Database and Web Services Group Project ~ Jacobs Club Manager

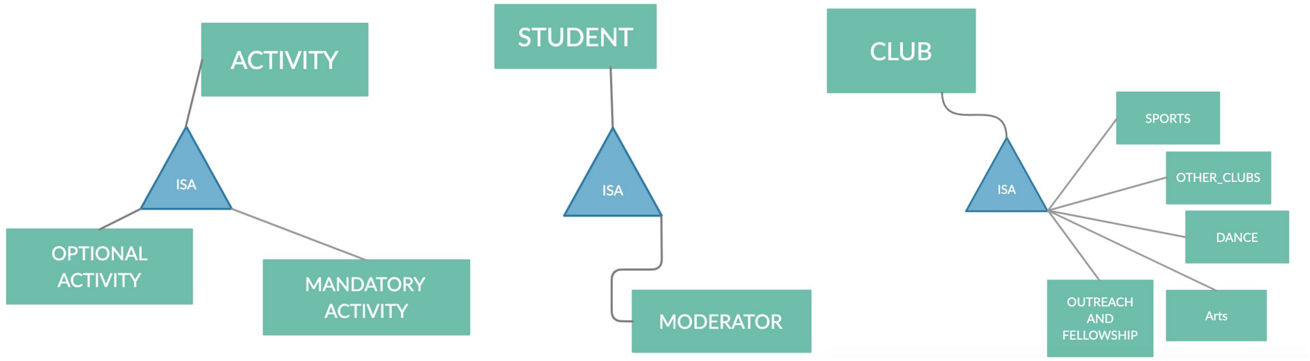
Group Members: Ankit Pandit, Arnav Singh, Nayan Man Singh Pradhan

Mapping Approach

A programmer may choose to implement an ISA hierarchy in three different mapping approaches:

1. Create a separate table for each relation per entity set.
2. Create a separate table for relations with only entity set with instances.
3. Create one single table for one big relation.

The ISA hierarchies in our ER diagram are as following:



1. **Optional Activity ISA Activity, Mandatory Activity ISA Activity**
   For this ISA hierarchy, we use the third mapping approach: Create one single table for one big relation. We chose to use this approach because the child entities do not have any attributes. This makes it convenient and easier to use one big table to classify the two child entities. We are using TINYINT(1) as our data type as we can use 0 as False and 1 as True values. While documenting this approach, we realized that we have not set an overlapping and covering constraint for the child activity yet. We will update this in the future submissions. We also chose the third approach as querying whether an activity is optional or mandatory is going to be faster for the user. We did not use the first approach because we do not have entities with attributes. We did not use the second approach because our Activity entity already had attributes that are relevant for both our child entities.

2. **Moderator ISA Student**

   For this ISA hierarchy, we use the first mapping approach: Create a separate table for each relation per entity set. We chose to use this approach because both students and moderators had their separate attributes, and we referenced students as a foreign key in the moderator table. (We can then use join to get the student attributes). Additionally, if we delete a student who is a moderator, then the corresponding moderator tuple also gets deleted. We did not use the second mapping approach because we only have one child for the student parent entity. We did not choose the third approach because students and moderators had their own corresponding attributes and merging them into the same table did not seem logical.

3. **Sports ISA Club, Other clubs ISA Club, Dance ISA Club, Arts ISA Club, Outreach and fellowship ISA Club**

   For this ISA hierarchy, we use the third mapping approach: Create one single table for one big relation. We chose to use this approach because the child entities do not have any attributes. This makes it convenient and easier to use one big table to classify the two child entities. We are using TINYINT(1) as our data type as we can use 0 as False and 1 as True values. While documenting this approach, we realized that we have not set an overlapping and covering constraint for the child activity yet. We will update this in the future submissions. We also chose the third approach as querying whether a club is a sports club, other club, art club, dance club, outreach and fellowship club is going to be faster for the user. We did not use the first approach because we do not have entities with attributes. We did not use the second approach because our Club entity already had attributes that are relevant for all our child entities.