

Regras:

Exercícios 96 a 105

96. O que são transações?

97. Explique o comando abaixo?

```
BEGIN TRANSACTION
```

```
update contas set saldo = saldo - 100 where num=1234;
```

```
update contas set saldo = saldo + 100 where num=4321;
```

```
COMMIT
```

98. Na questão anterior para desfazer a transação deve-se trocar o comando Commit por?

99. O que é Banco de Dados deve garantir com relação as transações?

100. Como pode ser finalizado uma transação?

101. O que faz uma transação falhar?

102. Quais são as propriedades de uma transação?

103. O que é deadlock?

```
CREATE TABLE EMPLOYEES (  
  EMPLOYEE_ID INT PRIMARY KEY,  
  FIRST_NAME VARCHAR(50),  
  LAST_NAME VARCHAR(50),  
  JOB_TITLE VARCHAR(50),  
  SALARY DECIMAL(10,2),  
  DEPARTMENT_ID INT,  
  SEXO CHAR(1)  
);
```

```
INSERT INTO EMPLOYEES (EMPLOYEE_ID, FIRST_NAME, LAST_NAME, JOB_TITLE, SALARY, DEPARTMENT_ID, SEXO) VALUES  
(130, 'Carlos', 'Souza', 'Analista', 4500.00, 10, 'M'),  
(131, 'Mariana', 'Lima', 'Gerente', 7800.00, 20, 'F'),  
(132, 'Pedro', 'Silva', 'Assistente', 2800.00, 10, 'M'),  
(133, 'Juliana', 'Oliveira', 'Analista', 4200.00, 30, 'F'),  
(134, 'Rafael', 'Pereira', 'Diretor', 15000.00, 40, 'M');
```

104. o funcionário 130 da tabela EMPLOYEES. Em seguida, utilize o ROLLBACK.

105: Exclua o funcionario 130 da tabela EMPLOYEES. Em seguida, utilize o COMMIT.

-- Por fim, tente usar o comando ROLLBACK e veja o que acontece.

106. Inicie uma transação e atualize o salário de um funcionário específico. Use ROLLBACK e verifique se o valor retornou ao original.

107. Inicie uma transação e atualize o salário de um funcionário específico. Use COMMIT e confirme que a alteração persistiu.

108. Exclua um funcionário usando BEGIN TRAN e depois use ROLLBACK para desfazer. Verifique se o registro ainda existe.

109. Exclua um funcionário usando BEGIN TRAN e depois COMMIT. Verifique que o registro foi removido permanentemente.

110. Atualize duas tabelas relacionadas (ex.: EMPLOYEES e DEPARTMENTS) dentro da mesma transação. Use COMMIT.

111. Repita o exercício anterior, mas use ROLLBACK e confirme que nenhuma tabela ficou parcialmente atualizada.

112. Use o nível de isolamento READ UNCOMMITTED e observe leituras sujas durante uma transação não finalizada.

113. Use o nível READ COMMITTED e mostre que leituras sujas não ocorrem.

114. Use REPEATABLE READ e mostre que leituras não repetíveis são impedidas.

115. Use SERIALIZABLE e observe que ele bloqueia leituras e escritas paralelas.

116. Crie uma transação que drope uma tabela. Utilize ROLLBACK para mostrar que a tabela permanece.

117. Crie uma transação que drope uma tabela. Use COMMIT e confirme que a tabela realmente foi excluída.

104. Quais foram suas dúvidas/dificuldades na aula e nesta Atividade?

105. O que pode ser melhorado na aula e nesta Atividade?