



Data Analyst : Hossam El-Din Mohamed

BUILDING AND MANAGING ROBUST DATABASES



About Project

Welcome to my presentation on SQL, a cornerstone technology in the world of data analytics. Today, I will be guiding you through a comprehensive SQL project designed as part of the DEPI Data Analytics Track. This project involves creating and manipulating databases through a series of structured query tasks, reflecting real-world applications in academic and enterprise environments.



1 - Create the Students, Courses, Enrollments, Professors, and CourseAssignments tables with appropriate columns and constraints.



```
create table Students (StudentID int Primary key, FirstName  
Varchar(50), LastName Varchar(50), DateOfBirth Date,  
EnrollmentDate Date, Email Varchar(100), Age int)
```

```
create table Courses (CourseID int Primary Key, CourseName  
Varchar(100), Credits int, Department Varchar (50))
```

```
create table Enrollments (EnrollmentID int Primary Key, StudentID  
int, CourseID int, Grade char(2), Semester varchar(10))
```

```
create table Professors (ProfessorID int Primary Key, FirstName  
varchar(50), LastName varchar(50), Department varchar(50), Email  
varchar(100))
```

```
create table CourseAssignments (AssignmentID int primary key,  
ProfessorID int, CourseID int, Semester varchar(10))
```

-	Tables
+	System Tables
+	FileTables
+	External Tables
+	Graph Tables
+	dbo.CourseAssignments
+	dbo.Courses
+	dbo.Enrollments
+	dbo.Professors
+	dbo.Students



2 - Insert data into the Students, Courses, Enrollments, Professors, and CourseAssignments tables. Provide at least 3 records for each table.



```
insert into Students Values
(1, 'John', 'Doe', '2000-01-01', '2018-09-01', 'john.doe@example.com', Null),
(2, 'Jane', 'Smith', '1999-05-15', '2017-09-01',
'jane.smith@example.com', 21),
(3, 'Robert', 'Brown', '2001-11-21', '2019-09-01',
'robert.brown@example.com', 25),
(4, 'Emily', 'Jones', '2002-03-03', '2020-09-01',
'emily.jones@example.com', 22),
(5, 'Michael', 'Davis', '1998-07-22', '2016-09-01',
'michael.davis@example.com', 23),
(6, 'Linda', 'Wilson', '2000-12-12', '2018-09-01',
'linda.wilson@example.com', 21),
(7, 'James', 'Taylor', '2001-03-14', '2019-09-01',
'james.taylor@example.com', 22),
(8, 'Sarah', 'Lee', '2002-07-07', '2020-09-01',
'sarah.lee@example.com', 23),
(9, 'David', 'Martin', '1999-11-11', '2017-09-01',
'david.martin@example.com', 24),
(10, 'Susan', 'Clark', '2001-05-05', '2019-09-01',
'susan.clark@example.com', 22)
```

```
insert into Courses Values
(1, 'Introduction to SQL', 3, 'Computer Science'),
(2, 'Data Structures', 4, 'Computer Science'),
(3, 'Database Management Systems', 3, 'Information Technology'),
(4, 'Algorithms', 4, 'Computer Science'),
(5, 'Operating Systems', 3, 'Information Technology'),
(6, 'Web Development', 3, 'Computer Science'),
(7, 'Computer Networks', 3, 'Information Technology');

insert into Enrollments Values
(1, 1, 2, 'A', 'Fall2020'),
(2, 2, 3, 'B+', 'Spring2021'),
(3, 3, 2, 'A-', 'Fall2021'),
(4, 4, 3, 'B', 'Spring2022'),
(5, 2, 2, 'A', 'Fall2020'),
(6, 5, 1, 'B+', 'Fall2020'),
(7, 6, 4, 'A', 'Spring2021'),
(8, 7, 5, 'B', 'Fall2021'),
(9, 8, 6, 'A-', 'Spring2022'),
(10, 9, 7, 'B+', 'Fall2020'),
(11, 10, 1, 'A', 'Spring2021'),
(12, 1, 3, 'B', 'Spring2021'),
(13, 2, 4, 'A-', 'Fall2021'),
(14, 3, 5, 'B+', 'Spring2022'),
(15, 4, 6, 'A', 'Fall2020');
```

```
insert into Professors Values
(1, 'Dr. Alice', 'Johnson', 'Computer Science', 'alice.johnson@example.com'),
(2, 'Dr. Bob', 'Miller', 'Information Technology', 'bob.miller@example.com'),
(3, 'Dr. Carol', 'Williams', 'Computer Science', 'carol.williams@example.com'),
(4, 'Dr. David', 'Jones', 'Information Technology', 'david.jones@example.com'),
(5, 'Dr. Emma', 'Brown', 'Computer Science', 'emma.brown@example.com');

insert into CourseAssignments Values
(1, 1, 1, 'Fall2020'),
(2, 1, 2, 'Spring2021'),
(3, 2, 3, 'Fall2020'),
(4, 3, 2, 'Spring2022'),
(5, 4, 4, 'Fall2021'),
(6, 5, 5, 'Spring2022'),
(7, 1, 6, 'Fall2020'),
(8, 2, 7, 'Spring2021');
```



3 - Update the email of the student with StudentID 1 to 'john.doe@newmail.com'.



update Students

Set email = 'john.doe@newmail.com'

where StudentID = 1;

StudentID	First Name	Last Name	Date Of Birth	Enrollment Date	Email	Age
1	John	Doe	2000-01-01	2018-09-01	john.doe@newmail.com	NULL
2	Jane	Smith	1999-05-15	2017-09-01	jane.smith@example.com	21
3	Robert	Brown	2001-11-21	2019-09-01	robert.brown@example.com	25
4	Emily	Jones	2002-03-03	2020-09-01	emily.jones@example.com	22
5	Michael	Davis	1998-07-22	2016-09-01	michael.davis@example.com	23
6	Linda	Wilson	2000-12-12	2018-09-01	linda.wilson@example.com	21
7	James	Taylor	2001-03-14	2019-09-01	james.taylor@example.com	22
8	Sarah	Lee	2002-07-07	2020-09-01	sarah.lee@example.com	23
9	David	Martin	1999-11-11	2017-09-01	david.martin@example.com	24
10	Susan	Clark	2001-05-05	2019-09-01	susan.clark@example.com	22



4 - Delete the record of the student with StudentID 5 from the Students table.



Delete from Students
where StudentID = 5;

	StudentID	FirstName	Last Name	Date Of Birth	Enrollment Date	Email	Age
1	1	John	Doe	2000-01-01	2018-09-01	john.doe@newmail.com	NULL
2	2	Jane	Smith	1999-05-15	2017-09-01	jane.smith@example.com	21
3	3	Robert	Brown	2001-11-21	2019-09-01	robert.brown@example.com	25
4	4	Emily	Jones	2002-03-03	2020-09-01	emily.jones@example.com	22
5	6	Linda	Wilson	2000-12-12	2018-09-01	linda.wilson@example.com	21
6	7	James	Taylor	2001-03-14	2019-09-01	james.taylor@example.com	22
7	8	Sarah	Lee	2002-07-07	2020-09-01	sarah.lee@example.com	23
8	9	David	Martin	1999-11-11	2017-09-01	david.martin@example.com	24
9	10	Susan	Clark	2001-05-05	2019-09-01	susan.clark@example.com	22



5 - Select all records from the Students table.



Select *
From Students;

	StudentID	FirstName	LastName	DateOfBirth	EnrollmentDate	Email	Age
1	1	John	Doe	2000-01-01	2018-09-01	john.doe@newmail.com	NULL
2	2	Jane	Smith	1999-05-15	2017-09-01	jane.smith@example.com	21
3	3	Robert	Brown	2001-11-21	2019-09-01	robert.brown@example.com	25
4	4	Emily	Jones	2002-03-03	2020-09-01	emily.jones@example.com	22
5	6	Linda	Wilson	2000-12-12	2018-09-01	linda.wilson@example.com	21
6	7	James	Taylor	2001-03-14	2019-09-01	james.taylor@example.com	22
7	8	Sarah	Lee	2002-07-07	2020-09-01	sarah.lee@example.com	23
8	9	David	Martin	1999-11-11	2017-09-01	david.martin@example.com	24
9	10	Susan	Clark	2001-05-05	2019-09-01	susan.clark@example.com	22



6- Select the FirstName and LastName of all students who enrolled after '2018-01-01'.



Select FirstName, LastName
From Students
Where EnrollmentDate > '2018-01-01';

1	John	Doe
2	Robert	Brown
3	Emily	Jones
4	Linda	Wilson
5	James	Taylor
6	Sarah	Lee
7	Susan	Clark



7 - Count the number of students in the Students table.



**Select Count(*) as Number_of_students
From Students;**

The screenshot shows a database interface with a teal header bar. On the left, there are two tabs: "Results" (highlighted in blue) and "Messages". The main area displays a single row in a table with one column labeled "Number_of_students". The value "9" is shown in a light blue cell with a black dashed border. The background of the main area is white.

Number_of_students
9



8 - Select all records from the Courses table.



Select *
From Enrollments;

	EnrollmentID	StudentID	CourseID	Grade	Semester
1	1	1	2	A	Fall2020
2	2	2	3	B+	Spring2021
3	3	3	2	A-	Fall2021
4	4	4	3	B	Spring2022
5	5	2	2	A	Fall2020
6	6	5	1	B+	Fall2020
7	7	6	4	A	Spring2021
8	8	7	5	B	Fall2021
9	9	8	6	A-	Spring2022
10	10	9	7	B+	Fall2020
11	11	10	1	A	Spring2021
12	12	1	3	B	Spring2021
13	13	2	4	A-	Fall2021
14	14	3	5	B+	Spring2022
15	15	4	6	A	Fall2020



9 - Select the CourseName and Credits for courses in the 'Computer Science' department.



```
Select CourseName, Credits  
From Courses  
Where Department = 'Computer Science';
```

CourseName	Credits
Introduction to SQL	3
Data Structures	4
Algorithms	4
Web Development	3



10 - Find the total number of credits offered by the 'Information Technology' department.



```
SELECT SUM(Credits) AS TotalCredits  
FROM Courses  
WHERE Department = 'Information Technology';
```

TotalCredits
9



11 - List each student's FirstName, LastName, and the names of the courses they are enrolled in.



Select FirstName, LastName, CourseName

From Students s

Join Enrollments e

on s.StudentID = e.StudentID

Join Courses c

on e.CourseID = c.CourseID;

	FirstName	LastName	CourseName
1	John	Doe	Data Structures
2	Jane	Smith	Database Management Systems
3	Robert	Brown	Data Structures
4	Emily	Jones	Database Management Systems
5	Jane	Smith	Data Structures
6	Linda	Wilson	Algorithms
7	James	Taylor	Operating Systems
8	Sarah	Lee	Web Development
9	David	Martin	Computer Networks
10	Susan	Clark	Introduction to SQL
11	John	Doe	Database Management Systems
12	Jane	Smith	Algorithms
13	Robert	Brown	Operating Systems
14	Emily	Jones	Web Development



12 - Find the names of courses that have more than one student enrolled. List the CourseName and the number of students.



```
Select c.CourseName, Count(e.StudentID)  
as num_of_stud  
from Courses c  
Join Enrollments e  
on c.CourseID = e.CourseID  
Group by CourseName  
Having Count(e.StudentID) >1
```

	CourseName	num_of_stud
1	Algorithms	2
2	Data Structures	3
3	Database Management Systems	3
4	Introduction to SQL	2
5	Operating Systems	2
6	Web Development	2



13 - Select all students and order them by their EnrollmentDate in descending order



Select *
From Students
Order by EnrollmentDate DESC;

	StudentID	FirstName	LastName	DateOfBirth	EnrollmentDate	Email	Age
1	4	Emily	Jones	2002-03-03	2020-09-01	emily.jones@example.com	22
2	8	Sarah	Lee	2002-07-07	2020-09-01	sarah.lee@example.com	23
3	3	Robert	Brown	2001-11-21	2019-09-01	robert.brown@example.com	25
4	7	James	Taylor	2001-03-14	2019-09-01	james.taylor@example.com	22
5	10	Susan	Clark	2001-05-05	2019-09-01	susan.clark@example.com	22
6	6	Linda	Wilson	2000-12-12	2018-09-01	linda.wilson@example.com	21
7	1	John	Doe	2000-01-01	2018-09-01	john.doe@newmail.com	NULL
8	2	Jane	Smith	1999-05-15	2017-09-01	jane.smith@example.com	21
9	9	David	Martin	1999-11-11	2017-09-01	david.martin@example.com	24



14 - Select all students and order them by their EnrollmentDate in descending order



Select FirstName, LastName

From Students s

left join Enrollments e

on s.StudentID = e.StudentID

where EnrollmentID is Null

FirstName	LastName



15- Find the average number of credits for each department. List the Department and the average credits.

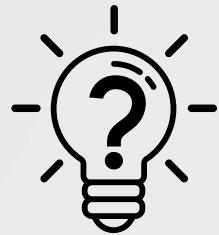


Select Avg(Credits) as Avg_Cred, Department
From Courses
Group by Department;

	Avg_Cred	Department
1	3	Computer Science
2	3	Information Technology



16 - List the FirstName, LastName, CourseName, and Grade of students enrolled in courses for the 'Fall2020' semester.



```
Select s.FirstName, s.LastName, c.CourseName, e.Grade  
From Courses c  
join Enrollments e  
on c.CourseID = e.CourseID  
join Students s  
on e.StudentID = s.StudentID  
Where e.Semester= 'Fall2020';
```

	FirstName	LastName	CourseName	Grade
1	John	Doe	Data Structures	A
2	Jane	Smith	Data Structures	A
3	David	Martin	Computer Networks	B+
4	Emily	Jones	Web Development	A



17 - List the CourseName and the number of students enrolled in each course.



Select c.CourseName, Count(e.StudentID)

From Enrollments e

Left join Courses c

on e.CourseID = c.CourseID

Group by c.CourseName;

	CourseName	(No column name)
1	Algorithms	2
2	Computer Networks	1
3	Data Structures	3
4	Database Management Systems	3
5	Introduction to SQL	2
6	Operating Systems	2
7	Web Development	2



18 - Find the FirstName, LastName, CourseName, and Grade of students who received a grade lower than 'B'.



Select s.FirstName, s.LastName, c.CourseName, e.Grade

From Students s

Join Enrollments e

on s.StudentID = e.StudentID

join Courses c

on e.CourseID = c.CourseID

Where Grade Not in ('B','B+','A-','A','A+')

FirstName	LastName
-----------	----------



19 - List each StudentID, FirstName, LastName, and the total number of enrollments.



```
SELECT s.StudentID, s.FirstName, s.LastName,  
COUNT(e.EnrollmentID) AS TotalEnrollments  
FROM Students s  
LEFT JOIN Enrollments e ON s.StudentID =  
e.StudentID  
GROUP BY s.StudentID, s.FirstName, s.LastName
```

	StudentID	First Name	Last Name	TotalEnrollments
1	1	John	Doe	2
2	2	Jane	Smith	3
3	3	Robert	Brown	2
4	4	Emily	Jones	2
5	6	Linda	Wilson	1
6	7	James	Taylor	1
7	8	Sarah	Lee	1
8	9	David	Martin	1
9	10	Susan	Clark	1



20 - List the CourseName of courses that have no enrollments.



```
Select CourseName  
From Courses c  
Left join Enrollments e  
on c.CourseID = e.CourseID  
Where EnrollmentID is null
```

CourseName



21 - List the top 3 students with the highest number of course enrollments. Include StudentID, FirstName, LastName, and TotalEnrollments.



```
Select top 3 s.StudentID, s.FirstName, s.LastName,  
Count(e.EnrollmentID) as Total_enrollment  
from Students s  
join Enrollments e  
on s.StudentID = e.StudentID  
Group by s.StudentID, s.FirstName, s.LastName  
Order By Count(e.EnrollmentID) Desc
```

	StudentID	FirstName	LastName	Total_enrollment
1	2	Jane	Smith	3
2	4	Emily	Jones	2
3	3	Robert	Brown	2



22 - List each professor's FirstName, LastName, and the names of the courses they are assigned to teach.



```
SELECT p.FirstName, p.LastName, c.CourseName  
FROM Professors p  
JOIN CourseAssignments ca  
ON p.ProfessorID = ca.ProfessorID  
JOIN Courses c  
ON ca.CourseID = c.CourseID
```

	FirstName	LastName	CourseName
1	Dr. Alice	Johnson	Introduction to SQL
2	Dr. Alice	Johnson	Data Structures
3	Dr. Bob	Miller	Database Management System
4	Dr. Carol	Williams	Data Structures
5	Dr. David	Jones	Algorithms
6	Dr. Emma	Brown	Operating Systems
7	Dr. Alice	Johnson	Web Development
8	Dr. Bob	Miller	Computer Networks



23 - Find the number of professors in each department. List the Department and the number of professors.



```
SELECT Department, COUNT(ProfessorID) AS  
NumberOfProfessors  
FROM Professors  
GROUP BY Department;
```

	Department	NumberOfProfessors
1	Computer Science	3
2	Information Technology	2



24 - Select all courses and order them by Credits in ascending order



Select *
From Courses
Order by Credits Asc;

	CourseID	CourseName	Credits	Department
1	1	Introduction to SQL	3	Computer Science
2	3	Database Management Systems	3	Information Technology
3	5	Operating Systems	3	Information Technology
4	6	Web Development	3	Computer Science
5	7	Computer Networks	3	Information Technology
6	4	Algorithms	4	Computer Science
7	2	Data Structures	4	Computer Science



25 - List the FirstName and LastName of professors who are teaching more than one course



```
Select FirstName, LastName, COUNT(ca.CourseID)  
as Num_of_courses  
from Professors p  
Left Join CourseAssignments ca  
on p.ProfessorID = ca.ProfessorID  
join Courses c  
on ca.CourseID = c.CourseID  
Group by FirstName, LastName  
Having COUNT(ca.CourseID) > 1;
```

FirstName	LastName	Num_of_courses
Dr. Alice	Johnson	3
Dr. Bob	Miller	2



26 - List the FirstName and LastName of professors who are teaching more than one course



```
SELECT s.FirstName,s.LastName,c.CourseName  
FROM Students s  
JOIN Enrollments e  
ON s.StudentID = e.StudentID  
JOIN Courses c  
ON e.CourseID = c.CourseID  
JOIN CourseAssignments ca  
ON c.CourseID = ca.CourseID  
JOIN Professors p  
ON ca.ProfessorID = p.ProfessorID  
WHERE p.FirstName = 'Dr. Alice' AND p.LastName = 'Johnson';
```

	FirstName	LastName	CourseName
1	John	Doe	Data Structures
2	Robert	Brown	Data Structures
3	Jane	Smith	Data Structures
4	Sarah	Lee	Web Development
5	Susan	Clark	Introduction to SQL
6	Emily	Jones	Web Development



27 - Find the total number of credits each student is enrolled in. List the StudentID, FirstName, LastName, and TotalCredits.



```
SELECT s.StudentID, s.FirstName, s.LastName, SUM(c.Credits)
AS TotalCredits
FROM Students s
JOIN Enrollments e
ON s.StudentID = e.StudentID
JOIN Courses c
ON e.CourseID = c.CourseID
GROUP BY s.StudentID, s.FirstName, s.LastName;
```

	StudentID	FirstName	LastName	TotalCredits
1	1	John	Doe	7
2	2	Jane	Smith	11
3	3	Robert	Brown	7
4	4	Emily	Jones	6
5	6	Linda	Wilson	4
6	7	James	Taylor	3
7	8	Sarah	Lee	3
8	9	David	Martin	3
9	10	Susan	Clark	3



28 - Find the number of students enrolled in each course for the 'Spring2021' semester. List the CourseID, CourseName, and StudentCount.



```
SELECT c.CourseID, c.CourseName, COUNT(e.StudentID) AS  
Num_of_student  
FROM Courses c  
JOIN Enrollments e  
ON c.CourseID = e.CourseID  
WHERE e.Semester = 'Spring2021'  
GROUP BY c.CourseID, c.CourseName;
```

	CourseID	CourseName	Num_of_student
1	1	Introduction to SQL	1
2	3	Database Management Systems	2
3	4	Algorithms	1



29 - List the CourseName and Department of courses that had enrollments in 'Spring2021'.



```
SELECT CourseName, Department  
from Courses c  
join Enrollments e  
on c.CourseID = e.CourseID  
WHERE e.Semester = 'Spring2021'
```

	CourseName	Department
1	Database Management Systems	Information Technology
2	Algorithms	Computer Science
3	Introduction to SQL	Computer Science
4	Database Management Systems	Information Technology



30 - For each student, list their StudentID, FirstName, LastName, and the date of their last enrollment.



```
SELECT s.StudentID, s.FirstName, s.LastName,  
MAX(s.EnrollmentDate) AS LastEnrollmentDate  
FROM Students s  
JOIN Enrollments e  
ON s.StudentID = e.StudentID  
GROUP BY s.StudentID, s.FirstName, s.LastName;
```

	StudentID	FirstName	LastName	LastEnrollmentDate
1	1	John	Doe	2018-09-01
2	2	Jane	Smith	2017-09-01
3	3	Robert	Brown	2019-09-01
4	4	Emily	Jones	2020-09-01
5	6	Linda	Wilson	2018-09-01
6	7	James	Taylor	2019-09-01
7	8	Sarah	Lee	2020-09-01
8	9	David	Martin	2017-09-01
9	10	Susan	Clark	2019-09-01



31 - Find the names of students who have enrolled in 'Data Structures' (CourseID 2).



```
SELECT s.FirstName, s.LastName  
FROM Students s  
JOIN Enrollments e  
ON s.StudentID = e.StudentID  
WHERE e.CourseID = 2
```

	FirstName	LastName
1	John	Doe
2	Robert	Brown
3	Jane	Smith



32 - List all students along with the total number of courses they are enrolled in.



```
SELECT s.StudentID, s.FirstName, s.LastName,  
COUNT(e.CourseID) AS TotalCourses  
FROM Students s  
LEFT JOIN Enrollments e  
ON s.StudentID = e.StudentID  
GROUP BY s.StudentID, s.FirstName, s.LastName
```

	StudentID	FirstName	LastName	TotalCourses
1	1	John	Doe	2
2	2	Jane	Smith	3
3	3	Robert	Brown	2
4	4	Emily	Jones	2
5	6	Linda	Wilson	1
6	7	James	Taylor	1
7	8	Sarah	Lee	1
8	9	David	Martin	1
9	10	Susan	Clark	1



33 - List all students with their enrolled courses and grades. Include StudentID, FirstName, LastName, CourseName, and Grade.



```
SELECT s.StudentID, s.FirstName, s.LastName,  
c.CourseName, e.Grade  
FROM Students s  
JOIN Enrollments e  
ON s.StudentID = e.StudentID  
JOIN Courses c  
ON e.CourseID = c.CourseID
```

	StudentID	FirstName	LastName	CourseName	Grade
1	1	John	Doe	Data Structures	A
2	2	Jane	Smith	Database Management Systems	B+
3	3	Robert	Brown	Data Structures	A-
4	4	Emily	Jones	Database Management Systems	B
5	2	Jane	Smith	Data Structures	A
6	6	Linda	Wilson	Algorithms	A
7	7	James	Taylor	Operating Systems	B
8	8	Sarah	Lee	Web Development	A-
9	9	David	Martin	Computer Networks	B+
10	10	Susan	Clark	Introduction to SQL	A
11	1	John	Doe	Database Management Systems	B
12	2	Jane	Smith	Algorithms	A-
13	3	Robert	Brown	Operating Systems	B+
14	4	Emily	Jones	Web Development	A



34 - Find the total number of enrollments for each student. List the StudentID, FirstName, LastName, and TotalEnrollments.



```
SELECT s.StudentID, s.FirstName, s.LastName,  
COUNT(e.EnrollmentID) AS TotalEnrollments  
FROM Students s  
LEFT JOIN Enrollments e  
ON s.StudentID = e.StudentID  
GROUP BY s.StudentID, s.FirstName, s.LastName
```

	StudentID	FirstName	LastName	TotalEnrollments
1	1	John	Doe	2
2	2	Jane	Smith	3
3	3	Robert	Brown	2
4	4	Emily	Jones	2
5	6	Linda	Wilson	1
6	7	James	Taylor	1
7	8	Sarah	Lee	1
8	9	David	Martin	1
9	10	Susan	Clark	1



35 - Create a view named EnrollmentSummary that displays StudentID, FirstName, LastName, CourseID, CourseName, and Grade from the Enrollments table.



CREATE VIEW EnrollmentSummary AS

```
SELECT s.StudentID, s.FirstName, s.LastName,  
e.CourseID, c.CourseName, e.Grade  
FROM Enrollments e  
JOIN Students s  
ON e.StudentID = s.StudentID  
JOIN Courses c  
ON e.CourseID = c.CourseID;
```

	StudentID	FirstName	LastName	TotalEnrollments
1	1	John	Doe	2
2	2	Jane	Smith	3
3	3	Robert	Brown	2
4	4	Emily	Jones	2
5	6	Linda	Wilson	1
6	7	James	Taylor	1
7	8	Sarah	Lee	1
8	9	David	Martin	1
9	10	Susan	Clark	1



36 - Find the average age of students enrolled in each course. List the CourseID, CourseName, and AverageAge.



```
SELECT c.CourseID, c.CourseName, AVG(s.Age)  
AS Average_Age  
FROM Courses c  
JOIN Enrollments e  
ON c.CourseID = e.CourseID  
JOIN Students s  
ON e.StudentID = s.StudentID  
GROUP BY c.CourseID, c.CourseName;
```

	CourseID	CourseName	Average_Age
1	1	Introduction to SQL	22
2	2	Data Structures	23
3	3	Database Management Systems	21
4	4	Algorithms	21
5	5	Operating Systems	23
6	6	Web Development	22
7	7	Computer Networks	24



37 - Find the highest and lowest grade received by students in each course. List the CourseID, CourseName, HighestGrade, and LowestGrade.



```
SELECT c.CourseID, c.CourseName,  
MAX(e.Grade) AS HighestGrade,  
MIN(e.Grade) AS LowestGrade  
FROM Courses c  
JOIN Enrollments e  
ON c.CourseID = e.CourseID  
GROUP BY c.CourseID, c.CourseName;
```

	CourseID	CourseName	HighestGrade	LowestGrade
1	1	Introduction to SQL	B+	A
2	2	Data Structures	A-	A
3	3	Database Management Systems	B+	B
4	4	Algorithms	A-	A
5	5	Operating Systems	B+	B
6	6	Web Development	A-	A
7	7	Computer Networks	B+	B+



38 - Calculate the average grade of students enrolled in each course. List the CourseID, CourseName, and AvgGrade.



```
SELECT c.CourseID, c.CourseName,  
AVG(CASE
```

```
    WHEN e.Grade = 'A' THEN 4.0  
    WHEN e.Grade = 'B+' THEN 3.5  
    WHEN e.Grade = 'B' THEN 3.0  
    WHEN e.Grade = 'C+' THEN 2.5  
    WHEN e.Grade = 'C' THEN 2.0  
    WHEN e.Grade = 'D+' THEN 1.5  
    WHEN e.Grade = 'D' THEN 1.0  
    WHEN e.Grade = 'F' THEN 0.0  
    ELSE NULL
```

```
END) AS AvgGrade
```

```
FROM Courses c
```

```
JOIN Enrollments e ON c.CourseID = e.CourseID
```

```
GROUP BY c.CourseID, c.CourseName
```

	CourseID	CourseName	AvgGrade
1	1	Introduction to SQL	3.750000
2	2	Data Structures	4.000000
3	3	Database Management Systems	3.166666
4	4	Algorithms	4.000000
5	5	Operating Systems	3.250000
6	6	Web Development	4.000000
7	7	Computer Networks	3.500000



39 - Find the total grades for students in each department. List the Department and TotalGrades.



```
SELECT p.Department,
       SUM(CASE
           WHEN e.Grade = 'A' THEN 4.0
           WHEN e.Grade = 'B+' THEN 3.5
           WHEN e.Grade = 'B' THEN 3.0
           WHEN e.Grade = 'C+' THEN 2.5
           WHEN e.Grade = 'C' THEN 2.0
           WHEN e.Grade = 'D+' THEN 1.5
           WHEN e.Grade = 'D' THEN 1.0
           WHEN e.Grade = 'F' THEN 0.0
           ELSE NULL
       END) AS TotalGrades
  FROM Enrollments e
  JOIN Students s
    ON e.StudentID = s.StudentID
  JOIN Courses c
    ON e.CourseID = c.CourseID
  JOIN Professors p
    ON c.Department = p.Department
 GROUP BY p.Department
```

	Department	TotalGrades
1	Computer Science	60.0
2	Information Technology	39.0



40 - List the FirstName, LastName, CourseName, and ProfessorName for each enrollment.



```
SELECT s.FirstName, s.LastName, c.CourseName  
FROM Enrollments e  
JOIN Students s  
ON e.StudentID = s.StudentID  
JOIN Courses c  
ON e.CourseID = c.CourseID  
JOIN CourseAssignments ca  
ON e.CourseID = ca.CourseID  
JOIN Professors p  
ON ca.ProfessorID = p.ProfessorID
```

	FirstName	LastName	CourseName
1	John	Doe	Data Structures
2	John	Doe	Data Structures
3	Jane	Smith	Database Management Systems
4	Robert	Brown	Data Structures
5	Robert	Brown	Data Structures
6	Emily	Jones	Database Management Systems
7	Jane	Smith	Data Structures
8	Jane	Smith	Data Structures
9	Linda	Wilson	Algorithms
10	James	Taylor	Operating Systems
11	Sarah	Lee	Web Development
12	David	Martin	Computer Networks
13	Susan	Clark	Introduction to SQL
14	John	Doe	Database Management Systems
15	Jane	Smith	Algorithms
16	Robert	Brown	Operating Systems
17	Emily	Jones	Web Development

CONTACT

Hussam.mo.saad@gmail.com



01128330688



Cairo, Egypt





THANK YOU