**People Table**

The ***People*** table will hold basic information such as First and Last names, Email, and a PID to identify a user. It will have a one to one relationship with the ***Students*** and ***Admins*** tables.

**Admins Table**

The ***Admins*** table will hold information about the admins. It will store their Username, Password and an AID which will be taken from the ***People’s*** PID.

**Students Table**

The ***Students*** table will store only the Birthday of the student to be used to determine if prospective students will be considered a regular student or an adult student. It will also store a SID which will be taken from the ***People’s*** PID.

**Sessions Table**

In our design, we are going to have session that saves a randomly generated number with a student’s email address so they can log back into a session without losing all of the work they have done so far. The **Sessions Table** stores that generated number with the student’s email address. This table has a one to one relationship with ***Students***.

**Student\_Classes Table**

The ***Student\_Classes*** table will hold the information connecting a student to a class. This is because many classes can have many students and many student can have many classes. This table will store the DID (Dutchess ID) with the SID (Student ID).

**D\_Classes Table**

The ***D\_Classes*** table will hold all information about Dutchess Community College courses. Subject, Credits, Course number, and Course name will all be stored along with a DID to identify classes. It will have a one to many relationship with ***Has\_Taken*** and ***Cred\_Transfers***.

**Cred\_Transfers Table**

The ***Cred\_Transfers*** tablewill hold all the information connecting a Marist class to a Dutchess Class. This will match the classes that transfer from Dutchess to Marist. This table will store DID (Dutchess ID) and MarID (Marist ID). It will act as a composite table since it has a many to one relationship with ***M\_Classes*** and ***D\_Classes***.

**M\_Classes Table**

The ***M\_Classes*** table will hold all information about Marist College courses. Subject, Course number, Credits, and Course name will all be stored along with a MarID to identify classes. It will have a one to many relationship ***Cred\_Transfers***. It also has a many to one relationship with ***M\_Minors Table*** and ***Major\_Classes Table.*** This means that a marsit course can count towards several different Majors and Minors.

**M\_Minors Table**

The ***M\_Minors Table*** will hold information about Minors at Marist. It will store the Minor name and the school the Minor belongs to. It will use a MinID to identify Minors. This table will have a many to one relationship with ***Minor\_Classes*** because a class can be part of several Minors.

**Minor\_Classes Table**

The ***Minor\_Classes*** tablewill hold information about classes that count towards specific Minors at Marist. It uses a composite key of MarID and MinID from the ***M\_Minors*** table to the ***M\_Classes*** table. The ***M\_Classes*** table has a one to many relationship with the ***M\_Minors*** table and the ***M\_Classes*** table this is because many classes can have many minors and many minors can have many classes. We need the ***Minor\_Classes*** composite table to denote this since you can not have a many to many relationship without one.

**M\_Majors Table**

The ***M\_Majors*** table will hold information about Majors at Marist. It will store the Major name and the school the Major belongs to. It will use a MJID to identify Majors. This table will have a many to one relationship with ***Major\_Classes*** because a class can be part of several Majors.

**Major\_Classes Table**

The ***Major\_Classes*** tablewill hold information about classes that count towards specific Majors at Marist. It uses a composite key of MarID and MJID from the ***M\_Majors*** table to the ***M\_Classes*** table. The ***M\_Classes*** table has a one to many relationship with the ***M\_Majors*** table and the ***M\_Classes*** table this is because many classes can have many majors and many majors can have many classes. We need the ***Major\_Classes*** composite table to denote this since you can not have a many to many relationship without one.