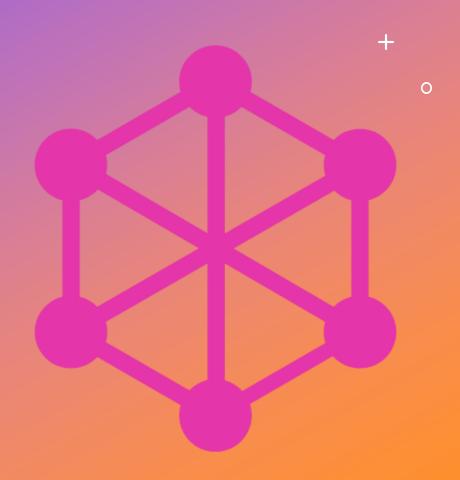
COMPOSING QUERIES



```
import { gql } from "@apollo/client";
                                                                                          queries.js
const getAuthorsQuery = gql`
        authors {
            name
            id
const getBooksQuery = gql`
        books {
            name
            id
const addBookMutation = gql`
    mutation AddBook($name: String!, $genre: String!, $authorId: ID!){
        addBook(name: $name, genre: $genre, authorId: $authorId){
            name
            id
                                                                                              Insert
                                                                                              code
export { getAuthorsQuery, getBooksQuery, addBookMutation };
```

```
queries.js
```

```
import React from "react";
import { useQuery, useMutation } from "@apollo/client";
import { getAuthorsQuery | addBookMutation } from '../queries/queries';
function DisplayAuthors(){
    const { loading, data } = useQuery(getAuthorsQuery);
   if(loading){
       return( <option disabled>Loading authors</option> );
   } else {
       return data.authors.map(author => {
            return( <option key={ author.id } value={author.id}>{ author.name }</option> );
       });
function AddBook() {
   const [book, setBook] = React.useState({ name: "", genre: "", authorId: "" });
   const [addBookMut, { dataMutation }] = useMutation(addBookMutation);
   const handleSubmit = e => {
        e.preventDefault();
       addBookMut({
            variables: {
             name: book.name,
             genre: book.genre,
              authorId: book.authorId,
export default AddBook;
```

