

Bases de Datos
Trabajo Práctico N°2

Fecha de Entrega: 29/06/2011

Fecha de Recuperatorio: 15/07/2011

Enunciado

En la segunda parte del Trabajo Práctico, el objetivo será desarrollar una herramienta vinculada a la recuperación ante fallas en una base de datos.

Una de las propiedades deseadas de todo motor de base de datos es que sea capaz de sobrevivir a fallas en el sistema. Estas fallas pueden producirse por diferentes causas pero provocan que el sistema interrumpa su funcionamiento normal. El *Recovery Manager* es el módulo responsable de asegurar que, al volver a arrancar el sistema, la base de datos quede en un estado consistente.

Típicamente, el *Recovery Manager* es notificado cada vez que una transacción inicia, hace commit, hace abort o modifica alguna de las tablas existentes. El módulo de recuperación es el encargado de almacenar esta información, que será necesaria para recuperar al sistema en el caso de que surja una falla. Para esto cuenta con una estructura llamada *Log* donde se guarda un registro por cada notificación que recibe por parte de las transacciones.

La herramienta que deberán desarrollar en esta parte de la materia se va a centrar en el algoritmo de recuperación conocido como REDO-Logging. Desde la cátedra se les proveerá una interfaz gráfica que les va a ayudar a probar su implementación y que les va a servir como base para desarrollar el Trabajo Práctico. El lenguaje de programación utilizado será *Java*. A continuación presentaremos en detalle los puntos que cada grupo debe resolver:

1) Incorporar Checkpointing No Quiescent

Actualmente, la herramienta soporta las directivas básicas del algoritmo de REDO: START, COMMIT, ABORT y UPDATE. En este punto deben crear las estructuras necesarias para que se pueda incorporar *Checkpointing No Quiescent*. Cabe aclarar que estos cambios deben reflejarse también en la interfaz gráfica de la herramienta.

2) Lograr que se generen sólo logs válidos

Una vez incorporadas las directivas de checkpointing, se pide agregar la lógica necesaria para que desde la herramienta se puedan generar *solamente* logs válidos. Para esto, previamente deberán definir (aclarándolo en el informe) cuáles son las reglas de validación necesarias. Por ejemplo, una transacción T no puede hacer COMMIT si previamente no hizo un START.

3) Recuperación ante una falla

Interesará conocer las acciones que deberá realizar el motor de base de datos si falla y cuenta con ese log para la recuperación.

Nuevamente, se deberá poder usar la herramienta para llevar esto a cabo y será necesario que se detallen paso a paso las acciones que el motor tiene que realizar para volver la base de datos a un estado consistente. Para realizar esto, deben hacer uso del algoritmo de REDO. Aclaramos que debe especificarse paso a paso lo que debería hacer el motor para recuperarse pero NO debe modificarse el log utilizado como entrada.

Casos de Test

Para validar el funcionamiento de la aplicación, se podrán utilizar dos tipos de test. El primero es utilizando la herramienta visual que permite generar manualmente el log y luego verificar sus resultados. Mientras tanto, el segundo es utilizando tests de unidad (en este caso usamos *JUnit*). La cátedra proveerá casos de test en ambos formatos.

Consideraciones

Para la resolución deberán tener en cuenta las siguientes consideraciones:

- Para la corrección no sólo se evaluará que lo implementado funcione correctamente sino también la calidad del código generado. Con calidad nos referimos a usar comentarios, nombre declarativo para las variables o métodos, uso de métodos auxiliares, evitar duplicar código, etc.
- La interfaz gráfica cuenta con la funcionalidad básica y es bienvenida cualquier modificación que tienda a mejorarla.

Entrega

La entrega deberá contar con el código que implementa lo pedido, un breve informe con detalles de implementación, decisiones tomadas y todo lo que crean conveniente y, finalmente, casos de test *proprios* (incluye la entrada y los resultados esperados).

En los laboratorios, se llevará a cabo una demo con su tutor asignado donde presentarán casos de test utilizando la herramienta visual. Además, la cátedra llevará sus propios tests de unidad para validar también el funcionamiento de la herramienta.

Bibliografía

"Database Management Systems", Raghu Ramakrishnan – Johannes Gehrke, 2° Edición