

Introduction to key-value databases

INTRODUCTION TO NOSQL

SQL


Jake Roach
Data Engineer

Key-value databases

Definition: A NoSQL database that uses simple key-value pairs (similar to dictionaries in Python) to store data.

- Can be searched by key, not the value
- Store values of type string, hash, or lists
- Performant for simple read and writes of data
- Data is stored in memory, instead of on disk

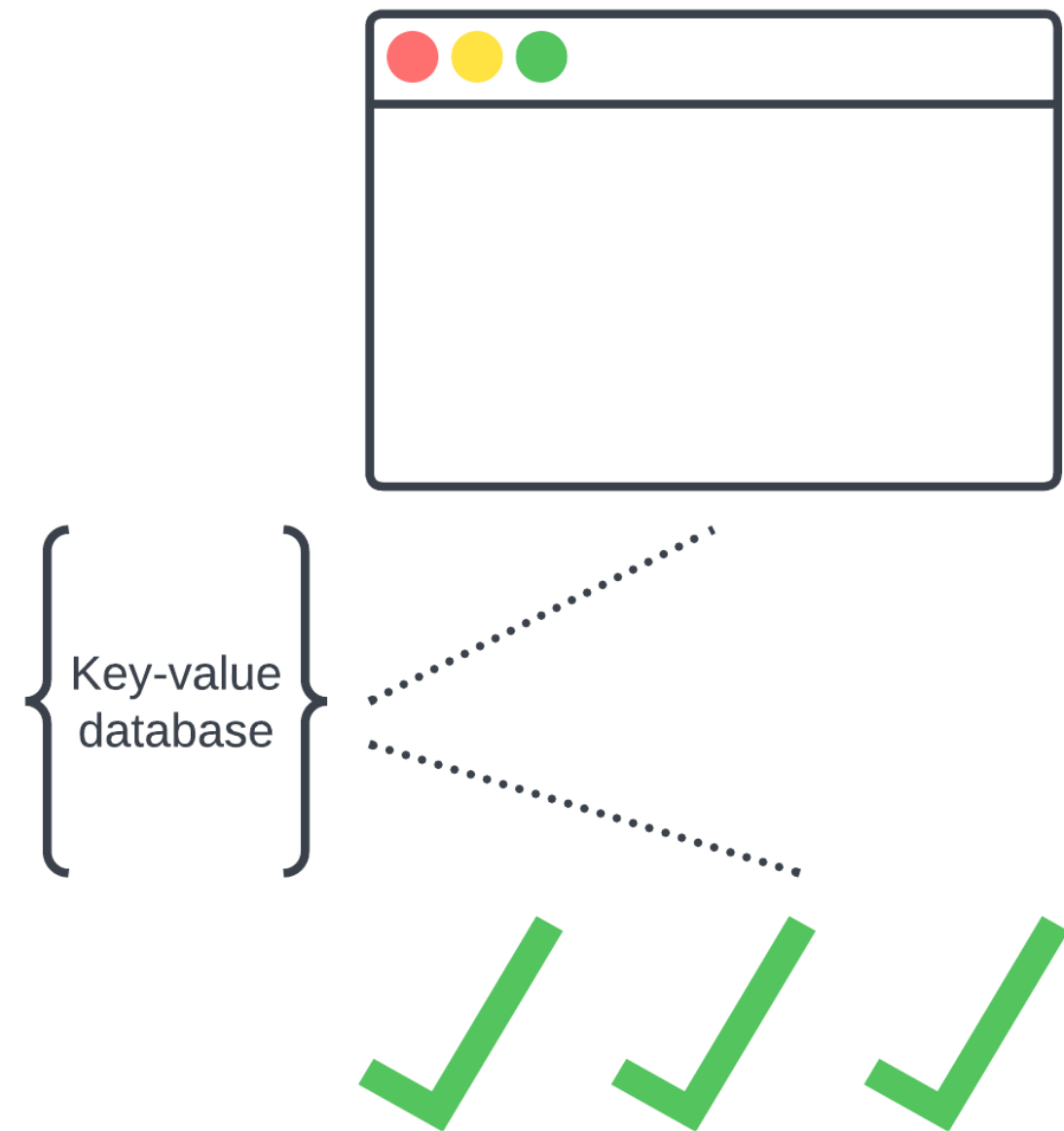
`"weather" : "sunny"`



retrieve value stored at
"weather"

¹ <https://redis.com/nosql/key-value-databases/>

Key-value database use-cases



Commonly used in web applications, for:

- Session management
- Caching frequently accessed data
- Track user preferences and behavior

Key-value databases source analytics platforms:

- Snowflake
- Redshift

Redis



`redis` library to interact with Redis using Python

- Connect to cluster running on `localhost`
- Exposed over port `6379`
- Decode responses, instead of bytes

```
import redis

# Create a connection to Redis cluster
r = redis.Redis(
    host="localhost",
    port=6379,
    decode_responses=True
)
```

Later, we'll:

- Store key-value pairs
- Retrieve key-value pairs

Let's practice!

INTRODUCTION TO NOSQL

Storing and retrieving key-value data

INTRODUCTION TO NOSQL

SQL

Jake Roach
Data Engineer

Storing key-value data

```
# Import redis, make a connection  
r = redis.Redis(...)
```

```
# Store a key-value pair  
r.set("username", "JDoe")
```

```
# Store another key-value pair  
r.set("age", 27)
```

```
# Overwrite an existing key  
r.set("username", "BSmith")
```

After making a connection to the Redis server:

- Pass a key and value to the `.set()` method
- Can pass native `int` or `float` types, will be stored as `strings`
- Can overwrite existing key-value pairs

Retrieving key-value data

Retrieve a key-value pair

```
# Store a key-value pair
r.set("username", "JDoe")

# Retrieve the key-value pair
username = r.get("username")

# Print the result
print(username)
```

JDoe

Overwrite a key-value pair

```
r.set("username", "BSmith")
username = r.get("username")
print(username)
```

BSmith

Attempt to access a key that does not exist

```
favorite_color = r.get("favorite_color")
print(favorite_color)
```

None

Storing complex key-value data

```
# Store a dictionary using .hset()
r.hset(
    "shopping_cart",
    mapping={
        "item_id": "1003",
        "quantity": 2,
        "price": 79.99
    }
)
```

```
# Retrieve the dictionary
r.hgetall("shopping_cart")
```

Can store more complex data types, like `dict` ionaries:

- `.hset()` , takes key and a `dict`
- Pass a key to `.hgetall()`

```
{
    'item_id': '1003',
    'quantity': '2',
    'price': '79.99'
}
```

Let's practice!

INTRODUCTION TO NOSQL

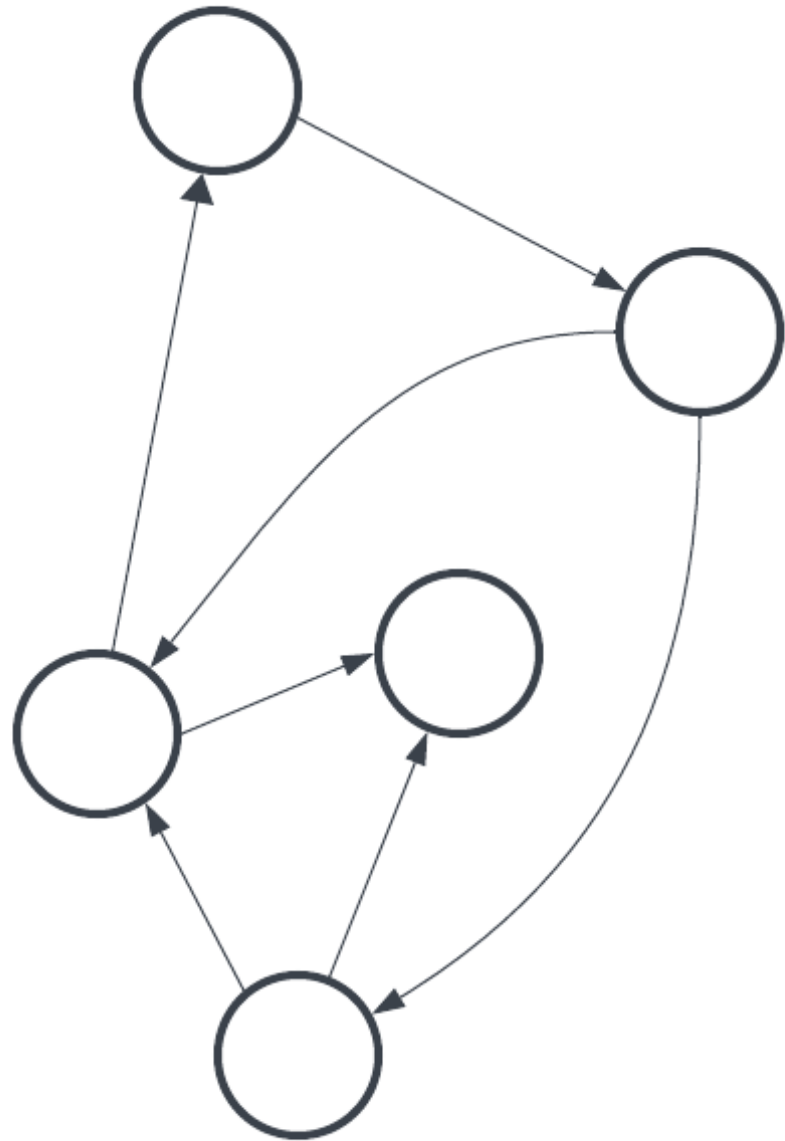
Graph databases

INTRODUCTION TO NOSQL



Jake Roach
Data Engineer

What are graph databases?

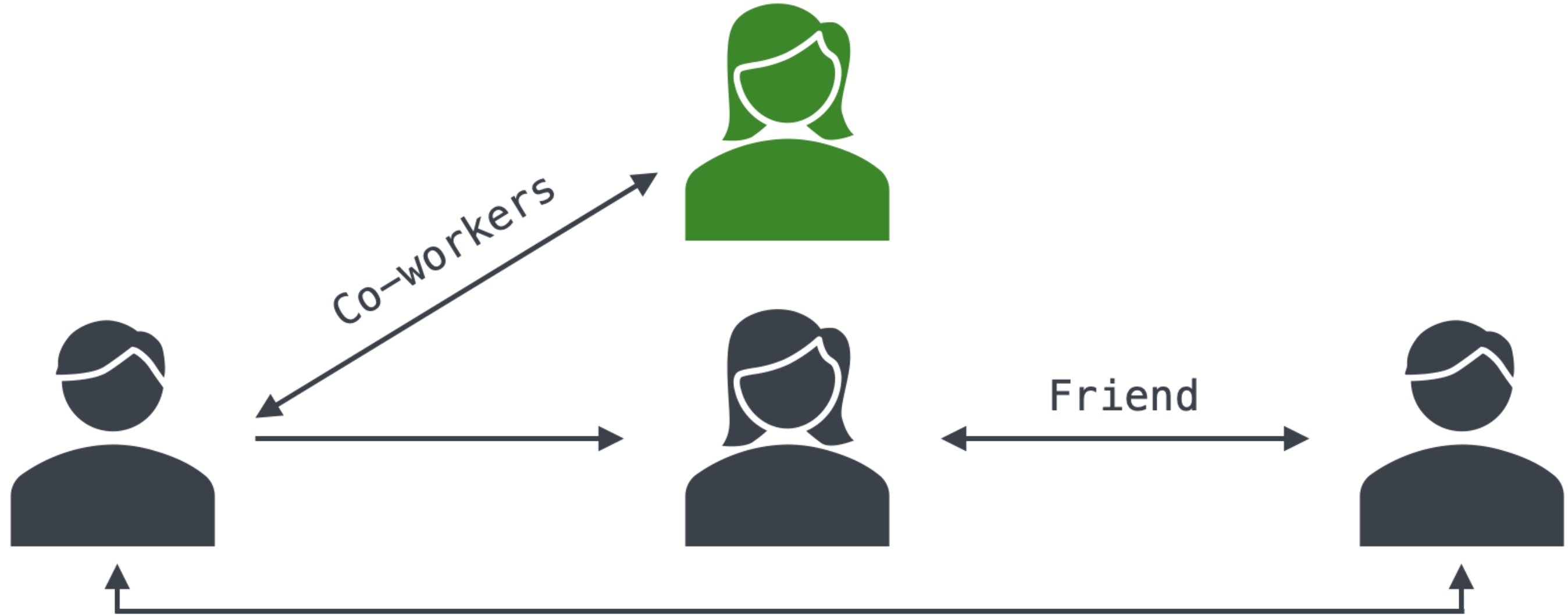


Definition: NoSQL data stores that persist data in a network of nodes and edges. In graph databases, each node represents an entity, and each edge represents a relationship between those entities.

- Graph query language
- Graph algorithms
- Performant and efficient

¹ <https://aws.amazon.com/nosql/graph/>

A closer look at graph databases



What are graph databases used for?

Graph databases are commonly used for:

- Social networks
- Recommendation engines
- Fraud detection
- Studying patterns and relationships



Graph database providers



Let's practice!

INTRODUCTION TO NOSQL

Wrapping up!

INTRODUCTION TO NOSQL



Jake Roach
Data Engineer

NoSQL data stores

NoSQL:

- How is it different from RDBMS?
- Basic characteristics

Document databases:

- Postgres JSON
- Semi-structured data
- Prepare data for analytics

Columnar databases:

- Snowflake
- Large, structured and semi-structured data

Key-value and graph databases:

- Redis
- Simple data, stored in key-value pairs
- Graph databases

What next?

- Continue to practice with real-world examples
- Build projects using DataCamp Workspace
- Document takeaways



Thank you!
INTRODUCTION TO NOSQL