

Transactions

Cada operação realizada dentro de um banco de dados, é tratada como uma transação independente, com início, meio e fim.

READ COMMITTED - Nível padrão de configuração na maioria dos BD 's, ler e trabalhar em cima de algo já aplicado no BD.

- Outro papel do controle transacional é impedir que um usuário altere uma informação ao mesmo tempo que outro usuário está alterando ela. Ou seja, “quem chegar primeiro vence”

Como é realizado:

Início

- Insert
- Update
- Delete

Outras operações como consulta simples (DQL), ou Query, mas o foco está nesses 3 acima.

Resolução

- Commit - Comando enviado ao BD, para confirmar minhas alterações. Enquanto não houver confirmação, outros usuários não verão as alterações que eu fiz.
- Rollback - É quando eu desisto da minha operação, e o banco desfaz a operação que eu fiz (podendo ser tanto inserção, alteração e até mesmo remoção de dados).

Fim

Aqui se tem os dados novos após o Commit, ou a resolução após o Rollback.

ACID

As transações trabalham por meio do conceito ACID que é:

Atomicidade

- O sistema deve garantir que toda operação seja executada com sucesso, sendo assim, um fim, que vai ser um Commit ou um Rollback (o mais normal de se acontecer é um Rollback).

Consistência

- O sistema deve garantir a unicidade de chaves, restrições de integridade lógica, etc. Eu não posso infringir as regras que violem o que eu modelei no DER e depois apliquei no DDL, vulgo banco de dados.

Isolamento

- O BD deve permitir várias transações podem acontecer simultaneamente no mesmo registro, ou parte dele.

Durabilidade

- Depois de aplicar o Commit, mesmo após erros, quedas de energias, etc, as alterações devem ser aplicadas mesmo assim após sua restauração.

Normalmente os bancos de dados, os SGBD 's, mantém um log transacional para no caso de algum erro no sistema ele mesmo assim aplicar os dados no sistema.