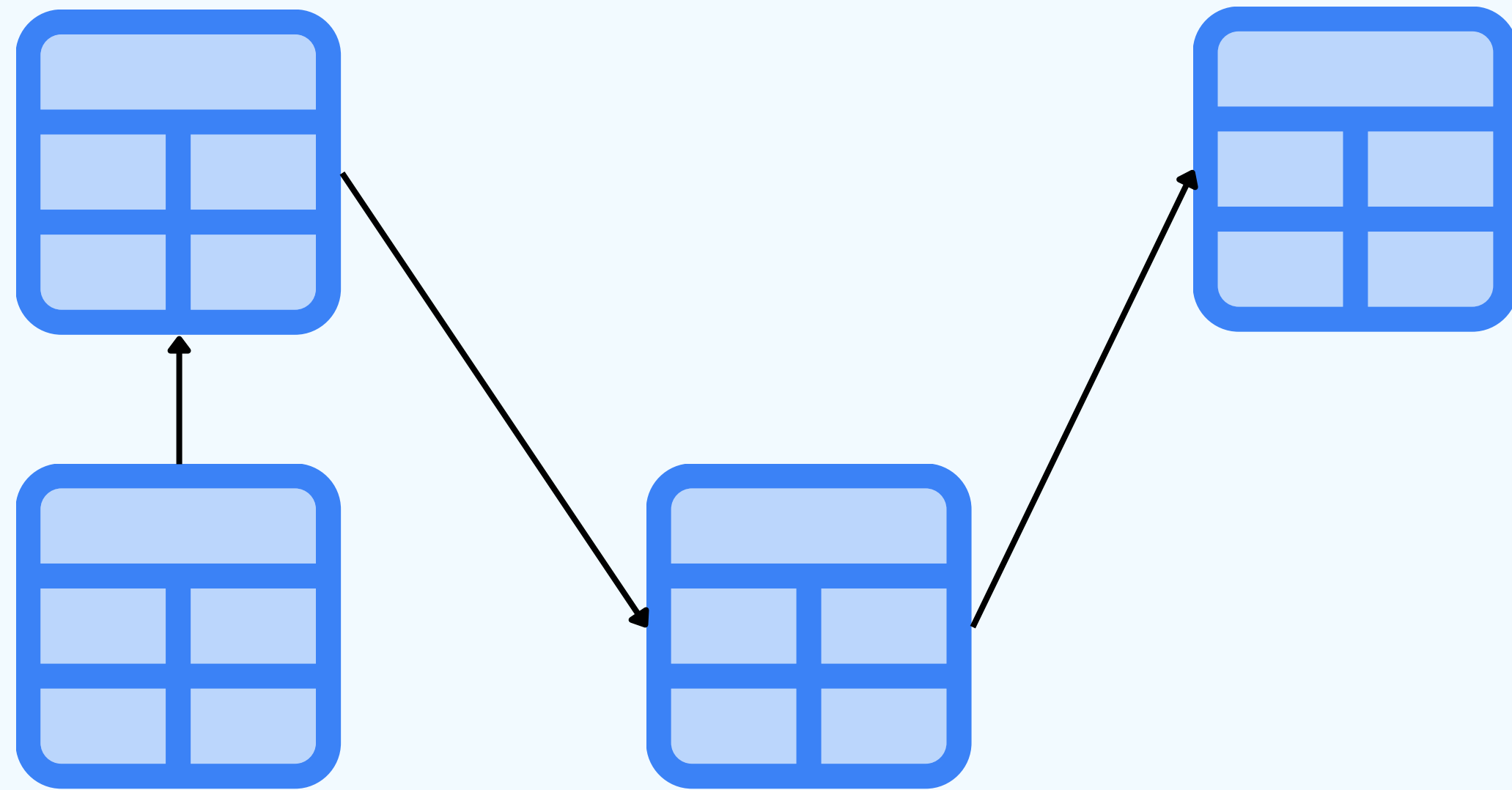


ALMACENAMIENTO DE DATOS

BBDD RELACIONALES VS NOSQL

Daniel Blanco Calviño

BASES DE DATOS RELACIONALES



BASES DE DATOS RELACIONALES

- **Esquema de datos predefinido.**
 - Información estable y consistente.
 - Lenguaje de consultas robusto y potente, **SQL**.

BASES DE DATOS RELACIONALES

- **Esquema de datos predefinido.**
 - Información estable y consistente.
 - Lenguaje de consultas robusto y potente, **SQL**.
- **Evita la duplicidad de información.**

BASES DE DATOS RELACIONALES

user

id	username	email	created_date
U1	daniel	daniel@email.com	1697300359
U2	noelia	noelia@email.com	1695300359

post

id	title	content	created_date	username	user_email
P1	First title	First content	1697300359	daniel	daniel@email.com
P2	Second title	Second title	1695300359	noelia	noelia@email.com

BASES DE DATOS RELACIONALES

user

id	username	email	created_date
U1	daniel	daniel@email.com	1697300359
U2	noelia	noelia@email.com	1695300359

post

id	title	content	created_date	user_id
P1	First title	First content	1697300359	U1
P2	Second title	Second title	1695300359	U2

BASES DE DATOS RELACIONALES

- **Esquema de datos predefinido.**
 - Información estable y consistente.
 - Lenguaje de consultas robusto y potente, **SQL**.
- **Evita la duplicidad de información.**
- **Transacciones ACID.**
 - Atomicity.
 - Consistency.
 - Isolation.
 - Durability.

TRANSACCIONES ACID

- **Atomicity**

- Las transacciones se ejecutan como una unidad atómica indivisible.
- O se ejecuta todo, o no surte ningún efecto.

TRANSACCIONES ACID

- Insertamos un pedido de tres televisores y dos ordenadores para el usuario U1.

order

id	user	total_price
01	U1	3400

order_line

id	product_id	quantity	price
----	------------	----------	-------



Fallo al insertar order lines. Estado inconsistente, no refleja la realidad.

TRANSACCIONES ACID

- Insertamos un pedido de tres televisores y dos ordenadores para el usuario U1.

order

id	user	total_price
----	------	-------------

order_line

id	product_id	quantity	price
----	------------	----------	-------



Operaciones atómicas. Se realiza **rollback**, dejando un estado válido.

TRANSACCIONES ACID

- **Atomicity**

- Las transacciones se ejecutan como una unidad atómica indivisible.
- O se ejecuta todo, o no surte ningún efecto.

- **Consistency**

- Si partimos de un estado consistente antes de ejecutar una transacción, va a seguir consistente cuando se finalice, haya finalizado satisfactoriamente o no.

TRANSACCIONES ACID

- **Atomicity**

- Las transacciones se ejecutan como una unidad atómica indivisible.
- O se ejecuta todo, o no surte ningún efecto.

- **Consistency**

- Si partimos de un estado consistente antes de ejecutar una transacción, va a seguir consistente cuando se finalice, haya finalizado satisfactoriamente o no.

- **Isolation**

- Una transacción no se verá afectada por otras transacciones concurrentes.

TRANSACCIONES ACID

Transacción insertando datos
en progreso

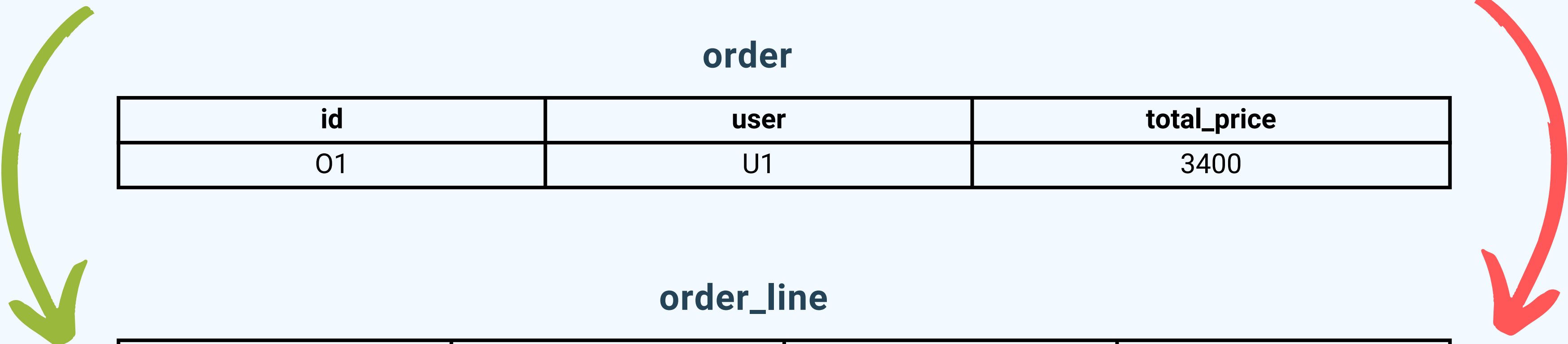
Transacción leyendo datos no
confirmados

order

id	user	total_price
O1	U1	3400

order_line

id	product_id	quantity	price
OL1	P1	3	2000



TRANSACCIONES ACID

- **Atomicity**

- Las transacciones se ejecutan como una unidad atómica indivisible.
- O se ejecuta todo, o no surte ningún efecto.

- **Consistency**

- Si partimos de un estado consistente antes de ejecutar una transacción, va a seguir consistente cuando se finalice, haya finalizado satisfactoriamente o no.

- **Isolation**

- Una transacción no se verá afectada por otras transacciones concurrentes.

- **Durability**

- Los efectos de las transacciones confirmadas van a mantenerse en la base de datos.

ASPECTOS POSITIVOS BBDD RELACIONALES



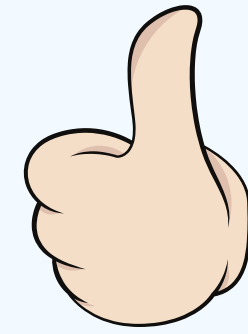
- **Sistemas robustos que llevan siendo usados muchas décadas.**
 - Transacciones ACID. Aseguran fiabilidad y consistencia total.
- **Lenguaje muy potente para realizar consultas complejas.**
 - Información almacenada de forma estructurada.
- **Eficiencia en el almacenamiento.**
 - Se reduce la duplicidad de información.
- **Estructura natural para los humanos.**
 - Intuitivo incluso para personas fuera del mundo de la informática.

ASPECTOS NEGATIVOS BBDD RELACIONALES



- **Estructura predefinida.**
 - Desventaja clara si necesitamos flexibilidad.
- **Más difícil de escalar** que las BBDD no relacionales.
 - Las transacciones ACID ralentizan el procesamiento.
- **Lecturas más lentas.**
 - Garantías de las transacciones ACID.
 - Joins entre tablas.

CUÁNDO USAR BBDD RELACIONALES



- Sistemas con **estructura de datos bien definida**.
- En casos dónde las garantías de las **transacciones ACID sean totalmente necesarias**.
- Si queremos realizar **consultas complejas** sobre nuestros datos.

BASES DE DATOS NOSQL

- **Documentales.**
 - MongoDB.
- **Clave-Valor.**
 - Redis.
 - DynamoDB.
- **Orientados a columnas.**
 - Apache Cassandra.
- **Grafos.**
 - Neo4J.
- **Series Temporales.**
 - InfluxDB



BASES DE DATOS NOSQL

- **No tienen un esquema predefinido.**
 - No es necesario conocer perfectamente la estructura de los datos con antelación.

BASES DE DATOS NOSQL

- **No tienen un esquema predefinido.**
 - No es necesario conocer perfectamente la estructura de los datos con antelación.
- Estructuras intuitivas para los programadores.
 - Objetos, listas, maps, grafos etc.
 - Se elimina la necesidad de librerías externas como los ORMs.

BASES DE DATOS NOSQL

- **No tienen un esquema predefinido.**
 - No es necesario conocer perfectamente la estructura de los datos con antelación.
- Estructuras intuitivas para los programadores.
 - Objetos, listas, maps, grafos etc.
 - Se elimina la necesidad de librerías externas como los ORMs
- Pocos soportan transacciones ACID. **Siguen el modelo BASE.**
 - Basically Available.
 - Soft State.
 - Eventually consistent.

BASE

- **Basically Available**

- La disponibilidad de prioriza mucho.
- Lecturas y escrituras posibles incluso en fallos parciales o particiones de red.
- Se busca una consistencia eventual.

BASE

- **Basically Available**

- La disponibilidad de prioriza mucho.
- Lecturas y escrituras posibles incluso en fallos parciales o particiones de red.
- Se busca una consistencia eventual.

- **Soft State**

- Se permite tener información en estados intermedios o transitorios.

BASE

- **Basically Available**

- La disponibilidad de prioriza mucho.
- Lecturas y escrituras posibles incluso en fallos parciales o particiones de red.
- Se busca una consistencia eventual.

- **Soft State**

- Se permite tener información en estados intermedios o transitorios.

- **Eventually consistent**

- No se garantiza una consistencia total en todo momento.
- Lecturas realizadas por diferentes usuarios pueden tener distintos resultados.
- Con el tiempo, se asegura que los datos estén en un estado consistente.

ASPECTOS POSITIVOS NOSQL



- **No es necesario un esquema de datos preestablecido.**
 - Si necesitamos un nuevo campo para algún registro, lo podemos insertar sin afectar a los demás.
- **Mayor escalabilidad y rendimiento.**
 - Las propiedades BASE sacrifican consistencia para ganar rendimiento.
- **Mayor disponibilidad.**
 - Las propiedades BASE nos ofrecen también esta característica.

ASPECTOS NEGATIVOS NOSQL



- **Falta de transacciones ACID completas.**
 - La mayoría de sistemas no relacionales sacrifican algunas propiedades ACID.
- **Falta de estándares claros y definidos.**
 - Sistemas de creación y uso más reciente. Son menos maduros.
- Las **consultas** específicas y complicadas son **mucho más complejas.**
 - La naturaleza no estructurada de los datos lo hace más difícil.

CUÁNDO USAR NOSQL



- Sistemas que **carecen** de una **estructura de datos fija o bien definida**.
- Sistemas para los que la **consistencia total no sea un requisito**.
 - Si no necesitamos ACID, podremos tener mayor rendimiento, escalabilidad y disponibilidad.
- Necesidad de **mayor rendimiento en las lecturas**.