(*) Schema

1) zobcs033Levels (level, className)

2) zobcs033Pool (Pool, PoolName, location)

3) zobcs033Staff (firstName, middleName, lastName, Suffix, salaried, Payamount, staffID)

4) zobcs033Classes (lessonIndex, level, SectionID, Semester, days, Time, Rol, Instructor, limit, Enrolled, Price)

5) zobcs033Enrollment (lessonIndex, SID, Status, Charged, AmountPaid, DateEnrolled)

6) zobcs033Students (SID, firstName, middleName, LastName, Suffix, birthday, localstreet, localcity, localPostalCode, localPhone)

20bcs035

weak entities

- The entities having no primary key are called as weak entities

- in this DBMS model The entity (20bcs035Enrollment) is weak entity.

- to make this entity strong we have to create a composite key which will act as primary key.

- The composite key will be (SID, dateEnrolled) which is unique for every tuple.

Relation	cardinality.
① 20bcs034levels	4
② 20bcs034Pool	3
③ 20bcs034Staff	7
④ 20bcs034classes	12
⑤ 20bcs034Enrollment	6
⑥ 20bcs034Students	10

Date

data redendency Scenarios:

- in the relation zobcs035classes the classes are assigned to the ~~Pool~~ entities Pool, students and staff if we delete a tuple for class we will loose the information about the Payment and the Pool also

- This redendency can be corrected by by dividing classes into 2 or more relations.

- have to add a primary key constraint to Enrollment entity as its a weak entity can cause data redendency

- in student and staff entities we can make a single column of name mearging other 3 which are related to name, it will be more feasible

⑥ classes ++ —— o| levels

1 to 1 relation.

⑦ students ++ —— o< Enrollments

1 to many relations.

Relationships between entities.

① Pool ++ —— o< classes

1 to 1 relation

② levels ++ —— o< classes

1 to many relatio

③ Staff ++ —— o< classes

1 to 1 relation

④ ~~students~~ ++ —— o< classes

Enrollment

1 to many relation

⑤ classes ++ —— ++ Pool

1 to 1