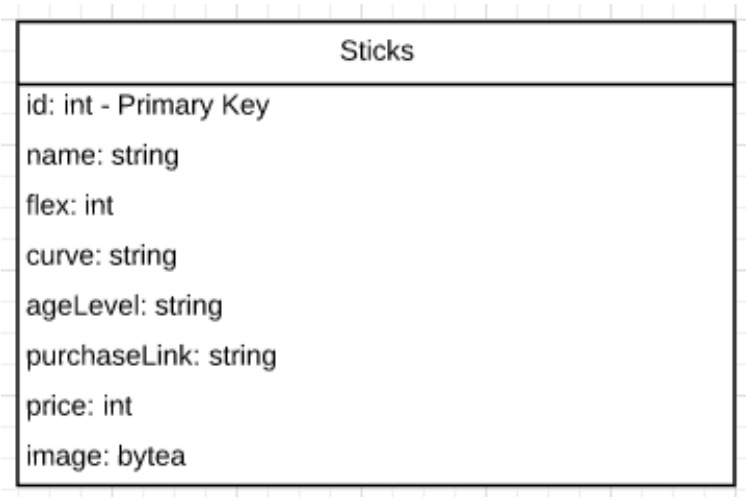
Database Design – Stick It

The Stick It application will use PostgreSQL as the application database.

The Stick It application will use a relational database to store its data in tables. The database will consist of two tables: the sticks table, which will store the data on the hockey sticks which can be returned to users as their result, and the suggestions table, which will store the data that is posted to it by users as they suggest new sticks to add to the sticks table. The tables will be set up as follows:

Sticks Table (MVP)

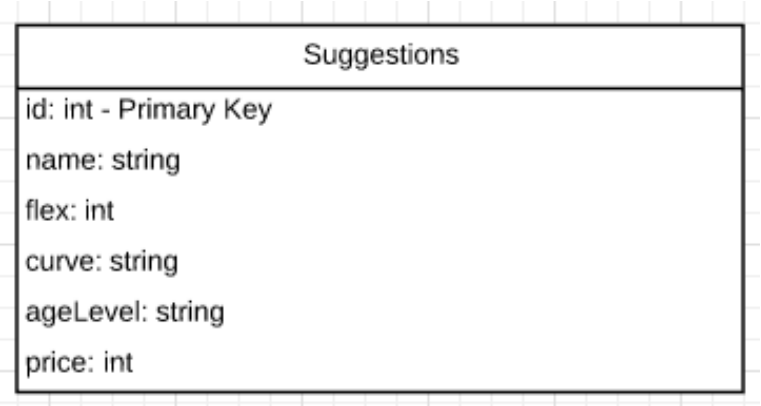


The table will consist of eight columns: id, name, flex, curve, ageLevel, purchaseLink, price, and image. The id column will serve as the primary key of the table and will be auto-assigned as each stick is entered in the table. This table will receive GET requests from the service endpoint. The database will then return one stick record as a JSON response to the service endpoint, which will allow the user to view their result.

An example of what the data would look like stored in the table would appear like:

Id	Name	Flex	Curve	ageLevel	purchaseLink	Price
1	Bauer XCX	100	P98	43	https://www.purehockey.com/product/bauer-vapor-1x-lite-grip-composite-hockey-stick-senior/itm/29896-41/	199
2	Bauer Supreme	70	PM9	24	https://www.purehockey.com/product/bauer-supreme-2s-pro-grip-composite-hockey-stick-junior/itm/35545-21/	249

Suggestions Table (Stretch)



The table will consist of six columns: id, name, flex, curve, ageLevel, and price. The id column will serve as the primary key of the table and will be auto-assigned as each stick is entered in the table. This table will receive POST requests from the service endpoint, which will post the data entered by the user into this table. The database administrator will be in charge of verifying the stick information entered by the user, and can add the suggested information into the sticks table following verification. At that point, the database administrator would be in charge of adding the purchase link and image for that stick, as that information is not requested from the user.

An example of what the data would look like stored in the table would appear like:

Id	Name	Flex	Curve	ageLevel	Price
1	Bauer XCX	100	P98	43	199
2	Bauer Supreme	70	PM9	24	249

Why PostgreSQL

At this point, PostgreSQL makes the most sense for my application, as all of the data stored in the databased will consist of the exact same information. Sticks are not likely to change drastically over time; therefore, I do not feel that I need the flexibility that a schema-less database such as MongoDB offers. Additionally, I believe that the strictly enforced schema of PostgreSQL will help to prevent potential errors by a student programmer like me. PostgreSQL also offers the ability to use functions and create stored procedures to allow it to find information in the database quickly and repeatedly. It also offers future flexibility for my application with features such as table inheritance, which allows tables to inherit data from other tables to keep data consistent. As NoSQL data types are frequently used across the software development field, PostgreSQL has added the ability to handle NoSQL data types.