Departamento de Ciências de Computação – SCC
Instituto de Ciências Matemáticas e de Computação – ICMC
Universidade de São Paulo – USP

Laboratório de Bases de Dados
Docente Responsável: Profa. Dra. Cristina Dutra de Aguiar Ciferri
Estagiária PAE: Jaqueline Joice Brito

Exercícios Práticos
Transações em Ambientes Multiusuário

Para a realização dessa prática, são necessários dois usuários: **Usuário1** e **Usuário2**. Escolha um amigo próximo a você e bom trabalho!

PARTE 1 – Transações READ ONLY

Usuário 1. Execute o script completo de criação e inserção de dados referente ao projeto **"empresa de aluguel de carros"**.

Usuário 1. Garanta privilégios ao usuário 2 para que ele possa realizar inserções, remoções, atualizações e consultas nas tabelas empresa e filial.

Usuário 1. Finalize a transação corrente.

Usuário 1. Inicie uma transação como READ ONLY

Usuário 1. Execute a seguinte consulta SQL. Quantas tuplas foram listadas?

```
SELECT empresa.emp_codigo, emp_nome, count(filial_nro) as quantidade
FROM empresa LEFT JOIN filial ON filial.emp_codigo = empresa.emp_codigo
GROUP BY empresa.emp_codigo, emp_nome
ORDER BY empresa.emp_codigo;
```

Usuário 1. Execute a seguinte atualização. O que aconteceu? Por que?

```
UPDATE empresa
SET emp_codigo = 5
WHERE emp_codigo = 4;
```

Usuário 2. Execute a seguinte consulta SQL. Quantas tuplas foram listadas?

```
SELECT Usuario1.empresa.emp_codigo, emp_nome, count(filial_nro) as quantidade
FROM Usuario1.empresa LEFT JOIN Usuario1.filial
ON filial.emp_codigo = empresa.emp_codigo
GROUP BY Usuario1.empresa.emp_codigo, emp_nome
ORDER BY Usuario1.empresa.emp_codigo;
```

Usuário 2. Execute a seguinte inserção. O que aconteceu? Por que?

```
INSERT INTO Usuario1.empresa
VALUES (10, 'teste READ ONLY', 'teste READ ONLY, ' ');
```

Usuário 2. Execute a seguinte consulta SQL. Quantas tuplas foram listadas?

```
SELECT Usuario1.empresa.emp_codigo, emp_nome, count(filial_nro) as quantidade
FROM Usuario1.empresa LEFT JOIN Usuario1.filial
ON filial.emp_codigo = empresa.emp_codigo
GROUP BY Usuario1.empresa.emp_codigo, emp_nome
ORDER BY Usuario1.empresa.emp_codigo;
```

Usuário 1. Execute a seguinte consulta SQL. Quantas tuplas foram listadas?

```
SELECT empresa.emp_codigo, emp_nome, count(filial_nro) as quantidade
FROM empresa LEFT JOIN filial ON filial.emp_codigo = empresa.emp_codigo
GROUP BY empresa.emp_codigo, emp_nome
ORDER BY empresa.emp_codigo;
```

Usuário 1. Finalize a transação corrente.

Usuário 1. Execute a seguinte consulta SQL. Quantas tuplas foram listadas?

```
SELECT empresa.emp_codigo, emp_nome, count(filial_nro) as quantidade
FROM empresa LEFT JOIN filial ON filial.emp_codigo = empresa.emp_codigo
GROUP BY empresa.emp_codigo, emp_nome
ORDER BY empresa.emp_codigo;
```

Usuário 2. Finalize a transação corrente.

Usuário 1. Execute a seguinte consulta SQL. Quantas tuplas foram listadas?

```
SELECT empresa.emp_codigo, emp_nome, count(filial_nro) as quantidade
FROM empresa LEFT JOIN filial ON filial.emp_codigo = empresa.emp_codigo
GROUP BY empresa.emp_codigo, emp_nome
ORDER BY empresa.emp_codigo;
```

Usuário 1. Finalize a transação corrente.

Usuário 2. Finalize a transação corrente.

PARTE 2 – Transações READ WRITE

Usuário 1. Inicie uma transação como READ WRITE

Usuário 1. Execute a seguinte consulta SQL.

```
SELECT empresa.emp_codigo, emp_nome, count(filial_nro) as quantidade
FROM empresa LEFT JOIN filial ON filial.emp_codigo = empresa.emp_codigo
GROUP BY empresa.emp_codigo, emp_nome
ORDER BY empresa.emp_codigo;
```

Usuário 1. Execute a seguinte atualização. O que aconteceu? Por que?

```
UPDATE empresa
SET emp_codigo = 5
WHERE emp_codigo = 4;
```

Usuário 2. Execute a seguinte inserção. O que aconteceu?

```
INSERT INTO Usuario1.empresa
VALUES (11, 'teste READ WRITE', 'teste READ WRITE, ' ');
```

Usuário 2. Finalize a transação corrente.

Usuário 1. Execute a seguinte consulta SQL. Quantas tuplas foram listadas? Por que?

```
SELECT empresa.emp_codigo, emp_nome, count(filial_nro) as quantidade
FROM empresa LEFT JOIN filial ON filial.emp_codigo = empresa.emp_codigo
GROUP BY empresa.emp_codigo, emp_nome
ORDER BY empresa.emp_codigo;
```

Usuário 1. Finalize a transação corrente.

Usuário 2. Finalize a transação corrente.

PARTE 3 – Transações no Modo Serializável

Usuário 1. Inicie uma transação no modo de isolação SERIALIABLE

Usuário 1. Execute a seguinte consulta SQL. Quantas tuplas foram listadas?

```
SELECT empresa.emp_codigo, emp_nome, count(filial_nro) as quantidade
FROM empresa LEFT JOIN filial ON filial.emp_codigo = empresa.emp_codigo
GROUP BY empresa.emp_codigo, emp_nome
ORDER BY empresa.emp_codigo;
```

Usuário 2. Execute a seguinte inserção. O que aconteceu? Por que?

```
INSERT INTO Usuario1.empresa
VALUES (12, 'teste SERIALIABLE', 'teste SERIALIABLE', ' ');
```

Usuário 1. Execute a seguinte consulta SQL. Quantas tuplas foram listadas?

```
SELECT empresa.emp_codigo, emp_nome, count(filial_nro) as quantidade
FROM empresa LEFT JOIN filial ON filial.emp_codigo = empresa.emp_codigo
GROUP BY empresa.emp_codigo, emp_nome
ORDER BY empresa.emp_codigo;
```

Usuário 2. Finalize a transação corrente.

Usuário 1. Execute a seguinte consulta SQL. Quantas tuplas foram listadas?

```
SELECT empresa.emp_codigo, emp_nome, count(filial_nro) as quantidade
FROM empresa LEFT JOIN filial ON filial.emp_codigo = empresa.emp_codigo
GROUP BY empresa.emp_codigo, emp_nome
ORDER BY empresa.emp_codigo;
```

Usuário 1. Execute comandos que permitam a leitura da tupla com emp_codigo = 12 inserida pelo usuário 2.

Usuário 1. Finalize a transação corrente.

Usuário 2. Finalize a transação corrente.

PARTE 4 – Transações no Modo Serializável + UPDATE

Usuário 1. Inicie uma transação no modo de isolação SERIALIABLE

Usuário 2. Execute a seguinte atualização. O que aconteceu?

```
UPDATE empresa
SET emp_codigo = 6
WHERE emp_codigo = 5;
```

Usuário 1. Execute a seguinte atualização. O que aconteceu?

```
UPDATE empresa
SET emp_codigo = 13
WHERE emp_codigo = 12;
```

Usuário 1. Finalize a transação corrente.

Usuário 2. Finalize a transação corrente.

PARTE 5 – Transações READ WRITE + UPDATE

Usuário 1. Inicie uma transação no modo de isolação READ WRITE

Usuário 2. Execute a seguinte atualização. O que aconteceu?

```
UPDATE empresa
SET emp_codigo = 7
WHERE emp_codigo = 6;
```

Usuário 1. Execute a seguinte atualização. O que aconteceu?

```
UPDATE empresa
SET emp_codigo = 13
WHERE emp_codigo = 12;
```

Usuário 1. Execute a seguinte consulta SQL. Quantas tuplas foram listadas?

```
SELECT empresa.emp_codigo, emp_nome, count(filial_nro) as quantidade
FROM empresa LEFT JOIN filial ON filial.emp_codigo = empresa.emp_codigo
GROUP BY empresa.emp_codigo, emp_nome
ORDER BY empresa.emp_codigo;
```

Usuário 2. Execute a seguinte consulta SQL. Quantas tuplas foram listadas?

```
SELECT empresa.emp_codigo, emp_nome, count(filial_nro) as quantidade
FROM empresa LEFT JOIN filial ON filial.emp_codigo = empresa.emp_codigo
GROUP BY empresa.emp_codigo, emp_nome
ORDER BY empresa.emp_codigo;
```

Usuário 1. Finalize a transação corrente.

Usuário 2. Finalize a transação corrente.