1)

An Entity-Relationship diagram with the following entities and attributes:

20BCS056_Pool — PoolName, Location

20BCS056_Levels — ClassName
- connected via relationship **Level** to 20BCS056_Classes

20BCS056_Classes — Enrolled, Time, Limit, SectionID, Semester, Price, Days
- connected via relationship **Pool** to 20BCS056_Pool
- connected via relationship **Instructed by** to 20BCS056_Staff
- connected via relationship **StudentID** to 20BCS056_Students

20BCS056_Staff — PayAmount, StaffID, FirstName, MiddleInitial, LastName, Suffix, Salaried

20BCS056_Enrollment — AmountPaid, DateEnrolled, Charged, LessonIndex, StudentID, Status
- connected via relationship **StudetID** to 20BCS056_Students

20BCS056_Students — LocalPostalCode, MiddleInitial, LocalPhone, FirstName, LastName, LocalCity, Suffix, LocalStreet, Birthday
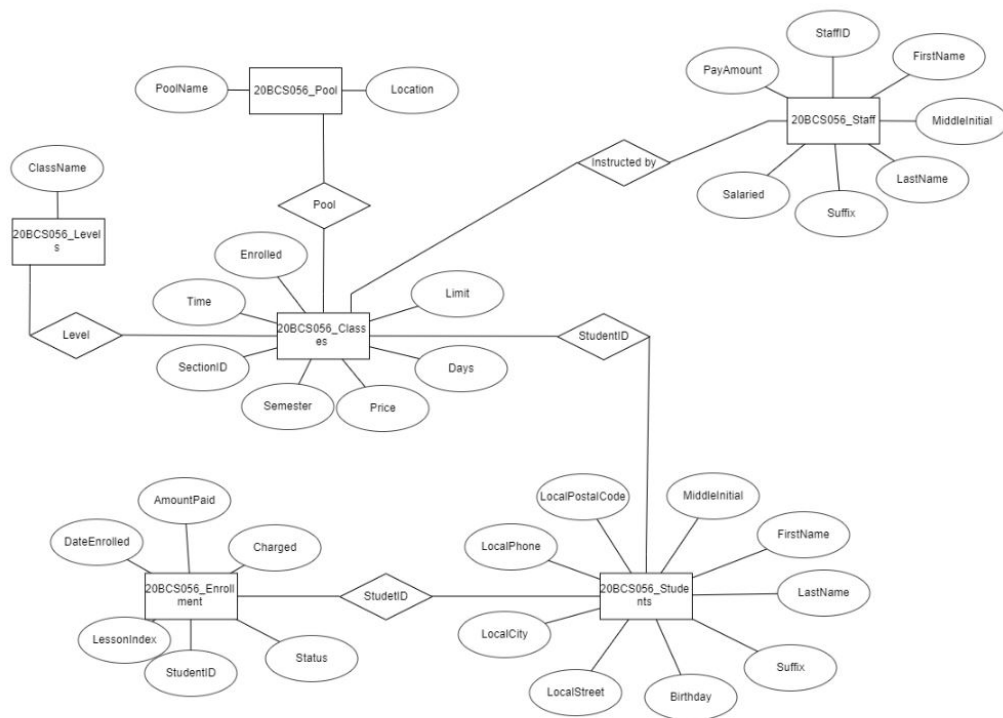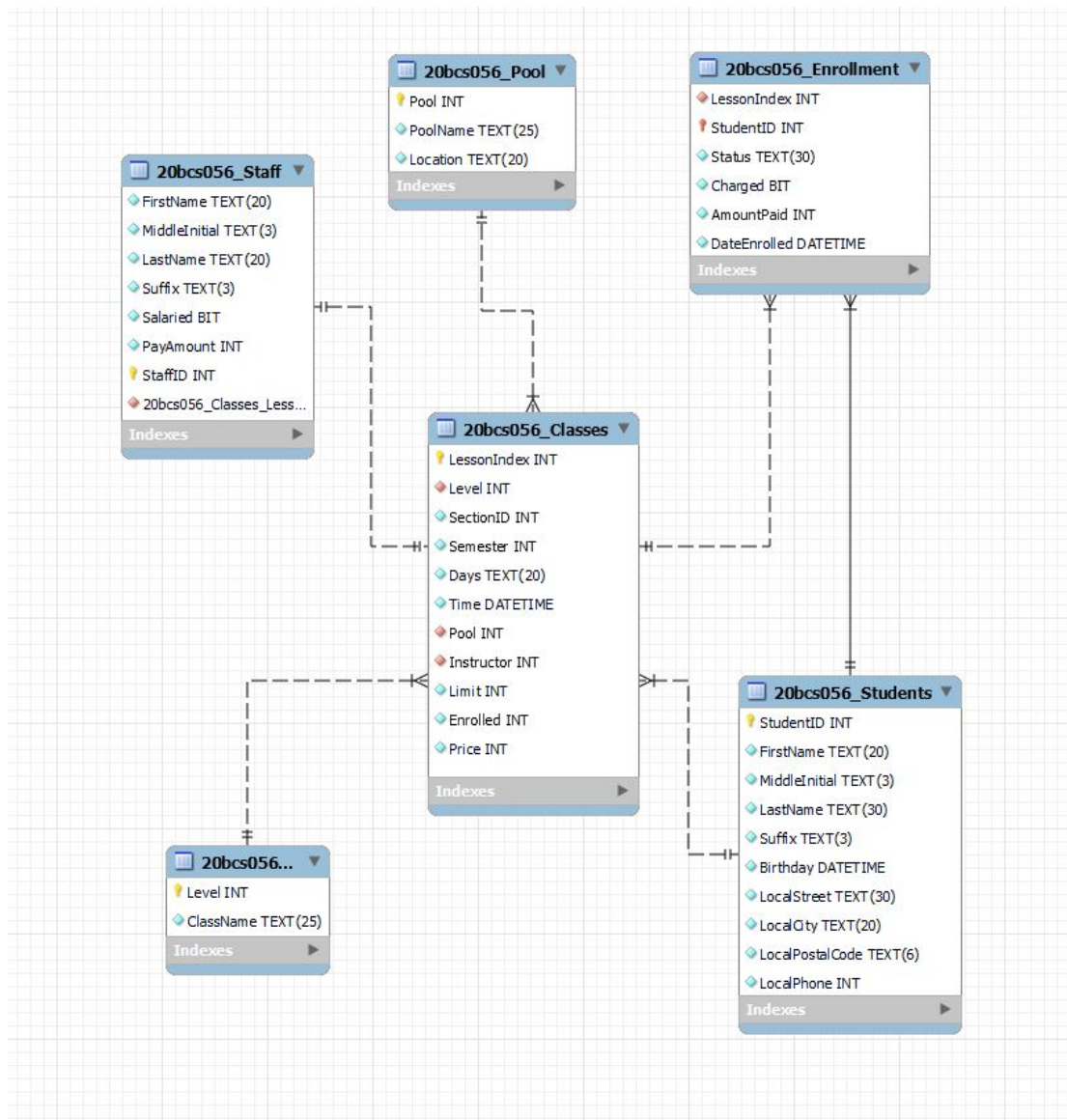
2) All the relationship in the above ERD iagram are binary relationships. the cardinality of these relationships is as follows:

      Students-Enrollment : 1 to many

      Enrollment-Classes : Many to 1

      Classes-Staff : Many to 1

      Classes-Pool: Many to 1

      Classes-Level: Many to 1

3) As we can see from the above diagram 1 inst can take many class but it is mandatory that each class have 1 instructor while the participation of staff is optional I.e. it is not necessary that an instructor has a class assigned to him.

Similarly a pool can be used to conduct many classes but it is mandatory for a class to be conducted inside a pool. There can be a pool where no classes are conducted hence it's participation in the relationship is optional.

A class has to be on some level (beginner, intermediate, advanced etc...) and there will be at least one class for every level.

In the above diagram we can see that enrollment is a weak entity as it has no unique primary key. One Student can enroll into many classes and many students can enroll for the same class. So we cannot use student id or class id to identify enrollment. So we use a composite key comprising of two foreign keys StudentID and LessonIndex and assign it as primary key. Thus we can give each enrollment a unique id and convert it into strong entity by identifying it using two strong entities.

Since a student has to enroll and a class must have enrolled students we can infer that a class will have at least one student.

There can be a situation when one class is conducted more than once in a day for the same group of students. It is possible that due to timing clashes that these classes are conducted by the same instructor in different pools or by different instructors in different pool or by different instructors in different pools. In any case there will be multiple records of that class with a change in only two or three fields. Due to this any updation in record might be tricky. This can be solved by creating another entity which can be identified through a combination of instructor id and lesson id.

4) Weak entity is absent here, as table has a primary key. So, each table can be identified individually. So, each table is a strong entity.

5)

20bcs056_Staff
- FirstName TEXT(20)
- MiddleInitial TEXT(3)
- LastName TEXT(20)
- Suffix TEXT(3)
- Salaried BIT
- PayAmount INT
- StaffID INT
- 20bcs056_Classes_Less...
- Indexes

Session
- LessonIndex INT
- StaffID INT
- Days TEXT(20)
- Pool INT
- Time DATETIME
- Indexes

20bcs056_Pool
- Pool INT
- PoolName TEXT(25)
- Location TEXT(20)
- Indexes

20bcs056_Classes
- LessonIndex INT
- Level INT
- SectionID INT
- Semester INT
- Days TEXT(20)
- Time DATETIME
- Pool INT
- Instructor INT
- Limit INT
- Enrolled INT
- Price INT
- Indexes

20bcs056_Enrollment
- LessonIndex INT
- StudentID INT
- Status TEXT(30)
- Charged BIT
- AmountPaid INT
- DateEnrolled DATETIME
- Indexes

20bcs056...
- Level INT
- ClassName TEXT(25)
- Indexes

20bcs056_Students
- StudentID INT
- FirstName TEXT(20)
- MiddleInitial TEXT(3)
- LastName TEXT(30)
- Suffix TEXT(3)
- Birthday DATETIME
- LocalStreet TEXT(30)
- LocalCity TEXT(20)
- LocalPostalCode TEXT(6)
- LocalPhone INT
- Indexes