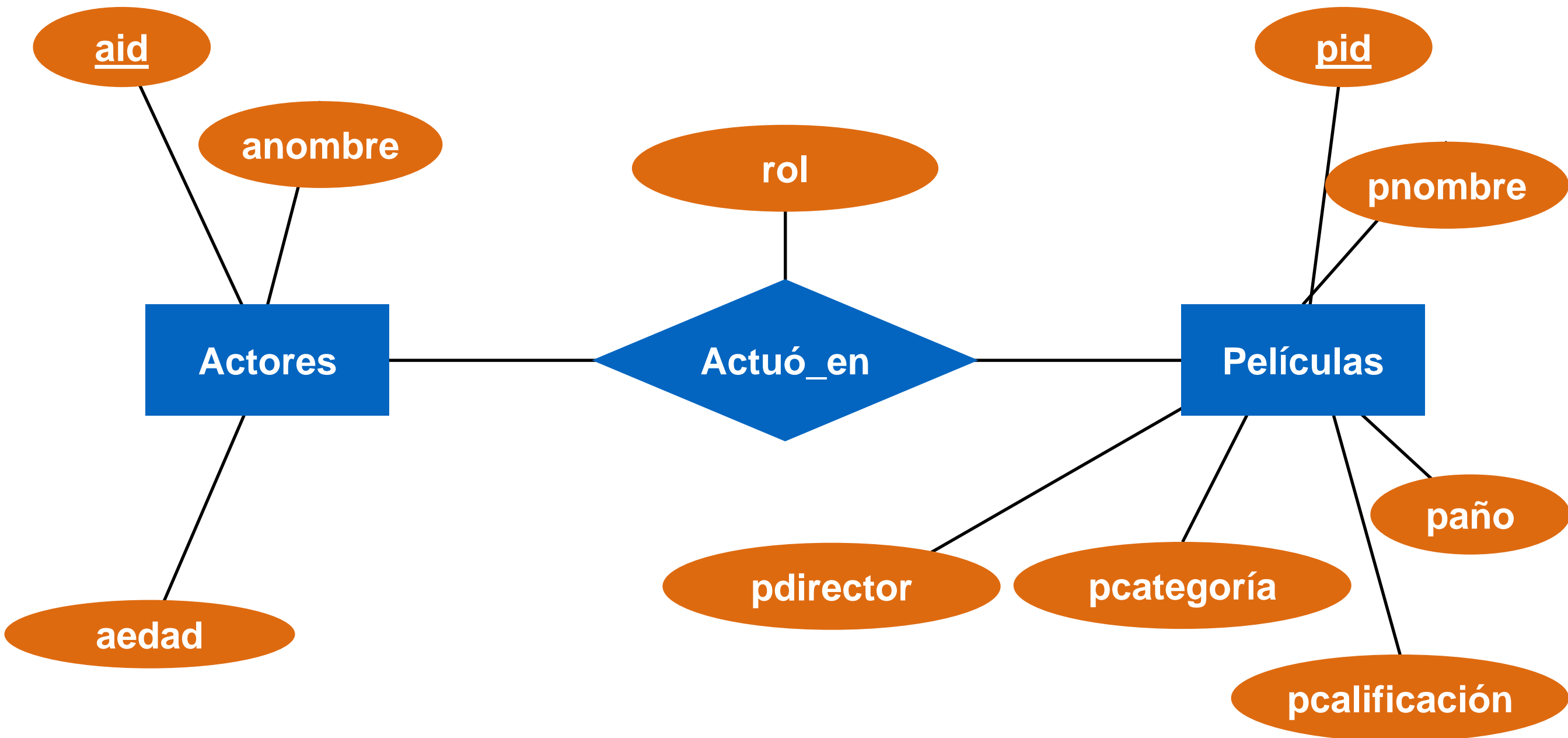


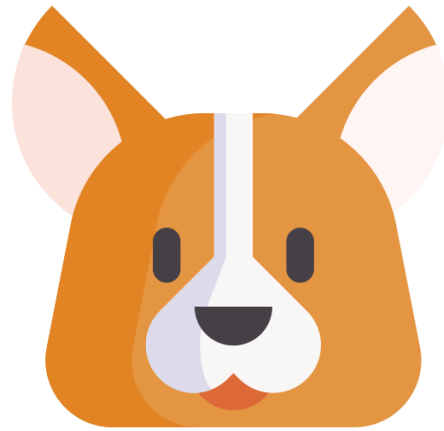
Te parece ver un ejemplo en vivo  
para ver que cosas se pueden  
hacer?

# Diagrama E/R Ejemplo en vivo



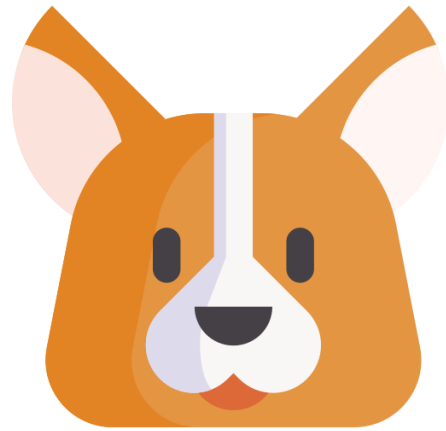


Ejemplo en vivo



Don P:

- Quiere comprar accesorios para mascotas
- Como el común de los mortales (o lomitos), no tiene por qué conocer el funcionamiento de la base de datos



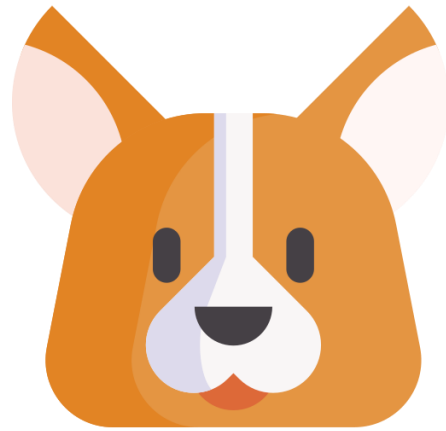
Me gustaría poder comprar mis  
accesorios en una tienda en línea



Si hubiese hecho mi programa en Python, desarrollaría una aplicación web que usara los datos



Don N., por supuesto que una aplicación web puede conectarse a la base de datos relacional



Don N., me contaron que al usar un  
framework web ya no hay que saber  
SQL





Entonces lo que aprendí no me va a servir de nada?

# ORM

Los frameworks web tienen librerías para abstraerse de la base de datos

Un ORM (Object-Relational Mapping) es una técnica para tratar a los datos de un sistema como objetos de un lenguaje de programación

# ORM

## Ejemplo - Modelos

```
from django.db import models
```

```
class Musician(models.Model):
```

```
    first_name = models.CharField(max_length=50)
```

```
    last_name = models.CharField(max_length=50)
```

```
    instrument = models.CharField(max_length=100)
```

```
class Album(models.Model):
```

```
    artist = models.ForeignKey(Musician,  
                               on_delete=models.CASCADE)
```

```
    name = models.CharField(max_length=100)
```

```
    release_date = models.DateField()
```

```
    num_stars = models.IntegerField()
```

# ORM

## Ejemplo - Consultas

Obtener todos los músicos:

```
>>> Musician.objects.all()
```

Obtener todos los músicos con nombre 'James':

```
>>> Musician.objects.filter(first_name='James')
```

Obtener todos los álbumes del artista con id 1:

```
>>> Musician.objects.get(id=1).album_set.all()
```

# ORM

Un ORM permite abstraerse de un sistema de bases de datos en particular

No es tan flexible como utilizar SQL, y no depende del desarrollador cómo se traducen las consultas

Nosotros instalamos la base de datos, pero el ORM se encarga de utilizarla



Es posible hacer una aplicación web  
sin saber SQL y sólo saber usar el  
ORM?



Sí, pero también es posible hacer una aplicación web en un editor en línea sin saber cómo funcionan

[Funciones](#)[Explora](#)[Suscripciones](#) [Plantillas](#)[Ayuda](#)[Entrar](#)

# Todo empieza con tu increíble página web

Wix combina la belleza con la más avanzada tecnología para crear tu increíble página web. Es fácil y gratis.

[Empieza ya](#)





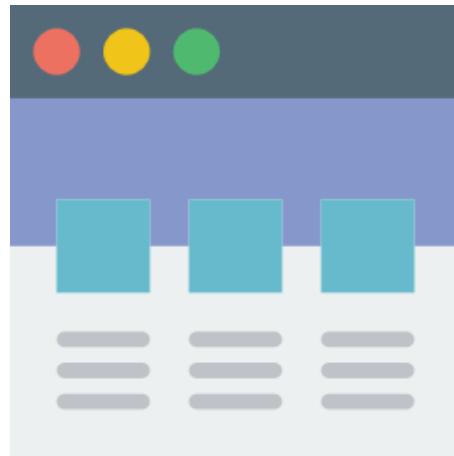
O también es posible armar un puente sin saber cómo hacerlo



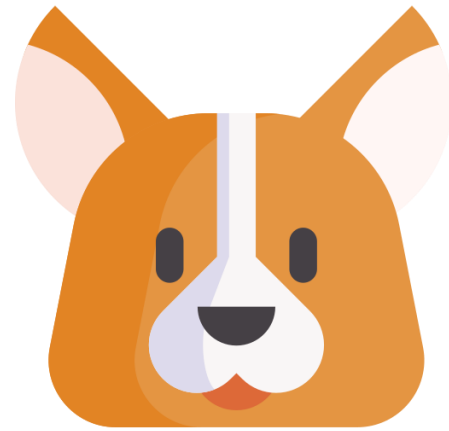


*“Es importante conocer cómo funciona SQL para construir aplicaciones de calidad”*

Profesor A, abril 2021



Y así, Don N. finalizó la  
aplicación web de su NebStore,  
pero...



Don N., su aplicación está  
funcionando demasiado lenta, quizás  
le está faltando...



**Big Data**

**Cloud Computing**

**Blockchain**

**Buzzword**

**Quantum Computing**

**Microservices**

**Deep Learning**

**Internet of Things**

**Data Science**





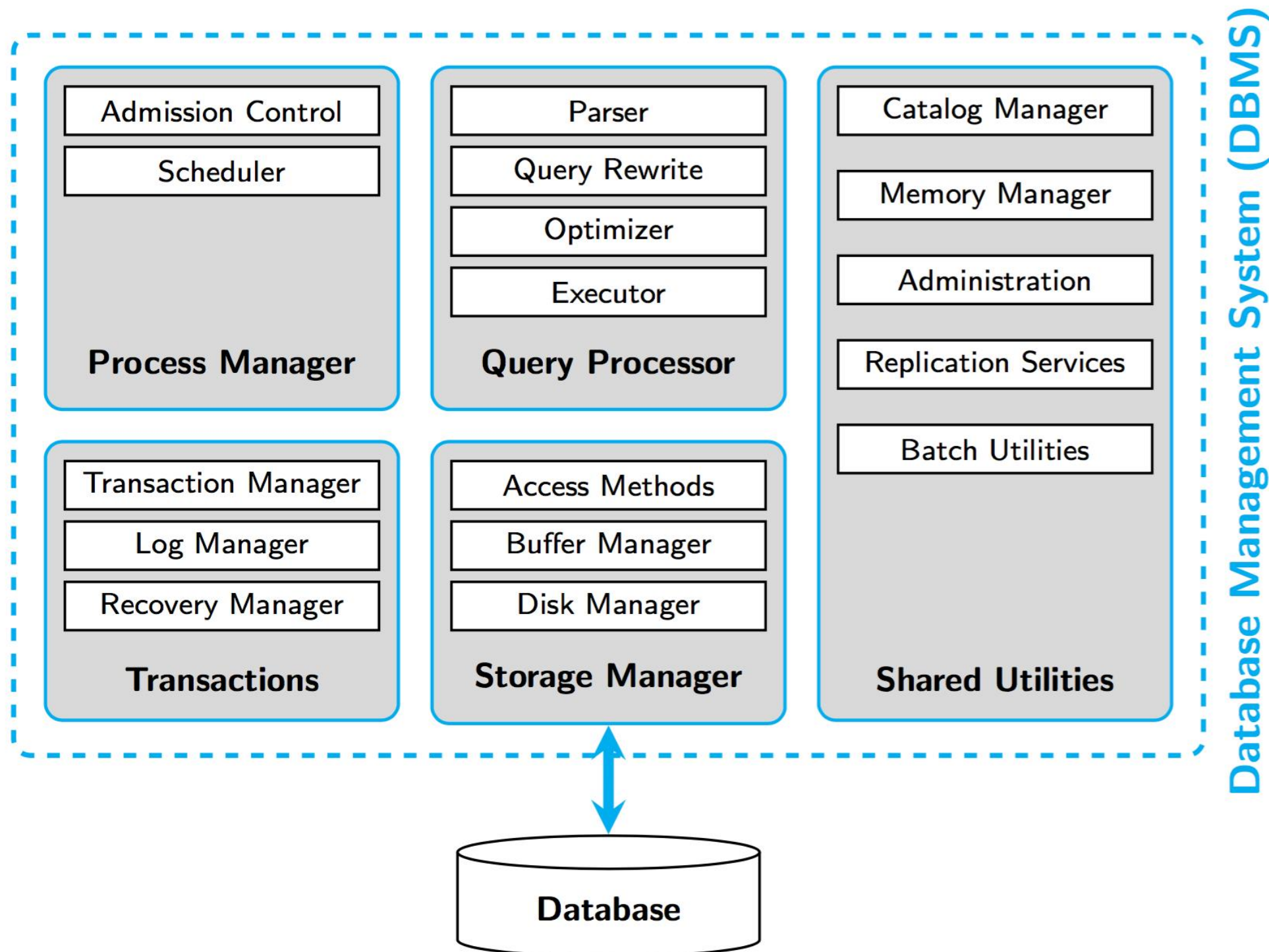
Profesor, me han contado que  
hay que usar NoSQL, porque  
funcionan mucho más rápido





A veces los problemas no son lo  
que parecen

# Cómo funciona una base de datos



# ¿Es SQL suficiente?

Un sistema puede funcionar lento porque:

- La base de datos está mal indexada
- Los datos están mal modelados
- Hay una cantidad de datos demasiado grande
- ...

# ¿Es SQL suficiente?

Aplicaciones grandes, como una red social de millones de usuarios, o un laboratorio de millones de registros, necesita técnicas diferentes

Pero una aplicación típica puede funcionar perfecto con una base de datos SQL!

# ¿Es SQL suficiente?

Además, para funcionar más rápido hay que transar algunas cosas, como la **consistencia** de los datos

En general, para asegurar la integridad de los datos, es recomendable usar un sistema SQL

# Tipos de DBMS

- Relacional
- Key - Value
- Column Store
- Graph Databases
- Document Store
- RDF
- In Memory Databases
- ...

# Tipos de DBMS

- Relacional

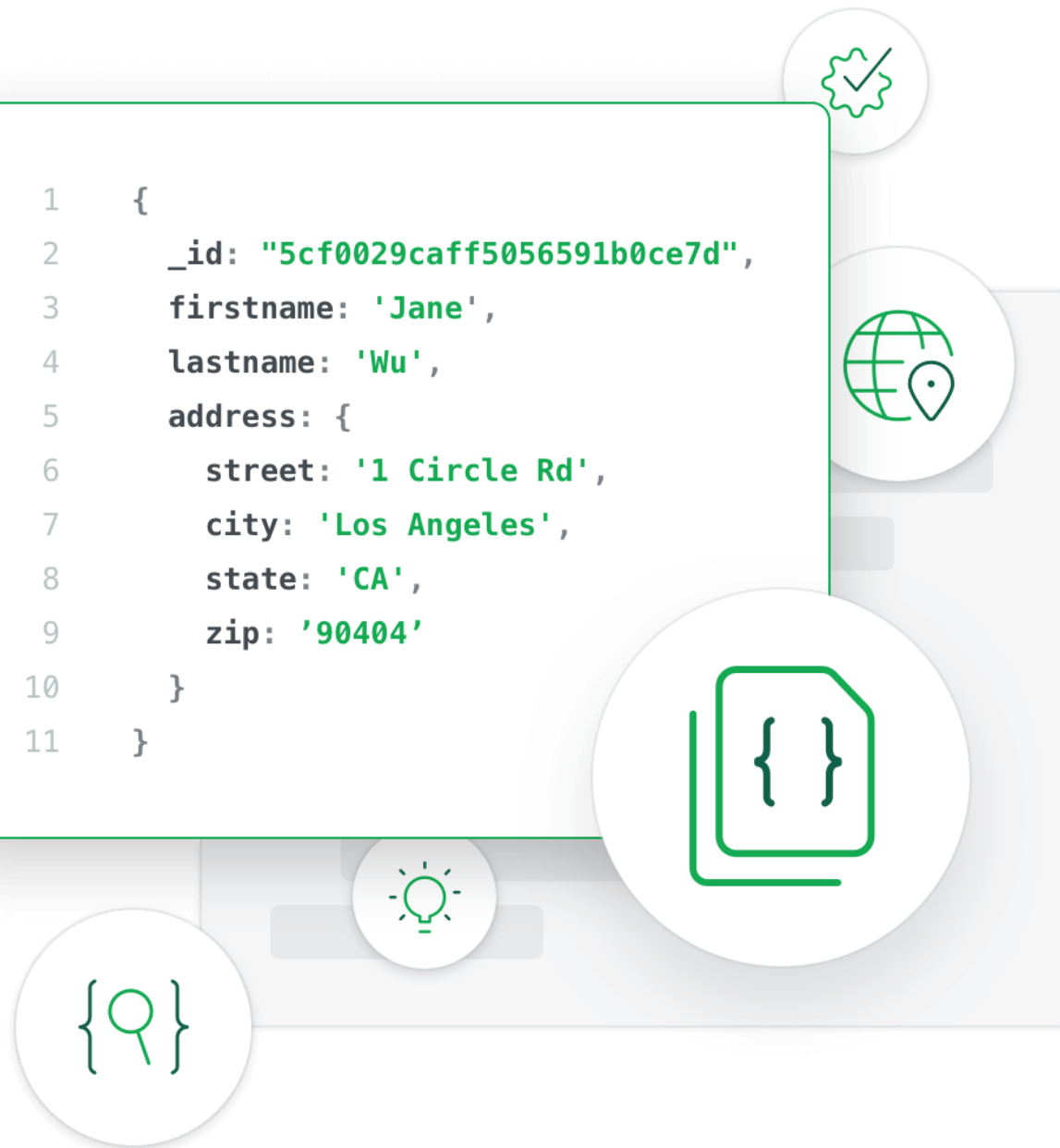


- SQLite
- PostgreSQL

# Tipos de DBMS

- Documentos  - MongoDB

```
1  {  
2    _id: "5cf0029caff5056591b0ce7d",  
3    firstname: 'Jane',  
4    lastname: 'Wu',  
5    address: {  
6      street: '1 Circle Rd',  
7      city: 'Los Angeles',  
8      state: 'CA',  
9      zip: '90404'  
10   }  
11 }
```





# Tipos de DBMS

- Documentos  - MongoDB

```
db.usuarios.find()
```

```
db.usuarios.find({uid: 2}, {}).pretty()
```

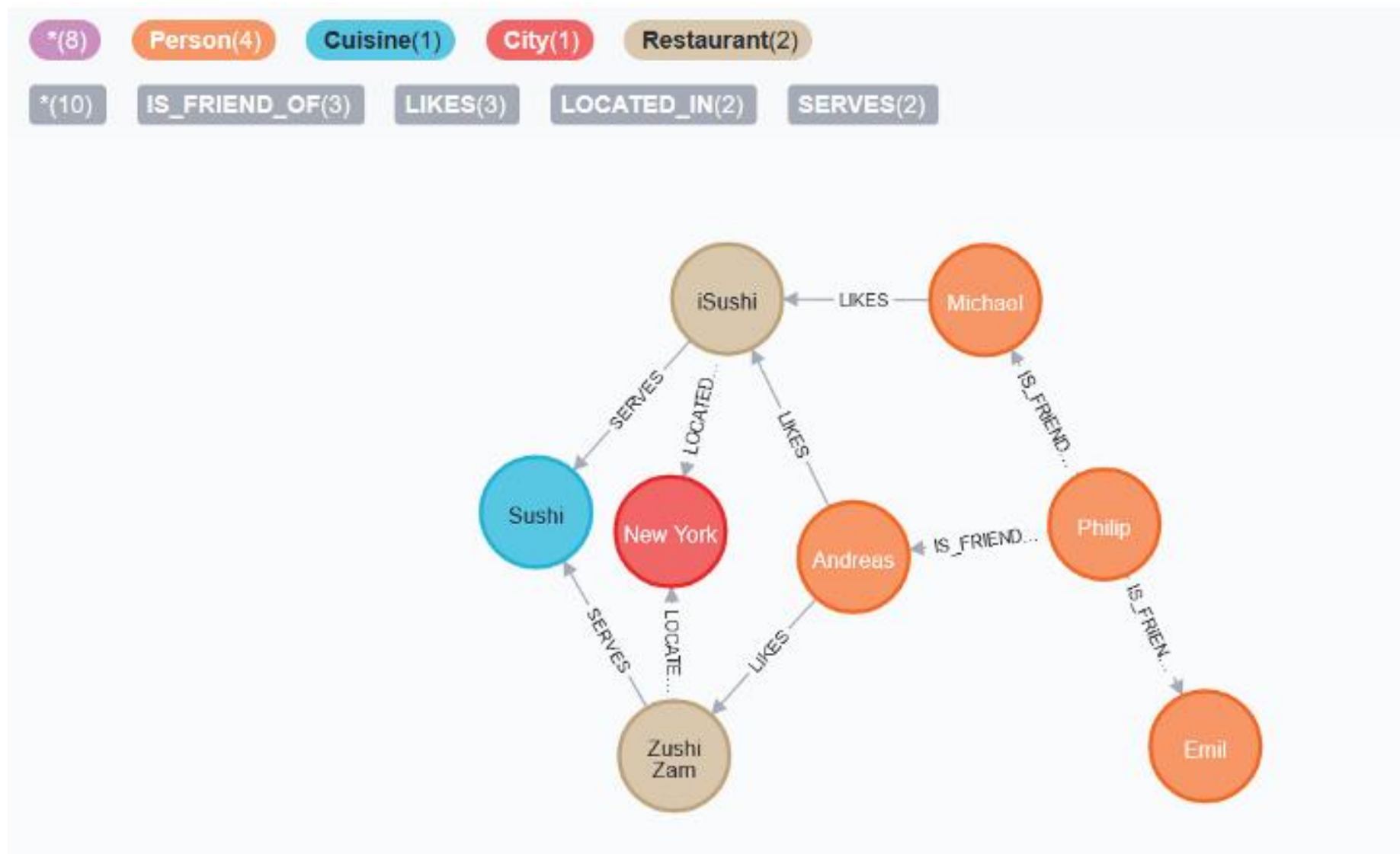
```
db.productos.find({price: {$gt: 300}}, {}).pretty()
```

# Tipos de DBMS

- Grafos



- Neo4j



# Tipos de DBMS

- Grafos



- Neo4j

```
MATCH (p:Person {name: 'Jennifer'})-[:LIKES]-(r:Restaurant)
RETURN r
```

# Tipos de DBMS

- Hay que usar cada sistema para lo que fue pensado
- No hay dejarse engañar por publicidad

# Tipos de DBMS

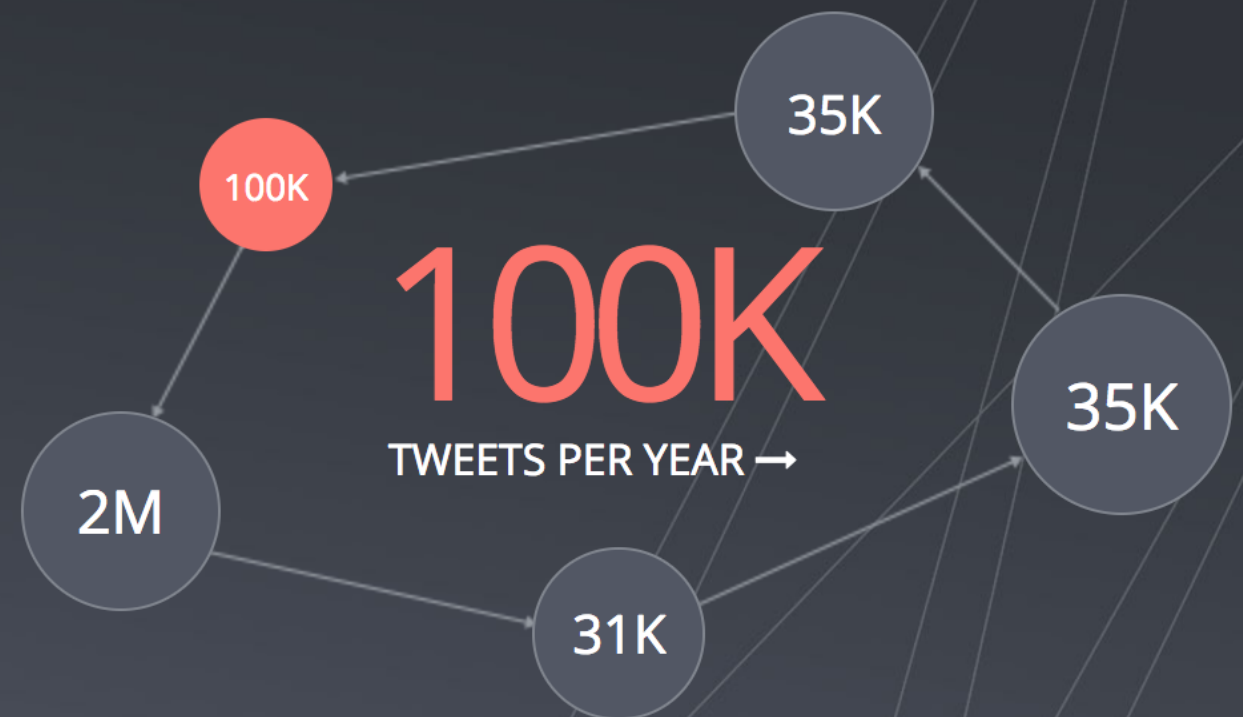
[PRODUCTS](#)[SOLUTIONS](#)[PARTNERS](#)[CUSTOMERS](#)[LEARN](#)[DEVELOPERS](#)

## (Neo4j) – [:LOVES] – (Developers)

World's leading graph database, with native graph storage and processing.

Property graph model and Cypher query language makes it easy to understand.

**Fast. Natural. Fun.**



# Tipos de DBMS



The world's most advanced  
open source database.



The world's most popular open source database

[MySQL.com](#)

[Downloads](#)

[Documentation](#)

[Developer Zone](#)

# Tipos de DBMS



[Home](#) [Download](#) [Documentation](#) [Community](#)

Manage massive amounts of data,  
fast, without losing sleep

# Tipos de DBMS



Why OrientDB? ▾

Products ▾

Learn ▾

Services ▾

Company ▾

Download

OrientDB - The World's First Distributed Multi-Model NoSQL Database with a Graph Database Engine



# Tipos de DBMS

## Do What You Could Never Do Before

What's been holding you back? MongoDB is the next-generation database that lets you create applications never before possible.

# Tipos de DBMS



[Home](#) [About](#) [Documentation](#) [Download](#) [License](#) [Support](#) [Purchase](#)

SQLite is a [self-contained](#), [high-reliability](#), [embedded](#), [full-featured](#), [public-domain](#), SQL database engine. SQLite is the [most used](#) database engine in the world. [More Info](#)

# Tipos de DBMS

[View Documentation](#)

[On Github](#)

## TITAN


### Distributed Graph Database

.zip

Titan is a scalable **graph database** optimized for storing and querying graphs containing hundreds of billions of vertices and edges distributed across a multi-machine cluster. Titan is a transactional database that can support **thousands of concurrent users** executing **complex graph traversals** in real time.

In addition, Titan provides the following features:

- Elastic and linear scalability for a growing data and user base.
- Data distribution and replication for performance and fault tolerance.
- Multi-datacenter high availability and hot backups.
- Support for **ACID** and **eventual consistency**.
- Support for various **storage backends**:
  - **Apache Cassandra**
  - **Apache HBase**
  - **Oracle BerkeleyDB**



# Tipos de DBMS

ORACLE®

Menu



Sign In ▾



Country/Region ▾



Call

## The Database's Future Is Autonomous

Powered by machine learning, Oracle Autonomous Database Cloud is 100 times more reliable, at half the cost.

Why autonomous is better →

Register: live briefing →



# Tipos de DBMS



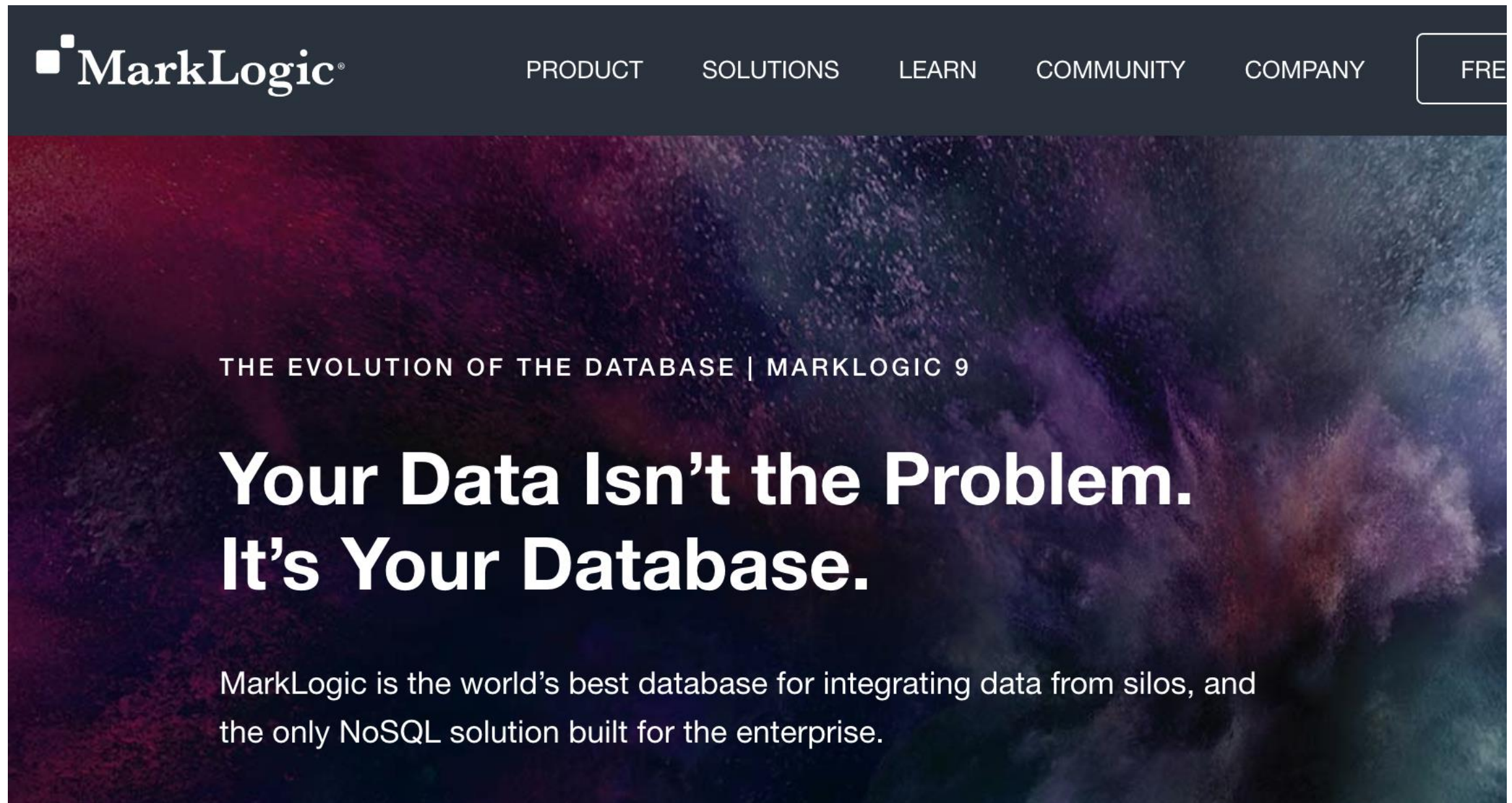
## Welcome to Apache HBase™

Apache HBase™ is the Hadoop database, a distributed, scalable, big data store.

Use Apache HBase™ when you need random, realtime read/write access to your Big Data. This project's goal is the hosting of very large tables -- billions of rows X millions of columns -- atop clusters of commodity hardware. Apache HBase is an open-source, distributed, versioned, non-relational database modeled after Google's [Bigtable: A Distributed Storage System for Structured Data](#) by Chang et al. Just as Bigtable leverages the distributed data storage provided by the Google File System, Apache HBase provides Bigtable-like capabilities on top of Hadoop and HDFS.



# Tipos de DBMS

The image is a screenshot of the MarkLogic website's hero section. It features a dark, textured background with shades of purple, blue, and red. At the top, there is a dark navigation bar with the MarkLogic logo on the left and several menu items in the center. On the right side of the navigation bar, there is a button labeled 'FREE'. Below the navigation bar, the text 'THE EVOLUTION OF THE DATABASE | MARKLOGIC 9' is displayed in a small, white, sans-serif font. The main headline, 'Your Data Isn't the Problem. It's Your Database.', is written in a large, bold, white, sans-serif font. Below the headline, a paragraph of text in a smaller, white, sans-serif font describes MarkLogic as the world's best database for integrating data from silos and the only NoSQL solution built for the enterprise.

**MarkLogic®** PRODUCT SOLUTIONS LEARN COMMUNITY COMPANY [FREE](#)

THE EVOLUTION OF THE DATABASE | MARKLOGIC 9

## Your Data Isn't the Problem. It's Your Database.

MarkLogic is the world's best database for integrating data from silos, and the only NoSQL solution built for the enterprise.

# Tipos de DBMS

<https://db-engines.com/en/ranking>

# Resumen

- Todos necesitamos manejar datos
- Salvo que queramos programar algoritmos más allá de nuestro problema, nos conviene usar un DBMS
- Hay que saber identificar el caso de uso de cada DBMS



# Links de interés

- Video de 20 min de SQL Básico:

[https://www.youtube.com/watch?v=K8bgmWoty0E&t=644s&ab\\_channel=Adri%C3%A1nSotoSu%C3%A1rez](https://www.youtube.com/watch?v=K8bgmWoty0E&t=644s&ab_channel=Adri%C3%A1nSotoSu%C3%A1rez)

- Video de 15 min de Agregación en SQL:

[https://www.youtube.com/watch?v=n-Ky1go7Alk&t=7s&ab\\_channel=Adri%C3%A1nSotoSu%C3%A1rez](https://www.youtube.com/watch?v=n-Ky1go7Alk&t=7s&ab_channel=Adri%C3%A1nSotoSu%C3%A1rez)

# Introducción a las bases de datos

Exploratorio en Computación  
Invitado: Valentina Álvarez