NODE.JS PT.2

CREATING AND INTEGRATING DATABASES IN YOUR WEBAPP

OUTLINE

- Databases
- MongoDB
- PSQL

DATABASES

DATABASES

DEFINITION

A database is an organized collection of structured information, or data, typically stored electronically in a computer system.

"image: Freepik.com".

DIFFERENT WAYS OF STORING DATA

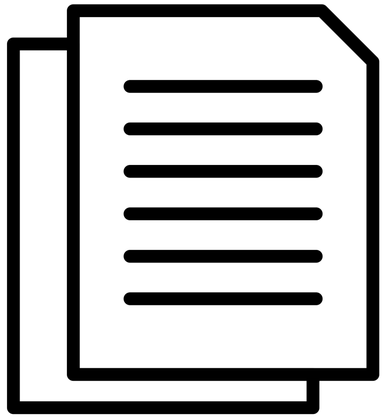
1. **CCPROG1:** Data is stored in memory

2. **CCPROG2 & CCPROG3:** Data is stored in a file (text file or binary file)

3. **CCINFOM ONWARDS:** Data is stored in a database or multiple databases

DATABASES

COMMON TYPES



NOSQL

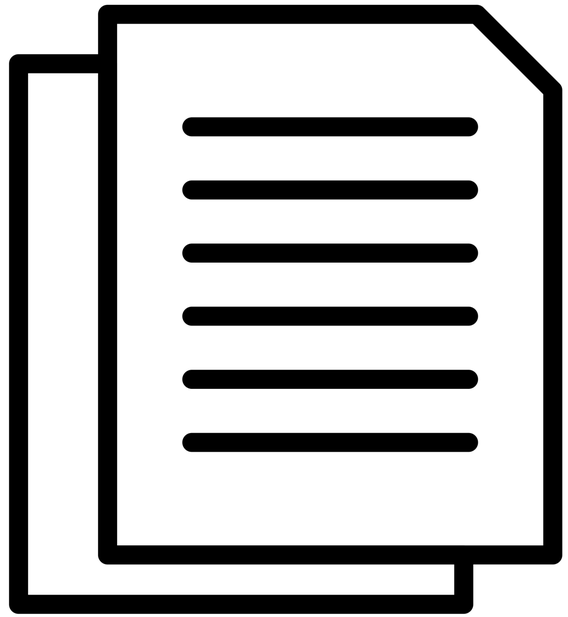
SQL

"image: Vecteezy.com and Flaticon.com".

NOSQL

DEFINITION

- **NoSQL** databases (document-based db) are **non-tabular databases** and store data differently than relational tables.
- These databases store semi-structured data and descriptions of that data in **document format**
- **Example:** MongoDB



"image: https://icons8.com/".

Document 1

```
{
  "id": "1",
  “name": "John Smith",
  "isActive": true,
  "dob": "1964-30-08"
}
```

Document 2

```
{
  "id": "2",
  “fullName": "Sarah Jones",
  “isActive": false,
  "dob": "2002-02-18"
}
```

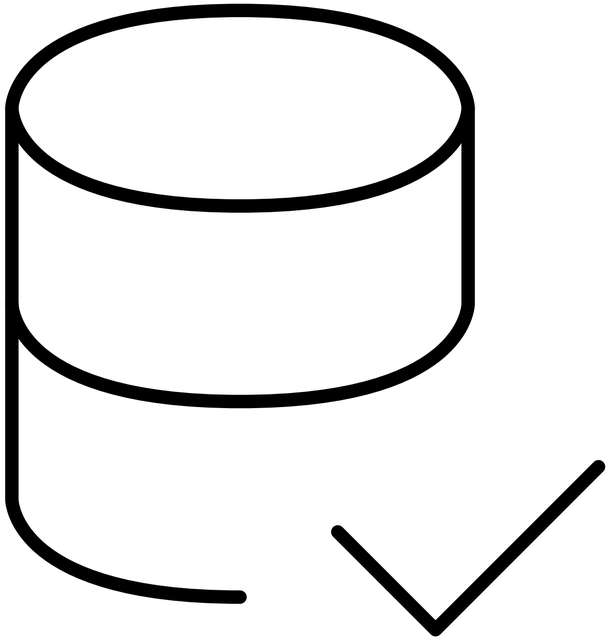
Document 3

```
{
  "id": "3",
  “fullName":
  {
    "first": "Adam",
    "last": "Stark"
  },
  "isActive": true,
  "dob": "2015-04-19"
}
```

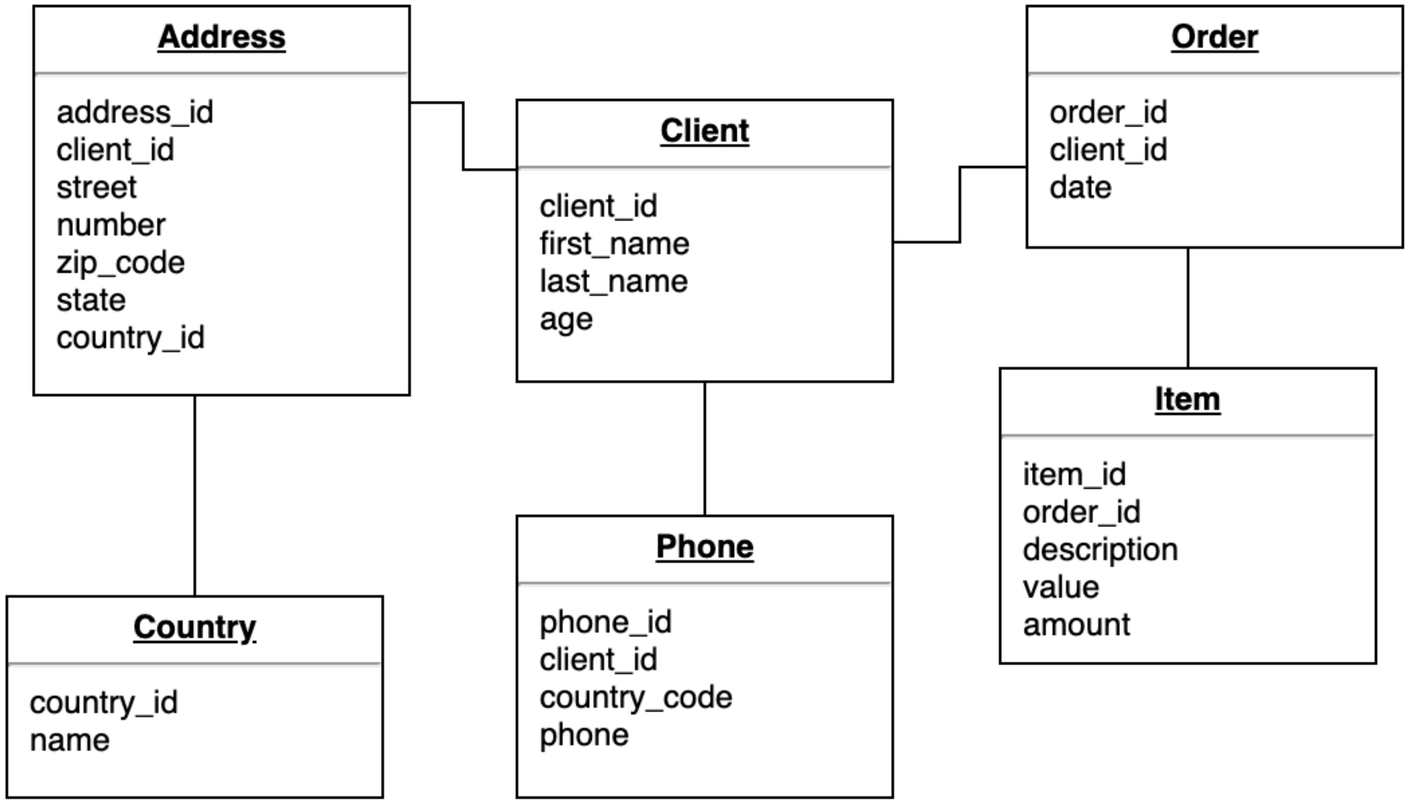
SQL

DEFINITION

- SQL stands for **Structured Query Language.** It's used for **relational databases.** A SQL database is a collection of tables that stores a specific set of structured data.
- A relational database is a type of database that stores and provides access to data points that are related to one another.
- **Example:** MySQL, Postgres



"image: https://icons8.com/".


Address
address_id
client_id
street
number
zip_code
state
country_id
Client
client_id
first_name
last_name
age
Order
order_id
client_id
date
Item
item_id
order_id
description
value
amount
Country
country_id
name
Phone
phone_id
client_id
country_code
phone

MONGODB

MONGODB

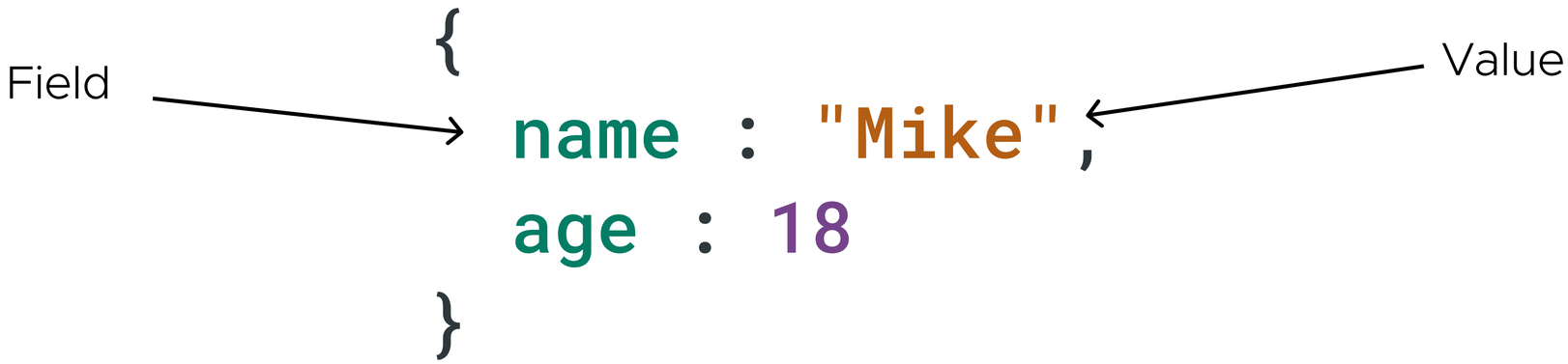
DEFINITION

- “MongoDB is a general purpose, **document-based, distributed database** built for modern application developers and for the cloud era.”
- MongoDB is a document database wherein one collection (table in RDBMS) holds different documents (tuple in RDBMS).
- Number of fields, content, and size of the document can differ from one document to another.

MONGODB

TYPICAL MONGODB DOCUMENT

A typical example of a MongoDB document which primarily consists of field and value pairs



- Documents are like JSON objects.
- The values of fields may include other documents, arrays, and arrays of documents.

MONGODB

DEMO

Using MongoDB Compass

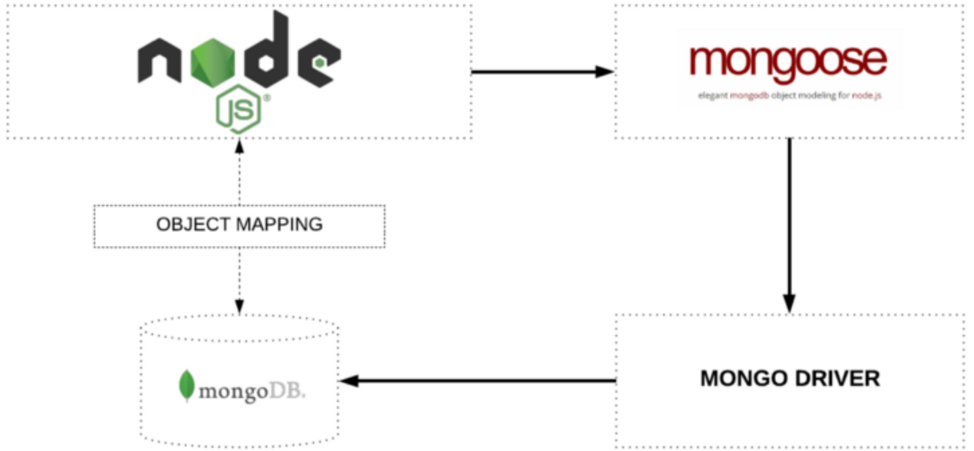
- Creating a database
- Creating a collection
- Inserting documents
- Finding documents
- Updating documents
- Deleting documents
- Droping a collection

MONGODB

NODE.JS INTEGRATION

Mongoose

- Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js.
- It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB.



MONGODB

NODE.JS INTEGRATION

Mongoose Demo

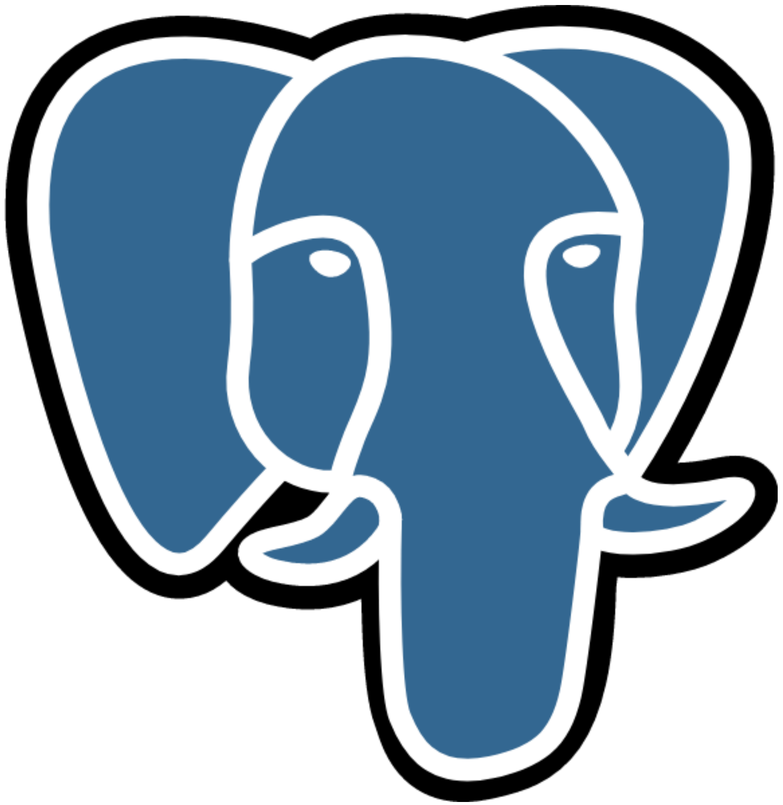
- Connecting to a database
- Defining a Schema
- Exporting a Model
- CRUD Operations

POSTGRE

POSTGRE

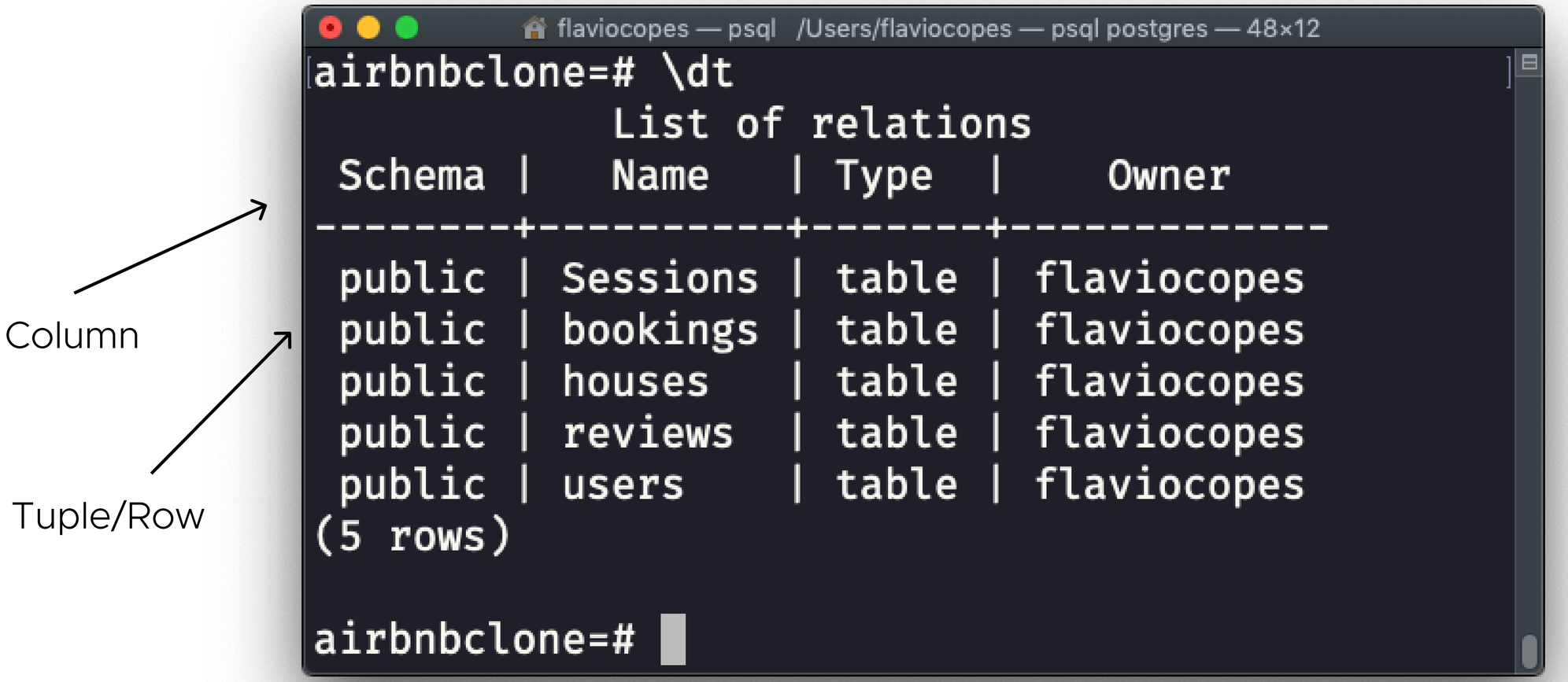
DEFINITION

- PostgreSQL is a powerful, open source **object-relational database system** with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.
- Free, open source, and highly extensible



POSTGRE

TYPICAL POSTGRE TABLE



POSTGRE

DEMO

Using pgAdmin

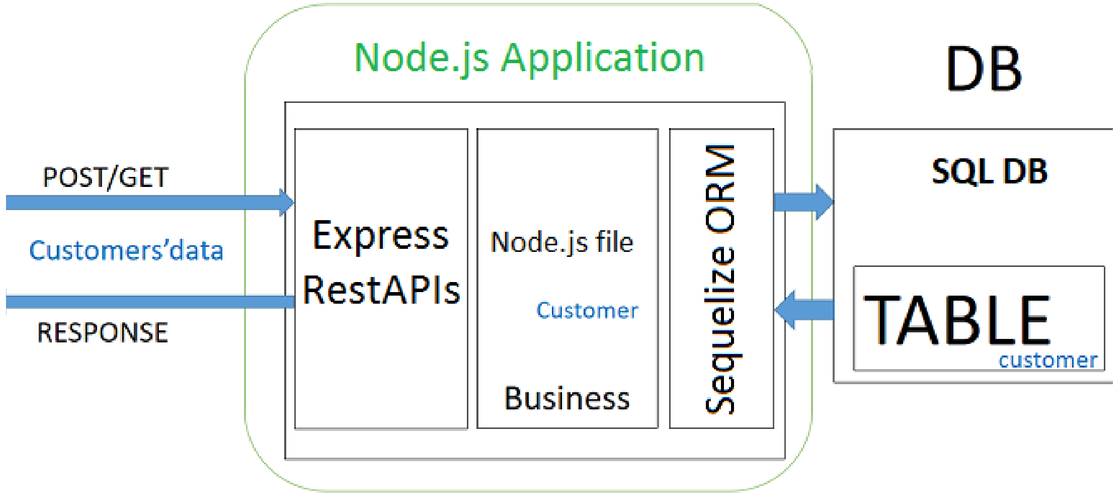
- Creating a database
- Creating a table
- Inserting records
- Finding records
- Updating records
- Deleting records
- Droping a table

POSTGRE

NODE.JS INTEGRATION

Sequelize

- Sequelize is a **promise-based Node.js ORM tool** for Postgres, MySQL, MariaDB, SQLite, Microsoft SQL Server, Amazon Redshift and Snowflake's Data Cloud. It features solid transaction support, relations, eager and lazy loading, read replication and more.



POSTGRE

NODE.JS INTEGRATION

Sequelize Demo

- Connecting to a database
- Creating a Model
- Exporting a Model
- CRUD Operations

SUMMARY

SUMMARY

TERMINOLGY AND CONCEPTS

PostgreSQL	MongoDB
ACID Transactions	ACID Transactions
Table	Collection
Row	Document
Column	Field
Secondary Index	Secondary Index
JOINs, UNIONs	Embedded documents, $lookup & $graphLookup, $unionWith
Materialized Views	On-demand Materialized Views
GROUP_BY	Aggregation Pipeline

Source: https://www.mongodb.com/compare/mongodb-postgresql

SUMMARY

QUERY LANGUAGE MAP

SQL	MongoDB

`CREATE TABLE users (`
`user_id VARCHAR(20) NOT NULL,`
`age INTEGER NOT NULL,`
`status VARCHAR(10));`
	Not Required

`INSERT INTO users(user_id, age, status)`
`VALUES ('bcd001', 45,"A");`
	`db.users.insert({`
	`user_id: "bcd001",`
	`age: 45,`
	`status: "A"`
	`})`

`SELECT *` `FROM users;`	`db.users.find()`

Source: https://www.mongodb.com/compare/mongodb-postgresql

SUMMARY

QUERY LANGUAGE MAP

```
UPDATE users
SET status = 'C'
WHERE age &gt; 25;
```

```
db.users.update(
    { age: { $gt: 25 } },
    { $set: { status: "C" } },
    { multi: true }
)
```

```
START TRANSACTION;
INSERT INTO orders
(order_id, product, quantity)
VALUES ('1a2b3c', 'T-shirt', '7');
UPDATE stock
SET quantity=quantity-7
WHERE product='T-shirt';
COMMIT;
```

```
session.startTransaction();
db.orders.insert ({
  order_id: '1a2b3c',
  product: 'T-shirt',
  quantity: 7
})
db.stock.update (
  { product: { $eq: 'T-shirt', } },
  { $inc: { quantity: -7 } }
})
session.commitTransaction();
```

Source: https://www.mongodb.com/compare/mongodb-postgresql

SUMMARY

WHEN TO USE NOSQL OR SQL

- **Depends**
- **NoSQL**
  - For fast development
  - Structure not needed
  - Data is scaling horizontally
- **SQL**
  - Data is structured
  - Involves transactions
  - Operations
- **Further Reading**
  - https://www.mongodb.com/compare/mongodb-postgresql

THANK YOU

QUESTIONS?