**1. Defina que es un Bloqueo (Lock)**

La consistencia de transacciones se basa en que:

* + Una transacción solo puede leer y escribir un elemento si se solicitó un bloqueo y éste no se ha liberado.
  + Si una transacción bloquea un elemento, debe liberarlo posteriormente

**2. Defina que un Desbloqueo (UnLock)**

El comando **Unlock** quita un bloqueo establecido dentro del contexto de la transacción que está activa en ese momento. Todos los bloqueos se mantienen en el contexto de la transacción actual. Cuando la transacción actual se confirma o se revierte, se liberan automáticamente todos los bloqueos definidos dentro de la transacción.

**3. Para qué sirve la tabla de Bloqueo (Table Lock)**

Para garantizar que no haya problemas entre transacciones el Planificador de Transacciones del SGBD utiliza la tabla de **lock** para bloquear ciertas operaciones de manera que sea seguro ejecutar toda transacción

Un Bloqueo (lock) es una estructura que solo puede ser adquirida por una hebra de ejecución (thread) a la vez. Si dos ejecuciones tratan de obtener un lock para actualizar una tabla, la primera que trate de obtenerlo tendrá acceso exclusivo a la tabla, la segunda debe esperar a que la primera lo desbloquee (unlock) para obtener el acceso.

Los locks pueden tener distintas granularidades:

Base de datos, Tablas, filas, atributos

**4. Cuáles son los niveles de granularidad de los Lock**

Para garantizar que no haya problemas entre transacciones el Planificador de Transacciones del SGBD utiliza la tabla de **lock** para bloquear ciertas operaciones de manera que sea seguro ejecutar toda transacción.

Las transacciones solicitan operaciones de Lectura/Escritura

El Planificador Bloquea el dato X al que tiene acceso las operaciones lectura/escritura de las transacciones.

La operación COMMINT y ROLLBACK libera el dato bloqueado por la transacción.

Ejecuta las operaciones de Lectura/Escritura de manera segura

**5. Qué condiciones debe cumplir un Plan que usa Lock para ser considerado "legal"**

Debe ser serializable si es un plan NO SERIE en todo caso el plan debe ser SERIE

**6. Que característica debe cumplir un Bloque de dos Fases (Two-phase locking - 2PL)**

En toda transacción, todos los locks preceden a todos los unlocks. El problema de 2PL, caer en un abrazo mortal (deadlock).

T1 no puede ejecutar *Look*(B) porque T2 ya lo ejecuto, entonces T1 debe esperar hasta que T2 ejecute *Unlook*(B), …por lo tanto T1 debe continuar

T2 no puede ejecutar *Look*(A) porque T1 ya lo ejecuto, entonces T2 debe esperar hasta que T1 ejecute *Unlook*(A), …por lo tanto T2 debe continuar

**7. Qué problema puede darse al usar Bloque de dos Fases “(2PL)**

Se puede dar una inconsistencia de datos y puede faltar datos y causar falla y falta de dineros en las transacciones.

**8. Explique el concepto de Bloqueo Compartido (Shared Lock).**

Shared lock.- Cuando un transacción **Ti**bloquea en modo **Share Lock (*Ls*)** el dato **X**, permiten que otra transacción **Tj** pueda solamente leer el dato **X**, por lo tanto **Tj** debe esperar hasta que **Ti** lo libere (Unlock) para poder escribir X

**9. Explique el concepto de Bloqueo Exclusivo (Exclusive Lock).**

Exclusive lock.- Cuando un transacción **Ti**bloquea en modo **Exclusive Lock (*Lx*)** el dato **X**, no permiten que otra transacción **Tj** pueda leer/escribir el dato **X**, por lo tanto **Tj** debe esperar hasta que **Ti** lo libere (Unlock).

**10. Defina el concepto de Nivel de Aislamiento en las Transacciones**

En SQL Server las transacciones especifican un nivel de aislamiento que define el grado en que se debe aislar una transacción de las modificaciones de recursos o datos realizadas por otras transacciones.

Los niveles de aislamiento de transacciones controlan si una operación de lectura que hace referencia a filas modificadas por otra transacción:

* + Se bloquea hasta que se libera el bloqueo exclusivo de la fila.
  + Recupera la versión confirmada de la fila que existía en el momento en el que se inició la instrucción o la transacción.
  + Lee la modificación de los datos no confirmada.

**11. Cite los Nivel de Aislamiento en SQL Server**

SQL Server dispone de los siguientes niveles de aislamiento de transacciones

|  |  |  |
| --- | --- | --- |
| **NIVEL** | **DESCRIPCION** | **EFECTO** |
| READ UNCOMMITTED  (Lectura no confirmada) | Especifica que las instrucciones pueden leer filas que han sido modificadas por otras transacciones pero todavía no se han confirmado. | Lectura Sucia  Lectura no repetible  Datos Fantasma |
| READ COMMITTED  (Lectura confirmada) | Especifica que las instrucciones no pueden leer datos que hayan sido modificados, pero no confirmados, por otras transacciones | Lectura no repetible  Datos Fantasma |
| REPEATABLE READ  (Lectura repetible) | Especifica que las instrucciones no pueden leer datos que han sido modificados pero aún no confirmados por otras transacciones y que ninguna otra transacción puede modificar los datos leídos por la transacción actual hasta que ésta finalice | Datos Fantasma |
| SNAPSHOT  (instantánea) | Especifica que los datos leídos por cualquier instrucción de una transacción sean la versión coherente, desde el punto de vista transaccional, de los datos existentes al comienzo de la transacción |  |
| SERIALIZABLE  (Serializable) | Especifica lo siguiente:  Las instrucciones no pueden leer datos que hayan sido modificados, pero aún no confirmados, por otras transacciones.  Ninguna otra transacción puede modificar los datos leídos por la transacción actual hasta que la transacción actual finalice.  Otras transacciones no pueden insertar filas nuevas con valores de clave que pudieran estar incluidos en el intervalo de claves leído por las instrucciones de la transacción actual hasta que ésta finalice. |  |

**12. Explique Bloqueo Compartido por tabla en SQL Server**

WITH (TabLock) , con esta instrucción se obtienen bloqueos compartido (Shared Lock) para una tabla, es decir puedo bloquear todas las filas de una tabla, mientras mi transacción este activa, sin embargo, otras transacciones solamente pueden leer los datos de la tabla, podraan escribir cuando se se libere el datos (COMMIT/ROLLBACK)

**13. Explique Bloqueo Exclusivo por tabla en SQL Server**

Bloqueos exclusivos (o de escritura): Cuando una transacción mantiene un bloqueo de este tipo sobre un objeto, ninguna otra transacción puede acceder a él, ni adquirir ningún tipo de bloqueo sobre ese objeto, hasta que sea liberado por la transacción que lo había retenido. Este tipo de bloqueos se utiliza cuando una transacción quiere actualizar algún objeto.

**14. Explique Bloqueo Compartido por fila en SQL Server**

Bloqueos compartidos (o de lectura): Cuando una transacción tiene sobre un objeto un bloqueo de tipo compartido, permite que otras transacciones retengan también ese mismo objeto en bloqueos compartidos, pero no exclusivos. Este tipo de bloqueo se utiliza cuando las transacciones no necesitan actualizar datos, pero quieren impedir cualquier modificación de éstos mientras son consultados.

**15. Explique Bloqueo Exclusivo por fila en SQL Server**

WITH (RowLock, Xlock) , con esta instrucción se obtienen bloqueos exclusivos por fila, es decir puedo bloquear la fila de una tabla, mientras mi transacción este activa

RowLock  = Bloquea la fila indicada en el Select.   
 Xlock =  Hace que el bloque sea exclusivo