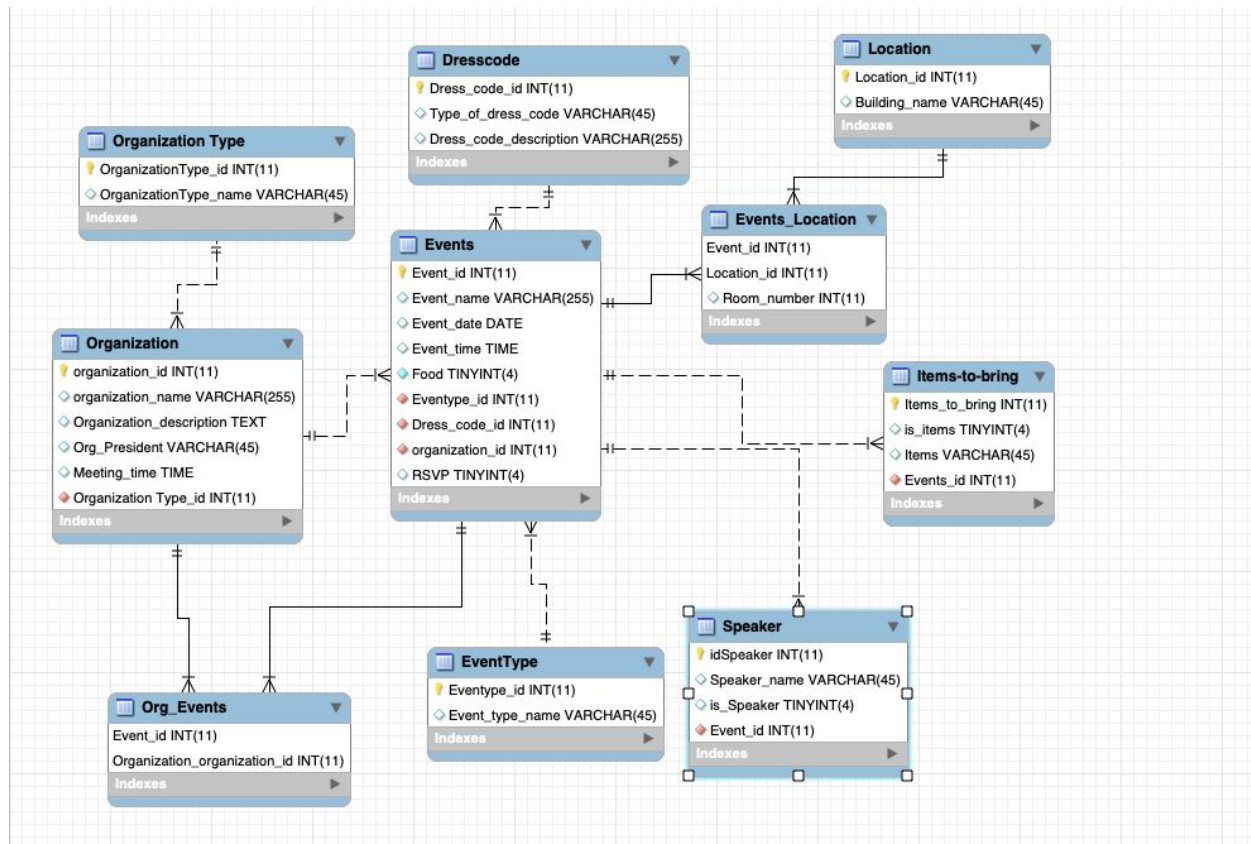


For this project, we built a database that will display campus events for students. We organized our database based on different type of events. For example, sports-related events would fall under a table designated for sports and so forth. The tables are generalized based on the categories that apply to the event. The events are organized and multiple tables and we have references (id) to the other tables in a one to many relationships. We will primarily focus on student-run organizations instead of university or faculty organizations. We decided to do student events because we know that there are a lot of events that happen on campus that many students aren't aware of. As Information Science majors, we specifically would like to know when tech-based organizations are inviting and hosting companies to review resumes or to hire students. In addition to that, we designed our database so that students know what attire to wear to each event. We were also concerned about being as detailed as possible therefore we added a column that informed students on what to bring to each event. For example, some events may require laptops. Our goal was to create a database that was accessible to students and when they attend these events they would be prepared for them. We have all had a challenging experience trying to find out when, where and what events were taking place on campus. This database is intended to facilitate that issue. There's a database for club meetings but not for actual events, therefore, it was essential to make a centralized database for students.

We wanted to give the students of the University of Maryland an easier way to figure out what's events are occurring on campus. This database will consolidate the event information on all of their events and make it easier for students to find the events and the information they want, so they can take advantage of more opportunities and connect with more people. In addition to helping students, this database will also benefit the clubs and organization the put on these events. By having their information in our database, their events will be publicized to more people and is easy to understand, people will get less confused about the event making them more likely to go.

This database will have information on the events occurring on the UMCP campus for one week. It lets students know whether or not there will be food at an event, whether or not they have to RSVP for an event, what to bring to the event, what to wear to the event, where the

event takes place, when the event takes place, and who is going to be speaking at the event as well as the hosting organization. This database will help keep students informed about events on campus and will decrease information overload and increase information accuracy.



The Screenshot above is a snap shot of our ERD. The way we structured is first we created our main table: Events. This table is what brings the database together, some of the major columns like the event name, event date and time are the most important to the content of our database. The events table is linked to a lot of other tables in our database. For example, Events has a one to many relationships between the items to bring table and speaker table. It also has many to one relationship between organization table, dress code table, and event type table. We also created two join table the first one consisted of a join between organizations and events, the

other join table combines location and events. The organization table also has a many to one relationship with organization type table.

We based our views on what most students would use our events for. For instance, the first query 'find\_events\_and\_orgs' lets the user locate an organization's events and we format the date so that it's easier for the user to know when to go to the event. Our next view was made to find all professional events that offer food. Many students wouldn't go to an event if there is no food that's why we find the ones that do. It also shows how well we connected tables that don't have a direct connection. Events and organizations don't have foreign keys connecting them but to account for this we made a linking table org\_events to easily bridge the gap between the two. The rest of the queries we test to check and see if our database works well with aggregate functions and others methods.

As a group, we all learned how to work together in a team when dealing with a large cumbersome project, build efficient queries and databases to be used in production and how to use and visualize and build databases. We learned that doing our research is very important. Utilizing each other background knowledge is always important. Some of us had prior knowledge and experience on where to find the data to import into our database. We have made tremendous progress over the past few weeks. Our professor suggested that we make more tables and make them more specific and detailed. Therefore it was easier for us to fill in sections such as required apparel for an event or items to bring for the event. These stipulations helped us to build our tables and make them easier to read. In order to retrieve our data, we were advised to use a calendar layout instead keeping track of all events through group chats. At first, we wanted to search through social media and group me's in order to track campus events but we quickly realized that this was not the most efficient way to import data. One of our group members suggested that we use the e-calendar provided by STAMP in order to search for events and event details. The calendar was very effective. Also, we were going to import the entire month of October but instead we only needed to use one week in October. If we had imported every event

held in October than we would have recorded hundreds of events which probably wouldn't have been the smartest way to do things since we still had to complete several other portions of our project. By recording only a week's worth of events we did not have to record as many events as anticipated. When we first began this project it was difficult for us to create more than 7 tables. However the more we took the aspects of the events apart we began to notice that we would need well over 7 tables. What we also learn is the import of data is not as easy as it sounds. Going through that process we ran into a lot of problems. Some CVS data could not upload to some of the tables so as a result, we had to insert tables like Org\_event and items to bring manually. That process showed how it's important to pay attention to detail when importing data to tables.

We've made a lot of significant changes from the first proposal. After receiving all the feedback from our professor and Teacher Assistant we understood the reasoning behind the changes that needed to be set in place. The first and most important thing we had to do was figure out the tables we were going to use. Initially, we miss understood what exactly makes something a table. In the first proposal, we seemed to have 2 tables (Organizations & Events) and we sliced row-wise based on Organization\_Category to artificially create multiple tables. In order to fix that we consulted with our professor and we created new sets of tables. As seen in the ERD above Events is our major table, we also have Event Type, Organization, Organization, RSVP, Dress Code, Location, Items to bring, and Speakers. As you can see we added a Speakers table because we felt as though the speakers are often a major part of events so you felt it was important that we have that as a part of our database. Another major thing we changed was deleting our RSVP table because we felt that making RSVP into a full table was not necessary for our Events database

The table Organization event is a linked table between events and organization with two composite primary keys. The logic behind that decision is that there's a possibility that each event is able to have multiple organizations hosting that particular event. We also have a link table We now feel confident in the structure of our database. We also clarified as a group the scope of our database. There are so many events that happen within a week on this campus so as a group we agreed on a week's worth of events is the scope of the database we are building.

INST327  
Team Project Final Submission  
12/6/18  
Team B

Sean Brown, Jepthe Cadet, Maimouna Camara, Martin Campos, Constance Hackett

In the future, it would be beneficial to expand our database. Instead of just importing stamp events we could import events that happen everywhere on campus. We could also expand to posting events that happen outside of the campus. We would definitely stay inside of the DC. Maryland, Virginia Area but this would allow for students to know what type of events are happening in their area. For international students or out-of-state students it may be difficult to find things to do, jobs or social events, by broadening this database we will give them a chance to get involved outside of the University of Maryland Campus. Another potential idea would be to incorporate more information to the database to make it more complex and be able to display more information about the events. Overall there are many possible ways to improve the database and facilitate

INST327  
Team Project Final Submission  
12/6/18  
Team B

Sean Brown, Jepthe Cadet, Maimouna Camara, Martin Campos, Constance Hackett

| ViewName                                     | Req.<br>A | Req.<br>B | Req.<br>C | Req.<br>D | Req.<br>E |
|--|-----------|-----------|-----------|-----------|-----------|
| find_events_and_orgs                         | x         | x         |           | x         |           |
| find_all_professionals_and_if_they_have_food | x         | x         | x         | x         |           |
| what_items_to_bring                          | x         | x         |           |           |           |
| all_non_causal_events_and_what_to_wear       | x         | x         |           |           | x         |
| events_per_day                               |           |           | x         |           |           |