

Administración de Bases de Datos

Concurrencia de Datos y Consistencia

Javier Román Pásaro y Raúl Vázquez Guerra



Tabla de Contenidos

01

Conceptos Previos

Nociones básicas de transacciones, visión general de los niveles de aislamiento de transacciones en Oracle.

02

Concurrencia de datos

Definición, problemas de concurrencia y mecanismos de gestión.

03

Consistencia de datos

Definición, problemas de consistencia y mecanismos de gestión.

04

Otros aspectos y conclusiones

Técnicas de diseño de un SBD. Influencia en el rendimiento y escalabilidad de la BD. Conclusiones

Tabla de Contenidos

01

Conceptos Previos

Nociones básicas de transacciones, visión general de los niveles de aislamiento de transacciones en Oracle.

02

Concurrencia de datos

Definición, problemas de concurrencia y mecanismos de gestión.

03

Consistencia de datos

Definición, problemas de consistencia y mecanismos de gestión.

04

Otros aspectos y conclusiones

Técnicas de diseño de un SBD. Influencia en el rendimiento y escalabilidad de la BD. Conclusiones

01

Conceptos Previos

Nociones básicas de transacciones, visión general de los niveles de aislamiento de transacciones en Oracle.

Transacciones

Serie de acciones llevadas a cabo por un único usuario o por un programa de usuario, que lee y actualiza el contenido de la BD

A ————— **C** ————— **I** ————— **D**

Atomicidad

Las transacciones son un todo o nada

Consistencia

Sólo se guardan datos válidos

Aislamiento

Las transacciones no se afectan entre sí

Durabilidad

Los datos escritos no se perderán

Aislamiento de transacciones

En Oracle, existen diferentes niveles de aislamiento de transacciones, que se utilizan para controlar cómo las transacciones interactúan entre sí y con los datos



Aislamiento de transacciones

En Oracle, existen diferentes niveles de aislamiento de transacciones, que se utilizan para controlar cómo las transacciones interactúan entre sí y con los datos



Permite a una transacción leer los datos que han sido modificados por otra transacción que aún no ha sido confirmada. No garantiza la integridad de los datos y puede llevar a lecturas inconsistentes.

Aislamiento de transacciones

En Oracle, existen diferentes niveles de aislamiento de transacciones, que se utilizan para controlar cómo las transacciones interactúan entre sí y con los datos



Permite a una transacción leer datos solo después de que han sido confirmados por otra transacción. Esto garantiza que las lecturas sean consistentes, pero aún pueden ocurrir problemas de lectura sucia en situaciones de alta concurrencia.

Aislamiento de transacciones

En Oracle, existen diferentes niveles de aislamiento de transacciones, que se utilizan para controlar cómo las transacciones interactúan entre sí y con los datos



Garantiza que una transacción pueda repetir una lectura y obtener el mismo resultado cada vez. Para lograr esto, las transacciones bloquean las filas de la tabla que están leyendo, lo que puede llevar a problemas de bloqueo en situaciones de alta concurrencia.

Aislamiento de transacciones

En Oracle, existen diferentes niveles de aislamiento de transacciones, que se utilizan para controlar cómo las transacciones interactúan entre sí y con los datos



Garantiza que las transacciones se ejecuten como si se hubieran ejecutado una tras otra en secuencia, lo que garantiza la integridad de los datos. Esto se logra bloqueando todas las filas de la tabla que se van a leer o modificar, lo que puede llevar a problemas de bloqueo significativos en situaciones de alta concurrencia.

Aislamiento de transacciones

Los beneficios de utilizar diferentes niveles de aislamiento de transacciones en Oracle son los siguientes:



Aislamiento de transacciones

Los beneficios de utilizar diferentes niveles de aislamiento de transacciones en Oracle son los siguientes:



El uso adecuado de los niveles de aislamiento de transacciones puede ayudar a controlar la concurrencia y evitar problemas de bloqueo y lectura sucia.

Aislamiento de transacciones

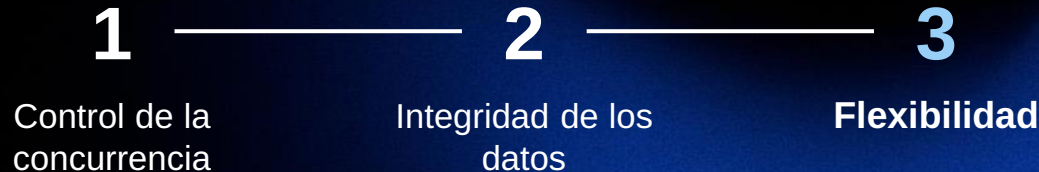
Los beneficios de utilizar diferentes niveles de aislamiento de transacciones en Oracle son los siguientes:



Los niveles de aislamiento más altos, como `SERIALIZABLE`, garantizan la integridad de los datos al bloquear todas las filas de la tabla que se van a leer o modificar.

Aislamiento de transacciones

Los beneficios de utilizar diferentes niveles de aislamiento de transacciones en Oracle son los siguientes:



Los diferentes niveles de aislamiento de transacciones permiten a los desarrolladores elegir el nivel adecuado para cada situación, equilibrando la integridad de los datos con la necesidad de un alto rendimiento.

Tabla de Contenidos

01

Conceptos Previos

Nociones básicas de transacciones, visión general de los niveles de aislamiento de transacciones en Oracle.

02

Concurrencia de datos

Definición, problemas de concurrencia y mecanismos de gestión.

03

Consistencia de datos

Definición, problemas de consistencia y mecanismos de gestión.

04

Otros aspectos y conclusiones

Técnicas de diseño de un SBD. Influencia en el rendimiento y escalabilidad de la BD. Conclusiones

02

Concurrencia de datos

Definición, problemas de concurrencia y mecanismos de gestión.

Concurrencia de datos

Capacidad de manejar múltiples solicitudes de acceso a la base de datos al mismo tiempo, de manera que varias transacciones puedan ejecutarse simultáneamente sin interferir entre sí.

Problemas de concurrencia:

- Lectura sucia
- Lectura no repetible
- Lectura fantasma

Técnicas de control de concurrencia:

- Protocolos basados en bloqueo
- Protocolos basados en marcas de tiempo
- Protocolos basados en grafos



Problemas de concurrencia

- **Lectura sucia**
- Lectura no repetible
- Lectura fantasma

Usuario 1	Usuario 2
BEGIN	BEGIN
UPDATE camiseta SET precio = 100	
	SELECT precio FROM camiseta
ROLLBACK	
	COMMIT

Se produce cuando una transacción lee los datos de una fila que ha sido modificada por otra transacción, pero que aún no ha sido confirmada. Esto puede causar que la transacción lea datos incorrectos o incompletos.

Problemas de concurrencia

- Lectura sucia
- **Lectura no repetible**
- Lectura fantasma

Usuario 1	Usuario 2
BEGIN	BEGIN
SELECT salario FROM empleados	
	UPDATE empleados SET salario = 100
	COMMIT
SELECT salario FROM empleados	
COMMIT	

Se produce cuando una transacción lee los mismos datos de una fila dos veces, pero los datos han sido modificados por otra transacción en el intervalo entre las dos lecturas, lo que hace que los datos leídos sean diferentes.

Problemas de concurrencia

- Lectura sucia
- Lectura no repetible
- **Lectura fantasma**

Usuario 1	Usuario 2
BEGIN	BEGIN
SELECT id FROM empleados WHERE salario > 70	
	UPDATE empleados SET salario = 50 WHERE id = 23
	COMMIT
SELECT id FROM empleados WHERE salario > 70	
COMMIT	

Se produce cuando una transacción intenta leer un conjunto de datos que cumple con ciertos criterios, pero luego otra transacción modifica o inserta datos que también cumplen con esos mismos criterios, lo que hace que la transacción original lea un conjunto de datos diferente.

Técnicas de control de concurrencia

- **Protocolos basados en bloqueo**

- * Reservas

- * 2 Fases

- Protocolos basados en marcas de tiempo

- Protocolos basados en grafos

Esta técnica utiliza bloqueos para controlar el acceso a los datos y garantizar la consistencia de los mismos en un entorno concurrente

Técnicas de control de concurrencia

- **Protocolos basados en bloqueo**

- * Reservas

- * 2 Fases

- Protocolos basados en marcas de tiempo

- Protocolos basados en grafos

Reglas básicas del bloqueo:

Bloqueo compartido: si una transacción tiene un bloqueo compartido sobre un elemento de datos, puede leer el elemento, pero no actualizarlo (escribir).

Bloqueo exclusivo: si una transacción tiene un bloqueo exclusivo sobre un elemento de datos, puede leer y actualizar (escribir) el elemento

Técnicas de control de concurrencia

- Protocolos basados en bloqueo

- * Reservas**

- * 2 Fases**

- Protocolos basados en marcas de tiempo
- Protocolos basados en grafos

Tipos de Candados:

1. Candados binarios
2. Candados compartidos y exclusivos.



Una transacción tiene que obtener la reserva de un elemento antes de poder operar sobre él. Utilizan Protocolos.

Técnicas de control de concurrencia

- Protocolos basados en bloqueo

* Reservas

* 2 Fases

- Protocolos basados en marcas de tiempo
- Protocolos basados en grafos

Tipos de Candados:

1. Candados binarios

Bloqueado

Desbloqueado

2. Candados compartidos y exclusivos.



Una transacción tiene que obtener la reserva de un elemento antes de poder operar sobre él. Utilizan Protocolos.

Técnicas de control de concurrencia

- Protocolos basados en bloqueo

* Reservas

* 2 Fases

- Protocolos basados en marcas de tiempo
- Protocolos basados en grafos

Tipos de Candados:

1. Candados binarios

2. Candados compartidos y exclusivos.

Bloqueado de Lectura
Bloqueado de Escritura
Desbloqueo



Una transacción tiene que obtener la reserva de un elemento antes de poder operar sobre él. Utilizan Protocolos.

Técnicas de control de concurrencia

- Protocolos basados en bloqueo

* Reservas

* 2 Fases

- Protocolos basados en marcas de tiempo
- Protocolos basados en grafos

Bloqueos en 2 fases		T1	T2
T1	T2	Bloquea(y,lec) Lee(y) Bloquea(x,esc+lec)	
Bloquea(y,lec)	Bloquea(x,lec)		Bloquea(x,lec)
Lee(y)	Bloquea(y,esc+lec)		<i>(se pone en espera hasta que T1 desbloquee x)</i>
Bloquea(x,esc+lec)	Lee(x)	Lee(x) x=x+y Escribe(x)	
Lee(x)	Desbloquee(x)	Desbloquee(y) Desbloquee(x)	
x=x+y	Lee(y)		Bloquea(y,esc+lec)
Escribe(x)	y=x+y		Lee(x)
Desbloquee(y)	Escribe(y)		Desbloquee(x)
Desbloquee(x)	Desbloquee(y)		

Una transacción T sigue el protocolo de bloqueo en dos fases si todas las operaciones de bloqueo preceden a la primera operación de desbloqueo

Técnicas de control de concurrencia

- Protocolos basados en bloqueo

- * Reservas

- * 2 Fases**

- Protocolos basados en marcas de tiempo
- Protocolos basados en grafos

Problemas

- Lectura No Confirmada
- Lectura No Repetible
- Resumen Incorrecto
- Actualización Perdida



Técnicas de control de concurrencia

- Protocolos basados en bloqueo

* Reservas

* 2 Fases

- Protocolos basados en marcas de tiempo
- Protocolos basados en grafos

Lectura no confirmada

T1: R(X) W(X)

R(Y) abort

PB2F: Lectura no confirmada

T2: R(X) W(X)

T1

T2

Bloquear_lectura(X)

R(X)

Bloquear_escritura(X)

W(X)

Bloquear_lectura(X)

Bloquear_lectura(Y)

esperar

R(Y)

esperar

abort

R(X)

Técnicas de control de concurrencia

- Protocolos basados en bloqueo

* Reservas

* 2 Fases

- Protocolos basados en marcas de tiempo
- Protocolos basados en grafos

Lectura no repetible

T1: R(X) W(X) commit PB2F: Lectura no repetible
T2: R(X) R(X) commit

T1

T2

Bloquear_lectura(X)

R(X)

Bloquear_lectura(X)

R(X)

Bloquear_escritura(X)

esperar

R(X)

esperar

Desbloquear(*)

esperar

commit

W(X).....

M(X).....

esperar

commit

esperar

esperar(desbloquear)

Técnicas de control de concurrencia

- Protocolos basados en bloqueo

* Reservas

* 2 Fases

- Protocolos basados en marcas de tiempo
- Protocolos basados en grafos

Resumen Incorrecto

T2: R(A)W(s) ... R(X)W(s)R(Y)W(s) ... R(Z);W(s) commit

T1: R(X)W(X) R(Y) W(Y) commit

Técnicas de control de concurrencia

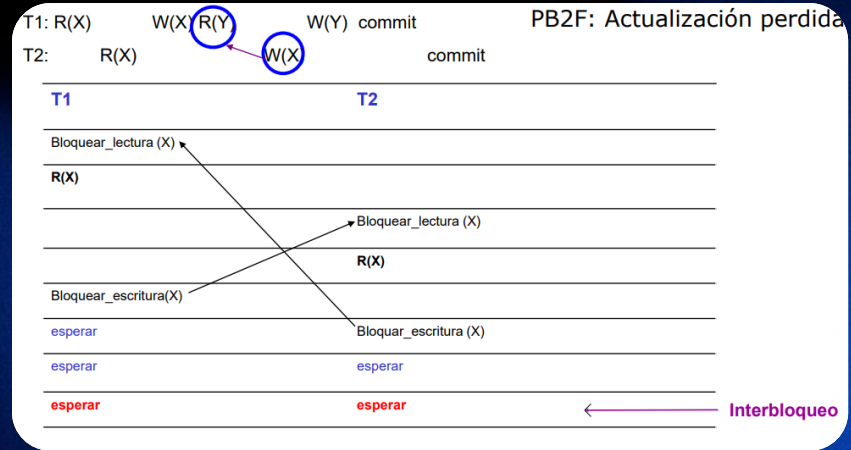
- Protocolos basados en bloqueo

* Reservas

* 2 Fases

- Protocolos basados en marcas de tiempo
- Protocolos basados en grafos

Actualización Perdida



Técnicas de control de concurrencia

- Protocolos basados en bloqueo

* Reservas

* 2 Fases

- Protocolos basados en marcas de tiempo

- Protocolos basados en grafos

Solución

PB2F estricto:

PB2F básico y además, no libera ninguna reserva hasta que acaba (con COMMIT o ABORT)



Técnicas de control de concurrencia

- Protocolos basados en bloqueo

- * Reservas

- * 2 Fases

- **Protocolos basados en marcas de tiempo**

- Protocolos basados en grafos

El control de consistencia basado en marcas temporales asigna marcas de tiempo y utiliza reglas de precedencia para asegurar la consistencia de los datos, revirtiendo automáticamente las operaciones que violan las reglas



Técnicas de control de concurrencia

- Protocolos basados en bloqueo

- * Reservas

- * 2 Fases

- Protocolos basados en marcas de tiempo

- **Protocolos basados en grafos**

El control de consistencia basado en grafo modela las transacciones como nodos y las operaciones como relaciones entre nodos.

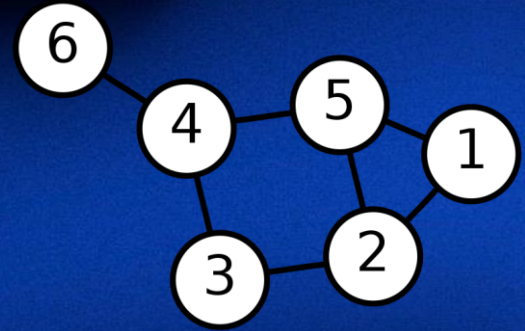


Tabla de Contenidos

01

Conceptos Previos

Nociones básicas de transacciones, visión general de los niveles de aislamiento de transacciones en Oracle.

02

Concurrencia de datos

Definición, problemas de concurrencia y mecanismos de gestión.

03

Consistencia de datos

Definición, problemas de consistencia y mecanismos de gestión.

04

Otros aspectos y conclusiones

Técnicas de diseño de un SBD. Influencia en el rendimiento y escalabilidad de la BD. Conclusiones

03

Consistencia de datos

Definición, problemas de consistencia y mecanismos de gestión.

Consistencia de datos

Los datos almacenados en la base de datos deben ser precisos y coherentes en todo momento. También se relaciona con la capacidad de mantener la integridad de los datos incluso en situaciones de fallas o errores.

Problemas de consistencia:

- Violación de restricciones de integridad
- Inconsistencia de datos
- Pérdida de datos
- Problemas de sincronización

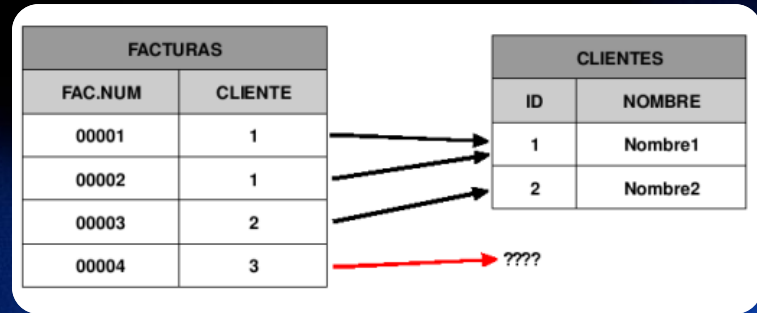
Mecanismos de control de consistencia:

- Validación de datos
- Reglas de integridad
- Transacciones atómicas
- Verificación de la coherencia de datos
- Control de versiones



Problemas de consistencia

- **Violación de restricciones de integridad**
- Inconsistencia de datos
- Pérdida de datos
- Problemas de sincronización



Las restricciones de integridad son reglas que se definen en una base de datos para garantizar que los datos almacenados sean precisos y coherentes. Si se violan estas reglas, puede haber inconsistencias en los datos.

Problemas de consistencia

- Violación de restricciones de integridad
- **Inconsistencia de datos**
- Pérdida de datos
- Problemas de sincronización

The diagram shows two identical 'Registro' (Registration) forms side-by-side, representing different states of a database. Both forms have the following fields: 'Nombre' (Name), 'Apellidos' (Surnames), 'E-mail', 'Usuario *' (Username), 'Contraseña' (Password), and 'Información General' (General Information). A red arrow points from the 'Usuario *' field in the left form to the 'Usuario *' field in the right form, highlighting a data inconsistency. The left form has 'Juan T' in the 'Nombre' field, 'Garcia' in the 'Apellidos' field, 'juan@test.test' in the 'E-mail' field, and 'matrix' in the 'Usuario *' field. The right form has 'George' in the 'Nombre' field, 'Jhous' in the 'Apellidos' field, 'g.jhous@test.test' in the 'E-mail' field, and 'matrix' in the 'Usuario *' field. The 'Contraseña' field in both forms is empty. The 'Información General' field in both forms contains a single line of text.

La inconsistencia de datos se produce cuando diferentes partes de la base de datos contienen información contradictoria o no coincidente. Esto puede ocurrir cuando una transacción falla o cuando una actualización de datos no se realiza correctamente.

Problemas de consistencia

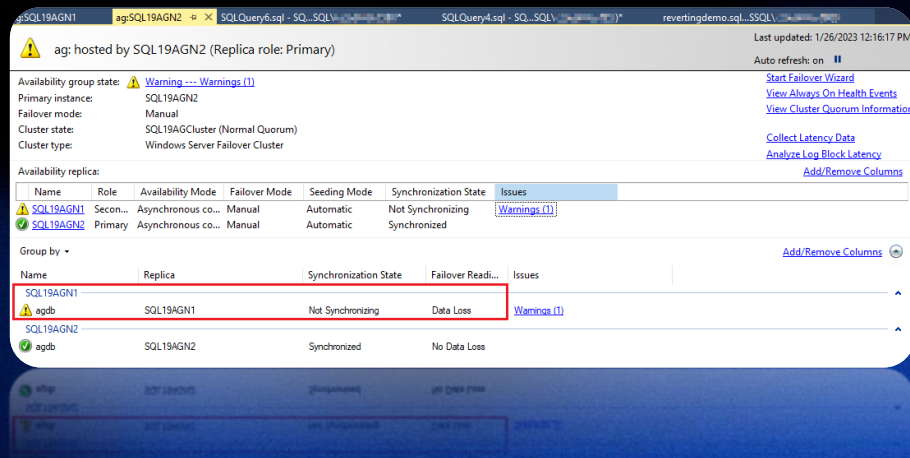
- Violación de restricciones de integridad
- Inconsistencia de datos
- **Pérdida de datos**
- Problemas de sincronización



La pérdida de datos se produce cuando se eliminan datos importantes de la base de datos por accidente o cuando hay fallas en el sistema que resultan en la pérdida de datos.

Problemas de consistencia

- Violación de restricciones de integridad
- Inconsistencia de datos
- Pérdida de datos
- **Problemas de sincronización**



ag: hosted by SQL19AGN2 (Replica role: Primary)

Last updated: 1/26/2023 12:16:17 PM

Auto refresh: on

Availability group state: [Warning --- Warnings \(1\)](#)

Primary instance: SQL19AGN2

Failover mode: Manual

Cluster state: SQL19AGCluster (Normal Quorum)

Cluster type: Windows Server Failover Cluster

Availability replica:

Name	Role	Availability Mode	Failover Mode	Seeding Mode	Synchronization State	Issues
SQL19AGN1	Secon...	Asynchronous co...	Manual	Automatic	Not Synchronizing	Warnings (1)
SQL19AGN2	Primary	Asynchronous co...	Manual	Automatic	Synchronized	

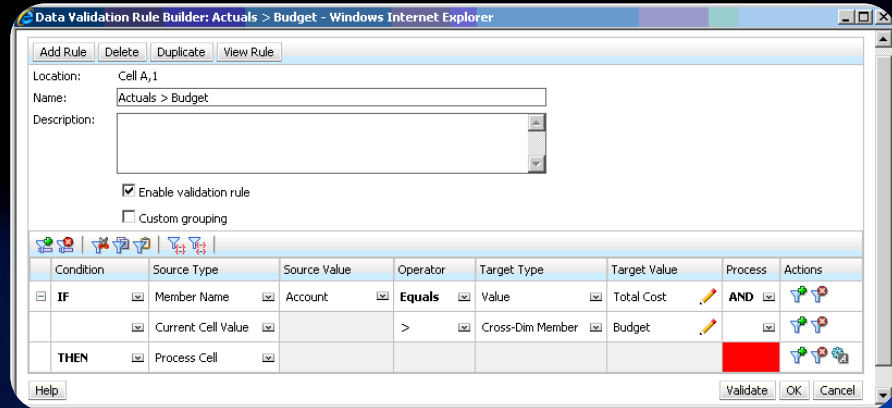
Group by

Name	Replica	Synchronization State	Failover Read...	Issues
SQL19AGN1	SQL19AGN1	Not Synchronizing	Data Loss	Warnings (1)
SQL19AGN2	SQL19AGN2	Synchronized	No Data Loss	

Los problemas de sincronización se producen cuando los datos en una base de datos no se actualizan correctamente en diferentes partes de la base de datos. Esto puede ser causado por problemas de red o por errores en la replicación de datos.

Mecanismos de control de consistencia

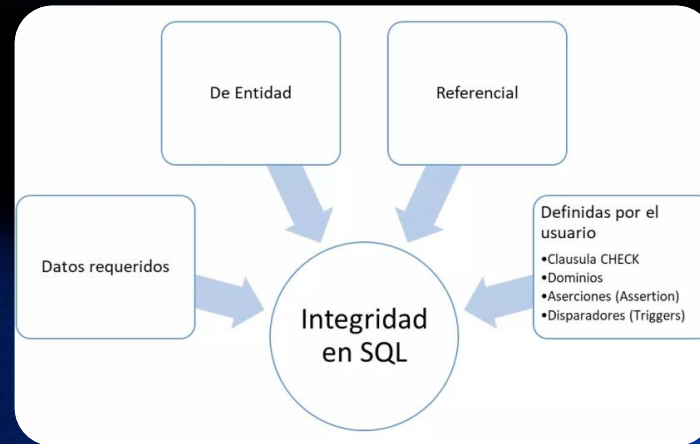
- **Validación de datos**
- Reglas de integridad
- Transacciones atómicas
- Verificación de la coherencia de datos
- Control de versiones



La validación de datos se utiliza para garantizar que los datos almacenados en el sistema sean precisos y estén en un formato coherente. Esto se logra mediante la definición de reglas de validación que restringen los tipos y formatos de datos que se pueden ingresar en el sistema (Definidas en el diccionario de datos).

Mecanismos de control de consistencia

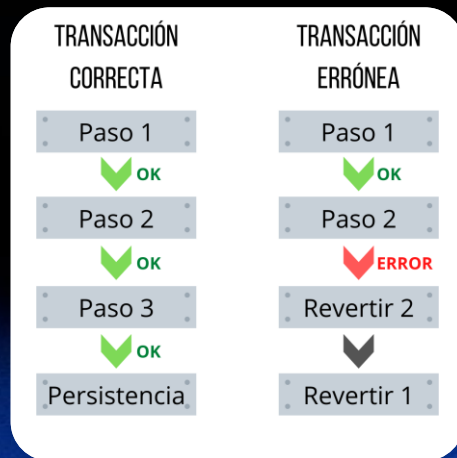
- Validación de datos
- **Reglas de integridad**
- Transacciones atómicas
- Verificación de la coherencia de datos
- Control de versiones



Las reglas de integridad se utilizan para garantizar que los datos almacenados en el sistema cumplan con ciertas restricciones y reglas definidas por el usuario. Por ejemplo, se puede definir una regla de integridad que impida que se ingresen valores negativos en un campo de cantidad.

Mecanismos de control de consistencia

- Validación de datos
- Reglas de integridad
- **Transacciones atómicas**
- Verificación de la coherencia de datos
- Control de versiones



Las transacciones atómicas se utilizan para garantizar que una serie de operaciones se realicen como una unidad indivisible. Esto significa que si una de las operaciones falla, todas las operaciones se deshacen y se restaura el estado anterior de los datos.

Mecanismos de control de consistencia

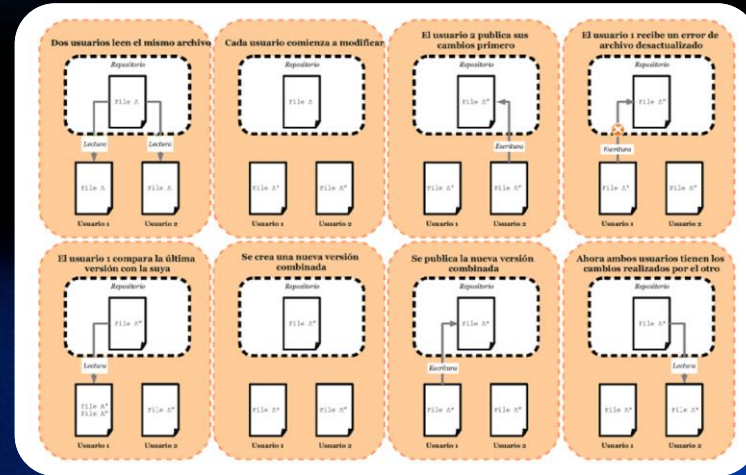
- Validación de datos
- Reglas de integridad
- Transacciones atómicas
- **Verificación de la coherencia de datos**
- Control de versiones



La verificación de la coherencia de datos se utiliza para garantizar que los datos almacenados en el sistema sean coherentes en todo momento. Esto se logra mediante la ejecución de procesos de verificación regulares que comparan los datos almacenados en el sistema con la fuente de datos original.

Mecanismos de control de consistencia

- Validación de datos
- Reglas de integridad
- Transacciones atómicas
- Verificación de la coherencia de datos
- **Control de versiones**



El control de versiones se utiliza para garantizar que los cambios en los datos se realicen de forma controlada y rastreable. Esto se logra mediante la creación de versiones de los datos y la implementación de un sistema de seguimiento que permita rastrear los cambios realizados en los datos.

Tabla de Contenidos

01

Conceptos Previos

Nociones básicas de transacciones, visión general de los niveles de aislamiento de transacciones en Oracle.

02

Concurrencia de datos

Definición, problemas de concurrencia y mecanismos de gestión.

03

Consistencia de datos

Definición, problemas de consistencia y mecanismos de gestión.

04

Otros aspectos y conclusiones

Técnicas de diseño de un SBD. Influencia en el rendimiento y escalabilidad de la BD.
Conclusiones

04

Otros aspectos y conclusiones

Técnicas de diseño de un SBD. Influencia en el rendimiento y escalabilidad de la BD. Conclusiones

Importancia de la concurrencia y la consistencia en las bases de datos

La concurrencia y la consistencia son dos propiedades fundamentales en las bases de datos que aseguran la integridad, fiabilidad y eficiencia de los datos almacenados

- Integridad de los datos
- Acceso simultáneo a los datos
- Transacciones seguras
- Prevención de bloqueos de datos
- Fiabilidad del sistema



Principales consideraciones a tener en cuenta al diseñar un SBD



Importante considerar el control de concurrencia, transacciones atómicas, diseño de la base de datos, escalabilidad, rendimiento, seguridad, respaldo y recuperación, mantenimiento, documentación y pruebas. Estas consideraciones aseguran que la base de datos sea eficiente, segura y escalable, y que pueda manejar un gran volumen de transacciones concurrentes