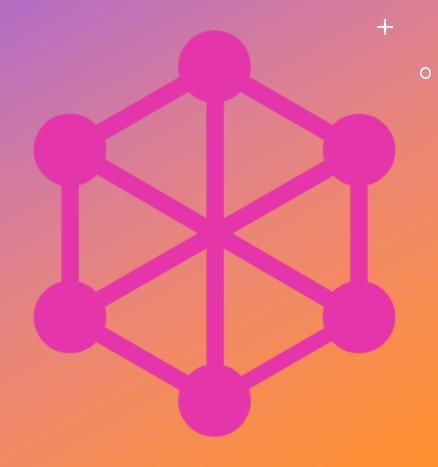
TYPE RELATIONS

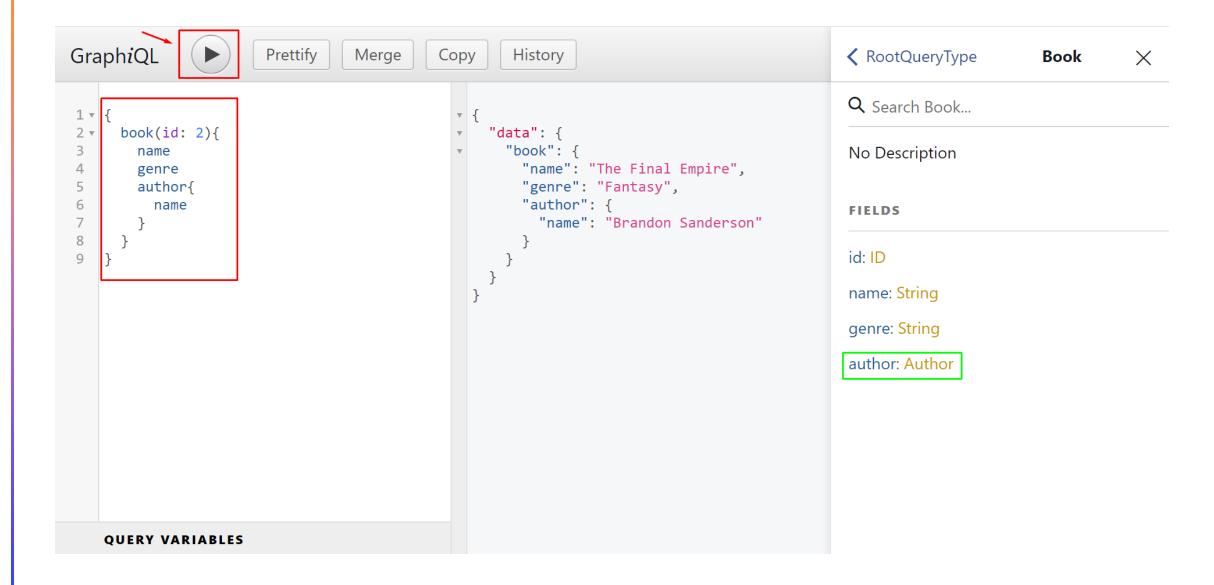
In this GraphQL tutorial I'll show you how we can form relations between our GraphQL Object Types so that we can query (or nest) related data. For example, so that we could query a book and ask for it's author.



... schema.js

```
// dummy data
var books = [
    { name: 'Name of the Wind', genre: 'Fantasy', id: '1', authorId: '1' },
      name: 'The Final Empire', genre: 'Fantasy', id: '2', authorId: '2' },
      name: 'The Long Earth', genre: 'Sci-Fi', id: '3', authorId: '3' },
];
                                                                                            Modify
var authors = [
                                                                                            code
    { name: 'Patrick Rothfuss', age: 44, id: '1' },
      name: 'Brandon Sanderson', age: 42, id: '2' },
     name: 'Terry Pratchett', age: 66, id: '3' }
];
const BookType = new GraphQLObjectType({
    name: 'Book',
    fields: ( ) => ({
       id: { type: GraphQLID },
       name: { type: GraphQLString },
                                                                                            Insert
       genre: { type: GraphQLString },
       author: {
                                                                                            code
            type: AuthorType,
            resolve(parent, args){
                console.log(parent);
               return .find(authors, { id: parent.authorId });
});
```

Go to http://localhost:4000/graphql



Go to http://localhost:4000/graphql

