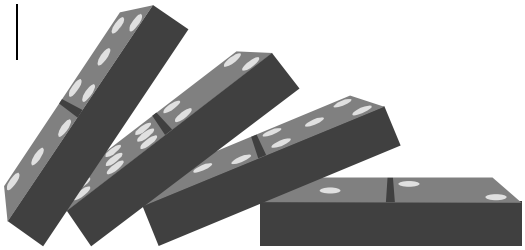

BD's distribuídas
Recuperación y control de concurrencia.

Objetivo:

Profundizar en las particularidades de la gestión de recuperación y concurrencia en estos entornos.

Desarrollo:

- **Recuperación de transacciones**
  - **Concepto de coordinador**
  - **Atomicidad de transacciones**
  - **Robustez del sistema**

- **Control de concurrencia**
  - **Datos en una sola localidad**
    - Concepto de coordinador único
  - **Datos repetidos en varias localidades**
  - **Gestión de bloqueos**

Sistemas Distribuidos.
Recuperación y control de concurrencia.

♦ Recuperación de transacciones

Debe garantizarse la atomicidad de las transacciones. El efecto final de una ejecución debe ser el mismo que si se ejecutara en una sola localidad.

Para conseguir esto, el gestor debe realizar funciones adicionales a las ya estudiadas. Dentro del conjunto de SGBD incluidos en el sistema distribuido, uno de ellos actúa como coordinador para la ejecución de cada transacción. El coordinador:

- Divide la transacción en varias subtransacciones que distribuye a las localidades apropiadas
- Coordina la terminación de la transacción, mediante el uso del commit en dos fases:

  - Fase 1.
    Ante una petición de commit de la transacción, el coordinador graba los registros de bitácora apropiados y envía una petición de preparación para commit al resto de los gestores participantes en la transacción.
    El resto de gestores realizan el proceso de commit como si funcionaran de forma aislada, y envían el resultado al coordinador (commit o rollback).

  - Fase 2.
    El coordinador recoge las respuestas de cada una de las localidades participantes. Si todas responden commit, el coordinador les envía el mensaje de commit, que cada una recibe y graba en su log. Si alguna de las respuestas es rollback o no es recibida en un tiempo determinado, el coordinador envía a todas las localidades el mensaje de rollback, y cada localidad debe deshacer su parte en la transacción.

- **Robustez**

  En sistemas distribuidos aparecen un conjunto de tipos de fallos que hacen más compleja la gestión de la recuperación. El gestor es capaz de mantener la consistencia de los datos si se le permite actuar. Sin embargo, hay situaciones que podrían requerir la actuación manual, como es el caso de localidades no coordinadoras en las que se ha perdido la conexión con la coordinadora para transacciones que se hallan entre la fase 1 y la 2 del proceso de commit. Estas transacciones se encuentran en estado en duda (in-doubt), y el gestor no puede tomar una decisión hasta que la conexión se restablezca (desconoce la decisión del coordinador). El gestor mantiene los bloqueos adquiridos por esa transacción hasta su resolución. En estas situaciones, lo más recomendable es no actuar manualmente y dejar que el gestor resuelva la situación, si es posible.

♦ Control de concurrencia

- **Datos en una sola localidad**

  Se diferencian con respecto a los sistemas centralizados en la mensajería añadida:
  - 2 mensajes por bloqueo
  - 1 mensaje por desbloqueo

  La gestión de bloqueos se complica, pues puede realizarse en varias localidades.

  Otra opción es disponer de un coordinador único de bloqueos para todas las localidades. En este caso, la mensajería intercambiada es la misma que se ha citado y la gestión de bloqueos sería igual que para un sistema centralizado. Sin embargo, pueden producirse cuellos de botella en el coordinador de bloqueos y si falla el coordinador el sistema completo queda inutilizable.

- **Datos replicados en varias localidades**

  Existen varias posibilidades:
  - Marcar una localidad como lugar necesario para solicitar bloqueo (primero y único)
  - Pedir el bloqueo a todas las localidades
  - Mixto

- **Gestión de bloqueos**

  El control de bloqueos mediante el grafo de espera se complica, pues aunque un grafo local no tenga ciclos, sí puede existir si se considera el conjunto de las localidades.

  Para observar esta globalidad, el algoritmo de construcción del grafo se complica:

  Se añade $t_i \rightarrow t_{ex}$ si t_i está esperando un dato de otra localidad que se halla bloqueado por cualquier otra transacción
  Se añade $t_{ex} \rightarrow t_i$ si existe una transacción t_i en otra localidad que está esperando un dato que se halla en esta localidad

  Cuando se detecta un ciclo en el que participa una t_{ex}, se intercambian mensajes con la otra localidad, para permitir la detección del deadlock y actuar en consecuencia.

---o---o---o---