

The background features a complex network diagram with numerous nodes (colored circles in green, orange, purple, and black) and connecting lines, set against a dark grey field. On the right side, there are large, overlapping, semi-transparent green geometric shapes. The overall aesthetic is modern and technical.

Transacciones

Realizado por:

Manuel Cerrejón Naranjo

Gonzalo Casquete Rodríguez



Índice

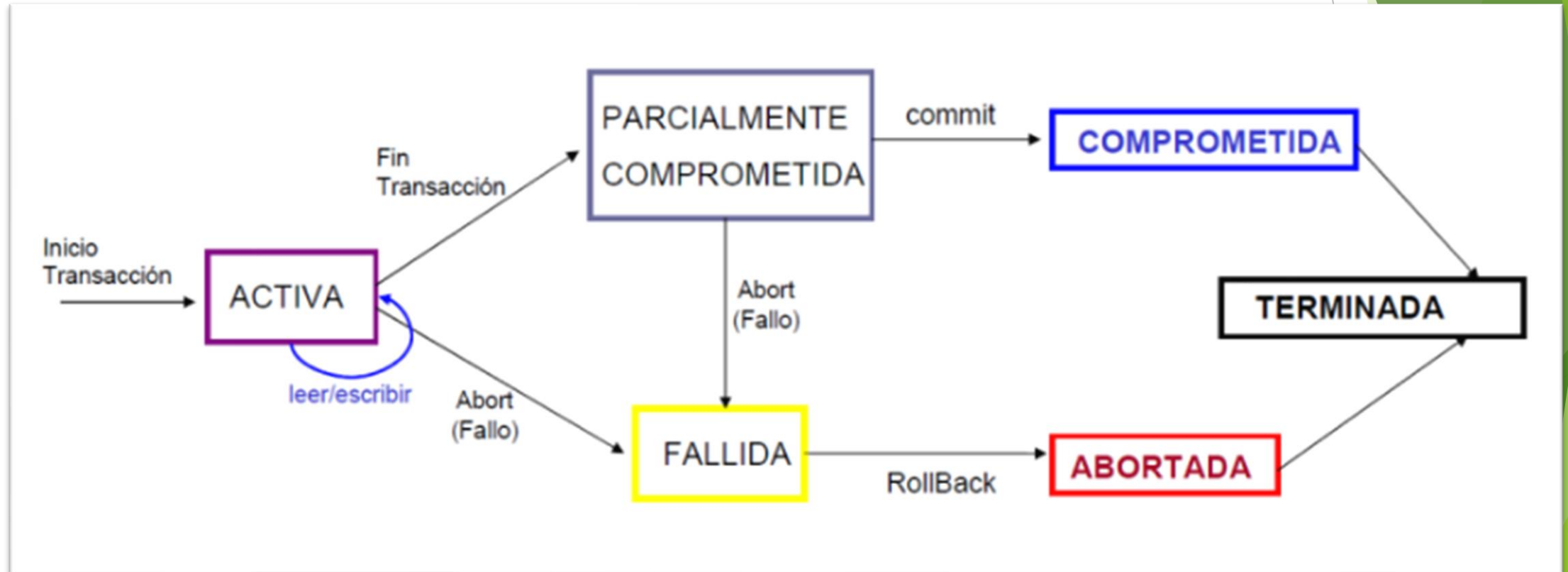
1. ¿Qué es una transacción?
2. Conceptos de transacciones
3. Visión general de control de transacciones
4. Transacciones Autónomas
5. Transacciones distribuidas



¿Qué es una transacción?

- Las transacciones son un conjunto de operaciones sobre los datos de una base de datos, la manera que tienen de ejecutarse dichas operaciones aseguran la integridad de dicha BD observando los principios de atomicidad, consistencia, aislamiento y duración.

Pasos en la ejecución de una transacción



Principios de las transacciones

Atomicidad

Consistencia

Aislamiento

Duración

Atomicidad

- ▶ Este principio es el encargado de observar que todas las operaciones que componen una transacción sean correctas, si alguna de ellas contiene una errata o es errónea se cancelara la transacción.
- ▶ Ejemplo:
 - ▶ La cuenta A ingresa 1000€ a la cuenta B:

Cuenta	Saldo bancario
A	500
B	1000

Ejemplo atomicidad

Cuenta	Saldo bancario
A	500



Retirar
saldo

Cuenta	Saldo bancario
A	-500



Ingresar saldo



Cuenta	Saldo bancario
B	1000

Rollback



Fallo en la transacción

Consistencia

La consistencia es la propiedad encargada de asegurar que los datos almacenados estén en un estado válido y coherente, en otras palabras, que cumplan una serie de reglas y restricciones definidas anteriormente.

Base de Datos

Producto	Stock
P001	22

Stock físico



P001 tiene 10 unidades



Inconsistencia

Aislamiento

- El aislamiento se refiere a la propiedad de que una transacción pueda mantener independencia entre transacciones que trabajen sobre los mismos datos.

Productos

Id	Nombre	Precio
P001	Manzana	2

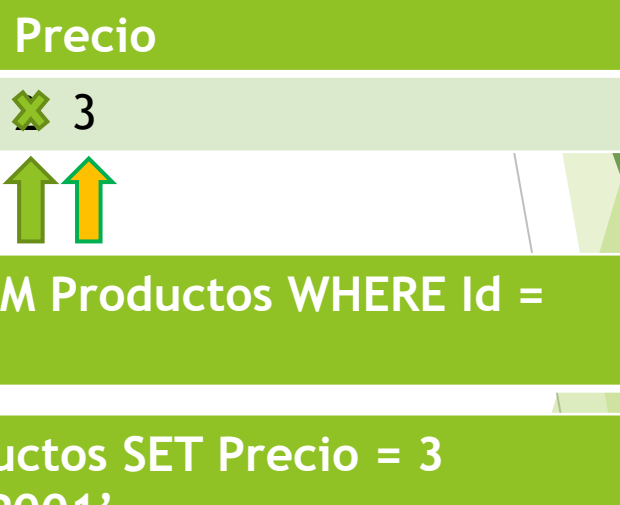
Transacciones

Usuario 1	SELECT * FROM Productos where Id like 'P001';
Usuario 2	Update Productos set precio=3 where Id like 'P001';
Usuario 3	SELECT * FROM Productos where Id like 'P001';

Aislamiento

Id	Nombre	Precio
P001	Manzana	✖ 3

Usuario 1	SELECT * FROM Productos WHERE Id = 'P001'
Usuario 2	UPDATE Productos SET Precio = 3 WHERE Id = 'P001'
Usuario 3	SELECT * FROM Productos WHERE Id = 'P001'



Durabilidad

- La durabilidad es una propiedad de las bases de datos que se refiere a la capacidad de una transacción para ser permanente y resistir cualquier tipo de falla o error en el sistema.

Id	Nombre	Precio
P001	Manzana	✖ 3



Usuario 2	UPDATE Productos SET Precio = 3 WHERE Id = 'P001'
-----------	---



Se produce un error
en el sistema



El cambio ha sido
guardado permanentemente

Control de transacciones

- ▶ El control de transacciones es un proceso que se utiliza para garantizar la integridad y consistencia de los datos en una base de datos, para ello se basa en los principios de ACID vistos anteriormente y utiliza distintos mecanismos para aplicarlos:
- ▶ COMMIT: Técnica que se utiliza para asegurar que todas las transacciones se realicen de forma segura y que los cambios realizados en la base de datos sean permanentes y duraderos.
- ▶ ROLLBACK: Mecanismo que se utiliza para restaurar la base de datos a su estado anterior en caso de que se produzca un fallo en el sistema, para poder volver a un estado anterior la base de datos hará uso de savepoints.
- ▶ SAVEPOINT: Estado que identifica la base de datos y deja almacenado de manera que en caso de error se realizara un rollback hasta dicho punto.

Ejemplo control de transacciones

Tiempo	Sesión	Explicación
t0	COMMIT;	Acaba cualquier transacción anterior almacenando los cambios.
t1	SET TRANSACTION NAME 'sal_update';	Inicio de una transacción.
t2	UPDATE employees SET salary = 7000 WHERE last_name = 'Banda';	Actualización salario de Banda a 7000.
t3	SAVEPOINT after_banda_sal;	Creación savepoint.
t4	UPDATE employees SET salary = 12000 WHERE last_name = 'Greene';	Actualización salario de Greene a 12000.

Ejemplo control de transacciones 2


Tiempo	Sesión	Explicación
t5	SAVEPOINT after_greene_sal;	Creación savepoint.
t6	ROLLBACK TO SAVEPOINT after_banda_sal;	Rollback al savepoint llamado “after_banda_sal”.
t7	UPDATE employees SET salary = 11000 WHERE last_name = 'Greene';	Actualización salario Greene a 11000.
t8	ROLLBACK;	Rollback de la transacción, se elimina cualquier cambio y acaba la transacción
t9	SET TRANSACTION NAME 'sal_update2';	Inicio de una transacción.

Ejemplo control de transacciones 2

Tiempo	Sesión	Explicación
t10	UPDATE employees SET salary = 7050 WHERE last_name = 'Banda';	Actualización salario Banda a 7050.
t11	UPDATE employees SET salary = 10950 WHERE last_name = 'Greene';	Actualización salario Greene a 10950.
t12	COMMIT;	Acaba cualquier transacción anterior almacenando los cambios.

Introducción a transacciones autónomas

Las transacciones autónomas permiten gestionar automáticamente las transacciones sin la intervención humana. Estas transacciones autónomas permiten que los sistemas de bases de datos sean más escalables, seguros y confiables.



Esto significa que las aplicaciones pueden iniciar transacciones con la base de datos sin preocuparse por la implementación de la lógica de gestión de transacciones, lo que reduce significativamente la complejidad del desarrollo de aplicaciones.



Además, las transacciones autónomas también proporcionan un alto nivel de seguridad y confiabilidad puesto que en caso que de alguna falle se realizara un rollback al último savepoint almacenado.

Ejemplo transacción autónoma

Tabla Productos

idProduc	Nombre	Precio
001	Manzana	5
002	Pera	5
003	Melón	5
004	Kiwi	5

INSERT



Tabla Productos

idProduc	Nombre	Precio
001	Manzana	5
002	Pera	5
003	Melón	5
004	Kiwi	5
005	Limón	5

Transacciones distribuidas

- Este tipo de transacciones destaca el hecho de que serán utilizadas por múltiples usuarios en diferentes sistemas informáticos y generalmente en lugares geográficamente dispersos, este tipo de transacciones son comunes en sistemas distribuidos o en entornos de computación en la nube.

Desventajas

Por regla general estas transacciones son más complejas que las transacciones centralizadas ya que requiere un mayor nivel de sincronización y coordinación, a su vez también puede generar problemas de seguridad por la propia naturaleza de los sistemas distribuidos.

Ventajas

Como ventajas podemos destacar el hecho de que son fácilmente escalables, permiten manejar grandes volúmenes de transacciones, tienen una gran tolerancia de fallos en uno o más sistemas y permite la colaboración entre múltiples participantes.

Ejemplo transacciones distribuidas

Base de datos Santander

Cuenta	Saldo
ES0315448631943153297274	1000\$

Una persona quiere realizar un pago a una cuenta bancaria de EEUU

```
SELECT Saldo FROM Santander WHERE Titular = 'Numero de cuenta' AND Banco = 'Santander' AND Saldo >= 500;
```

```
UPDATE Santander SET Saldo = Saldo - 500 WHERE Cuenta = 'Numero de Cuenta';
```



Cuenta	Saldo
ES0315448631943153297274	500\$

Base de datos JPMorgan Chase

Cuenta	Saldo
US4315448631943153297344	1000\$

El banco recibe la petición y tiene que autorizar la transacción

```
UPDATE JPMorgan Chase SET Saldo = Saldo + 500 WHERE Cuenta = 'Numero de Cuenta';
```



Cuenta	Saldo
US4315448631943153297344	1500\$

Preguntas

