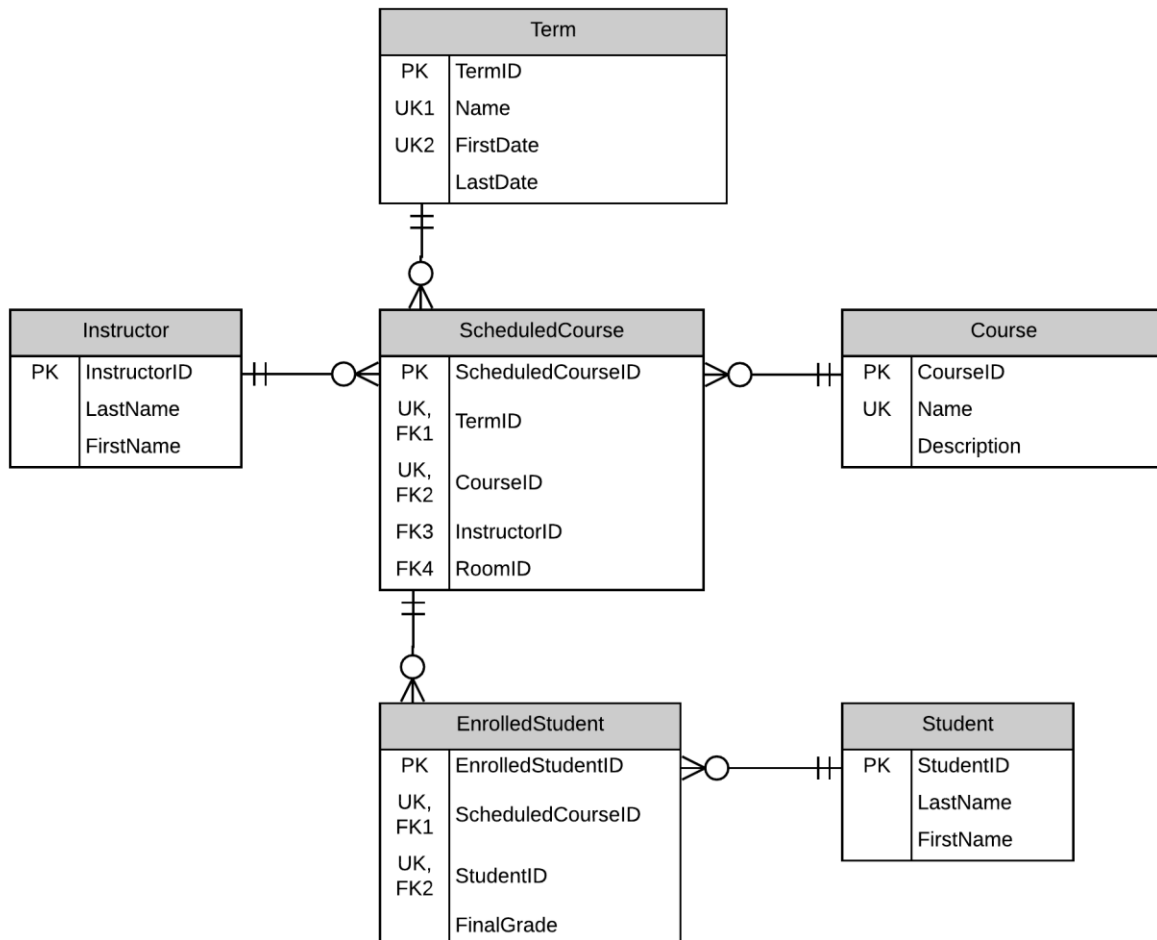


Exam 1 – Practice A Key

Database Design

The questions on this exam use tables from the below diagram. For the most part, the complete column specifications are unimportant for the purposes of this exam, but make note of the following:

- All columns in the database do not allow nulls. That is, they are defined with NOT NULL.
- All primary key columns and the referencing foreign key columns are of type INT.
- All Name columns (including LastName and FirstName) in all tables are of type NVARCHAR.
- The FinalGrade column in the EnrolledStudents table is of type DECIMAL(5, 2). This simply means it supports values from 0.00 to 100.00, such as 79.5 and 90.25.



1. Write a query to return all students enrolled in either CIS560 or CIS562 for the term of Fall 2018. "Fall 2018" will appear as a Name in the Term table, and "CIS560", for example, is the Name in Course.

Required Result Columns

LastName – The last name of the student as it appears in LastName of Student.

FirstName – The first name of the student as it appears in FirstName of Student.

StudentID – The identifier of the student as it appears in StudentID of Student.

CourseName – The name of the course as it appears in Name of Course.

Implementation Requirements

The results should be sorted by LastName, then FirstName, and finally StudentID all in ascending order.

```
SELECT S.LastName, S.FirstName, S.StudentID, C.Name AS CourseName
FROM Term T
    INNER JOIN ScheduledCourse SC ON SC.TermID = T.TermID
    INNER JOIN Course C ON C.CourseID = SC.CourseID
    INNER JOIN EnrolledStudent ES ON ES.ScheduledCourseID = SC.ScheduledCourseID
    INNER JOIN Student S ON S.StudentID = ES.StudentID
WHERE T.Name = N'Fall 2018'
    AND C.Name IN (N'CIS 560', N'CIS 562')
ORDER BY S.LastName ASC, S.FirstName ASC, S.StudentID ASC;
```

2. Write a query to return all instructors who are **not** teaching during the Fall 2018 term. "Fall 2018" will appear as the Name in the Term table to identify the Fall 2018 term.

Required Result Columns

InstructorID – The identifier of the instructor as it appears in InstructorID of Instructor.

LastName – The last name of the instructor as it appears in LastName of Instructor.

FirstName – The first name of the instructor as it appears in FirstName of Instructor.

Implementation Requirements

- The solution may only use joins. That is, it should contain no subqueries.
- Each table in your solution should be referenced only once.
- The results should be sorted by LastName, then FirstName, and finally InstructorID all in ascending order.

```
SELECT I.InstructorID, I.LastName, I.FirstName
FROM Instructors I
  LEFT JOIN ScheduledCourse SC ON SC.InstructorID = I.InstructorID
  LEFT JOIN Term T ON T.TermID = SC.TermID
  AND T.Name = N'Fall 2018'
WHERE T.TermID IS NULL
ORDER BY I.LastName ASC, I.FirstName ASC, I.InstructorID ASC;
```

3. Write a query as an alternate solution to *Question 2*. This solution **should use a subquery**, which can either be self-contained or correlated. All other requirements are the same as *Question 2*, including that **each table should be referenced only once**.

-- Solution with self-contained subquery

```
SELECT I.InstructorID, I.LastName, I.FirstName
FROM Instructors I
WHERE I.InstructorID NOT IN
(
  SELECT SC.InstructorID
  FROM Term T
    INNER JOIN ScheduledCourse SC ON SC.TermID = T.TermID
  WHERE T.Name = N'Fall 2018'
)
ORDER BY I.LastName ASC, I.FirstName ASC, I.InstructorID ASC;
```

-- Solution with correlated subquery

```
SELECT I.InstructorID, I.LastName, I.FirstName
FROM Instructors I
WHERE NOT EXISTS
(
  SELECT *
  FROM ScheduledCourse SC
    INNER JOIN Term T ON T.TermID = SC.TermID
  WHERE SC.InstructorID = I.InstructorID
    AND T.Name = N'Fall 2018'
)
ORDER BY I.LastName ASC, I.FirstName ASC, I.InstructorID ASC;
```