

| Curso “Especialización en IoT”

Módulo 6: Plataformas Clase 2

Docentes:
Sebastián Valerio Guerrero
Jonattan Silva Castillo



PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO

Bases de Datos



Qué es una **Base de Datos**?

Según Oracle:

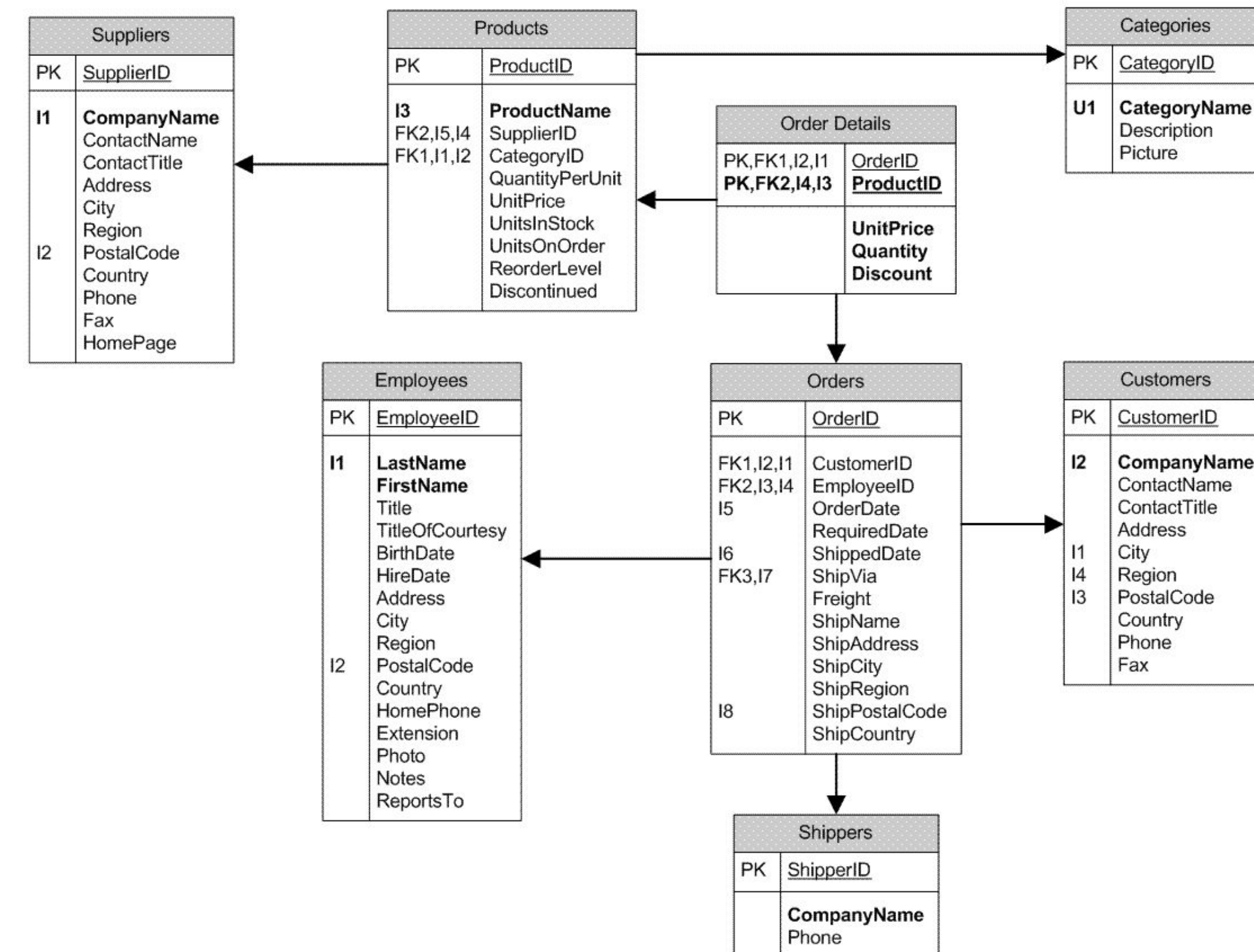
“Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un sistema de gestión de bases de datos (DBMS).”

En conjunto, los datos y el DBMS, junto con las aplicaciones asociadas a ellos, reciben el nombre de sistema de bases de datos, abreviado normalmente a simplemente base de datos.”



Modelo Relacional

- Almacenan y organizan puntos de datos con relaciones definidas para un acceso rápido.
- Usan tablas con datos organizados en filas (que contienen **entidades**) y columnas (que contienen **atributos** de entidad). Este proceso se conoce como **normalización**.
- Cada fila contiene un identificador único o una **clave** que une las tablas para establecer una relación.
- Altamente **estructurado** y representado mediante un esquema (lógico y físico).





Structured Query Language: SQL

Lenguaje de consulta estándar **SQL** es la interfaz estándar.

- Instrucciones SQL:
 - CREATE DATABASE se utiliza para crear una nueva base de datos vacía.
 - SELECT se utiliza cuando quieres leer (o seleccionar) tus datos.
 - INSERT se utiliza cuando quieres añadir (o insertar) nuevos datos.
 - UPDATE se utiliza cuando quieres cambiar (o actualizar) datos existentes.
 - DELETE se utiliza cuando quieres eliminar (o borrar) datos existentes.
 - TRUNCATE se utiliza cuando quieres vaciar (o borrar) todos los datos de la plantilla.

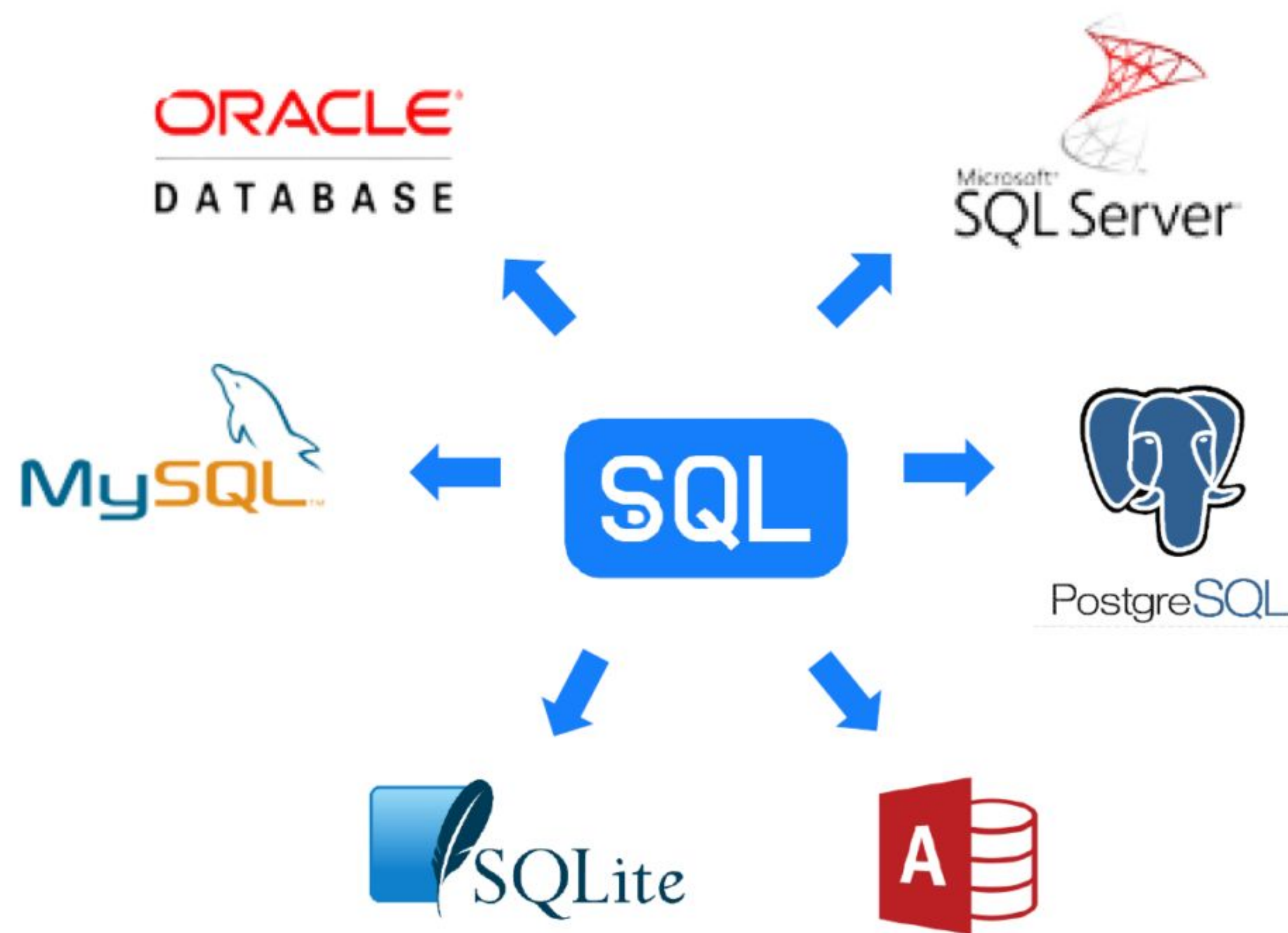
The screenshot shows the Microsoft SQL Server Management Studio interface. The main window displays a SQL script with the following commands:

```
CREATE TABLE SampleTable (ID INT, Col1 VARCHAR(100));  
INSERT INTO SampleTable (ID, Col1)  
VALUES (1, 'One'), (2, 'Two'), (3, 'Three');  
SELECT *  
FROM SampleTable;  
DROP TABLE SampleTable;
```

Below the script, the 'Results' tab is active, showing the output of the SELECT statement as a table:

	ID	Col1
1	1	One
2	2	Two
3	3	Three

Bases de Datos SQL

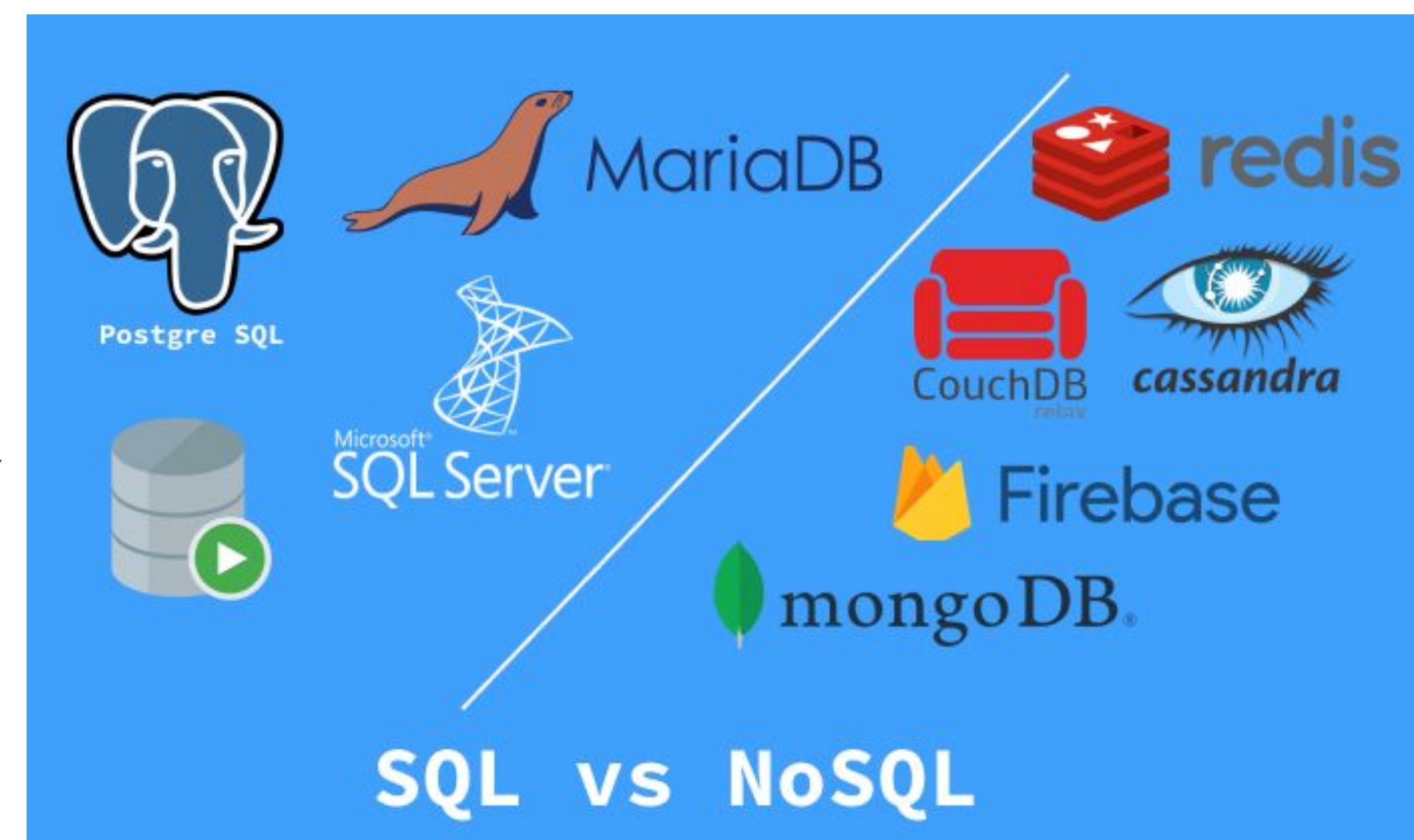


NoSQL

No SQL, Not only SQL. Hace referencia a lo **No** relacional o No estructurado.

Según AWS:

“Las bases de datos NoSQL están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento a escala.”





Ejemplo

Relacional

- Un registro de libros a menudo se normaliza y se almacena en **tablas separadas**, y las relaciones se definen mediante restricciones de claves primarias y externas. La tabla Libros tiene las columnas ISBN, Título del libro y Número de edición, la tabla Autores tiene las columnas IDAutor y Nombre de autor y, finalmente, la tabla Autor-ISBN tiene las columnas IDAutor e ISBN.

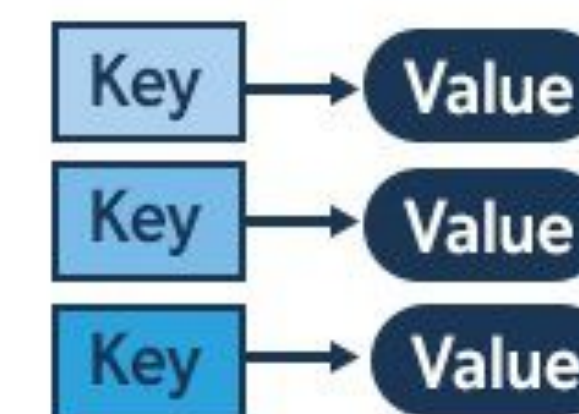
NoSQL

- El registro de un libro generalmente se almacena como un **documento JSON**. Para cada libro, el elemento, ISBN, Título del libro, Número de edición, Nombre autor y IDAutor se almacenan como atributos en un solo documento. En este modelo, los datos están optimizados para un desarrollo intuitivo y escalabilidad horizontal.

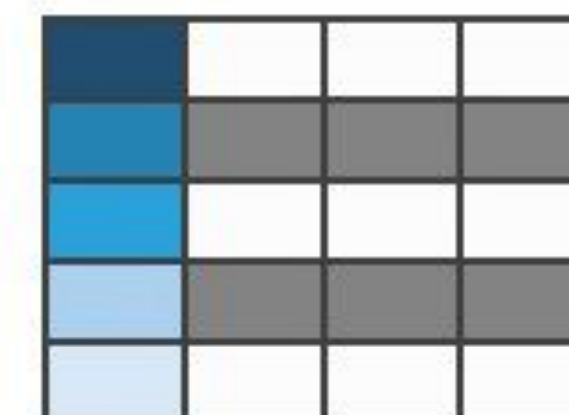
Tipos de NoSQL

- **Clave-valor:** altamente divisibles y permiten escalado horizontal a escalas que otros tipos de bases de datos no pueden alcanzar. Los casos de uso como juegos, tecnología publicitaria e **IoT** se prestan particularmente bien con este modelo.
- **Documentos:** los datos se representan a menudo como un objeto o un documento de tipo JSON porque es un modelo de datos eficiente e intuitivo para los desarrolladores.
- **Gráficos:** facilitan la creación y la ejecución de aplicaciones que funcionan con conjuntos de datos altamente conectados. Los casos de uso típicos para una base de datos de gráficos incluyen redes sociales, motores de recomendaciones, detección de fraude y gráficos de conocimiento.
- **Columnar:** creadas para tareas de consultas complejas y altamente analíticas. A diferencia de las bases de datos relacionales, las bases de datos en columnas almacenan sus datos por columnas, en lugar de por filas. Estas columnas se agrupan para formar subgrupos.

Key-Value



Column-Family

















Graph



Document



Bases de Datos NoSQL

Document Database	Graph Databases
 Couchbase  MarkLogic  mongoDB	 Neo4j  InfiniteGraph <small>The Distributed Graph Database</small>
Wide Column Stores	Key-Value Databases
 redis  amazon DynamoDB  AEROSPIKE  riak	 accumulo  HYPERTABLE INC  Cassandra  APACHE HBASE  Amazon SimpleDB



PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO

MongoDB

- Es una base de datos **NoSQL** orientada a documentos que apareció a mediados de la década de 2000.
- Se utiliza para almacenar volúmenes masivos de datos.
- No se basa en tablas y columnas. Los datos se almacenan como **colecciones y documentos**.
- Los documentos no tienen un esquema predefinido y los campos pueden añadirse a voluntad, con esto se facilita la representación de relaciones jerárquicas u otras estructuras complejas.
- Una característica importante es la **elasticidad** de sus entornos. Muchas empresas tienen clusters de más de 100 nodos para bases de datos que contienen millones de documentos.



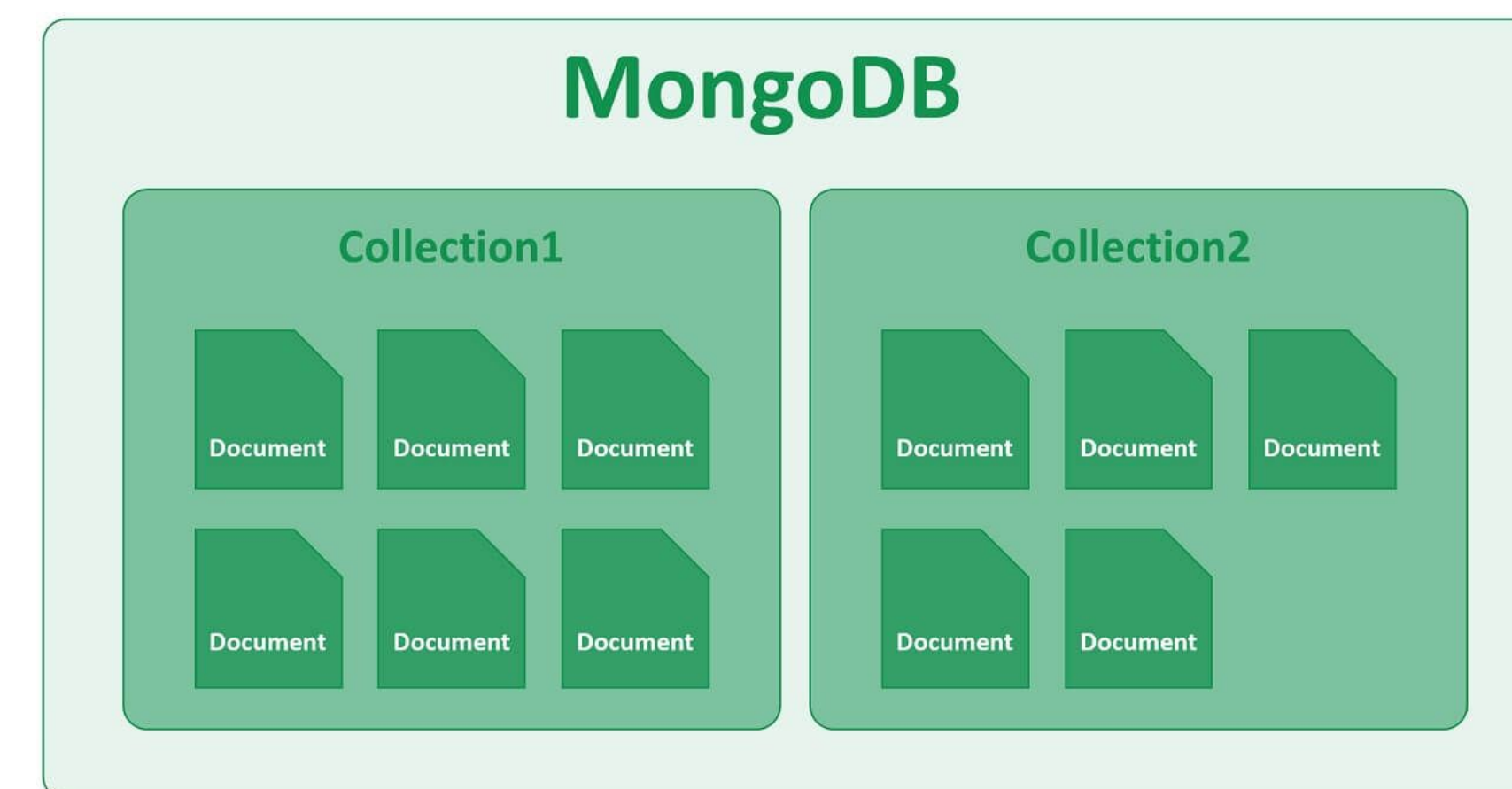
mongoDB

{ name: mongo, type: DB }



Conceptos de MongoDB

- Una base de datos es un **contenedor** de colecciones. Cada uno tiene su propio conjunto de archivos en el sistema de archivos. Un servidor MongoDB puede almacenar múltiples bases de datos.
- Una **colección** es un grupo de documentos de MongoDB. No tiene una estructura predefinida.
- Un **documento** es el equivalente a un registro en una base de datos tradicional. Se compone de campos de nombre y valor.
- «_id» es un campo **obligatorio** para cada documento. Representa un valor único y puede considerarse como la clave principal del documento para identificarlo dentro de la colección.



Conceptos de MongoDB

- **JSON** (JavaScript Object Notation) es un formato de texto plano para expresar datos estructurados.
- Está soportado por muchos lenguajes de programación.

```
{
  "_id": "5cf0029cafff5056591b0ce7d",
  "firstname": "Jane",
  "lastname": "Wu",
  "address": {
    "street": "1 Circle Rd",
    "city": "Los Angeles",
    "state": "CA",
    "zip": "90404"
  }
  "hobbies": ["surfing", "coding"]
}
```



Ejemplo 1

Habilitar nuestro entorno de trabajo:

- Vamos a crear una cuenta **MongoDB Cloud**:
 - <https://www.mongodb.com/cloud/atlas/register>
- Vamos a crear un **Cluster Gratis** de prueba.
- Agregar el paquete **pymongo** en PyCharm.





Cluster Compartido => Gratis

CLUSTERS > CREATE A SHARED CLUSTER

Create a Shared Cluster

Serverless

Dedicated

Shared

Cloud Provider & Region

GCP, Sao Paulo (southamerica-east1) ^

aws

Google Cloud

Azure

★ Recommended region ⓘ Ⓜ Dedicated tier region ⓘ

NORTH AMERICA / SOUTH AMERICA

EUROPE / MIDDLE EAST / AFRICA

AUSTRALIA

🇧🇷 Sao Paulo (southamerica-east1) ★

🇧🇪 Belgium (europe-west1) ★

🇦🇺 Sydney (australia-southeast1) ★ Ⓜ

🇺🇸 Iowa (us-central1) ★

🇵🇱 Warsaw (europe-central2) ★ Ⓜ

🇦🇺 Melbourne (australia-

FREE

Free forever! Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

Cancel

Create Cluster



Autenticación: usuario y contraseña

Security Quickstart

To access data stored in Atlas, you'll need to create users and set up network security controls. [Learn more about security setup](#)

1 How would you like to authenticate your connection?

Your first user will have permission to read and write any data in your project.

Username and Password

Certificate

Create a database user using a username and password. Users will be given the *read and write to any database* [privilege](#) by default. You can update these permissions and/or create additional users later. Ensure these credentials are different to your MongoDB Cloud username and password.

Username

mongoUser

Password 

mongoPassword2022

 Autogenerate Secure Password

 Copy


Create User




Acceso al cluster: Agregar mi IP actual

2 Where would you like to connect from?

Enable access for any network(s) that need to read and write data to your cluster.



My Local Environment
Use this to add network IP addresses to the IP Access List. This can be modified at any time.



Cloud Environment
Use this to configure network access between Atlas and your cloud or on-premise environment. Specifically, set up IP Access Lists, Network Peering, and Private Endpoints.

ADVANCED

Add entries to your IP Access List

Only an IP address you add to your Access List will be able to connect to your project's clusters.

IP Address

Description

Enter IP Address

Enter description

Add My Current IP Address

Add Entry



Conexión a DB: MongoDB Drivers

Connect to ClusterTest

✓ Setup connection security

Choose a connection method

Connect

Connect to your application



MongoDB Drivers

Connect your application to your cluster using MongoDB's native drivers



MongoDB Atlas Data API

Access your Atlas data over HTTPS via a fully managed and secure API



Access your data through tools



MongoDB Compass

Explore, modify, and visualize your data with MongoDB's GUI



MongoDB Shell

Interact with your cluster using MongoDB's interactive Javascript interface





Python: copiar la URI(Uniform Resource Identifier)

1 Select your driver and version

DRIVER

Python

VERSION

3.6 or later

2 Add your connection string into your application code

☐ Include full driver code example

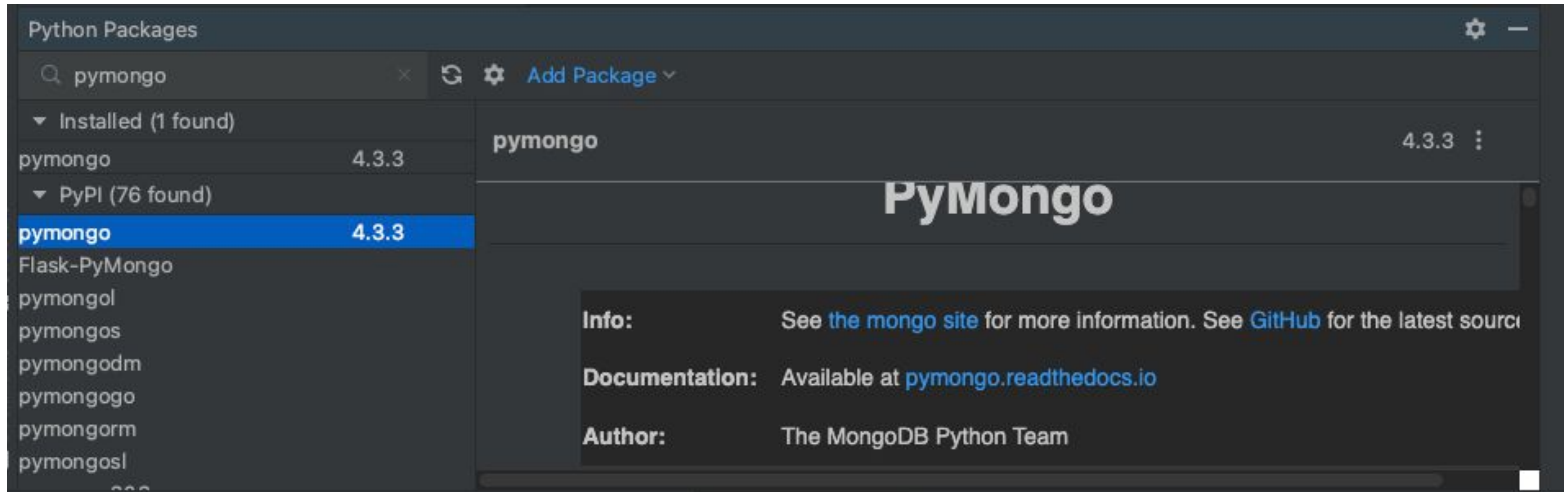
```
mongodb+srv://mongoUser:<password>@clustertest.xj97szq.mongodb.net/?  
retryWrites=true&w=majority
```



Replace **<password>** with the password for the **mongoUser** user. Ensure any option params are [URL encoded](#).



PyCharm: agregar pymongo





PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO

Ejemplo 1

Utilizando Python y la librería **pymongo**:

- Vamos a crear una aplicación que guarde documentos en una Mongo DB.





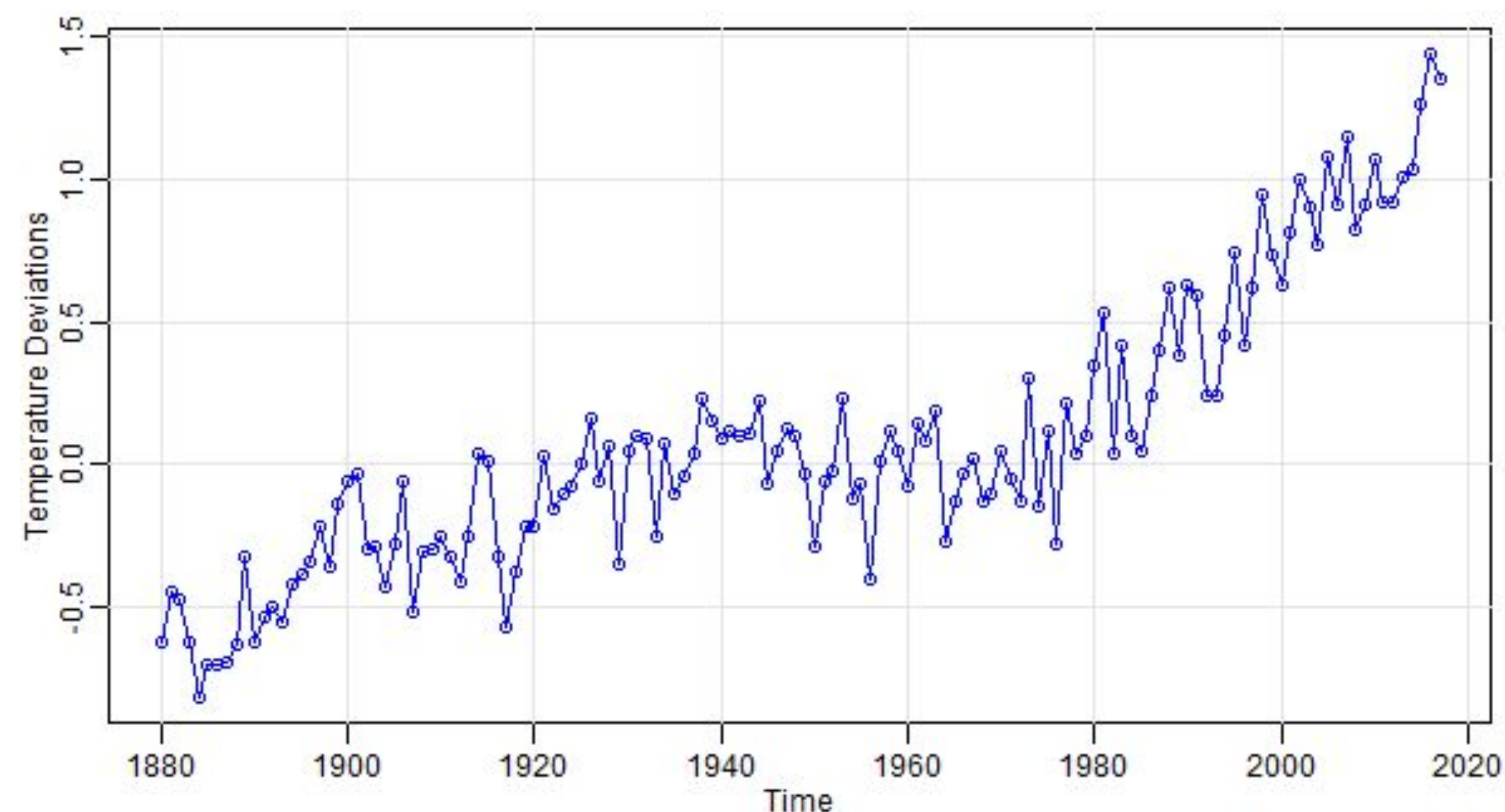
PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO



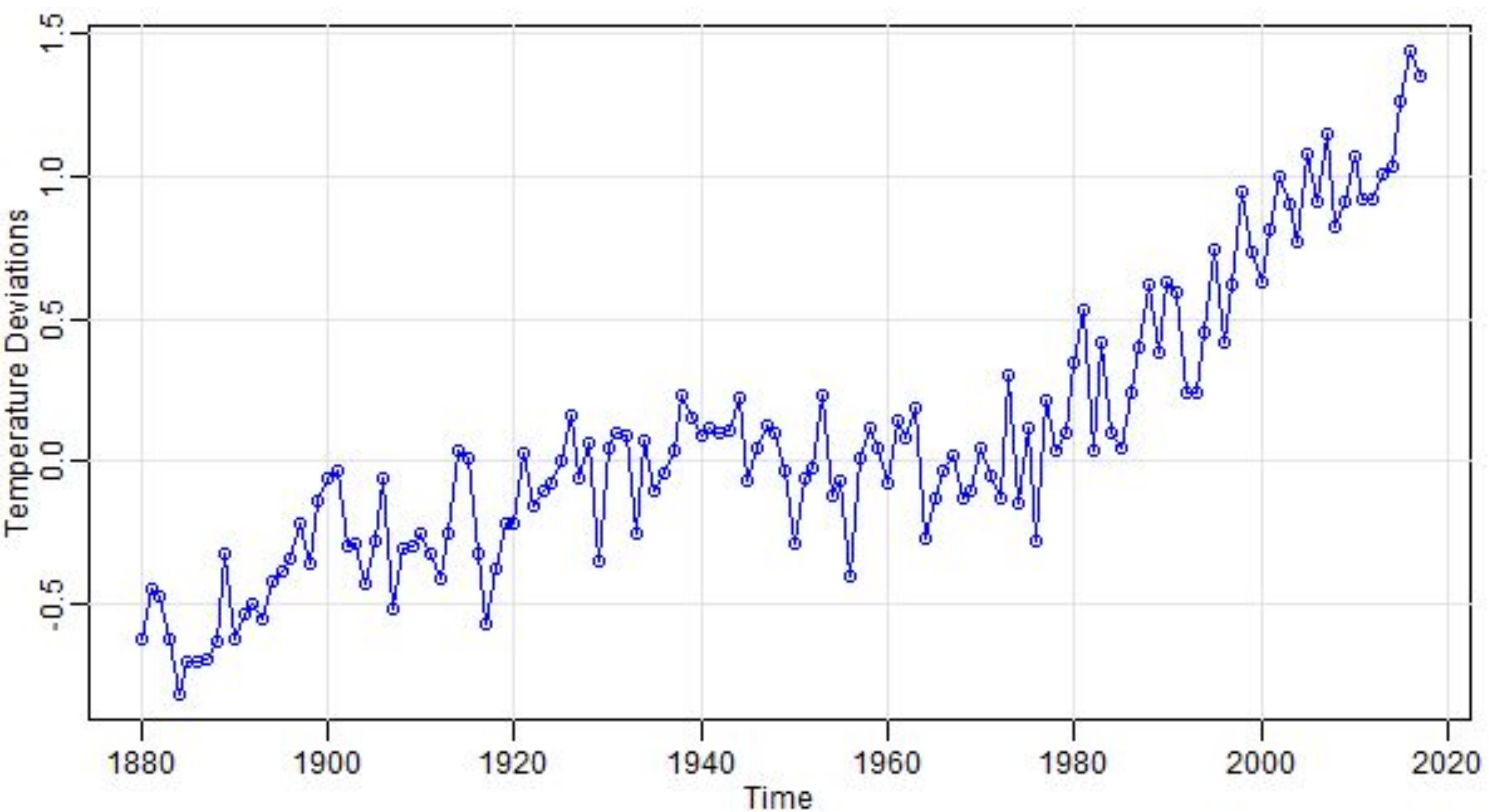
Series de tiempo

Los *datos de series de tiempo* son una colección de **observaciones** (comportamiento) para un **solo sujeto** (entidad) en **diferentes intervalos de tiempo**

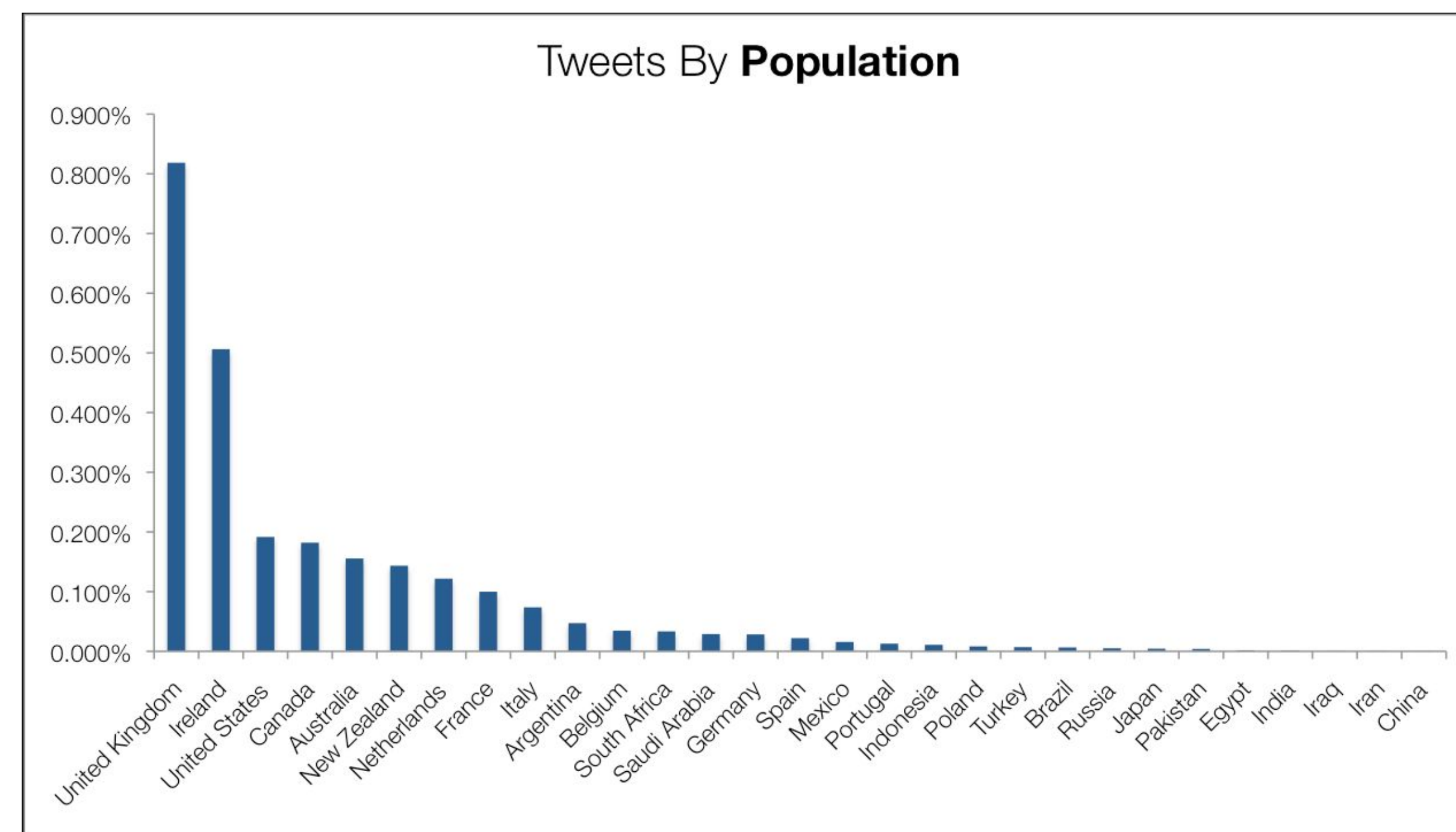
Por ejemplo: temperatura máxima, humedad y viento (los tres comportamientos) en la ciudad de Nueva York (entidad única) recopilados el primer día de cada año (múltiples intervalos de tiempo)



Datos de series de tiempo



Esto SI es una serie de Tiempo



Esto NO es una serie de Tiempo

Datos de series de tiempo

Los datos de series de tiempo se pueden clasificar en dos tipos:

- Mediciones recopiladas a intervalos de tiempo **regulares** (métricas)
- Mediciones recopiladas en intervalos de tiempo **irregulares** (eventos)

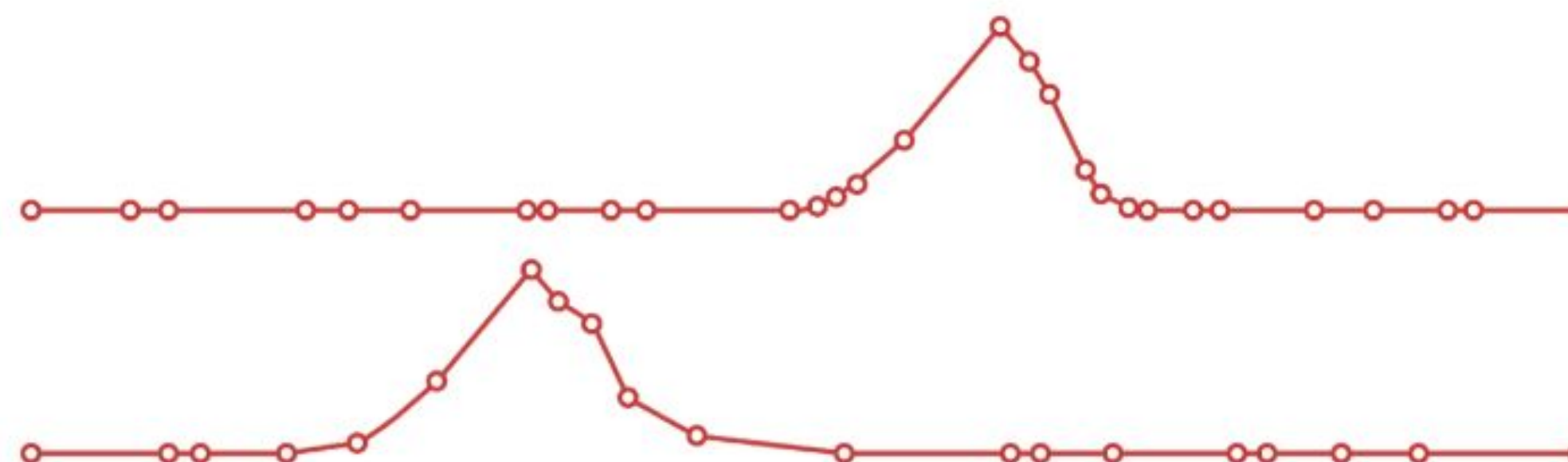
Metrics (Regular)

Measurements
gathered at regular
time intervals



Events (Irregular)

Measurements
gathered at irregular
time intervals





Datos de series de tiempo

Un punto de medida se denomina **Data Point** y está compuesto por:

- **Time**: almacena marcas de tiempo. Muestra la fecha y la hora, en formato [RFC3339](#) UTC, asociada con datos particulares.
- Los *Fields* se componen de **Field Key** y **Field Value**.
 - **Field Key**: String que indica a que corresponde cada valor obtenido.
 - **Field Value**: son los datos; pueden ser string, float, enteros o booleanos. Un field value **siempre** está asociado con una marca de tiempo

<https://www.epochconverter.com/>



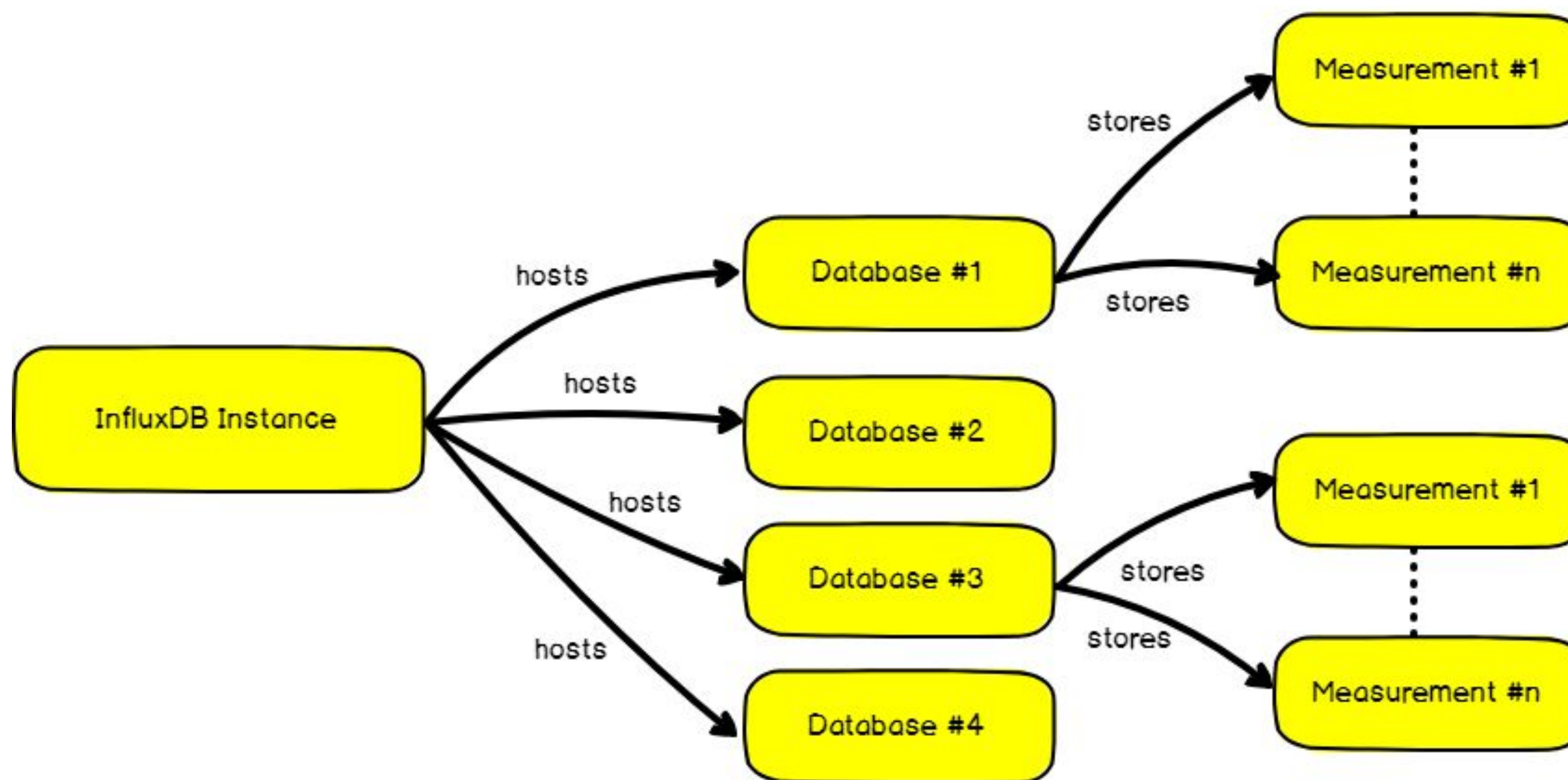
Datos de series de tiempo

Un punto de medida se denomina ***Data Point*** y está compuesto por:

- Los *Tags* se componen de ***Tag Key*** y ***Tag Value***
 - Los *tags* son opcionales. No se necesita tener *tags* en la estructura de datos, pero generalmente es una buena idea usarlas porque, a diferencia de los *fields*, los *tags* están indexadas.

InfluxDB

InfluxDB Internals



InfluxDB

Measurement #1

cpu_metrics

Data are aggregated over time!

Sensor	Temperature	time
Sensor 1	40	12/14 @ 15:16 pm
Sensor 1	38	13/14 @ 11:16 am
Sensor 1	41	12/11 @ 11:16 am
Sensor 1	38	11/11 @ 12:14 am



```
SELECT * FROM cpu_metrics WHERE temperature='40'
```



Tags vs Fields

Tags vs Fields

Measurement #1

cpu_metrics

Sensor	Temperature	time
Sensor 1	40	12/14 @ 15:16 pm
Sensor 1	38	13/14 @ 11:16 am
Sensor 1	41	12/11 @ 11:16 am
Sensor 1	38	11/11 @ 12:14 am

+

tag? field?

Location
USA
France
USA
Luxembourg

In a SQL world, would this column be indexed? yes → tag

no

Is this column giving additional information to a measurement? yes → tag

no

field



Ejemplo 2: Configuración InfluxDB

Para utilizar InfluxDB existen dos formas.

La primera es descargar la BBDD e instalarla en algún sistema operativo.

InfluxDB soporta los siguientes SO:

Platform

Docker Image

Docker Image

macOS (Homebrew)

macOS (tar)

Ubuntu & Debian

Ubuntu & Debian (ARM 64-bit)

RedHat & CentOS

RedHat & CentOS (ARM 64-bit)

Linux Binaries (64-bit)

Linux Binaries (ARM 64-bit)

Windows Binaries (64-bit) - using PowerShell



Ejemplo 2: Configuración InfluxDB

Para utilizar InfluxDB existen dos formas.

La segunda forma es utilizar InfluxDB Cloud. En esta actividad utilizaremos esta opción.

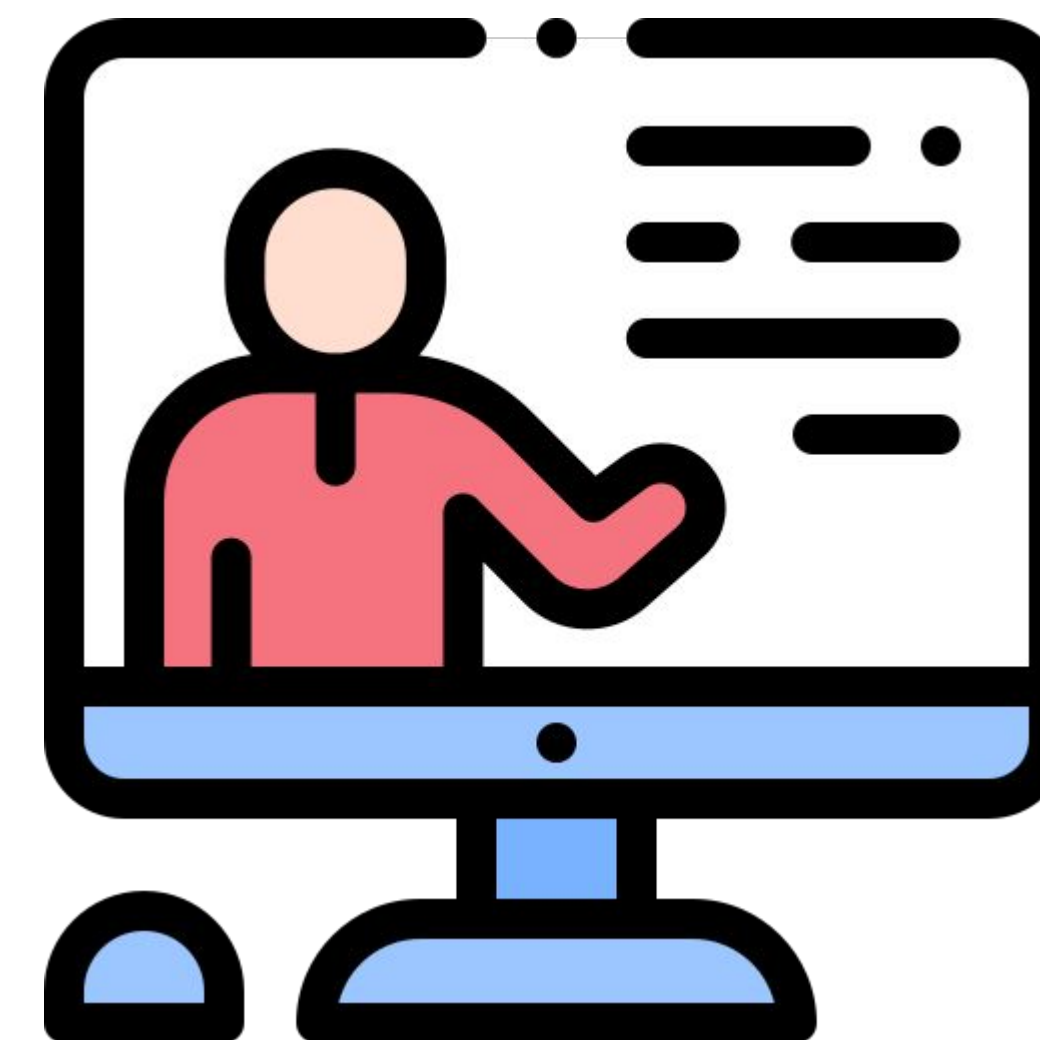
Para lo anterior, vaya a “<https://www.influxdata.com/>” y haga click en **Log In**. Escoga la opción **InfluxDB Cloud 2.0** y acceda con sus credenciales de Google.



Ejemplo 2

Utilizando Python y la librería **influxdb-client**:

- Vamos a crear una aplicación que guarde datos en una Influx DB.





PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO

Trabajo Grupal



Grupos de trabajo P3

Grupo	Nombre	Apellido Paterno	Apellido Materno	Sala
1	CHANTAL ALEJANDRA	SALCEDO	BURGOS	1
	SERGIO ISMAEL	MORAN	VALDES	
	BRIAN LANTANO	GARCIA	SIMONTTI	
1	VALENTIN ANTONIO	SEPULVEDA	PIZARRO	2
	MIGUEL ANTONIO	SOTO	VEGA	
	GABRIEL ALEJANDRO BENITO	SAAVEDRA	RAMIREZ	
	ANDRES ESTEBAN	ORELLANA	RIQUELME	
2	ARLETTE IVONNE	ARAYA	ARAYA	3
	HECTOR EDUARDO	MEZA	SALGADO	
	ALVARO RODRIGO	HENRIQUEZ	PACHECO	
2	VICTOR	REYES	DURAN	4
	JULIO CESAR	ESPARZA	HENRIQUEZ	
	RENATO HUMBERTO	VARGAS	DIAZ	
	ALEX EDUARDO	VEGA	PROVOSTE	
3	GUSTAVO ADOLFO	CORDOVA	CATALAN	5
	CHRISTIAN DANIEL	SILVA	BARRERA	
	ITALO IVAN	IBACACHE	VARGAS	
3	ADRIAN MICHEL	TAPIA	COLL	6
	VICTOR ANDRES	ESPINOZA	MONARES	
	JOSE LUIS	MENDEZ	VASQUEZ	

Grupo	Nombre	Apellido Paterno	Apellido Materno	Sala
4	FELIPE EDUARDO ALEXIS	ARCOS	ZAMORA	7
	DAVID ALEXANDRO	GIRALT	CHACANA	
	RICARDO ANTONIO	PEZO	FRIAS	
4	ROGELIO EDGARDO	MORENO	ARANEDA	8
	MARIO ERNESTO	VILLANUEVA	GUTIERREZ	
	RUBEN ANTONIO	CONTRERAS	OTAIZA	
5	JAVIER ANTONIO	SILVA	RUIZ	9
	CARLOS IGNACIO	VALDERRAMA	GUERRA	
	CARLOS ALBERTO	ASTUDILLO	VASQUEZ	
5	JOSUE ALEXANDER	ISTURIZ	PEREZ	10
	ADOLFO ALBERTO	BRAVO	SILVA	
	KENY OSCAR	CORTES	GONZALEZ	
6	CLAUDIO ANDRES	JERALDO	PASTEN	11
	JOSE DANIEL	CASTRO	CERDA	
	HUMBERTO ALEJANDRO	CHAVEZ	PINO	
6	PATRICIO ERNESTO	MORENO	TORRES	12
	ARIEL FRANCISCO	MARTINEZ	GAVILAN	
	LUIS HUGO	BONILLA	YANEZ	

Ejercicio 1

En Salas de Grupo:

- En el **dispositivo** Wemos deben publicar en un mensaje **MQTT** la temperatura ambiente con el sensor DS18B20 cada 15 segundos.
- En **Python** deben crear una aplicación tipo “**microservicio ETL**” que **extraiga** los mensajes desde el servicio MQTT, **transforme** los mensajes recibidos y los **almacene** como datos en InfluxDB.
- Para MQTT, vamos a utilizar estos parámetros:
 - **broker** = "test.mosquitto.org"
 - **port** = 1883
 - **topic** = "pucv/iot/m6/pX/gY" *#reemplazar X e Y por paralelo y sala correspondiente*



Feedback



PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO





Cierre

En esta clase vimos:

1. Distintos servicios de bases de datos NoSQL.
2. Aprendimos, ejercitando en grupo, cómo inyectar datos usando librerías de Python.