

University of Colorado  
Department of Computer Science  
CSCI 3308  
Milestone 4  
**Purple Cobras**

**Team Members**

Kin Cheung Leung kile2101@colorado.edu	Rachel Mamich rama7226@colorado.edu
Drake Rutherford drru8281@colorado.edu	Christian Simons chsi2839@colorado.edu
Qinglu Sun qisu4484@colorado.edu	

# 1 DBMS Software and SQL Queries

Team Purple Cobras chose postgresql as the database language. The database will use three separate tables, “logininfo”, “userinfo”, and “classes”. “logininfo” will hold all the usernames and passwords, and will be queried upon login. “userinfo” will hold each user’s school, availability, and contact information. This table will be queried to display the user’s profile page and the results of their study buddy searches. “classes” will use a concatenated username/class key to list every class every user is attending. “classes” will be queried to find all users attending a certain class. All three tables will be connected by the username column.

The code to initialize and edit the database can be found at the following link:

<https://github.com/PurpleCobras3308/code/tree/master/sql>

The code that exists within the above Github repository can also be found below with descriptions of what each script does.

The database initialization will look like the following:

```
CREATE TABLE logininfo (  
username VARCHAR(45) NOT NULL,  
password VARCHAR(45) NOT NULL,  
PRIMARY KEY (username)  
);  
  
CREATE TABLE userinfo (  
username VARCHAR(45) NOT NULL,  
school VARCHAR(45) NOT NULL,  
availability VARCHAR(45) NOT NULL,  
contact VARCHAR(45) NOT NULL,  
PRIMARY KEY (username)  
); /*other columns may be added to userinfo as the  
profile page becomes more fleshed out*/  
  
CREATE TABLE classes (  
username VARCHAR(45) NOT NULL,  
class VARCHAR(45) NOT NULL,  
PRIMARY KEY (username, class)  
);
```

Users will be able to query the database to discover other users taking the same classes. The query will look like the following:

```
SELECT [userinfo columns]  
FROM classes c INNER JOIN userinfo u ON c.username = u.username  
WHERE c.class = [user input];
```

Log-in information will be verified using the “login” table. A “login” query to get the information to display a user’s profile page looks like the following:

```
SELECT [userinfo columns]  
FROM userinfo u INNER JOIN logininfo l ON u.username = l.username  
WHERE l.username = [user input] AND l.password = [user input];
```

The query for a user editing their profile information will look like the following:

```
UPDATE userinfo  
SET [column] = [user input]  
WHERE username = [user’s username];
```

The query for a user to add a class to the “classes” table will look like the following:

```
INSERT INTO classes (username, class)  
VALUES ([user’s username], [user input]);
```

## 2 Data Model

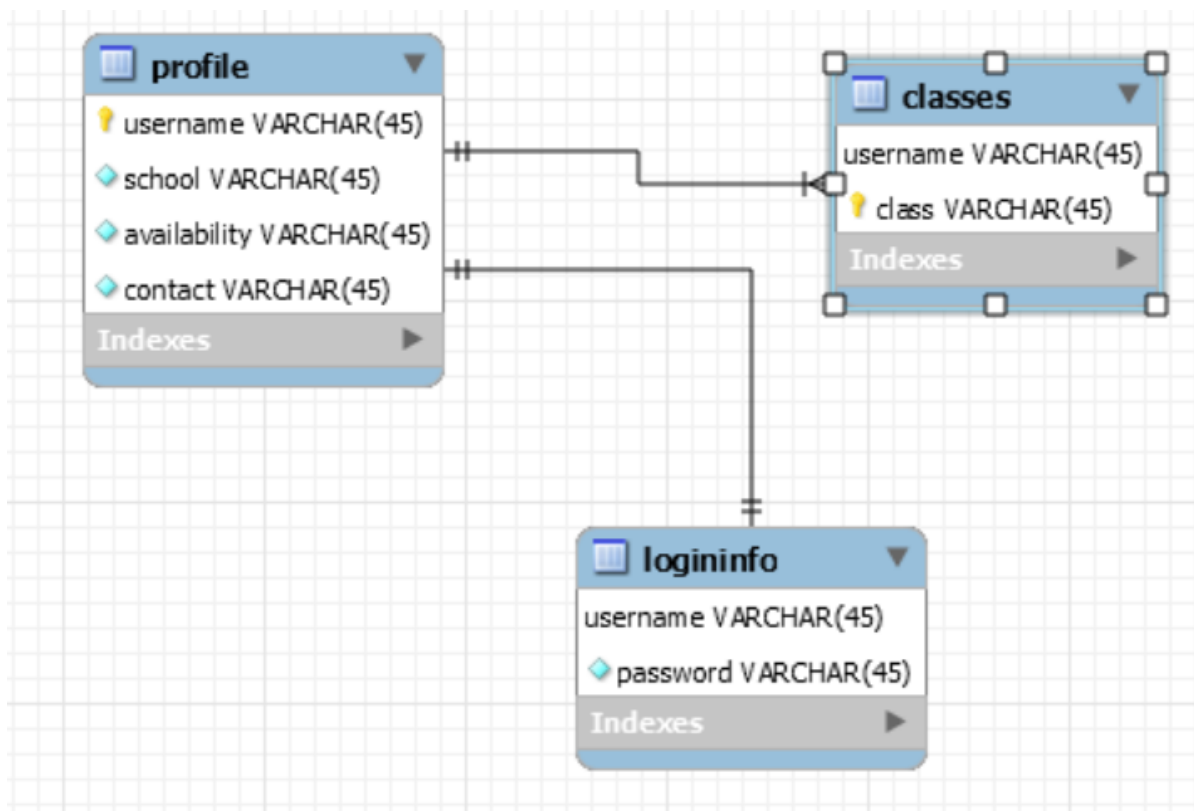


Figure 1: The diagram for the app's PostgreSQL database. For every row in the profile table, there is exactly one corresponding row in the "logininfo" table containing the user's password, and one or more corresponding rows in the classes table containing the user's classes. "username" is a primary key in every table.