**Consultas**

* **CUENTA\_1 (S1)** = 101
* **CUENTA\_2 (S1)** = 104

CASO 1: Serializable S1 Y S2

CASO 2: Read Commited S1 Y S2

|  |  |  |
| --- | --- | --- |
| **Sesión 1 (S1)** | **T** | **Sesión 2 (S2)** |
| Defina nivel de Aislamiento para iniciar Bloque de Transacción T1 en S1  **SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;** | t1 |  |
| **S1T1 - Operación 1: modificar el saldo**   1. Seleccione una cuenta que llamemos CUENTA \_1 específica de su Base de datos sobre la cual quiere hacer esta consignación y elabore la sentencia SQL respectiva que modifica el saldo de está cuenta sumándole 1 millón de pesos.   UPDATE Cuentas  SET Saldo = Saldo + 1000000.00  WHERE IDCuenta = 101;    Se actualiza el saldo de IDCUENTA 101 a 1,005,000 solo en la sesión 1, dado que no aparece en otras sesiones el cambio, no se ha hecho commit. Nadie más puede modificar o leer este rango de índice hasta que S1 termine, se hace commit o rollback | t2 | No se refleja el cambio de la S1 en la S2 |
| **S1T1 - Operación 1: registro en el log**   1. Para la misma CUENTA \_1 específica de su Base de datos sobre la cual quiere hacer esta consignación de 1 millón de pesos, elabore la sentencia SQL para insertar en el log de operaciones sobre cuentas el registro de la consignación realizada.   INSERT INTO OperacionesCuentas (IDOperacionCu, Monto, Fecha, TipoOC, idCuenta, idPuntoAtencion) VALUES (operacionescuentas\_sequence.nextval,1000000, TO\_DATE('2024-04-28', 'YYYY-MM-DD'),'Consignar',101,4 )    Se inserta un nuevo log en OperacionesCuentas con el MONTO de 1,000,000 de tipo ‘Consignar’ en la cuenta 101. No se ve esto en otras sesiones, solo en esta sesión se ve el cambio, se completa sin interferencias ya que no hay transacciones activas en S2 que afecten CUENTA\_2. Similar a t2, esto bloquea el rango. | t3 | No se refleja el cambio de la S1 en la S2, no  se realiza el insert en la tabla de OperacionesCuentas de la S2 |
|  |  |  |
| **S1T1 – Operación 2: modificar el saldo**   1. Seleccione una cuenta que llamemos CUENTA\_2 *(diferente a la cuenta CUENTA\_*1) específica de su Base de datos y elabore la sentencia SQL respectiva que modifica el saldo de la cuenta restándole 50.000.   UPDATE Cuentas  SET Saldo = Saldo - 50000  WHERE IDCuenta = 104;    Se realiza la actualizacion del saldo de idcuenta 104 a 15000, dado que se le restan 50000. Bloquea la cuenta para S2 hasta el fin de la transacción | t4 | Defina nivel de Aislamiento para iniciar Bloque de Transacción T2 en la sesión S2  **SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;** |
| **S1T1 - Operación 2: registro en el log**   1. Para la misma CUENTA\_2 específica de su Base de datos sobre la cual quiere hacer este retiro, elabore la sentencia SQL para insertar en el log de operaciones sobre cuentas el registro del retiro de 50.000 pesos.   INSERT INTO OperacionesCuentas (IDOperacionCu, Monto, Fecha, TipoOC, idCuenta, idPuntoAtencion) VALUES (operacionescuentas\_sequence.nextval,50000, TO\_DATE('2024-04-28', 'YYYY-MM-DD'),'Retirar',104,5 )    Se realiza el insert en la tabla sin ningún problema, con el saldo de 50000, el idcuenta de 104. El id de la operación cuenta es 12. | t5 | Operaciones Cuentas permanece igual, es la misma tabla que en el t3 |
|  | t6 | **S2T2 - Operación 1: modificar el saldo**  Para la misma CUENTA\_1 (que escogió en la Sesión 1, tiempo t2), elabore la sentencia SQL respectiva que modifica el saldo de esta cuenta específica de su Base de datos, para descontar 30.000 pesos del saldo correspondientes a la cuota de manejo de la cuenta CUENTA\_1.    UPDATE Cuentas  SET Saldo = Saldo - 30000  WHERE IDCuenta = 101;    Puesto que hay un bloqueo sobre la fila de IDCUENTA 101 en cuentas por la modificación del t2 de la S1. Se queda cargando, esperando a que se desbloquee el candado para dicha fila para poder modificarla, espera a un commit o rollback que finalice la transacción de S1 |
| Cierre exitosamente este primer Bloque de Transacción S1T1(se confirma el registro de las operaciones)  COMMIT  Esto libera los bloqueos en CUENTA\_1 Y CUENTA\_2. | t7 | El error se produce dado que se trata de hacer el update de t6 pero puesto que en el nivel de aislamiento serializable, Oracle garantiza que una transacción vea los datos en un estado consistente, tal y como estaban al inicio de la transacción y que no vea cambios realizados por otras transacciones que se comprometieron después de que comenzó esta transacción. Este error se produjo porque la transacción (S2T2) intenta modificar datos que han sido cambiados por otra transacción (S1T1) que se comprometió después de que tu transacción comenzara. Oracle no puede mantener el aislamiento serializable en este caso porque los datos que S2T2 está intentando modificar ya no están en el estado que estaban cuando S2T2 comenzó.  CUENTAS: No Se modifica    OPERACIONESCUENTAS: No se modifica |
|  | t8 | **S2T2 - Operación 1: registro en el log**  Para la misma CUENTA\_1 específica de su Base de datos elabore la sentencia SQL para insertar en el log de operaciones sobre cuentas el registro del retiro de 30.000 pesos correspondientes a la cuota de manejo de la CUENTA\_1.  INSERT INTO OperacionesCuentas (IDOperacionCu, Monto, Fecha, TipoOC, idCuenta, idPuntoAtencion) VALUES (operacionescuentas\_sequence.nextval,  30000, TO\_DATE('2024-04-28', 'YYYY-MM-DD'),'Retirar',101,4 )    En el escenario donde S2T2 comenzó su operación antes del COMMIT de S1T1 en CUENTA\_1, S2T2 trabajaba con una "instantánea" de los datos antes de las modificaciones de S1T1. Al intentar S2T2 modificar el saldo y registrar cambios en el log de CUENTA\_1, se encontró con que S1T1 ya había confirmado cambios sobre la misma cuenta. El error ORA-08177 surge porque S2T2 intenta operar sobre una versión desactualizada de los datos, en conflicto con las garantías del aislamiento serializable que Oracle busca mantener. Esto ocurre al detectar que los datos objetivo han sido alterados post-inicio de S2T2, invalidando la operación según las restricciones de integridad transaccional serializable. |
|  | t9 | **S2T2 - Operación 1: modificar el saldo**  Para la misma CUENTA\_2 específica de su Base de datos, elabore la sentencia SQL respectiva que modifica el saldo de está cuenta sumándole 5.000 de intereses sobre la CUENTA\_2.  UPDATE Cuentas  SET Saldo = Saldo + 5000  WHERE IDCuenta = 104;    En el caso específico de t9 donde se intenta actualizar el saldo de IDCUENTA = 104 en la sesión S2, el error ORA-08177 surge porque esta transacción S2T2 comenzó antes de que la transacción S1T1 en la sesión S1 hiciera COMMIT en t7, que incluyó una modificación sobre la misma cuenta. Aunque S2T2 está operando bajo el nivel de aislamiento serializable, intenta modificar datos que ya fueron cambiados por otra transacción después de su inicio. Oracle, al mantener un nivel de aislamiento serializable, garantiza que las transacciones no vean cambios realizados por otras transacciones que se completan durante su ejecución. Al detectar que IDCUENTA = 104 fue modificada por S1T1 después del inicio de S2T2, Oracle lanza el error ORA-08177 para prevenir que S2T2 actúe sobre una base de datos que ya no refleja su estado inicial, manteniendo así la consistencia y aislamiento requeridos por el nivel serializable. |
|  | t10 | **S2T2 - Operación 1: registro en el log**  Para la misma CUENTA\_2 específica de su Base de datos, elabore la sentencia SQL para insertar en el log de operaciones sobre cuentas el registro de la consignación de 5.000 de intereses realizada.  INSERT INTO OperacionesCuentas (IDOperacionCu, Monto, Fecha, TipoOC, idCuenta, idPuntoAtencion) VALUES (operacionescuentas\_sequence.nextval,  5000, TO\_DATE('2024-04-28', 'YYYY-MM-DD'),’Consignar’,104,5 )    Mismo error descrito anteriormente |
| Consulte el saldo de la cuenta CUENTA\_1 y CUENTA\_2, para ello haga la consulta SQL respectiva  SELECT SALDO  FROM CUENTAS  WHERE IDCUENTA = 101    Se muestra el saldo de la IDCUENTA 101 perfectamente, no hay bloqueos de lectura sobre esta fila  SELECT SALDO  FROM CUENTAS  WHERE IDCUENTA = 104    Se muestra el saldo de la IDCUENTA 104 perfectamente, no hay bloqueos de lectura sobre esta fila | T11 |  |
|  | T12 | Cierre exitosamente este primer Bloque de Transacción S2T2 (se confirma el registro de la operaciones)  COMMIT  Se actualiza la tabla OperacionesCuentas, se ven las modificaciones de la S1 |
| Consulte el saldo de la cuenta CUENTA\_1 y CUENTA\_2, para ello haga la consulta SQL respectiva  SELECT SALDO  FROM CUENTAS  WHERE IDCUENTA = 101    Se muestra el saldo de la IDCUENTA 101 perfectamente, con sus modificaciones, no hay bloqueos de lectura sobre esta fila  SELECT SALDO  FROM CUENTAS  WHERE IDCUENTA = 104    Se muestra el saldo de la IDCUENTA 104 perfectamente con sus modificaciones, no hay bloqueos de lectura sobre esta fila | t13 | Consulte el saldo de la cuenta CUENTA\_1 y CUENTA\_2, para ello haga la consulta SQL respectiva  SELECT SALDO  FROM CUENTAS  WHERE IDCUENTA = 101  SELECT SALDO  FROM CUENTAS  WHERE IDCUENTA = 104    Se muestra el saldo de la IDCUENTA 101 perfectamente, con sus modificaciones de la S1 , no hay bloqueos de lectura sobre esta fila, se muestra lo mismo que en el select de la S1  SELECT SALDO  FROM CUENTAS  WHERE IDCUENTA = 104    Se muestra el saldo de la IDCUENTA 104 perfectamente con sus modificaciones de la S1, no hay bloqueos de lectura sobre esta fila, se muestra lo mismo que en el select de la S1 |

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

|  |  |  |
| --- | --- | --- |
| **Sesión 1 (S1)** | **T** | **Sesión 2 (S2)** |
| Defina nivel de Aislamiento para iniciar Bloque de Transacción T1 en S1  SET TRANSACTION ISOLATION LEVEL READ COMMITTED; | t1 |  |
| **S1T1 - Operación 1: modificar el saldo**  Seleccione una cuenta que llamemos CUENTA \_1 específica de su Base de datos sobre la cual quiere hacer esta consignación y elabore la sentencia SQL respectiva que modifica el saldo de está cuenta sumándole 1 millón de pesos.  UPDATE Cuentas  SET Saldo = Saldo + 1000000.00  WHERE IDCuenta = 101;    S1 modifica el saldo de CUENTA\_1, aumentándolo en 1,000,000 de pesos. Bajo Read Committed, esta modificación es visible para otras transacciones solo después del COMMIT. No hay bloqueos prolongados sobre los datos leídos por otras transacciones, solo se bloquean los datos escritos hasta que la transacción finalice. | t2 | La tabla Cuentas no presenta los cambios de S1 |
| **S1T1 - Operación 1: registro en el log**  Para la misma CUENTA \_1 específica de su Base de datos sobre la cual quiere hacer esta consignación de 1 millón de pesos, elabore la sentencia SQL para insertar en el log de operaciones sobre cuentas el registro de la consignación realizada.  INSERT INTO OperacionesCuentas (IDOperacionCu, Monto, Fecha, TipoOC, idCuenta, idPuntoAtencion) VALUES (operacionescuentas\_sequence.nextval,1000000, TO\_DATE('2024-04-28', 'YYYY-MM-DD'),'Consignar',101,4 )    S1 inserta un nuevo registro en la tabla OperacionesCuentas para documentar la consignación en CUENTA\_1. Al igual que la actualización del saldo, este registro no es visible para otras transacciones hasta el COMMIT de S1. | t3 | La tabla operaciones Cuentas no tiene los cambios de S1 |
|  |  |  |
| **S1T1 – Operación 2: modificar el saldo**   1. Seleccione una cuenta que llamemos CUENTA\_2 *(diferente a la cuenta CUENTA\_*1) específica de su Base de datos y elabore la sentencia SQL respectiva que modifica el saldo de la cuenta restándole 50.000.   UPDATE Cuentas  SET Saldo = Saldo - 50000  WHERE IDCuenta = 104;    S1 procede a actualizar CUENTA\_2 (IDCuenta = 104), reduciendo el saldo por 50,000 pesos. Esta modificación sigue el mismo principio de visibilidad que la primera operación. | t4 | Defina nivel de Aislamiento para iniciar Bloque de Transacción T2 en la sesión S2  SET TRANSACTION ISOLATION LEVEL READ COMMITTED; |
| **S1T1 - Operación 2: registro en el log**   1. Para la misma CUENTA\_2 específica de su Base de datos sobre la cual quiere hacer este retiro, elabore la sentencia SQL para insertar en el log de operaciones sobre cuentas el registro del retiro de 50.000 pesos.   INSERT INTO OperacionesCuentas (IDOperacionCu, Monto, Fecha, TipoOC, idCuenta, idPuntoAtencion) VALUES (operacionescuentas\_sequence.nextval,50000, TO\_DATE('2024-04-28', 'YYYY-MM-DD'),'Retirar',104,5 )    Se inserta un registro en la misma tabla para el retiro efectuado de CUENTA\_2. | t5 | OperacionesCuentas es igual que t3 S2 |
|  | t6 | **S2T2 - Operación 1: modificar el saldo**  Para la misma CUENTA\_1 (que escogió en la Sesión 1, tiempo t2), elabore la sentencia SQL respectiva que modifica el saldo de esta cuenta específica de su Base de datos, para descontar 30.000 pesos del saldo correspondientes a la cuota de manejo de la cuenta CUENTA\_1.    UPDATE Cuentas  SET Saldo = Saldo - 30000  WHERE IDCuenta = 101;    En t5, S2 parece quedar "cargando" al intentar modificar el saldo de CUENTA\_1. Esto puede ocurrir porque, aunque en Read Committed las transacciones solo ven cambios ya confirmados, un bloqueo sobre la fila específica puede persistir si S1 aún no ha liberado el bloqueo tras actualizar CUENTA\_1, y S2 debe esperar a que S1 complete su COMMIT en t7 para proceder. |
| Cierre exitosamente este primer Bloque de Transacción S1T1(se confirma el registro de las operaciones)  COMMIT  Cuentas:    Esta igual que antes del commit, igual que t4 S1, todavía no se ve el cambio de la S2 que acaba de hacer porque la S2 no ha hecho commit  OperacionesCuentas:    Esta igual que antes del commit, igual que t5 S1 | t7 | Se actualiza satisfactoriamente el update de t6 de la S2 que estaba esperando a que se liberara el candado  Cuentas:    Cuentas en IDCUENTA 101 le resta 30000 a el saldo que tenia IDCUENTA 101 en la s1, se actualiza a los datos nuevos con el commit antes de restar el valor. Aparece el nuevo valor de IDCUENTA 104 con su modificación de la S1  OperacionesCuentas:    Se actualiza con los cambios de la S1.  Aquí se evidencia que es read commited, pues cuando hay un commit en alguna transacción directamente se actualiza todo en las otras sesiones read commited. |
| OperacionesCuentas no se actualiza con la modificación de S2, queda igual que t7 | t8 | **S2T2 - Operación 1: registro en el log**  Para la misma CUENTA\_1 específica de su Base de datos elabore la sentencia SQL para insertar en el log de operaciones sobre cuentas el registro del retiro de 30.000 pesos correspondientes a la cuota de manejo de la CUENTA\_1.  INSERT INTO OperacionesCuentas (IDOperacionCu, Monto, Fecha, TipoOC, idCuenta, idPuntoAtencion) VALUES (operacionescuentas\_sequence.nextval,  30000, TO\_DATE('2024-04-28', 'YYYY-MM-DD'),'Retirar',101,4 )    Se realiza el insert sobre operacionesCuentas, Ahora, los registros insertados por S1 son visibles en esta tabla, pero como cada inserción es independiente, no hay conflicto directo. |
| Cuentas no se modifica con la modificación de S2, queda igual que en t7 | t9 | **S2T2 - Operación 1: modificar el saldo**  Para la misma CUENTA\_2 específica de su Base de datos, elabore la sentencia SQL respectiva que modifica el saldo de está cuenta sumándole 5.000 de intereses sobre la CUENTA\_2.  UPDATE Cuentas  SET Saldo = Saldo + 5000  WHERE IDCuenta = 104;    S2 ahora modifica el saldo de CUENTA\_2 sumando 5,000 de intereses. Esta operación verá el saldo actualizado de CUENTA\_2 como fue modificado y confirmado en t7 por S1 |
| OperacionesCuentas no se actualiza con la modificación de S2, queda igual que t7 | t10 | **S2T2 - Operación 1: registro en el log**  Para la misma CUENTA\_2 específica de su Base de datos, elabore la sentencia SQL para insertar en el log de operaciones sobre cuentas el registro de la consignación de 5.000 de intereses realizada.  INSERT INTO OperacionesCuentas (IDOperacionCu, Monto, Fecha, TipoOC, idCuenta, idPuntoAtencion) VALUES (operacionescuentas\_sequence.nextval,  5000, TO\_DATE('2024-04-28', 'YYYY-MM-DD'),’Consignar’,104,5 )    OperacionesCuentas se actualiza con el insert, este se hace sin ningún problema. Se inserta un registro de la consignación de intereses para CUENTA\_2. |
| Consulte el saldo de la cuenta CUENTA\_1 y CUENTA\_2, para ello haga la consulta SQL respectiva  SELECT SALDO  FROM CUENTAS  WHERE IDCUENTA = 101    Dado que no se ha hecho commit en la S2, no se ve el saldo actualizado con las modificaciones de S2 sino el saldo que hay desde t7. Se reflejan las propiedades de read committed  SELECT SALDO  FROM CUENTAS  WHERE IDCUENTA = 104    Dado que no se ha hecho commit en la S2, no se ve el saldo actualizado con las modificaciones de S2 sino el saldo que hay desde t7. Se reflejan las propiedades de read committed | T11 |  |
| Ddao que se hace commit en la S2, las tablas se actualizan a los cambios que se hicieron en S2  Cuentas:    OperacionesCuentas: | T12 | Cierre exitosamente este primer Bloque de Transacción S2T2 (se confirma el registro de la operaciones)  COMMIT |
| Consulte el saldo de la cuenta CUENTA\_1 y CUENTA\_2, para ello haga la consulta SQL respectiva  SELECT SALDO  FROM CUENTAS  WHERE IDCUENTA = 101    Dado que ya se hizo la actualización de las modificaciones de S2 con el commit de S2, entonces el saldo de IDCUENTA 101 se actualiza  SELECT SALDO  FROM CUENTAS  WHERE IDCUENTA = 104    Dado que ya se hizo la actualización de las modificaciones de S2 con el commit de S2, entonces el saldo de IDCUENTA 104 se actualiza | t13 | Consulte el saldo de la cuenta CUENTA\_1 y CUENTA\_2, para ello haga la consulta SQL respectiva  SELECT SALDO  FROM CUENTAS  WHERE IDCUENTA = 101    El saldo es el mismo que estaba en la tabla cuentas  SELECT SALDO  FROM CUENTAS  WHERE IDCUENTA = 104    El saldo es el mismo que estaba en la tabla cuentas |