**Database Design Question:**

**Scenario**: You are tasked with designing a database for a university exam system. The system must track **Teachers**, **Students**, **Courses**, **Questions**, and **Exams**.

* A **Teacher** teaches many **Courses**, and each **Course** is taught by many **Teachers**
* A **Student** enrolls in many **Courses** and can take many **Exams**.
* A **Course** contains many **Questions**, and each **Question** belongs to one **Course**.
* **Exams** are created using **Questions** from the respective course.
* **Students** must be enrolled in a **Course** to be eligible to take the **Exam**.
* **A student’s result for an exam** is tracked via a relationship that captures the **Score** and **Status**.

**Requirements:**

1. **Student**: Must be able to enroll in courses, take exams, and have results stored for each exam.
2. **Course**: Contains questions, and each course can have many students.
3. **Exam**: Formed using **Questions**, and can have multiple students who take the exam.
4. **Exam Results**: Must capture the student’s performance in each exam, including **Score** and **Status**.

**Your Task:**

1. Design an **Entity-Relationship Diagram (ERD)** to model the system.
   * Represent all the entities: **Teacher**, **Student**, **Course**, **Question**, **Exam**, and the relationship between them.
   * Show the relationships between entities with cardinalities and indicate where **many-to-many** relationships exist.