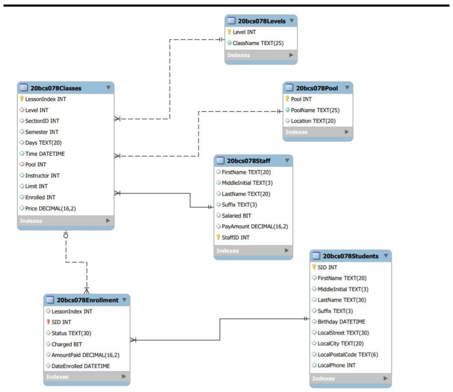
DBMS CLASS HACKATHON

Name: M Manjunath
Roll no.: 20bcs078

Entity-Relationship diagram:



All the relationships in the above ERD diagram are binary relationships.

cardinality of these relationships is as follows:

. Students-Enrollment: One to Many

• Enrollment-Classes: Many to One

• Classes-Staff: Many to One

Classes-Pool: Many to One

Classes-Level: Many to One

As we can see from the above diagram one instructor can take many classes but it is mandatory that each class have an instructor (from staff) while the participation of staff is optional i.e. it is not necessary that an instructor has a class assigned to him/her.

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 SID 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

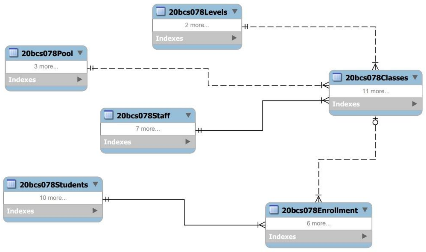
infer that a class will have at least one student.

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.

Code

```
CREATE TABLE 20bcs078Levels(
Level INTEGER PRIMARY KEY,
ClassName TEXT(25) NOT NULL);
```

```
CREATE TABLE 20bcs078Pool(
Pool INTEGER PRIMARY KEY,
ClassName TEXT(25) NOT NULL,
Location TEXT(20));

CREATE TABLE 20bcs078Staff(
StaffID INTEGER PRIMARY KEY,
FirstName TEXT(20),
MiddleInitial TEXT(3),
LastName TEXT(20),
Suffix TEXT(3),
Salaried BIT,
PayAmount INTEGER);

CREATE TABLE 20bcs078Students(
SID INTEGER PRIMARY KEY,
FirstName TEXT(20),
MiddleInitial TEXT(3),
LastName TEXT(30),
Suffix TEXT(3),
Birthday DATETIME,
LocalStreet TEXT(30),
LocatCity TEXT(20),
LocalPostalCode TEXT(6),
LocalPhone INT);

CREATE TABLE 20bcs078Classes(
LessonIndex INTEGER PRIMARY KEY,
Level INTEGER,
SectionID INTEGER,
Semester INT,
Days TEXT(20),
Time DATETIME,
Pool INTEGER,
Instructor INTEGER,
ClassLimit INT,
Enrolled INT,
Price INTEGER,

FOREIGN KEY (Level) REFERENCES 20bcs078Levels(Level),
FOREIGN KEY (Pool) REFERENCES 20bcs078Pool(Pool),
FOREIGN KEY (Instructor) REFERENCES 20bcs078Staff(StaffID));

CREATE TABLE 20bcs078Enrollment(
LessonIndex INTEGER,
SID INTEGER PRIMARY KEY,
Status TEXT(30),
Charged BIT,
AmountPaid INTEGER,
DateEnrolled DATETIME,

CONSTRAINT Enrollment_FK1 FOREIGN KEY (SID) REFERENCES
20bcs078Classes(LessonIndex),
CONSTRAINT Enrollment_FK2 FOREIGN KEY (SID) REFERENCES
20bcs078Students(SID)
);
```


20bcs078Levels
2 more...
Indexes
20bcs078Pool
3 more...
Indexes
20bcs078Classes
11 more...
Indexes
20bcs078Staff
7 more...
Indexes
20bcs078Students
10 more...
Indexes
20bcs078Enrollment
6 more...
Indexes