

# Introducing queries

INTRODUCTION TO SQL



**Izzy Weber**

Curriculum Manager, DataCamp

# What is SQL useful for?

patrons

card_num	name	member_year	total_fine
54378	Izzy	2012	9.86
94722	Maham	2020	0
45783	Jasmin	2022	2.05
90123	James	1989	0

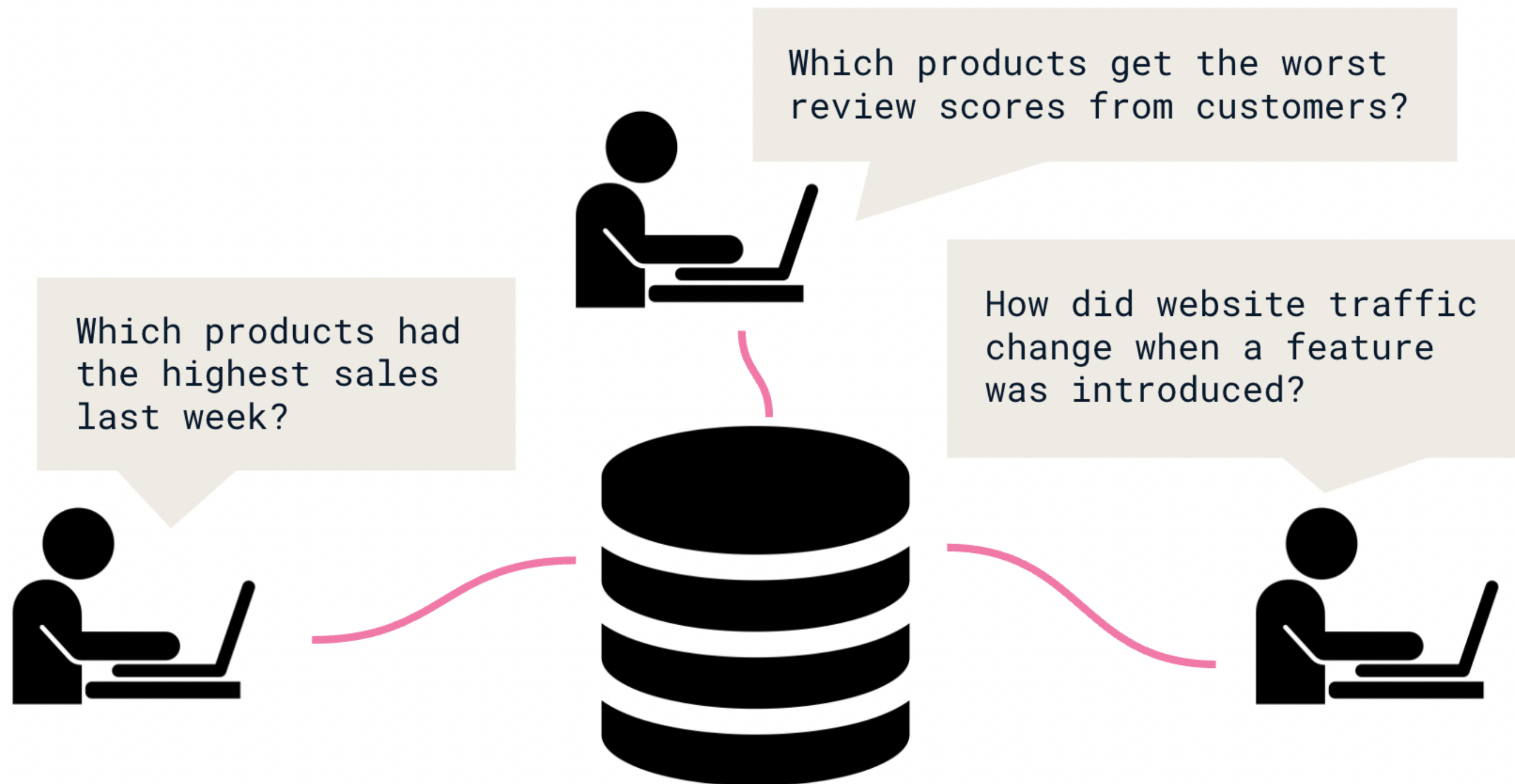
books

id	title	author	genre	pub_year
638	Being Mortal	Atul Gawande	Non-Fiction	2015
912	Educated	Tara Westover	Non-Fiction	2018
322	Night	Elie Wiesel	Non-Fiction	1956
156	Where the Wild Things Are	Maurice Sendak	Childrens	1963

checkouts

id	start_date	due_date	card_num	book_id
567	2022-05-13	2022-05-27	54378	638
568	2022-06-10	2022-06-24	54378	322
569	2022-06-27	2022-07-11	45783	156
570	2022-08-14	2022-08-28	90123	912

# Best for large datasets



# Keywords

*Keywords* are reserved words for operations

**SELECT** name

patrons

card_num	name	member_year	total_fine
54378	Izzy	2012	9.86
94722	Maham	2020	0
45783	Jasmin	2022	2.05
90123	James	1989	0

Common keywords: **SELECT** , **FROM**

**FROM** patrons

patrons	
card_num	INT
name	VARCHAR
member_year	INT
total_fine	NUMERIC

checkouts	
id	INT
start_date	DATE
due_date	DATE
card_num	INT

books	
id	INT
title	VARCHAR
author	VARCHAR
genre	VARCHAR
pub_year	INT

# Our first query

```
SELECT name  
FROM patrons;
```

```
| name |  
|-----|  
| Izzy |  
| Maham |  
| Jasmin |  
| James |
```

- Query results often called *result set*

## patrons

card_num	name	member_year	total_fine
54378	Izzy	2012	9.86
94722	Maham	2020	0
45783	Jasmin	2022	2.05
90123	James	1989	0

# Selecting multiple fields

```
SELECT card_num, name
FROM patrons;
```

card_num	name
54378	Izzy
94722	Maham
45783	Jasmin
90123	James

```
SELECT name, card_num
FROM patrons;
```

name	card_num
Izzy	54378
Maham	94722
Jasmin	45783
James	90123

# Selecting multiple fields

```
SELECT name, card_num, total_fine
FROM patrons;
```

card_num	name	total_fine
54378	Izzy	9.86
94722	Maham	0
45783	Jasmin	2.05
90123	James	0

# Selecting all fields

```
SELECT *  
FROM patrons;
```

card_num	name	member_year	total_fine
54378	Izzy	2012	9.86
94722	Maham	2020	0
45783	Jasmin	2022	2.05
90123	James	1989	0



# Let's practice!

INTRODUCTION TO SQL

# Writing queries

INTRODUCTION TO SQL



**Izzy Weber**

Curriculum Manager, DataCamp

# Aliasing

Use *aliasing* to rename columns

```
SELECT name AS first_name, year_hired  
FROM employees;
```

first_name	year_hired
-----	-----
Darius	2020
Raven	2017
Eduardo	2022
Maggie	2021
Amy	2020
Meehir	2021

# Selecting distinct records

```
SELECT year_hired  
FROM employees;
```

```
| year_hired |  
|-----|  
| 2020      |  
| 2017      |  
| 2022      |  
| 2021      |  
| 2020      |  
| 2021      |
```

```
SELECT DISTINCT year_hired  
FROM employees;
```

```
| year_hired |  
|-----|  
| 2020      |  
| 2017      |  
| 2022      |  
| 2021      |
```

# DISTINCT with multiple fields

employees

id	name	dept_id	job_level_id	year_hired
54378	Darius	1	3	2020
94722	Raven	2	3	2017
45783	Eduardo	2	1	2022
90123	Maggie	3	2	2011
67284	Amy	2	2	2009
26148	Meehir	3	3	2021

```
SELECT dept_id, year_hired
FROM employees;
```

```
| dept_id | year_hired |
|-----|-----|
| 1       | 2020       |
| 2       | 2017       |
| 2       | 2022       |
| 3       | 2021       |
| 2       | 2020       |
| 3       | 2021       |
```

# DISTINCT with multiple fields

```
SELECT DISTINCT dept_id, year_hired  
FROM employees;
```

dept_id	year_hired
1	2020
2	2017
2	2022
3	2021
2	2020

# Views

- A *view* is a virtual table that is the result of a saved SQL `SELECT` statement
- When accessed, views automatically update in response to updates in the underlying data

```
CREATE VIEW employee_hire_years AS
SELECT id, name, year_hired
FROM employees;
```

# Using views

```
SELECT id, name  
FROM employee_hire_years;
```

id	name
54378	Darius
94722	Raven
45783	Eduardo
90123	Maggie
67284	Amy
26148	Meehir



# Let's practice!

INTRODUCTION TO SQL

# SQL flavors

INTRODUCTION TO SQL



**Izzy Weber**

Curriculum Manager, DataCamp

# SQL flavors

- Both free and paid
- All used with relational databases
- Vast majority of keywords are the same
- All must follow universal standards
- Only the additions on top of these standards make flavors different



<sup>1</sup> Table flatlay photo created by freepik [www.freepik.com](http://www.freepik.com)

# Two popular SQL flavors

## PostgreSQL

- Free and open-source relational database system
- Created at the University of California, Berkeley
- "PostgreSQL" refers to both the PostgreSQL database system and its associated SQL flavor

## SQL Server

- Has free and paid versions
- Created by Microsoft
- T-SQL is Microsoft's SQL flavor, used with SQL Server databases

# Comparing PostgreSQL and SQL Server

- Like dialects of the same language
- Example: limiting number of results

PostgreSQL:

```
SELECT id, name  
FROM employees  
LIMIT 2;
```

id	name
54378	Darius
94722	Raven

SQL Server:

```
SELECT id, name  
FROM employees  
TOP 2;
```

id	name
54378	Darius
94722	Raven

# Choosing a flavor

Just like with ice cream, any flavor is probably a good choice!



# Let's practice!

INTRODUCTION TO SQL

# Congratulations!

INTRODUCTION TO SQL

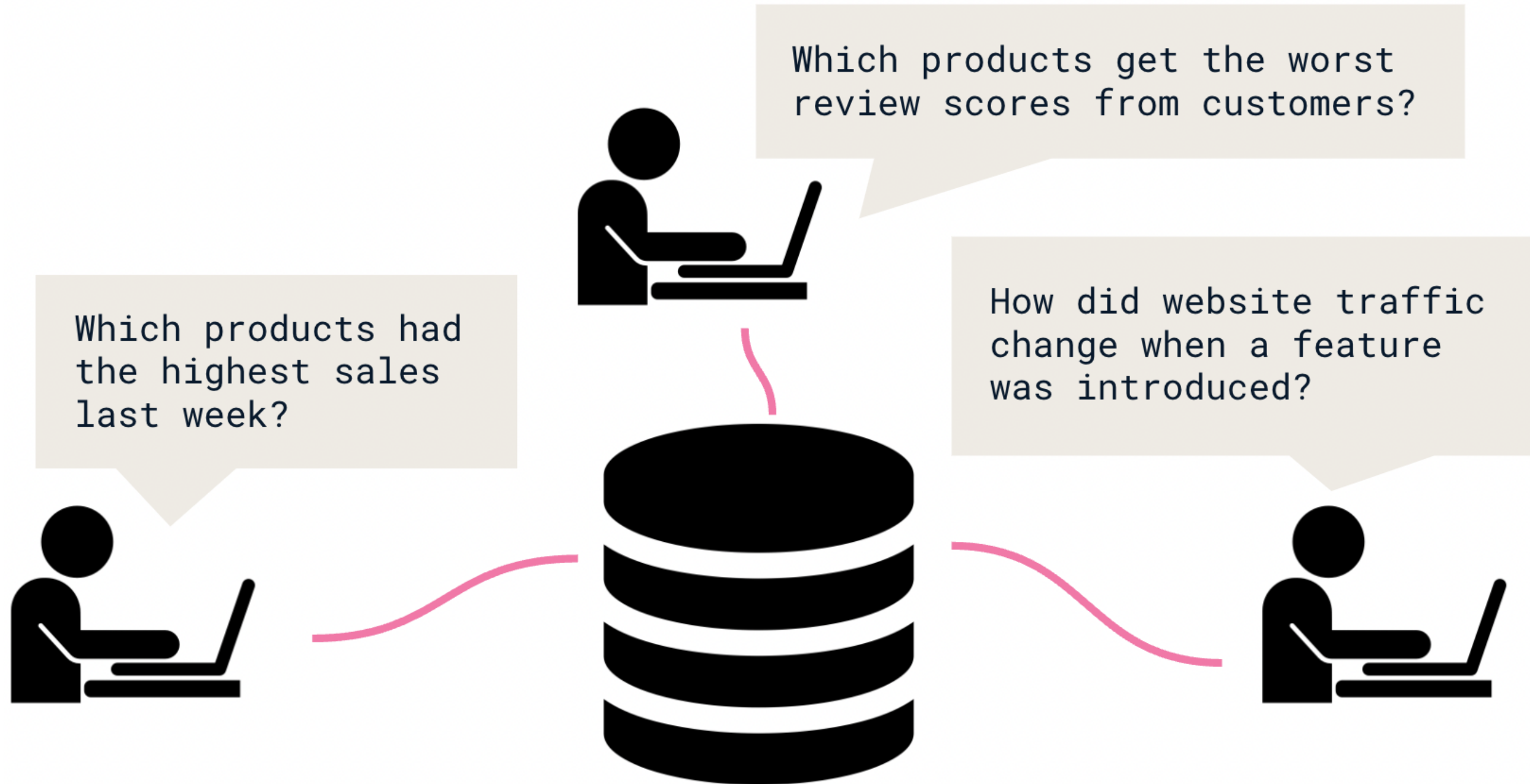


**Izzy Weber**

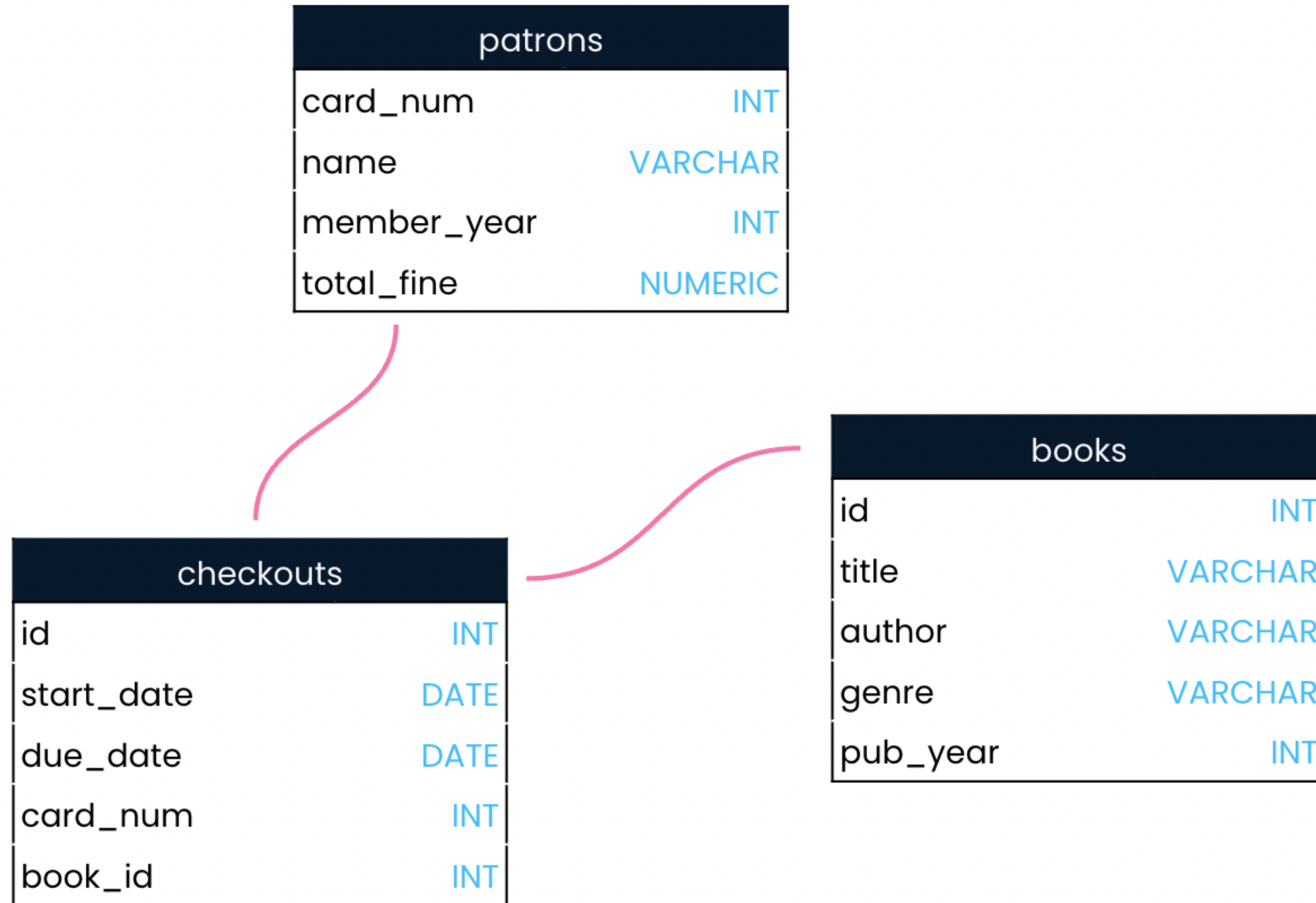
Curriculum Manager, DataCamp



# What you've learned

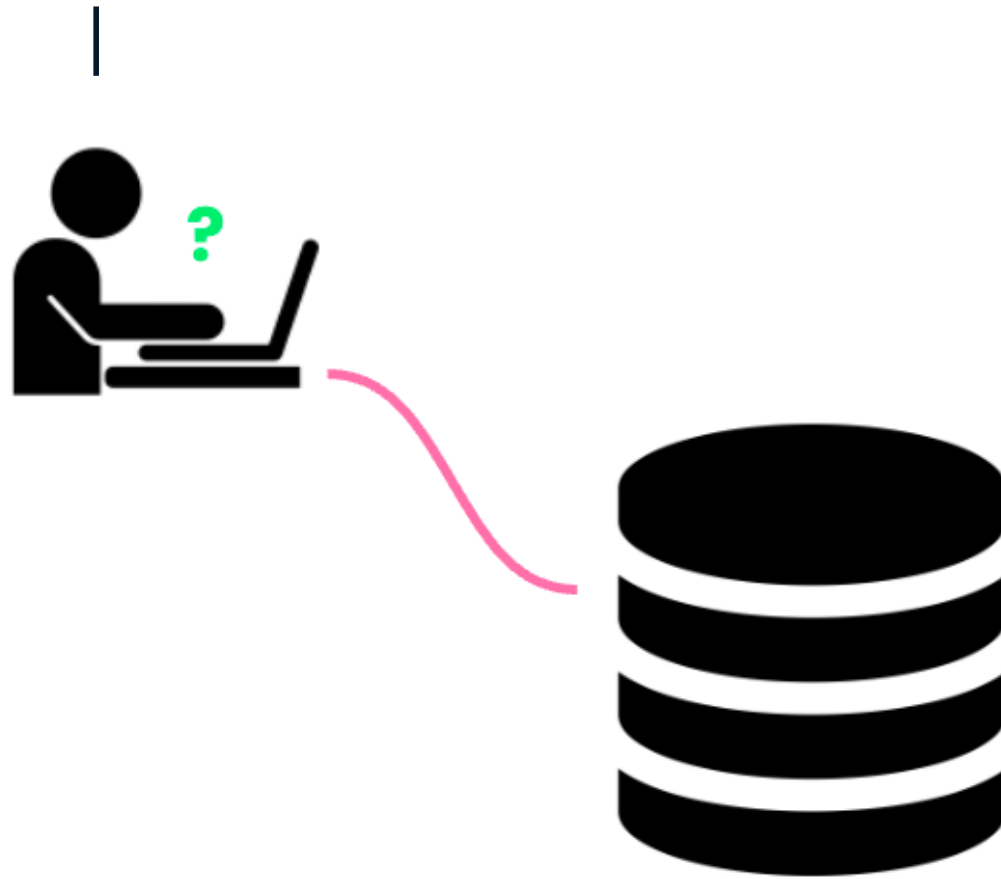


# What you've learned



# What you've learned

```
SELECT DISTINCT genre AS unique_genre  
FROM books  
LIMIT 15;
```



# Where to go next

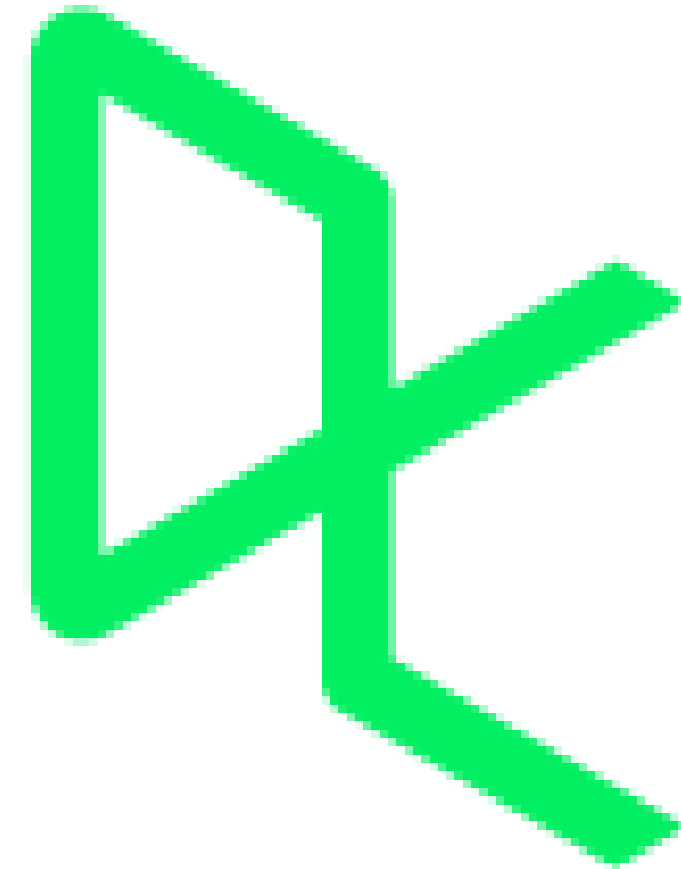
The next step is to learn more keywords and to choose which flavor you'll learn them in!

Learn PostgreSQL on DataCamp:

- Intermediate SQL Queries

Learn SQL Server on DataCamp:

- Introduction to SQL Server



**Thank you!**  
INTRODUCTION TO SQL