BANCO DE DADOS II

Transactions

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

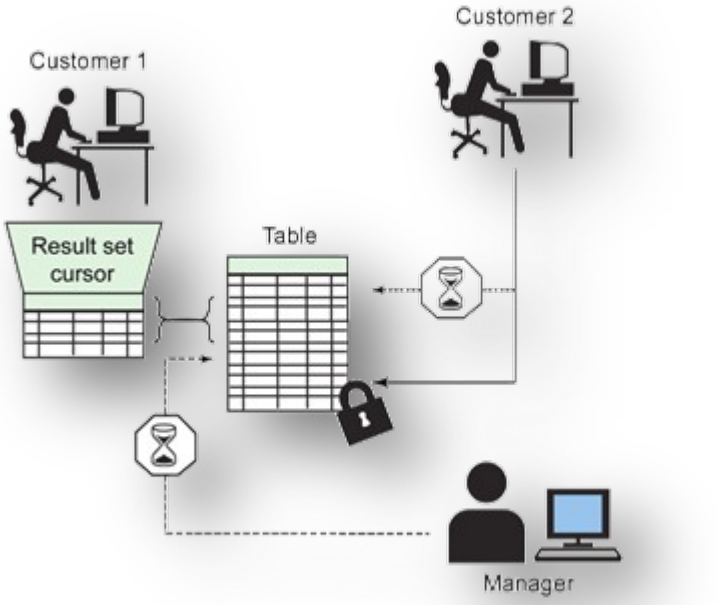
Msc. Eliezio Soares
eliezio.soares@ifrn.edu.br | https://docente.ifrn.edu.br/elieziosoares

CONTROLE DE CONCORRÊNCIA E GARANTIA DE CONSISTÊNCIA

Situações problemas:

- O que acontece se vários usuários editam, ao mesmo tempo, uma mesma tupla em um banco de dados?

- O que ocorre se um usuário executa uma query ao mesmo tempo que outro executa uma declaração DELETE?



CONTROLE DE CONCORRÊNCIA E GARANTIA DE CONSISTÊNCIA NO POSTGRESQL

Para gerenciar o acesso concorrente aos dados e garantir a consistência dos dados é implementado um modelo multiversão (Multiversion Concurrency Control - MVCC).

Cada comando SQL, portanto, vê uma versão do banco de dados (snapshot dos dados), um estado, independente de como os dados estejam atualmente.

Isso ocorre para evitar que algum comando resulte em uma visão inconsistente dos dados produzida por transações concorrentes realizando mudanças nas mesmas tuplas, **isolando transações** para cada sessão.

TRANSACTIONS

Para iniciar uma transação:

- START TRANSACTION (SQL-Standard)
- BEGIN TRANSACTION (PostgreSQL)

```
START TRANSACTION [ transaction_mode [, ...] ]

where transaction_mode is one of:

    ISOLATION LEVEL { SERIALIZABLE | REPEATABLE READ | READ COMMITTED | READ UNCOMMITTED }
    READ WRITE | READ ONLY
    [ NOT ] DEFERRABLE
```

```
BEGIN [ WORK | TRANSACTION ] [ transaction_mode [, ...] ]

where transaction_mode is one of:

    ISOLATION LEVEL { SERIALIZABLE | REPEATABLE READ | READ COMMITTED | READ UNCOMMITTED }
    READ WRITE | READ ONLY
    [ NOT ] DEFERRABLE
```

TRANSACTIONS

START / BEGIN TRANSACTION

- Uma vez iniciada a transação, os comandos SQL associados terão seus resultados persistidos somente após a efetivação da transação.

- Work | Transaction
  - Não possuem efeito, mantém-se apenas por compatibilidade. São opcionais.

- Transaction_mode
  - ISOLATION LEVEL { SERIALIZABLE | REPEATABLE READ | READ COMMITTED | READ UNCOMMITTED }
  - READ WRITE | READ ONLY
  - [NOT] DEFERRABLE **(extensão PostgreSQL)**

TRANSACTIONS MODE ISOLATION LEVEL

- READ COMMITTED
  - Uma declaração pode ver apenas linhas comitadas (confirmadas) antes da query começar.
  - **Default Level**

- REPEATABLE READ
  - Todas as declarações da transação atual podem ver apenas linhas comitadas antes da primeira query ou modificação de dados executada nessa transação.

- SERIALIZABLE (**mais rigorosa**)
  - Todas as declarações da transação atual podem ver apenas linhas comitadas antes da primeira query ou modificação de dados executada nessa transação.

  - Qualquer execução concorrente de um conjunto de transações SERIALIZABLE deve produzir o mesmo efeito nos dados como se cada transação estivesse sendo executada uma após a outra (em série).

- READ UNCOMMITTED
  - SQL Standard
  - No PostgreSQL é tratada como uma READ COMMITED

TRANSACTIONS MODE ISOLATION LEVEL

- Exceto no caso das transações SERIALIZABLE, as demais produzem fenômenos resultantes da interação entre transações concorrentes.
- FENÔMENOS:
  - **Dirty Read**
    - Uma transação lê dados escritos por uma transação concorrente não comitada.
  - **Nonrepeatable Read**
    - Uma transação relê dados já lidos previamente e encontra dados modificados por outra transação (comitados enquanto esta transação estava em execução).
  - **Phantom Read**
    - Uma transação re-executa uma query retornando um conjunto de linhas que satisfaz uma condição de busca e se depara com um resultado modificado por outra transação recentemente comitada.
  - **Serialization Anomaly**
    - O resultado da confirmação com sucesso de um grupo de transações é inconsistente com todas as ordens possíveis de execução dessas transações uma por vez.

TRANSACTIONS MODE ISOLATION LEVEL

Isolation Level	Dirty Read	Nonrepeatable Read	Phantom Read	Serialization Anomaly
Read uncommitted	Allowed, but not in PG	Possible	Possible	Possible
Read committed	Not possible	Possible	Possible	Possible
Repeatable read	Not possible	Not possible	Allowed, but not in PG	Possible
Serializable	Not possible	Not possible	Not possible	Not possible

TRANSACTIONS MODE

- READ WRITE
  - Declara que a transação realiza escrita e leitura
- READ ONLY
  - Declara que a transação somente lê dados, não modifica-os
  - [NOT] DEFERRABLE **(extensão PostgreSQL)**

"The DEFERRABLE transaction property has no effect unless the transaction is also SERIALIZABLE and READ ONLY.
When all three of these properties are selected for a transaction, the transaction may block when first acquiring its snapshot, after which it is able to run without the normal overhead of a SERIALIZABLE transaction and without any risk of contributing to or being canceled by a serialization failure. This mode is well suited for long-running reports or backups."

COMMIT

- COMMIT confirma a transação atual.
- Todas as modificações feitas pela transação passam a ser visíveis para outras e são persistidas no banco de dados, mesmo que um problema ocorra.

```
COMMIT;
```

ROLLBACK

- ROLLBACK aborta a transação atual.
- Todas as modificações feitas pela transação são descartadas.

```
ROLLBACK;
```

EXEMPLO

```
BEGIN;
        insert into tstdel values (1,'teste', current_timestamp);
        insert into tstdel values (2,'teste2', current_timestamp);
        SELECT * FROM tstdel;
COMMIT;
```

EXEMPLO

```
BEGIN WORK;
        insert into tstdel values (1,'teste', current_timestamp);
        insert into tstdel values (2,'teste2', current_timestamp);
        SELECT * FROM tstdel;
COMMIT WORK;
```

EXEMPLO

```
BEGIN TRANSACTION;
        insert into tstdel values (1,'teste', current_timestamp);
        insert into tstdel values (2,'teste2', current_timestamp);
        SELECT * FROM tstdel;
COMMIT TRANSACTION;
```

EXEMPLO

```
BEGIN;
        insert into tstdel values (1,'teste', current_timestamp);
ROLLBACK;
```

EXEMPLO

```
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;

delete from historicos_escolares where mat_alu = 501923 and cod_disc = 501931

COMMIT;
```

DÚVIDAS?

REFERÊNCIAS BIBLIOGRÁFICAS

http://paposql.blogspot.com.br/2013/05/os-conceitos-da-propriedade-acid.html

PostgreSQL 9.6.6 Documentation. Disponível em: <https://www.postgresql.org/docs/9.6/static/mvcc-intro.html>. Acesso em 18 Out. 2017.