('2024-01-01', 'YYYY-MM-DD')

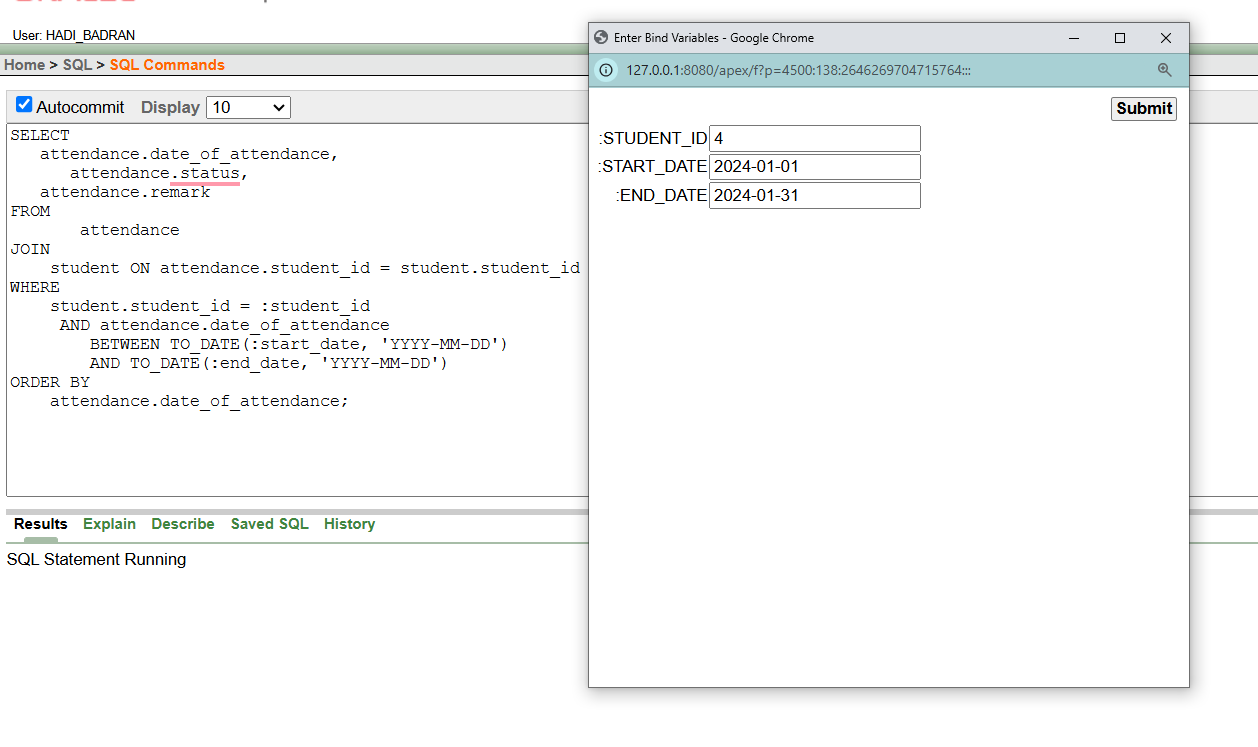
AND TO\_DATE('2024-01-31', 'YYYY-MM-DD')

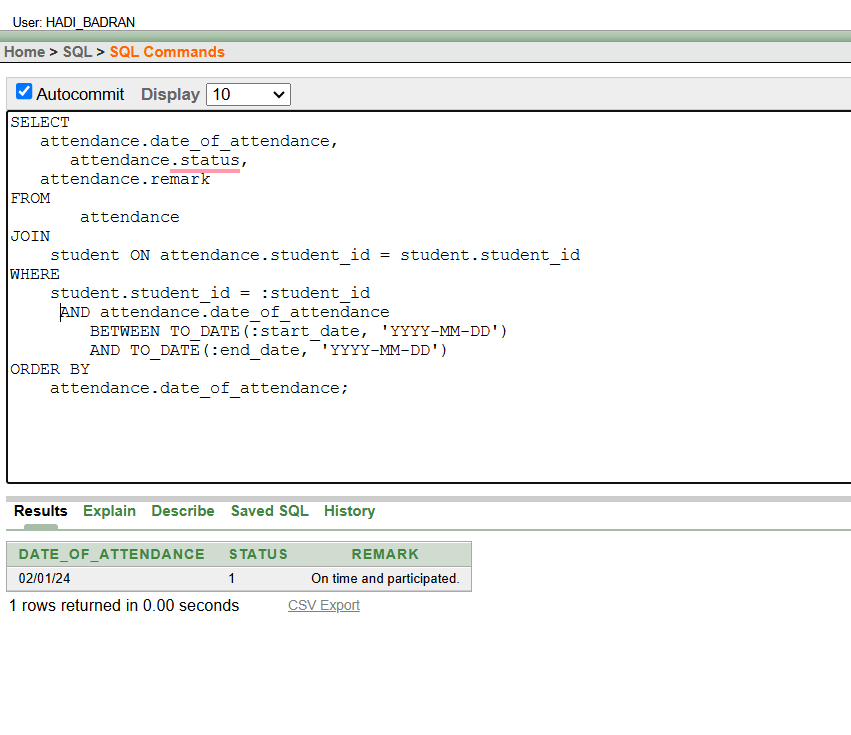
ORDER BY

attendance.date\_of\_attendance;

ومن خلالuserنضع الشرط:

WHERE student.student\_id = :student\_id and attendance.date\_of\_attendance BETWEEN TO\_DATE(:start\_date, 'YYYY-MM-DD') AND TO\_DATE(:end\_date, 'YYYY-MM-DD')





5) List the parents' contact information for all students in a particular grade.

في الاستعلام يريد ارجاع معلومات الاتصال ل parent لجميع الطلاب الي موجودين في صف معين .

نختار معلومات الاتصال ل parent من جدول parent مثل phone,email,name,mobile ثم للحصول على صف معين نربط الجداول باستخدام المفاتيح الاجنبيه :

student.parent\_id = parent.parent\_id: لربط كل طالب بولي أمره.

classroom\_student.student\_id = student.student\_id: لربط الطلاب بالفصول الدراسية.

classroom.classroom\_id = classroom\_student.classroom\_id: لربط الطلاب بالفصول التي ينتمون

grade.grade\_id = classroom.grade\_id: لربط الفصول الدراسية بالصفوف الدراسية.

لتصفية الطلاب وأولياء أمورهم الذين ينتمون إلى صف دراسي معين باستخدام grade\_id.

select

parent.fname || ' ' || parent.lname as parent\_name,

parent.email,

parent.phone,

parent.mobile,

grade.name as grade\_name

from

parent

join

student on parent.parent\_id = student.parent\_id

join

classroom\_student on student.student\_id = classroom\_student.student\_id

join

classroom on classroom\_student.classroom\_id = classroom.classroom\_id

join

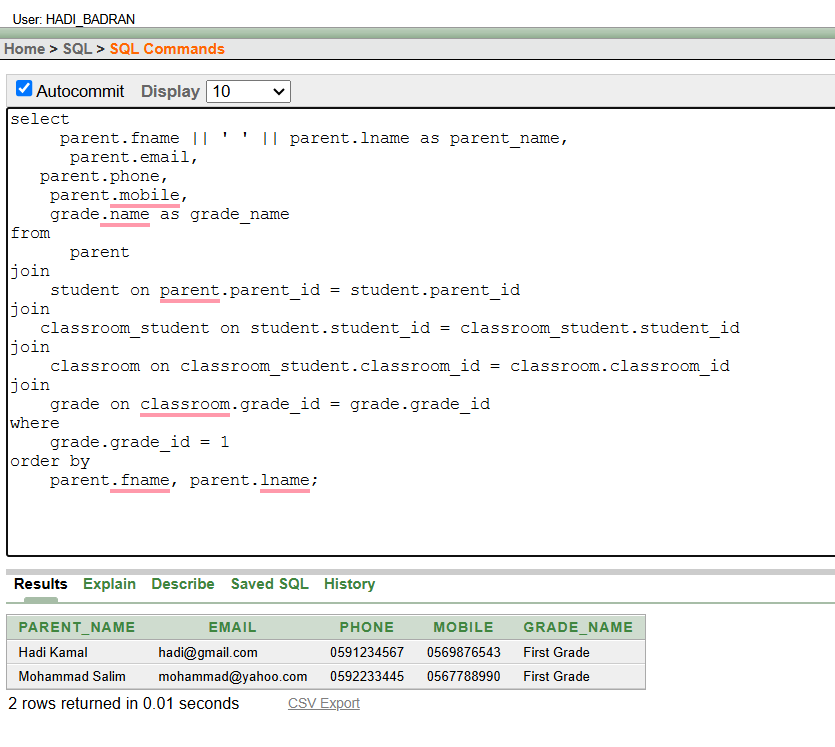
grade on classroom.grade\_id = grade.grade\_id

where

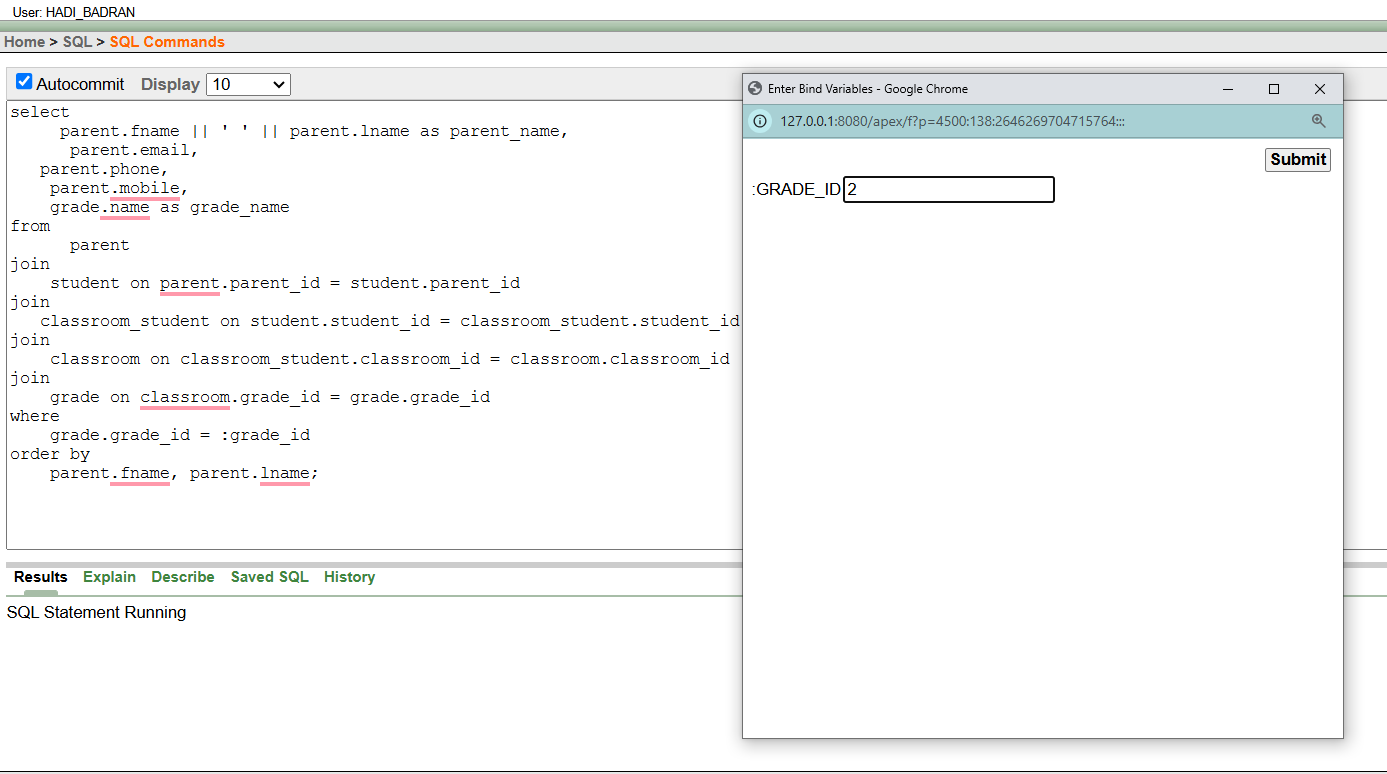
grade.grade\_id = 1

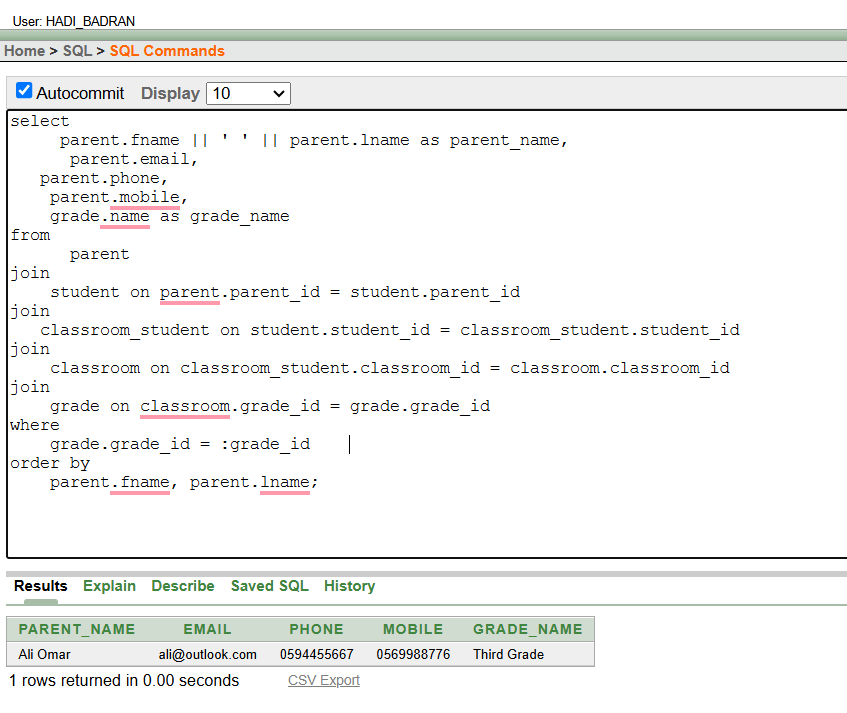
order by

parent.fname, parent.lname;



او من خلال شرط:where grade\_id=:grade\_id





6) Fetch the average marks for each exam type for a specific course

المطلوب حساب متوسط العلامات لكل نوع امتحان لكورس معين .

رابطنا بين الجداول عبر exam\_result و exam باستخدام exam\_id، ثم ربطنا exam بـ exam\_type باستخدام exam\_type\_id.

استخدمنا course\_id في WHERE لتصفية البيانات حسب المادة المحددة.

قمنا بحساب متوسط العلامات لكل نوع امتحان باستخدام AVG() بعد تحويل العلامات إلى أرقام باستخدام TO\_NUMBER().

To number:تستخدم لتحويل النصوص الى ارقام لحساب المعدل لها

**select**

**exam\_type.name as exam\_type,**

**avg(to\_number(exam\_result.marks)) as average\_marks**

**from**

**exam\_result**

**join**

**exam on exam\_result.exam\_id = exam.exam\_id**

**join**

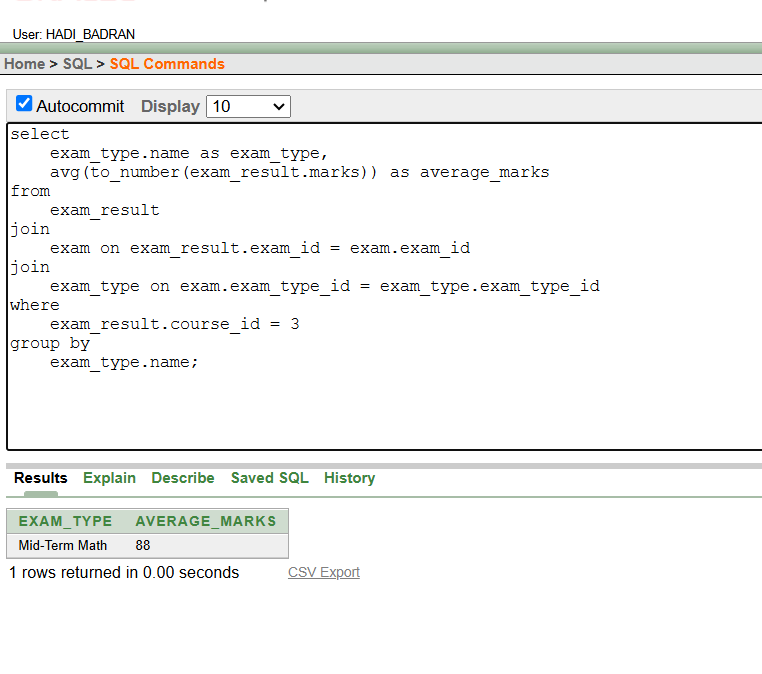
**exam\_type on exam.exam\_type\_id = exam\_type.exam\_type\_id**

**where**

**exam\_result.course\_id = 3**

**group by**

**exam\_type.name**;



7) Display all courses along with their associated grade and teacher.

**بدو الدوره الدراسية مع الصف الدراسي المرتبط بها، بالإضافة إلى اسم المعلم المسؤول عن التدريس في هذا الصف**

ربط جدول course مع جدول grade باستخدام grade\_id. هذا يتيح لنا معرفة الصف الدراسي المرتبط بكل دورة.

ربطنا جدول grade مع جدول classroom باستخدام grade\_id لنعرف الفصل الذي ينتمي إليه الصف الدراسي.

بطنا classroom مع teacher باستخدام teacher\_id للحصول على اسم المعلم المسؤول عن الفصل.

**select**

**course.name as course\_name,**

**grade.name as grade\_name,**

**teacher.fname || ' ' || teacher.lname as teacher\_name**

**from**

**course**

**join**

**grade on course.grade\_id = grade.grade\_id**

**join**

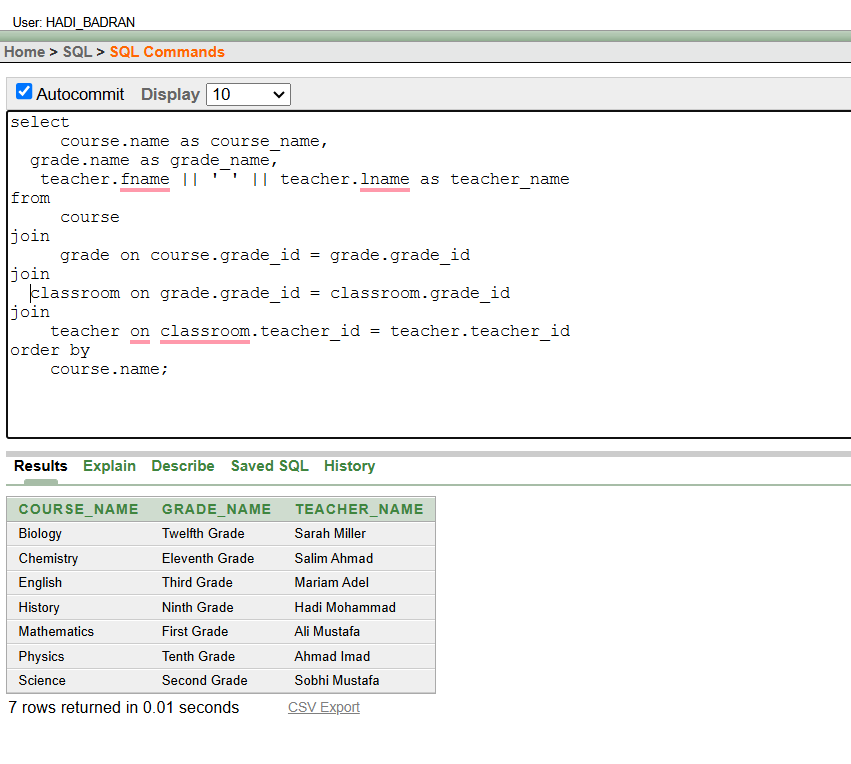
**classroom on grade.grade\_id = classroom.grade\_id**

**join**

**teacher on classroom.teacher\_id = teacher.teacher\_id**

**order by**

**course.name**;



8) List all classrooms along with the number of students in each.

يطلب عرض جميع الفصول الدراسيه مع عدد الطلاب في كل فصل

في هذا الاستعلام، **جبنا عمود** classroom.classroom\_id لعرض **رقم الفصل الدراسي**، و **جبنا عمود** classroom.year لعرض **السنة الدراسية** الخاصة بالفصل، بالإضافة إلى **جبنا عمود** classroom.section لعرض **القسم** الموجود في الفصل الدراسي. بعد ذلك، قمنا باستخدام **دالة** count(classroom\_student.student\_id) لحساب **عدد الطلاب** في كل فصل دراسي. كما قمنا **بربط جدول** classroom **مع جدول** classroom\_student باستخدام classroom\_id لربط الفصول الدراسية بالطلاب. ولتجميع النتائج بشكل صحيح، استخدمنا group by classroom.classroom\_id, classroom.year, classroom.section لتجميع البيانات حسب **رقم الفصل الدراسي**، **السنة الدراسية**، و **القسم**. وفي النهاية، قمنا **بترتيب النتائج حسب** classroom.classroom\_id للحصول على ترتيب النتائج حسب **رقم الفصل الدراسي**.

select

classroom.classroom\_id,

classroom.year,

classroom.section,

count(classroom\_student.student\_id) as student\_count

from

classroom

left join

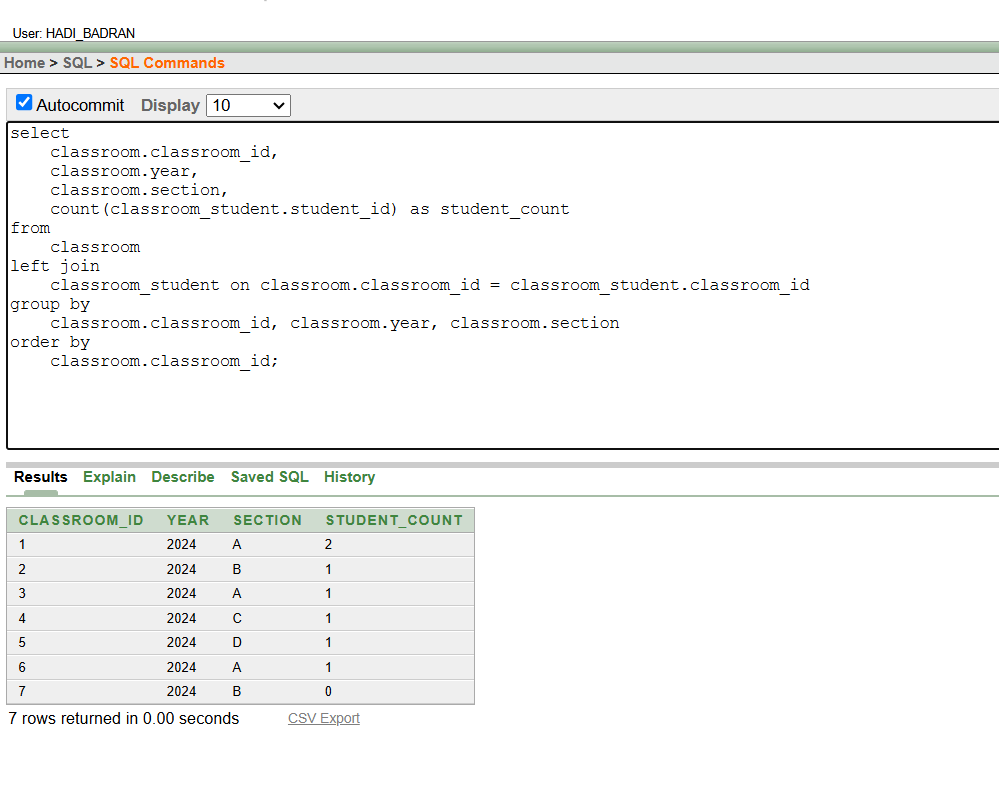
classroom\_student on classroom.classroom\_id = classroom\_student.classroom\_id

group by

classroom.classroom\_id, classroom.year, classroom.section

order by

classroom.classroom\_id;



**5) REPORT GENERATION:**

**the attached file**

****

**Classroom-wise student performance (average marks for each classroom)**

في هذا الاستعلام، **قمنا بإنشاء أو استبدال View** باسم report1 الذي يعرض **متوسط العلامات** لكل **فصل دراسي**. بدأنا أولًا ب**جلب بيانات الفصول الدراسية** من جدول classroom، ثم **قمنا بربط جدول** classroom **مع جدول** classroom\_student باستخدام classroom\_id لربط الطلاب بالفصول الدراسية. بعد ذلك، **قمنا بربط جدول** classroom\_student **مع جدول** exam\_result باستخدام student\_id للحصول على العلامات الخاصة بكل طالب. **تم استخدام دالة** AVG() **لحساب متوسط العلامات** لكل فصل دراسي، كما قمنا **بتجميع النتائج حسب** classroom\_id**,** year**, و** section. أخيرًا، **تم ترتيب النتائج حسب** classroom\_id للحصول على عرض مرتب للفصول الدراسية.

create or replace view report1 as

select

classroom.classroom\_id,

classroom.year,

classroom.section,

avg(to\_number(exam\_result.marks)) as average\_mark

from

classroom

join

classroom\_student on classroom.classroom\_id = classroom\_student.classroom\_id

join

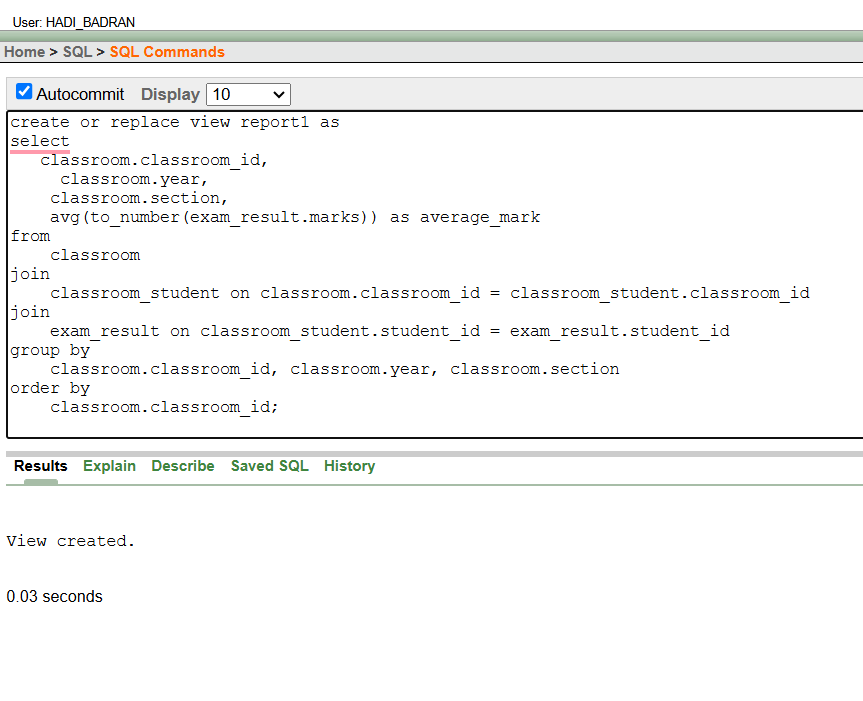
exam\_result on classroom\_student.student\_id = exam\_result.student\_id

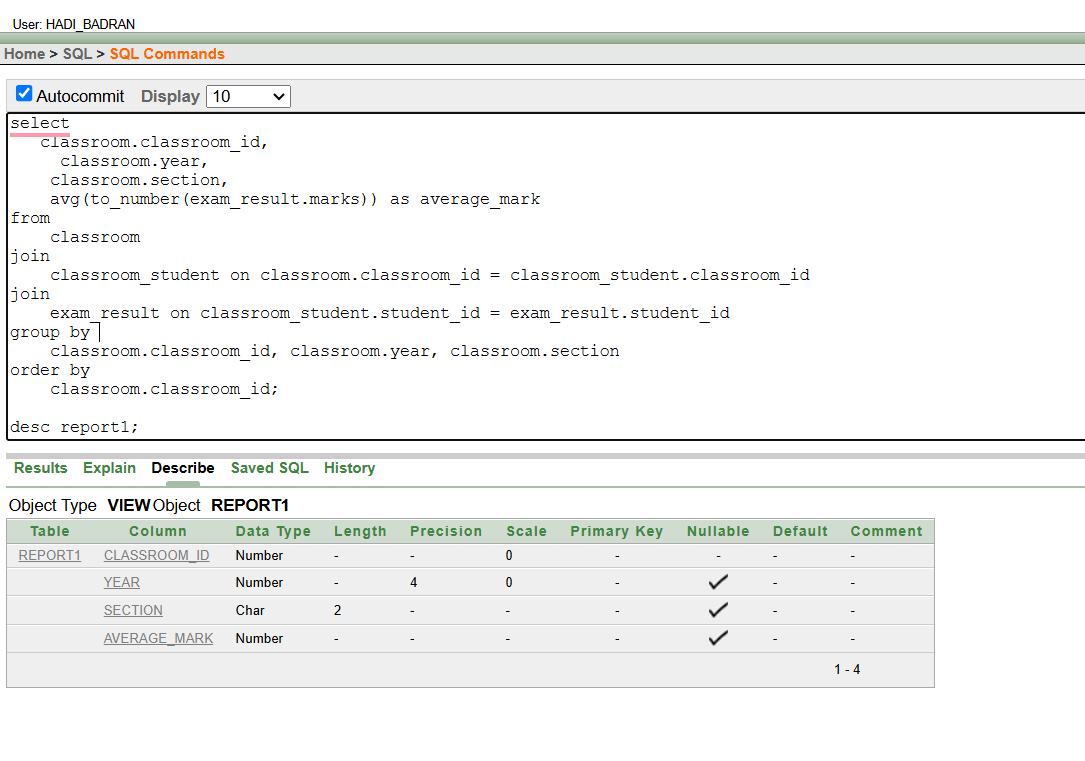
group by

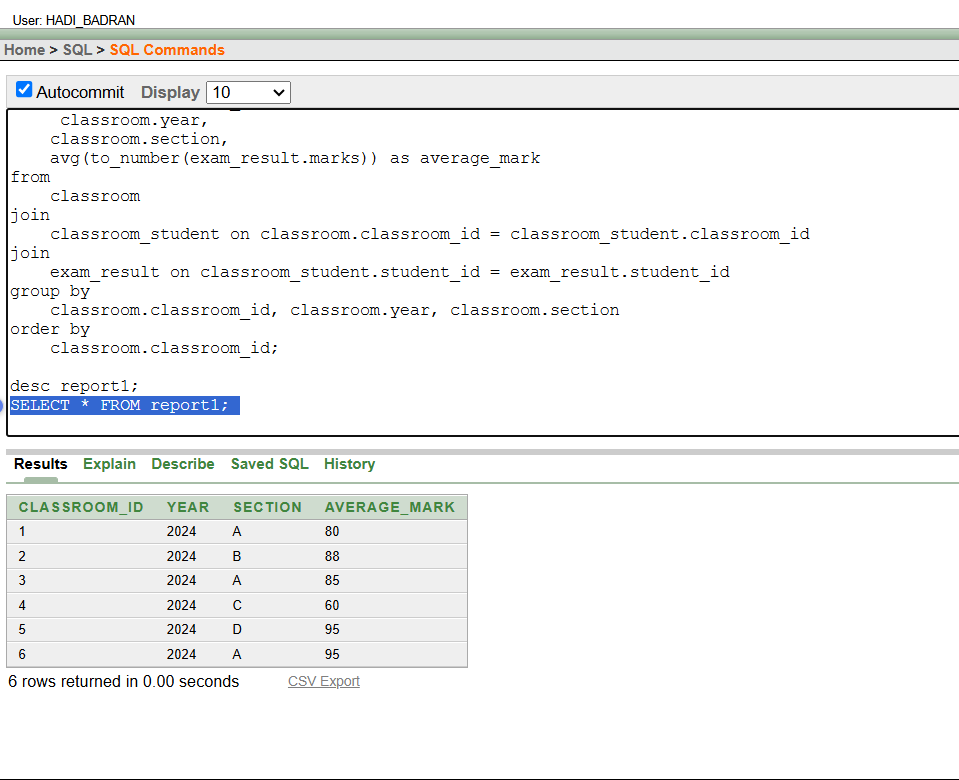
classroom.classroom\_id, classroom.year, classroom.section

order by

classroom.classroom\_id;







**Teacher-wise course details and the grades associated with those courses.**

**لاستعلام يعرض تفاصيل الدورات الدراسية التي يُدرسها كل معلم، بالإضافة إلى الصفوف الدراسية المرتبطة بتلك الدورات**

قمنا بإنشاء View باسم report2 والذي يعرض تفاصيل المعلمين والدورات الدراسية المرتبطة بالصفوف الدراسية. في البداية، **قمنا بربط جدول المعلمين (**teacher**) مع جدول الفصول الدراسية (**classroom**)** باستخدام teacher\_id، مما يتيح لنا معرفة المعلم المسؤول عن كل فصل دراسي. ثم، **ربطنا جدول الفصول الدراسية مع جدول الصفوف الدراسية (**grade**) باستخدام** grade\_id، لتحديد الصف الدراسي الذي ينتمي إليه الفصل. بعد ذلك، **قمنا بربط جدول الصفوف الدراسية مع جدول الدورات الدراسية (**course**) باستخدام** grade\_id، مما يمكننا من معرفة الدورات الدراسية المرتبطة بالصفوف الدراسية.

create or replace view report2 as

select

teacher.teacher\_id,

teacher.fname || ' ' || teacher.lname as teacher\_name,

course.course\_id,

course.name as course\_name,

grade.grade\_id,

grade.name as grade\_name

from

teacher

join

classroom on teacher.teacher\_id = classroom.teacher\_id

join

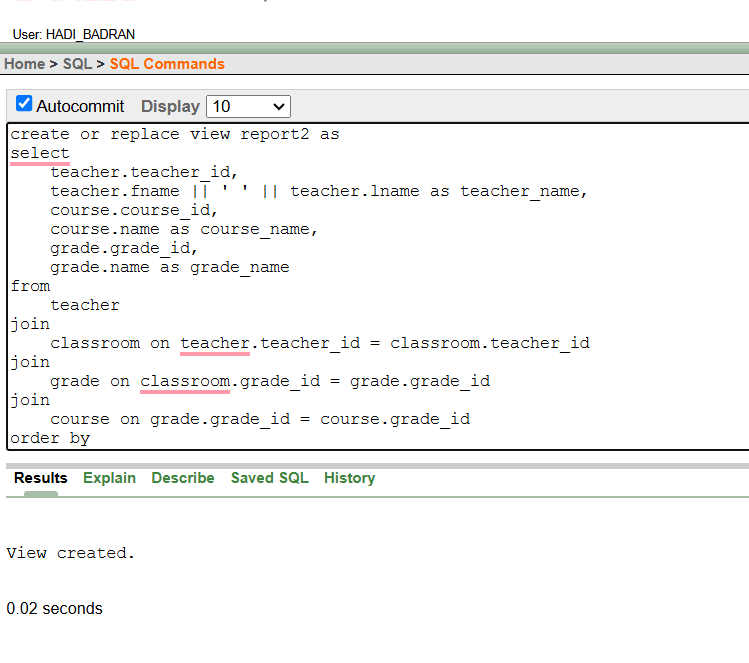
grade on classroom.grade\_id = grade.grade\_id

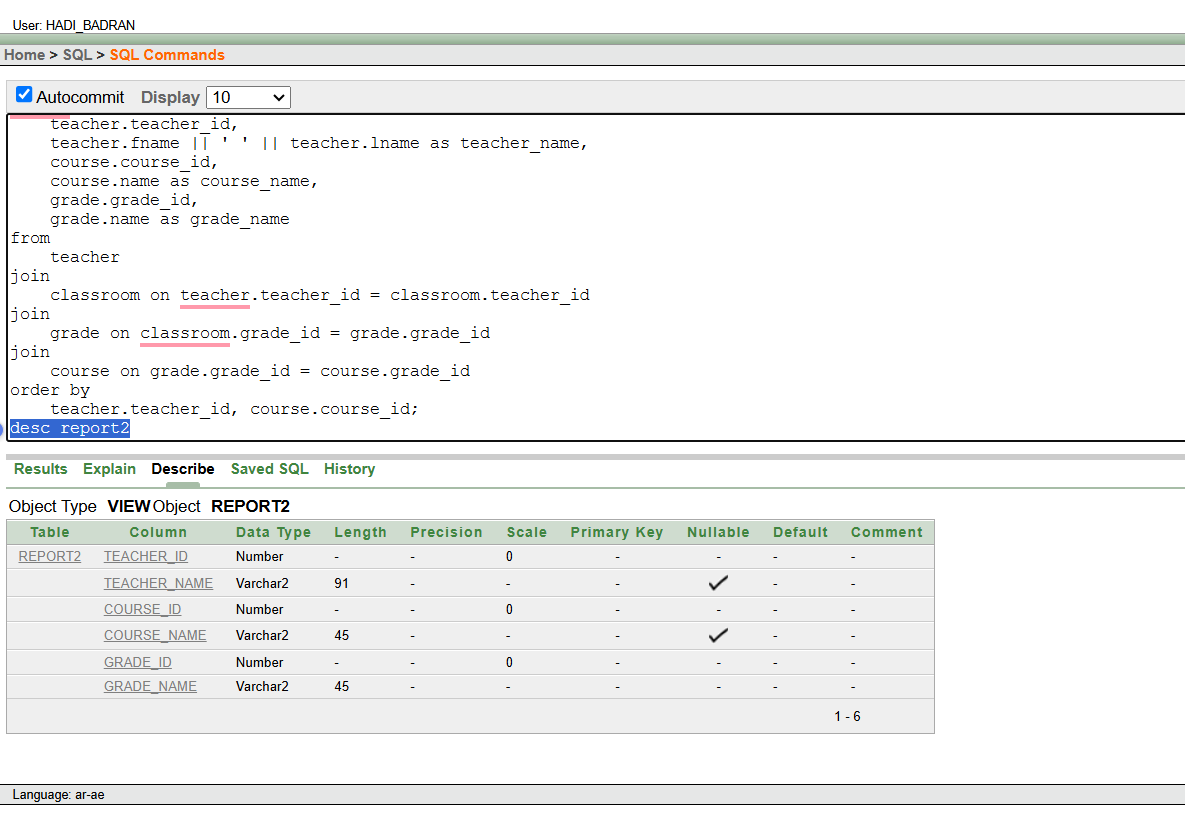
join

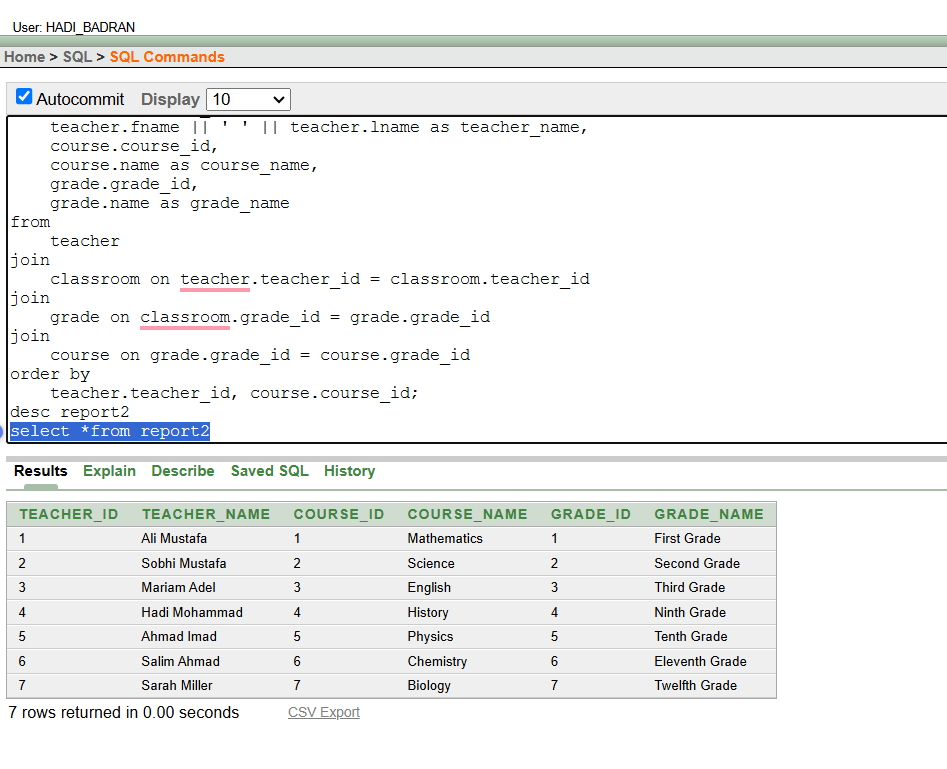
course on grade.grade\_id = course.grade\_id

order by

teacher.teacher\_id, course.course\_id;







**Name:hadi mohammad badran**

**Number:202212503**

**Dr.Areen Naji**