

TEMA 6
TRANSACCIONES Y CONTROL DE CONCURRENCIA DISTRIBUIDO.

Nota: Los ejercicios se entregan de forma individual en cualquier formato.

1.1. PREGUNTAS.

1. Defina el concepto de estado de consistencia de una BD.
2. Describa brevemente las operaciones del SQL que se emplean para realizar el control transaccional de una BD.
3. Describir brevemente cada una de las propiedades ACID de las transacciones.
4. Defina brevemente cada uno de los siguientes problemas que pueden ocurrir cuando 2 transacciones concurrentes interactúan con un mismo dato.
 - a. Lost Update (Actualización perdida o sobrescrita).
 - b. Dirty Read (Lectura sucia o no confirmada.)
 - c. Non repeatable read (lecturas no repetibles)
5. Mencione los diferentes criterios que se emplean para clasificar a las transacciones y mencione cada clasificación.
6. ¿Cuál es la finalidad del uso de una transacción autónoma en Oracle?
7. Describir brevemente en los 2 mecanismos de control de concurrencia empleados tanto en BD centralizadas como en BDD
8. Mencione las principales características de los 2 puntos de vista empleados para implementar los mecanismos mencionados en la pregunta anterior.
9. Mencione las principales características de los siguientes componentes fundamentales para un correcto control de concurrencia en una BDD:
 - a. Transaction Manager
 - b. Scheduler
 - c. Procesador de datos.
10. ¿Por qué razón al Scheduler se le conoce como Lock Manager?
11. ¿Cuál es la principal diferencia ente el protocolo 2PL y el protocolo 2PL estricto?
12. Describa las diferencias del protocolo 2PL que se emplea en BD centralizadas Vs el protocolo 2PL empleado en BDD llamado Protocolo de 2 fases de sitio primario.
13. Mencione las diferencias entre el Protocolo de 2 fases de sitio primario y el protocolo de 2 fases distribuido.
14. Describa brevemente el concepto de timestamp de una transacción.
15. Describa la fórmula que define a la regla de ordenamiento por Timestamp: **Regla TO**.
16. Defina los conceptos $rts(x)$ y $wts(x)$
17. Par una operación de escritura que pertenece a una transacción T_i , ¿Qué acciones tomará el Scheduler si se cumple la condición $ts(T_i) < wts(x)$?
18. Describa brevemente los 3 algoritmos empleados para implementar el control de concurrencia distribuido empleando ordenamiento de Timestamps.
19. Describa las 3 técnicas empleadas para manejar Deadlocks. Indique cuál de las 3 es la más empleada.
20. Describa brevemente los 3 algoritmos que se emplean en una BDD para detectar y resolver la ocurrencia de un deadlock

1.2. NIVELES DE AISLAMIENTO

1.2.1. Ejercicio 1

Suponer que 2 transacciones realizan las siguientes actividades en una BD que no tiene habilitado **ningún** nivel de aislamiento. Llenar las columnas de la siguiente tabla:

Tiempo	Txn1	Valor de X	Txn2	Valor de X	Nombre del problema que se genera	Nivel de aislamiento que resuelve el problema	¿Cuál debería haber sido el valor correcto de X?
T0	x=100				Dirty Read	Read Committed	120
T1	Start	NA		NA			
T2	r(x)	100	start	NA			
T3	x<-x+1	101		NA			
T4	w(x)	101		NA			
T5		NA	r(x)	101			
T6		NA	x<-x+20	121			
T7		NA		NA			
T8	Rollback	100		NA			
T9			w(x)	121			
T10			commit	121			

1.2.2. Ejercicio 2.

Reescribir la tabla anterior aplicando el nivel de aislamiento indicado en la última columna, verificar que efectivamente el valor de X es el correcto.

Tiempo	Txn1	Valor de X	Txn2	Valor de X	Nombre del problema que se genera	Nivel de aislamiento que resuelve el problema	¿Cuál debería haber sido el valor correcto de X?
T0	x=100				Dirty Read	Read Committed	120
T1	Start	NA		NA			
T2	r(x)	100	start	NA			
T3	x<-x+1	101		NA			
T4	w(x)	101		NA			
T5		NA	r(x)	100			
T6		NA	x<-x+20	120			
T7		NA		NA			
T8	Rollback	100		NA			
T9			w(x)	120			
T10			commit	120			

1.2.3. Ejercicio 3

Suponer que 2 transacciones realizan las siguientes actividades en una BD con nivel de aislamiento READ COMMITTED. En Oracle es el nivel por default. Explícitamente es equivalente a ejecutar:

```
SQL> set transaction isolation level read committed;
```

Completar la siguiente tabla.

Tiempo	Txn1	Valor de X	Txn2	Valor de X	Nombre del problema que se genera	Estrategia que soluciona el problema	¿Cuál debería haber sido el valor correcto de X?
T0	x=500				Lost Update Txn1 está sobrescribiendo el valor de X en T8 que fue actualizado en T3 por Txn2	Emplear Control de concurrencia optimista. Update condicionado: Update X = 505 Where X = 500 500 es el valor inicial de X. Si X cambia, la sentencia UPDATE no afecta registros. Las aplicaciones deben detectar esta condición y reintentar la transacción completa. Al reintentar, Txn1 leerá 510 en lugar de 500, Por	515
T1		NA	r(x)	500			
T2		NA	x<-x+10	510			
T3		NA	w(x)	510			
T4	r(x)	500		NA			
T5		NA	r(x)	510			
T6	x<-x+5	505	commit	510			
T7	w(x)	505		NA			
T8	commit	505		NA			

						lo tanto cuando termine Txn1 X valdrá 510+5 = 515 que es el valor esperado y correcto.	
--	--	--	--	--	--	--	--

En Oracle existe la cláusula SELECT FOR UPDATE:

```
Select * from my_table for update of my_column;
```

Esta instrucción bloqueará cualquier otra transacción que intente modificar el contenido de la columna `my_column`. El bloqueo puede ser a nivel de registro o tabla. Si esta instrucción se aplica a Txn1, Txn2 no podrá leer ni escribir el valor de X hasta que Txn1 termine. Ojo: Esto no resuelve el problema de LOST UPDATE ya que posterior a que Txn1 termine, Txn2 se reanuda y actualizaría a 510, lo que provoca que se pierda el valor actualizado por Txn1.

1.2.4. Ejercicio 4

Suponer que 2 transacciones realizan las siguientes actividades en una BD con nivel de aislamiento READ COMMITTED. Completar la siguiente tabla. Considerar que Txn2 hace varias operaciones y lee 2 veces el valor de X de la BD. Se espera que el valor de X no debe cambiar para que los cálculos sean correctos.

Tiempo	Txn1	Valor de X	Txn2	Valor de X	Nombre del problema que se genera	Nivel de aislamiento que resuelve el problema	¿Cuál debería haber sido el valor correcto de X al concluir TXn2?
T0	x=100				Lecturas no repetibles	Repeatable Read	206 (Ver el siguiente ejercicio para comprobar este resultado).
T1	r(x)	100		NA			
T2	x<-x+4	104		NA			
T3	w(x)	104		NA			
T4		NA	y=r(x)	x=100,y=100			
T5	commit	104					
T6		NA	y<-x+3	x=100,y=103			
T7		NA	z=r(x)	x=104,z=104			
T8		NA	z<-x+3	x=104,z=107			
T9		NA	x=y+z	x=103+107			
T10		NA	w(x)	210			
T11		NA	commit	210			

1.2.5. Ejercicio 5

Reescribir la tabla anterior aplicando el nivel de aislamiento repeatable read, verificar que efectivamente el valor de X es el correcto.

Tiempo	Txn1	Valor de X	Txn2	Valor de X	Nombre del problema que se genera	Nivel de aislamiento que resuelve el problema	¿Cuál debería haber sido el valor correcto de X al concluir TXn2?
T0	x=100				Lecturas no repetibles	Repeatable Read	206
T1	r(x)	100		NA			
T2	x<-x+4	104		NA			
T3	w(x)	104		NA			
T4		NA	y=r(x)	x=100,y=100			
T5	commit	104					
T6		NA	y<-x+3	x=100,y=103			
T7		NA	z=r(x)	x=100,z=100			
T8		NA	z<-x+3	x=100,z=103			
T9		NA	x=y+z	x=103+103			
T10		NA	w(x)	206			
T11		NA	commit	206			

En Oracle se puede especificar la siguiente instrucción para aumentar el nivel de aislamiento al iniciar Txn2:

```
SQL> set transaction isolation level serializable;
```

1.2.6. Ejercicio 6

Suponer que 2 transacciones realizan las siguientes actividades en una BD con nivel de aislamiento READ COMMITTED. Completar la siguiente tabla. Considerar que Txn2 está realizando el cálculo de un reporte por lo cual el número de registros no debe modificarse.

Tiempo	Txn1	Valor de X	Txn2	Valor de X	Nombre del problema que se genera	Nivel de aislamiento que resuelve el problema	¿Cuál debería haber sido el valor correcto de 'r' al concluir TXn2?
T0	Considerar los siguientes datos iniciales:				Lecturas fantasma	Serializabe	10
T1	x = select count(*) from temp where c =100	5		NA			
T2		NA	insert into temp values (6,100)	NA			
T3	z = x	z=5 x=5		NA			
T4			commit;	NA			
T5	x = select count(*) from temp where c =100	6		NA			
T6	y = x	x=6 y=6		NA			
T7	r=y+z	r=11		NA			

De forma similar al ejercicio anterior, en Oracle se puede especificar la siguiente instrucción para aumentar el nivel de aislamiento al iniciar Txn2:

```
SQL> SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
```

Al aplicar esta instrucción, en T5 se obtendría x = 5, y por lo tanto r = 10.

1.2.7. Ejercicio 7

Suponer que se tiene la siguiente información almacenada en una BD.

PRODUCTO

Producto_id	Cantidad
1001	10
1002	20
1003	15
1004	5

Considerar la siguiente secuencia de acciones realizadas por 2 transacciones concurrentes en una BD donde se ha configurado un nivel de aislamiento "Read Committed" (lecturas confirmadas). Las 2 transacciones están actualizando las existencias de 4 productos que se venden en una tienda de auto servicio.

Tiempo	Txn1	Valor de la variable Cn para Txn1	Txn2	Valor de la variable Cn para Txn1
T0	start declare c1 number ,c2 number, c3 number, c4 number, c5 number,c6 number;	NA	start declare c1 number ,c2 number, c3 number, c4 number, c5 number, c6 number;	NA
		NA	update producto set cantidad = 5 where producto_id =1001	NA
T1	select cantidad into c1 from producto where producto_id=1001;	C1 = 10	select cantidad into c1 from producto where producto_id=1001;	C1 = 5
T2	select cantidad into c2 from producto where producto_id=1002;	C2 = 20	select cantidad into c2 from producto where producto_id=1002;	C2 = 20
T3	select cantidad into c3 from producto where producto_id=1003;	C3 = 15	select cantidad into c3 from producto where producto_id=1003;	C3 = 15
T4		NA	update producto set cantidad = 10 where producto_id =1004	NA
T5		NA	select cantidad into c5 from producto where producto_id=1004	C5 = 10
T6		NA	select sum(cantidad) into c6 from producto;	50
T7		NA	commit;	NA
T8	select cantidad into c4 from producto where producto_id=1004	10		
T9	select sum(cantidad) into c7 from producto;	55		NA
T10	commit;	NA		NA

- A. Llenar los espacios vacíos de la tercera y quinta columna para conocer los valores de C1,...,C5.
- B. ¿Cuál de los 2 valores obtenidos en C6 y C7 debería ser el correcto para asegurar que el total de productos obtenidos es el correcto al terminar ambas transacciones?
 C6 es el correcto, es decir al final existen 50 productos ya que solo Txn2 hizo cambios a la BD.
 - Txn1 hizo una lectura en T1 antes que T2 terminara, por lo que se leyó el valor 10.
 - Txn1 hizo una lectura en T8 después que T2 terminara, por lo que se leyó el valor que actualizó T2, el valor 10.
 Lo anterior provoca que Txn haga un mal calculo ya que un dato no lo vio y el otro si.
- C. ¿Cuál de los 3 problemas que define el estándar SQL en cuanto a concurrencia de transacciones es el que se está presentando? Y en ¿qué tiempo se está provocando el problema.
 Lectura no repetible, ocurre en T8 al leer un valor 10 para el producto 1004 el cual fue modificado por Txn2 mientras Txn1 estaba en ejecución.
- D. ¿Cuál de los 4 niveles de aislamiento que define el estándar SQL podría solucionar el siguiente problema? ¿Y a qué transacción se le debe aplicar?
 Repeatable Reads, se debe aplicar a Txn1

1.2.8. Ejercicio 8 (no resuelto).

La siguiente tabla de datos muestra un listado de los productos que ofrece una tienda y una bandera que indica si el producto es defectuoso o no. La base de datos opera de 9 am a 6 pm en donde un cliente puede reportar si el producto esta defectuoso o si funciona adecuadamente. Suponer que un “becario” (recién graduado) decide ejecutar una serie de consultas a la base de datos dentro de este horario para generar un reporte que contiene lo siguiente:

- En un archivo txt debe almacenar por fecha el número de productos defectuosos que existen en la tabla producto. Se le solicita generar el reporte para los 2 días mostrados a continuación. Para la tabla de datos, el archivo se debe visualizar de la siguiente manera:

Fecha	#defectuosos
31/12/2009	2
01/01/2010	1

- Este archivo será enviado al proveedor de los productos para notificarle la cantidad de productos que se le enviarán para su reemplazo.
- Posteriormente, el becario deberá generar otro archivo txt con la lista de todos los identificadores de los productos que están defectuosos considerando las mismas fechas que el reporte anterior. Dicho archivo será enviado al almacén para que los productos defectuosos sean extraídos y llevados al proveedor para su reemplazo. Para la tabla de datos el archivo se deberá visualizar de la siguiente manera.

```

Producto_id
1001
1002
1003

```

Tabla de datos:

PRODUCTO

Producto_id	Defectuoso	Fecha de reporte
1001	1	31/12/2009
1002	1	01/01/2010
1003	1	01/01/2010
1004	0	01/01/2010
1005	0	01/01/2010

- Suponer que el becario representa a la transacción **Txn1** de la siguiente tabla, y al mismo tiempo un cliente representado por la transacción **Txn2** realiza otras operaciones:

Tiempo	Txn1 (becario)	Txn2 (cliente)
	start	start
T1	El becario ejecuta esta sentencia para generar el primer txt: select fecha_reporte,count(*) from producto where defectuoso = 1 and fecha_reporte in('31/12/2009','01/01/2010') group by fecha_reporte order by 1;	
T2		El cliente reporta un producto como defectuoso: insert into producto values(1006,1,'01/01/2010');
T3		commit;
T4	El becario ejecuta esta sentencia para generar el segundo txt: select producto_id from producto where defectuoso =1 and fecha_reporte in('31/12/2009','01/01/2010') order by fecha_reporte;	
T5	commit;	

- Ambos archivos son enviados a sus destinos.
- Al día siguiente el gerente recibe una llamada en la que se le notifica que los valores que se enviaron en los archivos son incorrectos y no corresponden con los productos que le fueron enviados por parte del almacén.
- Le solicitan al becario revisar el problema lo más pronto posible.

- Con base al escenario, escribir el contenido de los archivos que se le enviaron al proveedor y al almacén.
- ¿Qué problema fue el que generó esta ambigüedad?
- ¿Qué debería realizar el becario para solucionar el problema y poder conservar su trabajo?

1.3. CONTROL DE CONCURRENCIA.**1.3.1. Ejercicio 1**

Suponga que las siguientes transacciones se ejecutan en una BDD configurada con un control de concurrencia basada en bloqueos con nivel de aislamiento "Read Committed".

PRODUCTO

Producto_id	Cantidad
1001	10
1002	20
1003	15
1004	5

Tiempo	Txn1	Txn2
T1	start	start
T2		select cantidad from producto where producto_id=1001;
T3		update producto set cantidad = 12 where producto_id=1001;
T4	update producto set cantidad =(select cantidad from producto where producto_id = 1001)+5 where producto_id=1001;	
T5		commit;
T6	commit;	

- A. ¿Existirá Dead Lock?
No
- B. ¿Existirá estado inconsistente en la BD al terminar esta sentencia de operaciones?
Lost Update. El valor actualizado por Txn2 es sobrescrito por Txn1
- C. ¿Qué valor contendrá el campo CANTIDAD después de ejecutar estas operaciones?
15
- D. ¿Qué valor contendrá el campo CANTIDAD si ambas transacciones se ejecutaran de forma secuencial T2, y después T1?
17
- E. ¿Existirá algún bloqueo?
SI
- F. En caso de generarse un bloqueo, escriba la sentencia SQL que tendrá que esperar a que el Lock del dato al que desea acceder sea liberado.
**update producto set cantidad =(
select cantidad from producto where producto_id = 1001
) +5 where producto_id=1001;**
- G. ¿Cuál de las 2 sentencias commit se va a ejecutar primero?
La sentencia commit de Txn2

1.3.2. Ejercicio 2

Suponga que las siguientes transacciones se ejecutan en una BDD configurada con un control de concurrencia basada en bloqueos con nivel de aislamiento "Read Committed".

PRODUCTO

Producto_id	Cantidad
1001	10
1002	20
1003	15
1004	5

Tiempo	Txn1	Txn2
T1	start	start
T2	select cantidad where producto_id=1004;	
T3		select cantidad where producto_id=1004;
T4	Update producto set cantidad = 20 where producto_id =1004	
T5	commit;	

- A. ¿Existirá Dead Lock?
No
- B. ¿Existirá algún bloqueo?
No
- C. En caso de generarse un bloqueo, escriba la sentencia SQL que tendrá que esperar a que el Lock del dato al que desea acceder sea liberado.
NA
- D. Observar que la transacción Txn2 nunca realizó commit. ¿Qué problema existirá con la sentencia de Txn1 en T4?
Ningún problema.
- E. Para poder ejecutar la sentencia en T2 para Txn1, ¿se requiere adquirir algún Lock?, en caso afirmativo, indicar el tipo de lock que se debe adquirir para poder acceder al dato.
Si, Read Lock.
- F. Para poder ejecutar la sentencia en T4 para Txn1, ¿se requiere adquirir algún Lock?, en caso afirmativo, indicar el tipo de lock que se deberá adquirir para poder ejecutar la sentencia.
Si, Write Lock

1.3.3. Ejercicio 3

Suponga que las siguientes transacciones se ejecutan en una BDD configurada con un control de concurrencia basada en bloqueos con nivel de aislamiento "Read Committed".

PRODUCTO

Producto_id	Cantidad
1001	10
1002	20
1003	15
1004	5

Tiempo	Txn1	Txn2	Txn3
T1	start	start	start
T2		Update producto set cantidad = 100 where producto_id =1003;	
T3	Update producto set cantidad = 2000 where producto_id =1003;		
T4	Commit;		
T5	start		
T6	Select cantidad From producto Where producto_id =1003;		
T7	Commit;		
T8			Select cantidad From producto Where producto_id =1003;
T9			Commit;

Observar que Txn2 nunca hace commit.

- A. ¿Existirá Dead Lock?
No
- B. ¿Cuántos bloqueos existirán en total?

1 en T3 para Txn1

- C. ¿Qué valor para el campo cantidad se obtendrá en T5 para Txn3?

15, Observar que Txn2 nunca termina, por lo que el bloqueo ocurrido en T3 para Txn1 provoca que Txn1 no avance y se quede bloqueada en el tiempo T3. La única transacción que termina es Txn3 Por lo tanto el se obtiene el valor original para producto_id =1003 que es 15.

- D. ¿Qué valor se obtendrá en T6 para Txn1?

Txn1 se queda bloqueada al no terminar Txn2

1.3.4. Ejercicio 4 (no resuelto)

Empleando los datos de la tabla PRODUCTO del ejercicio anterior, llenar la siguiente tabla con los pasos necesarios para provocar un Dead Lock.

Tiempo	Txn1	Txn2
T1		
Etc..		

1.3.5. Ejercicio 5

Considere la siguiente secuencia de operaciones que realiza un sistema de reservaciones de venta de boletos de avión para 2 clientes que acceden al sitio web en el siguiente orden. La BDD está configurada con un nivel de aislamiento "Read committed", y se desea que la aplicación permita el mayor nivel de concurrencia posible. El asiento seleccionado por ambos clientes está disponible (ocupado =0) en el tiempo T1.

Tiempo	Txn1 (cliente 1)	Txn2 (cliente 2=)
T1	start	Start
T2	select ocupado from vuelo where asiento ='12A'	select ocupado from vuelo where asiento ='12A'
T3	El cliente confirma	
T4	update vuelo set ocupado = 1 where asiento ='12A';	El cliente confirma
T6		update vuelo set ocupado = 1 where asiento ='12A';
T7	commit;	
T8		commit;

- A. ¿Existirá Dead Lock?

No

- B. ¿Existirá algún bloqueo?, ¿en qué tiempo?

Si, en T6 para Txn2, Txn2 se espera por un momento hasta que Txn1 termine.

- C. Observar que el sistema esta asignando el mismo asiento a ambos clientes lo cual representaría una inconsistencia en la BD.

- ¿Qué cambios en las sentencias anteriores o configuraciones se deberán realizar para evitar este problema?, recordando, se debe conservar el mayor nivel de concurrencia posible.

Agregar una condición al UPDATE (update condicionado)

update vuelo set ocupado = 1 where asiento ='12A' and ocupado =0;

- ¿Qué tipo de control de concurrencia será la mejor opción para resolver el error?

Control de concurrencia optimista.

- Suponiendo que el error ha sido corregido, ¿Qué cliente obtendrá el asiento deseado y qué cliente recibirá un error indicando que alguien más acaba de "ganar" el asiento?

El cliente 2 recibirá un error debido a que la sentencia update condicionada no actualizaría registro alguno. Cuando Txn1 hace commit, se libera el Lock y Txn2 reanuda su ejecución lanzando el Update condicionado. En ese punto, el asiento ya fue ocupado por Txn, por lo que la condición ocupado =0 ya no se cumple y no se haría el cambio (0 registros modificados). Esta condición la detectaría el sistema y generará un error para mostrarle al cliente indicando que le han ganado el asiento.