



Sean Farrell

2/26/2025

CPSC 4620 Taylor

Project Part 1:

Rationale for Yoga School Management System

Core Entities and Key Design Decisions:

Student Entity:

The Student entity is the foundation of our system, capturing essential information required by rule #1: student identification number, last name, first name, date of birth, and join date. We added a Level_ID foreign key to efficiently track each student's current yoga level while maintaining the system's ability to record their full progression history through the Student level entity. This design choice supports rule #9's requirement that students' progress through levels must be tracked, while providing quick access to current level information. This approach maintains data integrity by ensuring students attributes DATE_JOINED exist for students.

Leader as a Subtype:

Leaders are implemented as a subtype of Student (rule #2), using a partial disjoint relationship since a student either is or isn't a leader. This approach maintains data integrity by ensuring students attributes DATE_START_AS_LEADER) only exist for actual leaders. The disjoint subtype relationship enforces the business rule that "all leaders are also students, but clearly not all students are leaders." This implementation avoids NULL values that would occur if leader attributes were placed in the Student table. Leader has a weak relationship with the Class entity (1:M), due to every class requiring at least one leader attribute per class (rule 3)

Class and Class Session:

The separation between Class and Class Session entities directly addresses rules #4 and #7. Class represents the recurring class offering with consistent attributes (time, day, location, level), while Class Session captures individual occurrences with date-specific information. This separation is crucial for accommodating variable attendance and leadership roles at each session. Class Session is designed as a weak entity because its existence depends on a Class.

Rule #4 is satisfied by storing level, time, day, and location in the Class entity, allowing the system to distinguish between different classes offered at the same location on different days or times.

Level and Requirements:

The Level entity with its associated Level Requirement entity addresses rule #8, allowing multiple requirements to be stored for each level. The one-to-many relationship ensures that "each requirement is associated with only one particular level" while supporting that "most levels have numerous level requirements."

Bridge Tables and Relationships:

Student Level Entity:

This junction table is critical for tracking student progression through yoga levels as required by rule #9. By storing the DATE_AWARDED attribute for each level a student achieves, we maintain a complete history of student advancement. This design supports the business rule that "every level that a student attains is kept in the system" and that "new students joining the school are automatically given the beginner level." The mandatory relationship between Student and Student Level (1:N) ensures each student has at least one level, satisfying the requirement that new students start with the beginner level.

Attendance Entity:

The Attendance entity implements the many-to-many relationship between students and class sessions described in rules #5 and #6. This allows the system to track which students attend which sessions without forcing any student to attend specific sessions. The optional nature of the relationship from Class Session to Attendance (1:0..N) supports the rule that "some class sessions may not be attended by any students," while the optional relationship from Student to Attendance (1:0..N) accommodates that "new students will not have attended any class sessions yet."

Session Leader Entity:

This junction table addresses the complex leadership structure described in rule #7, where "a given class session may have a head leader and many assistant leaders." By including a Role attribute (head/assistant), the system can track the different roles leaders play in each session. The mandatory relationship from CLASS_SESSION to SESSION_LEADER (1:N) ensures that "it will always have at least the one head leader," while allowing multiple leaders to assist. This entity also supports tracking when leadership roles may be reversed in different sessions of the same class. This also allows volunteers leaders who are not assigned to any class due to the cardinality between leader and Session Leader being 0:M satisfying rule 3.

Discriminators and Constraints:

Leader Status: The STATUS attribute in Leader (paid/volunteer) serves as a discriminator that supports rule #3's note that "some leaders, especially volunteer leaders, may not be assigned to any class."

Leader Role: The Role attribute in SESSION_LEADER (head/assistant) discriminates between leadership types for each session as required by rule #7.

Project Constraints: This design is fully normalized to 3NF, with all entities properly related through primary and foreign keys, and all business rules represented through appropriate cardinalities, connectivities, and relationship verbs.