

**Q-1**

Explanation of ER

* 20bcs077\_Levels

Strong Entity

Level - PK

* 20bcs077\_Pool

Strong Entity

Pool - PK

* 20bcs077\_Staff

Strong Entity

StaffID - PK

* 20bcs077\_Classes

Strong Entity

LessonIndex - PK

Level, Pool, Instructor - FK

* 20bcs077\_Enrollment

Weak Entity

SID - FK

* 20bcs077\_Students

Strong Entity

SID - PK

**Q-2**

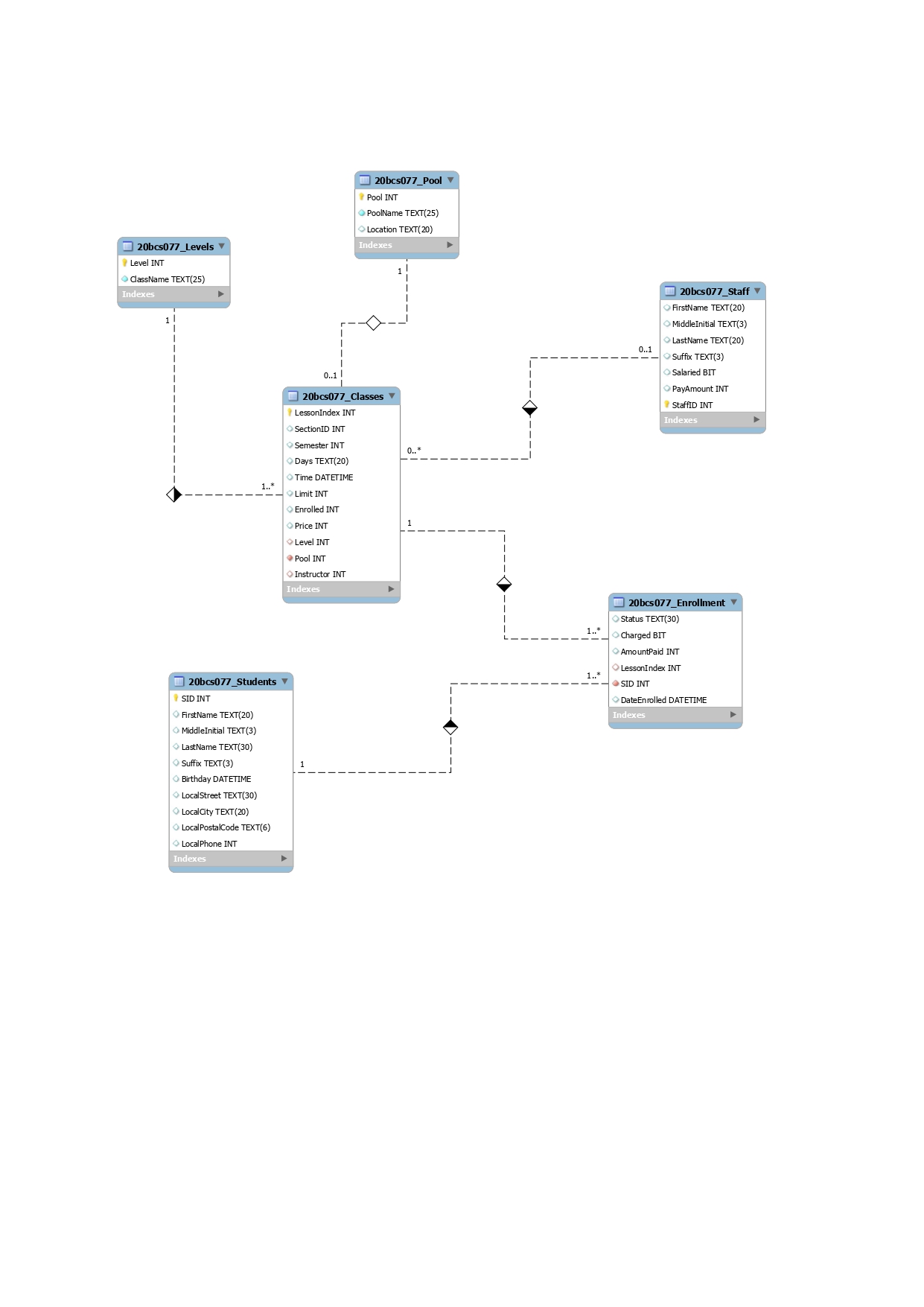
**Cardinality**

| **Cardinality** | **Entity1** | **Entity2** |
| --- | --- | --- |
| **One to Many** | **Pool** | **Classes** |
| **One to Many** | **Level** | **Classes** |
| **One to Many** | **Staff** | **Classes** |
| **One to Many** | **Classes** | **Enrollment** |
| **One to Many** | **Student** | **Enrollment** |

**Degree**

| **Entity** | **Degree** |
| --- | --- |
| **Level** | **1** |
| **Pool** | **1** |
| **Enrollment** | **2** |
| **Staff** | **1** |
| **Classes** | **4** |
| **Students** | **1** |

**Q-3**

****

**Q-4**

In Enrollment there is no Primary key so it is Weak Entity and we can make it strong entity either by making one of key as Primary key or by adding one more table to it.

We can use the Composite key as {SID, LessonIndex} which becomes Primary Key or we can add a new attribute named “enrollId” and assign it as Primary Key.

**Q-5**

Data redundancy is defined as the storing of the same data in multiple locations.

In our case:

1. In the Classes table we have the same data for sectionID, Semester for a lot of different Lesson Index which leads to repetition of SectionID and SemesterID again and again. We can remove it by creating a new table which handles Lesson Index and doesn't allow further anomalies to happen.
2. For multiple subjects data like date, time and limit might be the same so we can create a new table which handles these attributes.