

CORREÇÃO

Exercício 1

- a. (V) No T-SQL, não existe booleano
- b. (F) Não dá pra armazenar dados monetários em FLOAT
- c. (V) Valores armazenados em VARCHAR(MAX) e em VARCHAR(10) ocupam o mesmo espaço
- d. (F) Dá para armazenar pontos flutuantes em BIGINT
- e. (F) Não vai dar erro

```
DECLARE
    @pi INT = 3.14156295,
    @sin_pi FLOAT = SIN(@pi);

-- Vai dar erro pois @pi ainda não está na memória
```

Exercício 2

a)

```
BEGIN TRY
    CREATE DATABASE bd_primeira_prova;
END TRY
BEGIN CATCH
    PRINT('Banco já existente')
END CATCH;
GO
USE bd_primeira_prova;
```

b)

```

IF NOT EXISTS (SELECT 1 FROM sys.tables WHERE name = 'tb_prova')
BEGIN
    CREATE TABLE ....
    -- Resto do script da criação da tabela
END;

```

Exercício 3

```

DECLARE
    @i INT = 1;
WHILE @i <= 100
BEGIN
    DECLARE
        @nome VARCHAR(MAX),
        @x FLOAT = RAND() * (200 - 20) + 20,
        @y FLOAT = RAND() * (200 - 20) + 20,
    SET @nome = LEFT(CONVERT(VARCHAR(MAX), NEWID()), 20);
    SET @nome = REPLACE(@nome, '0', 'a');
    SET @nome = REPLACE(@nome, '1', 'a');
    SET @nome = REPLACE(@nome, '2', 'a');
    SET @nome = REPLACE(@nome, '3', 'a');
    SET @nome = REPLACE(@nome, '4', 'a');
    SET @nome = REPLACE(@nome, '5', 'a');
    SET @nome = REPLACE(@nome, '6', 'a');
    SET @nome = REPLACE(@nome, '7', 'a');
    SET @nome = REPLACE(@nome, '8', 'a');
    SET @nome = REPLACE(@nome, '9', 'a');
    SET @nome = REPLACE(@nome, '-', 'a');

    INSERT INTO tb_prova
        (nome, x, y)
    VALUES
        (@nome, @x, @y);

```

```
    SET @i = @i + 1;  
END;
```

Exercício 4

```
CREATE TABLE tb_copia  
(  
    id INT PRIMARY KEY, -- Não vai ter o identity  
    nome VARCHAR(MAX),  
    x FLOAT,  
    y FLOAT  
);  
  
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;
```

Exercício 5

a)

```
CREATE OR ALTER FUNCTION dbo.f(@x FLOAT, @inf FLOAT, @sup FLOAT)
RETURNS FLOAT
AS
BEGIN
    DECLARE
        @w FLOAT;

    IF @sup < @inf
    BEGIN
        SET @w = @sup;
        SET @sup = @inf;
        SET @inf = @w;
    END;

    DECLARE
        @retorno FLOAT = @x;
    IF @x < @inf
    BEGIN
        SET @retorno = @inf;
    END;

    IF @x > @sup;
    BEGIN
        @retorno = @sup;
    END;

    RETURN @retorno;
END;
```

b)

```
CREATE OR ALTER PROCEDURE ex_5
@inf FLOAT,
```

```
@sup FLOAT,  
AS  
BEGIN  
    UPDATE  
        tb_prova  
    SET  
        x = dbo.f(x, @inf, @sup),  
        y = dbo.f(y, @inf,, @sup);  
END;
```

c)

```
CREATE OR ALTER TRIGGER dbo.ex5  
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;
```