Table 1: Roles
Brief Explanation of Name: This table will include information about any user that interacts with any CS help queue, whether they are a student or TA. Users can be a student in some classes and a TA for others.
Entity Represented: User - a human at BYU that has some involvement with CS classes
Evidence of Normalization: We were originally going to have a name column as well for displaying on the front end. However, we realized that NetID and Name are functionally dependent, so we decided to just cut the Name column entirely. We considered a separate name table but now plan to investigate using existing BYU API's instead of storing the data ourselves.
Column Names:

NetID: a user's BYU NetID, part of the composite key
CourseID: an identifier for a specific CS class, part of the composite key
Role: the role the student (should be either TA or student)

Schema:
Roles(NetID, CourseID, Role)

- Foreign Key CourseID references Courses

Table 2: HelpInstances
Brief Explanation of Name: A record of all student/TA interactions
Entity Represented: This table holds representations of every time a student gets in line on the queue.
Evidence of Normalization: None of the columns are functionally dependent on anything but our internal ID.
Column Names:

ID: an internal ID, primary key
CourseID: an internal ID for identifying courses
StudentNetID: the BYU NetID of a student
QuestionText: the textual description of the question provided by the student
EnqueueTime: the time the student joins the queue
DequeueTime: the time the student is removed from the queue
StartHelpTime: the time the TA begins helping the student
TANetId: the BYU NetID of the TA helping the student

Schema:
HelpInstances(ID, CourseID, StudentNetID, QuestionText, EnqueueTime, DequeueTime, StartHelpTime, TANetId)

- Foreign Key StudentNetID, CourseID references Roles
- Foreign Key TANetID, CourseID references Roles

Table 3: Courses
Brief Explanation of Name: A list of all CS courses with a queue and corresponding settings
Entity Represented: A CS course

Evidence of Normalization: None of the columns are functionally dependent on anything but our internal Course ID. We considered using CourseName as the identifier, but the course change could change (ex: 401R becomes 474) and we didn't want to make it difficult to update names.
Column Names:

CourseID: An internal object identifier that uniquely identifies a course
CourseName: The name of a course (e.x. CS142)
QueueOpen: A value indicating if the queue is currently open

Schema:
Courses(CourseID, CourseName, QueueOpen)

How the Entities Relate:
Users are associated with Courses. A User can either be a student in a course, or a TA in a course. A user can be a TA in some courses and a student in others. Each course has a collection of help instances associated with it that include information about the TA and student involved in the interaction. The time information is used by the front end to display statistics such as the average wait time, or which TA's have helped the most students. The settings for each CS course help queue vary slightly so the Courses table keeps track of these differences. Each individual queue can be turned on and off as necessary. The Course table also stores the external name of the course.