Laboratorio: Gestión de Transacciones

diciembre 2011

Objetivo de la práctica

Asentar los conocimientos sobre gestión de transacciones y concurrencia presentados en las clases teóricas y repasar los conocimientos sobre *triggers*.

Enunciado

Para conocer el número de visitas a ciertas páginas de un sitio web, se utiliza una tabla en una base de datos Oracle en la que se almacena el identificador de cada página, un nombre simbólico de la página y el número de visitas que recibe. Cada vez que un usuario accede a una página, se debe incrementar el número de visitas correspondiente.

El esquema relacional y una posible extensión inicial son las que pueden verse en la **Figura 1**.

Páginas	
PK	**id**
	nombre contador

paginas

id	nombre	contador
1	Portada	0
2	Noticias	0
3	Resumen	0
4	Comentarios	0
5	Eventos	0

Figura 1: Modelo relacional y extensión inicial

Ejercicios

1. Cree una tabla en Oracle acorde al enunciado y al esquema relacional de la **Figura 1**. Utilice una secuencia y un *trigger* para simular que la columna id tiene un comportamiento *autonumérico* como en Microsoft Access o en MySQL. Use valores por defecto para la columna contador.

2. Inserte los datos correspondientes a la extensión inicial de la **Figura 1** usando el código más sencillo posible al aprovechar el trigger y los valores por defecto creados en el ejercicio anterior (inserte filas asignando valores sólo a la columna nombre).

3. Abra una ventana SQL*Plus y pida a su compañero de prácticas que haga lo mismo en su ordenador. Asegúrense de que el nivel de aislamiento es `READ COMMITED` mediante el siguiente comando en SQL*Plus:

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

   Escojan cualquiera de las páginas insertadas en el ejercicio anterior y ejecuten el siguiente comando en SQL*Plus:

   ```
   select contador from paginas where id = <ID de la página seleccionada>;
   ```

El objetivo de cada alumno es sumar al valor leído en la consulta anterior un número distinto (p.e. un alumno le suma 100 y otro le suma 200). Actualicen cada uno los datos con la siguiente instrucción en SQL*Plus:

```
update paginas set contador = <nuevo valor>
where id = <ID de la página seleccionada>;

commit;
```

¿Hay pérdida de actualizaciones? ¿Es esto siempre un problema?

4. Repitan el ejercicio anterior pero esta vez, cambien el primer comando por:

```
select contador from paginas
where id = <ID de la página seleccionada>
for update;
```

¿Sigue habiendo pérdida de actualizaciones? ¿Qué ocurre? ¿Por qué?

5. Vuelvan a hacer el ejercicio 3 pero esta vez no utilicen la cláusula *for update* en el *select* y cambien el primer comando por el siguiente:

```
set transaction isolation level serializable;
```

¿Sigue habiendo pérdida de actualizaciones? ¿Qué ocurre? ¿Por qué?

6. Oracle no permite *lecturas sucias*, aunque se puede intentar reproducir el escenario descrito en las clases de teoría. Para ello, tanto usted como su compañero de prácticas, vuelvan a establecer el nivel de aislamiento de transacciones al valor por defecto con el siguiente comando en SQL*Plus:

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

A continuación, ejecuten las siguientes transacciones:

Transacción 1	Transacción 2
select contador **from** paginas **where** id=<ID seleccionado>; **update** paginas **set** contador = <nuevo valor> **where** id=<ID seleccionado>;	
	select contador **from** paginas **where** id=<ID seleccionado>; **update** paginas **set** contador = <nuevo valor> **where** id=<ID seleccionado>; **commit;**
rollback;	

¿Hay lecturas sucias en la transacción 2? ¿Y si se cambia el *rollback* final de la transacción 1 por un *commit*?

7. Vuelvan a repetir el ejercicio anterior usando en ambas transacciones (1) *select ... for update* y (2) nivel de aislamiento serializable. ¿Qué ocurre? ¿Por qué?

8. Asegurándose que están en nivel de aislamiento `READ COMMITED`, reproduzca con su compañero el escenario de *lectura no repetible* visto en clase. Ejecuten cada uno una transacción de las siguientes:

Transacción 1	Transacción 2
select * **from** paginas **where** id=<ID seleccionado>;	
	select * **from** paginas **where** id=<ID seleccionado>; **update** paginas **set** contador = <nuevo valor> **where** id=<ID seleccionado>; **commit;**
select * **from** paginas **where** id=<ID seleccionado>; **commit;**	

¿Son las dos lecturas de la transacción 1 iguales? ¿Por qué?

9. Vuelvan a repetir el anterior ejercicio usando nivel de aislamiento `SERIALIZABLE`. ¿Qué ocurre? ¿Por qué?

10. Asegurándose que están en nivel de aislamiento `READ COMMITED`, reproduzca con su compañero el escenario de *lectura fantasma* visto en clase. Ejecuten cada uno una transacción de las siguientes:

Transacción 1	Transacción 2
select * **from** paginas;	
	select * **from** paginas; **insert into** paginas(nombre) **values** ('Fantasma'); **commit;**
select * **from** paginas; **commit;**	

¿Son las dos lecturas de la transacción 1 iguales? ¿Aparece la tupla fantasma? ¿Por qué?

11. Vuelvan a repetir el anterior ejercicio usando nivel de aislamiento serializable. ¿Qué ocurre? ¿Por qué?