Library Management System

Tables

- **Students**
- **Books**
- **Students_Books** (Join Table for Students and Books)
- **Libraries** (Additional Table for Tracking Borrowing Details)

TABLE RELATIONSHIP :

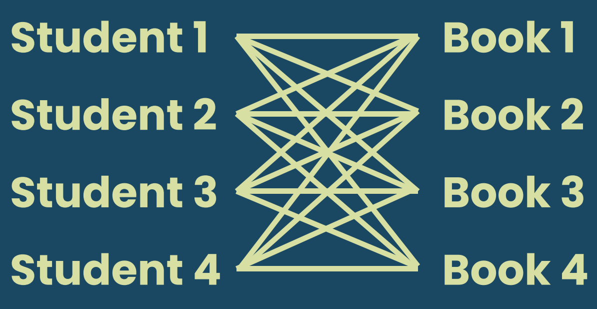
Many-to-Many between Students and Books :

- **Student.belongsToMany(Book, { through: StudentBook }) ;**
  - Defines that a student can borrow many books.

$$\begin{tikzcd}
 & \text{Book 1} \\
\text{Student} \arrow[ru] \arrow[r] \arrow[rd] \arrow[rdd] & \text{Book 2} \\
 & \text{Book 3} \\
 & \text{Book 4}
\end{tikzcd}$$

- **Book.belongsToMany(Student, { through: StudentBook }) ;**
  - Defines that a book can be borrowed by many students.

$$\begin{tikzcd}
 & \text{Student 1} \\
\text{Book} \arrow[ru] \arrow[r] \arrow[rd] \arrow[rdd] & \text{Student 2} \\
 & \text{Student 3} \\
 & \text{Student 4}
\end{tikzcd}$$



One-to-Many between Students and Library :

- **Student.hasMany(Library) ;**
  - A student can have many borrowing records.
- **Library.belongsTo(Student) ;**
  - Each borrowing record belongs to one student.

$$\begin{tikzcd}
 & \text{Record 1 (Each record belongs to one student)} \\
\text{Student} \arrow[ru] \arrow[r] \arrow[rd] \arrow[rdd] & \text{Record 2 (Each record belongs to one student)} \\
 & \text{Record 3 (Each record belongs to one student)} \\
 & \text{Record 4 (Each record belongs to one student)}
\end{tikzcd}$$

One-to-Many between Books and Library :

- **Book.hasMany(Library) ;**
  - A book can have many borrowing records.
- **Library.belongsTo(Book) ;**
  - Each borrowing record belongs to one book.

$$\begin{tikzcd}
 & \text{Record 1 (Each record belongs to one book)} \\
\text{Book} \arrow[ru] \arrow[r] \arrow[rd] \arrow[rdd] & \text{Record 2 (Each record belongs to one book)} \\
 & \text{Record 3 (Each record belongs to one book)} \\
 & \text{Record 4 (Each record belongs to one book)}
\end{tikzcd}$$

Sequelize

- Sequelize plays a key role as an ORM (Object-Relational Mapping) tool, facilitating interactions between the application and the **MySQL** database.

1. Database Connection and Configuration
2. Model Definition
3. Relationships and Associations
4. Database Synchronization
   a. The " **sequelize.sync({ force: false })** " command synchronizes the models with the database schema. This means that it creates the necessary tables in the database if they don't already exist, based on the defined models and relationships.
5. CRUD Operations
   a. " **Student.create** " creates a new student record.
   b. " **Student.findAll** " fetches all student records.
   c. " **Student.findByPk** " retrieves a student by their primary key.
   d. " **Student.update** " updates an existing student's information.
   e. " **Student.destroy** " deletes a student record.