You don't need an ORM

Who am I

👋 Hello! I'm Giacomo

💖 I love functional programming

⭐ I'm part of Gleam's core team

Our running example

```sql
create table if not exists book (
  isbn text primary key,
  title text not null,
  ratings_by_star int[] not null default array[0, 0, 0, 0, 0]
);
create table if not exists genre (
  isbn text not null references book(isbn),
  name text not null,
  primary key (isbn, name)
);
```

We should use the right language for the job

We should use the right language for the job

That's Java

We should use the right language for the job

That's Python

Java

We should use the right language for the job

That's Gleam

Python

Java

We should use the right language for the job

That's SQL!

Sticking to plain old SQL

🔃 Makes us more intentional about what queries are sent to the database

Sticking to plain old SQL

🔃 Makes us more intentional about what queries are sent to the database

🗂️ Forces us to think about what data we're fetching

… we once identified an extremely costly query that joined 12 tables … **Many of these problematic queries are generated by ORMs,** so it’s important to carefully review the SQL they produce and ensure it behaves as expected.

Scaling PostgreSQL to power 800 million users

Sticking to plain old SQL

🔃 Makes us more intentional about what queries are sent to the database

🗃️ Forces us to think about what data we're fetching

🕳️ Nudges us towards a pit of success

We should use the right language for the job

That's SQL!

We should use the right language for the job

That's SQL! ***Or is it?***

```
pub fn list_books(db: Connection) {
  "select book.isbn, book.title
   from book limit 10"
  ▷ pog.query
  ▷ pog.returning({
    use isbn ← decode.field(0, decode.string)
    use title ← decode.field(1, decode.string)
    decode.success(Book(isbn:, title:))
  })
  ▷ pog.execute(db)
}
```

```
pub fn list_books(db: Connection) {
  "select book.isbn, book.title
   from book limit 10"
  ▷ pog.query
  ▷ pog.returning({
    use isbn ← decode.field(0, decode.string)
    use title ← decode.field(1, decode.string)
    decode.success(Book(isbn:, title:))
  })
  ▷ pog.execute(db)
}
```

```
pub fn list_books(db: Connection) {
  "select book.isbn, book.title
   from book limit 10"
  ▷ pog.query
  ▷ pog.returning({
    use isbn ← decode.field(0, decode.string)
    use title ← decode.field(1, decode.string)
    decode.success(Book(isbn:, title:))
  })
  ▷ pog.execute(db)
}
```

```
pub fn list_books(db: Connection) {
  "select book.isbn, book.title
   from book limit 10"
  |> pog.query
  |> pog.returning({
    use isbn <- decode.field(0, decode.string)
    use title <- decode.field(1, decode.string)
    decode.success(Book(isbn:, title:))
  })
  |> pog.execute(db)
}
```

```
pub fn list_books(db: Connection) {
  "select book.isbn, book.title
   from book limit 10"
  ▷ pog.query
  ▷ pog.returning({
    use isbn ← decode.field(0, decode.string)
    use title ← decode.field(1, decode.string)
    decode.success(Book(isbn:, title:))
  })
  ▷ pog.execute(db)
}
```

```
pub fn list_books(db: Connection) {
  "select book.isbn, book.title
   from book limit 10"
  |> pog.query
  |> pog.returning({
    use isbn <- decode.field(0, decode.string)
    use title <- decode.field(1, decode.string)
    decode.success(Book(isbn:, title:))
  })
  |> pog.execute(db)
}
```

```
pub fn list_books(db: Connection) {
  "select book.isbn, book.title, book.average_rating
   from book limit 10"
  |> pog.query
  |> pog.returning({
    use isbn <- decode.field(0, decode.string)
    use title <- decode.field(1, decode.string)
    decode.success(Book(isbn:, title:))
  })
  |> pog.execute(db)
}
```

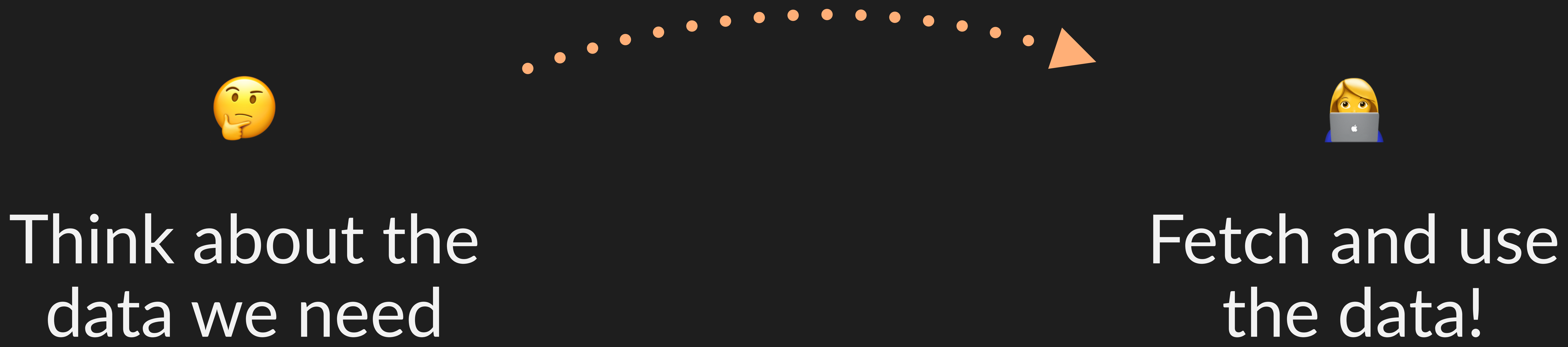
```
pub fn list_books(db: Connection) {
  "select book.isbn, book.average_rating, book.title
   from book limit 10"
  ▷ pog.query
  ▷ pog.returning({
    use isbn ← decode.field(0, decode.string)
    use title ← decode.field(1, decode.string)
    decode.success(Book(isbn:, title:))
  })
  ▷ pog.execute(db)
}
```

🤔

Think about the data we need

👩‍💻

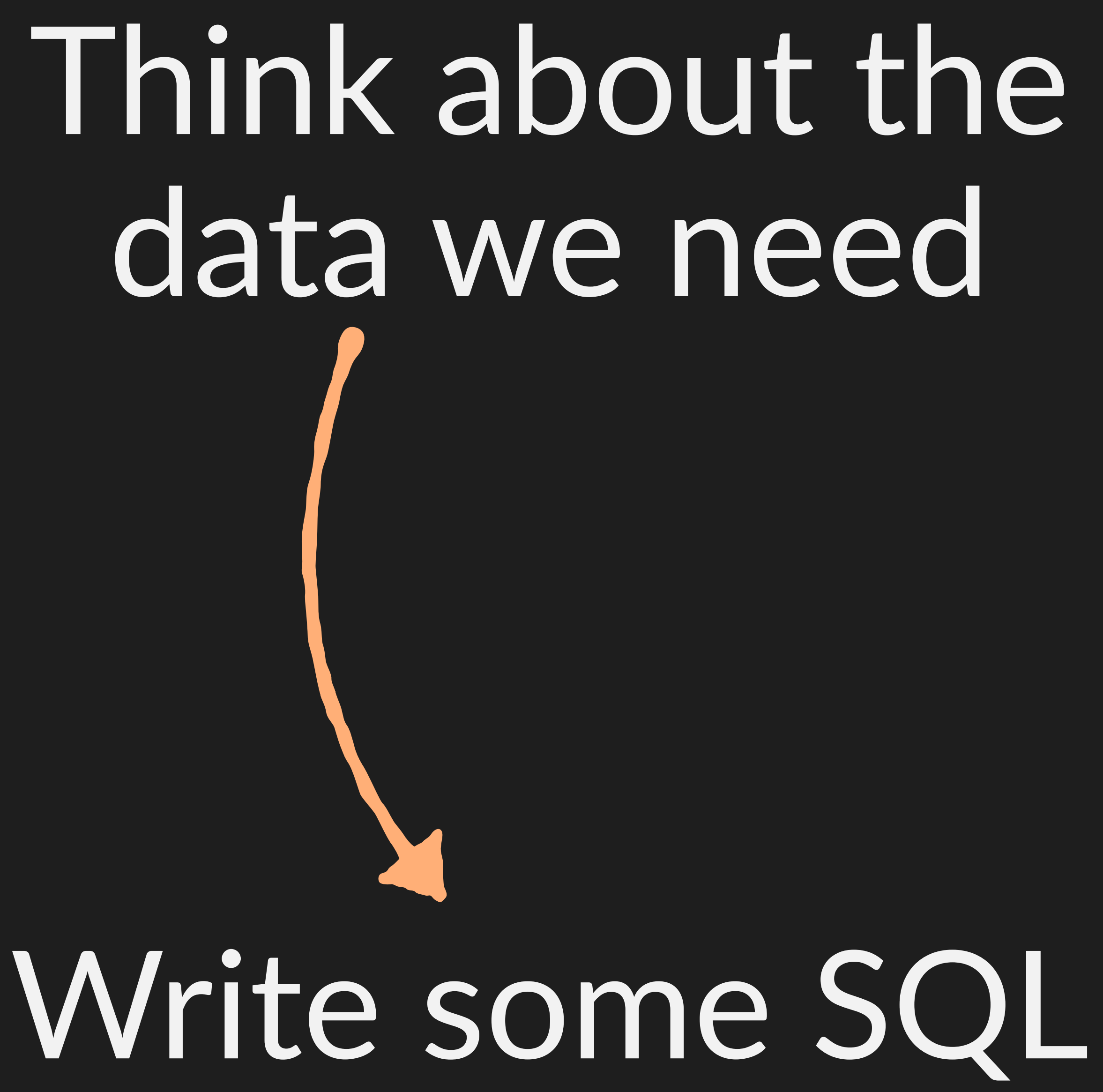
Fetch and use the data!



Think about the data we need

Fetch and use the data!

🤔

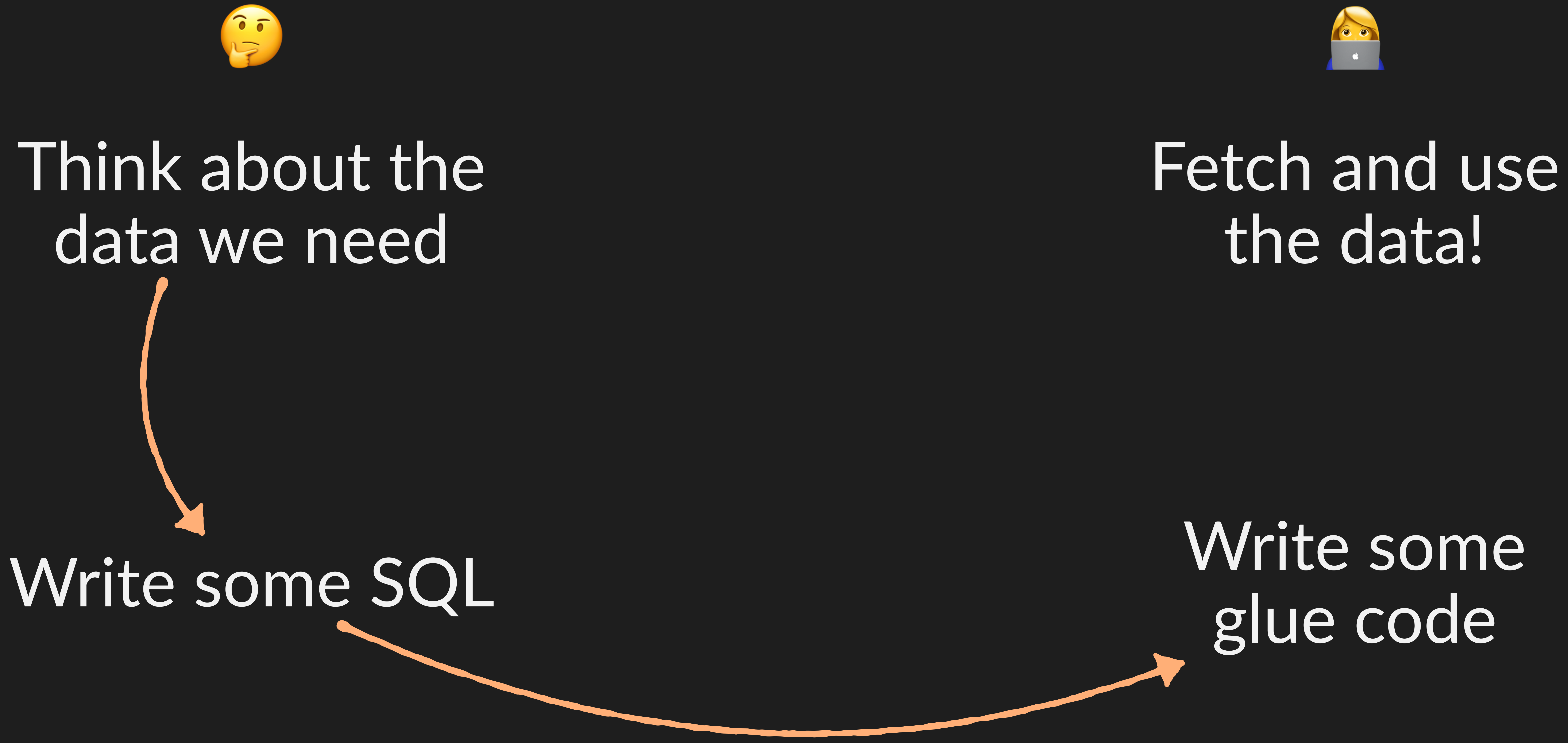


Think about the data we need

Write some SQL

👩‍💻

Fetch and use the data!



🤔

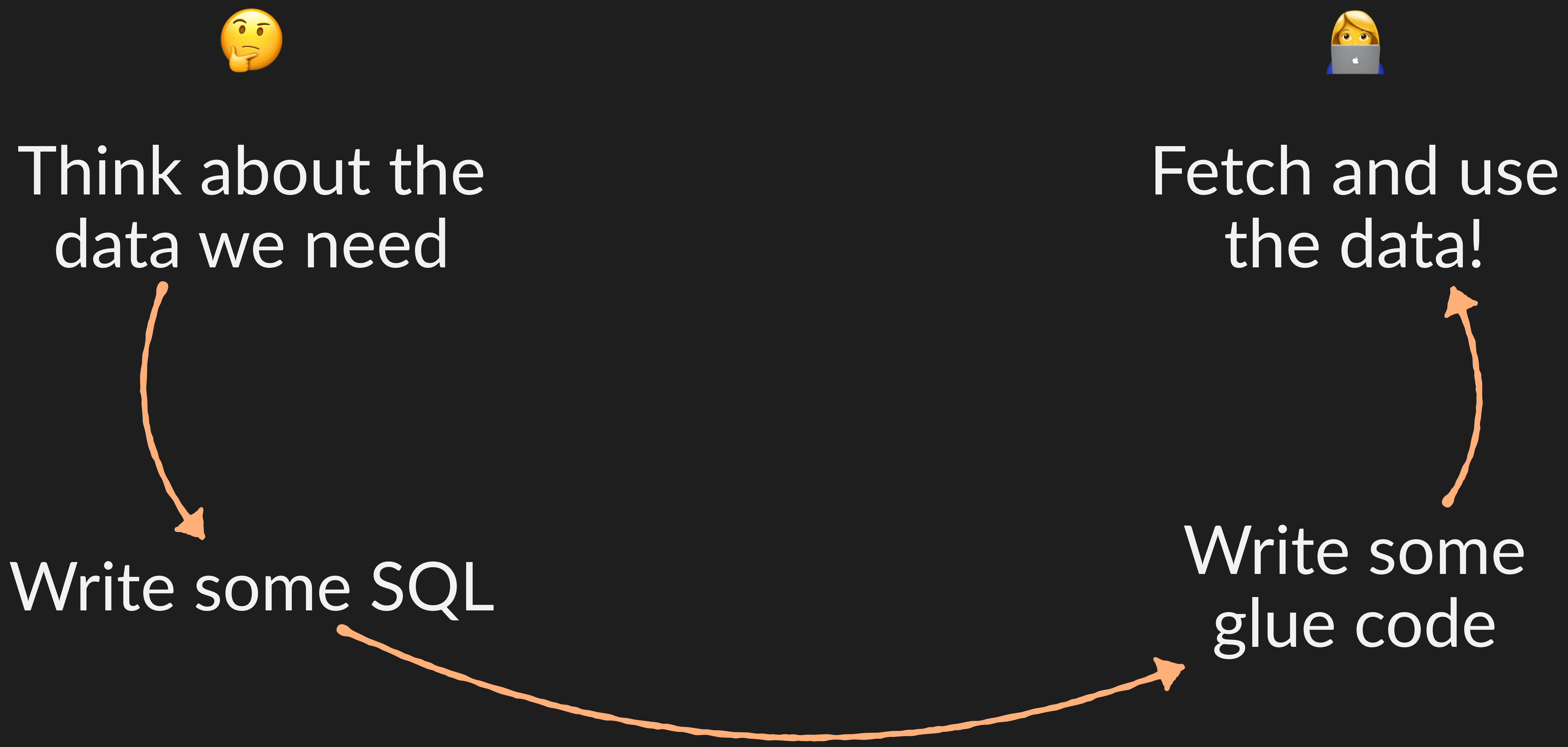
Think about the data we need

Write some SQL

Write some glue code

👩‍💻

Fetch and use the data!



🤔

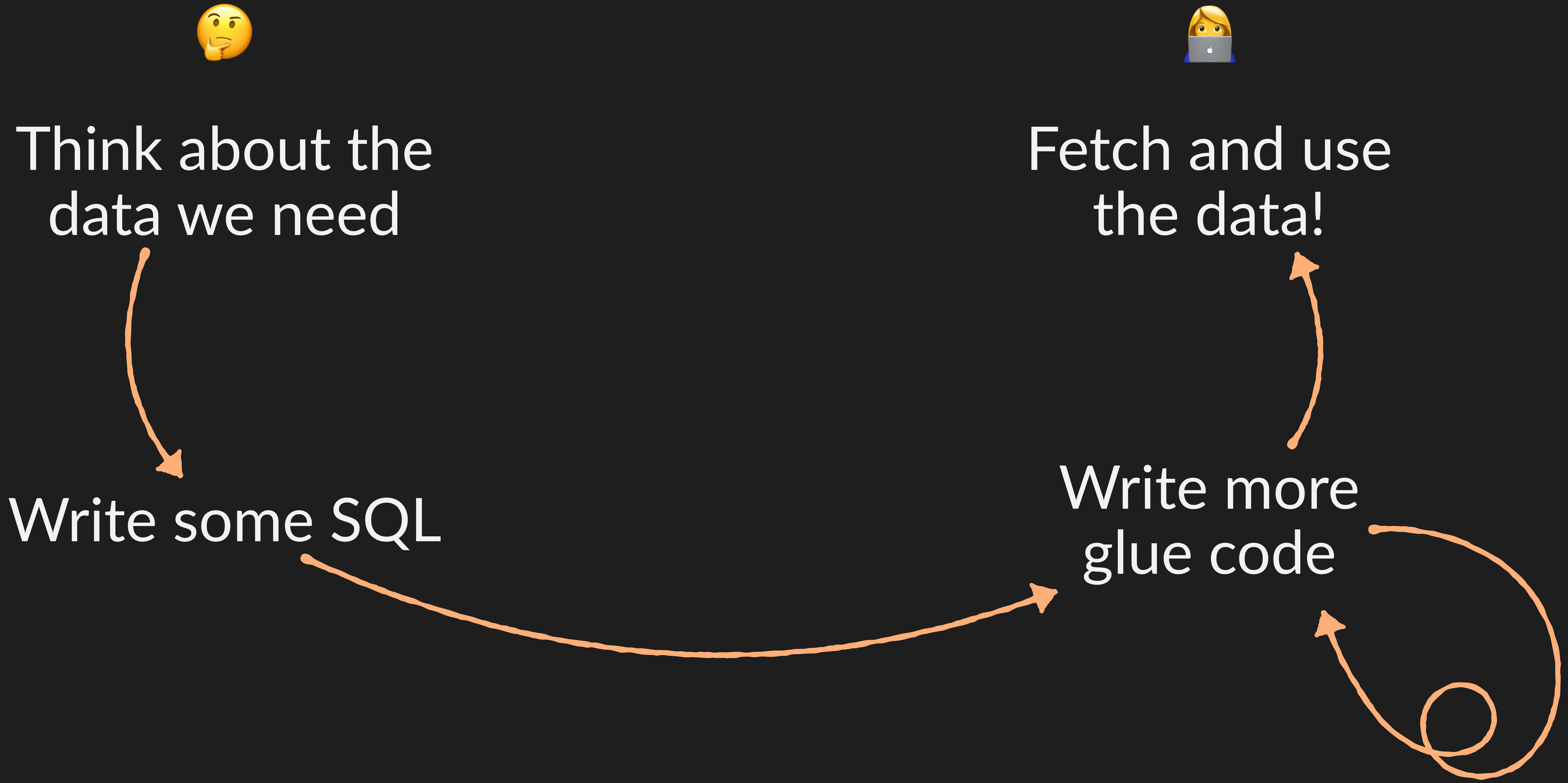
Think about the data we need

Write some SQL

Write some glue code

👩‍💻

Fetch and use the data!



🤔

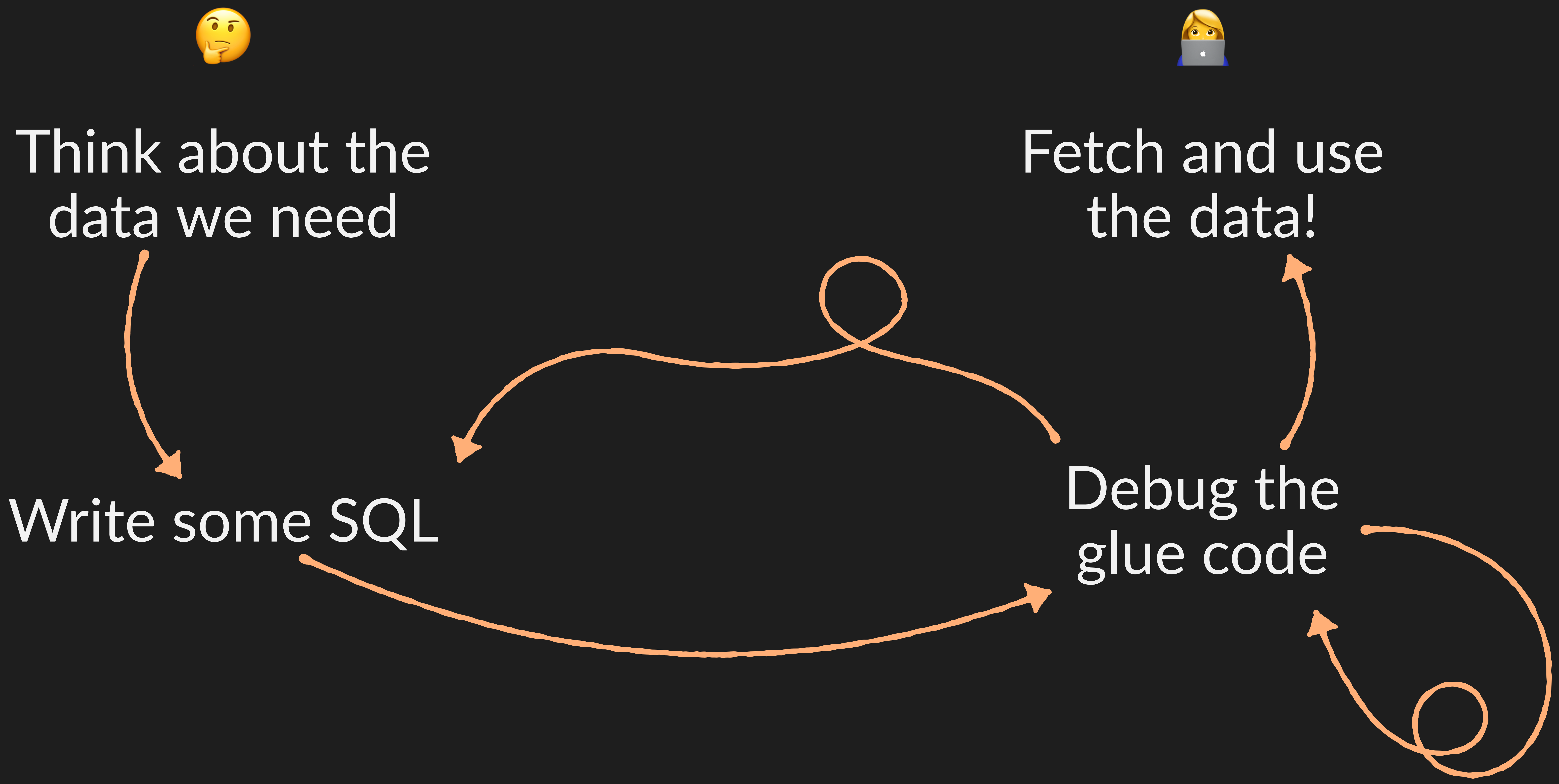
Think about the data we need

Write some SQL

Write more glue code

👩‍💻

Fetch and use the data!



🤔

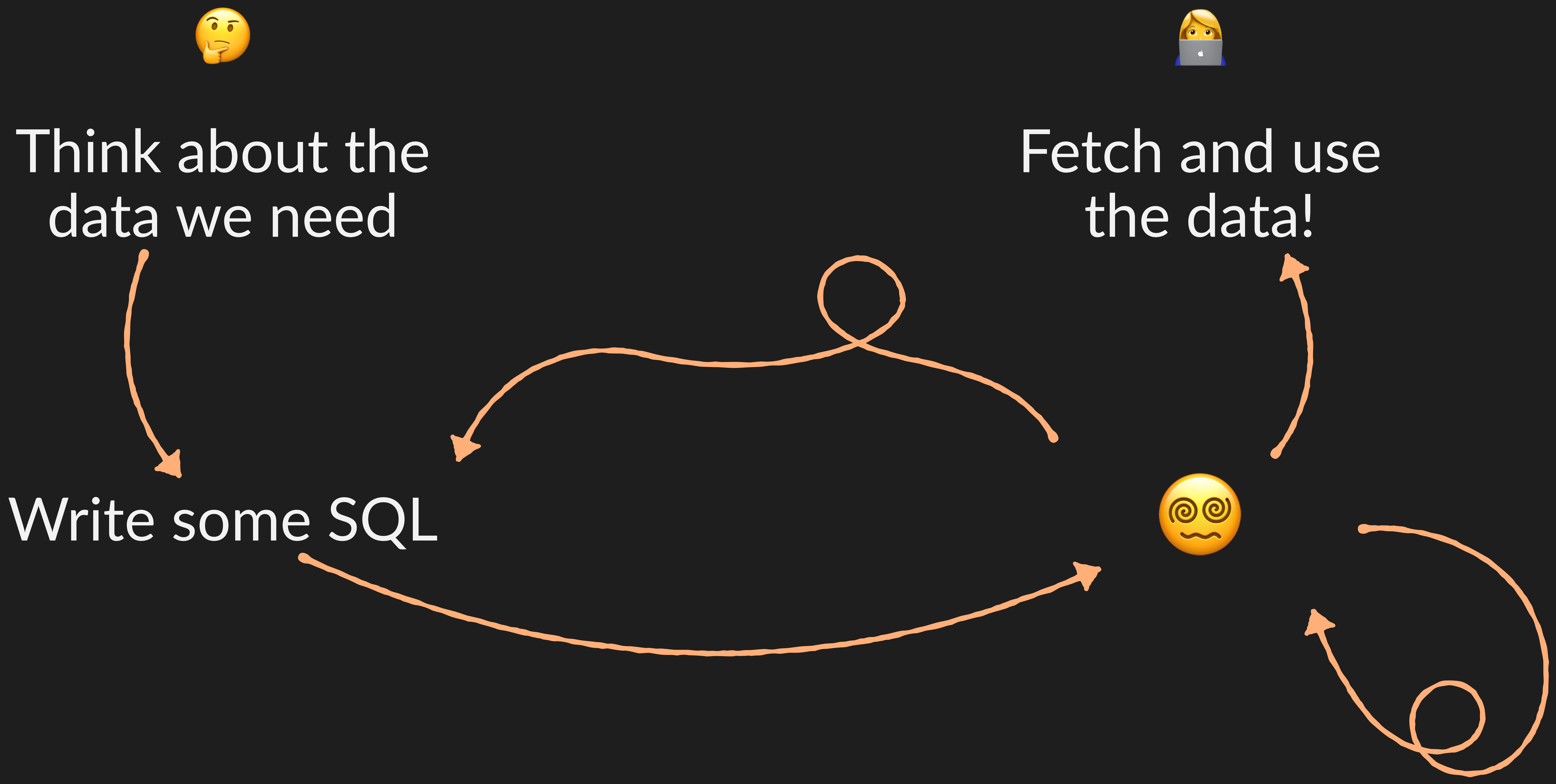
Think about the data we need

Write some SQL

Debug the glue code

👩‍💻

Fetch and use the data!



🤔

Think about the data we need

Write some SQL

😵‍💫

👩‍💻

Fetch and use the data!

We should use the right language for the job

That's SQL! ***Or is it?***

We should use the right language for the job

That's SQL! ***Or is it? It could be!***

🤔 Is it possible to embrace SQL and not compromise on developer experience?

```
book_ranking_query(db)
```

```gleam
case book_ranking_query(db) {
  Ok(rows) -> todo
  Error(_) -> panic as "pesky database"
}
```

Enter Squirrel 🐿️

Time for a live demo!

Great developer experience

🌱 It's plain old SQL files

Great developer experience

🌱 It's plain old SQL files

⚒️ Errors happen at build time

```
> gleam run -m squirrel
```

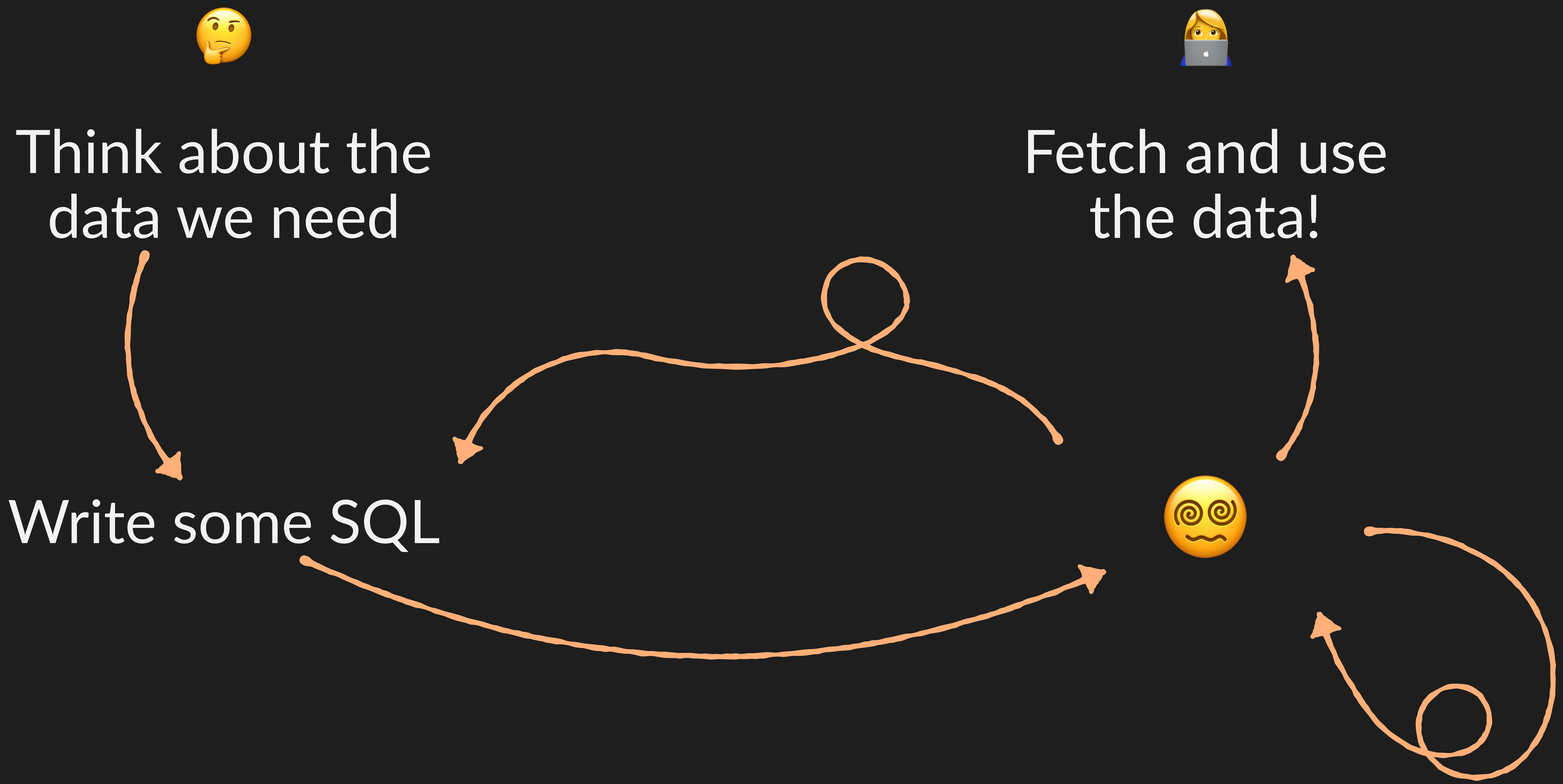
```
Error: Invalid query [42803]
    ╭─ ./src/app/sql/book_and_genres.sql
    │
  1 │ select book.title, genre.name
                         ╰─ column "genre.name" must appear in
                            the GROUP BY clause or be used in an
                            aggregate function
  2 │ from book join genre using(isbn)
  3 │ group by book.isbn
    ┆
```

Great developer experience

🌱 It's plain old SQL files

⚒️ Errors happen at build time

🍃 Refactoring is a breeze



🤔

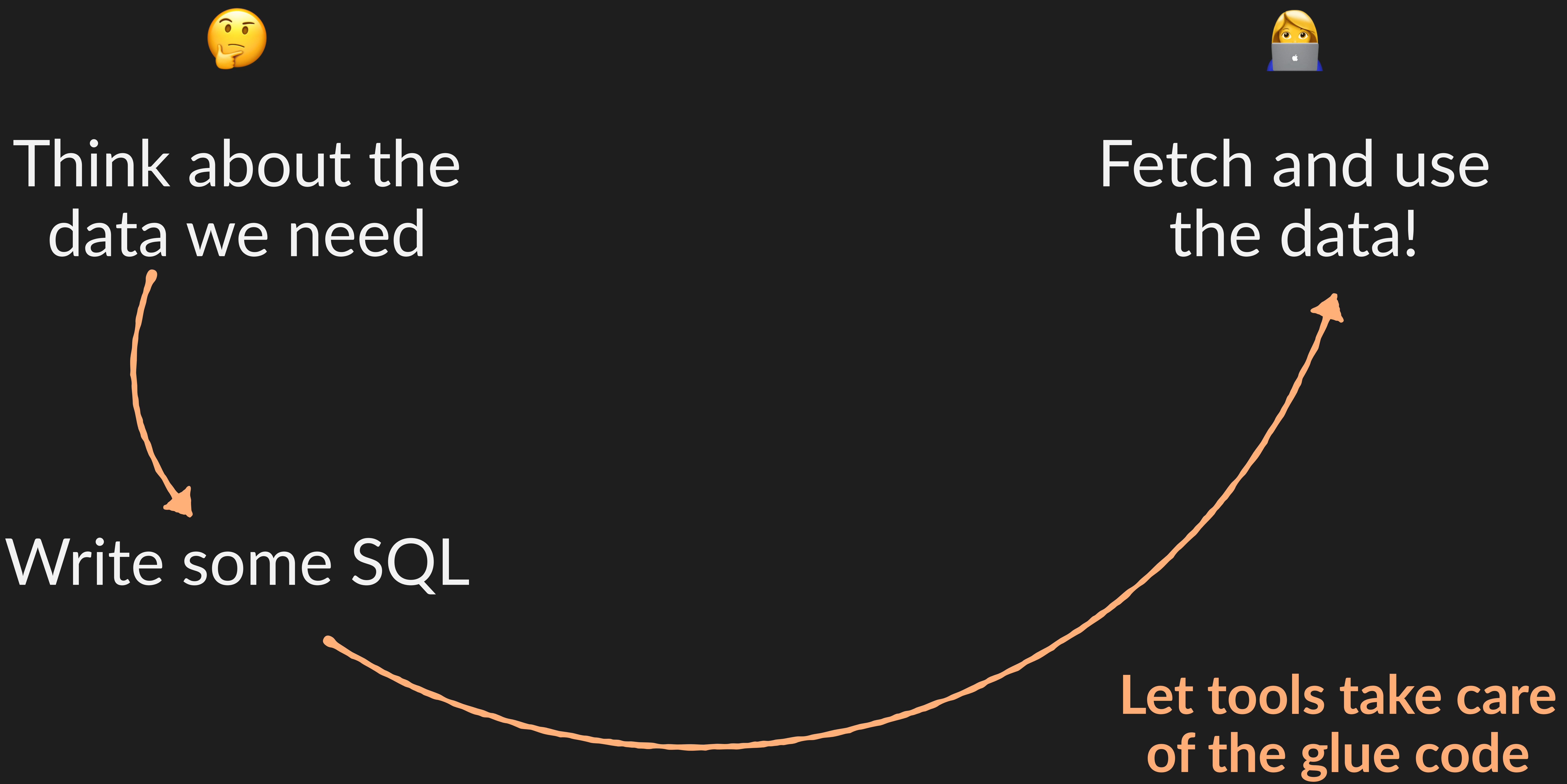
Think about the data we need

Write some SQL

😵‍💫

👩‍💻

Fetch and use the data!



🤔

Think about the data we need

Write some SQL

👩‍💻

Fetch and use the data!

Let tools take care of the glue code

You don't need an ORM

Share your feedback!

giacomocavalieri.me/feedback/fosdem-2026

Share your feedback!

Let's keep in touch!

🦋 *giacomocavalieri.me/socials*

✉️ *info@giacomocavalieri.me*

🧑‍💻 *github.com/giacomocavalieri/squirrel*