

# CS411 PT1 Stage2 - Database Design

Matt Straczek, Peyton Murray, Qi Yu, Hao Qi

February 20, 2023

## 1. Summary

This project intends to build a review website, RateMyRSOs, that allows students at UIUC to post reviews and ratings to UIUC's Registered Student Organizations (RSOs). Therefore, our team is designing a database for keeping track of reviews for different RSOs posted by different users.

We have 5 entities in our database. Firstly, we want to capture information about RSOs, users and reviews. For RSO, we want to keep track of their name, contact information, year of establishment, as well as some external links such as links to their website, Facebook and Instagram account. We also want to know the officers of different RSOs as well as the departments which different RSOs are affiliated with. Therefore, we introduce two more entities in our database to keep track of the information of different departments, including their names, addresses, and contact information, and to capture details about different officers including their name, contact information, and the RSOs they manage. For users, we want to collect information about their username, passwords, names and types (admin or students). For reviews, we want to know the author of each review and the RSO that the review is about. Additional information related to reviews are the content, associated rating, date, and the number of likes each review receives. Finally, for RSOs, Officers and Reviews database, we use an ID to identify different entries, which are RSOId, OfficerId, and ReviewId respectively.

Different entities in our database are also related to each other. We assume that one RSO can be managed by multiple officers but have to be run by at least one officer. We also assume that an officer can run multiple RSO at the same time. For the relationship between RSOs and Departments, we assume that an RSO can be affiliated with multiple department, and one department can have multiple RSOs as well. Users can self-identify as a member of multiple RSOs, and an RSO can have multiple user as members. For reviews, one piece of review must be reported by one user and must be associated with one RSO. In the meantime, one RSO can have multiple reviews and one user can post multiple reviews as well. Finally, we also want to enable users to report about reviews, in case the users think there's something wrong with any reviews they see. The reports must have some content. One user can report multiple reviews and one review can also be reported by multiple users.

## 2. Relational Schema

We can translate our ER diagram into relational schema as follows

```
RSOs (  
    RSOId INT [PK],  
    ContactEmail VARCHAR(255),  
    YearEstablished INT,  
    WebsiteLink VARCHAR(255),  
    FacebookLink VARCHAR(255),  
    InstagramLink VARCHAR(255)  
)  
  
Officers (  
    OfficerId INT [PK],  
    RSOId INT [FK to RSOs.RSOId],  
    Name VARCHAR(255),  
    ContactEmail VARCHAR(255)  
)
```

```

Departments (
    Name VARCHAR(255) [PK],
    Address VARCHAR(255),
    ContactPhone VARCHAR(20),
    ContactEmail VARCHAR(255)
)

Users (
    Username VARCHAR(255) [PK],
    Password VARCHAR(30),
    Type VARCHAR(20),
    Name VARCHAR(255)
)

Reviews (
    ReviewId INT [PK],
    Username VARCHAR(255) [FK to Users.Username],
    RS0Id INT [FK to RS0s.RS0Id],
    Content VARCHAR(1024),
    Rating INT,
    NumLikes INT,
    PostDate DATE
)

Managements (
    OfficerId INT [FK to Officers.OfficerId],
    RS0Id INT [FK to RS0s.RS0Id],
    (OfficerId, RS0Id) [PK]
)

Affiliations (
    RS0Id INT [FK to RS0s.RS0Id],
    DeptName VARCHAR(255) [FK to Departments.Name],
    (RS0Id, DeptName) [PK]
)

Memberships (
    Username VARCHAR(255) [FK to Users.Username],
    RS0Id INT [FK to RS0s.RS0Id],
    (Username, RS0Id) [PK]
)

Reports (
    Username VARCHAR(255) [FK to Users.Username],
    ReviewId INT [FK to Reviews.ReviewId],
    Content VARCHAR(1024),
    (Username, ReviewId) [PK]
)

```

### 3. ER Diagram

Our conceptual database design (ER diagram) is shown in Figure 1.

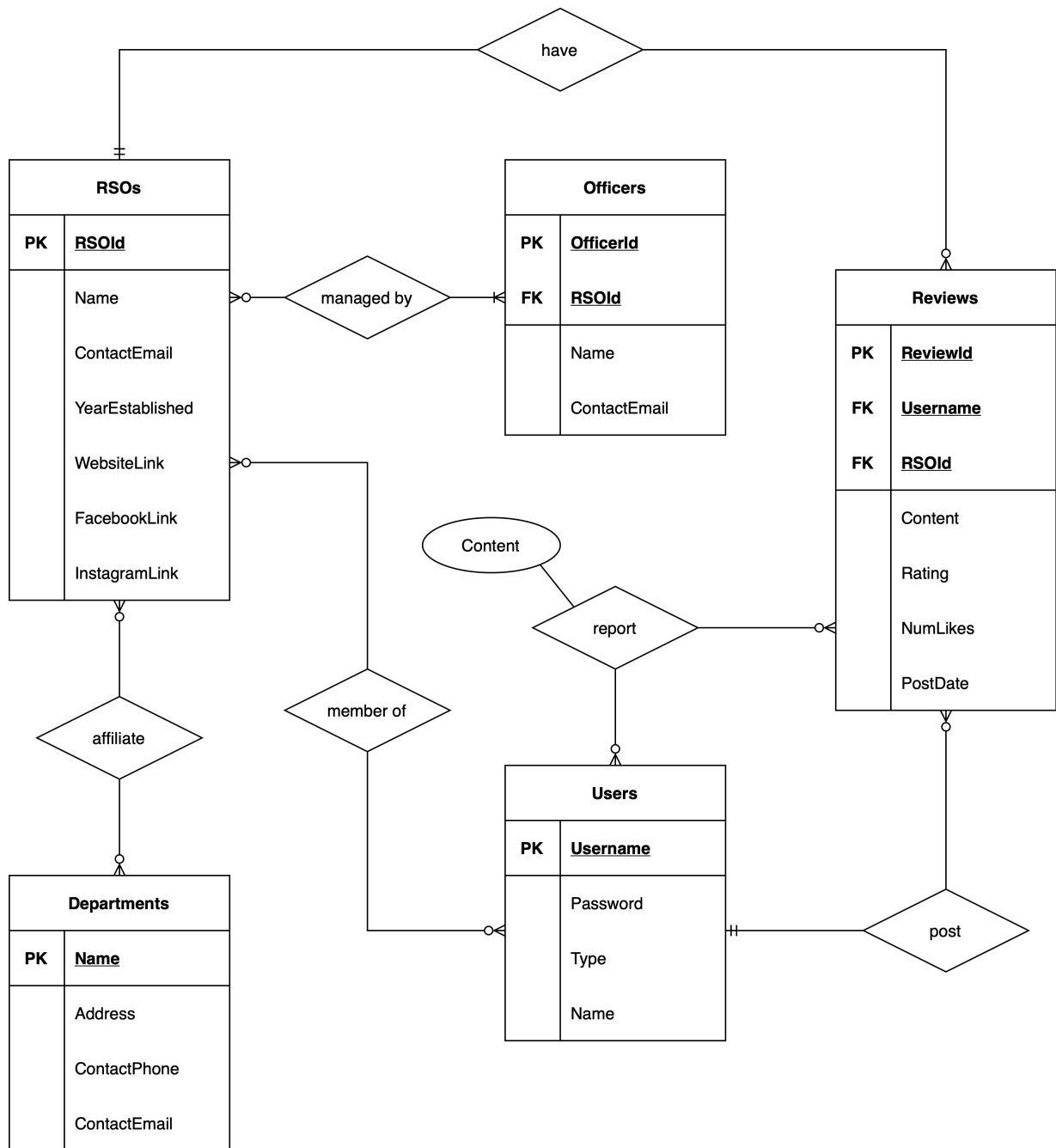


Figure 1: ER Diagram