Universidade Tecnológica Federal do Paraná
Curso: Tecnologia em Análise de Sistemas
Disciplina: Banco de Dados Avançados

LABORATÓRIO
Transações no PostgreSQL

Crie uma nova base de dados

```
postgres=# CREATE DATABASE exercicio;
postgres=# \c exercicio
```

Crie uma tabela e insira valores

```
exercicio=# create table aluno (id serial primary key, nome varchar(50));
exercicio=# INSERT into aluno (nome) values ('Huguinho'), ('Zezinho'), ('Luisinho'), ('Juquinha'),
('Joaozinho');
```

Criamos 5 alunos em uma nova base. Agora vamos ao primeiro cenário. Para isto será necessário que você abra duas instâncias do psql ou qualquer outra ferramenta cliente.

Inicie uma transação em cada uma executando em ambas a instrução:

```
exercicio=# begin;
```

Na primeira, execute o seguinte comando:

```
exercicio=# UPDATE aluno set nome = 'Jose da Silva' where id = 2;
exercicio=# select * from aluno;
    id |     nome
    ----+---------------
    1 | Huguinho
    3 | Luisinho
    4 | Juquinha
    5 | Joaozinho
    2 | Jose da Silva
    (5 rows)
```

Veja que o nome de Zezinho agora é Jose da Silva.
Realize a mesma consulta na segunda instância do psql:

```
exercicio=# select * from aluno;
    id |   nome
    ----+-----------
    1 | Huguinho
    2 | Zezinho
    3 | Luisinho
    4 | Juquinha
    5 | Joaozinho
    (5 rows)
```

Como você atentou, os alunos continuam com seus nomes intactos. O que a primeira transação modificou não foi sentido pela segunda. O motivo é que se a primeira voltar atrás, a segunda já teria lido e tomado suas

próprias decisões baseando-se em dados inconsistentes. Confira como faríamos para reverter (execute na primeira transação):

```
exercicio=# ROLLBACK ;
exercicio=# select * from aluno;
        id |   nome
        ----+-----------
        1 | Huguinho
        2 | Zezinho
        3 | Luisinho
        4 | Juquinha
        5 | Joaozinho
        (5 rows)
```

Nossa base voltou ao estágio em que estava no momento do BEGIN.

Unrepeatable Reads

Leituras não repetíveis acontecem quando uma transação lê algum dado em determinado momento, posteriormente outra transação altera ou deleta e executa o COMMIT. A primeira volta a ler os mesmos dados e constata que eles não estão mais como antes. Este não chega a ser um problema em muitos sistemas, a ponto do PostgreSQL, em seu nível de isolamento padrão, permitir que ocorra. Vejamos:

PSQL1: Iniciamos uma nova transação (aquela anterior foi desfeita com ROLLBACK) e atualizamos novamente o nome de Zezinho.

```
exercicio=# BEGIN ;
        exercicio=# UPDATE aluno set nome = 'Jose da Silva' where id = 2;
```

PSQL2: Como visto antes, nada foi percebido pela outra transação.

```
        exercicio=# select * from aluno;
        id |   nome
        ---+-----------
        1 | Huguinho
        2 | Zezinho
        3 | Luisinho
        4 | Juquinha
        5 | Joaozinho
        (5 rows)
```

PSQL1: Confirmamos a alteração

```
        exercicio=# COMMIT ;
```

PSQL2: Agora a alteração foi sentida

```
        exercicio=# select * from aluno;
        id |     nome
        ---+--------------
        1 | Huguinho
        3 | Luisinho
        4 | Juquinha
        5 | Joaozinho
        2 | Jose da Silva
        (5 rows)
```

Phantom Read

O problema das leituras fantasmas é parecido com o das leituras não repetíveis, o que muda é que uma das transações em vez de alterar o conjunto de dados, insere um dado novo.

Vamos refazer a dinâmica. Primeiramente, encerre as transações das duas instâncias do psql. Neste momento, tanto faz encerrar com COMMIT ou com ROLLBACK. Se a transação não estiver em curso, você receberá uma mensagem como esta:

WARNING: there is no transaction in progress

PSQL1: Inicie a transação e insira duas novas alunas

```
# begin;
BEGIN
exercicio=# INSERT INTO aluno (nome) values ('Chiquinha'), ('Mariazinha');
INSERT 0 2
```

PSQL2: Inicie a transação e consulte a lista de alunos

```
# begin;
BEGIN
exercicio=# select * from aluno;
id |     nome
----+---------------
1 | Huguinho
3 | Luisinho
4 | Juquinha
5 | Joaozinho
2 | Jose da Silva
(5 rows)
```

Ninguém novo até agora.

PSQL1: Confirme sua transação com um COMMIT

```
exercicio=# COMMIT ;
COMMIT
```

PSQL2: Veja que as novas alunas agora estão presentes

```
exercicio=# select * from aluno;
id |     nome
----+---------------
1 | Huguinho
3 | Luisinho
4 | Juquinha
5 | Joaozinho
2 | Jose da Silva
6 | Chiquinha
7 | Mariazinha
(7 rows)
```