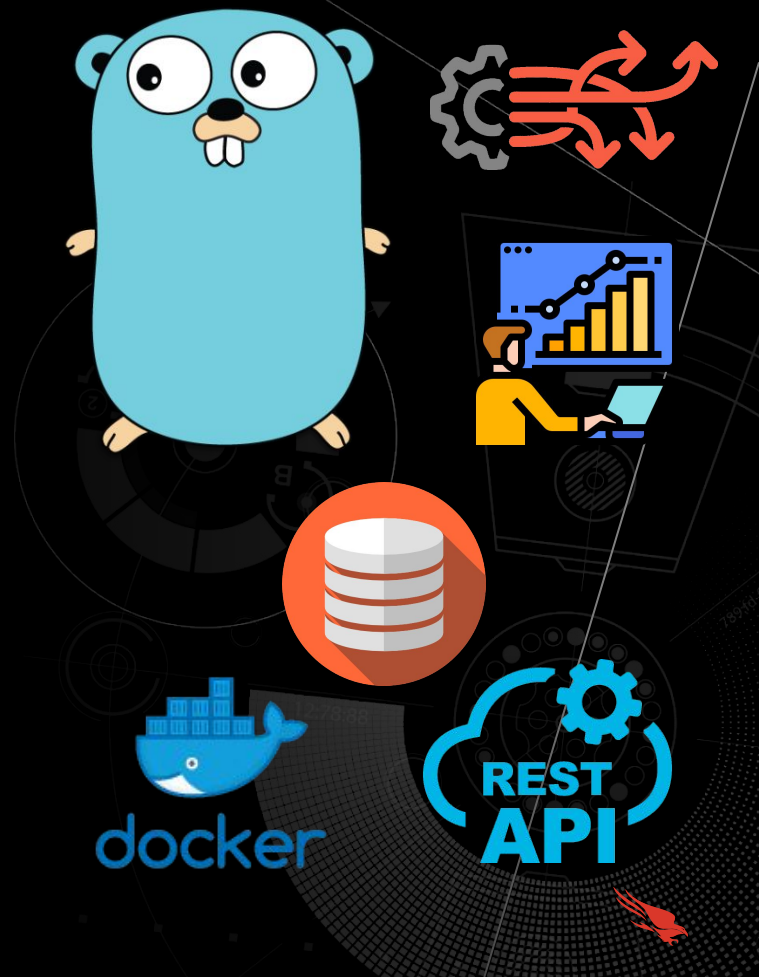
Databases

CrowdStrike HEROES - Cloud Workshop

CROWDSTRIKE

Date	Topic
July 25 - 16:00	Intro to golang
July 26 - 16:00	Intro to golang (continuation)
July 27 - 16:00	Multithreading
July 28 - 16:00	Rest API
July 29 - 16:00	Unit testing, logging and monitoring
August 1 - 16:00	Workshop and Q&A
August 2 - 16:00	Deployments/Docker
August 3 - 16:00	**Databases**
August 4 - 16:00	Databases extended
August 5 - 13:00	Microservices contest (4h with Awards)



CROWDSTRIKE

What we're talking about here today

Intro

The why and the how

Environment setup

Docker it up, go it up, and Postgres the hell out of it

CRUD!

What a database is, and how the basic operations look like

Join the club

Constraints, joins, migrations

Stretching the cord

Bulk inserts at scale, and how to optimize for large data volumes

ACID

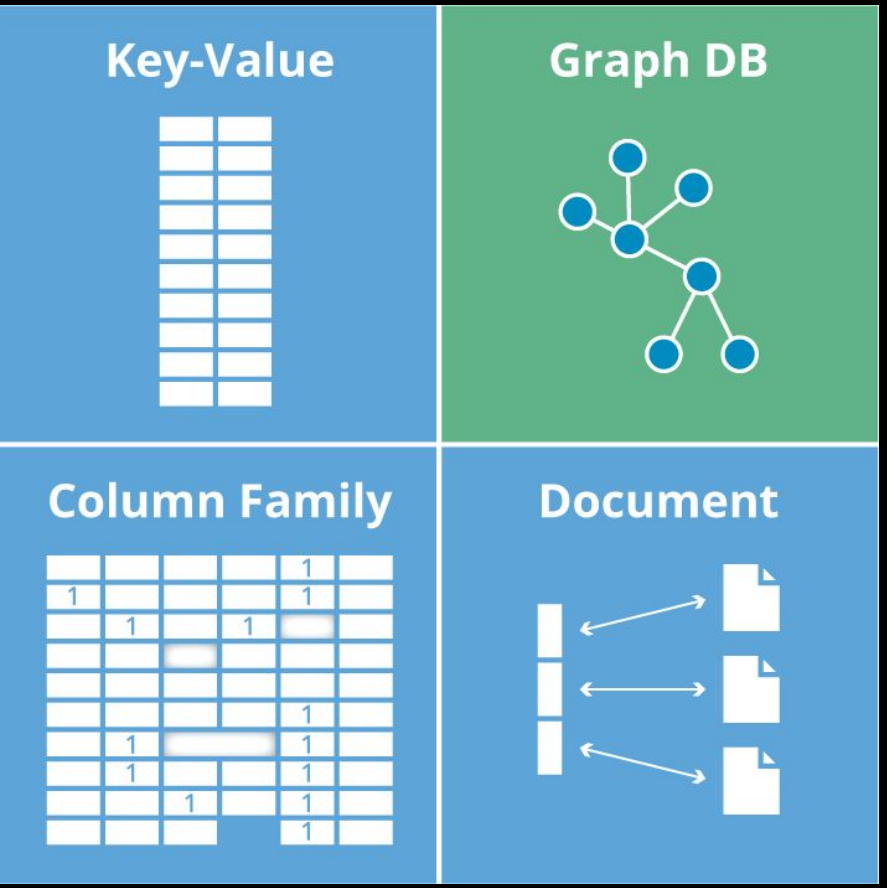
Because this is a fun workshop

A database is where you keep your data

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

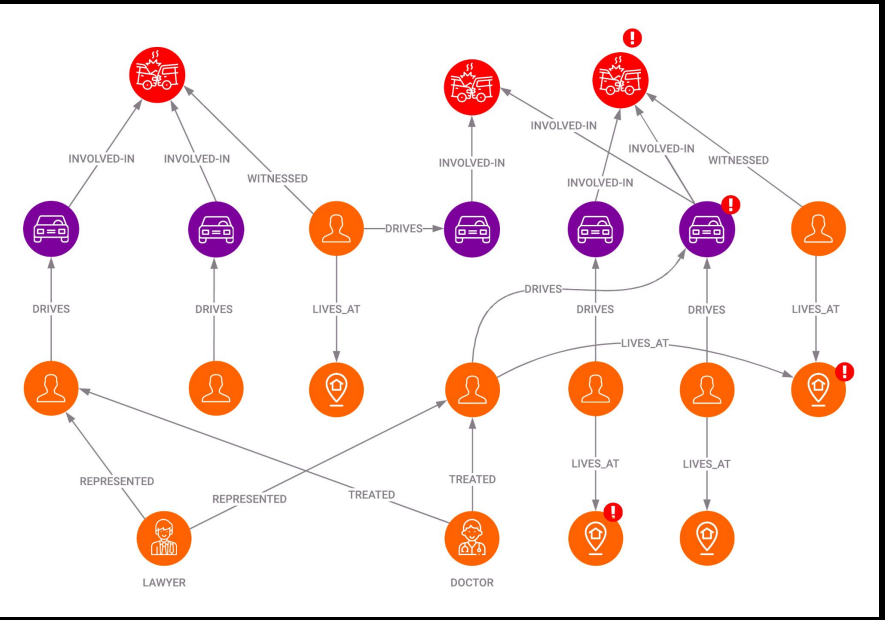
Relational databases

Relational databases became dominant in the 1980s. Items in a relational database are organized as a set of tables with columns and rows.



NoSQL databases

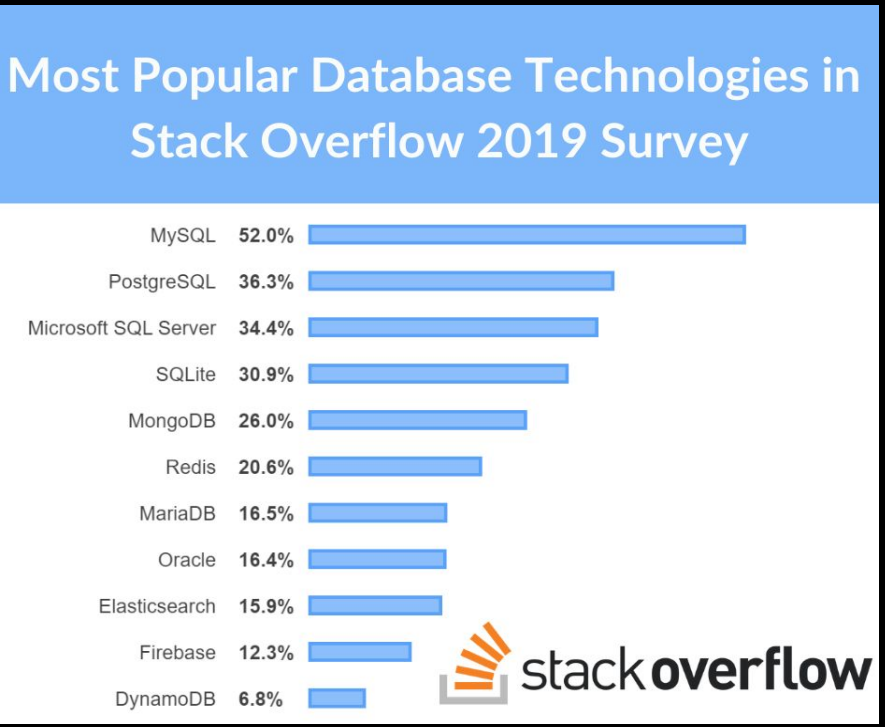
A NoSQL, or nonrelational database, allows unstructured and semistructured data to be stored and manipulated



Graph databases

A graph database stores data in terms of entities and the relationships between entities.

CROWDSTRIKE



Intro

The one and only: PostgreSQL

- Free, open source, relational database
- Older than you: Initial release was on July 8th 1996
- Very popular

CROWDSTRIKE

Environment setup

Docker, Golang, Postgres

CROWDSTRIKE

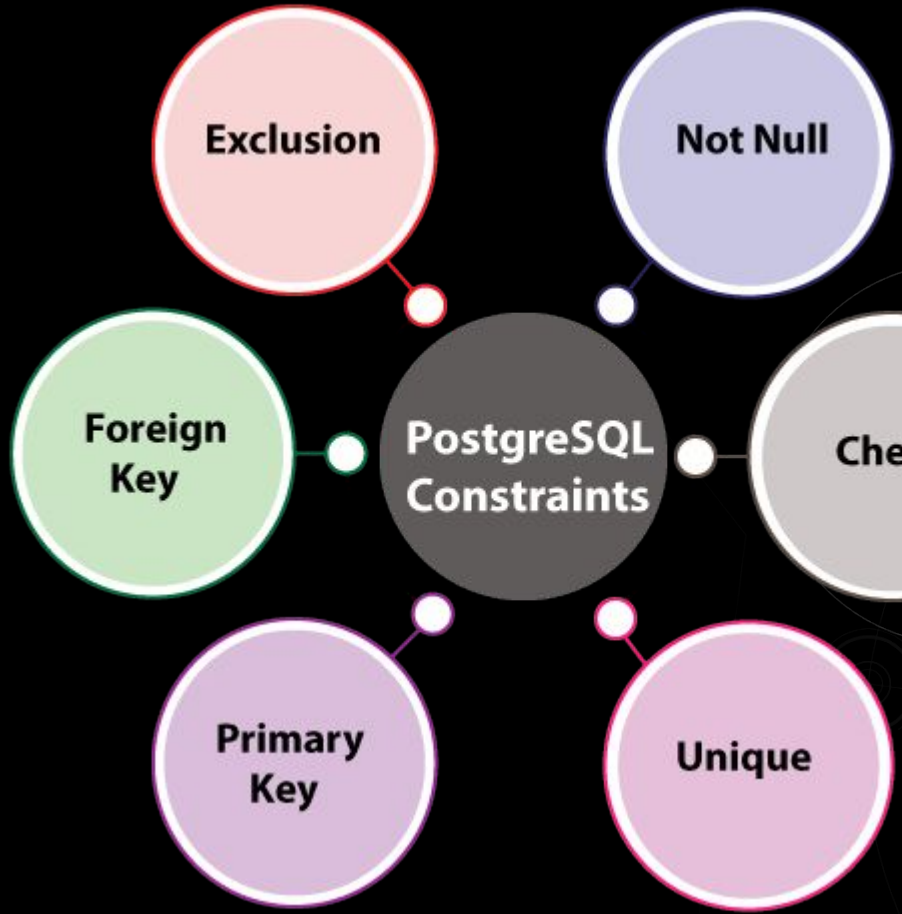
CRUD!

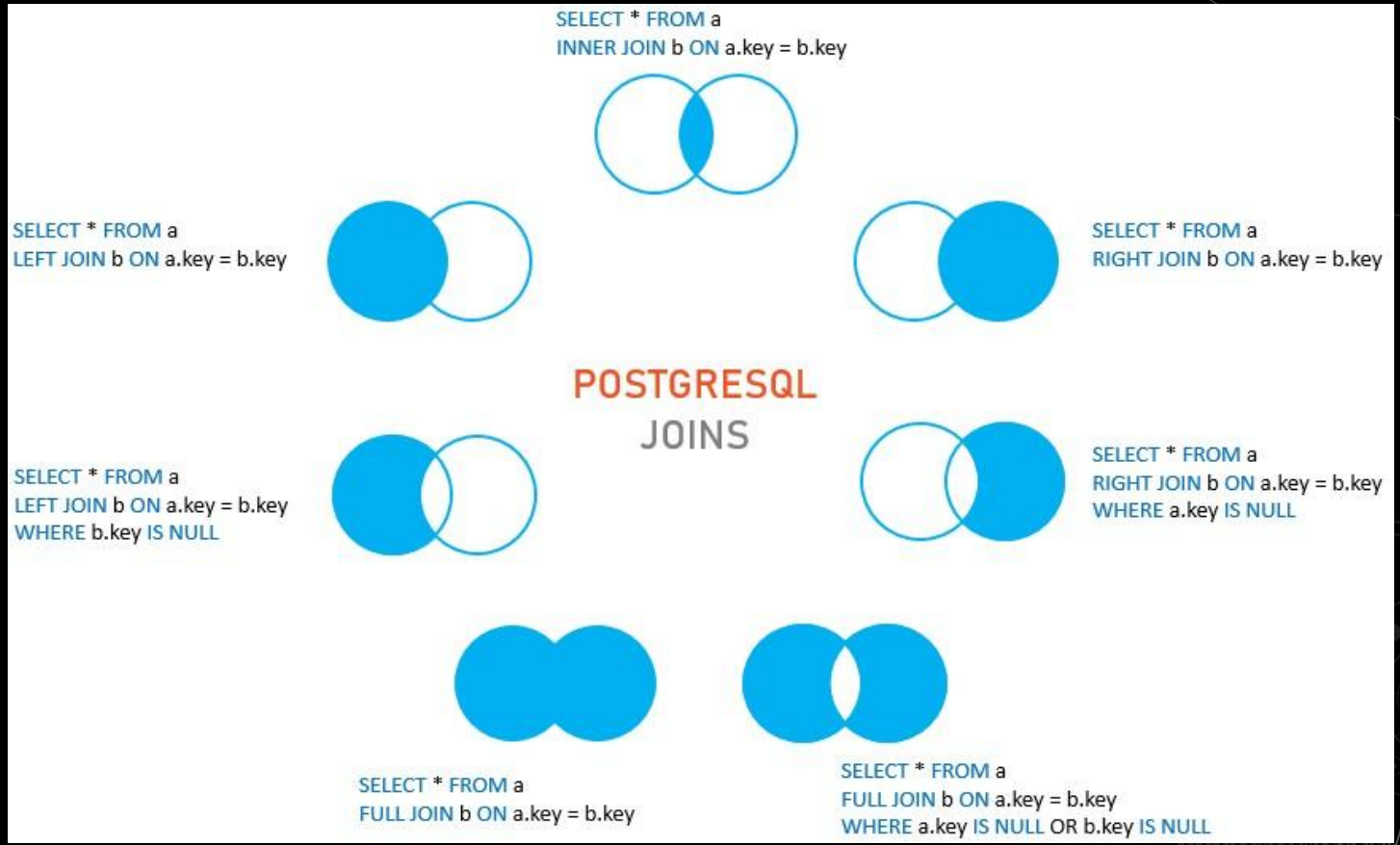
Create, Read, Update, Delete operations

CROWDSTRIKE

Join the club

Constraints, joins, and migrations


Exclusion
Not Null
Foreign Key
PostgreSQL Constraints
Check
Primary Key
Unique


SELECT * FROM a
INNER JOIN b ON a.key = b.key
SELECT * FROM a
LEFT JOIN b ON a.key = b.key
SELECT * FROM a
RIGHT JOIN b ON a.key = b.key
POSTGRESQL
JOINS
SELECT * FROM a
LEFT JOIN b ON a.key = b.key
WHERE b.key IS NULL
SELECT * FROM a
RIGHT JOIN b ON a.key = b.key
WHERE a.key IS NULL
SELECT * FROM a
FULL JOIN b ON a.key = b.key
SELECT * FROM a
FULL JOIN b ON a.key = b.key
WHERE a.key IS NULL OR b.key IS NULL

CROWDSTRIKE

Stretching the cord

Bulk inserts, Indexes, B-tree

CROWDSTRIKE

ACID

Atomicity, Consistency, Isolation, Durability


Atomicity
Commits finish an entire operation successfully or the database rolls back to its prior state
Consistency
Any change maintains data integrity or is cancelled completely
Isolation
Any read or write will not be impacted by other reads or writes of separate transactions
Durability
Successful commits will survive permanently

CROWDSTRIKE

What we talked about today

Intro

The why and the how

Environment setup

Docker it up, go it up, and Postgres the hell out of it

CRUD!

What a database is, and how the basic operations look like

Join the club

Constraints, joins, migrations

Stretching the cord

Bulk inserts at scale, and how to optimize for large data volumes

ACID

Because this is a fun workshop

CROWDSTRIKE

Any questions?