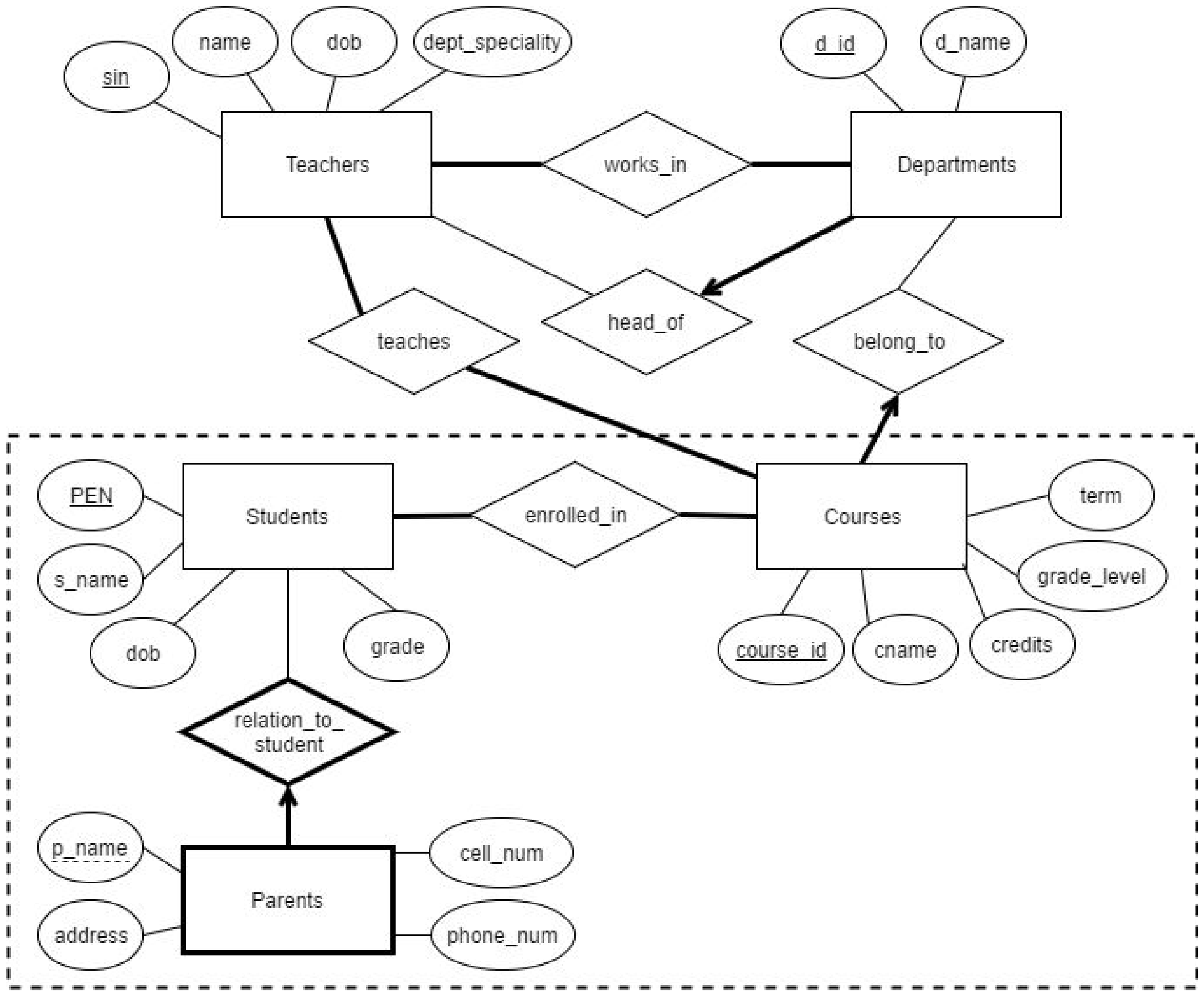
Sofia Bautista
301292776
IAT 352 (D101)
03/19/2017

P02 Technical Report

ER Diagram:



Design:
The database design for my second project expanded upon the in-lecture exercises drawing ER diagrams in Week 6 for a database holding student, professor, and course information.

A large part of the design consists of the entity sets and relationship sets between *courses* and their corresponding *teachers* and d*epartments*. As a result, the *students, courses,* and *parents* entity sets are aggregated, since many courses are each taught by at least one teacher and belong to only one department. In particular, within this aggregation is a weak entity set between *students* and *parents*, because there is one student (one owner) to more than one parent (many weak entities).

Relationship sets like *works_in, teaches,* and *enrolled_in* signify many-to-many and total participation key constraints, since there are many teachers working in at least one department, many teachers teaching at least one course, and many students enrolled in at least one course.

On the other hand, relationship sets like *head_of* and *belong_to* signify many-to-one and both total and partial participation key constraints, since each department is managed by one teacher and each course belongs to one department.

Implementation:
At first, I had difficulty implementing nine tables as separate queries in PHP because in-lecture and tutorial, we were only taught handling two queries at a time in procedural notation. It took time for me to properly understand and implement using *mysqli_multi_query()* in object-oriented notation to submit all nine tables at the same time. Nevertheless, I also learned that the order in which the queries go in is important as well, especially if they defined foreign keys using the primary keys from other tables. For example, tables like *departments, students,* and *teachers* are queried first because they only use primary keys, while *head_of*, *enrolled_in, teaches,* and *works_in* should go last because each of them use two foreign keys from two different tables.

There was also a lot of work involved in using sessions properly to display the most updated username to the user. I had to rearrange many *session_start()*'s and *session_destroy()*'s for the *$login_session* variable to be accessible from start to finish of the user's visit. I attempted to develop the functionality of displaying the new username to a user if they change their username while keeping them logged in. After spending a couple of days working on this one feature, I had to compromise and settled on automatically logging the user out subsequent to changing their account information, and then having them log in again using their new account information to start a new session.

In relation, after researching various online sources, I learned that it is important to start a session on every new page a user navigates through to keep the current session going. However, when a user logs out and tries to access the index.html/homepage again, I came across a “too many redirects” error. It was due to the fact that I had a conditional in *session.php* which defined every instance when a session ends and redirects using *header()* to the same page, and thus, created an infinite loop. Again, by placing the proper amount of *session_destroy()* functions and a new conditional at the top of every navigable page defining to start a session only when the *$login_session* variable is accessible, I was able to counteract this error.