## Discovery

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Entities | Attributes | | Users | **User ID, Username, Password, First Name, Last Name, email, last update** | | Students | **Student ID, User ID, enrollment date** | | Instructors | **Instructor ID, user ID, last update** | | ADMINS | **Admin ID, user ID, last update** | | Courses | **Course ID, course name, course subject, instructor id** | | Enrollments | **student id**, **course id, progress %**, **completion** **status**, **last updated** | | Compensations | **Compensation ID, instructor ID**, **enrollment bonus,** **course success bonus, last update** | |  |

## Relationships And Cardinality

**Relationships:**

|  |  |  |
| --- | --- | --- |
| Students | Are | Users |
| Admins | Are | Users |
| Instructors | Are | Users |
| Students | Enroll | Courses |
| Instructors | Teach | Courses |
| Instructors | Receive | Compensations |
| Enrollment | Tracks | Courses |

**Cardinality:**

|  |  |  |  |
| --- | --- | --- | --- |
| Users | – Students | One to One | For one user, there can be one student. |
| Users | – Admins | One to One | For one user, there can be one admin |
| Users | -Instructors | One to One | For one user, there can be one instructor |
| Instructors | – Courses | One to Many | One instructor can teach multiple courses |
| Courses | – Instructors | Many to one | Many courses can have one instructor |
| Instructors | - Compensations | One to many | One instructor can receive multiple compensations |
| Compensation | – Instructors | Many to one | Many compensations can be for one instructor |
| Students | – Enrollments | One to many | One student can enroll in multiple courses |
| Courses | – Enrollments | One to many | One course can have many enrollments |
| Enrollments | – Courses | Many to one | Many enrollments can belong to one course |

# SuperType & SUBTYPE ENTITIES

**Supertype: Users**

**Subtypes: Students, Instructors, Administrators**

Reasoning: Share many common features, first name, last name, username, all of them need a user ID for other tables but are also distinct enough to have unique features so that they need to be identified. Students need a student ID to register for courses, Instructors need their ID for creating courses, or checking on student progress.

# STRONG & WEAK ENTITIES

**Strong Entities: Users, Students, Admins, Instructors, Courses**

**Weak Entities: Enrollments, Compensations**

All the entities are strong except for enrollments and compensations due to their ability to exist independently, with their own unique identifiers. Enrollments and Compensations are weak particularly due to their dependence on other primary keys. Enrollments relies on student\_id and course\_id, a composite primary key of which comes from other tables. Without these keys it would not exist. Compensations specifically relies on instructor\_id, despite having a compensation\_id to track the values in the table. With instructor\_id, its existence is tied to instructors. If there is no instructor present, it would not exist.

# A paper with writing on it Description automatically generatedapply normalization

A close-up of a paper

Description automatically generatedHere are some of the first ideas for the schema model:

A close-up of a list

Description automatically generated

After putting some of the ideas into a MySQL Model:

A computer screen shot of a computer

Description automatically generatedThe compensations table was the one that changed the most, with its merging of the course\_success\_compensations and enrollment\_compensation. This was done to reduce redundancy, as having 3 different IDs, and three different tables all created for 2 slightly different compensations felt unnecessary and overly complicated. EnrollmentID, and userEnrollmentDate were dropped to establish enrollments as a weak table with a primary composite key reliant on students and courses. Multiple tables had their ID datatypes change, such as course id which I decided to make more similar to Avila’s registration course IDs.

Final Schema Model:

A screenshot of a computer

Description automatically generated