



## Ciclo 2

# Semana 6

## *Persistencia en Bases de Datos*

### Lectura 4 - ¿Cuándo implementar bases de datos SQL Y NoSQL?



## | ¿Cuándo implementar bases de datos SQL Y NoSQL?

Para ilustrar una de las diferencias entre bases de datos SQL y NoSQL, vamos a dar un ejemplo. Imaginemos por un momento una ciudad como Medellín (Colombia), donde todas las personas hablan el mismo idioma, por tanto, es la única forma de que todos los residentes se comuniquen e interactúen. Si se cambia ese idioma se perjudican todos los residentes.

Las bases de datos relacionales utilizan un lenguaje de consulta estructurado para la manipulación de datos, estas se conforman por filas, columnas y registros y se almacenan por tablas. Para manipular los datos en SQL, se requiere primero determinar la estructura de estos, si se cambia la estructura de uno de los datos, puede perjudicar todo el sistema, ya que las tablas están relacionadas.

Ahora imaginemos otra ciudad como Bogotá (Colombia) y pensemos hipotéticamente que en cada uno de los hogares se habla un idioma diferente, todos interactúan distinto y no hay entendimiento entre todos, pero nadie afecta a nadie. Las bases de datos no relacionales tienen un esquema dinámico, no se requiere la estructura de los datos para su manipulación. Los datos se pueden almacenar de cualquier manera, columnas, documentos, gráficos, etc, y cada documento puede tener su propia estructura, sin afectar los demás, puede agregar más campos a medida que se avanza. Están conformadas por documentos, campos y datos del documento, además, se almacenan por colecciones.



## ¿Cuándo utilizar SQL o NoSQL?

### SQL

Cuando el volumen de mis datos no crece o lo hace poco a poco.

Cuando las necesidades de proceso se pueden asumir en un sólo servidor.

Cuando no tenemos picos de uso del sistema por parte de los usuarios más allá de los previstos.

### NoSQL

Cuando el volumen de mis datos crece muy rápidamente en momentos puntuales.

Cuando las necesidades de proceso no se pueden prever.

Cuando tenemos picos de uso del sistema por parte de los usuarios en múltiples ocasiones.

Tanto SQL como NoSQL son tipos de bases de datos recomendadas para utilizar a la hora de comenzar con tus proyectos, cada una de ellas con ventajas y desventajas.

Por ejemplo, los sistemas contables, o de inventario, son sistemas que requieren transacciones de varias filas, para este tipo de trabajos la mejor opción son las bases de datos SQL (MySQL). Si los sistemas son de gestión de contenido, aplicaciones móviles, sistemas de análisis en tiempo real, bases de datos con un crecimiento rápido, con un esquema descentralizado, la mejor opción son bases de datos NoSQL (MongoDB).

Algunas de las ventajas de SQL: mayor soporte y más variedad de herramientas debido a que lleva más tiempo en el mercado, es útil para manejar y obtener los datos, permite agregar otros servidores de SQL, por ejemplo, una persona puede acceder a la base de datos de otra.

Como desventajas de SQL están: No es flexible (antes de ingresar los objetos, deben estar correctamente validados), mientras más compleja sea la base de datos, requiere mayor procesamiento y eso se puede ver reflejado en el rendimiento y consumo de recursos.

## Semana 6 | Persistencia en Bases de Datos

Ahora, conozcamos algunas de las ventajas de las bases de datos NoSQL: permite una alta escalabilidad (ayuda a reducir la carga de trabajo), flexible a diferentes tipos de datos, los datos deben cumplir con el tipo de dato definido, y algunas desventajas: la integridad de los datos se afecta por el poco soporte, menos seguridad al ejecutar consultas, no existe estandarización, en la mayoría de los casos son poco compatibles con las bases de datos SQL, casi todo su mantenimiento se debe realizar por consola debido a que existen pocas herramientas.

Muchas personas piensan que deben migrar sus bases de datos a tecnologías NoSQL ya que es lo “nuevo”, pero es un pensamiento errado. NOSQL no es el reemplazo de SQL, simplemente es un modelo diferente que ofrece ventajas y soluciones a problemas que poseen las bases de datos relacionales.