Duke Conversations

CS 316

Noah Burrell, Anne Driscoll, KImberly Eddleman, Summer Smith, Sarp Uner

Application

The Problem

Duke Conversations is a program run through the provost's office, not through the UCAE, that holds dinners twice or three times a dinner with faculty members, as a way to bring together the Duke community. The goal of the program is to create small group interaction between faculty members and students, and build an atmosphere of academic engagement.

Currently, the program accepts dinner applications through google forms, where individuals on the planning committee accept or reject applicants.

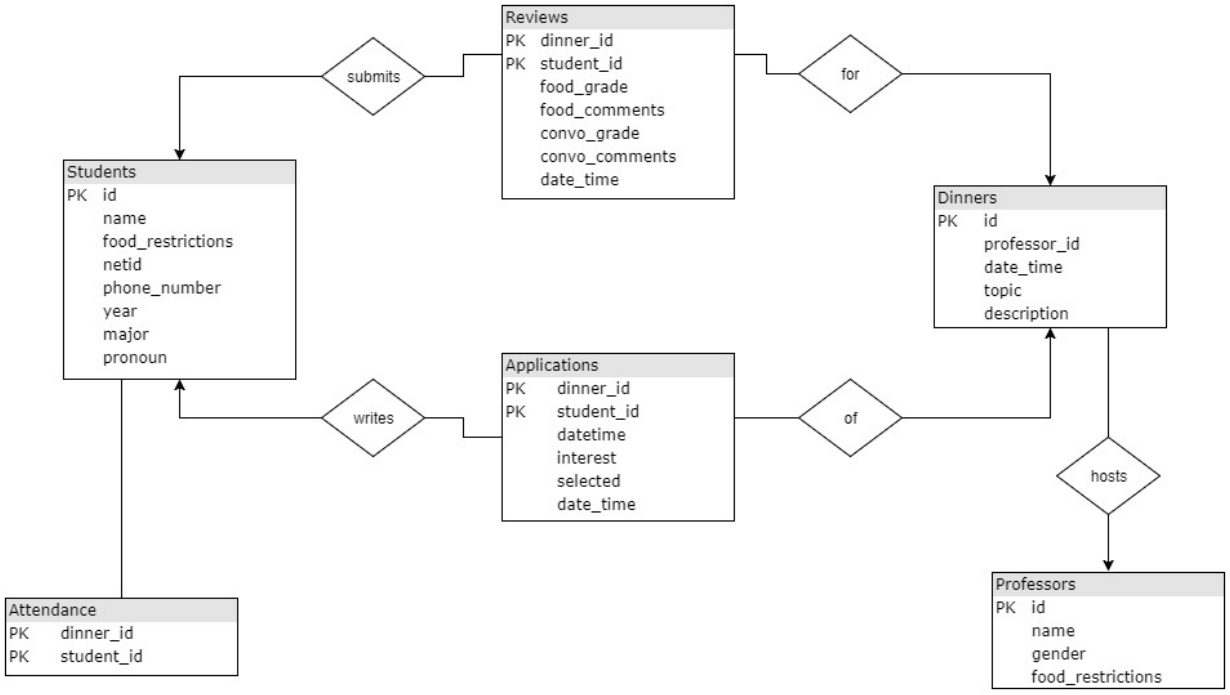
The App

The application we intend to build will help the organizing team keep track of attendees. By keeping count of how many times each person has applied, and if they've attended previous dinners, we'll be able to ensure that the selection process is standardized, and the team won't have to try to remember the names of all the people that went to (or applied to) dinners that they didn't manage.

To do that, we'll have to create a login process for both sets of parties, a way for people to apply, and a way for planners to select applicants.

Database Design

The ER diagram for the database is provided below. Several changes have been made from the original schema implemented. Several fields have been changed to help the team better pursue their data needs, including adding enhanced information in the reviews, additional information on professors, and eliminating unnecessary fields.



Note that referential integrity constraints are necessary, as, for example, a non-existent student in the database should not be able to submit an application for a non-existent dinner. For more likely definitions, an existent student should not be able to submit a review for an existent dinner that they did not attend. In the system used in the first report, we implemented psql triggers to enforce existence conditions. In the section discussing platform choices this will be further discussed.

Data

We have compiled the applications from a few of the dinners last semester, totaling 600 applications. This sample set includes variables for all the past questions, with NA's for the applications that were not asked that question originally. As questions have been phased in over time, there are many individuals that have answered different questions at different times. For occasions where an individual has given different answers at different times, the most recent answer was chosen.

We are currently using a simple test database while we build the platform. This database contains several items in each table, and was generated to resemble the types of inputs that we should see in reality.

Platform Choice

Overall Implementation

We have chosen to implement the web app in python using Django. We chose Django over Flask because it provides many of the features we will need built in. Django allows us out of the box to create user logins, and provides preliminary admin data editing capabilities.

The front end is primarily done using an adapted version of bootstrap themes, to create a responsive user interface.

We have begun implementation of the admin side of the platform, and have nearly finished the HTML/CSS design portion of the user side of the platform. The admin side doesn't require HTML work, since Django provides a basic user interface. Should additional admin needs arise, we will also implement those.

Database Implementation

As we are using Django, we have to use the built in system for data storage. Though Django ultimately uses SQLite, the user inputs the schema through a separate set of model definitions. Having created appropriate model definitions, we are able to check the generated SQL code to confirm it matches our original SQL code.

Because of the details of Django, some details of the models have changed slightly. When there is no individual primary key, Django generates a single primary key attribute. As such, we have implemented a unique together method, along with an index, so that the double attribute keys can still function as needed despite not being the primary keys.