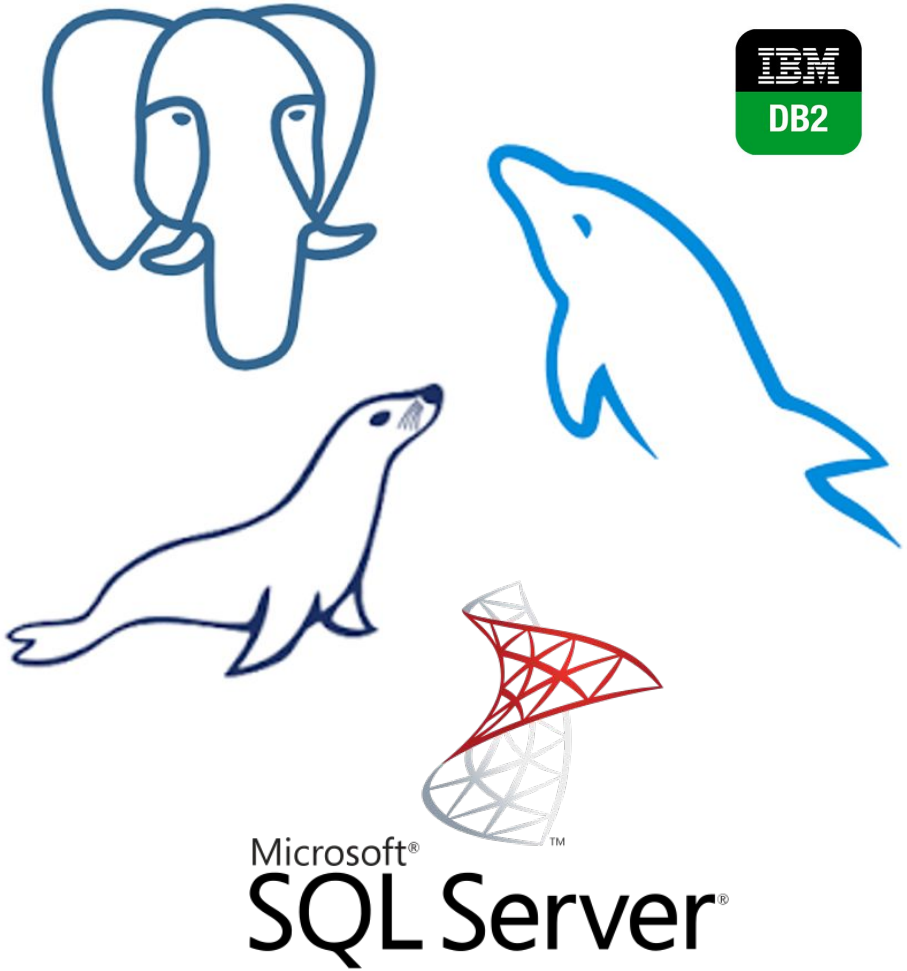
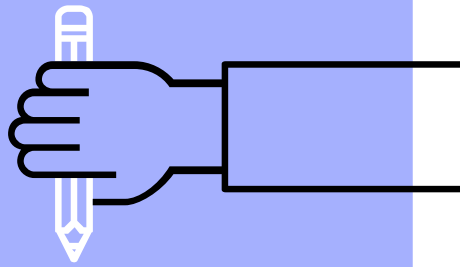

To SQL or NoSQL
A tale of two paradigms
#devFestBerlin

HELLO!

My name is Victoria Perez Mola

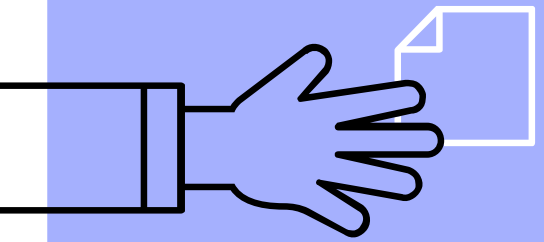
I am here because I love talking about databases.

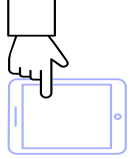




1. Databases

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.

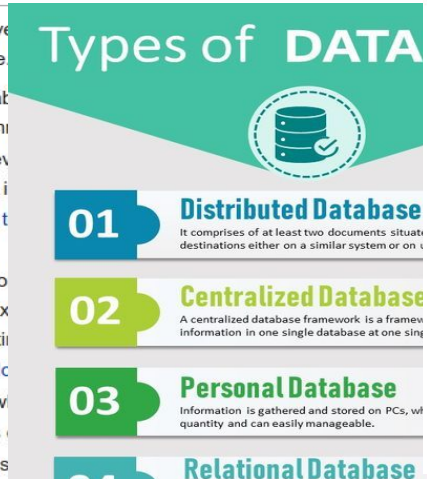
– *C.J. Date*

Types of Databases

Classification

One way to classify databases involve
banking, manufacturing, or insurance

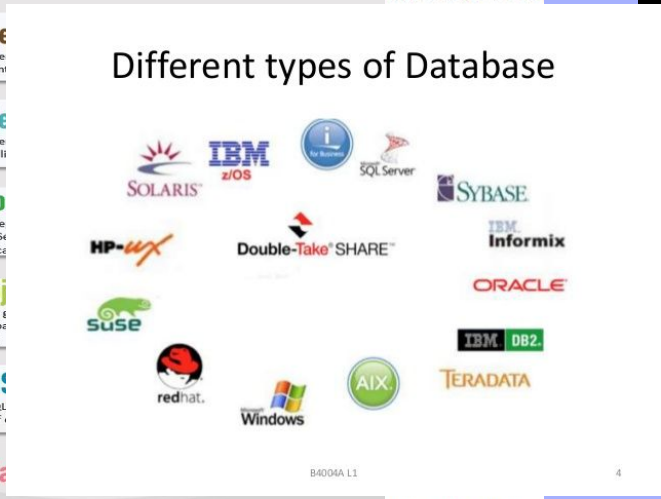
- An in-memory database is a datab
time is critical, such as in telecom
- An active database includes an ev
provide active database features i
- A cloud database relies on cloud t
browser and Open APIs.
- Data warehouses archive data fro
access to operational data. For ex
data warehousing include extracti
- A deductive database combines lo
- A distributed database is one in w
- A document-oriented database is
- An embedded database system is
maintenance.[21]
- End-user databases consist of dat
them are much simpler than full-fl
- A federated database system com
DBMSs, possibly of different types
- Sometimes the term *multi-databas*
application. In this case, typically
- A graph database is a kind of NoS
databases such as triplestores an
- An array DBMS is a kind of NoSQ
- In a hypertext or hypermedia data
large amounts of disparate inform
- A knowledge base (abbreviated K
representing problems with their s
- A mobile database can be carried
- Operational databases store detai
demographic information about a
parts inventory, and financial data
- A parallel database seeks to impr



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.

ctions of documen



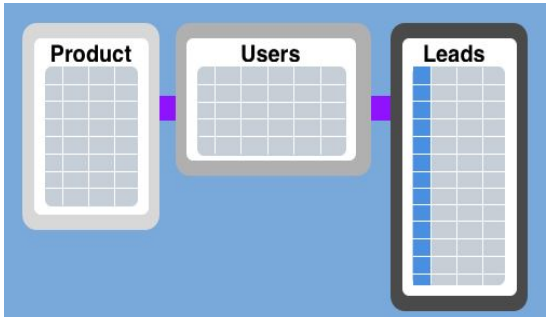
data, building inde

Again, Types of Databases

Relational

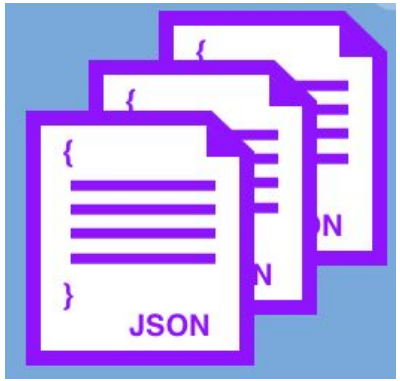
SQL

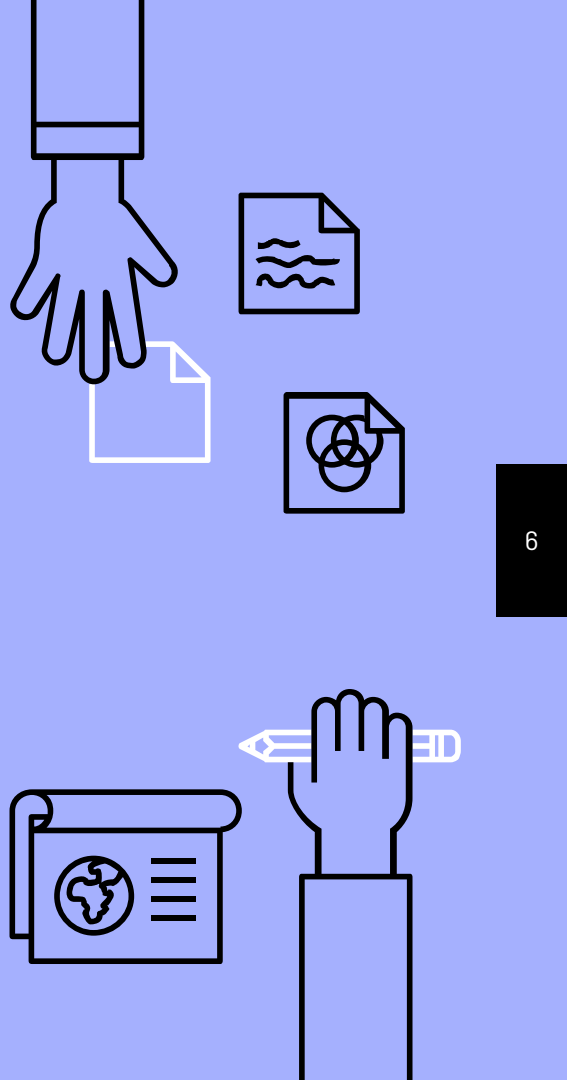
RDBMS

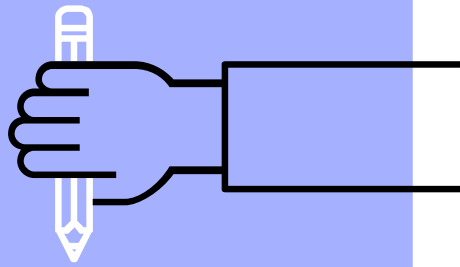


Non Relational

NoSQL

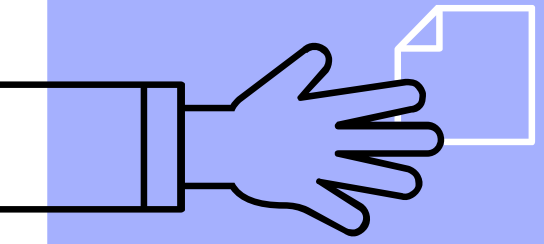






2. Relational Databases

(and SQL)

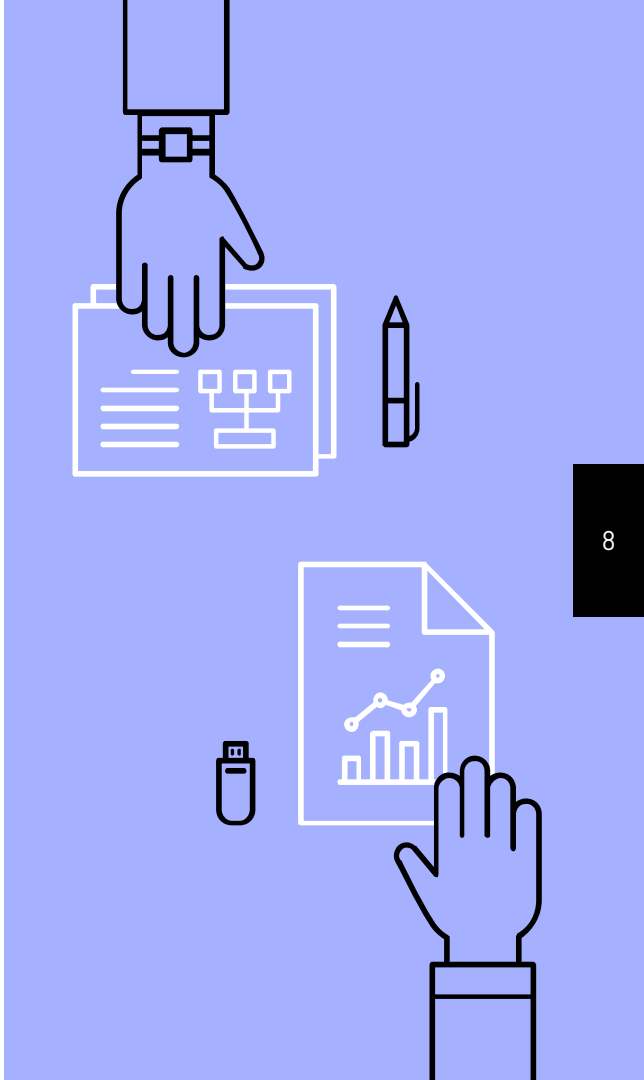


Relational Databases

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

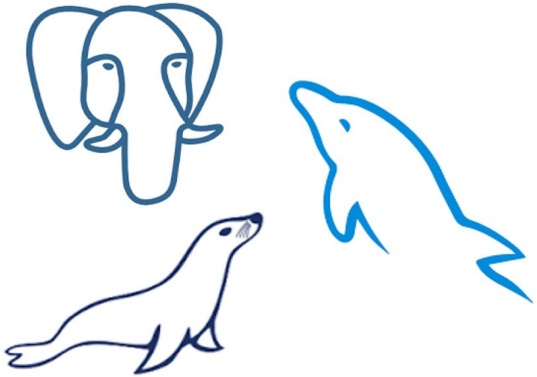
Name	City	Age
Maria	Seattle	20
Luis	Toronto	25
Martin	Berlin	22

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

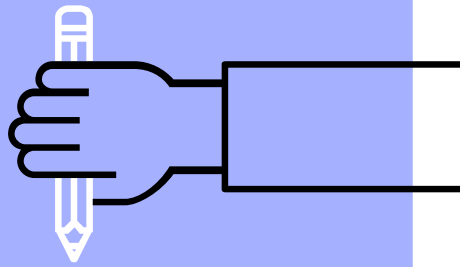


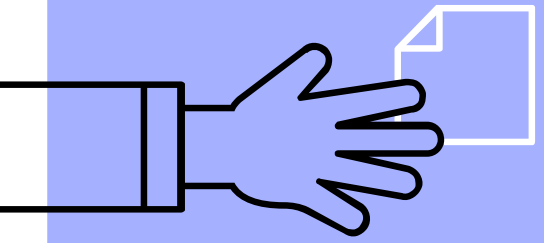
Structured Query Language

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



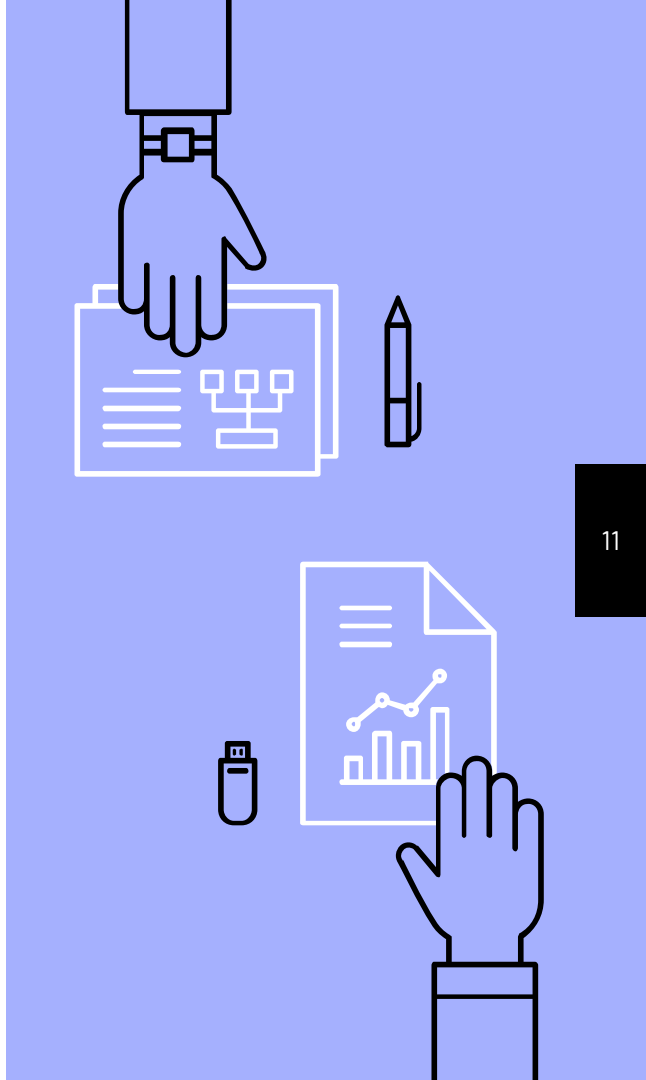
3. NoSQL

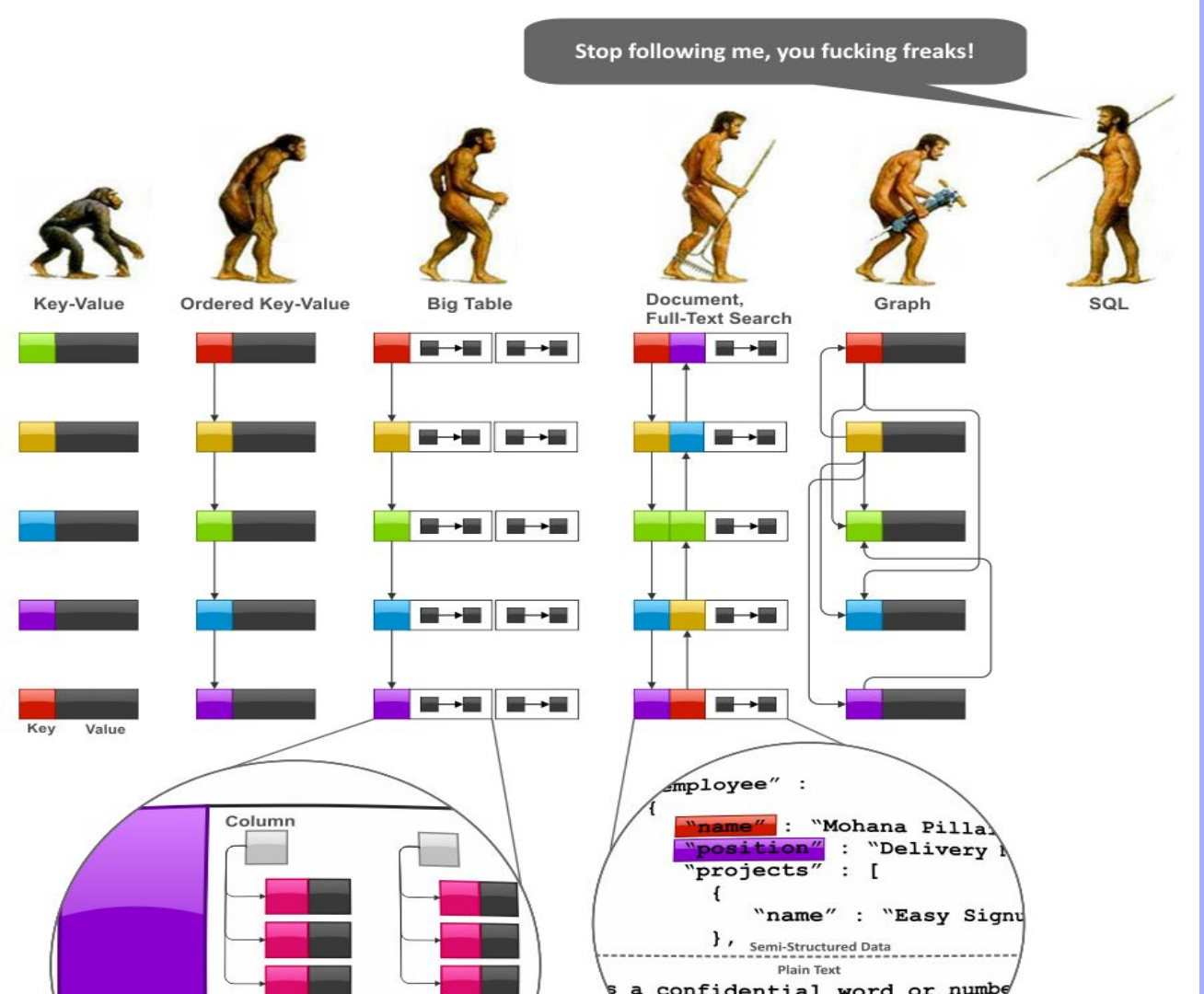


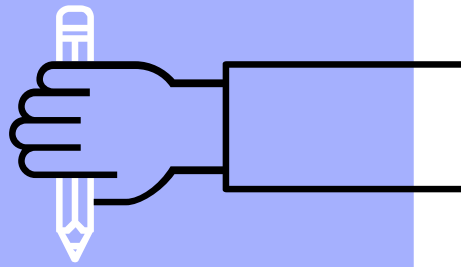


NoSQL

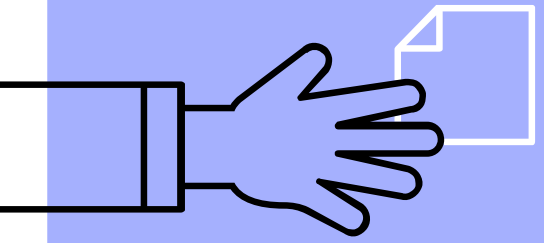
- Appears in the 90's
- Explodes in 2010
- Depends mostly on the vendor
- Different subtypes

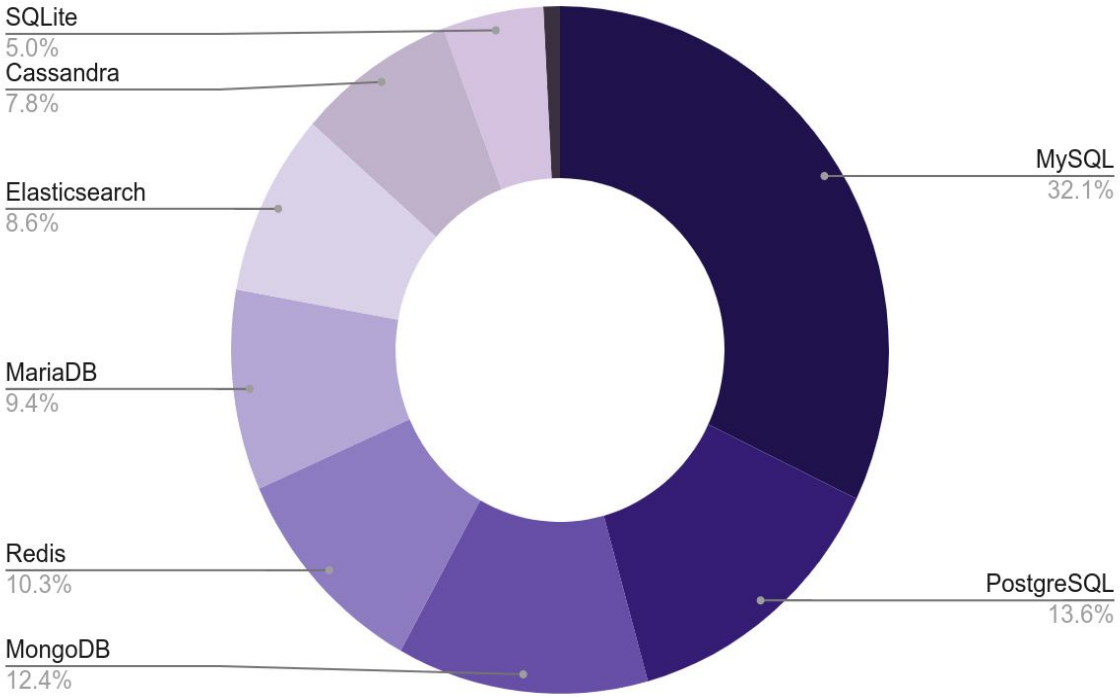



Stop following me, you fucking freaks!
Key-Value
Ordered Key-Value
Big Table
Document, Full-Text Search
Graph
SQL
Key
Value
Column
"name" : "Mohana Pilla
"position" : "Delivery
"projects" : [
{
"name" : "Easy Signu
},
Semi-Structured Data
Plain Text



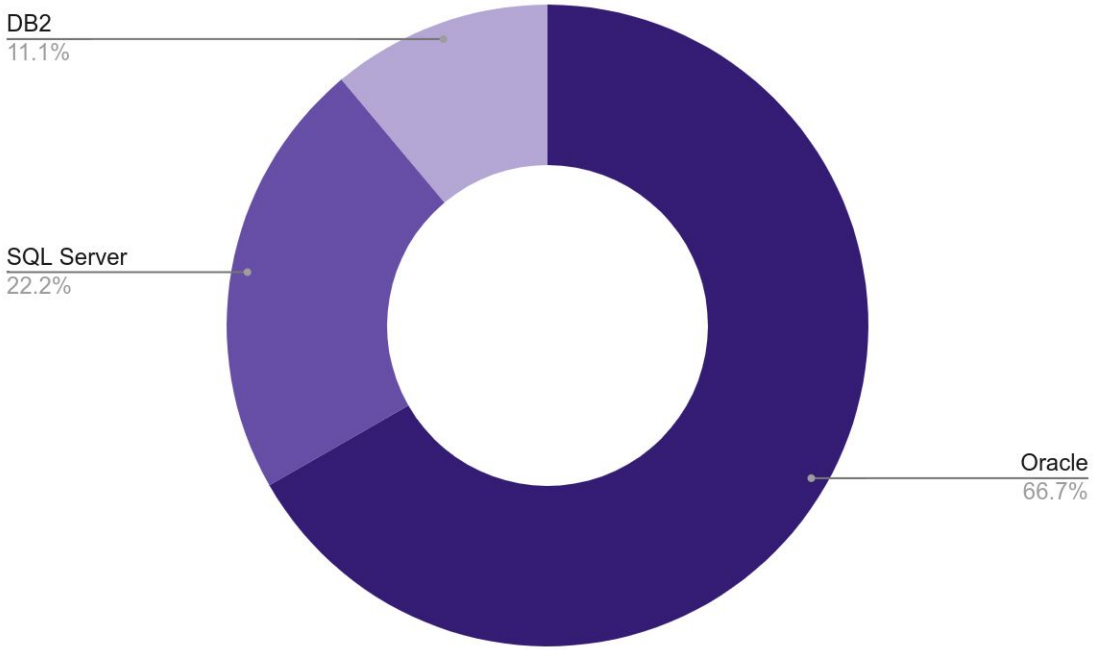
4. RDBMS vs NoSQL





Source: dzone.com/

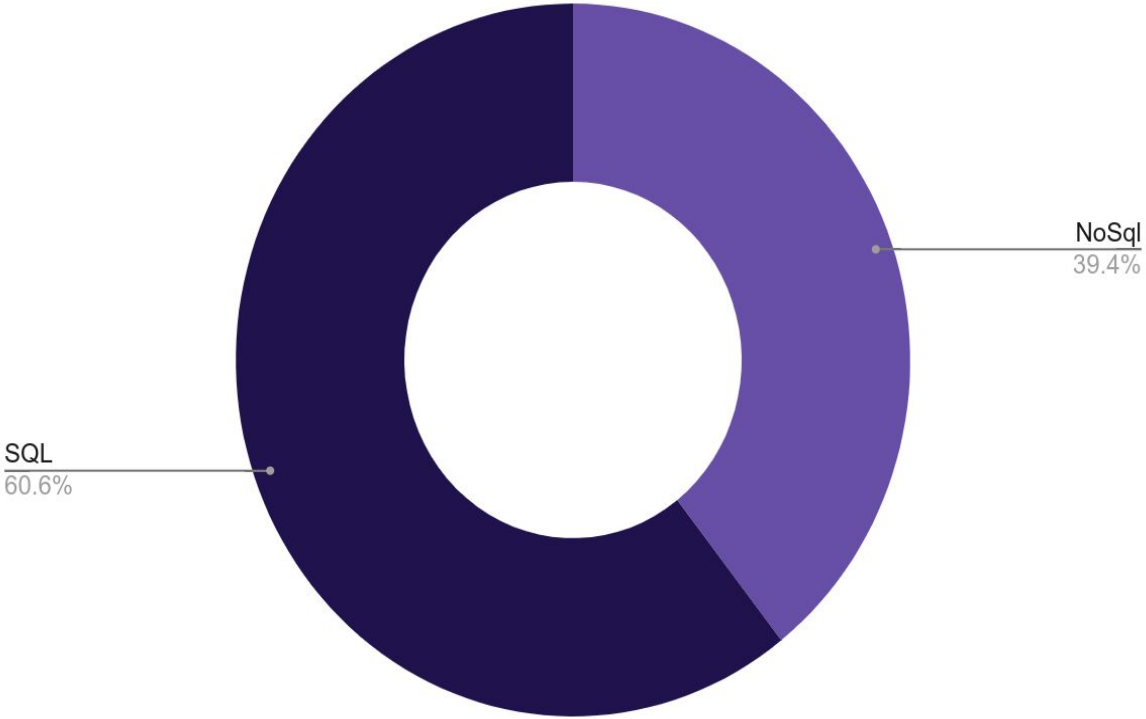
Top open source Databases



Source: dzone.com/

Top commercial databases.

Oracle represents 2/3 of the Market.



Source: dzone.com/

Relational Databases are dominating the market.

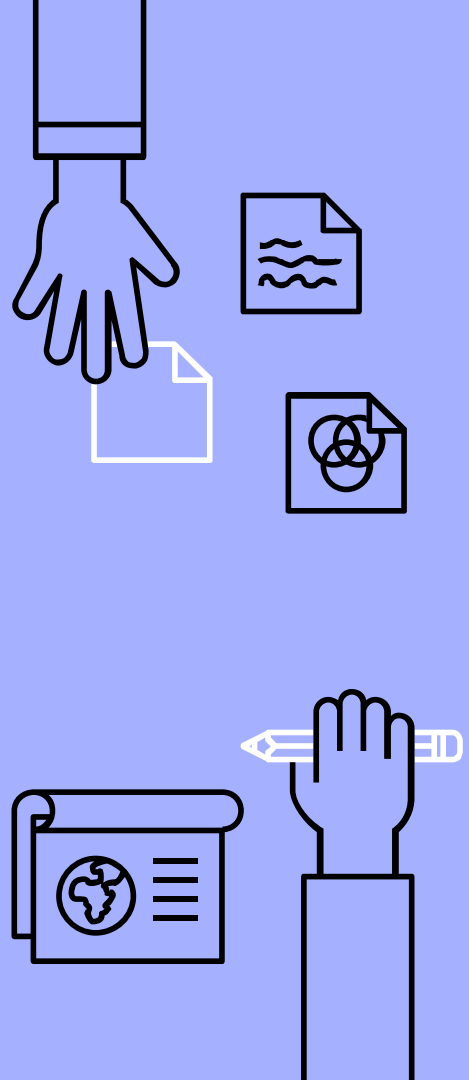
RDBMS vs NoSQL

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



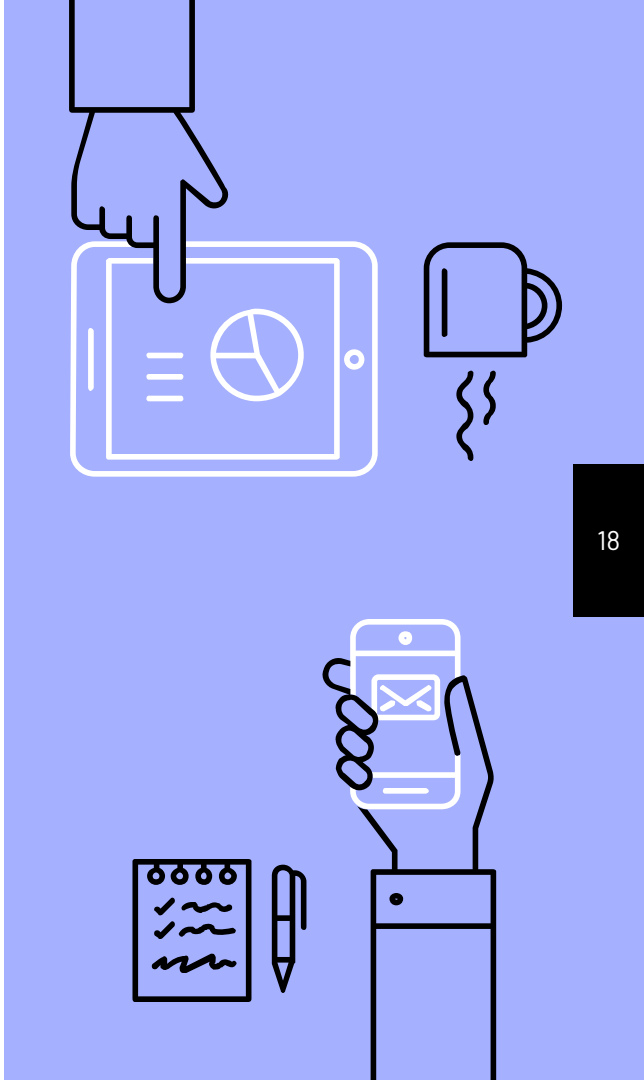
Selection requirements

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



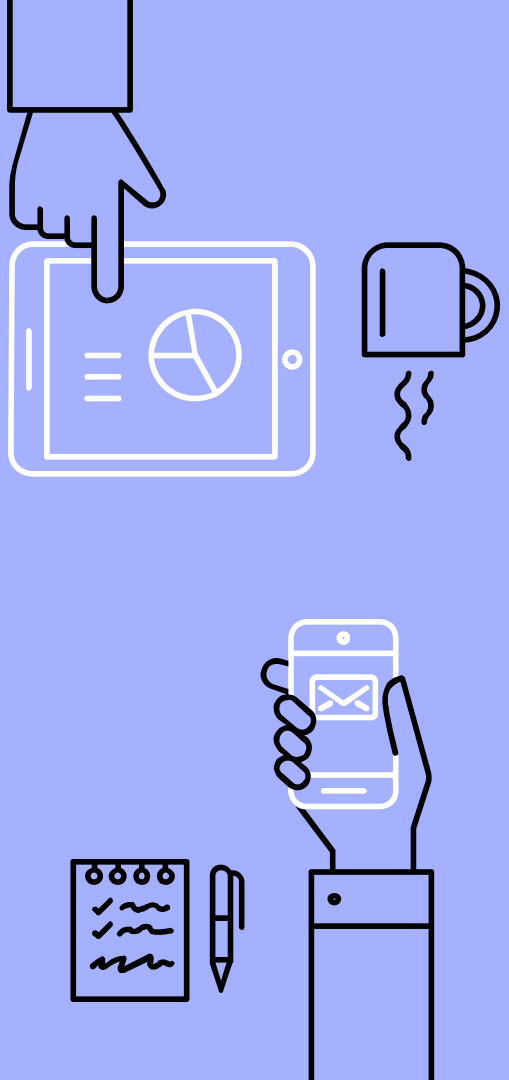
Common uses

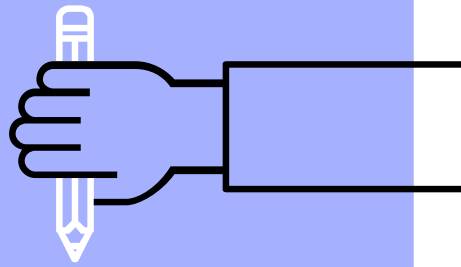
Relational

- Accounting, finance
- Banking systems
- Transaction management systems

NoSQL

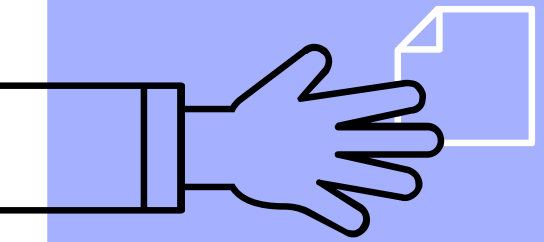
- Mobile apps
- Real-time analytics
- Content management
- Personalization
- IoT apps





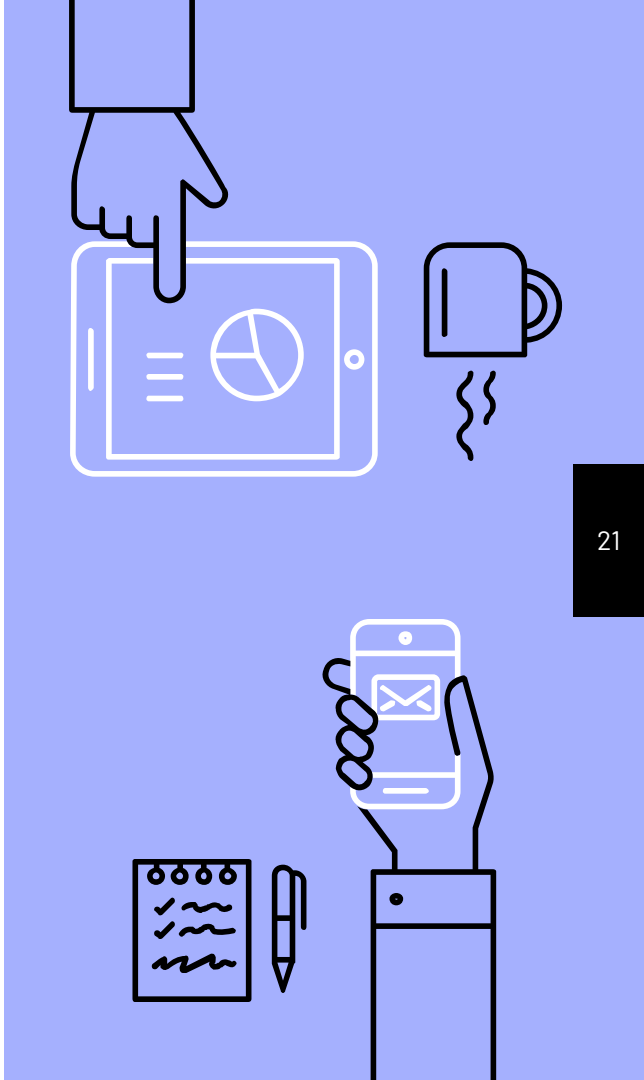
5. Use case #1

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

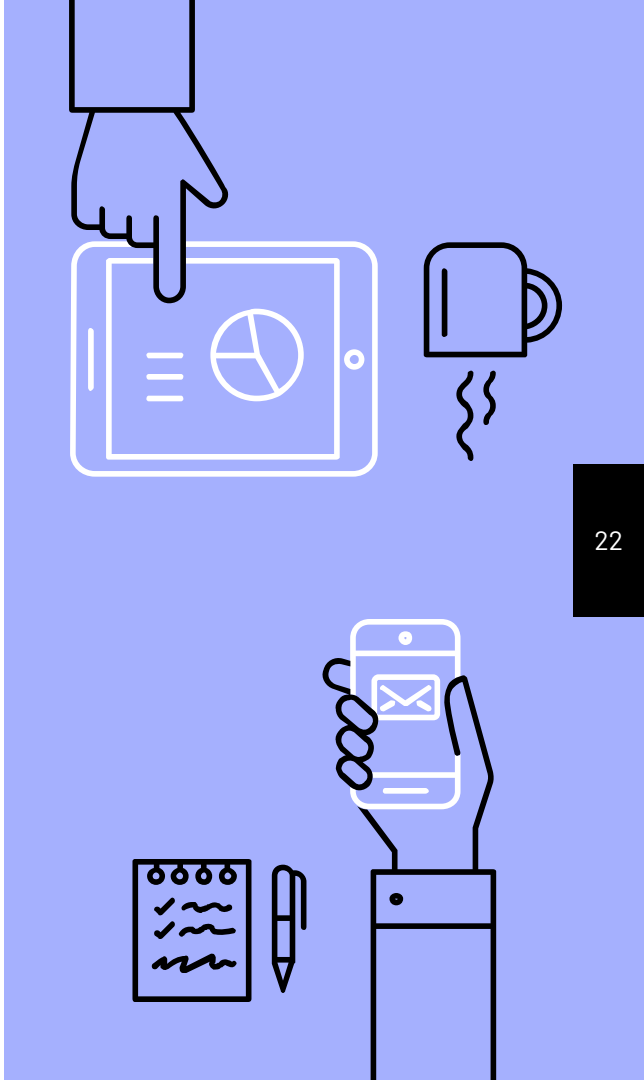
Ebay implemented Couchbase server in 2014.

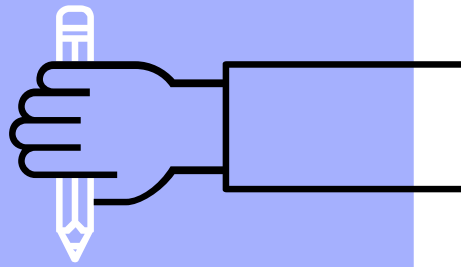
Outcames

- 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

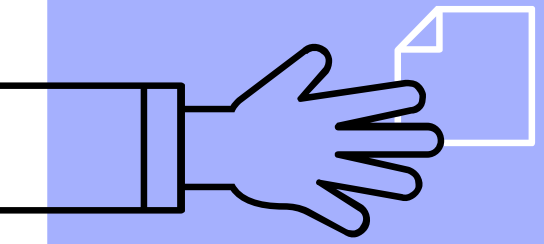
- N1QL: the power of SQL and the flexibility of JSON
- In-memory database
- Cross datacenter replication





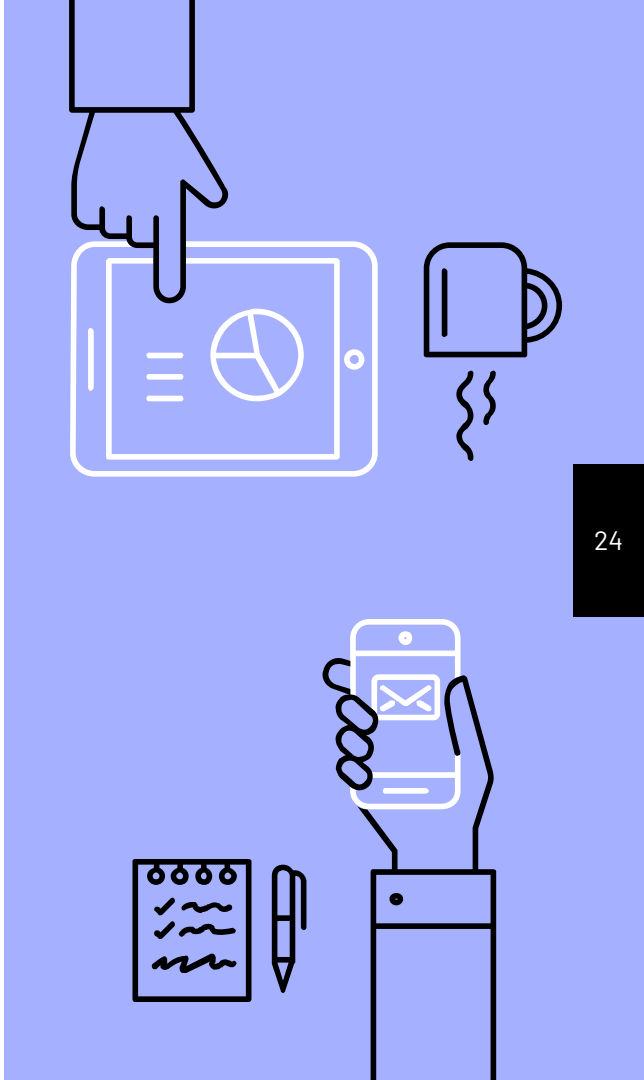
5. Use case #2

Flexcoin and Poloniex:
from MongoDB to nothing



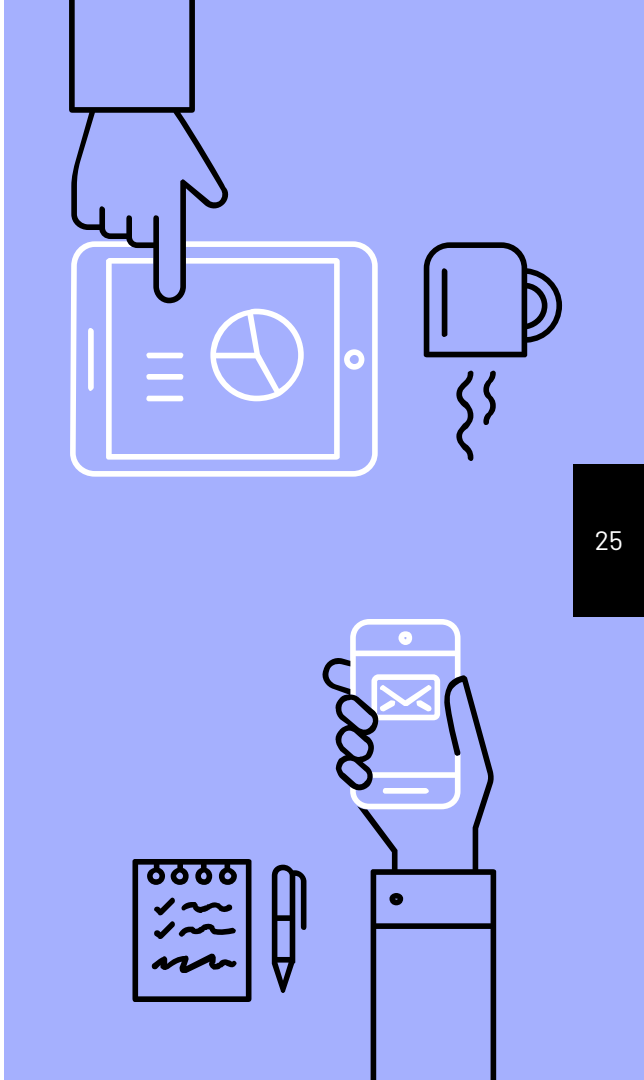
Flexcoin:NoSQL goes wrong

- 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.

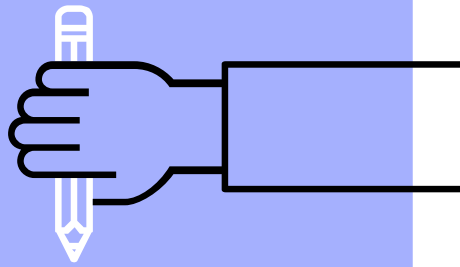


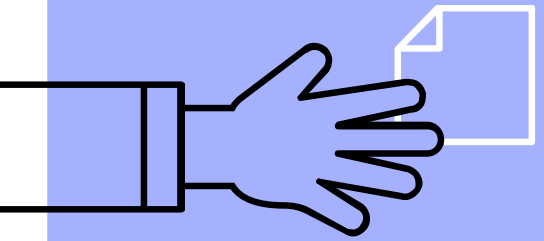
Poloniex:NoSQL goes wrong

- 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.

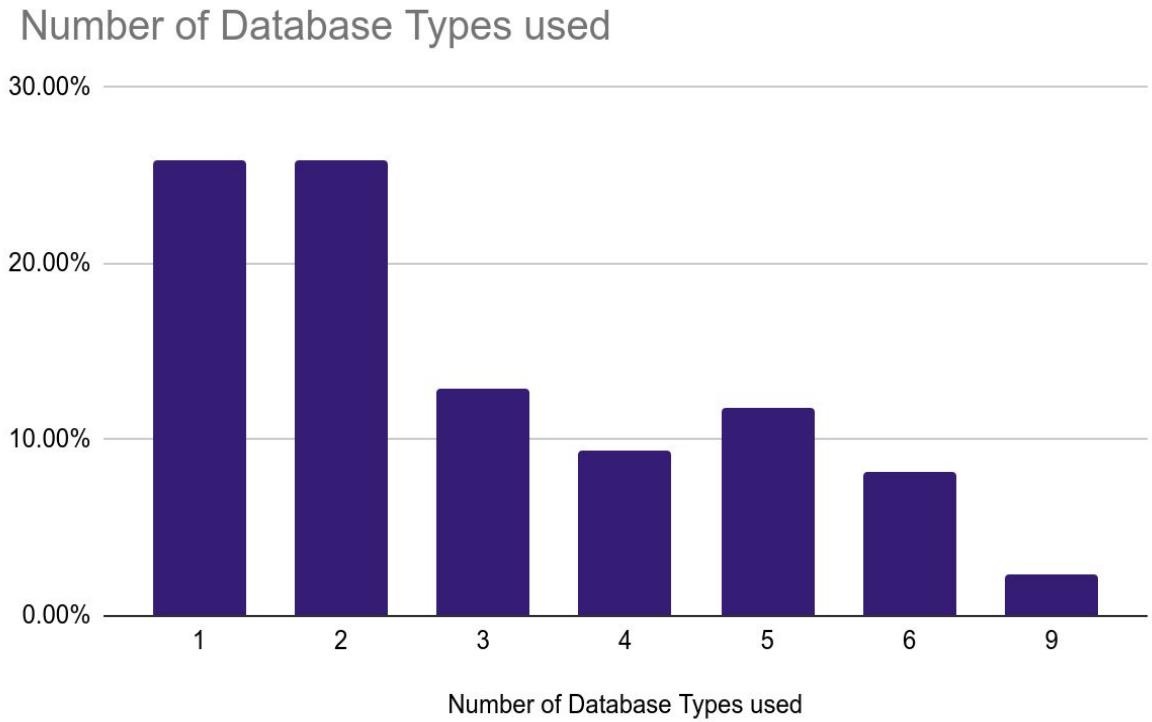


6. Conclusion





Understand your data and business needs



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
```