Firebase is a realtime NoSQL database that uses a JSON structure.

Unlike MySQL which uses tables, NoSQL uses either a hierarchy or tree like model.

**MySQL & NoSQL (JSON) Comparison**

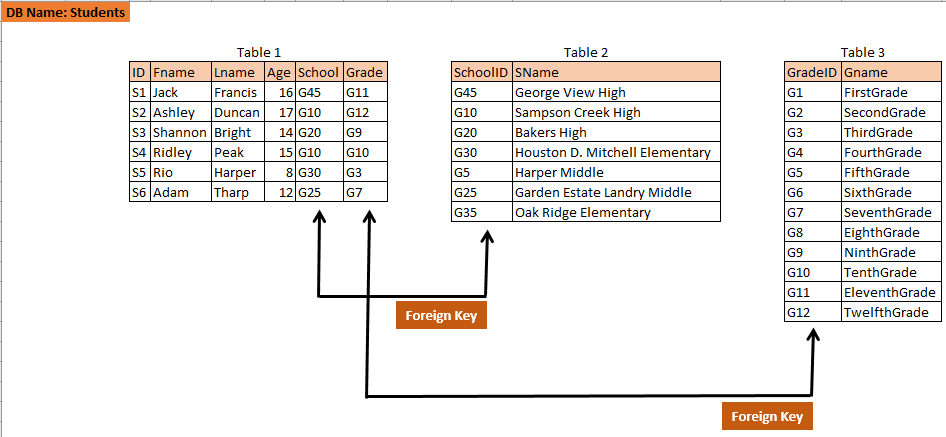
1. MySQL structure

In this structure, tables are required to make the relationships work. These tables are not dynamic and cannot hold sub-categories. Data that poses deletion or update anomalies must be put into their own table. In the case that a relationship has a many-to-many relationship such as:

* Issue: many books are written by many authors.

A bridge table would need to be introduced to reduce the many to many relationships

* Solution: 1 book can have many authors; 1 author can write many books



Notice how each “subject/topic” are in their own respective tables instead of just one. Students are in their own table, schools are in there own and grades are in their own.

In Table1 each student is given an ID for the school and grade they are attending. S1 attends George View High and is in the 11th grade.

2. NoSQL Structure (JSON) [Firebase]

In the JSON structure, the way data is store can be dynamic. Meaning not data doesn’t necessarily have to be in their own “document”. You can have sub-categories of attributes if you want, although Firebase suggest only doing this to a limit of 1-2 sub-categories per attribute.

In firebase, you have a Database and the data is stored in “Collections” and a collection consists of “Documents”. You can think of it as this:

**Database**

-> Collections

->Documents

-> Fields

For example, I have a website that keeps track of all students in the district. We need first and last name, age, grade, and school.

In firebase it would look like this:

**Students** [database]

->StudentInfo [collection]

->S1 [document id]

->FName: Jack [attribute]

->LName: Francis [attribute]

->Age: 16 [attribute]

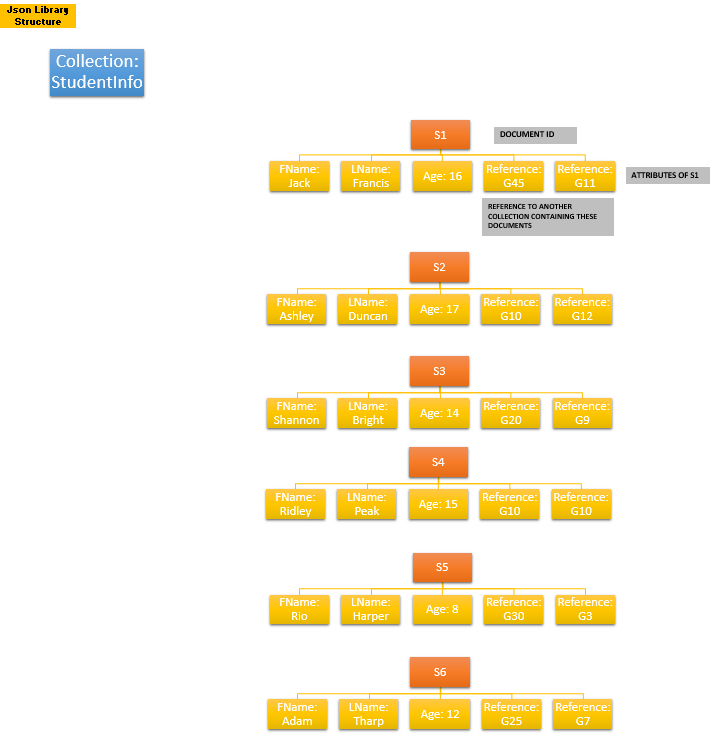
->School: reference [attribute]

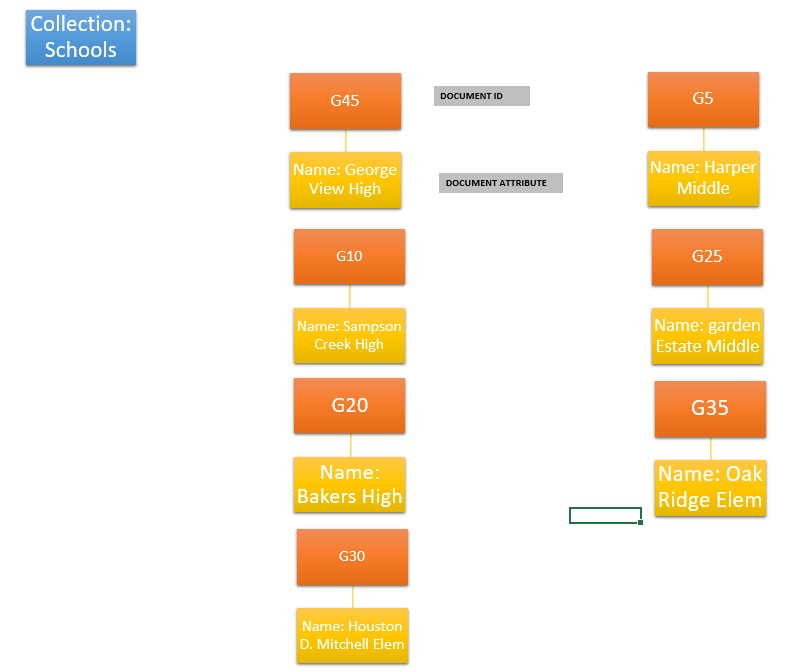
->path: /school/G45

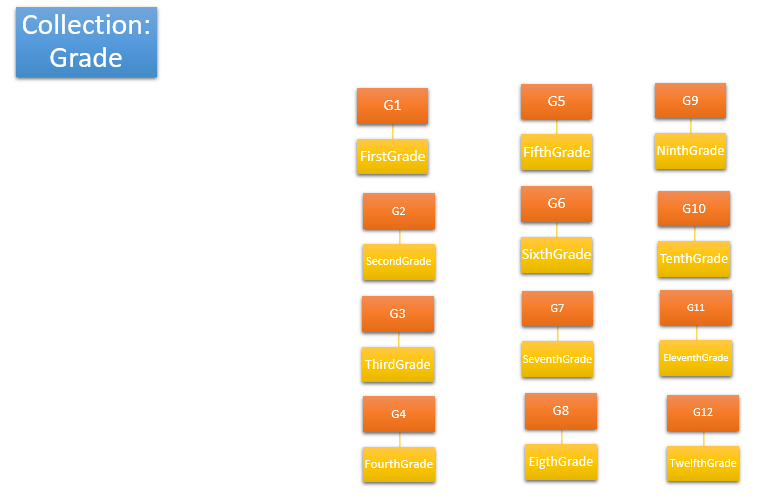
->Grade: reference [attribute]

->path: /Grade/G11

Reference is equivalent to Foreign Keys in a MySQL database. The reference requires a path to the collection and then to the ID containing the value of the correct Field.







So overall, you want to group common things into the same collection.

If you require user login, there should be a *User’s* collection only storing the user’s login credentials and profile information.

Same as above, all of my *Students* are in a separate collection from the school and grade. Using a reference to grab the information from the other collections and syncing it to the correct student’s document.

**Things to avoid in JSON**

While you can nest data in this structure, it advised to not do so because it’s just bad practice. For example, you could very well do this:

**Students** [database]

->StudentInfo [collection]

->S1 [document id]

->FName: Jack [attribute]

->LName: Francis [attribute]

->Age: 16 [attribute]

->BirthMonth: August

->School: George View High

->Grade: Eleventh

Notice how the *School* and *Grade* attributes are no longer references. This is because they no longer have their own separate collections and are just added to the *Student* collection. Instead *School* is an original Field for this student and *Grade* is now a sub-category of school.

Also, see how I added a sub-category to *Age*. The persons *BirthMonth*is now a sub-category of Age.

**Why is the above bad???**

It is simply bad to store the data this way because just like MySQL and other databases, you run the risk of anomalies. Also, it is inefficient to store data that will have multiple relationships to one document. For example:

Jack is the only student who has George View High in his document. However, the more students that get added to the collection, you’ll realize that Jack is not the only one who attends George View High.

So, it is better to list the school in a separate collection that will hold all schools and give each school their own unique document ID so that it can be referenced to multiple students.

**Terms to compare**

|  |  |
| --- | --- |
| MySQL | JSON |
| Database | Database |
| Table | Collection |
| Record | Document |
| Attributes | Field |

**Firebase Console**

