

Mini Project

Erfan Shafagh, Matthew Lee

Project Specifications

Researchers (Entries are researchers)

researcher_id (Primary) - int
first_name - varchar
last_name - varchar
email - varchar
organization - varchar

Competitions (Entries are calls for proposals)

competition_id (Primary) - int
title - varchar
deadline - date
description - varchar
area - varchar
status - varchar

Applications (Entries are submissions/proposals for a competition)

application_id (Primary)
competition_id (Foreign to Competition)
principal_investigator (Foreign to Researchers)
requested_amount - int
status - varchar
awarded_amount - int
awarded_date - date

Collaborators (Entries are researchers who worked on a specific proposal)

application_id (Foreign to Application)
researcher_id (Foreign to Researchers)

Reviewers (Entries are reviewers, who also may be researchers)

reviewer_id (Primary)
researcher_id (Foreign to Researchers)

Conflicts (Entries are researchers who have conflict with specific reviewer)

reviewer_id (Foreign to Reviewers)
researcher_id (Foreign to Researchers)

Assignments (Entries are the assignment of reviewers to competitions)

reviewer_id (Foreign to Reviewer)
competition_id (Foreign to Competitions)
deadline - date
status - varchar

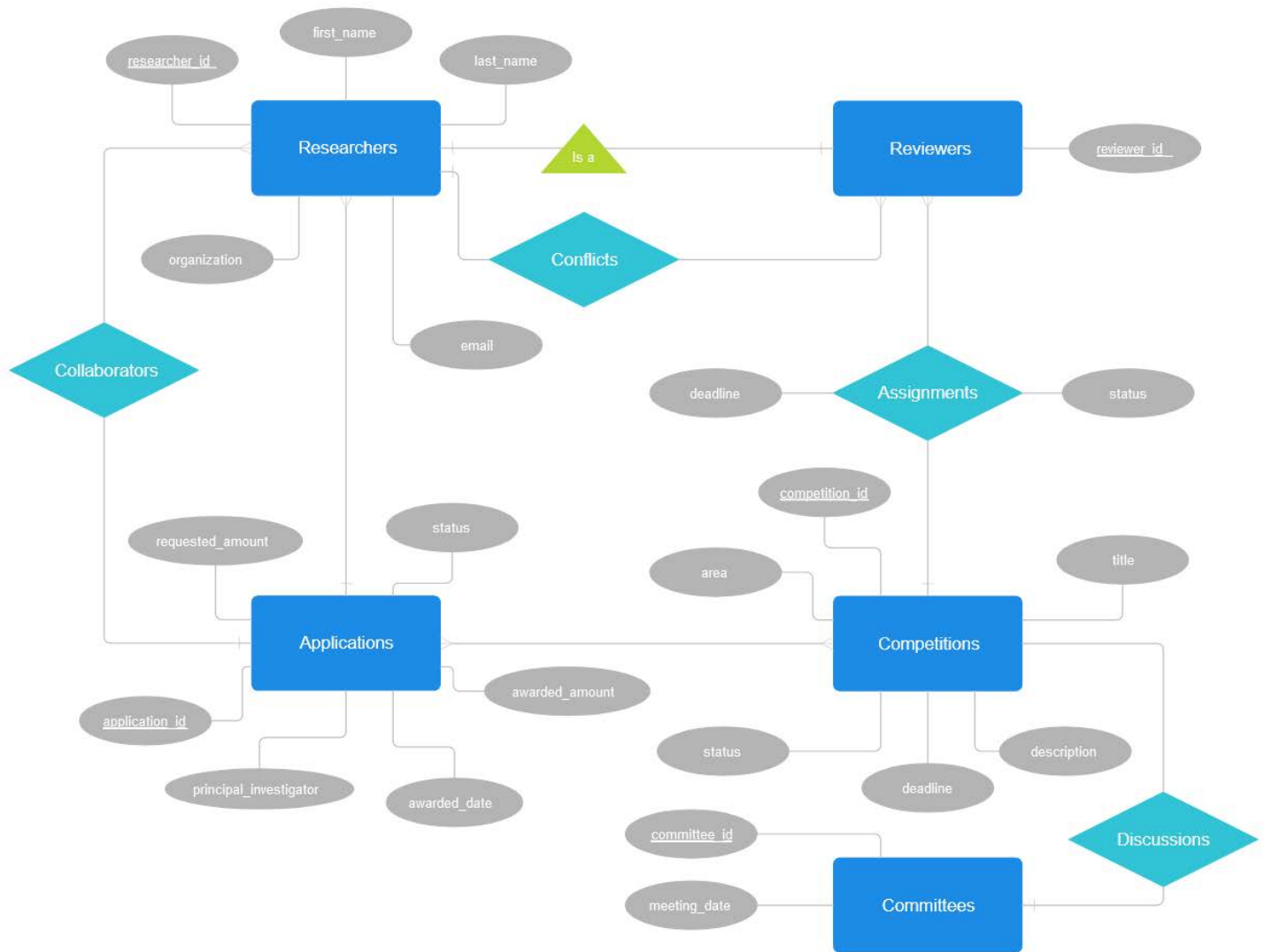
Committees (Entries are the committee meetings at a specific day)

committee_id (Primary) - int
meeting_date - date

Discussions (Entries are the competitions discussed/voted on at a specific committee meeting)

committee_id (Foreign to Committees)
competition_id (Foreign to Competitions)

ER Diagram



Design Analysis

To verify the design has no anomalies, we can evaluate the functional dependencies of the schema:

$\text{researcher_id} \rightarrow \text{first_name}, \text{last_name}, \text{email}, \text{organization}$

$\text{competition_id} \rightarrow \text{title}, \text{deadline}, \text{description}, \text{area}$

$\text{application_id} \rightarrow \text{competition_id}, \text{principal_investigator}, \text{requested_amount}, \text{status}, \text{awarded_amount}, \text{awarded_date}$

$\text{reviewer_id} \rightarrow \text{researcher_id}$

$\text{reviewer_id}, \text{competition_id} \rightarrow \text{deadline}, \text{status}$

$\text{committee_id} \rightarrow \text{meeting_date}$

For our schema to not allow anomalies, we can evaluate if its in BCNF form or not. We can do this by evaluating if each functional dependency of the system follows one of two rules:

For functional dependency $X \rightarrow Y$

1. $Y \subseteq X$ (FD is trivial)
2. X is a key

$\text{researcher_id} \rightarrow \text{first_name}, \text{last_name}, \text{email}, \text{organization}$
 researcher_id is a key, so **BCNF**

competition_id → *title, deadline, description, area*

competition_id is a key, so **BCNF**

application_id → *competition_id, principal_investigator, requested_amount, status, awarded_amount, awarded_date*

application_id is a key, so **BCNF**

reviewer_id → *researcher_id*

reviewer_id is a key, so **BCNF**

reviewer_id, competition_id → *deadline, status*

reviewer_id, competition_id is a key, so **BCNF**

committee_id → *meeting_date*

committee_id is a key, so **BCNF**

All the functional dependencies of our system follow **BCNF**, meaning that our system should not have any anomalies.