1st created the database using

CREATE DATABASE StudentCourseManagement;

2-Creating the tables

```sql
-- Students table
CREATE TABLE Students (
    student_id INT IDENTITY(1,1) PRIMARY KEY,--IDENTITY(1,1) is used to generate unique, auto-incrementing values for a column.
    first_name VARCHAR(100),
    last_name VARCHAR(100),
    email VARCHAR(255),
    date_of_birth DATE
);

-- Courses table
CREATE TABLE Courses (
    course_id INT IDENTITY(1,1) PRIMARY KEY,
    course_name VARCHAR(255),
    course_description TEXT
);

-- Instructors table
CREATE TABLE Instructors (
    instructor_id INT IDENTITY(1,1) PRIMARY KEY,
    first_name VARCHAR(100),
    last_name VARCHAR(100),
    email VARCHAR(255)
);

-- Enrollments table
CREATE TABLE Enrollments (
    enrollment_id INT IDENTITY(1,1) PRIMARY KEY,
    student_id INT,
    course_id INT,
    enrollment_date DATE,
    FOREIGN KEY (student_id) REFERENCES Students(student_id),
    FOREIGN KEY (course_id) REFERENCES Courses(course_id)
);
```

3-Insert Data

```sql
-- Inserting sample data for Students
INSERT INTO Students (first_name, last_name, email, date_of_birth)
VALUES
  ('Michael', 'Matta', 'michaelj@example.com', '2001-12-10'),
  ('Ahmed', 'idk', 'ahmed@example.com', '2001-12-11'),
  ('Mohamed', 'idk2', 'mohamed@example.com', '2001-12-12'),
  ('Mostafa', 'idk3', 'mostafa@example.com', '2001-12-12'),
  ('Marawan', 'idk4', 'marawan@example.com', '2001-12-13'),
  ('Kerolos', 'idk5', 'kerolos@example.com', '2001-12-14'),
  ('Omar', 'idk6', 'omar@example.com', '2001-12-15'),
  ('Salah', 'idk7', 'salah@example.com', '2001-12-16'),
  ('Foad', 'idk8', 'foad@example.com', '2001-12-17'),
  ('Youssef', 'idk9', 'youssef@example.com', '2001-12-18'),
  ('Zayn', 'idk10', 'zayn@example.com', '2001-12-19');

-- Inserting sample data for Courses
INSERT INTO Courses (course_name, course_description)
VALUES
  ('Mathematics', 'An introductory course on Mathematics'),
  ('Physics', 'Basic principles of Physics'),
  ('Computer Science', 'Fundamentals of Computer Science'),
  ('Web Development', 'Building interactive websites'),
  ('Data Analysis', 'Extracting insights from data');

-- Inserting sample data for Instructors
INSERT INTO Instructors (first_name, last_name, email)
VALUES
  ('Ahmed', 'Essam Azab', 'ahmed.azab@example.com'),
  ('idk1', 'idk4', 'alicebrown@example.com'),
  ('idk2', 'idk3', 'bobwhite@example.com');


-- Inserting sample data for Enrollments
INSERT INTO Enrollments (student_id, course_id, enrollment_date)
VALUES
  (1, 1, '2024-01-01'),
  (2, 2, '2024-01-02'),
  (3, 3, '2024-01-03'),
  (4, 4, '2024-01-04'),
  (5, 1, '2024-01-05'),
  (3, 2, '2024-01-06'),
  (6, 3, '2024-01-07'),
  (7, 4, '2024-01-08'),
  (8, 5, '2024-01-09'),
  (9, 1, '2024-01-10'),
  (10, 2, '2024-01-11'),
  (2, 3, '2024-01-12'),
  (5, 4, '2024-01-13'),
  (4, 5, '2024-01-14'),
  (6, 5, '2024-01-15'),
  (1, 1, '2024-01-16'),
  (7, 3, '2024-01-17'),
  (8, 2, '2024-01-18'),
  (9, 4, '2024-01-19'),
  (11, 3, '2024-01-20');
```

4- Basic Queries

```sql
-- Select all students
SELECT * FROM Students;

-- Select all courses
SELECT * FROM Courses;

-- Select all enrollments with student names and course names
SELECT Enrollments.enrollment_id, Students.first_name, Students.last_name, Courses.course_name, Enrollments.enrollment_date
FROM Enrollments
JOIN Students ON Enrollments.student_id = Students.student_id
JOIN Courses ON Enrollments.course_id = Courses.course_id;
```

120 %

Results Messages

	student_id	first_name	last_name	email	date_of_birth
1	1	Michael	Matta	michaelj@example.com	2001-12-10
2	2	Ahmed	idk	ahmed@example.com	2001-12-11
3	3	Mohamed	idk2	mohamed@example.com	2001-12-12
4	4	Mostafa	idk3	mostafa@example.com	2001-12-12
5	5	Marawan	idk4	marawan@example.com	2001-12-13
6	6	Kerolos	idk5	kerolos@example.com	2001-12-14
7	7	Omar	idk6	omar@example.com	2001-12-15
8	8	Salah	idk7	salah@example.com	2001-12-16

	course_id	course_name	course_description
1	1	Mathematics	An introductory course on Mathematics
2	2	Physics	Basic principles of Physics
3	3	Computer Science	Fundamentals of Computer Science
4	4	Web Development	Building interactive websites
5	5	Data Analysis	Extracting insights from data

	enrollment_id	first_name	last_name	course_name	enrollment_date
1	1	Michael	Matta	Mathematics	2024-01-01
2	2	Ahmed	idk	Physics	2024-01-02
3	3	Mohamed	idk2	Computer Science	2024-01-03
4	4	Mostafa	idk3	Web Development	2024-01-04

5- Advanced Queries

```sql
-- Select students who enrolled in a specific course (i choose math)
SELECT Students.first_name, Students.last_name
FROM Students
JOIN Enrollments ON Students.student_id = Enrollments.student_id
WHERE Enrollments.course_id = 1;

-- Select courses with more than 5 students enrolled
SELECT Courses.course_name, COUNT(Enrollments.student_id) AS num_students
FROM Courses
JOIN Enrollments ON Courses.course_id = Enrollments.course_id
GROUP BY Courses.course_name
HAVING COUNT(Enrollments.student_id) > 5;

-- Update a student's email
UPDATE Students
SET email = 'neweupdatedmail@updated.com'
WHERE student_id = 1;

-- Delete a course that no students are enrolled in
DELETE FROM Courses
WHERE course_id NOT IN (SELECT DISTINCT Enrollments.course_id FROM Enrollments);

-- Calculate the average age of students (assuming current year is 2024)
SELECT AVG(DATEDIFF(YEAR, date_of_birth, GETDATE())) AS avg_age
FROM Students;

-- Find the course with the maximum enrollments
SELECT TOP 1 Courses.course_name, COUNT(Enrollments.student_id) AS num_students
FROM Courses
JOIN Enrollments ON Courses.course_id = Enrollments.course_id
GROUP BY Courses.course_name
ORDER BY num_students DESC;


-- List courses along with the number of students enrolled
SELECT Courses.course_name, COUNT(Enrollments.student_id) AS num_students
FROM Courses
JOIN Enrollments ON Courses.course_id = Enrollments.course_id
GROUP BY Courses.course_name;
```

Results Messages

	first_name	last_name
1	Michael	Matta
2	Marawan	idk4
3	Foad	idk8
4	Michael	Matta

	course_name	num_students

	avg_age
1	23

	course_name	num_students
1	Computer Science	5

	course_name	num_students
1	Computer Science	5
2	Data Analysis	3
3	Mathematics	4
4	Physics	4
5	Web Development	4

6- Join Queries

```sql
-- Select all students with their enrolled courses
SELECT Students.first_name, Students.last_name, Courses.course_name
FROM Students
JOIN Enrollments ON Students.student_id = Enrollments.student_id
JOIN Courses ON Enrollments.course_id = Courses.course_id;

-- List all instructors and their courses.

-- I added a new column so we can find a relation to do the query based on it
ALTER TABLE Enrollments
ADD instructor_id INT;

UPDATE Enrollments
SET instructor_id = CASE
    WHEN course_id = 1 THEN 1
    WHEN course_id = 2 THEN 2
    WHEN course_id = 3 THEN 3
    ELSE 1
END;

SELECT Instructors.first_name, Instructors.last_name, Courses.course_name
FROM Instructors
JOIN Enrollments ON Instructors.instructor_id = Enrollments.instructor_id
JOIN Courses ON Enrollments.course_id = Courses.course_id
GROUP BY Instructors.first_name, Instructors.last_name, Courses.course_name;


-- Find students who are not enrolled in any course
SELECT Students.first_name, Students.last_name
FROM Students
LEFT JOIN Enrollments ON Students.student_id = Enrollments.student_id
WHERE Enrollments.student_id IS NULL;
```

Results | Messages

	first_name	last_name	course_name
1	Michael	Matta	Mathematics
2	Ahmed	idk	Physics
3	Mohamed	idk2	Computer Science
4	Mostafa	idk3	Web Development
5	Marawan	idk4	Mathematics
6	Mohamed	idk2	Physics
7	Kerolos	idk5	Computer Science
8	Omar	idk6	Web Development
9	Salah	idk7	Data Analysis

	first_name	last_name	course_name
1	Ahmed	Essam Azab	Data Analysis
2	Ahmed	Essam Azab	Mathematics
3	Ahmed	Essam Azab	Web Development
4	idk1	idk4	Physics
5	idk2	idk3	Computer Science

	first_name	last_name
1	Zayn	idk10

7- Subqueries and Set Operations

```sql
-- Select students enrolled in more than one course
SELECT Students.first_name, Students.last_name
FROM Students
JOIN Enrollments ON Students.student_id = Enrollments.student_id
GROUP BY Students.first_name, Students.last_name
HAVING COUNT(Enrollments.course_id) > 1;


-- Find courses taught by a specific instructor (here i find MR Ahmed Essam courses)
SELECT Courses.course_name
FROM Courses
JOIN Enrollments ON Courses.course_id = Enrollments.course_id
WHERE Enrollments.instructor_id = 1
GROUP BY Courses.course_name;

-- Select the top 3 students with the most enrollments
SELECT TOP 3 Students.first_name, Students.last_name, COUNT(Enrollments.course_id) AS number_of_enrollments
FROM Students
JOIN Enrollments ON Students.student_id = Enrollments.student_id
GROUP BY Students.first_name, Students.last_name
ORDER BY number_of_enrollments DESC;

-- Combine results of students enrolled in more than one course and courses with more than one student enrolled
SELECT 'Student' AS type, Students.first_name, Students.last_name, NULL AS course_name
FROM Students
JOIN Enrollments ON Students.student_id = Enrollments.student_id
GROUP BY Students.first_name, Students.last_name
HAVING COUNT(Enrollments.course_id) > 1
UNION
SELECT 'Course' AS type, NULL AS first_name, NULL AS last_name, Courses.course_name
FROM Courses
JOIN Enrollments ON Courses.course_id = Enrollments.course_id
GROUP BY Courses.course_name
HAVING COUNT(Enrollments.student_id) > 1;
```

8- Functions and Stored Procedures:

```sql
-- Create a stored procedure to add a new student
CREATE PROCEDURE AddNewStudent
    @first_name VARCHAR(100),
    @last_name VARCHAR(100),
    @email VARCHAR(255),
    @date_of_birth DATE
AS
BEGIN
    INSERT INTO Students (first_name, last_name, email, date_of_birth)
    VALUES (@first_name, @last_name, @email, @date_of_birth);
END;
GO

-- Create a function to calculate the age of a student based on their date of birth
CREATE FUNCTION CalculateStudentAge (@date_of_birth DATE)
RETURNS INT
AS
BEGIN
    RETURN DATEDIFF(YEAR, @date_of_birth, GETDATE()) -
           CASE
               WHEN DATEADD(YEAR, DATEDIFF(YEAR, @date_of_birth, GETDATE()), @date_of_birth) > GETDATE()
               THEN 1
               ELSE 0
           END;
END;
GO
```

9- Aggregate Functions and Grouping

```sql
-- Calculate the total number of students
SELECT COUNT(*) AS total_students
FROM Students;

-- Calculate the average, minimum, and maximum number of enrollments per course
SELECT
    AVG(enrollment_count) AS average_enrollments,
    MIN(enrollment_count) AS minimum_enrollments,
    MAX(enrollment_count) AS maximum_enrollments
FROM (
    SELECT course_id, COUNT(student_id) AS enrollment_count
    FROM Enrollments
    GROUP BY course_id
) AS course_enrollments;
```

120 %

Results | Messages

	total_students
1	11

	average_enrollments	minimum_enrollments	maximum_enrollments
1	4	3	5

10- Additional Tasks

```sql
-- Select all students with aliases for complex column names
SELECT
    Students.first_name AS SFN,
    Students.last_name AS SLN,
    Courses.course_name AS CCN
FROM
    Students
JOIN
    Enrollments ON Students.student_id = Enrollments.student_id
JOIN
    Courses ON Enrollments.course_id = Courses.course_id;

--Use CASE to categorize students based on their age.
SELECT
    first_name AS SFN,
    last_name AS SLN,
    date_of_birth,
    CASE
        WHEN DAY(date_of_birth) <= 10 THEN 'Early birth'
        WHEN DAY(date_of_birth) BETWEEN 11 AND 15 THEN 'Mid birth
        ELSE 'Late birth'
    END AS birth_category
FROM
    Students;

-- Use EXISTS to find courses with at least one enrolled student
SELECT course_name
FROM Courses
WHERE EXISTS (
    SELECT course_id
    FROM Enrollments
    WHERE Enrollments.course_id = Courses.course_id
);

--Create comments in SQL for clarity
/*hello world :)
this is a comment
good bye*/
```

Results | Messages

	SFN	SLN	CCN
1	Michael	Matta	Mathematics
2	Ahmed	idk	Physics
3	Mohamed	idk2	Computer Science
4	Mostafa	idk3	Web Development
5	Marawan	idk4	Mathematics
6	Mohamed	idk2	Physics
7	Kerolos	idk5	Computer Science
8	Omar	idk6	Web Development

	SFN	SLN	date_of_birth	birth_category
1	Michael	Matta	2001-12-10	Early birth
2	Ahmed	idk	2001-12-11	Mid birth
3	Mohamed	idk2	2001-12-12	Mid birth
4	Mostafa	idk3	2001-12-12	Mid birth
5	Marawan	idk4	2001-12-13	Mid birth
6	Kerolos	idk5	2001-12-14	Mid birth
7	Omar	idk6	2001-12-15	Mid birth
8	Salah	idk7	2001-12-16	Late birth

	course_name
1	Mathematics
2	Physics
3	Computer S...
4	Web Devel...
5	Data Analysis