### **1. ¿Cuál es la principal diferencia entre SQL y NoSQL?**

La principal diferencia está en la forma en la que se organizan los datos. En SQL los datos se organizan en tablas con columnas y filas, lo que requiere de una estructura fija antes de guardar los datos. En NoSQL, los datos son guardados en colecciones y documentos, lo que permite tener una estructura más flexible, ideal para datos cambiantes o una gran cantidad de datos.

### **2.Menciona dos ventajas de usar una base de datos NoSQL sobre una SQL.**

**-Estructura flexible:** No hay que definir un esquema fijo, por lo que se puede almacenar distintos tipos de datos sin tener que modificar el esquema, lo que hace a noSQL ideal para modelos de datos que cambian con frecuencia.

**-Escalabilidad:** NoSQL está diseñado para funcionar en sistemas distribuidos, lo que permite la posibilidad de escalar horizontalmente de forma sencilla. Mientras la app o web crece, en lugar de mejorar el hardware de un solo servidor como lo haría SQL, el escalado horizontal de NoSQL permite agregar más servidores, haciendo al sistema menos costoso y más flexible.

### **3.¿Qué es una clave primaria y cuál es su función?**

La clave primaria es la columna o conjunto de columnas que identifica de forma única a cada fila de una tabla en una base de datos relacional. Ningún valor se puede repetir, no puede ser null, e idealmente, nunca debería cambiar. La función principal de una clave primaria es identificar cada registro, pero tiene otras funciones como la posibilidad de que otras tablas puedan usar la clave primaria como clave foránea, creando relaciones entre tablas. También hace que algunas búsquedas, actualizaciones o bajas sean más rápidos al apuntar directamente al id de la fila a buscar.

### **4.En NoSQL, ¿Qué significa anidar documentos ? ¿Cómo podrías relacionarlo a lo visto en SQL?**

En NoSQL, la anidación significa tener documentos adentro de otros documentos. Entonces se pueden guardar estructuras completas y jerárquicas en un solo objeto BSON.

Por ejemplo, un documento de una persona, puede tener dentro un documento ‘direccion’ con propiedades como ‘calle’, ‘numero’, ‘ciudad’. En SQL para tener algo así, necesitaríamos tener dos tablas y hacer la relación entre ellas mediante una clave foránea.

### **5.Menciona dos ventajas de usar SQL en vez de NoSQL**

**-Consistencia de datos:** Las bases de datos SQL siguen las cuatro propiedades principales que garantizan la integridad y consistencia de las transacciones, conocidas como ACID (atómicas, consistentes, aisladas y durables). Esto significa que las transacciones son seguras y los datos siempre son consistentes incluso si hay fallos.

**-Consultas:** Permite hacer consultas complejas entre muchas tablas usando métodos como JOIN, y usar agrupamientos, filtros avanzados, subconsultas, etc, con un buen rendimiento y optimización.

### **6.¿Qué tipo de base de datos usarías para un sistema bancario? Justifica tu respuesta**

Usaría una base de datos relacional, principalmente por la seguridad de las transacciones (propiedades ACID), evitando posibles grandes errores como que una se envíe dos veces, saldos incorrectos o transacciones incompletas. Además, SQL maneja de forma muy eficiente cuando hay muchas operaciones al mismo tiempo, y controla bien las relaciones complejas entre datos, ideal para un sistema bancario donde los datos suelen estar muy relacionados.

### **7.Define ACID en bases de datos relacionales.**

ACID son un grupo de propiedades fundamentales para garantizar la fiabilidad e integridad de los datos. Las propiedades son:

**-Atomicidad (Atomicity):** Una transacción se completa totalmente o no se ejecuta nada. Si en el medio algo falla, se revierte toda la operación para no dejar nada incompleto.

**-Consistencia (Consistency):** La base de datos siempre pasa de un estado válido a otro válido, respetando las reglas definidas.

**-Aislamiento (Isolation):** Cada transacción se ejecuta de forma independiente, sin interferirse entre sí.

-Durabilidad (Durability): Cuando la transacción se confirma, los cambios quedan guardados permanentemente. Si hay un fallo o pasa algo después, no afecta a los cambios ya guardados.

### **8.Explica el concepto de "sharding" en bases de datos NoSQL.**

Sharding es una técnica para dividir una base de datos grande en partes más pequeñas llamadas shards (fragmentos), y distribuirlas en varios servidores. Cada shard contiene una porción diferente de los datos, y juntos forman la base de datos completa. Es la técnica base para la escalabilidad horizontal, donde en lugar de crecer un servidor, se agregan más y se reparten los datos entre ellos.

### **9.¿Qué es un "documento" en MongoDB?**

Un documento es un objeto similar al JSON que representa un registro completo. Es flexible, respeta jerarquías, y puede tener otros documentos o arreglos dentro.

{

"nombre": "Pedro",

"direccion": {

"calle": "Moreno 1111",

"ciudad": "Olavarría"

},

"telefonos": ["123123", "123124"],

"activo": true

}

En este ejemplo de documento, contiene dentro un subdocumento (direccion) y un arreglo (telefonos)

### **10.¿Cuáles son los tipos principales de bases de datos NoSQL?**

Hay cuatro tipos principales, y se diferencian en la forma en la que almacenan y manejan los datos.

1. **Bases de datos de documentos:** almacenan datos como objetos en formato JSON o BSON, tienen un esquema muy flexible, ideal para datos jerárquicos. La más conocida es MongoDB.
2. **Bases de datos clave-valor:** almacenan datos en pares clave-valor, donde la clave es un identificador único y el valor es el dato asociado a esa clave. Son muy rápidas y simples.
3. **Bases de datos de columnas:** almacenan datos por columnas en lugar de filas. Son excelentes para consultas masivas y análisis a gran escala.
4. **Bases de datos de grafos:** almacenan datos como nodos y relaciones (aristas). Son muy eficientes para representar y consultar relaciones complejas. El uso más común es en redes sociales.