# Introduction to NoSQL

Using Databases for their purpose

# Table of Contents

**What are Databases?**
Accessing Information
**SQL vs NoSQL**
*Popular Databases*
Visual Representation
Types of Scaling
Varieties of NoSQL Databases
**Use Cases**
Demos

# What are Databases?

A **Database** is a collection of information that is organized so that it can be easily accessed, managed and updated.

This information is usually accessed by **queries** and the **structure** is defined by the developer

Nearly all the applications in the world require a database and the database of choice varies by use case.

**There is no jack of all trades database. Each database technology has it’s use case.**

Databases typically come in two different flavours, **SQL and NoSQL**

# Accessing Information

Data can be accessed, mutated and inserted from or into a database using a **query**.

Before we can do that, we must define a **structure** of how the data can be represented.

**Scalability** is the capability of a system, network, or process to handle a growing amount of work, or its potential to be enlarged to accommodate that growth.

For this example, I will be using MongoDB.

```
// Structure
{
    item:  String,
    qty: String,
    tags: [],
    size: {h: Number, w: Number, uom: String}
}
```

# Accessing Information

```
// Inventory Structure
{
    item:  String,
    qty: String,
    tags: [],
    size: {h: Number, w: Number, uom: String}
}
```

```
db.collection('inventory').insertOne(
    {
        item: 'paper',
        qty: 25,
        tags: ['very cool', 'red'],
        size: { h: 14, w: 21, uom: 'cm' }
    },
);
```

```
db.collection('inventory').updateOne(
    { item: 'paper' },
    {
        $set: { 'size.uom': 'cm', qty: 69 },
        $currentDate: { lastModified: true }
    }
);
```

```
db.collection('inventory').find({ qty: 25 });
```

# Popular Databases







**SQL**






**NoSQL**

# What is SQL?

SQL or Structured Query Language is a **relational database** where it's data structures are **fixed and predefined**.

**Made in the 70s and is proven to be one of the most reliable database system in the world.**

**Usually scales vertically.** Generally faster for complex queries.

Great for: Data Integrity and Complex Queries

| address_id | street | city | state | zip_code |
|---|---|---|---|---|
| 1 | 807 Grandrose Ave. | Yonkers | NY | 10701 |
| 2 | 26 Market Drive | Forest Hills | NY | 11375 |
| 3 | 60 Myers Dr. | Amityville | NY | 11701 |
| 4 | 9782 Indian Spring Lane | Harlingen | TX | 78552 |
| 5 | 167 James St. | Los Banos | CA | 93635 |
| 6 | 755 East Henry Lane | Central Islip | NY | 11722 |
| 7 | 8165 Baker Ave. | Franklin Square | NY | 11010 |
| 8 | 669 S. Gartner Street | San Pablo | CA | 94806 |
| 9 | 683 West Kirkland Dr. | East Northport | NY | 11731 |
| 10 | 684 Westport Drive | Ballston Spa | NY | 12020 |
| 11 | 720 Thompson Lane | Rego Park | NY | 11374 |
| 12 | 973 Yukon Avenue | Encino | CA | 91316 |
| 13 | 55 Cambridge Street | Plainview | NY | 11803 |

# What is NoSQL?

NOT SQL

Thanks for Coming

# What is NoSQL?

NoSQL is what it sounds like.

NoSQL databases are **non-relational** and is typically called a distributed database. It’s schema is **dynamic.**

**Usually scales horizontally.** Generally faster for small queries.

Great for: Large Unstructured Datasets (Big Data), Flexibility, Real-time systems, and Rapid development

# What they look like

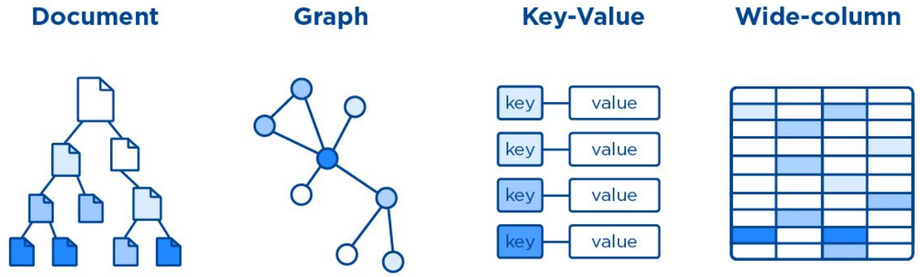
Usually just a table or tables

F5 O

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Rating | Name | Assets | NAVPS | MER | LoadFees | Ret_1Mth | Ret_3Mth | Ret_1Yr | Ret_3Yr |
| 2 | 5 | ABC Fundamental Value | 535.9 | 18.61 | 2.00 | N | -3.4 | 2.6 | 15.2 | 12.8 |
| 3 | 1 | AIC Advantage | 1235.1 | 68.92 | 2.45 | O | -2.9 | -1.9 | 2.1 | -0.3 |
| 4 | 4 | AIC PPC Canadian Pool | 50.9 | 13.30 | 2.80 | O | -1.6 | 1.5 | 15.7 | |
| 5 | 4 | Assante Canadian Equity Div Po | 59.4 | 14.36 | 0.43 | O | -3.4 | 1.6 | 13.1 | 7.1 |
| 6 | 3 | Assumption/FDI True North-B | | 19.88 | 4.28 | N | -3.0 | 1.5 | 14.8 | 5.3 |
| 7 | 3 | BMO Special Equity | 399.8 | 22.88 | 2.52 | N | -3.1 | -1.8 | 11.3 | 8.7 |
| 8 | 1 | Canada Life Gens Cdn Eq (Temp) | 9.2 | 14.42 | 3.54 | R | -1.6 | 0.6 | 2.0 | 1.4 |
| 9 | 3 | CDA Common Stock (Altamira) | 42.8 | 48.31 | 0.97 | N | -1.7 | 2.3 | 10.7 | 6.1 |
| 10 | 2 | CIBC Canadian Small Companies | 104.6 | 19.59 | 2.62 | N | -5.4 | -0.5 | 8.8 | 4.0 |
| 11 | 3 | CI Signature Canadian GIF-B | 4.7 | 12.26 | 3.78 | O | -2.0 | 0.7 | 14.4 | 5.6 |
| 12 | 4 | Clarica SF Canadian Equity | 53.9 | 14.98 | 3.69 | N | -1.5 | 4.1 | 19.3 | 6.0 |
| 13 | 4 | Clarica Canadian Equity | 104.1 | 15.18 | 3.35 | N | -1.5 | 4.2 | 20.0 | 6.5 |
| 14 | 3 | Clarington Canadian Core Port | 32.7 | 13.40 | 2.71 | O | -1.5 | 1.9 | 8.8 | |
| 15 | | Co-operators CdnFstCl (AIM) VAII | 1.0 | 128.85 | 2.94 | D | -3.2 | 2.5 | 20.6 | |
| 16 | 3 | Concordia Equity | 15.0 | 32.79 | 2.43 | D | -0.8 | 2.6 | 12.8 | 5.6 |
| 17 | 4 | Desj Fn Pool Bissett Cdn Equ | 13.4 | 17.62 | 0.50 | N | -1.0 | 2.7 | 15.1 | 7.9 |
| 18 | | Disciplined Leadrshp Cdn Equ A | 66.7 | 11.22 | 2.47 | F | -4.3 | 1.6 | | |
| 19 | 3 | Empire Premier Equity | 248.1 | 214.01 | 1.56 | F | -0.9 | 2.2 | 13.0 | 6.2 |
| 20 | 4 | Ferique Equity | 263.7 | 54.73 | 0.75 | N | -1.7 | 3.4 | 12.7 | 8.1 |
| 21 | 3 | Fidelity Cdn Disciplined Equ-T | 8.8 | 24.87 | 2.42 | O | -3.3 | 1.8 | 12.0 | |
| 22 | | Franklin Temp Cdn Growth Port | 16.9 | 10.99 | 2.54 | O | -2.5 | 0.7 | | |
| 23 | 1 | GWL Canadian Opp. (M) DSC | 12.8 | 267.94 | 2.93 | D | -1.7 | -1.3 | -3.4 | -0.7 |
| 24 | | Hartford Growth & Income B | 22.3 | 12.16 | 2.60 | D | 0.0 | 4.0 | | |
| 25 | 2 | GWL Smaller Company (M) NL | 10.7 | 303.19 | 3.06 | N | 0.2 | -0.5 | 2.8 | 2.1 |
| 26 | 4 | IA Canadian Conservative Equ | 103.8 | 19.86 | 2.56 | O | -0.5 | 0.9 | 13.8 | 10.6 |
| 27 | 1 | Hathaway Focus Plus Canadian | 8.2 | 6.27 | 3.00 | O | -3.1 | -2.8 | 3.8 | |
| 28 | 3 | IAP Canadian Equity Growth | 4.1 | 13.17 | 2.74 | R | -2.3 | 1.7 | 10.6 | 6.6 |

**SQL**

What??




**NoSQL**

# Types of Scaling

We want to scale a database with this machine

**Buy better hardware on your machine (More RAM, CPU .. etc)**

**Vertical Scaling**

**Just buy more machines (Distributed System)**

**Horizontal Scaling**

# Common Misconceptions

NoSQL is NOT always faster and NOT always better to scale than SQL.

Turning on NoSQL doesn’t mean it automatically scales. SQL can scale too.

NoSQL is not always SAFE to write data or READ to (Not ACID Compliant).
Typically NoSQL stages your data before writing it into a database.

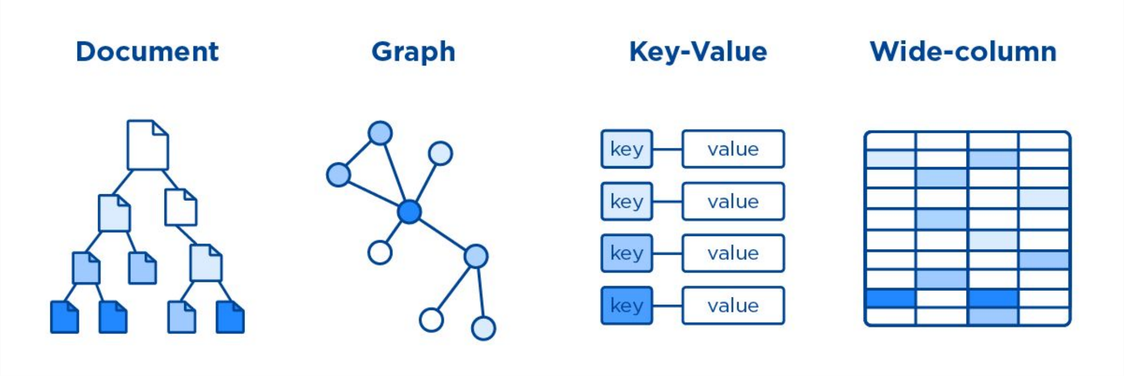
**When to use SQL**

- When you know your data is consistent or unchanging
- ACID compliance (Atomicity, Consistency, Isolation, Durability)
- i.e requiring data integrity, data has 100% accuracy and consistency of data stored in a database

**When to use NoSQL**

- When you have massive amounts of unstructured data
- Require data to be spread across multiple machines
- High Availability

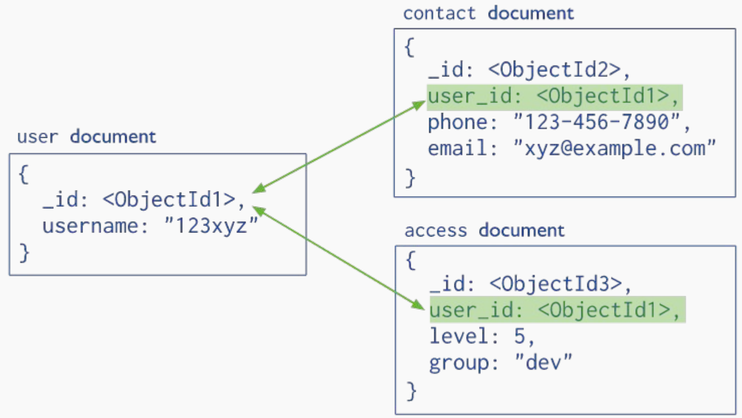
# Types of NoSQL Databases



# Examples of NoSQL Databases

**MongoDB** - Document Based

- Used to store JSON-Like documents
- Makes it easy to store objects across an application by using the same document model
- Flexible Models that allow you to store data of any structure
- Example: Content Management




**Redis** - Key-Value

- Can be used for efficient caching
- Counting e.g. Leaderboards

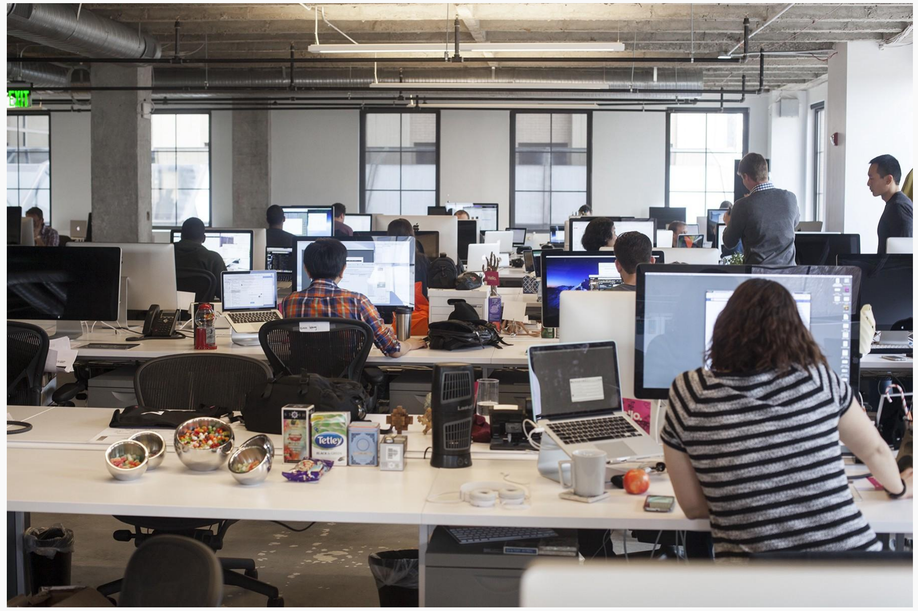
| Key | Value |
|---|---|
| K1 | AAA,BBB,CCC |
| K2 | AAA,BBB |
| K3 | AAA,DDD |
| K4 | AAA,2,01/01/2015 |
| K5 | 3,ZZZ,5623 |

# Installing MongoDB

**Installing MongoDB**
https://www.mongodb.com/download-center/community

**Viewing Data inside your MongoDB database:**
**https://robomongo.org/**

**Using a cloud solution: MongoDB Atlas (No need to install)**
https://www.mongodb.com/cloud/atlas



# Accessing Information

```
// Inventory Structure
{
    item:  String,
    qty: String,
    tags: [],
    size: {h: Number, w: Number, uom: String}
}
```

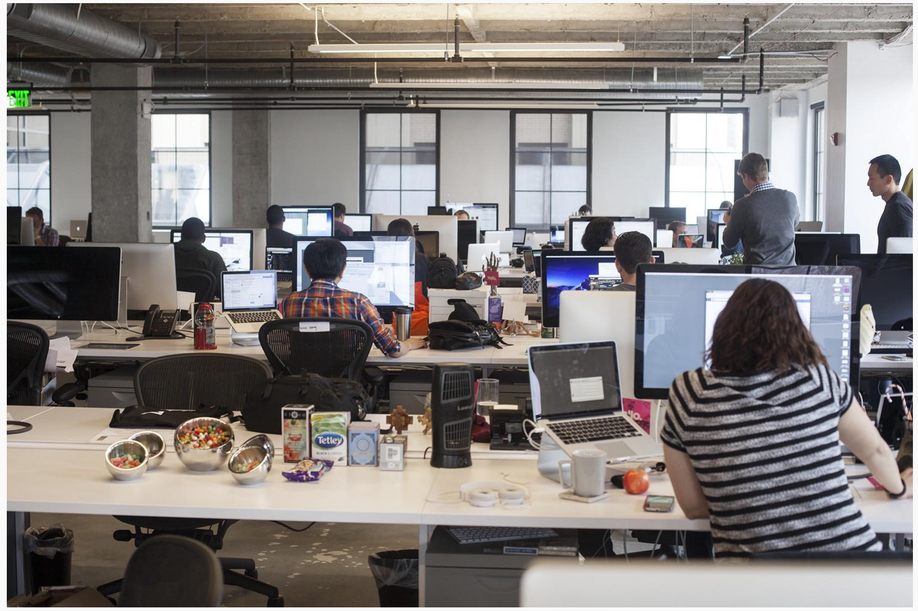
```
db.collection('inventory').insertOne(
    {
        item: 'paper',
        qty: 25,
        tags: ['very cool', 'red'],
        size: { h: 14, w: 21, uom: 'cm' }
    },
);
```

```
db.collection('inventory').updateOne(
    { item: 'paper' },
    {
        $set: { 'size.uom': 'cm', qty: 69 },
        $currentDate: { lastModified: true }
    }
);
```

```
db.collection('inventory').find({ qty: 25 });
```

# DEMO

**Demo Repo and Slides**
https://csec.club/htv



Thanks for Coming