

# Tema 6

Concurrencia y consistencia de datos.

# Concurrencia de datos y consistencia en un entorno multiusuario

- En los sistemas monousuario no existen problemas con la integridad de los datos.
- En los sistemas multiusuario aparecen problemas al acceder a los datos de forma concurrente, y surge la necesidad de crear mecanismos para obtener resultados validos y consistentes.

# Concurrencia de datos y consistencia en un entorno multiusuario

- ¿Qué es la concurrencia?

La concurrencia consiste en acceder a un mismo dato varios usuarios a la vez.

- ¿Qué es la consistencia?

La consistencia asegura que los datos que estamos viendo no cambien hasta que acabemos la transacción.

# Concurrencia de datos y consistencia en un entorno multiusuario

- ¿Cómo podemos solucionarlo?

El modelo definido por los investigadores es la serialización, que consiste en simular que las transacciones están siendo ejecutadas de una en una en serie (aislamiento).

Sin embargo el rendimiento del sistema puede verse comprometido.

Por tanto hay que encontrar la combinación aislamiento/rendimiento adecuada.

# Fenómenos prevenibles y niveles de aislamiento de transacción

- El estándar ANSI/ISO SQL (SQL92) define cuatro niveles de aislamiento de transacción con diferentes grados de impacto en el rendimiento de procesamiento de transacciones.
- Estos niveles de aislamiento se definen en términos de tres fenómenos que deba evitarse entre las operaciones que se ejecutan simultáneamente.

# Fenómenos prevenibles y niveles de aislamiento de transacción

- Lectura sucia:

Una transacción lee datos que han sido escritos por otra transacción que no se ha confirmado todavía.

- Lectura no repetible:

Una transacción relee datos que previamente ha leído y descubre que otra transacción confirmada ha modificado o borrado los datos.

- Lectura fantasma:

Una transacción vuelve a ejecutar una consulta que devuelve un conjunto de filas que satisfacen una condición de búsqueda y encuentra que otra transacción confirmada ha insertado nuevas filas que satisfacen la condición.

# Niveles de aislamiento SQL92

*Table 13–1 Preventable Read Phenomena by Isolation Level*

Isolation Level	Dirty Read	Nonrepeatable Read	Phantom Read
Read uncommitted	Possible	Possible	Possible
Read committed	Not possible	Possible	Possible
Repeatable read	Not possible	Not possible	Possible
Serializable	Not possible	Not possible	Not possible

Oracle ofrece el modo de lectura confirmada y niveles serializables de aislamiento, así como un modo de sólo lectura que no es parte de SQL92.

La lectura confirmada es el modo de lectura predeterminada.

# Mecanismos de bloqueo

Los bloqueos o cerraduras son los mecanismos de los cuales nos servimos para impedir la interacción destructiva entre las transacciones que acceden al mismo recurso.





# Mecanismos de bloqueo

Los recursos incluyen dos tipos generales de objetos:

- Objetos de usuario,
  - tablas y filas (estructuras y datos).
- Objetos no visibles para usuarios,
  - estructuras de datos compartidos en las filas de memoria y datos del diccionario.

# Aspectos de la gestión de la concurrencia y consistencia de datos por parte de Oracle :

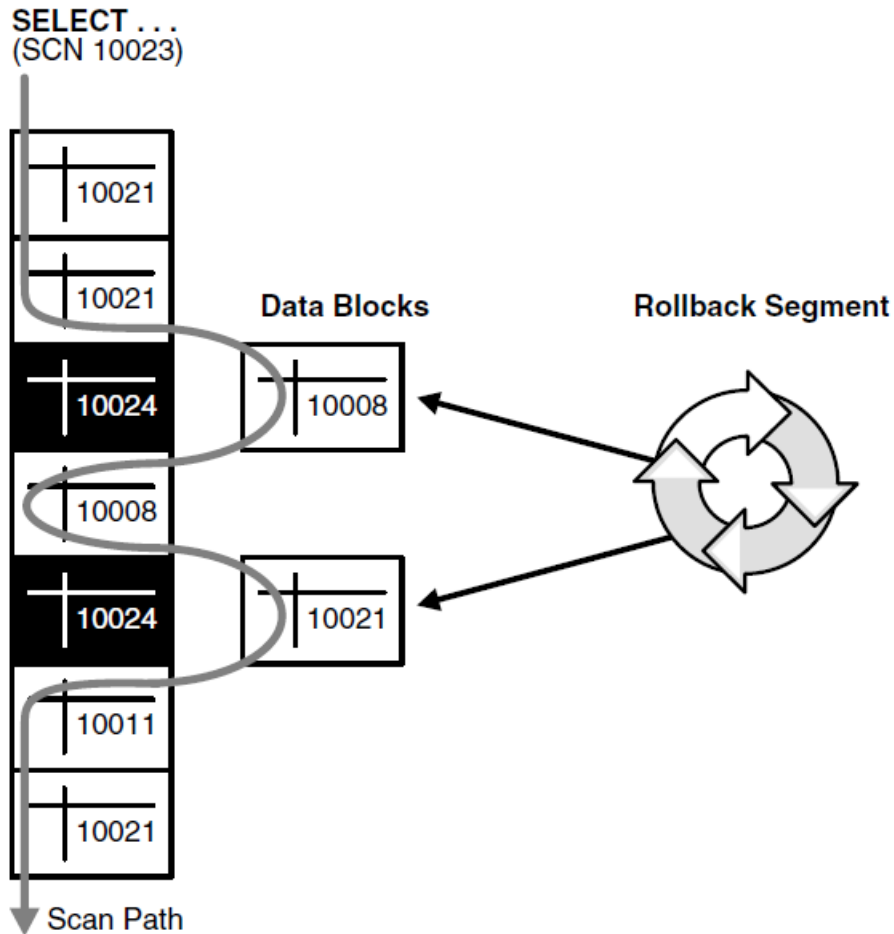
- Control de concurrencia multiversión.
- Declaración de coherencia de lectura.
- Transacción coherencia de lectura.
- Consistencia de lectura con Real Application Clusters (RAC).
- Niveles de aislamiento de Oracle.
- Comparación de lectura confirmada y aislamiento serializable.
- Elección del nivel de aislamiento.

# Control de concurrencia Multiversion

Oracle proporciona automáticamente una coherencia de lectura a una consulta para que todos los datos de la consulta se vea desde un punto único en el tiempo (coherencia de lectura a nivel de instrucción).

Oracle también puede proporcionar una coherencia de lectura a todas las consultas en una transacción (coherencia de lectura a nivel de transacción).

# Control de concurrencia Multiversion



Para poder llevar a cabo la consistencia a nivel de lectura Oracle emplea los segmentos de rollback.

En dichos segmentos se encuentran los viejos valores de los datos que han sido modificados por las transacciones sin confirmar o que se han comprometido recientemente.

# Lectura consistente a nivel de sentencia

- Consistencia de las sentencias.
- Datos devueltos por una consulta responden todos a un solo momento en el tiempo.
- No lecturas sucias.
- La consulta no ve los cambios confirmados después de la ejecución.

# Lectura consistente a nivel de transacción

- Oracle ofrece la opción de consistencia de lectura a nivel de transacción.
- Cuando se ejecuta una transacción en modo serializable, todos los accesos de datos reflejan el estado de la base de datos como en el momento en el que la transacción comenzó.
- No produce lecturas no repetibles ni fantasmas.

# Niveles de aislamiento en Oracle

- Lectura confirmada:
  - Nivel de aislamiento por defecto.
  - NO lecturas sucias.
  - Posibilidad de lectura no repetible y fantasmas.
- Serialización:
  - Cambios justo cuando la transacción comenzó.
  - Cambios hechos por la propia transacción → INSERT, UPDATE y DELETE.
  - NO lecturas no repetibles ni fantasmas.
- Solo lectura:
  - Cambios justo cuando la transacción comenzó.
  - No permite la declaración de INSERT, UPDATE o DELETE.

# Configuración de los niveles de aislamiento

Para una sola transacción:

- *SET TRANSACTION ISOLATION LEVEL READ COMMITTED;*
- *SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;*
- *SET TRANSACTION READ ONLY;*



# Configuración de los niveles de aislamiento

Para todas las transacciones posteriores:

- SET TRANSACTION → ALTER SESSION:
  - *ALTER SESSION SET ISOLATION\_LEVEL SERIALIZABLE;*
  - *ALTER SESSION SET ISOLATION\_LEVEL READ COMMITTED;*

### *Read Committed and Serializable Transactions*

	Read Committed	Serializable
Dirty write	Not possible	Not possible
Dirty read	Not possible	Not possible
Nonrepeatable read	Possible	Not possible
Phantoms	Possible	Not possible
Compliant with ANSI/ISO SQL 92	Yes	Yes
Read materialized view time	Statement	Transaction
Transaction set consistency	Statement level	Transaction level
Row-level locking	Yes	Yes
Readers block writers	No	No
Writers block readers	No	No
Different-row writers block writers	No	No
Same-row writers block writers	Yes	Yes
Waits for blocking transaction	Yes	Yes
Subject to cannot serialize access	No	Yes
Error after blocking transaction terminates	No	No
Error after blocking transaction commits	No	Yes

# Bloqueo a nivel de fila

- Se realiza en ambos niveles de aislamiento.
- Cuando una transacción utiliza una fila la bloquea hasta su confirmación.
- Si otra transacción quiere utilizarla tendrá que esperar hasta que se confirme o se deshaga y se libere el bloqueo.

# Bloqueo a nivel de fila

La transacción que bloquea la fila termina con:	Transacciones modo lectura confirmada	Transacciones modo serializable
Rollback	Puede actualizar la fila	Puede actualizar la fila
Commit	Puede actualizar la fila	Error “Cannot serialize access”

- Cannot serialize access:
  - Se quiere acceder a una fila que ha sido modificada y confirmada después de comenzar la transacción

# Integridad referencial

- Oracle no realiza bloqueos de lectura, por tanto, un dato leído puede ser modificado por otra transacción.
- Las inconsistencias de datos pueden sucederse en la BD.
- Será el desarrollador quien garantice la integridad referencial.

# Transacciones distribuidas

- Definición:
  - En un ambiente de BD distribuida, una transacción modifica datos en múltiples BD físicas.
- Los servidores que participan en transacciones serializables, deben soportar TODOS el modo de aislamiento serializable.
- Los servidores que participan en transacciones de lectura confirmada, NO tienen que soportar el modo de aislamiento serializable.

# Elección de un nivel de aislamiento

- Dependerá de:
  - El eficiencia de ejecución de las aplicaciones,
  - la consistencia de datos necesitada y
  - requerimientos de las aplicaciones.
- Oracle proporciona altos niveles de consistencia, concurrencia y rendimiento.
- Analizaremos para que situaciones conviene usar uno u otro.

# Modo de lectura confirmada

- Proporciona una mayor concurrencia, pero con mayor riesgo por las transacciones fantasma y las lecturas no repetibles.
  - Entornos de alto rendimiento con elevados ratios de llegada de transacciones.
  - Entornos con un muy bajo ratio de llegada de transacciones.



# Modo de lectura confirmada

- Para muchas aplicaciones, no es necesario el desarrollo de lógicas que protejan de errores.
  - No hay que controlar el “**Cannot serialize access error**”.
  - En muchas situaciones, es muy difícil la aparición de lecturas fantasmas y lecturas no repetibles.

# Modo serializable

- Es más adecuado para ambientes con grandes bases de datos y pequeñas transacciones que actualicen sólo pocas filas.
- Proporciona una mayor consistencia al proteger contra lecturas fantasmas y lecturas no repetibles.

# Inmovilizar la base de datos.

- El sistema está inmovilizado si no hay sesiones activas, excepto SYS y SYSTEM.
- Los administradores de la BD son los únicos que pueden realizar operaciones con la BD en estado inmovilizado, que de otra forma no podrían ser realizadas de forma segura.
  - Operaciones que:
    - Fallarían si hay transacciones concurrentes.
    - Cuyo efecto intermedio podría ser perjudicial para las transacciones o consultas de usuarios concurrentes.

# Inmovilizar la base de datos.

- Para los sistemas que deben operar continuamente, es crítica la habilidad de realizar tales acciones sin apagar la base de datos.
- El gestor bloquea todas las acciones de otros usuarios que no sean SYS o SYSTEM y volverán a operar cuando la BD vuelva a sus estado normal.
- Los usuarios no recibirán ningún mensaje de error adicional por estar la BD imovilizada.

# Proceso de inmovilización de la base de datos.

*Administrador de BD*

“ALTER SYSTEM QUIESCE RESTRICTED”

*Gestor de Recursos de la BD*

Prohíbe que se activen sesiones distintas de SYS y SYSTEM

Espera por todas las transacciones iniciadas por usuarios no privilegiados

↓  
BASE DE DATOS INMOVILIZADA

*Gestor de Recursos de la BD*

↓  
Si la instancia se ejecuta en modo servidor compartido, se bloquean los posibles logins

# Inmovilizar la base de datos.

- Durante el estado de inmovilización, no se podrá cambiar el plan de Gestión de Recursos de ninguna forma.
- *ALTER SYSTEM UNQUIESCE*: Cambia el estado de todas las instancias al estado normal.

# Bloqueos en Oracle

- Los bloqueos (locks) son mecanismo que impiden que dos transacciones accedan al mismo recurso:
  - Objetos visibles
    - tablas
    - filas
  - Objetos no visibles
    - estructuras de datos compartidos de memoria
    - datos del diccionario.
- Son automáticos según se ejecutan las instrucciones SQL, al no requerir ninguna acción especial, el usuario no debe preocuparse de ellos.
- Proporcionan la concurrencia de datos y la integridad entre las transacciones.

# Modos de Bloqueo

- Oracle utiliza dos modos de bloqueo :
  - Exclusive Lock:
    - Se obtienen para modificar los datos (operaciones de escritura) y para prevenir que los recursos asociados sean compartidos.
  - Share Lock:
    - Dependiendo de las operaciones involucradas, permite que el recurso asociado sea compartido.
    - Varios usuarios pueden compartir datos de lectura, por lo que diversas transacciones pueden adquirir bloqueos compartidos del mismo recurso.



# Duración de los Bloqueos

- Los bloqueos realizados por una instrucción, se mantienen hasta la finalización de la transacción en la que se encuentra.
- Al confirmar o deshacer una transacción, Oracle libera todos los bloqueos realizados por sus instrucciones.
- También libera los bloqueos realizados después de un Savepoint, cuando se ejecuta un Rollback a dicho Savepoint.

# Deadlock

- Puede darse cuando dos o más usuarios esperan por los datos bloqueados entre ellos.
- Son situaciones que impiden que las transacciones continúen trabajando.
- Oracle detecta automáticamente las situaciones de deadlock y las resuelve haciendo un Rollback en uno de los procesos (instrucción o transacción) involucrados, liberando sus bloqueos y manda un mensaje a la transacción.

ORA-00060: deadlock detected while waiting for resource

# Evitar los Deadlocks

- No existen recetas infalibles para evitarlos, sino buenas prácticas.
- No son un defecto de Oracle, responde a carencias de conocimiento de los mecanismos de bloqueo por parte del código ejecutado por múltiples sesiones que comparten recursos concurrentes.

# Tipos de Bloqueos

## 1. DML Locks:

- protegen los datos
  - bloqueo de tabla.

## 2. DDL Locks:

- protegen las estructuras de los objetos del sistema
  - definiciones de tablas y vistas.

## 3. Internal Locks y latches:

- protegen la estructura interna de la BD así como los archivos de datos.
- Son totalmente automáticos

# Bloqueo por defecto para INSERT, UPDATE, DELETE y SELECT ... FOR UPDATE

- Las características de bloqueo de estas declaraciones son las siguientes:
  - La transacción que contenga una instrucción DML adquiere bloqueos exclusivos de fila, en las filas que modifica la instrucción.
  - La transacción que contenga una instrucción DML no necesita adquirir bloqueos de fila en las filas seleccionadas por una subconsulta o una consulta implícita, como una consulta en la cláusula WHERE.

# Bloqueos DDL (DDL Locks)

- Un bloqueo del diccionario de datos protege a la definición de un objeto de esquema, mientras que una operación en curso DDL actúa sobre o se refiere al objeto.
- Oracle adquiere un bloqueo del diccionario de manera automática en nombre de cualquier transacción DDL que lo requiera.
- Los bloqueos DDL evitan que los objetos referenciados en el procedimiento se modifiquen o se caigan antes de que la compilación del procedimiento esté completa.

# Bloqueos DDL (DDL Locks)

- Los bloqueos DDL se dividen en tres categorías:
  - Bloqueos exclusivos DDL (Exclusive DDL Locks)
  - Bloqueos compartidos DDL (Share DDL Locks)
  - Bloqueos de análisis frágil (Breakable Parse Locks)

# Duración de los bloqueos DDL

- La duración de un bloqueo DDL depende de su tipo:
  - El bloqueo exclusivo y el compartido duran la ejecución de la sentencia DDL y la confirmación automática (commit).
  - Un bloqueo de análisis frágil persiste mientras que la sentencia asociada SQL permanece en la zona compartida.



# Cierres y bloqueos internos (Latches and Internal Locks)

- Protegen la base de datos interna y estructuras de memoria.
- Ambos son inaccesibles para los usuarios.

# Cierres (Latches)

- Los cierres son simples, mecanismos de bajo nivel de serialización empleados para proteger las estructuras de datos compartidos en el área global del sistema (SGA).
- La implementación de los cierres depende del sistema operativo, en particular en lo que respecta a cuándo y cuánto tiempo un proceso esperará un cierre.

# Bloqueos internos (Internal Locks)

- Los bloqueos internos son:
  - De más alto nivel.
  - Mecanismos más complejos que los cierres.
  - Sirven una variedad de propósitos.
    - Bloqueos del diccionario cache.
    - Bloqueos de archivo y de log.
    - Tablespace and rollback segment locks.

# EXPLICIT (MANUAL) DATA LOCKING

- Oracle usa cerrojos de manera automática.
- Es posible sobrescribir estos mecanismos.
- Es útil en las siguientes ocasiones:
  - Aplicaciones que necesitan producir datos sin reflejar cambios hechos por otras transacciones.
  - Necesidad de una aplicación de tener acceso exclusivo a algún recurso (sin esperar).

# BLOQUEO DE GESTIÓN DE SERVICIOS

- Permite a desarrolladores incluir declaraciones en bloques PL/SQL para:
  - Solicitar un bloqueo de tipo específico.
  - Dar el bloqueo de un unico nombre reconocible en otro procesimiento en el mismo o en otro caso.
  - Cambiar el tipo de bloqueo.
  - Liberar bloqueo.
- Éstos cerrojos heredan las funcionalidades de los cerrojos de Oracle permitiendo la detección de deadlock y no entran en conflicto.
- Todos estos servicios se encuentran el paquete DBMS\_LOCK.

# ORACLE FLASHBACK QUERY

- Utilizando multiversion Oracle read-consistency proporciona la capacidad de restaurar datos mediante la aplicación de deshacer.
  - Para ello sintoniza de forma automática un parámetro llamado "período de retención de deshacer" y recoge estadística de uso.
- La velocidad de ésta operación sólo depende del tamaño de los datos y el numero de cambios que se hayan hecho sobre éstos.
- Permite auditar las filas de una tabla obteniendo información sobre la transacción que cambio las filas y el momento en el que fue cambiado.