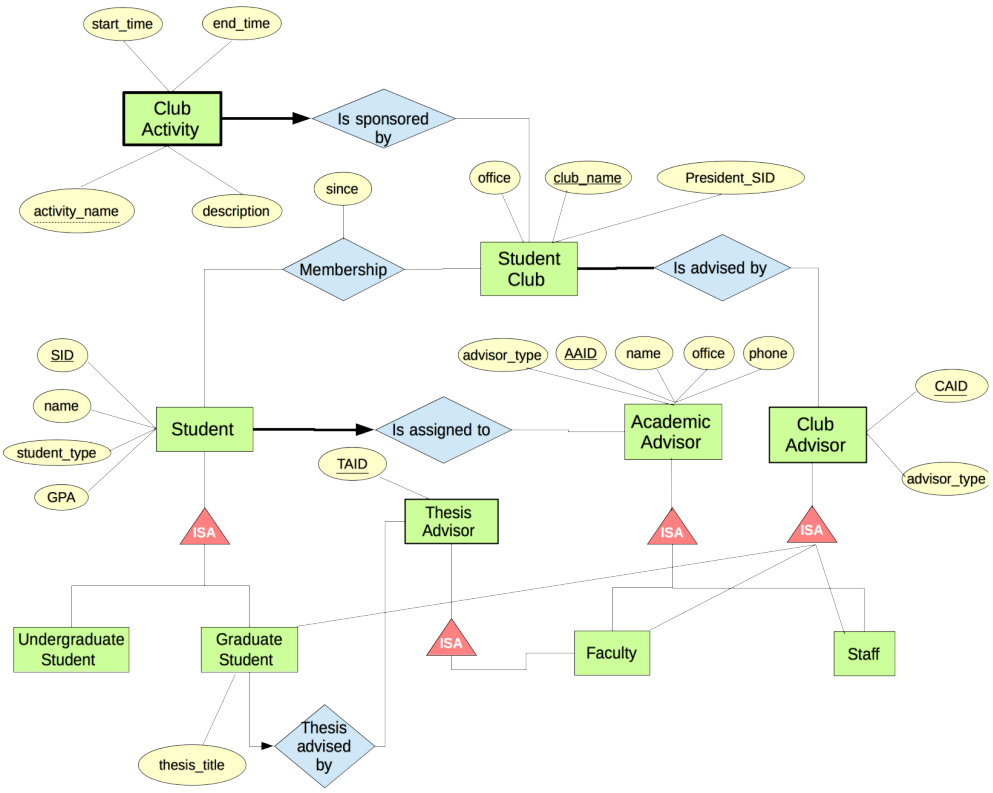
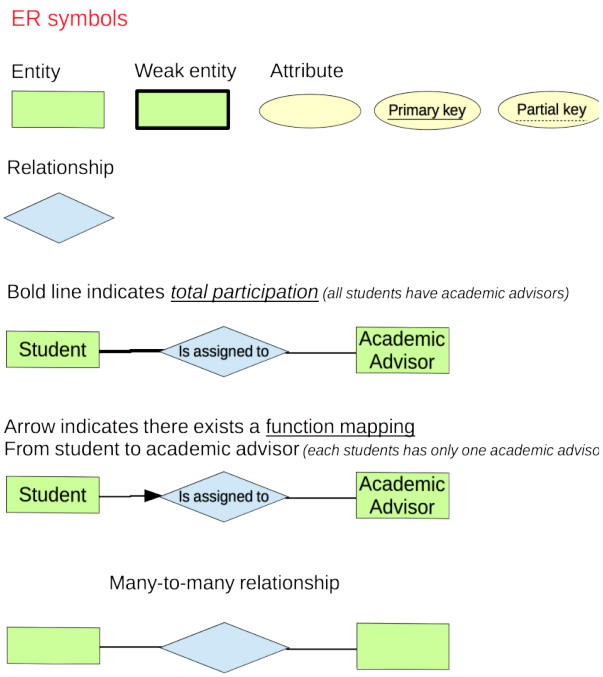
CS460-Prog4: Sample Design
Shuo Yang

Conceptual Design/ER Diagram:



ER Symbols Explanation:



Assumptions:

- each student club can have multiple advisors; each club advisor can advise multiple clubs.
- a thesis advisor can advise multiple graduate students.

Logical Design/Relational Schema:

1. **Academic_Advisor** (AAID, name, office, phone, advisor_type)

   This table corresponds to the `Academic Advisor` entity in the ER diagram.
   `AAID` represents a unique academic advisor ID; `advisor_type` can only be "Faculty" or "Staff".

2. **Student** (SID, name, student_type, gpa, advisor_id)

   This table corresponds to the `Student` entity in the ER diagram.
   `SID` represents a unique student ID; `advisor_id` is the foreign key refers to the primary key `AAID` of the `Academic_Advisor`; `student_type` can only be either "Graduate" or "Undergraduate".

3. **Student_Club** (club_name, office, president_SID)

   This table corresponds to the `Student Club` entity in the ER diagram.
   `club_name` represents a unique club name (within department); `president_SID` is foreign key refers to the primary key `SID` of the `Student` table, it's the student ID of the president of the club. This value cannot be NULL in order to satisfy the constraint that "each club must have at least one member to exist"

4. **Club_Membership** (SID, club_name, since)

   This table corresponds to the `Membership` relationship in the ER diagram.

5. **Grad_Club_Advisor** (CAID, SID)

This table corresponds to part of the `Club Advisor` entity in the ER diagram. Specifically, it represents those graduate students who are club advisors.
`CAID` represents a unique club advisor ID for graduate student. `SID` is the foreign key refers to the primary key `SID` of the `Student` table. Namely, the student ID of the graduate student who advises a club.

6. **Faculty_Staff_Club_Advisor** (CAID, AAID)

   This table corresponds to part of the `Club Advisor` entity in the ER diagram. Specifically, it represents those faculty and staff who are club advisors.
   `CAID` represents a unique club advisor ID for faculty and staff. `AAID` is the foreign key refers to the primary key `AAID` of the `Academic Advisor` table. Namely, the academic adviosr ID of the faculty or staff who advises a club.

7. **Club_Advised_By** (CAID, club_name, advisor_type)

   This table corresponds to the `is advised by` relationship in the ER diagram.
   `CAID` is the foreign key refers to the primary key `CAID` of either `Faculty_Staff_Club_Advisor` or `Grad_Club_Advisor` table depends on the value of `advisor_type`. `advisor_type` can only be either "Faculty_Staff" or "Graduate_Student".

8. **Club_Activity** (activity_name, club_name, start_time, end_time)

   This table corresponds to the `Club Activity` entity in the ER diagram.

9. **Thesis_Advised_By** (SID, AAID, thesis_title)

   This table corresponds to the `Thesis advised by` relationship in the ER diagram.
   `SID` is the student ID of the graduate who has a thesis advisor. `AAID` is the academic advisor ID of the faculty who advises the thesis.

Design Decisions:

- $AAID$ and $CAID$ are treated differently. $AAID$ is pre-assigned to each faculty and staff member, regardless they are currently advising students or not. $CAID$ are generated as needed, if a club advisor no longer advises a club, his/her $CAID$ will be deleted.
- Because club advisor can be graduate student, faculty or staff, we need two relations: **Grad_Club_Advisor** and **Faculty_Staff_Club_Advisor** for the entity **Club Advisor** in the conceptual design.
- $CAID$ are not unique across **Grad_Club_Advisor** and **Faculty_Staff_Club_Advisor** relations. It's only unique within each relation. So it's possible to have both a graduate student and a faculty member with $CAID$ of 1.
- Since not every graduate student has a thesis advisor, so it is not a good practice to store thesis advisor ID directly in **Student** relation, because then we will have a lot of NULL foreign keys.
- All the information regarding to graduate student is captured in **Thesis_Advised_By** relation, so there is no need to have a specific relation for graduate student.
- There is no need to have specific relations for undergraduate student, faculty or staff unless there are information specific to these three entities we need to model, for example, faculty may have research areas they are focusing on, staff may have specific duties to perform.

Normalization Analysis:

Recall that a relation R is in 3NF if and only if both of the following conditions holds:

- R is in 2NF.

- Every non-prime attribute of R is non-transitively dependent on every key of R.

A relation R is in 2NF if it is in 1NF and every non-prime attribute of the table is dependent on the whole of every candidate key. (no partial dependency)

A relation R is in 1NF if the value of each attribute contains only a single value from its corresponding domain which contains only atomic values.

Note that trivial FDs are ignored in the following analysis.

1. **Academic_Advisor** (AAID, name, office, phone, advisor_type)

   Candidate key: $AAID$
   Primary key: $AAID$

   FDs: $AAID \rightarrow name$; $AAID \rightarrow office$; $AAID \rightarrow phone$; $AAID \rightarrow advisor\_type$;

   The table is in 1NF assuming that each advisor can only have a single office and a phone.

   The table is in 2NF because the only candidate key contains a single attribute, there is no partial dependency in this table.

   The table is in 3NF because the only candidate key contains a single attribute and there are no FDs existing among non-prime attributes.

2. **Student** (SID, name, student_type, gpa, advisor_id)

   Candidate key: SID
   Primary key: SID

   FDs: $SID \rightarrow name$; $SID \rightarrow gpa$; $SID \rightarrow advisor\_id$; $SID \rightarrow student\_type$;

   The table is in 1NF assuming that each student has only one name, advisor, gpa and is either graduate or undergraduate student.

   The table is in 2NF because the only candidate key contains a single attribute, there is no partial dependency in this table.

   The table is in 3NF because the only candidate key contains a single attribute and there are no FDs existing among non-prime attributes.

3. **Student_Club** (club_name, office, president_SID)

   Candidate key: $club\_name$
   Primary key: $club\_name$

   FDs: $club\_name \rightarrow office$; $club\_name \rightarrow president\_SID$;

   The table is in 1NF assuming that each student club has only one office and one president who is a student.

   The table is in 2NF because the only candidate key contains a single attribute, there is no partial dependency in this table.

   The table is in 3NF because the only candidate key contains a single attribute and there are no FDs existing among non-prime attributes.

4. **Club_Membership** (SID, club_name, since)

   Candidate key: $\{SID, club\_name\}$
   Primary key: $\{SID, club\_name\}$

   FDs: $\{SID, club\_name\} \rightarrow since$;

   The table is in 1NF because each student joined a club at a specific time.

   The table is in 2NF because neither SID nor $club\_name$ can functionally determine $since$.

   The table is in 3NF because there only a single non-prime attribute $since$ in this relation and it doesn't functionally determine SID or $club\_name$, that means there is no transitive dependency in this relation.

5. **Grad_Club_Advisor** (CAID, SID)

   Candidate key: $CAID$, SID
   Primary key: $CAID$

   FDs: $CASID \rightarrow SID$; $SID \rightarrow CAID$;

   The table is in 1NF assuming that each graduate student has been assigned with only one $CAID$.

   The table is in 2NF and 3NF because there are no non-prime attributes in the relation.

6. **Faculty_Staff_Club_Advisor** (CAID, AAID)

   Candidate key: $CAID$, $AAID$
   Primary key: $CAID$

   FDs: $CASID \rightarrow AAID$; $AAID \rightarrow CAID$;

   The table is in 1NF assuming that each faculty or staff has been assigned with only one $CAID$.

   The table is in 2NF and 3NF because there are no non-prime attributes in the relation.

7. **Club_Advised_By** (CAID, club_name, advisor_type)

   Candidate key: $\{CAID, club\_name, advisor\_type\}$
   Primary key: $\{CAID, club\_name, advisor\_type\}$

   FDs:

   The table is in 1NF, 2NF and 3NF because there are no non-prime attributes in the relation.

   Note that $CAID$ doesn't functionally determine $club\_name$ because a graduate student with $CAID = 1$ could advise club "A", while a faculty with $CAID = 1$ could advise club "B". Vice versa.

8. **Club_Activity** (activity_name, club_name, start_time, end_time)

   Candidate key: $\{club\_name, activity\_name\}$
   Primary key: $\{club\_name, activity\_name\}$

   FDs: $\{club\_name, activity\_name\} \rightarrow start\_time$; $\{club\_name, activity\_name\} \rightarrow end\_time$

   The table is in 1NF because each activity starts and ends at a specific time.

   The table is in 2NF because neither $activity\_name$ nor $club\_name$ can functionally determine $start\_time$ or end_time.

   The table is in 3NF because there are no FDs existing among non-prime attributes and no non-prime attributes can functionally determine prime attributes.

9. **Thesis_Advised_By** (SID, AAID, thesis_title)

   Candidate key: SID
   Primary key: SID

   FDs: $SID \rightarrow AAID$; $SID \rightarrow thesis\_title$;

   The table is in 1NF assuming that each student has been assigned with only one thesis advisor and work on one thesis topic.

   The table is in 2NF because the only candidate key contains a single attribute, there is no partial dependency in this table.

   The table is in 3NF because the only candidate key contains a single attribute and there are no FDs existing among non-prime attributes.