### Overview

The project involves creating a comprehensive school management system using MongoDB as the database backend and Java Swing for the front-end interface. This system manages various aspects of a school's operations, including students, teachers, departments, courses, exam schedules, results, classes, and attendance records. The data for this system is imported from JSON files into MongoDB collections. Each collection is interrelated through primary and foreign keys, ensuring data integrity and facilitating complex queries.

### Technologies Used

* **Database**: MongoDB
* **Frontend**: Java Swing
* **Data Import**: JSON files

### Project Implementation Details

### Database Design

* MongoDB is used as the database backend due to its flexibility and scalability.
* Collections are designed to represent different entities within the school system, with appropriate relationships established using primary and foreign keys.

### Frontend Development with Java Swing

* Java Swing is utilized to create a user-friendly graphical interface for interacting with the database.
* Various forms and panels are designed to allow users to view, add, edit, and delete records from different collections.
* GUI components such as buttons, text fields, tables, and dialogs are used to facilitate user interactions.

### Importing JSON Data

* JSON files containing sample data for students, teachers, departments, courses, results, exam schedules, classes, and attendance are imported into MongoDB.
* Data import scripts or utilities are developed to read JSON files and insert data into the corresponding MongoDB collections.

### Interfacing MongoDB with Java Swing

* Java MongoDB driver (such as MongoDB Java Driver or Morphia) is used to connect to the MongoDB database from the Java Swing application.
* CRUD (Create, Read, Update, Delete) operations are implemented to interact with MongoDB collections based on user actions in the GUI.
* The JSON collections in our school management system project are intricately linked, representing the complex web of relationships inherent in an educational ecosystem. At the core of this interconnected network are the students, teachers, departments, courses, results, exam schedules, classes, and attendance records, each playing a crucial role in the functioning of the institution.

### Relationships among the Json Files:

* Students are associated with departments through a one-to-many relationship, where each student belongs to a single department. Additionally, students enroll in multiple courses, forming a many-to-many relationship between students and courses, facilitated by an array of course identifiers within the student collection. Teachers, much like students, are affiliated with departments, indicating their area of expertise and responsibility within the institution. They also teach multiple courses, resulting in a many-to-many relationship between teachers and courses, similar to that of students.
* Departments serve as organizational units within the school, with each department headed by a designated teacher. This one-to-one relationship between departments and teachers underscores the hierarchical structure of the institution. Courses, on the other hand, are affiliated with specific departments, reflecting their academic focus and alignment with departmental objectives. Furthermore, courses are taught by individual teachers, establishing a direct link between course offerings and instructional staff.
* Results, exam schedules, classes, and attendance records are intricately intertwined with students, courses, and teachers, forming a complex network of dependencies. Results are tied to both students and courses, reflecting individual academic achievements within specific courses. Exam schedules dictate the timing and logistics of assessments for various courses, ensuring coherence and coordination across the academic calendar. Classes, representing instructional sessions, are associated with both courses and teachers, indicating the alignment of pedagogical content with instructional delivery.
* Attendance records capture student participation in classes, with each record linked to a specific student and class. This relationship underscores the importance of monitoring student engagement and accountability within the learning environment. Together, these interconnected JSON collections encapsulate the multifaceted nature of school operations, highlighting the interdependencies and interactions that drive educational excellence.

### Database Schema

### Attendance

### 

### Classes

### 

### Course

### 

### Department

### 

### Exams

### 

### Result

### 

### Student

### 

### Teacher

### 

### Json Files

### Class Json

### 

### Attendance Json

### 

### Department Json

### 

### Course Json

### 

### Result Json

### 

### Exam Schedule Json

### 

### Student Json

### 

### Teacher Json

### 

### Importing Json MongoDB Shell/Compass

### MongoDB Shell

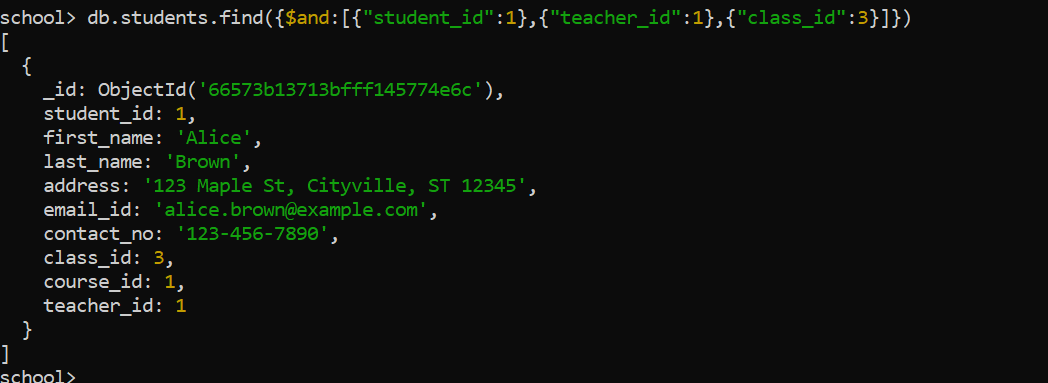
### 

### MongoDB Compass

### 

**MongoDB Queries**

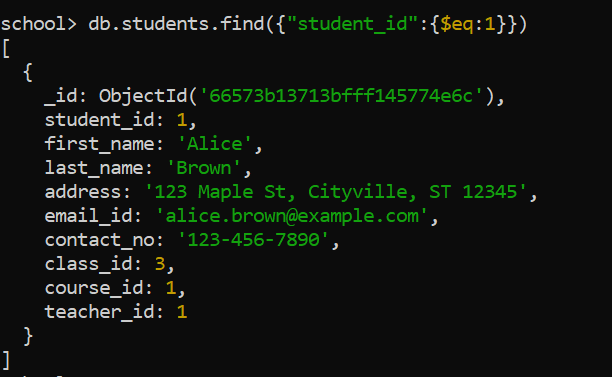
**And Operation**

****

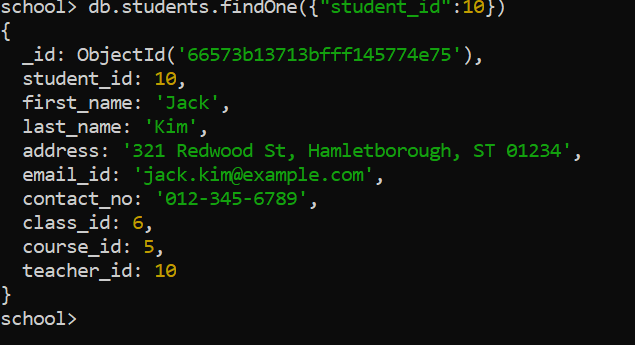
**Delete One**

****

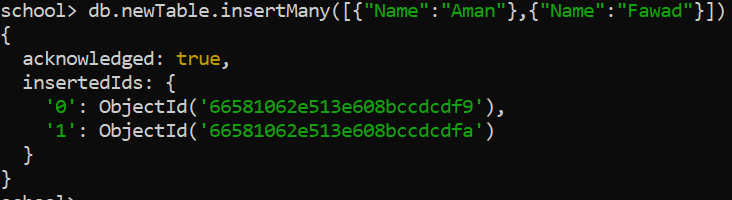
**Find with condition**

****

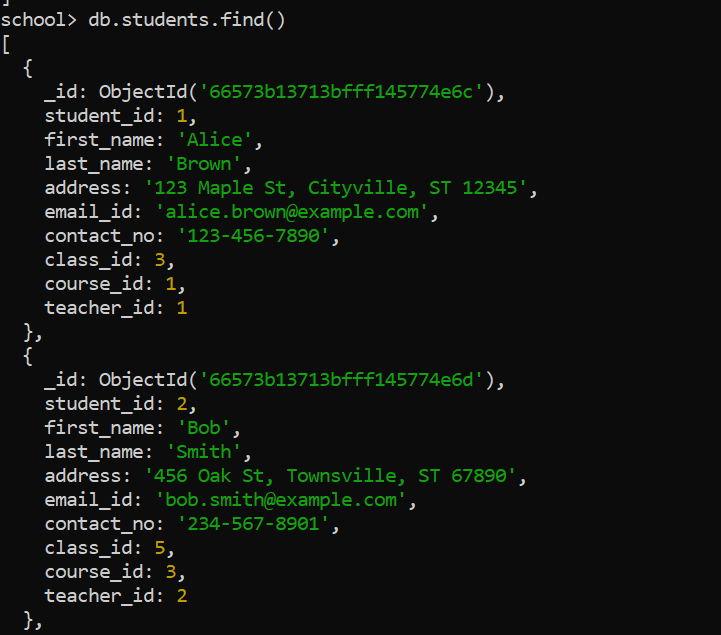
**Find One**

****

**Insert Many**

****

**Find All Collection**

****