Universidade Federal do Ceará
Campus Quixadá
CSGBD
Prof.ª Lívia Almada

Essa atividade deve ser executada no terminal via psql.

1- Crie um banco de dados e uma tabela chamada "T", com três colunas: id (do tipo inteiro echave primária), s (do tipo cadeia de caracteres com um comprimento variando de 1 a 40 caracteres), e si (número inteiro).

CREATE TABLE T (id INT NOT NULL PRIMARY KEY, s VARCHAR(40), si SMALLINT);

Insira algumas linhas na tabela recém criada:

```
INSERT INTO T (id, s) VALUES (1, 'first');
INSERT INTO T (id, s) VALUES (2, 'second');
INSERT INTO T (id, s) VALUES (3, 'third');
SELECT * FROM T ;
```

Um comando "SELECT * FROM T" confirma que as três novas linhas foram anexadas a tabela (observe os valores NULL registrados na coluna "si").

Tendo em mente o que foi dito até agora sobre o tópico de transações SQL, é feita uma tentativa de cancelar / reverter a transação atual, emitindo o seguinte comando:

```
ROLLBACK;
SELECT * FROM T ;
```

1.1 Qual foi o resultado obtido? As ações foram desfeitas? Procure o significado do modo AUTOCOMMIT, explique-o relacionando com o resultado obtido.

Agora execute os seguintes comandos em SQL.

```
START TRANSACTION;
INSERT INTO T (id, s) VALUES (4, 'fourth');
SELECT * FROM T ;
ROLLBACK;
```

SELECT * FROM T;

1.2 - Compare os resultados obtidos dos últimos comandos com o do item 1.

2 – Desabilite o AUTOCOMMIT. Pelo terminal execute :

\set AUTOCOMMIT OFF

Verificando a modificação:
\echo :AUTOCOMMIT

2.1 – Exclua todas as linhas da tabelca, com exceção de uma.

DELETE FROM T WHERE id > 1;
COMMIT;

2.2 - Insira novas linhas novamente.
INSERT INTO T (id, s) VALUES (6, 'sixth');
INSERT INTO T (id, s) VALUES (7, 'seventh');
SELECT * FROM T;

ROLLBACK;
SELECT * FROM T;

Execute COMMIT; para finalizar a transação.

Qual é a vantagem / desvantagem de usar a instrução "SET TRANSACTION", como comparado ao uso do "SET AUTOCOMMIT", para desligar o SGBD do padrão AUTOCOMMIT?

Já é sabido que algumas instruções SQL são categorizadas como sendo da Linguagem de Definição de Dados (DDL), e alguns outros como sendo da Linguagem de Manipulação de Dados (DML). Exemplos do primeiro são instruções como CREATE TABLE, CREATE INDEX e DROP TABLE, enquanto exemplos da segunda categoria (DML) são instruções como SELECT,
INSERT e DELETE. Tendo em mente o que precede, vale a pena investigar ainda mais as ações da instrução ROLLBACK. Execute o seguinte.

INSERT INTO T (id, s) VALUES (9, 'will this be committed?');
CREATE TABLE T2 (id INT);

```
INSERT INTO T2 (id) VALUES (1);
SELECT * FROM T2;
ROLLBACK;
COMMIT;
```

2.3- Quais as conclusões que você tira a respeito do ROLLBACK sobre criação de tabelas?

3 – Ligue o AUTOCOMMIT novamente. Considere a base de dados condominio utilizada anteriormente. Considere ainda a existência de duas transações, os quais são mostradas a seguir. Inicie cada uma delas em um console diferente, simulando que há dois clientes.

BEGIN
TRANSACTION ISOLATION LEVEL SERIALIZABLE;
select pg_sleep(10);
UPDATE l01_morador SET nrofamiliares=8 WHERE mcpf=4;
select pg_sleep(60);
SELECT * FROM l08_despesa;
COMMIT;
	BEGIN
	TRANSACTION ISOLATION LEVEL SERIALIZABLE;
	UPDATE l01_morador SET nrofamiliares=10 WHERE mcpf=4;
	COMMIT;

3.1 - Ao executar concorrente as duas transações acima, **existe algum conflito entre elas? Reporte o erro ocorrido e explique o que o acasinou**.

3.2 - Qual é o estado que você espera obter na tupla com “mcpf=4” da tabela l01_morador, se as transações abaixo forem executadas de maneira serial?

3.3 – Agora altere as transações para executarem no nível Read uncommitted. Houve alguma alteração na execução? Explique.

3.4. Explique cada um dos níveis de isolamento possíveis para execução das transações.