

BANCO DE DADOS II

Ministrado por Professor(Doutor) Guilherme Dal Bianco,

Desenvolvido por Jardel O. Duarte.

Documentação escrita Trabalho Transaction DB

0- Criar o esquema do banco de dados utilizando a linguagem de programação. Exemplo: usar a linguagem para criar a tabela, usar o drop caso ela já exista, etc. O banco de dados deverá fazer referência a movimentações bancárias, contendo no mínimo uma tabela de clientes(nome, cpf, dataNascimento, saldoEmConta,...)

1- Como construir uma transação na linguagem Javascript.

```
async function execute() {

  try {

    today = new Date();

    //Init the postgres transaction

    await conn.query("BEGIN");

    //Insert 100k tuples

    try {

      //Declare string for sql statement

      const sqlString = `INSERT INTO clientes (id, nome, cpf, datanascimento, saldoemconta) VALUES ($1, $2, $3, $4, $5);`;

      const values = [1, 'Maggie', 42, '2013-10-10', 4200];

      //Pass sql string to the query method

      await conn.query(sqlString, values, function(err, result) {

        console.log("client.query() SQL result:", result);

        if (err) {

          console.log("\nclient.query():", err);

        }

      });

    }

  }

}
```

```
        //Rollback before executing another transaction

        conn.query("ROLLBACK");

        console.log("Transaction ROLLBACK called");

    }

    else {

        conn.query("COMMIT");

        console.log("client.query() COMMIT row count:", result.rowCount);

    }

});

} catch (er) {

    //Rollback before executing another transaction

    conn.query("ROLLBACK");

    console.log("client.query():", er);

    console.log("Transaction ROLLBACK called");

}

}

finally {

    console.log("Client is released");

}

}
```

Return;

Table created in postgresql

Client is released

client.query() COMMIT row count: 1

{ id: 1,

nome: 'Maggie',

cpf: 42,

datanascimento: 2013-10-10T03:00:00.000Z,

saldoemconta: 4200 }

Was inserted 0 rows in 16 ms(t)

(Função que garante uma transação funcional em Javascript).

2- Devo setar a flag de autocommit como false, ou já é implícito da linguagem? Fazer testes via código para comprovar tal afirmação.

Segundo algumas pesquisas foi confirmado que sim, o autocommit é incluído no javascript, ainda que existam poucas documentações que trazem definições sobre a biblioteca que tem esta flag. Foram muitas buscas e poucas resoluções, as propostas encontradas não tiveram nenhuma assertividade quando testadas, entretanto a intenção de fazer mais buscas relacionadas ao autocommit é interessante.

3- A linguagem exige o comando de “commit” para finalizar a transação?

Não, esta observação foi verificado quando inserido diversas tuplas em um bd postgres... Porém não foram feitas análises relacionadas a diferença de tempo de transações com commit e sem commit para melhores acurácias, mas tudo indica que tem a mesma média de tempo supondo que a estrutura da transação continua sendo a mesma (outro detalhe é a garantia sobre a flag de autocommit estar setada como true).

Suposições a serem desconsideradas em caso de err...

Funções a serem implementadas:

1- Fazer a inserção de no mínimo 80000 tuplas, criadas aleatoriamente, em uma tabela de clientes utilizando transações implícitas vs. utilizando transações explícitas (comando insert a cada nova inserção). Compare o tempo de inserção. Explique o motivo da variação do tempo.

```
//Was inserted 80000 rows in 75781 ms(t) explicits
```

```
//Was inserted 80000 rows in 167724 ms(t) implicits
```

Essa diferença pode ser considerada à partir da estruturação da transação, normalmente quando trabalhando com transações se trabalha com grandes grupos de dados, ou seja a ação acontece com disparos sequenciais, diferente de uma inserção sem instruções (apenas queries, mesmo que diversas queries). Podemos afirmar isto com o nível ACID proposto nas transações que significa: atomicidade, consistência, isolamento e durabilidade.

2- Implementar uma função que causa um rollback em uma transação entre o processo de movimentação de uma conta para outra. Ou seja, o erro ocorre depois de descontar o valor de uma conta e antes de adicionar o valor na outra conta.

```
async function execute(){

  try {

    today = new Date();

    //Init the postgres transaction

    await conn.query("BEGIN");

    //Insert 100k tuples

    try {

      await conn.query( `update clientes set saldoemconta = 2000 where
id = 1`, function(err, result) {

        // console.log("client.query() SQL result:", result);

        if (err) {

          console.log("\nclient.query():", err);

          //Rollback before executing another transaction

          conn.query("ROLLBACK");

          console.log("Transaction ROLLBACK called");
```

```

    }

    else {

        conn.query("ROLLBACK");

        // console.log("client.query() COMMIT row count:",
result.rowCount);

    }

});

} catch (er) {

    //Rollback before executing another transaction

    conn.query("ROLLBACK");

    console.log("client.query():", er);

    console.log("Transaction ROLLBACK called");

}

}

finally {

    console.log("Client is released");

}

}

```

Return:

Client is released

Transaction ROLLBACK called

{ id: 2,

nome: 'John Lennon',

cpf: 12345671,

datanascimento: 2020-11-01T03:00:00.000Z,

saldoemconta: 3800 }

{ id: 1,

nome: 'Paul McCartney',

cpf: 1234567,

datanascimento: 2020-11-01T03:00:00.000Z,

saldoemconta: 2000 }

Was inserted 0 rows in 19 ms(t)

Obs: após finalizar a transação fiz um select * from clientes no postgres e os dados não haviam sido alterados.

3- Implementar uma função que gere um commit em uma transação com mais de uma operação de escrita.

```
async function execute() {
  try {
    today = new Date();
    //Init the postgres transaction
    await conn.query("BEGIN");
    //Insert 100k tuples
    try {
      await conn.query( `insert into clientes (id, nome, cpf,
dataNascimento, saldoEmConta)
      values (1, 'Paul McCartney', 1234567, '2020-11-01', 4000),
              (2, 'John Lennon', 12345671, '2020-11-01', 3800)` ,
function(err, result) {
      // console.log("client.query() SQL result:", result);
```

```

        if (err) {
            console.log("\nclient.query():", err);
            //Rollback before executing another transaction
            conn.query("ROLLBACK");
            console.log("Transaction ROLLBACK called");
        }
        else {
            conn.query("COMMIT");
            console.log("client.query() COMMIT row count:", result.rowCount);
        }
    });
} catch (er) {
    //Rollback before executing another transaction
    conn.query("ROLLBACK");
    console.log("client.query():", er);
    console.log("Transaction ROLLBACK called");
}
}
finally {
    console.log("Client is released");
}
}

```

Return:

Table created in postgresql

Client is released

client.query() COMMIT row count: 2

{ id: 1,

nome: 'Paul McCartney',

cpf: 1234567,

datanascimento: 2020-11-01T03:00:00.000Z,

saldoemconta: 4000 }

{ id: 2,

nome: 'John Lennon',

cpf: 12345671,

datanascimento: 2020-11-01T03:00:00.000Z,

saldoemconta: 3800 }

Was inserted 0 rows in 9 ms(t)

UFFS

Chapecó, Brasil 2020