CS405 Assignment 1 Solution - Fall 2026

Question 1: Entity Relationship Diagram

Entities and their key attributes:

Entity	Key Attributes	Other Attributes
Learner	Learner_ID (Primary Key)	Learner_Name
Course	Course_ID (Primary Key)	Course_Title, Required_Specialization
Instructor	Instructor_ID (Primary Key)	Instructor_Name, Specialization
Enrollment (Associative Table)	Learner_ID + Course_ID (Composite Primary Key)	Enrollment_Date
CourseAssignment (Associative Table)	Instructor_ID + Course_ID (Composite Primary Key)	

Relationships with Cardinality

Relationship	Cardinality	Meaning
Learner to Enrollment	1 to N	One learner can have many enrollments; each enrollment belongs to one learner.
Course to Enrollment	1 to N	One course can have many enrollments; each enrollment belongs to one course.
Instructor to CourseAssignment	1 to N	One instructor can be assigned to many courses; each assignment belongs to one instructor.
Course to CourseAssignment	1 to N	One course can be assigned to many instructors; each assignment belongs to one course.

Many-to-Many Relationships and Their Resolution

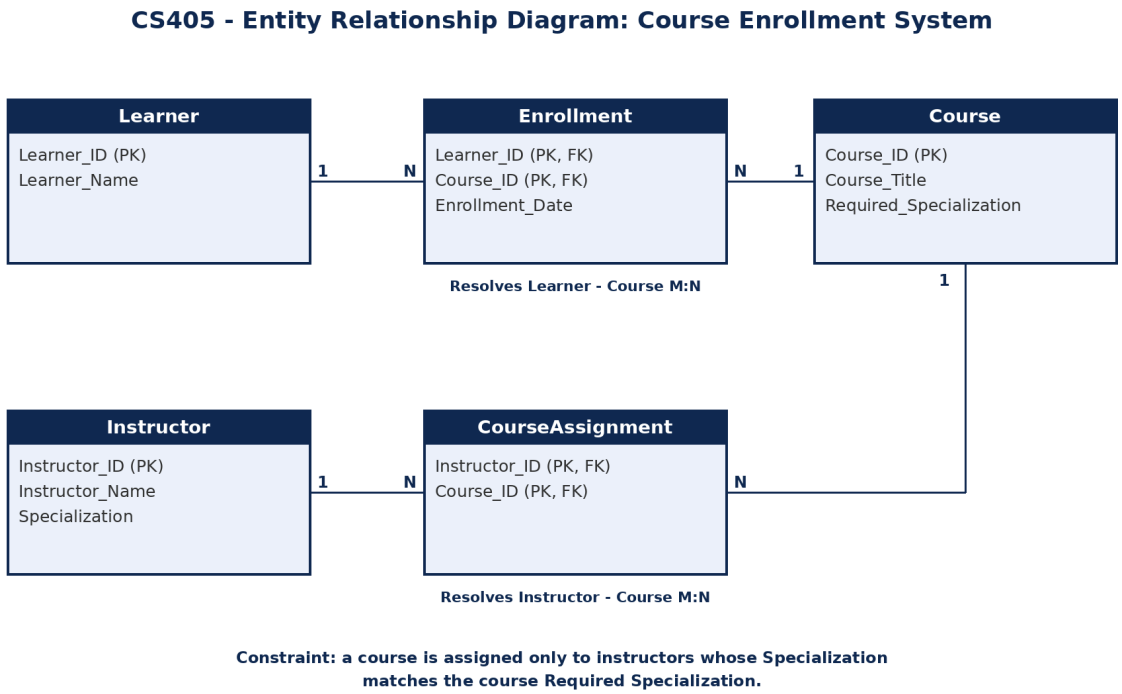
1. Learner and Course have a many-to-many relationship: one learner can enroll in many courses, and one course can have many learners. It is resolved by the associative table Enrollment, whose composite primary key is Learner_ID + Course_ID.

2. Instructor and Course have a many-to-many relationship: one instructor can teach many courses, and one course can be assigned to many instructors. It is resolved by the associative table CourseAssignment, whose composite primary key is Instructor_ID + Course_ID.

Specialization Matching Constraint

A course must be assigned only to instructors with a matching specialization. In this design, the Instructor table stores Specialization and the Course table stores Required_Specialization, and an assignment in CourseAssignment is valid only when the instructor Specialization matches the course Required Specialization.

Entity Relationship Diagram



Question 2: SQL Query

List the course title and the number of learners enrolled in each course, showing only courses that have more than 2 enrolled learners, sorted by the enrollment count in descending order:

```
SELECT c.Course_Title, COUNT(DISTINCT e.Learner_ID) AS Total_Learners
FROM Course c INNER JOIN Enrollment e ON c.Course_ID = e.Course_ID
GROUP BY c.Course_ID, c.Course_Title HAVING COUNT(DISTINCT
e.Learner_ID) > 2 ORDER BY Total_Learners DESC;
```

How it works: the INNER JOIN connects each course to its enrollments through Course_ID. GROUP BY groups the rows per course. COUNT(DISTINCT Learner_ID) counts each learner once per course. HAVING keeps only courses with more than 2 learners. ORDER BY Total_Learners DESC sorts the result from the highest enrollment count to the lowest.