Entities

Entities	Attribute
Departments	DepartmentID, DepartmentName
Subjects	SubjectID, SubjectName, Credits, DepartmentID
Teachers	TeacherID, FName, LName, Email, DepartmentID
Students	StudentID, FName, LName, DateOfBirth, Gender, GradeLevel, Address, City, Country
Enrollments	EnrollmentID, StudentID, SubjectID, TeacherID, EnrollmentDate
Grades	GradeID, EnrollmentID, Score, LetterGrade, ExamType

Relationship

Relationship	Type
Departments → Subjects	One-to-Many
Enrollments → Grades	One-to-One
Subjects → Enrollments	One-to-Many
Students → Enrollments	One-to-Many
Teachers ↔ Subjects	One-to-Many
Departments → Teachers	One-to-Many

User

Type		Permissions
Admin	المسؤول عن النظام بالكامل	1-اضافه و حذف الكيانات 2-تعيين المدرسين للمواد الدراسية 3-تعديل البيانات الأساسية لي النظام 4-إدارة حسابات باقي المستخدمين
Teachers	يقوم بتدريس الطلاب	1-عرض المواد التي يُدرّسه 2-تسجيل الدرجات وتقارير الأداء للطلاب 3-رؤية قائمة الطلاب الذين يدرسون لديه 4-تحديث بعض البيانات المتعلقة بمادته فقط 5-لا يستطيع التعديل على بيانات المدرسين الآخرين
Students	يستخدم النظام لعرض بيناته فقط	1-عرض المواد التي سجل فيها 2-مشاهدة درجاته لكل مادة 3-رؤية معلومات المدرسين للمواد المسجلة فيها 4-لا يمكنه تعديل أي بيانات في النظام
Parents	متابعه أداء الطالب	1-رؤية درجات الطالب والتقارير الأكاديمية 2-تتبع التسجيلات الحالية والغيابات إن وجدت 3-استقبال إشعارات حول الأداء الدراسي 4-لا يمكنه تعديل أي بيانات
Staff	التعامل مع تسجيل الطلاب والمقررات	1-تسجيل الطلاب في المواد الدراسية 2-تحديث بيانات الطلاب (مثل العنوان أو رقم الهاتف) 3-إنشاء الجداول الدراسية وطباعتها 4-لا يمكنه تعديل الدرجات ولا لديه صلاحية كاملة على النظام

Query

1-

```
mysql> SELECT
    ->     s.student_id,
    ->     s.first_name,
    ->     s.last_name,
    ->     sb.subject_name
    -> FROM
    ->     Students s
    -> JOIN
    ->     Enrollments e ON s.student_id = e.student_id
    -> JOIN
    ->     Subjects sb ON e.subject_id = sb.subject_id
    -> WHERE
    ->     sb.subject_name = 'Introduction to Programming';
+------------+------------+-----------+-----------------------------+
| student_id | first_name | last_name | subject_name                |
+------------+------------+-----------+-----------------------------+
|          1 | Alice      | Wong      | Introduction to Programming |
+------------+------------+-----------+-----------------------------+
1 row in set (0.038 sec)
```

2-

```
mysql> SELECT
    ->     t.teacher_id,
    ->     t.first_name,
    ->     t.last_name,
    ->     d.department_name
    -> FROM
    ->     Teachers t
    -> JOIN
    ->     Departments d ON t.department_id = d.department_id;
+------------+------------+-----------+--------------------+
| teacher_id | first_name | last_name | department_name    |
+------------+------------+-----------+--------------------+
|          1 | Mohamed    | Gehad     | Computer Science   |
|          2 | Mahmoud    | Khamis    | Mathematics        |
|          3 | Youssef    | Khalifa   | Physics            |
|          4 | Mohammed   | 3zzat     | Biology            |
|          5 | Abdullah   | Bayoud    | Chemistry          |
|          6 | Ahmed      | Al-Jiar   | English Literature |
|          7 | Mariam     | Wilson    | History            |
|          8 | Marwan     | Musa      | Art                |
|          9 | Jessica    | Anderson  | Music              |
+------------+------------+-----------+--------------------+
```

3-

```
mysql> SELECT
    ->     sb.subject_name,
    ->     COUNT(e.student_id) AS total_students
    -> FROM
    ->     Subjects sb
    -> LEFT JOIN
    ->     Enrollments e ON sb.subject_id = e.subject_id
    -> GROUP BY
    ->     sb.subject_name;
+-----------------------------+----------------+
| subject_name                | total_students |
+-----------------------------+----------------+
| Introduction to Programming |              1 |
| Data Structures             |              1 |
| Calculus                    |              1 |
| Quantum Mechanics           |              1 |
| Biology Basics              |              1 |
| Organic Chemistry           |              1 |
| Shakespeare Studies         |              1 |
| World War II History        |              1 |
| Painting Techniques         |              1 |
| Music Theory                |              1 |
+-----------------------------+----------------+
10 rows in set (0.060 sec)
```

4-

```
mysql> SELECT
    ->     s.student_id,
    ->     s.first_name,
    ->     s.last_name,
    ->     g.score,
    ->     g.letter_grade
    -> FROM
    ->     Students s
    -> JOIN
    ->     Enrollments e ON s.student_id = e.student_id
    -> JOIN
    ->     Grades g ON e.enrollment_id = g.enrollment_id
    -> WHERE
    ->     g.score > 90;
+------------+------------+-----------+-------+--------------+
| student_id | first_name | last_name | score | letter_grade |
+------------+------------+-----------+-------+--------------+
|          2 | Bob        | Lee       | 92.00 | A-           |
|          1 | Alice      | Wong      | 95.00 | A+           |
|          6 | Liam       | Hernandez | 91.50 | A-           |
|          9 | Emma       | Lopez     | 94.00 | A+           |
+------------+------------+-----------+-------+--------------+
4 rows in set (0.037 sec)
```

5-

```
mysql> SELECT
    ->     sb.subject_name
    -> FROM
    ->     Subjects sb
    -> LEFT JOIN
    ->     Enrollments e ON sb.subject_id = e.subject_id
    -> WHERE
    ->     e.enrollment_id IS NULL;
Empty set (0.006 sec)
```

Introduction:

The Educational System Database Project is designed to manage data related to students, teachers, subjects, and grades. The project is built using the Relational Database Model, with entities and relationships converted into SQL tables.

Design:

- **Departments:** Stores academic department information.
- **Teachers:** Stores teacher data and their association with departments.
- **Subjects:** Stores subject data and their link to departments.
- **Students:** Stores student information.
- **Enrollments:** Tracks student enrollment in subjects.
- **Grades:** Records student grades for subjects.

Queries:

Five SQL queries were written to perform various tasks:

1. Display students enrolled in a specific subject.
2. Show teachers and their respective departments.
3. Display the number of students enrolled in each subject.
4. Show students who scored above 90.
5. List subjects with no student enrollments.

Results:

The project provides a solid foundation for managing educational system data.

It can be extended in the future to support user interfaces (Web/Mobile) or include additional features such as classrooms, timetables, or reporting tools.