

Fotos da correção da primeira prova de Laboratório de Banco de Dados

Nome: Guilherme Silva Bisaio - ADS - 4º Semestre

02 | Correção da PROVA 1 (T-SQL Avançado)
70 | 1- (V) No T-SQL, não existe booleano! → PostgreSQL tem
2024 | (F) Não dá para armazenar dados monetários em FLOATS
(V) Valores armazenados em VARCHAR(MAX) e em VARCHAR(70) ocupam a mesma qtd de memória!
(F) Dá para armazenar pontos flutuantes em BIGINT.
(F) Não vai dar erro
DECLARE @pi INT = 3.14156295,
@sin_pi FLOAT = SIN(@pi); → ainda não existe
2 - a) BEGIN TRY
CREATE DATABASE bd-primeira-prova;
END TRY
BEGIN CATCH
PRINT('BD já existe');
END CATCH;
GO
USE bd-primeira-prova;

b) IF NOT EXISTS (SELECT 1 FROM sys.tables WHERE name = 'tb-prova')
BEGIN
CREATE... (resto da criação)
END

3 - DECLARE @i INT = 1;
WHILE @i <= 100
BEGIN
DECLARE @nome VARCHAR(MAX);
SET @nome = CONVERT(VARCHAR(MAX), NEWID()); ← valor aleatório com
valor de 1-1
SET @nome = LEFT(@nome, 20);
SET @nome = REPLACE(@nome, '0', 'a'); ← garante de que tenha
seu caractere
→ Repetir para os demais 1, 2, ..., 100
DECLARE @x FLOAT = (20 + RAND() * 780);
@y FLOAT = // ;
INSERT INTO tb-prova
VALUES (@nome, @x, @y);
SET @i = @i + 1;
END;

```

4- CREATE TABLE tb_copia (
    id INT PRIMARY KEY, -- NOT NULL IDENTITY
    =
);

```

```

DECLARE cursor_1 CURSOR
FOR
SELECT * FROM tb_prova;

```

```

OPEN cursor_1;

```

```

DECLARE @j INT = 1;
@n INT = (SELECT COUNT(*) FROM tb_prova);

```

```

WHILE @j <= @n

```

```

BEGIN

```

```

    DECLARE @id INT,
            @nome VARCHAR(MAX),
            @x FLOAT,
            @y FLOAT;

```

```

    FETCH NEXT FROM cursor_1 INTO @id, @nome, @x, @y;

```

```

    INSERT INTO tb_copia
    VALUES (@id, @nome, @x, @y);

```

```

    SET @j = @j + 1;

```

```

END;

```

```

CLOSE cursor_1;

```

```

DEALLOCATE cursor_1;

```


b) CREATE OR ALTER PROCEDURE [dbo.] ex5itemb

@inf FLOAT,

@sup FLOAT

AS

BEGIN

~~UPDATE~~ UPDATE tb_prova

SET x = dbo.f(x, @inf, @sup),

y = dbo.f(y, @inf, @sup);

END;

c) CREATE OR ALTER TRIGGER [dbo.] ex5itemc

ON tb_prova

AFTER INSERT

AS

BEGIN

UPDATE tb_prova

SET x = dbo.f(x, 100, 150),

y = dbo.f(y, 100, 150)

WHERE id IN (SELECT id FROM INSERTED);

END;

