



Universidad de Guadalajara

Alumno: Sánchez Gómez Edgardo Enrique

Código: 218401479

Asignatura: Administración de Bases de Datos

Práctica 10: Concurrencia y bloqueos en Oracle

Fecha de entrega: 24/Marzo/2024

A) INVESTIGACION:

Realice una investigación sobre bloqueos en oracle incluyendo los siguientes tipos de bloqueos, adjunte un archivo de la investigación

HECHA A MANO cubriendo el documento adjunto completamente:

- **Introduction to Data Concurrency and Consistency in a Multiuser Environment**
- **How Oracle Manages Data Concurrency and Consistency**
- **How Oracle Locks Data**
- **Overview of Oracle Flashback Query**

Mínimo 10 cuartillas.

Concurrencia y bloqueos en Oracle.

► Introducción a la Concurrencia y Consistencia de Datos en un Entorno Multiusuario:

En una base de datos de un solo usuario, el usuario puede modificar los datos en la base de datos sin preocuparse de que otros usuarios modifiquen los mismos datos al mismo tiempo. Sin embargo, en una base de datos multiusuario, las declaraciones dentro de múltiples transacciones simultáneas pueden actualizar los mismos datos. Las transacciones que se ejecutan al mismo tiempo deben producir resultados significativos y consistentes. Por lo tanto, el control de la concurrencia de datos y la consistencia de los datos es vital en una base de datos multiusuario.

- Concurrencia de datos significa que muchos usuarios pueden acceder a los datos al mismo tiempo.
- Consistencia de los datos significa que cada usuario ve una vista coherente de los datos, incluidos los cambios visibles realizados por las transacciones propias del usuario y las transacciones de otros usuarios.

Para describir el comportamiento de transacción consistente cuando las transacciones se ejecutan al mismo tiempo, los investigadores de bases de datos han definido un modelo de aislamiento de transacciones llamado serialización. El modo serializable de comportamiento de la transacción trata de garantizar

que las transacciones se ejecuten de tal manera que parezcan ejecutarse ^{una} a la vez, o en serie, en lugar de simultáneamente.

Si bien este grado de aislamiento entre las transacciones es generalmente deseable, ejecutar muchas aplicaciones en este modo puede comprometer seriamente el rendimiento de la aplicación. El aislamiento completo de las transacciones que se ejecutan simultáneamente podría significar que otra transacción no puede realizar un inserto en la tabla consultada por otra transacción.

En resumen, las consideraciones del mundo real generalmente requieren un compromiso entre el aislamiento perfecto de la transacción y el rendimiento.

Oracle ofrece dos niveles de aislamiento, proporcionando a los desarrolladores de aplicaciones modos operativos que preservan la consistencia y proporcionan un alto rendimiento.

• Fenómenos prevenibles:

+ Dirty reads: Una transacción lee datos que han sido escritos por otra transacción que aún no se ha confirmado (`commit`).

+ Nonrepeatable (fuzzy) reads: Una transacción vuelve a leer los datos que ha leído previamente y encuentra que otra transacción comprometida ha modificado o eliminado los datos.

+ Phantom reads (o phantoms): Una transacción vuelve a ejecutar una consulta que devuelve un conjunto de filas que satisface una condición de búsqueda y encuentra que otra transacción confirmada ha insertado filas adicionales que satisfacen la condición.

• Descripción general de los mecanismos de bloqueo:

En general, las bases de datos multiusuario utilizan algún tipo de bloqueo de datos para resolver los problemas asociados con la concurrencia de datos, la coherencia y la integridad. "Cerraduras" son mecanismos que impiden la interacción destructiva entre transacciones que acceden al mismo recurso.

Los recursos incluyen dos tipos generales de objetos:

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

► ¿Cómo gestiona Oracle la concurrencia y consistencia de datos?

Lo realiza en entornos de bases de datos multiusuario mediante un modelo de consistencia multiversión, junto con varios tipos de bloqueos y transacciones. Este enfoque permite que múltiples usuarios accedan y modifiquen datos simultáneamente, manteniendo al mismo tiempo una vista coherente y significativa de los datos para cada usuario.

• Concurrency de datos:

Se refiere a la capacidad de permitir que muchos usuarios accedan a los datos al mismo tiempo. Oracle logra una alta concurrencia mediante el uso de bloqueos a nivel de fila y su sistema de control de concurrencia multiversion. Este sistema asegura que las operaciones de lectura y escritura no se bloqueen entre sí, permitiendo así un acceso concurrente eficiente a los datos.

• Consistencia de datos:

La consistencia de datos asegura que cada usuario vea una vista coherente de los datos, incluidos los cambios realizados por sus propias transacciones y las transacciones de otros usuarios. Oracle proporciona consistencia de lectura a nivel de sentencia automáticamente, lo que significa que todos los datos que una consulta ve, provienen de un único punto en el tiempo. Además, Oracle puede proporcionar consistencia de lectura a nivel de transacción, asegurando que todas las consultas dentro de una transacción vean los datos de manera coherente con respecto a un único punto en el tiempo.

• Modos de aislamiento:

Oracle ofrece dos modos de aislamiento, proporcionando a los desarrolladores de aplicaciones con operatividades que preservan la consistencia y ofrecen rendimiento elevado.

Estos modos de aislamiento son el nivel de aislamiento "read committed" y "serializable", los cuales permiten un alto grado de concurrencia sin sacrificar la consistencia de los datos.

En particular, el modo serializable asegura que todas las lecturas dentro de una transacción sean consistentes con respecto al inicio de la transacción, mientras que el modo "read committed" proporciona una vista coherente de los datos comprometidos antes de que comenzara cada consulta.

• Bloqueos y control de concurrencia:

Oracle utiliza diversos tipos de bloqueos para gestionar el acceso concurrente a los datos y mantener la integridad de los mismos. Los bloqueos de datos, bloqueos DDL y bloqueos internos son mecanismos que previenen la interacción destructiva entre transacciones que acceden al mismo recurso.

Oracle asigna automáticamente los bloqueos necesarios durante la ejecución de sentencias SQL, utilizando el nivel menos restrictivo de bloqueo necesario para mantener la consistencia y la concurrencia.

• Gestión de la consistencia con segmentos de deshacer:

Oracle utiliza la información almacenada en los segmentos de deshacer para proporcionar vistas consistentes de los datos. Los segmentos de deshacer contienen los valores antiguos de los datos que han sido modificados por transacciones no confirmadas o recientemente confirmadas. Esto permite a Oracle reconstruir versiones anteriores de los datos para consultas en ejecución, asegurando que cada consulta devuelva datos

consistentes con respecto al número de cambio del sistema {SCN} registrado al inicio de la ejecución de la consulta.

► ¿Cómo bloquea Oracle los datos?

Oracle utiliza diversos tipos de bloqueos para garantizar la concurrencia y la integridad de los datos en entornos de bases de datos multiusuario.

Estos bloqueos se dividen en dos categorías principales:

- Bloques DML {diseñados para proteger datos}
 - Bloques DDL {diseñados para proteger la estructura de objetos de esquema.
-
- Bloques DML {Data Manipulation Language}:
Son utilizados para proteger datos en tablas y filas.
Existen dos tipos de bloque DML:
 - Bloques de tabla: Se utilizan para proteger entidades completas, como tablas. Estos bloques se utilizan para garantizar que ninguna transacción pueda modificar la tabla hasta que la transacción actual termine su operación.
 - Bloques de fila: Los bloqueos de fila se utilizan para proteger las filas individuales en una tabla. Estos bloqueos se utilizan para garantizar que ninguna transacción pueda modificar una fila hasta que la transacción actual termine su operación.
 - Bloques DDL {Data Definition Language}:
Los bloqueos DDL se utilizan para proteger la estructura

de objetos de esquema, como procedimientos y tablas. Estos bloqueos se utilizan para garantizar que ninguna transacción pueda modificar el esquema de un objeto hasta que la transacción actual termine su operación.

• Tipos de bloqueos:

- **Bloqueos exclusivos:** Se utilizan para proteger recursos que requieren modificaciones, como tablas y esquemas. Solo una transacción puede obtener un bloqueo exclusivo a la vez, lo que garantiza que ningún otra transacción pueda modificar el recurso mientras la transacción actual lo tiene bloqueado.

- **Bloqueos compartidos:** Se utilizan para proteger recursos que pueden ser leídos por varias transacciones al mismo tiempo.

Estos bloqueos se utilizan para garantizar que varias transacciones puedan leer un recurso sin interferirse entre sí.

- **Bloqueo automático:**

Oracle realiza el bloqueo automáticamente para garantizar la concurrencia y la integridad de los datos. Los bloques se liberan automáticamente cuando una transacción termina su operación, lo que permite a otras transacciones acceder al recurso.

- **Bloqueo de esquemas:**

Oracle utiliza bloques de esquema para proteger la estructura

de objetos de esquema, como procedimientos y tablas. Estos bloques se utilizan para garantizar que ninguna transacción pueda modificar el esquema de un objeto hasta que la transacción actual termine su operación.

- **Bloqueo de bases de datos:**

Protegen una base de datos completa; estos bloques se utilizan para garantizar que ninguna transacción pueda modificar una base de datos hasta que la transacción actual termine su operación.

- **Deadlocks:**

Ocorre cuando dos o más transacciones se bloquean mutuamente, esperando que la otra libere sus bloqueos. Oracle detecta automáticamente las situaciones que pueden generarlas y resuelve los deadlocks liberando uno de los bloqueos, lo que permite que las transacciones restantes continúen.

La transacción cuyo bloqueo fue liberado recibe un error y debe reintentar su operación.

- **Duración de los bloqueos:**

Los bloqueos se mantienen hasta que la transacción que los posee se compromete o se deshace. Esto asegura que los cambios realizados por la transacción sean atómicos y consistentes.

Oracle ha diseñado su sistema de bloqueo para maximizar la concurrencia mientras mantiene la integridad de los datos. Los bloqueos son fundamentales para el manejo de transacciones

en bases de datos Oracle, y el sistema está optimizado para manejarlos de manera eficiente y efectiva.

► Visión general de Oracle Flashback Query:

Oracle Flashback Query es una característica de Oracle Database que permite a los usuarios consultar versiones pasadas de los datos en la base de datos, sin necesidad de restaurar los datos desde una copia de seguridad. Esta funcionalidad es útil para recuperar información perdida o modificada accidentalmente, realizar análisis forenses de los datos y simplificar las tareas de recuperación de aplicaciones.

• Características principales:

+ Acceso a datos históricos:

Flashback Query permite a los usuarios ver el estado de los datos en un punto anterior en el tiempo, utilizando una sintaxis de consulta SQL simple. Esto es posible gracias a la capacidad de Oracle de mantener y gestionar versiones históricas de los datos en la base de datos.

+ Simplicidad de uso:

La tecnología Flashback Query se integra directamente con el lenguaje SQL estándar, lo que significa que los usuarios pueden utilizar comandos familiares para acceder a versiones anteriores de los datos. No se requiere una configuración compleja o herramientas adicionales.

+ Recuperación de datos:

Una de las aplicaciones más valiosas de Flashback Query es la capacidad de recuperar datos que han sido borrados

o modificados accidentalmente. Los usuarios pueden identificar rápidamente los cambios no deseados y restaurar los datos a su estado anterior.

+ Análisis de cambios:

Flashback Query también facilita el análisis de cómo han cambiado los datos a lo largo del tiempo, permitiendo a los usuarios rastrear modificaciones, identificar tendencias o realizar auditorías de cambios en los datos.

• Requisitos y configuración:

Para utilizar Oracle Flashback Query, la base de datos Oracle debe estar configurada para mantener información suficiente sobre las versiones anteriores de los datos. Esto generalmente implica configurar y gestionar adecuadamente el área de almacenamiento de deshacer (undo space) de Oracle, que es donde se almacenan los datos necesarios para reconstruir versiones anteriores de los datos.

• Consideraciones:

+ Rendimiento: Puede tener implicaciones en el rendimiento de la base de datos, especialmente si se accede a datos históricos de periodos largos.

Es importante planificar y gestionar adecuadamente los recursos de la base de datos para minimizar el impacto.

+ Limitaciones de tiempo: La capacidad de acceder a datos históricos está limitada por la cantidad de información de deshacer que la base de datos Oracle mantiene. Los administradores de bases de datos deben configurar cuidadosamente estos parámetros según las necesidades de recuperación de datos y los requisitos de retención de información histórica.

B) PRACTICA:

Genere cuando menos dos tipos de bloqueos en la base de datos, para evidenciarlos deberá consultar las vistas de locks en oracle, adjunte imágenes de la consulta de vistas de bloqueos en sqlplus.

1. Acceder al usuario privilegiado (Terminal 1), y desde otras 2 terminales acceder al usuario mortal (Terminal 2 y Terminal 3).
2. Desde una sesión del usuario mortal (Terminal 3) ejecutamos una consulta que activa un bloqueo de transacción, en este caso en la tabla alumnos.

Terminal 1 (Left):

```
SQL> select * from v$lock;
```

SID	TY	LOCK_TYPE	ID1	ID2

OBJECT_NAME				

USERNAME				

129	TM	Table Lock	74189	0
ALUMNOS				
EDGARDOM				
98	TM	Table Lock	74189	0
ALUMNOS				
EDGARDOM				

SID	TY	LOCK_TYPE	ID1	ID2

OBJECT_NAME				

USERNAME				

126	CF	Control File Lock	0	0

129	TX	Transaction Lock	524308	1635

SID	TY	LOCK_TYPE	ID1	ID2

OBJECT_NAME				

USERNAME				

EDGARDOM				

SQL>				

Terminal 2 (Top Right):

```
SQL> lock table alumnos in exclusive mode;
```

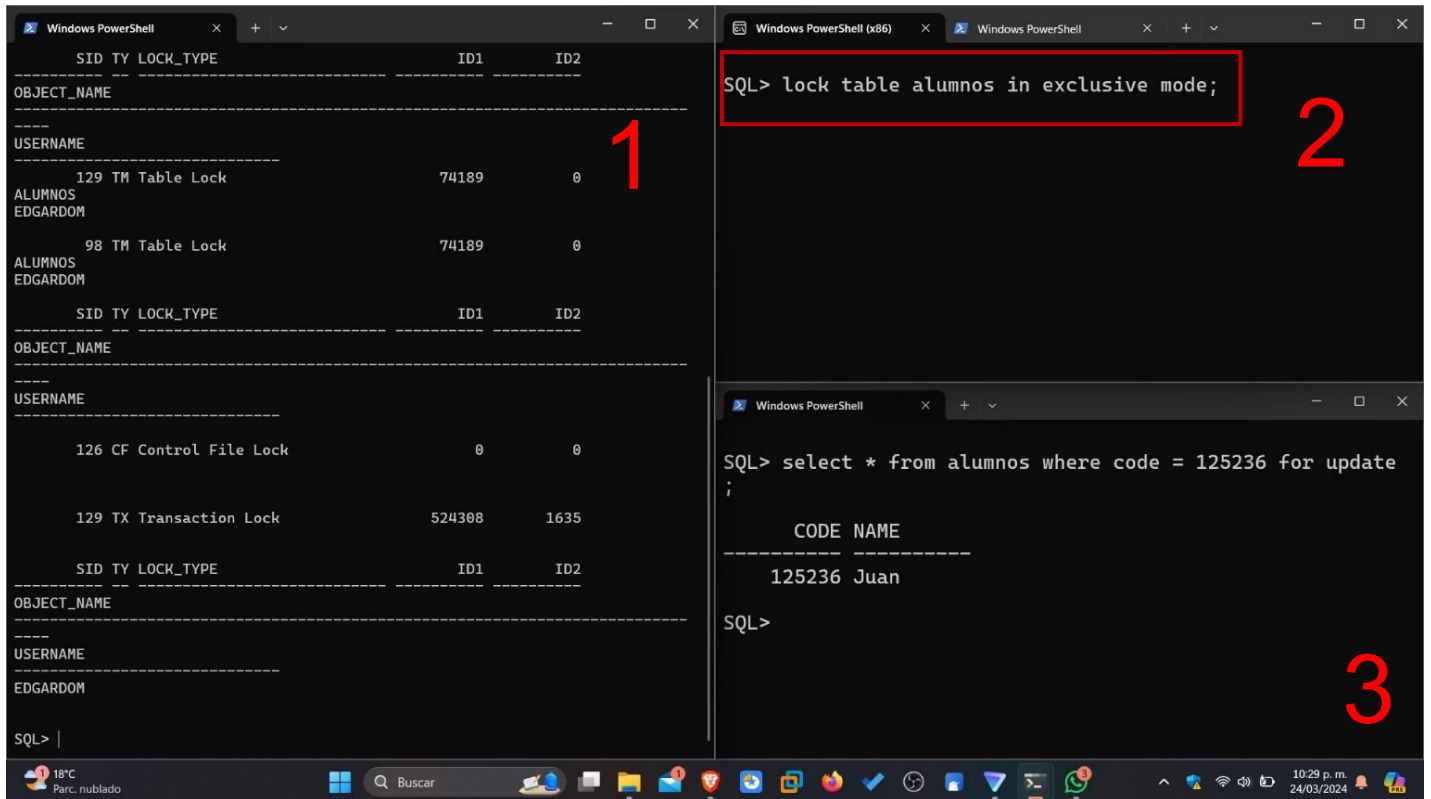
Terminal 3 (Bottom Right):

```
SQL> select * from alumnos where code = 125236 for update
;
```

CODE	NAME
125236	Juan

SQL>

3. Desde la terminal con la otra sesión mortal (Terminal 2) intentamos hacer un bloqueo exclusivo en la misma tabla que usamos en el paso anterior.
NOTA: Notamos que queda la transacción en espera, debido al bloqueo generado.



4. En la terminal con el super usuario (Terminal 1) ejecutamos un query que nos otorga los bloqueos activos en la base de datos:

```
SELECT s.sid,
       l.type,
       DECODE(l.type,
              'TM', 'Table Lock',
              'TX', 'Transaction Lock',
              'UL', 'User Lock',
              'DL', 'DDL Lock',
              'BL', 'Block Lock',
              'DX', 'Distributed Transaction Lock',
              'CF', 'Control File Lock',
              'IS', 'Instance State Lock',
              'PI', 'Parallel DML Lock',
              'PR', 'Profile Lock',
              'Unknown') AS lock_type,
       l.id1,
       l.id2,
       o.object_name,
```

```

s.username
FROM v$lock l
LEFT JOIN dba_objects o ON l.id1 = o.object_id
LEFT JOIN v$session s ON s.sid = l.sid
WHERE l.type IN ('TM', 'TX', 'UL', 'DL', 'BL', 'DX', 'CF', 'IS', 'PI',
'PR');

```

The image shows three Windows PowerShell windows running SQL*Plus commands. A red box labeled '1' highlights the output of the query from the first window, showing lock information for user EDGARDOM. A red box labeled '2' shows the command to lock the 'alumnos' table. A red box labeled '3' shows the command to update a record in the 'alumnos' table.

1

SID	TY	LOCK_TYPE	ID1	ID2
129	TM	Table Lock	74189	0
98	TM	Table Lock	74189	0
126	CF	Control File Lock	0	0
129	TX	Transaction Lock	524308	1635

2

```

SQL> lock table alumnos in exclusive mode;

```

3

```

SQL> select * from alumnos where code = 125236 for update
;

```

CODE	NAME
125236	Juan

C) DEADLOCK:

Generar un interbloqueo

1. Acceder al usuario privilegiado (Terminal 1), y desde otras 2 terminales acceder al usuario mortal (Terminal 2 y Terminal 3).
2. Desde una sesión del usuario mortal (Terminal 1) ejecutamos una consulta que activa un bloqueo de transacción, en este caso en la tabla alumnos, en la fila con id 1.
3. Desde una sesión del usuario mortal (Terminal 2) ejecutamos una consulta que activa un bloqueo de transacción, en este caso en la tabla alumnos, en la fila con id 2.
4. Desde la Terminal 1 intentamos hacer lo mismo para el id 2, y desde la Terminal 2 para el id 1.

NOTA: Ambas consultas quedarán esperando ya que ocupan su recurso mutuamente; sin embargo, gracias a la configuración de Oracle, detectará un interbloqueo y liberará uno de los dos, arrojando un mensaje de error.

The screenshot displays three Windows PowerShell windows illustrating a deadlock scenario in Oracle:

- Terminal 1 (Left):** Shows the user 'EDGARDOM' and a table lock on 'ALUMNOS' for ID 1. The lock is held by the user 'EDGARDOM'.
- Terminal 2 (Top Right):** Shows the user 'Pedro' and an error message 'ORA-00060: detectado interbloqueo mientras se esperaba un recurso' after attempting to update ID 2. The error message is in Spanish.
- Terminal 3 (Bottom Right):** Shows the user 'Juan' and a successful update of ID 1 after a rollback. The update is successful.

Bibliografía:

Database concepts. (s. f.).

https://docs.oracle.com/cd/B19306_01/server.102/b14220/consist.htm