Design Rational

Benjamin McDonnough, Kolby Turner

The STUDENT entity is used to store all of the information for each student including their, STUDENT_ID (primary key because it is unique and incrementable), STUDENT_LNAME, STUDENT_FNAME, STUDENT_DOB and STUDENT_JOINDATE. This entity is used to complete requirement 1. This entity also has a strong relationship with LEADER to make sure that all leaders are first students. However, a student doesn't have to be a leader, hence the (0,1).

The LEADER entity is an extension to the STUDENT entity to record information that is specific to a leader such as LEADER_STARTDATE and LEADER_PAIDSTATUS (which fulfils requirement 2). STUDENT_ID is both a primary key and a foreign key since LEADER is an extension of STUDENT. A leader may be assigned to a CLASS, hence the (0,N) relationship with CLASS (fulfils requirement 3).

The CLASS entity is used to define the zen/yoga classes. This includes things like the CLASS_ID (primary key because it is unique), CLASS_STARTTIME, CLASS_LOCATION, LEVEL_NAME (foreign key to the LEVEL entity since difficulty level is a requirement for a class) and STUDENT_ID (foreign key to the LEADER entity since a leader must be assigned to a class). The 1:1 relationship with LEADER is because each class must have an assigned leader and the 0:N is because a leader may not be assigned to a class and because a leader can be assigned to multiple classes.

The SESSION entity is used to represent the dated instances of each class. Since each class meets multiple times a week, this is used to track each specific meeting time. It has SESSION_ID (primary key because it is unique), SESSION_DATE, and CLASS_ID as the foreign key since it must be tied to a class. The tie between CLASS and SESSION is 1:N because each class has many sessions (multiple per week), but each session belongs to only one CLASS.

The ATTENDENCE entity is a bridge table between STUDENT and SESSION since each student has multiple sessions and each session has multiple students. The SESSION_ID and STUDENT_ID are used to form a composite key because it is tracking which student attended which class. Both connections (to STUDENT and to SESSION) are weak since attendance is dependent on both.

The SESSION_LEADER entity is also a bridge table and represents leaders who participate in individual sessions. The weak links (to LEADER and SESSION) are there to allow multiple leaders to assist/lead a session while still making sure that the head leader is identified).

The LEVEL entity is used to track the different levels for each session. It uses LEVEL_ID (primary key again because it is guaranteed to be unique), LEVEL_DESC, and LEVEL_NAME. This entity has a strong relationship with LEVEL_REQUIREMENT because each level can have multiple requirements, but each requirement can only be tied to one level. LEVEL is also tied to CLASS and STUDENT (via STUDENT_LEVEL) to assign their level.

The LEVEL_REQUIREMENT entity is used to store all the requirements for a specific level. It uses LEVEL_ID (both primary and foreign key) and the LEVEL_REQS. It has a weak relationship with LEVEL since it is dependent on LEVEL.

The STUDENT_LEVEL entity is a bridge table that manages the many to many relationships between STUDENT and LEVEL. The LEVEL_ID and STUDENT_ID are used to tie STUDENT_LEVEL to STUDENT and LEVEL and AWARDED_DATE is used to ensure that the level attainment is timestamped for accurate recordkeeping.

Zen Fest