

Prova Final - Laboratório de Banco de Dados

Prof.: Dr. Rodrigo Colnago Contreras

19 de junho de 2024

Nome do Aluno:

RA:

1 Salvamento e Recuperação de Dados

1. (1.0) Apresente:

- (a) quais são e detalhe a fundamentação dos componentes que envolvem a teoria de *Backup*.
- (b) quatro tipos de *Backup* juntamente com sua fundamentação.

2. (1.0) Julgue como verdadeiro (V) ou falso (F) e apresente a devida justificativa para tal com respeito aos itens a seguir:

- (a) () Independente de configuração, um *backup* do tipo incremental é menor do que um *backup* do tipo diferencial.
- (b) () B_0 é o estado inicial do banco de dados, ou seja, é apenas a estrutura do banco de dados sem nenhum registro tabular.
- (c) () Sejam t_1 e t_2 dois instantes de tempos do banco de dados tais que $t_1 < t_2$. Então, é sabido que o número de registros de B_{t_1} é inferior ao número de registros de B_{t_2} .
- (d) () O número de registros de qualquer estado do banco B_t é sempre maior que o número de registros de um *backup* diferencial, mesmo que este seja referente a um estado posterior a t .

3. (1.5) Escreva um procedimento que conduza um *backup* sobre o banco de dados **BD_prova**. Especificamente, esse procedimento deve receber dois parâmetros literais:

- **@tipo_de_backup**: parâmetro literal que pode assumir os valores “full” e “diff”. Caso assuma o primeiro, o procedimento deve conduzir um *backup* completo. Caso assuma o segundo, deve conduzir um *backup* diferencial.
- **@diretorio**: parâmetro literal que indica o endereço e o nome do diretório em que o banco de dados deve ser salvo. Por exemplo, esta variável pode assumir o valor “C:/Backup”.

Além disso, o arquivo final de backup deve seguir a respectiva nomenclatura:

BD_bak_<tipo>_<ano>_<mes>_<dia>_<hora>_<minuto>.bak,

na qual,

- <tipo>: é o conteúdo da variável @tipo_de_backup;
 - <ano>: é o ano em que o *backup* será feito;
 - <mes>: é o mês em que o *backup* será feito;
 - <dia>: é o dia em que o *backup* será feito;
 - <hora>: é a hora em que o *backup* será feito;
 - <minuto>: é o minuto em que o *backup* será feito.
4. (1.5) Suponha que o banco de dados BD_prova tenha uma tabela tb_prova. Altere o procedimento definido no exercício anterior para que o mesmo execute o *backup* apenas se a tabela tb_prova teve, pelo menos, 10000 atualizações desde o último *backup* independente do tipo.

2 Processamento de Transações

5. (1.0) Descreva o significado de operações, de transação e de concorrência sobre um banco de dados.
6. (1.0) Explique o que significa a estratégia ACID para processamento de transações em um banco de dados. Além disso, apresente quatro tipos de isolamentos existentes no dialeto T-SQL destacando as diferenças e as semelhanças de cada um com respeito a gravações e a leituras no banco de dados.
7. (1.5) Suponha que em no BD_prova exista uma tabela tb_cliente com a coluna saldo do tipo numérico real e com a chave primária id do tipo numérico inteiro. Crie um procedimento que efetue da **maneira mais segura possível** a transferência de crédito de um cliente para o outro.
Dica: utilize o conceito de transações.
8. (1.5) No código a seguir, o uso do conceito de *savepoints* é aplicado no dialeto T-SQL. Indique quais devem ser as respostas das seleções em forma tabular em cada uma das indicações de comentários (*), (**), (***) e (****).

```

1 DROP TABLE IF EXISTS tb_ex_save_point;
2 CREATE TABLE tb_ex_save_point(
3     id          INT PRIMARY KEY IDENTITY(1,1),
4     valor INT
5 );
6
7 INSERT INTO tb_ex_save_point VALUES(0);
8
9 SELECT * FROM tb_ex_save_point; -- (*)
10
11 BEGIN TRANSACTION tr1
12
13     INSERT INTO tb_ex_save_point VALUES(1);
14
15     SAVE TRANSACTION tr2
16
17         SELECT * FROM tb_ex_save_point; -- (**)
18
19         INSERT INTO tb_ex_save_point VALUES(2);
20

```

```

21      ROLLBACK TRANSACTION tr2
22
23      SAVE TRANSACTION tr3
24
25          INSERT INTO tb_ex_save_point VALUES(3);
26
27          DECLARE @count INT =
28              (SELECT COUNT(*) FROM tb_ex_save_point);
29
30          IF @count <= 2
31          BEGIN
32              COMMIT TRANSACTION tr3;
33          END
34          ELSE
35          BEGIN
36              ROLLBACK TRANSACTION tr3;
37
38              INSERT INTO tb_ex_save_point VALUES(4);
39          END;
40
41      SELECT * FROM tb_ex_save_point; -- (***)
42
43      COMMIT TRANSACTION tr1;
44
45
46      SELECT * FROM tb_ex_save_point; -- (****)

```

Boa Prova!