

CONCURRENCIA DE DATOS Y CONSISTENCIA

-ADMINISTRACIÓN DE BASES DE DATOS-



Pablo Cordon Hidalgo
Javier Fernández Olivo

ÍNDICE

1.	Introducción a la Concurrency de Datos y Consistencia.....	2
1.1.	Consistencia de lectura multiversional.....	3
1.1.1.	Consistencia de lectura a nivel de transacción.....	3
1.1.2.	Consistencia de lectura y segmentos de deshacer.....	3
2.	Niveles de aislamiento de transacción de Oracle.....	5
2.1.	Nivel de aislamiento de lectura confirmada.....	5
2.1.1.	Conflictos de escritura en transacciones de lecturas confirmadas.....	5
2.2.	Nivel de aislamiento serializable.....	6
2.3.	Nivel de aislamiento de solo lectura.....	7
3.	Descripción general del Mecanismo de bloqueo en Oracle.....	7
3.1.	Modos de bloqueo.....	8
3.2.	Duración del bloqueo.....	8
3.3.	Bloqueos e interbloqueos (DeadLock).....	8
3.4.	Bloqueos DML.....	9
3.4.1.	Bloqueos de fila.....	9
3.4.2.	Bloqueos de tablas.....	10
3.4.3.	Bloqueos y claves ajenas (Foreign Keys).....	11
3.5.	Bloqueos DDL.....	13
3.6.	Bloqueos de Sistema.....	13
4.	Bloqueos de datos manuales.....	14
5.	Bloqueos definidos por el usuario.....	15
6.	Conclusiones.....	15
7.	Bibliografía.....	16

1. INTRODUCCIÓN A LA CONCURRENCIA DE DATOS Y CONSISTENCIA

En el mundo de la informática en general, y más concretamente en el ámbito de las bases de datos, tenemos dos tipos de sistema: Monousuario y Multiusuario.

En los sistemas Monousuario, como su propio nombre indica, es un sólo usuario el que puede acceder y modificar la Base de Datos cada vez, sin preocuparse de que nadie más esté trabajando con esta al mismo tiempo.

En los sistemas Multiusuario, sin embargo, varios usuarios pueden trabajar con la Base de Datos al mismo tiempo, generando consultas, modificando o añadiendo nuevos datos... Esto puede generar ciertos problemas si no se controla adecuadamente.

Las transacciones que se ejecutan simultáneamente deben producir resultados consistentes. Es por esto que una correcta base de datos multiusuario debe proporcionar lo siguiente:

- **Concurrencia de datos:** Asegura que los usuarios pueden acceder a los datos al mismo tiempo
- **Consistencia de datos:** Asegura que los usuarios pueden ver una vista consistente de los datos y los cambios realizados mediante las transacciones de los mismos

Es a raíz de esto que surge el concepto de **serializabilidad**:

“los efectos de un conjunto de transacciones simultáneas deben dar el mismo resultado que si se ejecutan en serie las transacciones individuales y que si cada una de las transacciones tuviera un uso exclusivo del sistema.”

Si bien este grado de aislamiento entre transacciones es generalmente deseable, si se ejecutan muchas transacciones en modo serializable el rendimiento de la aplicación puede comprometerse seriamente.

En resumen, las consideraciones del mundo real generalmente requieren un compromiso entre un perfecto aislamiento y un buen rendimiento de las transacciones.

1.1. CONSISTENCIA DE LECTURA MULTIVERSIONAL

En Oracle Database, la multiversionalidad, es la capacidad de materializar múltiples versiones de datos simultáneamente. Las principales características de esta consistencia son:

- ❑ Consultas de lectura consistente: Los datos devueltos por dicha consulta son consistentes y están confirmados con respecto a un único punto en el tiempo.
- ❑ Consultas sin bloqueo: Los lectores y escritores de datos no se bloquean entre sí

1.1.1 CONSISTENCIA DE LECTURA A NIVEL DE TRANSACCIÓN

Oracle Database también puede proporcionar coherencia de lectura a todas las consultas en una transacción. En este caso, cada instrucción en una transacción observa datos en el mismo momento en que la transacción comenzó.

Las consultas realizadas por una transacción serializable observan los cambios realizados por la propia transacción. Un pequeño ejemplo sería; una transacción que actualiza una tabla “empleados” y luego cualquier consulta hacia esa tabla podrá ver dicha actualización.

1.1.2 CONSISTENCIA DE LECTURA Y SEGMENTOS DE DESHACER

Para manejar el modelo de consistencia de la lectura multiversion, la base de datos debe crear tipos de datos de lectura consistente cuando una tabla es consultada y actualizada simultáneamente.

Concretamente en una base de datos Oracle esto se consigue haciendo que una vez un usuario modifica algún dato, la base de datos crea “undo entries” o “entradas de deshacer”, que son escritas a “undo segments” o “segmentos de

deshacer". Estos segmentos contienen los antiguos valores de los datos que han sido cambiados por las transacciones recientes. Creándose así, múltiples versiones de los mismos datos, todos en diferentes momentos en el tiempo.

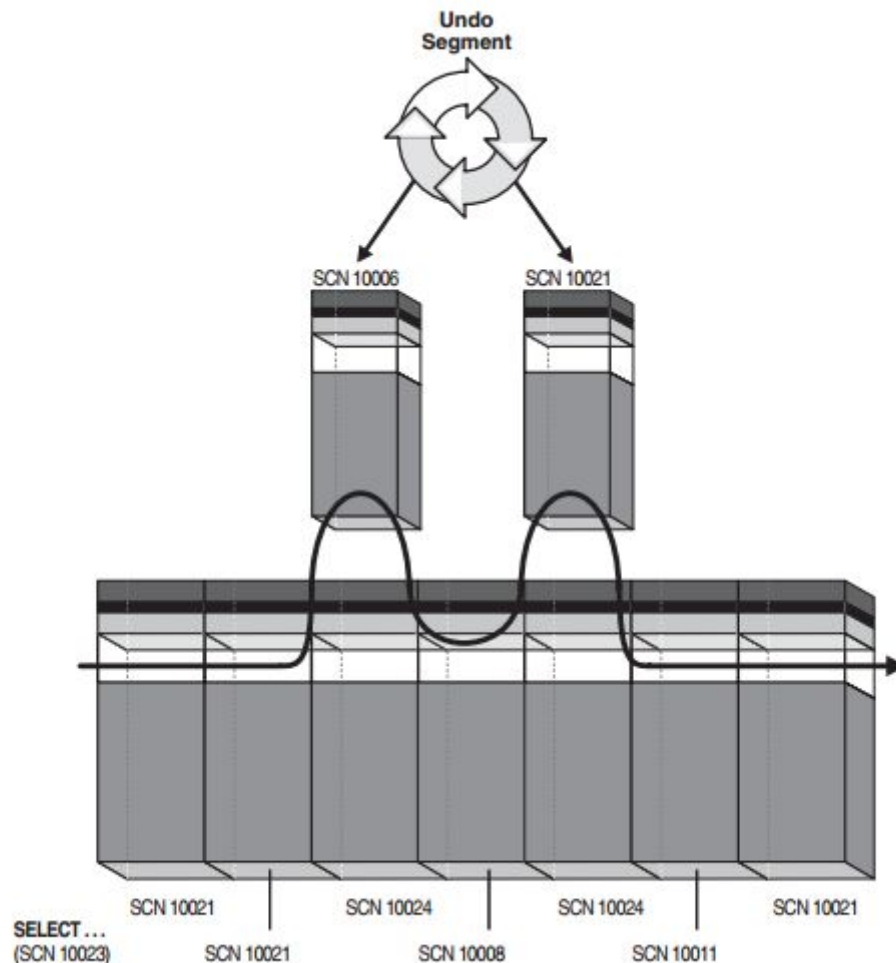


Imagen de ejemplo de consulta de un "segmento deshacer".

La base de datos usa un mecanismo llamado SCN para garantizar el orden de las transacciones. Cuando la sentencia SELECT entra en la fase de ejecución, dicha base de datos determina el SCN guardado justo en el momento en que la consulta fue ejecutada, en nuestro ejemplo, esta SC sería la 10023. La consulta solo podría ver los datos confirmados con respecto a SCN 10023.

En la imagen de ejemplo también podemos observar que los bloques con SC después del SCN 10023 indican datos modificados, que serían los SCN 10024. La sentencia SELECT requiere una versión del bloque que es consistente con los cambios confirmados.

2. NIVELES DE AISLAMIENTO DE TRANSACCIÓN DE ORACLE

En general las bases de datos de Oracle proporcionan los siguientes niveles de aislamiento:

- ☐ Nivel de aislamiento de lectura confirmada.
- ☐ Nivel de aislamiento serializable.
- ☐ Nivel de aislamiento de solo lectura.

2.1. NIVEL DE AISLAMIENTO DE LECTURA CONFIRMADA

Este es el nivel de aislamiento por defecto, en el cual toda consulta ejecutada por una transacción puede ver únicamente los datos confirmados antes de que la consulta hubiera empezado. Esto es apropiado para entornos donde en las bases de datos sea poco probable que las transacciones entren en conflicto.

2.1.1. CONFLICTOS DE ESCRITURA EN TRANSACCIONES DE LECTURA CONFIRMADA

Un conflicto de escritura ocurre cuando la transacción intenta cambiar una fila actualizada por una transacción concurrente no confirmada, también llamada transacción de bloqueo. La transacción de lectura confirmada espera a que la transacción de bloque termine y libere la fila bloqueada. Hay dos opciones posibles:

1. Si la transacción de bloqueo hace un “rolls back”, o reinicio, entonces la transacción que estaba en espera procede a cambiar la fila que previamente estaba bloqueada como si la otra transacción nunca hubiera existido.

2. Si la transacción de bloqueo se confirma y libera su estado bloqueado, entonces la transacción que estaba esperando procede con la actualización de la nueva fila cambiada.

2.2. NIVEL DE AISLAMIENTO SERIALIZABLE

En este nivel, una transacción solo ve los cambios confirmados en el momento que la transacción empezó y los cambios realizados por la propia transacción. Esto opera en un entorno que hace parecer como que ningún otro usuario estuviera modificando datos en la base de datos.

Se utiliza dicho nivel en los siguiente entornos:

- ☐ Con grandes bases de datos y transacciones cortas que actualicen solo unas pocas filas.
- ☐ Donde la probabilidad de que dos transacciones concurrentes modifiquen la misma fila es relativamente baja.
- ☐ Donde las transacciones relativamente largas de duración son principalmente sólo de lectura.

La base de datos Oracle permite a la transacción serializable modificar una fila únicamente si cambia la fila realizadas por otras transacciones fueron ya confirmadas. Se produciría un error cuando la transacción serializable intente actualizar o eliminar datos cambiados por una transacción diferente que se confirmó después de que la serializable hubiera empezado:

ORA-08177: Cannot serialize access for this transaction

Cuando se recibe el error ORA-08177, la aplicación puede hacer lo siguiente:

- ☐ Confirmar el trabajo ejecutado hasta ese punto
- ☐ Ejecutar declaraciones adicionales diferentes, por ejemplo después de hacer un “roll back” regresar a un “savepoint” o punto de guardado establecido antes de la transacción.
- ☐ Hacer “roll back” o reinicio a la transacción completa.

2.3. NIVEL DE AISLAMIENTO DE SOLO LECTURA

Este nivel es igual que el serializable pero sólo de lectura. Las transacciones no permiten que los datos se modifiquen en la transacción a menos que el usuario sea SYS. Por lo tanto, las transacciones de solo lectura no son susceptibles al error ORA-08177. Las transacciones de solo lectura son útiles para generar informes donde los contenidos deben ser consistentes respecto al momento en que comenzó la transacción.

3. DESCRIPCIÓN GENERAL DEL MECANISMO DE BLOQUEO EN ORACLE.

Un bloqueo es un mecanismo que previene las iteraciones destructivas, que son iteraciones que actualizan datos incorrectamente o alteran incorrectamente estructuras de datos, entre las transacciones que acceden a los datos compartidos.

La base de datos mantiene varios tipos de bloqueos, dependiendo de la operación que creó el bloqueo. En general, la base de datos utiliza dos tipos de bloqueos: bloqueos exclusivos y bloqueos compartidos. Solo se puede obtener un bloqueo exclusivo en un recurso como una fila o una tabla, pero muchos bloqueos compartidos se pueden obtener en un solo recurso.

Las siguientes reglas resumen el comportamiento de bloqueo de la base de datos Oracle para lecturas y escrituras:

- ☐ Una fila se bloquea solo cuando la modifica una escritura.
- ☐ Una escritura de una fila bloquea a una escritura concurrente de la misma fila.
- ☐ Una lectura nunca bloquea a una escritura.
- ☐ Una escritura nunca bloquea a una lectura.

3.1. MODOS DE BLOQUEOS

La base de datos de Oracle usa dos modos de bloqueo en una base de datos multiusuario:

- ❑ Modo de bloqueo exclusivo: Este modo previene los recursos asociados de ser compartidos. Una transacción obtiene un bloqueo exclusivamente cuando esta modifica cualquier dato. La primera transacción que bloquea un recurso exclusivamente es la única transacción que puede alterar el recurso hasta que el bloqueo es liberado.
- ❑ Modo de bloqueo compartido: Este modo permite que los recursos asociados puedan ser compartidos, dependiendo de la operación involucrada. Múltiples usuarios que estén leyendo un dato puede compartir ese dato, aguantando el bloqueo compartido para prevenir el acceso concurrente por un escritor que necesita un bloqueo exclusivo.

3.2. DURACIÓN DEL BLOQUEO

La base de datos de Oracle libera automáticamente un bloqueo cuando se produce algún evento, de modo que la transacción ya no requiere el recurso, por ejemplo cuando se realice un “commit” o un “roll back”, o también cuando una transacción después de un “savepoint” cuando se ha hecho un “roll back”. Sin embargo, sólo las transacciones que no esperan para los recursos previamente bloqueados pueden adquirir bloqueos en los recursos disponibles. Las transacciones en espera continúan esperando hasta después de que la transacción original se confirme o revierta por completo.

3.3. BLOQUEOS E INTERBLOQUEOS (DEADLOCKS)

Un interbloqueo o deadlock se da cuando dos o más usuarios esperan datos bloqueados entre ellos. Estos evitan que algunas transacciones continúen funcionando.

La base de datos Oracle detecta automáticamente los interbloqueos y los resuelve haciendo un “roll back” hasta una declaración involucrada en el interbloqueo, liberando un conjunto de filas en conflicto con bloqueos. Esta declaración pertenece a la transacción que detecta el interbloqueo.

3.4. BLOQUEOS DML


Un bloqueo DML o bloqueo de datos, garantiza la integridad de los datos a los que se acceden concurrentemente por múltiples usuarios. Por ejemplo este tipo de bloqueo impide que dos clientes compren la última copia de un libro disponible de una librería. Las declaraciones DML adquieren los siguientes tipos de bloqueos:


- ❑ Bloqueos de fila (TX).
- ❑ Bloqueos de tabla (TM).




3.4.1 BLOQUEOS DE FILA

Un bloqueo de fila o bloqueo TX, es un bloqueo en una simple fila de una tabla. Una transacción adquiere una fila bloqueada por cada fila modificada por diferentes sentencias como “INSERT, UPDATE, DELETE, MERGE O SELECT..FOR UPDATE”. Este bloqueo existe hasta que la transacción se confirme o haga un “roll back”.

Si la transacción obtiene un bloqueo para una fila, entonces también obtiene un bloqueo para la tabla que contiene dicha fila. Esto previene el conflicto en operaciones DDL que puedan sobrescribir datos cambiados en la transacción actual.

Table EMPLOYEES 

EMPLOYEE_ID	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	MANAGER_ID	DEPARTMENT_ID
 100	King	SKING	17-JUN-87	AD_PRES		90
101	Kochhar	NKOCHHAR	21-SEP-89	AD_VP	100	90
102	De Hann	LDEHANN	13-JAN-93	AD_VP	100	90
103	Hunold	AHUNOLD	03-JAN-90	IT_PROG	102	60

	Table lock acquired
	Exclusive row lock (TX) acquired
	Row being updated

La base de datos Oracle también tiene un almacén de bloqueos de filas y guarda en él la información del bloqueo en una bloque de datos que contiene dicho bloqueo de fila. La base de datos utiliza un mecanismo de cola para la obtención de los bloqueos de fila.

3.4.2 BLOQUEOS DE TABLAS

Un bloqueo de tablas o bloqueo TM, es obtenido por una transacción cuando la tabla es modificada por cualquier sentencia “INSERT, UPDATE, DELETE, MERGE, SELECT...FOR UPDATE or LOCK TABLE”. Este tipo de bloqueo se puede dar en los siguientes modos:

- ❑ **Fila compartida (RS):** Este bloqueo, también llamado bloqueo de tabla subcompartido (SS), indica que la transacción que tiene el bloqueo de la tabla tiene filas bloqueadas en la tabla y tiene la intención de actualizarlas. Un bloqueo de fila compartida es el modo menos restrictivo de bloqueo de tabla, que ofrece el mayor grado de concurrencia para una tabla.
- ❑ **Fila exclusiva de un bloqueo de tabla (SX):** Este bloqueo, también llamado bloqueo de tabla subexclusivo (SX), generalmente indica que la transacción que tiene el bloqueo ha actualizado las filas de la tabla o emitido SELECT ... FOR UPDATE. Un bloqueo SX permite que otras transacciones consulten, inserten, actualicen, eliminen o bloqueen filas al mismo tiempo en la misma tabla. Por lo tanto, los bloqueos SX permiten múltiples transacciones para obtener bloqueos SX y SS simultáneos para la misma tabla.

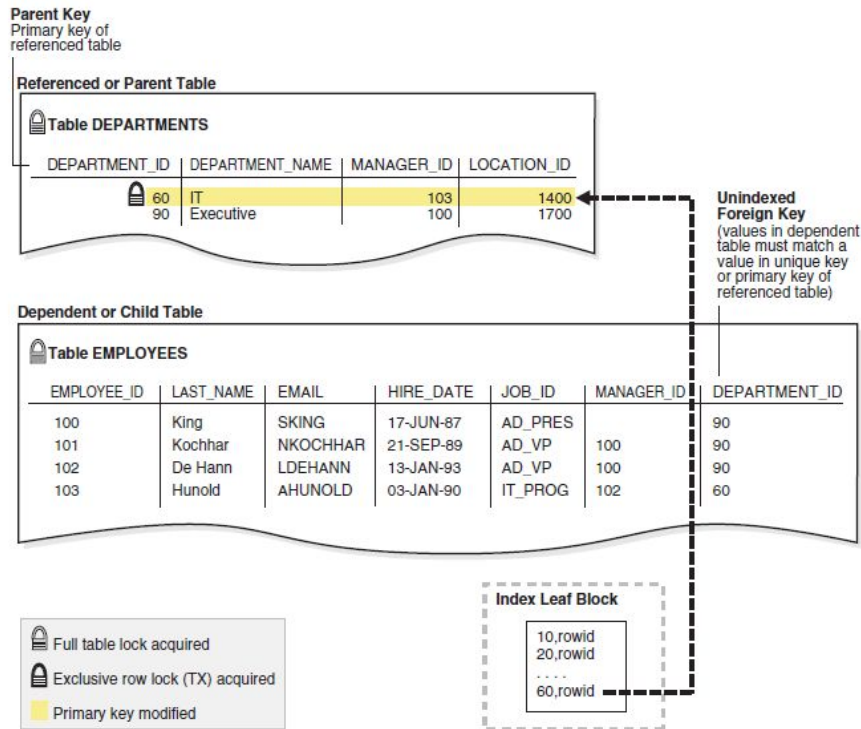
- ❑ **Bloqueo de tabla compartido (S):** Un bloqueo de tabla compartido obtenido por una transacción permite que otras transacciones consulten la tabla (sin usar SELECT ... PARA UPDATE), sin embargo, las actualizaciones solo se permiten si una sola transacción contiene el bloqueo de tabla compartido.
- ❑ **Bloqueo exclusivo de tabla en una fila compartida (SRX):** Este bloqueo, también llamado bloqueo de tabla subexclusivo compartido (SSX), es más restrictivo que un bloqueo de tabla compartido. Solo una transacción a la vez puede adquirir un bloqueo SSX en una tabla determinada. Un bloqueo de SSX obtenido por una transacción permite que otras transacciones consulten la tabla (excepto SELECT ... FOR UPDATE) pero no actualizar la tabla.
- ❑ **Bloqueo de tabla exclusivo (X):** Este bloqueo es el más restrictivo, lo que impide que otras transacciones ejecuten cualquier tipo de declaración de DML u obtengan cualquier tipo de bloqueo en la tabla.

3.4.3 BLOQUEOS Y CLAVES AJENAS (FOREIGN KEYS)

Las bases de datos de Oracle maximizan el control de concurrencia de las relaciones de claves primarias con las claves ajenas dependientes. El comportamiento del bloqueo depende de si las columnas de clave ajena están indexadas, es decir, tienen índices. Si no lo están, entonces la tabla secundaria probablemente será bloqueada con más frecuencia, se producirán bloqueos y se reducirá la concurrencia.

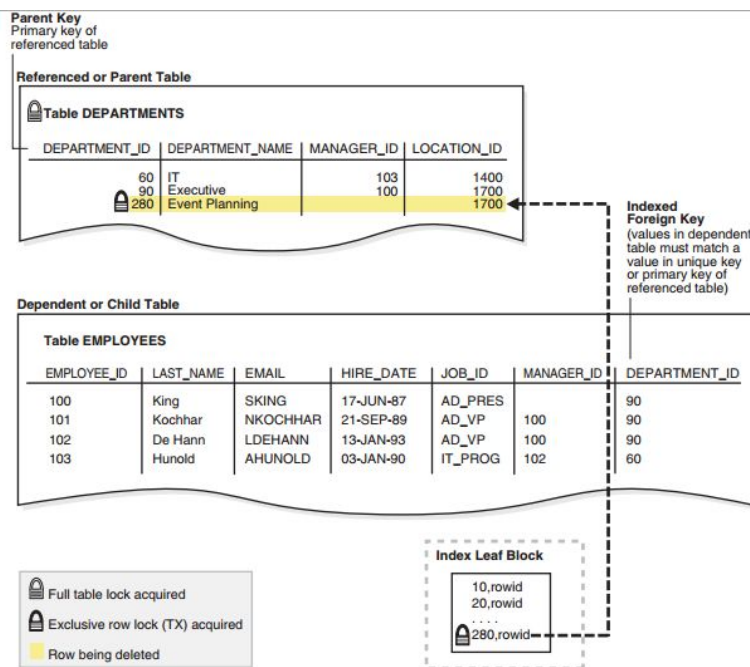
Para que se produzca un bloqueo de tabla completo en la tabla secundaria se deben producir dos condiciones:

1. No existe un índice en la columna de clave ajena de la tabla secundaria
2. Una sesión modifica una clave primaria o combina filas en la tabla principal.



La base de datos no requiere un bloqueo de tabla completo en la tabla secundaria cuando se cumplen estas dos condiciones:

1. Existe un índice en la columna de clave ajena en la tabla secundaria.
2. Una sesión modifica una clave primaria o combina filas en la tabla principal.



3.5. BLOQUEOS DDL

Un bloqueo de diccionario de datos (DDL) protege la definición de un objeto de esquema mientras la operación continúa de DDL actúa sobre el objeto o se refiere al objeto. Solo objetos de esquema individuales que están modificados o referenciados están bloqueados durante las operaciones DDL. La base de datos nunca bloquea todo el diccionario de datos.

Los usuarios no pueden solicitar bloqueos DDL explícitamente. Por ejemplo, si un usuario crea un procedimiento almacenado, la base de datos Oracle adquiere automáticamente los bloqueos DDL para todos los objetos de esquema a los que se hace referencia en la definición del procedimiento.

3.6. BLOQUEOS DE SISTEMA

La base de datos Oracle utiliza varios tipos de bloqueos de sistema para proteger la base de datos interna y la estructura de la memoria. Estos mecanismos son inaccesibles para los usuarios porque los estos no tienen control sobre su ocurrencia o duración.

4. BLOQUEOS DE DATOS MANUALES

La base de datos Oracle realiza el bloqueo automáticamente para garantizar la concurrencia de datos, integridad de los datos y la consistencia de lectura a nivel de declaración. Sin embargo, se pueden anular manualmente los mecanismos de bloqueo predeterminados. La modificación del bloqueo predeterminado es útil en situaciones como las siguientes:

- ❑ Las aplicaciones requieren consistencia de lectura a nivel de transacción o lecturas repetibles.
- ❑ Las aplicaciones requieren que una transacción tenga acceso exclusivo a un recurso para que la transacción no tenga que esperar a que se completen otras transacciones.

Se puede anular el bloqueo automático a nivel de sesión o a nivel de transacción. En el nivel de sesión, una sesión puede establecer el nivel de aislamiento de transacción requerido con la sentencia ALTER SESSION. En el nivel de transacción, las transacciones que incluyen las siguientes instrucciones SQL anulan el bloqueo predeterminado de la Base de Datos Oracle:

- ❑ La sentencia SET TRANSACTION ISOLATION LEVEL.
- ❑ La sentencia LOCK TABLE.
- ❑ La sentencia SELECT ... FOR UPDATE.

Los bloqueos adquiridos por los enunciados anteriores se liberan después de que finaliza la transacción o se realice un rollback. Si el bloqueo predeterminado se deshabilita en cualquier nivel, entonces el administrador de la base de datos o el desarrollador de la aplicación debe asegurarse de que los bloqueos deshabilitados funcionen correctamente.

5. BLOQUEOS DEFINIDOS POR EL USUARIO

Con los servicios de administración de bloqueos de Oracle Database, el usuario puede definir sus propios bloqueos para una aplicación específica, como por ejemplo serializar el acceso al registro de mensajes en los archivos del sistema.

Los bloqueos definidos por el usuario tienen todas las funcionalidades que tiene un bloqueo de Oracle Database, y nunca entran en conflicto pues los definidos por el usuario contienen el prefijo **UL**

Los servicios de administración de bloqueos de Oracle Database están disponibles a través de “*procedures*” en el paquete *DBMS_LOCK*. Pueden incluirse sentencias en bloques *PL_SQL* que:

- Soliciten un bloqueo de un tipo específico
- Den al bloqueo un nombre único reconocible en otro “*procedure*” de la misma u otra instancia
- Cambiar el tipo de bloqueo
- Liberar el bloqueo

6. CONCLUSIONES

La base de datos Oracle mantiene una concurrencia y consistencia de los datos haciendo uso de las transacciones serializables. Para ello usa diferentes niveles de aislamiento y modos de bloqueos que proporcionan cierta consistencia de los datos y estructuras de la base de datos.

Gracias a nuestro trabajo hemos podido entender y afianzar los conocimientos dados en la asignatura, más concretamente sobre la concurrencia en la base de datos Oracle, pudiendo ver con claridad como Oracle gestiona la concurrencia de las diferentes transacciones. También los distintos problemas que se pueden dar si no se controla y mantiene dicha concurrencia y como Oracle lo soluciona.

7. BIBLIOGRAFÍA

- Material ofrecido por el profesor en la moodle de la asignatura
- <https://docs.oracle.com/database/121/CNCPT/consist.htm#CNCPT221>
- <https://www.sqlshack.com/index-foreign-key-columns-sql-server/>
- https://www.ibm.com/support/knowledgecenter/es/SSLKT6_7.6.0/com.ibm.mt.doc/gp_intfrmwk/r_if_xml_obj_schema.html