



# To SQL or NoSQL

A tale of two paradigms

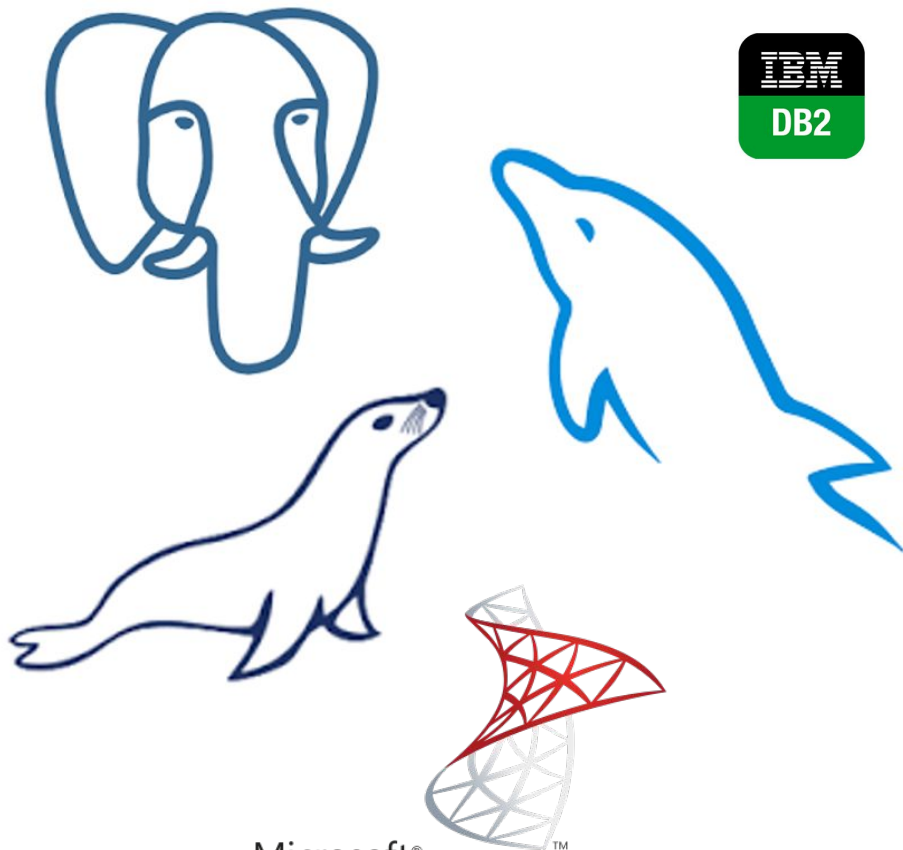
#devFestBerlin



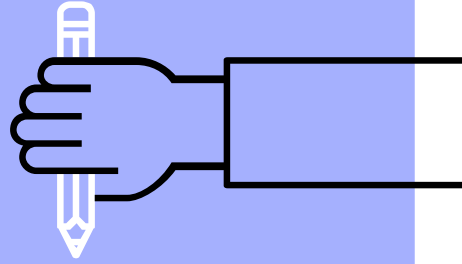
# HELLO!

My name is  
Victoria Perez Mola

I am here because I love  
talking about  
databases.

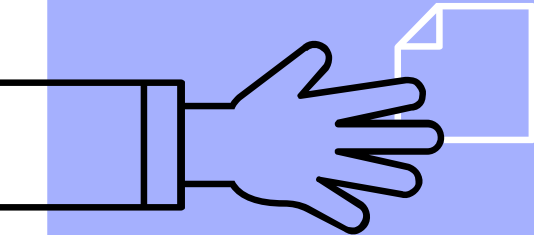


Microsoft®  
**SQL Server®**

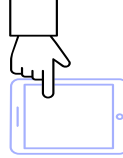


# 1. Databases

Let's start with the basics

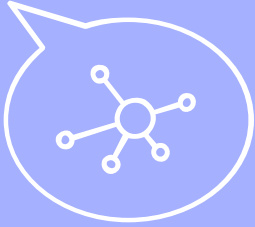


“



*A database is a collection of persistent data that is used by the application system of some given enterprise.*

– C.J. Date



# Types of Databases

## Classification

One way to classify databases involves their application, such as banking, manufacturing, or insurance.

- An **in-memory database** is a database where time is critical, such as in telecommunications.
- An **active database** includes an event-driven architecture that provides active database features like triggers and stored procedures.
- A **cloud database** relies on cloud technologies like web browser and Open APIs.
- **Data warehouses** archive data from operational data. For example, data warehousing includes extracting data from operational databases and loading it into a data warehouse.
- A **deductive database** combines logic and data.
- A **distributed database** is one in which data is distributed across multiple sites.
- A **document-oriented database** is a database that stores data in a document format.
- An **embedded database** system is a database that is integrated into an application's maintenance.<sup>[21]</sup>
- End-user databases consist of data that is used by end-users. They are much simpler than full-fledged databases.
- A **federated database system** combines data from multiple DBMSs, possibly of different types, into a single logical database.
- Sometimes the term *multi-database* is used to refer to a collection of application databases. In this case, typically each database is a separate DBMS.
- A **graph database** is a kind of NoSQL database that stores data in a graph structure.
- An **array DBMS** is a kind of NoSQL database that stores data in an array structure.
- In a **hypertext** or **hypermedia** database, data is organized into large amounts of disparate information.
- A **knowledge base** (abbreviated **KB**) is a database representing problems with their solutions.
- A **mobile database** can be carried on a mobile device.
- **Operational databases** store detailed information about a company's operations, such as demographic information about a company's customers, parts inventory, and financial data.
- A **parallel database** seeks to improve performance by distributing data across multiple processors.

## Types of DATA



01

### Distributed Database

It comprises of at least two documents situated at different destinations either on a similar system or on different systems.

02

### Centralized Database

A centralized database framework is a framework in which all information is in one single database at one single location.

03

### Personal Database

Information is gathered and stored on PCs, with a small quantity and can easily be manageable.

04

### Relational Database

It is described by a set of tables from where data can be accessed. Relational database can store a large amount of information in a set of tables, which are linked to each other.

05

### Operational Database

An operational database is a database that is used to store and manage data that is used in a business's day-to-day operations.

06

### Hierarchical Database

In a hierarchical database, data is organized into a hierarchy that is based on parent-child relationships.

07

### Cloud Database

It is a database that is hosted on a cloud platform and can be accessed from anywhere.

08

### Object-oriented Database

It is a database that stores data in the form of objects, which are linked to each other.

09

### NoSQL Database

NoSQL is a set of database management systems that are designed to handle large amounts of unstructured data.

Depending upon the usage requirements,

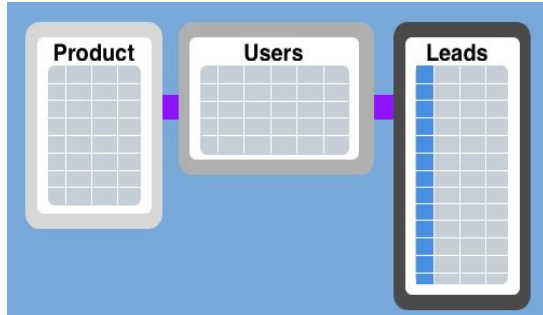
1. Centralised database.
2. Distributed database.
3. Personal database.
4. End-user database.
5. Commercial database.
6. NoSQL database.
7. Operational database.
8. Relational database.
9. Cloud database.
10. Object-oriented database.
11. Graph database.

## Different types of Database

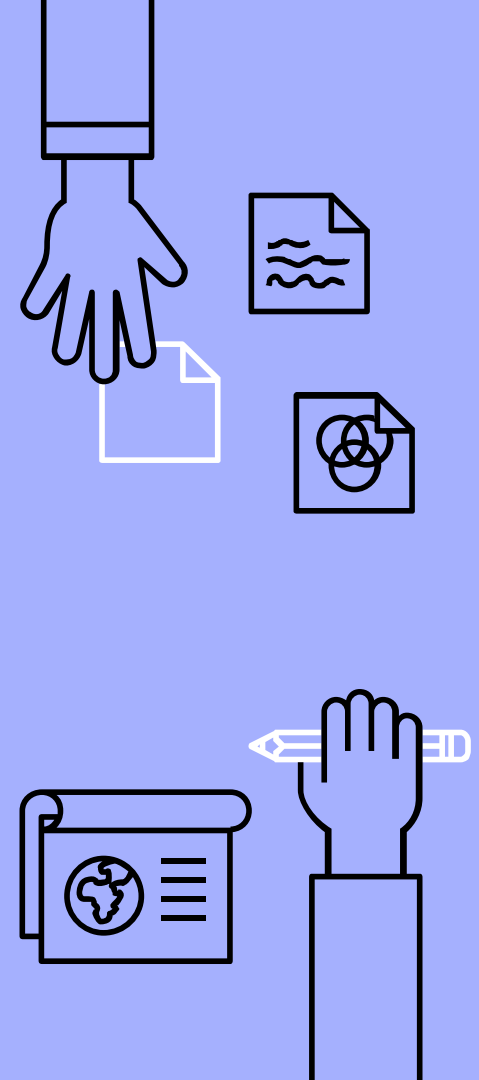
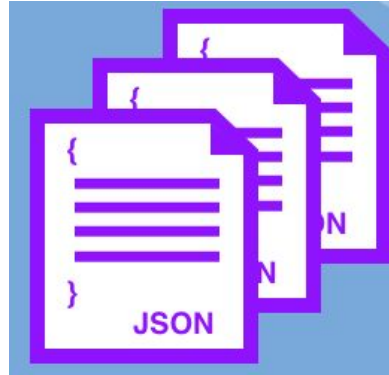


# Again, Types of Databases

Relational  
SQL  
RDBMS

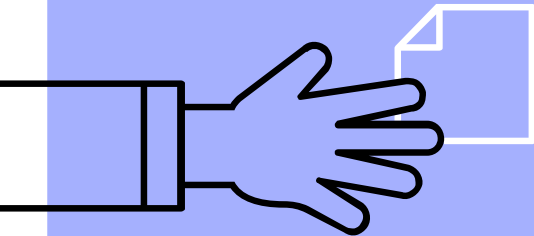
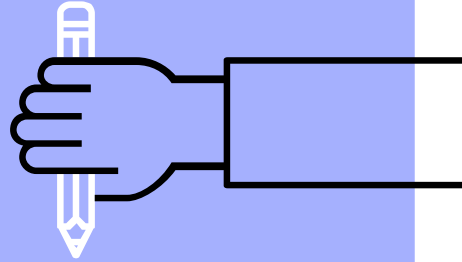


Non Relational  
NoSQL



## 2. Relational Databases

(and SQL)



# Relational Databases

- ▶ Created in the 60's
- ▶ *Structured* data
- ▶ Set of tables: columns and rows
- ▶ Relations among tables

Name	City	Age
Maria	Seattle	20
Luis	Toronto	25
Martin	Berlin	22

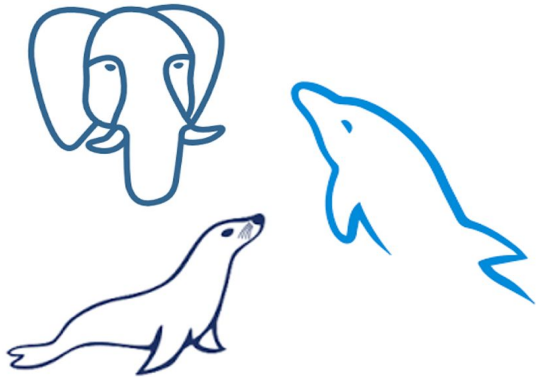
Name	Date	Payment
Maria	11/01/2019	Card
Maria	11/02/2019	Card
Luis	12/02/2019	Paypal
Maria	11/03/2019	Card





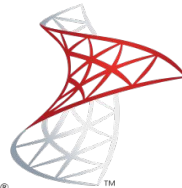
# Structured Query Language

- ▷ Appears in the 70's
- ▷ Standardized by ANSI
- ▷ Variations according vendors



**ORACLE®**

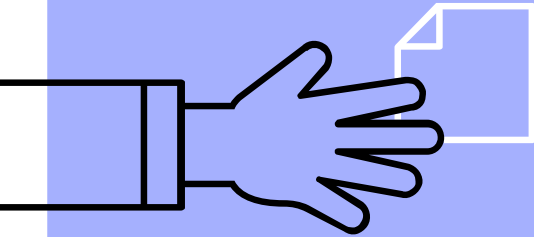
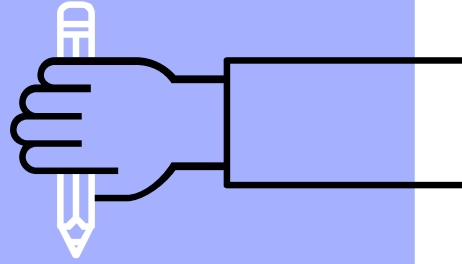
**DATABASE**



Microsoft®  
**SQL Server®**



### 3. NoSQL

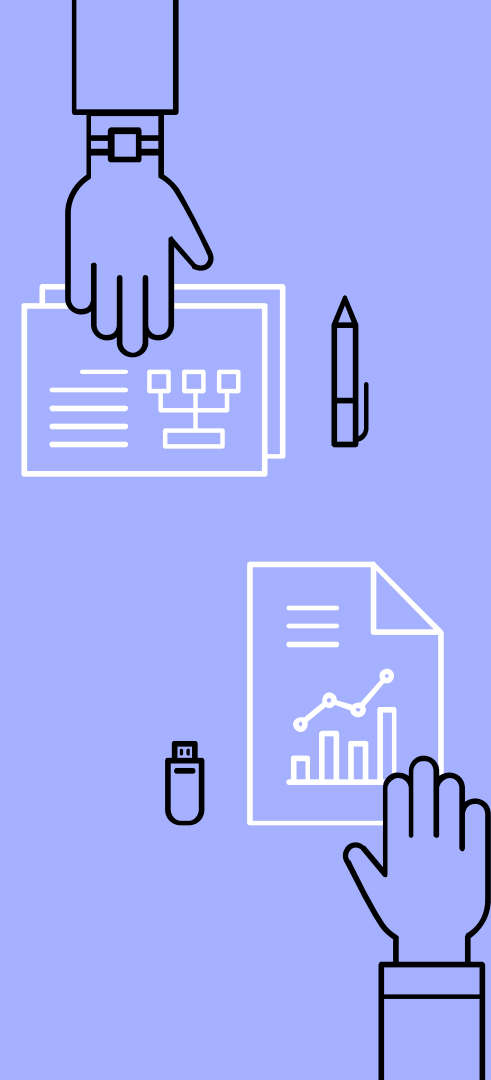
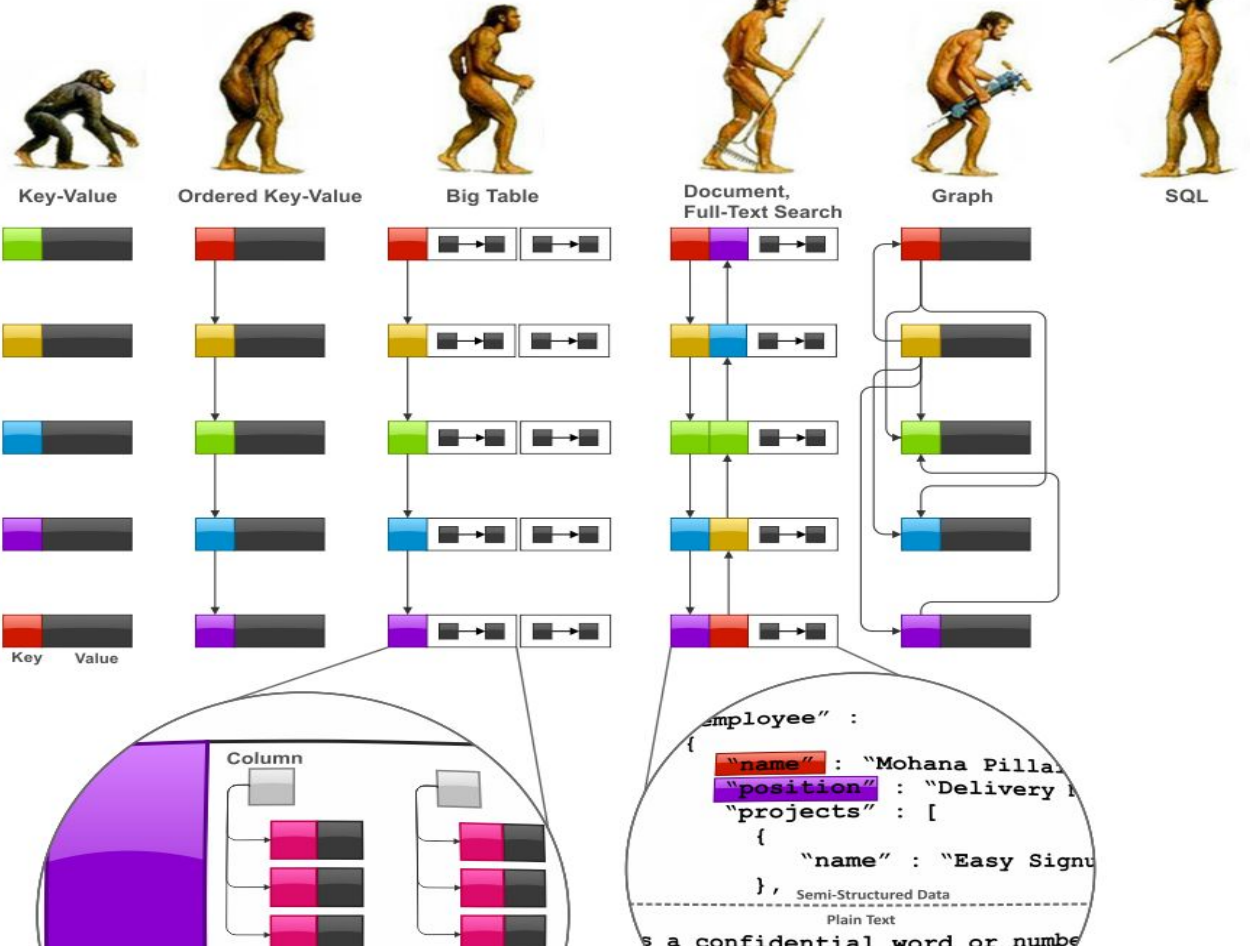


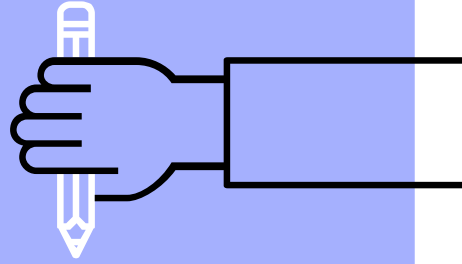
# NoSQL

- ▷ Appears in the 90's
- ▷ Explodes in 2010
- ▷ Depends mostly on the vendor
- ▷ Different subtypes

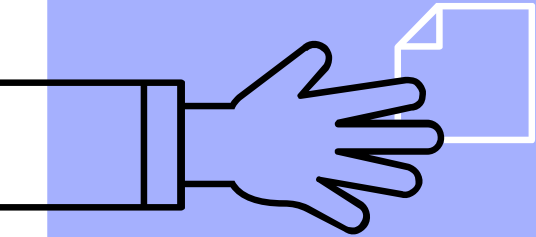


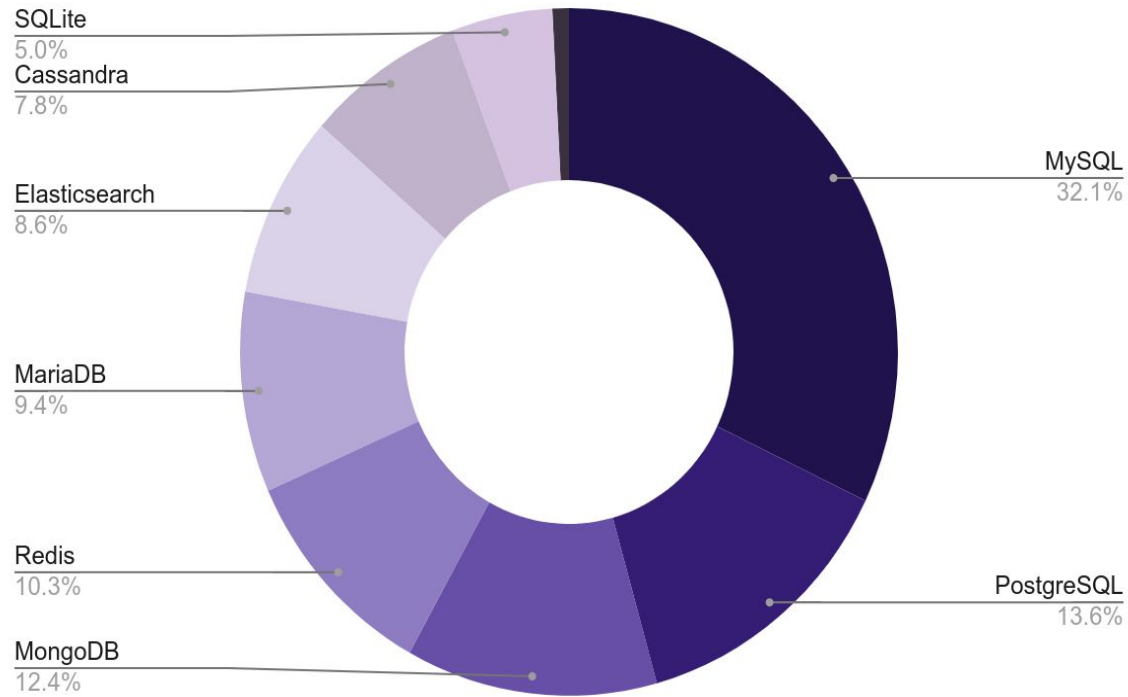
Stop following me, you fucking freaks!





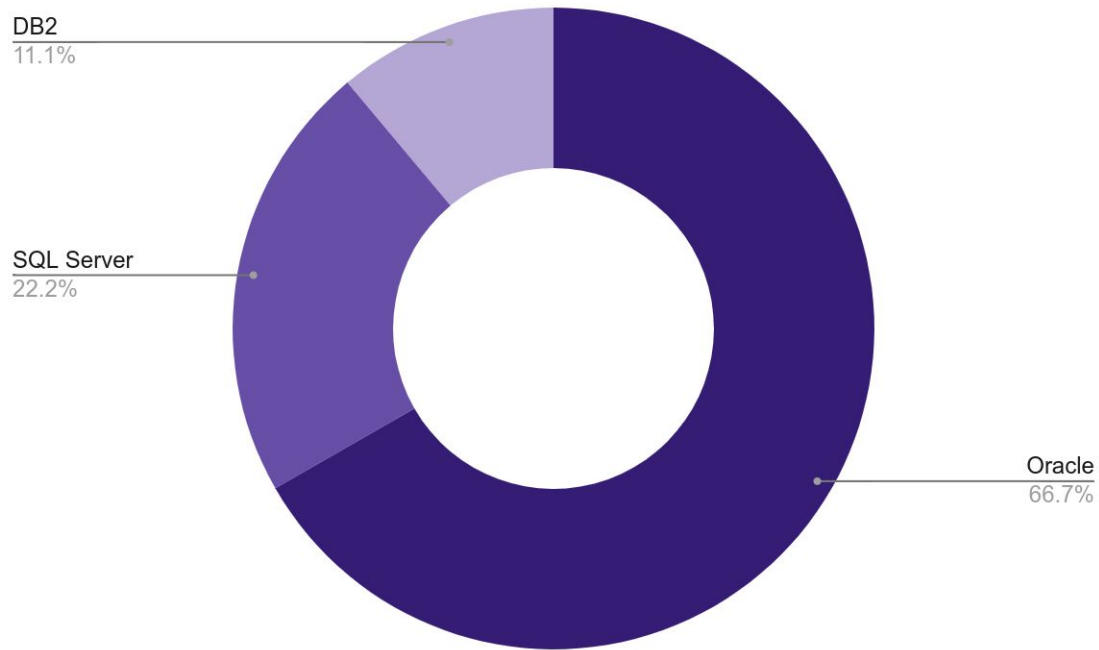
## 4. RDBMS vs NoSQL





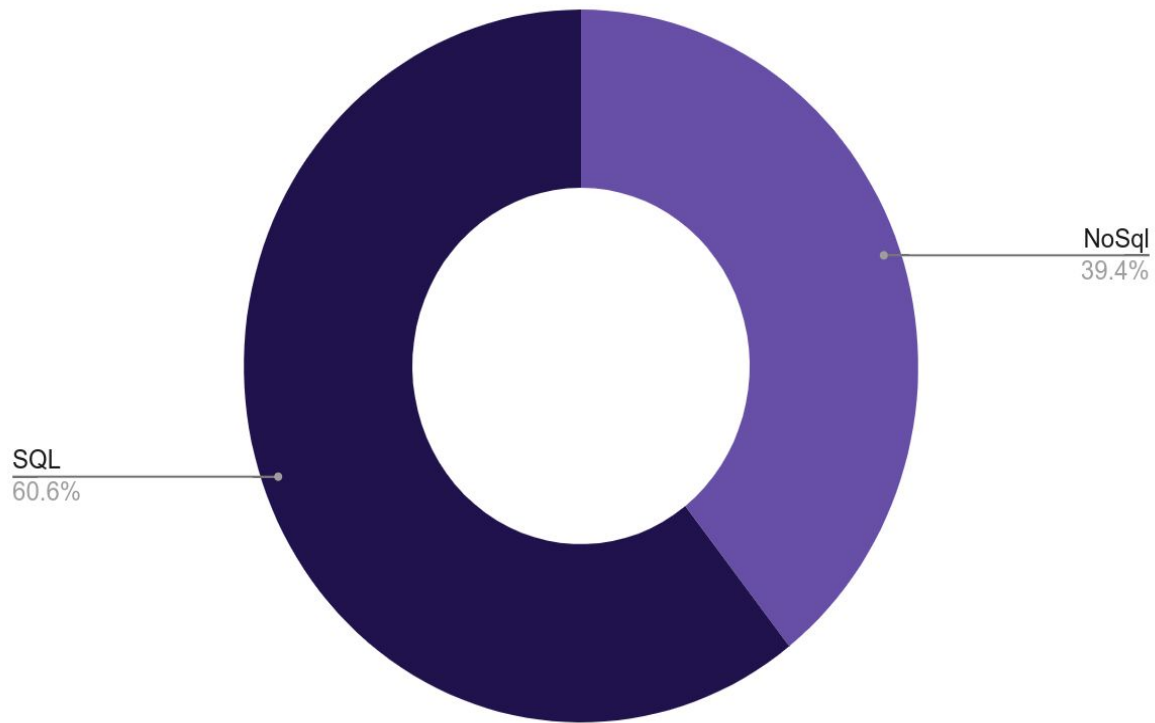
Source: dzone.com/

## Top open source Databases



Source: dzone.com/

Top commercial  
databases.  
Oracle represents  
2/3 of the Market.



Source: dzone.com/

Relational  
Databases are  
dominating the  
market.



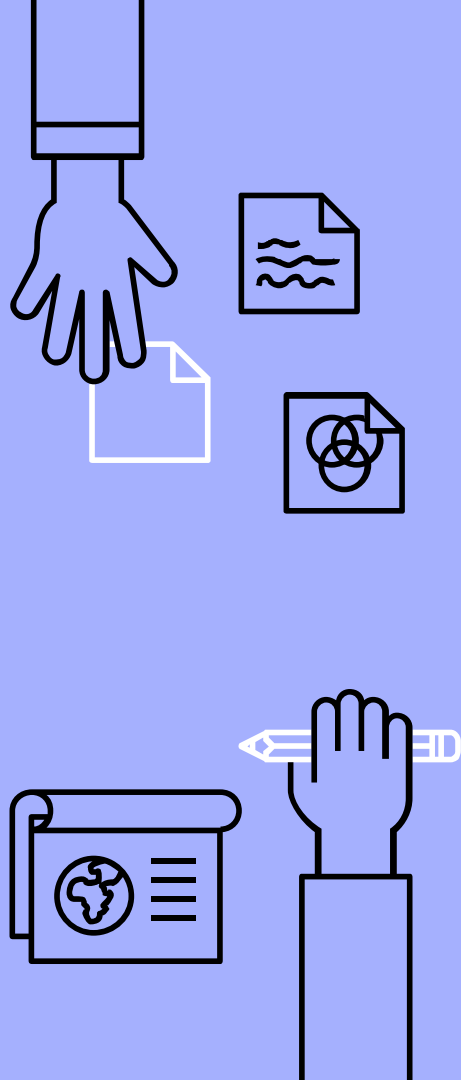
# RDBMS vs NoSQL

## RDBMS

- ▶ Requires a defined and structured schema
- ▶ Huge community supporting it
- ▶ Needs more managed scalability.
- ▶ SQL, transferable skills
- ▶ Many fantastic reporting tools
- ▶ Can offer performance problems for big data
- ▶ Support ACID

## NoSQL

- ▶ Allows the persistence of any data in the “document”
- ▶ Small community
- ▶ Easy scalability
- ▶ No structured query language
- ▶ Few reporting tools that are difficult to standardize.
- ▶ Excellent performance on big data.
- ▶ Limited support for joins
- ▶ Data is denormalized, requiring mass updates



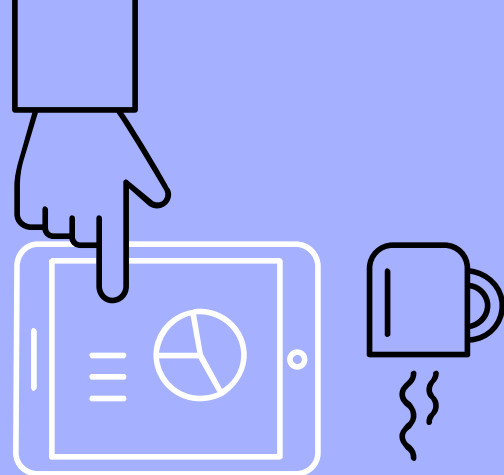
# Selection requirements

## Relational

- ▶ Medium-to-large-scale databases
- ▶ Fairly low concurrency
- ▶ ACID is a must
- ▶ Data highly correlated
- ▶ Wide assortment of data

## NoSQL

- ▶ Large scale db
- ▶ High concurrency
- ▶ ACID can be relegated
- ▶ Narrow set of data



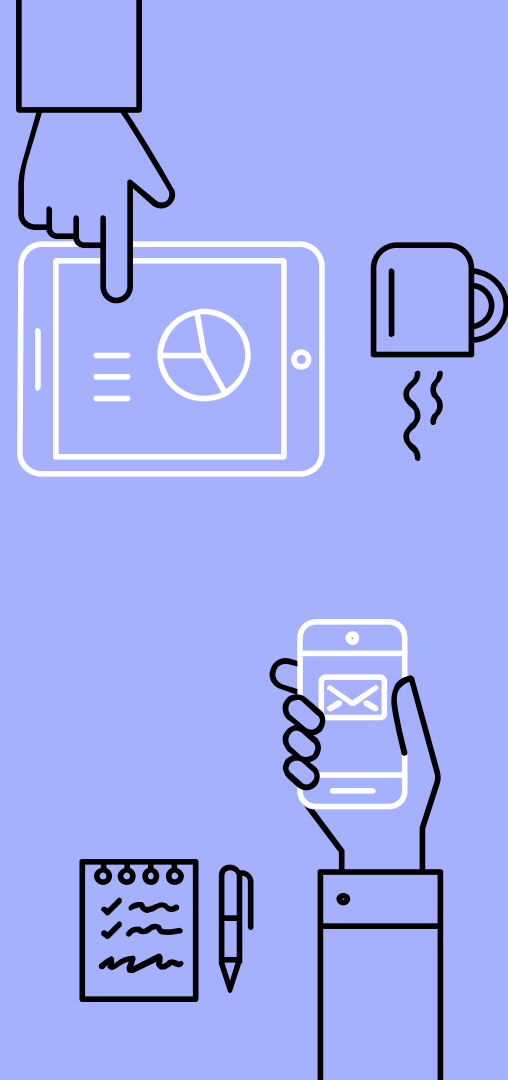
# Common uses

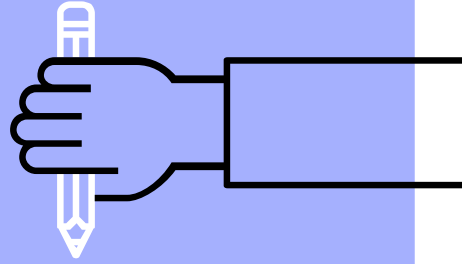
## Relational

- ▶ Accounting, finance
- ▶ Banking systems
- ▶ Transaction management systems

## NoSQL

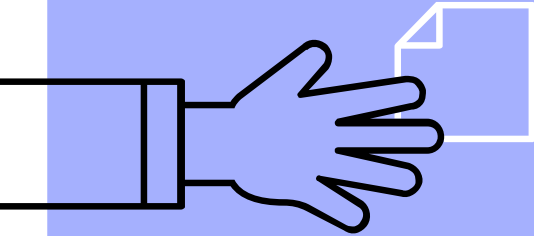
- ▶ Mobile apps
- ▶ Real-time analytics
- ▶ Content management
- ▶ Personalization
- ▶ IoT apps





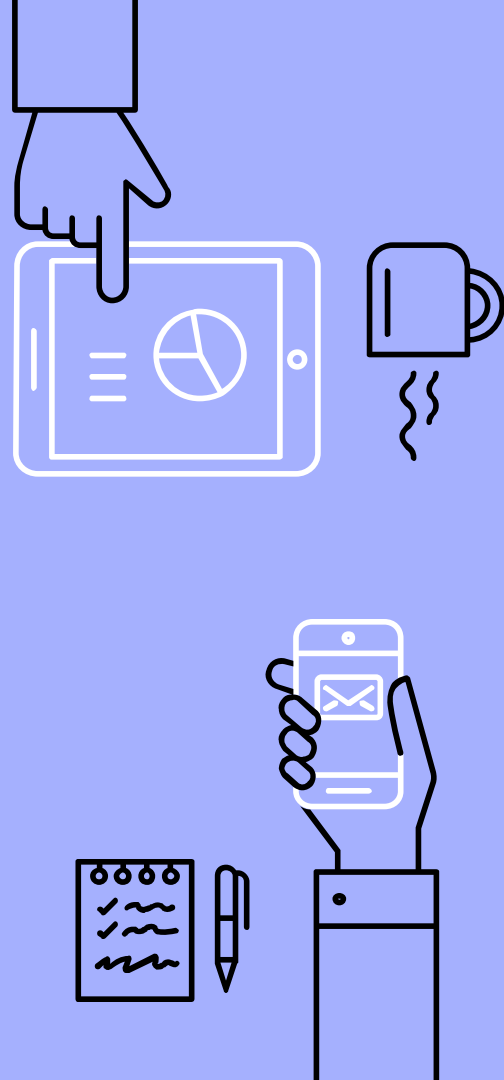
## 5. Use case #1

Ebay: from Oracle to  
Couchbase



# RDBMS goes wrong

- ▶ Oracle licensing, hardware, and support costs made scaling overly expensive
- ▶ Oracle's ACID features impacted performance of a key e-commerce application
- ▶ Oracle lacked native sharding and replication features



# Couchbase to the rescue

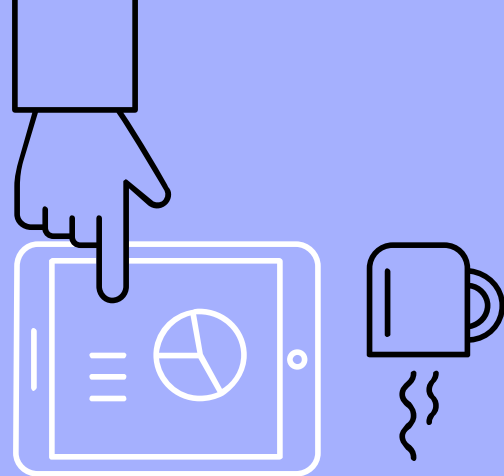
Ebay implemented Couchbase server in 2014.

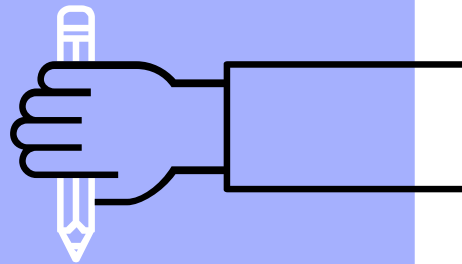
## Outcomes

- ▶ Linear scalability and high read/write throughput keep applications highly responsive even as users increase
- ▶ Location-aware low latency querying boosts performance for 110B Couchbase calls per day
- ▶ Auto-sharding distributes data evenly across nodes, while cross datacenter replication keeps sites highly available
- ▶ Flexible schema increases developer agility

## Key features

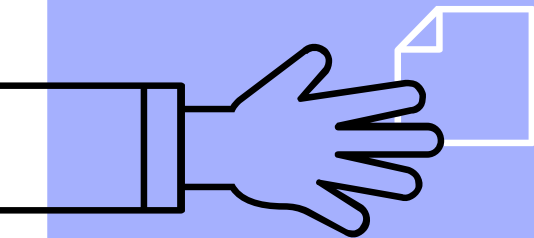
- ▶ N1QL: the power of SQL and the flexibility of JSON
- ▶ In-memory database
- ▶ Cross datacenter replication





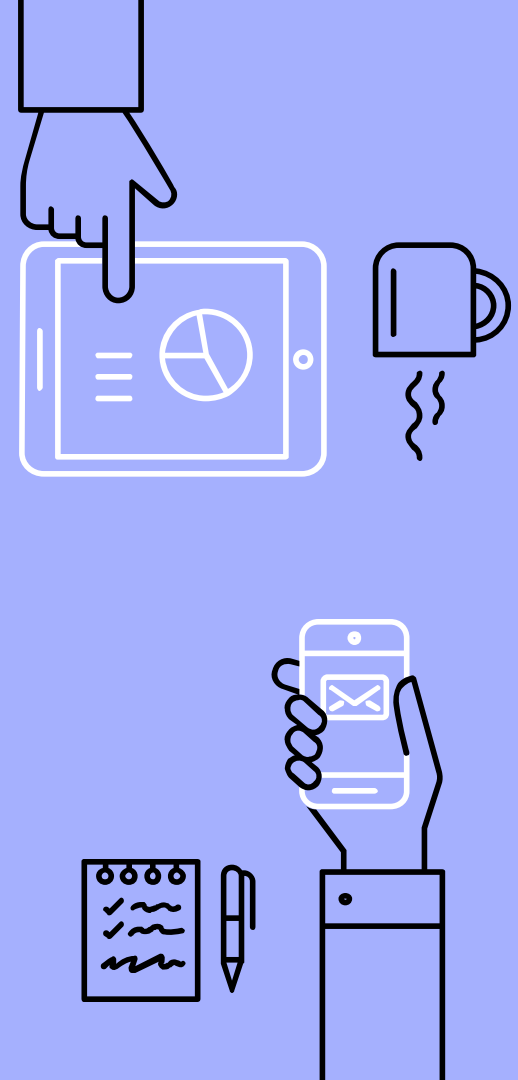
## 5. Use case #2

Flexcoin and Poloniex:  
from MongoDB to nothing



# Flexcoin:NoSQL goes wrong

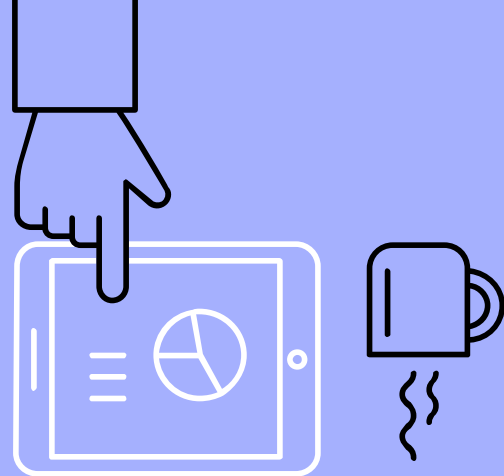
- ▶ Bitcoin exchange shut down in March 2014
- ▶ Flaw in the code which allows transfers between flexcoin users.
- ▶ Sent thousands of simultaneous requests until the sending account was overdrawn, before balances were updated.



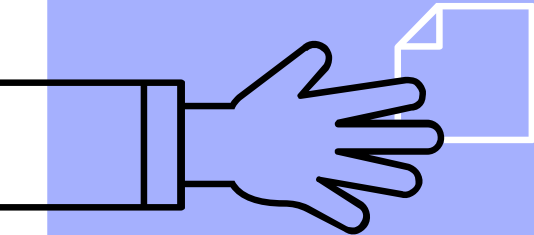
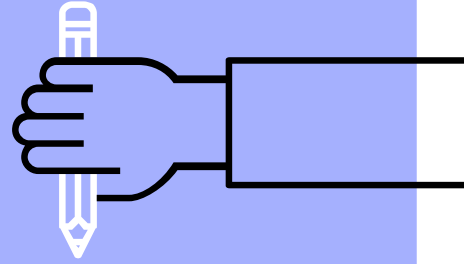


# Poloniex:NoSQL goes wrong

- ▶ Bitcoin exchange attacked in March 2014
- ▶ Vulnerability in the code that takes withdrawals.
- ▶ Placed several simultaneous withdrawals which resulted successful even though there was no real funds to cover.

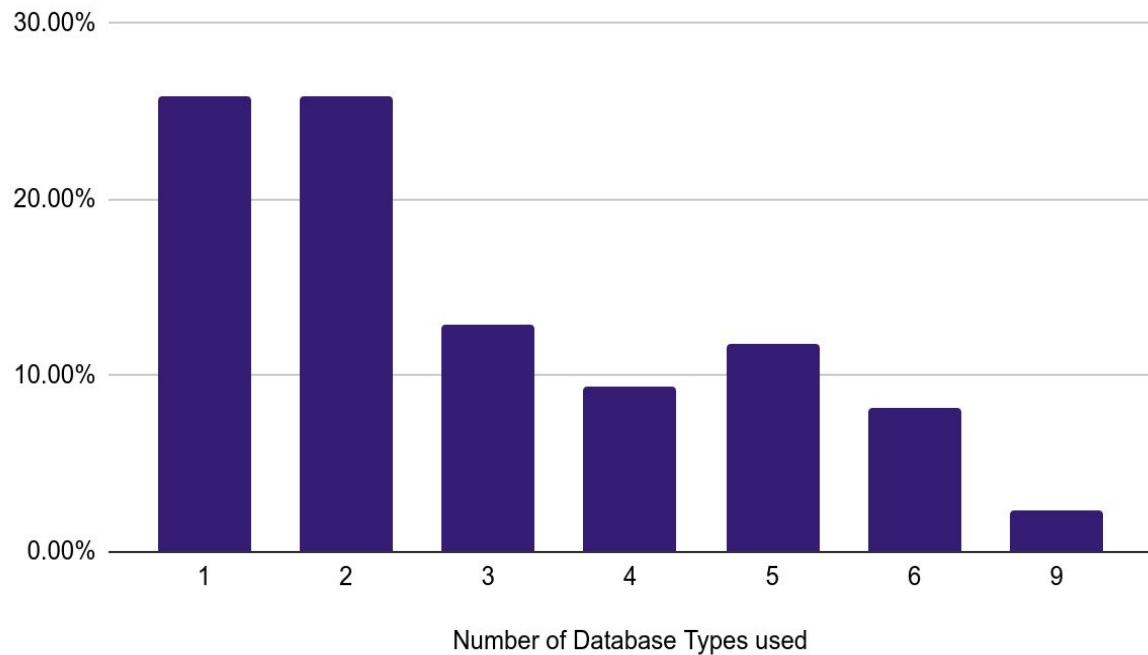


## 6. Conclusion



Understand your data and  
business needs

## Number of Database Types used



Source: DZone

Sometimes your  
business needs are  
not “data”  
compatible.  
Try Polyglot  
Persistence.

# THANK YOU!

```
1 SELECT 'Goodbye', 'Thank you'  
2 FROM devfest  
3 WHERE location = 'Berlin'  
4 AND year = 2019  
5
```