Módulo 4.1: Introducción a Bases de datos

MSc. Carlos Cedeño

En un sistema de IoT, los dispositivos (sensores, actuadores, etc.) generan un volumen masivo de datos de forma continua. Gestionar, almacenar y analizar esta información es imposible sin una base de datos robusta y bien elegida. Ella es el cerebro central que permite dar sentido a todo lo que los dispositivos recolectan.

1. Importancia de las Bases de Datos en IoT

La gestión de datos es el corazón de cualquier aplicación de IoT. Sin una base de datos, los datos generados por los sensores simplemente se perderían. Su importancia radica en:

- **Almacenamiento Persistente:** Guardan de forma segura y permanente los flujos de datos (telemetría) que envían los dispositivos.
- **Análisis de Datos:** Permiten realizar consultas complejas sobre los datos históricos para identificar patrones, detectar anomalías o predecir fallos. Por ejemplo, analizar la vibración de un motor a lo largo del tiempo para predecir cuándo necesitará mantenimiento.

- **Toma de Decisiones en Tiempo Real:** Facilitan el acceso rápido a la información para que los sistemas puedan tomar decisiones inmediatas, como apagar una máquina si la temperatura excede un umbral crítico.
- **Gestión de Dispositivos:** Almacenan información sobre los propios dispositivos, como su estado (online/offline), configuración, ubicación y versión de firmware.
- **Escalabilidad:** Un buen sistema de base de datos puede crecer para manejar miles o millones de dispositivos y los petabytes de datos que estos generan.

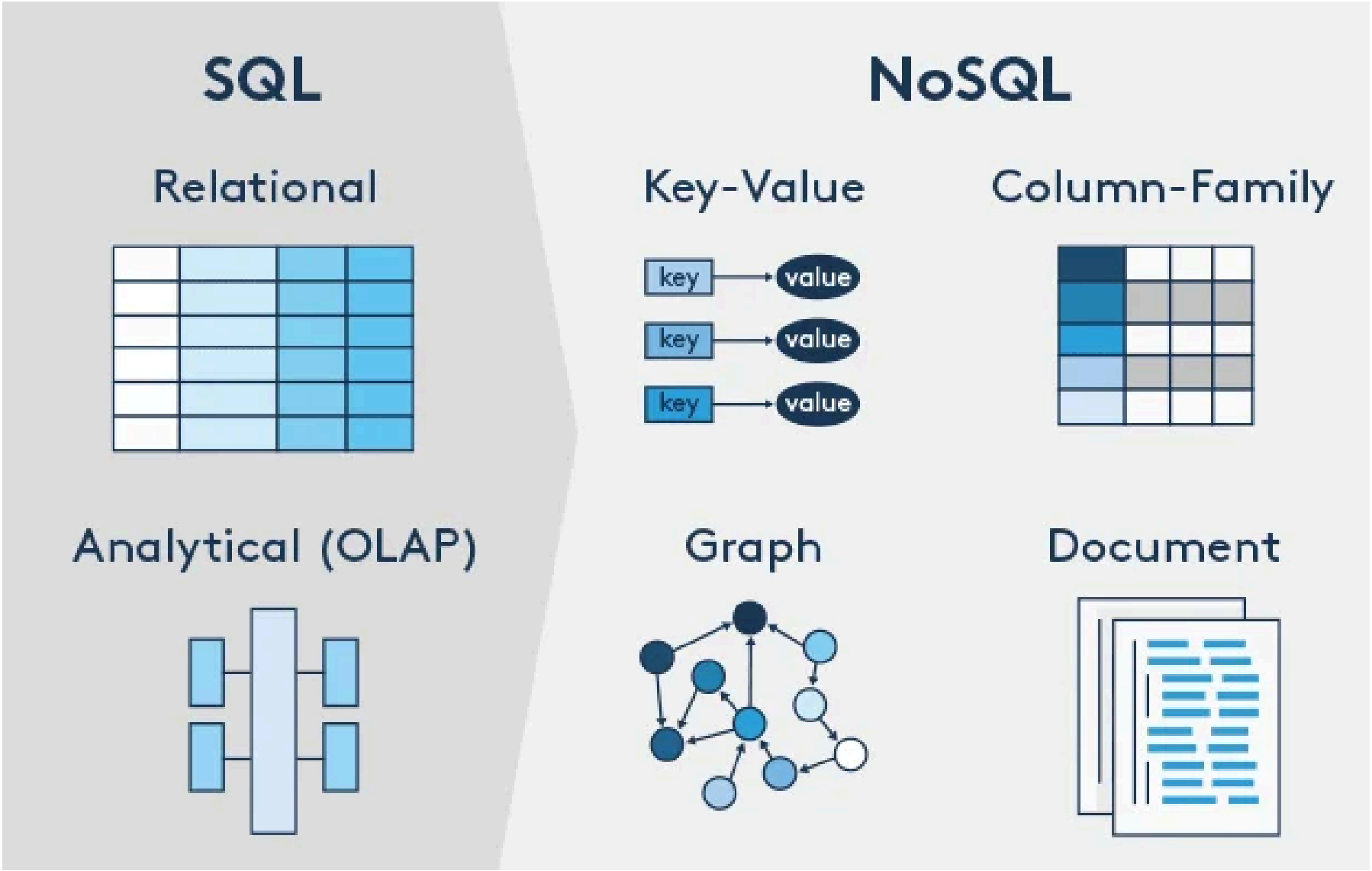
2. ¿Cuándo Usar una Base de Datos en IoT?

Prácticamente **siempre** que desarrolles una solución de IoT necesitarás una base de datos. Específicamente, se vuelve indispensable cuando necesitas:

- **Guardar un historial** de las mediciones de los sensores (temperatura, humedad, posición GPS, etc.).
- **Analizar tendencias** a lo largo del tiempo.
- **Crear dashboards y visualizaciones** para que un usuario pueda monitorear el sistema.
- **Entrenar modelos de Machine Learning** con datos históricos.
- **Gestionar una flota de dispositivos** y conocer el estado de cada uno.
- **Ofrecer una API** para que otras aplicaciones consuman los datos recolectados.

3. Tipos de Bases de Datos

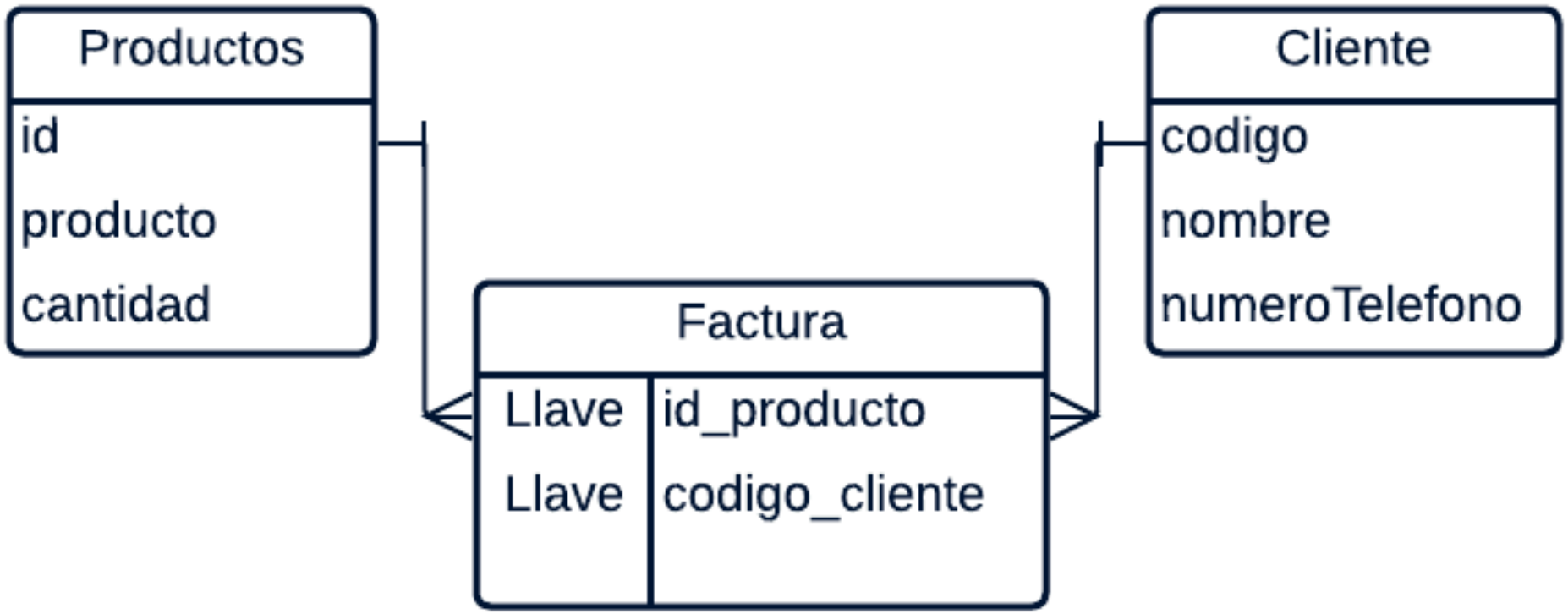
Las bases de datos se dividen principalmente en dos grandes familias: **Relacionales (SQL)** y **No Relacionales (NoSQL)**. La elección entre una y otra es una de las decisiones de arquitectura más importantes en un proyecto de IoT.


SQL
NoSQL
Relational
Key-Value
key
value
key
value
key
value
Column-Family
Analytical (OLAP)
Graph
Document

4. Bases de Datos Relacionales (SQL)

Estas son las bases de datos "clásicas". Organizan los datos en **tablas** con filas y columnas, con un **esquema estricto y predefinido**. Las relaciones entre tablas se establecen mediante claves (Primary Keys y Foreign Keys).

El lenguaje estándar para interactuar con ellas es **SQL (Structured Query Language)**. Garantizan la consistencia de los datos a través de las propiedades **ACID** (Atomicidad, Consistencia, Aislamiento y Durabilidad).


Productos
id
producto
cantidad
Cliente
codigo
nombre
numeroTelefono
Factura
Llave
id_producto
Llave
codigo_cliente

5. Bases de Datos No Relacionales (NoSQL)

Las bases de datos NoSQL surgieron para resolver las limitaciones de las relacionales frente a los desafíos del Big Data, como los que presenta IoT: grandes volúmenes de datos, alta velocidad y variedad de formatos.

No usan tablas, sino diferentes modelos de almacenamiento, y tienen un **esquema dinámico o flexible**. Esto las hace ideales para datos no estructurados o semi-estructurados (como JSON). Suelen priorizar la **disponibilidad y la escalabilidad horizontal** (añadiendo más servidores) sobre la consistencia estricta (modelo BASE: Basically Available, Soft state, Eventually consistent).

student_id	age	score
1	12	77
2	12	68
3	11	75


```
[
 {
  "student_id":1,
  "age":12,
  "score":77
 },
 {
  "student_id":2,
  "age":12,
  "score":68
 },
 {
  "student_id":3,
  "age":11,
  "score":75
 }
]
```

6. Comparativa: Relacional (SQL) vs. No Relacional (NoSQL)

Característica	Bases de Datos Relacionales (SQL)	Bases de Datos No Relacionales (NoSQL)
Modelo de Datos	Tablas con filas y columnas. Estructura rígida.	Documentos, clave-valor, grafos, columnas. Flexible.
Esquema	**Estricto y predefinido.** Se debe definir antes de insertar datos.	**Dinámico.** Se pueden añadir campos sobre la marcha.
Escalabilidad	**Vertical.** Se escala aumentando la potencia de un solo servidor (CPU, RAM).	**Horizontal.** Se escala añadiendo más servidores al clúster.

7. ¿Cuándo Usar Una u Otra en IoT?

La mejor estrategia en IoT a menudo es **híbrida**: usar el tipo de base de datos correcto para el trabajo correcto. No es una guerra de SQL vs. NoSQL, sino una colaboración.

Usa una Base de Datos Relacional (SQL) cuando:

- Los datos son **muy estructurados** y el esquema no va a cambiar con frecuencia.
- La **integridad y consistencia** de los datos son críticas (por ejemplo, para facturación, gestión de usuarios o configuraciones críticas de dispositivos).
- Necesitas realizar **consultas complejas con `JOIN`s`** entre diferentes tablas.
- **Ejemplo de uso en IoT:** Almacenar la información de los clientes, los dispositivos que han comprado, las reglas de configuración de cada dispositivo y las relaciones entre ellos.

Usa una Base de Datos No Relacional (NoSQL) cuando:

- Necesitas ingerir un **volumen masivo de datos a alta velocidad** (miles de escrituras por segundo). Este es el caso típico de la telemetría de sensores.
- Los datos son **semi-estructurados o no estructurados** (por ejemplo, un sensor puede enviar 3 campos y otro puede enviar 5).
- La **escalabilidad horizontal** es una prioridad. Sabes que vas a pasar de 100 a 1,000,000 de dispositivos.
- La **disponibilidad** es más crítica que la consistencia inmediata.
- **Ejemplo de uso en IoT (específicamente una de Series Temporales como InfluxDB):** Almacenar cada lectura de temperatura, humedad y CO_2 que un sensor envía cada 5 segundos. Luego, poder consultar rápidamente: "¿Cuál fue la temperatura promedio de ayer entre las 2 PM y las 4 PM?".