

AMPLIACION de BASES DE DATOS

(Profesor : Héctor Gómez Gauchía)

Práctica 3 SEMANA 6 y 7: Transacciones, niveles aislamiento, bloqueos implícitos y explícitos

Resultado:

- Instrucciones y Resultados de las pruebas de cada apartado en un fichero de texto.
- los ficheros .sql de lo que desarrolles en sql y plsql,
- Haz lista de dudas concretas sin resolver sobre tus respuestas para consultar con el profesor

Modo de entrega: No se Entrega

- Si terminas alguno de los apartados indicado en pizarra, avisa al profesor para puntuar la parte de participación en clase.
- Los conceptos de esta práctica se evalúan en un examen de control (tal y como se indica en la Ficha Docente), cuya fecha se avisará a tiempo.

MATERIALES para estudiar la práctica

- Teoría: Tema Transacciones / Teoria-Transacciones-y-Concurrencia.pdf
/ Transacciones-Posibles-Situaciones.pdf
- Ejemplos de PLSQL triggers y procs.: Tema PL/SQL / Ejercicios . . . / PLSQL-ejemplos/
. . . / PLSQLejemplosEjecutablesTeoria
- Ejemplos de PLSQL Dinámico : Tema PL/SQL / Ejercicios . . . / PLSQL-ejemplos/
. . . / PLSQL-dinamico-para-CV.rar

→ Recuerda durante toda la práctica usar en el editor `set serveroutput on` y `el set autocommit off`

→ Ejecuta la **BDejemplo.sql** que está en esta práctica para empezar de cero

APARTADO 2.- Rollback, checkpoints, bloqueos, transacciones autónomas, privilegios

→ Crea las pruebas necesarias (guarda en un fichero las pruebas, breves explicaciones de ellas y sus resultados)

a.- Varios aspectos de usuarios, gestión de espacio, tablas públicas, triggers, procedimientos

Queremos dar una comisión a todos los alumnos por cada nueva inversión. Para ello se pide lo siguiente:

- Crear en el editor del sqlDeveloper, una tabla
`Notificaciones(Usuario_Origen, Fecha, DNI_cli, NombreE, Tipo , Comision)`
`Usuario_Origen` es el usuario que está enviando la notificación (el que ejecuta el procedimiento)
`Fecha`: es el systimestamp del momento de crear la notificación
`Comision`: Calculada respecto a la cantidad de la inversión como se indica debajo
El resto de atributos son los de la inversión que activa el trigger.
- Para que cualquiera pueda insertar filas en esa tabla, da permiso al público.
 - Hacer un trigger **prac36** que después de que se crea una inversión en la tabla **invierte** llame al procedimiento **RegalaComisiones (parámetros necesarios)**
 - Hacer el procedimiento **RegalaComisiones (parámetros necesarios)**, que debe hacer lo siguiente:
Todos los usuarios de Oracle de la asignatura ABD se benefician de cada nueva inversión, siempre que se cumplan ciertas condiciones. Este procedimiento les informa de que les ha tocado una comisión del 2% del

importe de la inversión, escribiendo una fila en la tabla Notificaciones (de cada usuario que cumpla las condiciones) con la cantidad resultante.

- Las condiciones son las siguientes:

- Si existe la tabla Notificaciones de ese usuario.
- Si tiene menos 1800 bloques de espacio libre en su tablespace.
- Si hacemos un sorteo con un número aleatorio (que obtenemos una vez para cada usuario) y coincide con la última cifra del nombre de usuario de Oracle.

b.- ¿Cómo funcionan los Checkpoints y Rollback?

Modifica el procedimiento **RegalaComisiones** para que incluya lo siguiente:

- Asumiendo que cada usuario en el bucle, está recibiendo simultáneamente comisiones de varios usuarios: si, después de insertar nuestra comisión comprobamos que tiene ya comisiones por más de 100 euros, entonces deshacemos la nuestra con rollback.

c.- ¿Cómo funcionan los Rollback?

Modifica el procedimiento **RegalaComisiones** para que incluya lo siguiente:

- Después de haber hecho el proceso para todos los usuarios, si hemos dado más de 1000 euros de comisiones en total, deshacemos todo y volvemos a repetir el proceso entero (como mucho hasta tres veces en total) hasta que el total sea menor de 1000 euros.

d.- ¿Cómo funcionan los bloqueos explícitos de tabla?

Modifica el procedimiento **RegalaComisiones** para que incluya lo siguiente:

- Mientras se está haciendo el proceso del apartado anterior (*si hemos dado más de 1000 euros..*) no queremos que nadie modifique la tabla, así que bloquea esa tabla Notificaciones del usuario al que escribimos, para que nadie la modifique mientras haces el proceso.

Cuando pruebes puede que tu proceso se quede esperando porque haya otros usuarios han bloqueado alguna tabla.

e.- ¿Cuándo terminan los bloqueos de tabla?

En el apartado anterior, el problema es que se quedan bloqueadas todas las tablas. ¿cómo lo resuelves?. Es razonable?

f.- Para entender: ¿cómo funcionan las transacciones autónomas?

-- Ejecuta el procedimiento TRAB_T_1_LINEA_AUTONOMA.sql (instrucciones ejecución al final de ese mismo script) que llama al procedimiento PONE_LINEA_AUTONOMA.sql

- ¿Qué números de transacción da?
- ¿Porqué hay varias transacciones?

g.- ¿Cómo funcionan las transacciones autónomas?

Modifica el trigger **prac36** para que incluya lo siguiente:

- Queremos que, aunque se hiciera rollback a la inversión (en tabla Invierte), la comisión no se anulara
- ¿Qué tendrías que poner para que este trigger se active después de otro supuesto trigger tr_anterior sobre la misma tabla Invierte (comenta la instrucción, puesto que tr_anterior no existe)?
- Haz un procedimiento Prueba_Autónoma para comprobar resultado: inserta en tabla invierte para que se active el trigger y consultando datos de número de transacción en varios momentos, después haz rollback y comprueba qué se mantiene y qué se ha quitado.

h.- Qué características de una instalación con varias transacciones hacen que una transacción deba ser serializable

i.- Qué características de una instalación con varias transacciones hacen que una transacción deba ser read committed

j.- ¿Qué nivel de aislamiento u otro recurso aplicarías para estas situaciones?:

j.1.- Unas transacciones actualizan la tabla ValoresBolsa. Qué nivel de aislamiento se necesita en otra transacción para que vea cuanto antes los últimos cambios? ¿Harías algo en esas transacciones para que se vean los cambios cuanto antes? . Explica el porqué.

j.2.- Unas transacciones actualizan la tabla ControlPuerta, que puede estar abierta, cerrada o bloqueada. ¿Qué harías para que cada transacción vea el estado inmediatamente después de la actualización de la otra transacción? Explica el porqué.

j.3.- Unas transacciones actualizan la tabla MovtosVisa. Nuestra transacción hace el cierre fin de mes a las doce de la noche del último día, contabiliza y marca cada movimiento como procesado. Y al final suma el total de los movimientos marcados. Qué nivel de aislamiento necesitamos? Explica el porqué.

j.4.- Unas transacciones actualizan la tabla MovtosVisa. Nuestra transacción solo lista los movimientos entre dos fechas. Qué tipo de transacción definiremos para agilizar el proceso? Indica la instrucción.

k.- ¿Qué resultado dará cada situación descrita?:

k.1.- En este trozo de procedimiento: ¿qué sucede con el segundo rollback?

```
Savepoint SP1;  
  Proc-update1();  
Savepoint SP2;  
  Proc-update2();  
IF fallo1 then rollback savepoint SP1 end if;  
IF fallo2 then rollback savepoint SP2 end if;
```

k.2.- En este trozo de procedimiento: ¿qué sucede con el segundo rollback?

```
Savepoint SP1;  
  Update . . . .  
  Rollback;  
  Update. . . .  
  Rollback to savepoint SP1;
```