

LISTA GERAL

Arquivo SQL

ExerciciosPropostos.sql

Script do bd_exames

```
BEGIN TRY CREATE DATABASE bd_exames;

USE bd_exames;

END TRY BEGIN CATCH USE bd_exames;

END CATCH IF EXISTS(
    SELECT
        1
    FROM
        sys.tables
    WHERE
        name = 'tb_exames'
) BEGIN DROP TABLE tb_exames;

END;

CREATE TABLE tb_exames (
    id INT PRIMARY KEY IDENTITY(1, 1),
    -- Identificador único do paciente
    nome VARCHAR(100),
    -- Nome completo do paciente
    data_exame DATE,
    -- Data em que o exame foi realizado
    colesterol_total DECIMAL(7, 2),
```

```

-- Colesterol Total
-- Limite Normal: Menos de 200 mg/dL ([0,200])
LDL DECIMAL(7, 2),
-- Lipoproteína de Baixa Densidade
-- Limite Normal: Menos de 100 mg/dL ([0,100])
HDL DECIMAL(7, 2) -- Lipoproteína de Alta Densidade
-- Limite Normal: Mais de 60 mg/dL ([0,60])
);

```

1. Escreva uma PROCEDURE que adicione um valor inteiro @n de registros aleatórios na tabela acima considerando os seguintes requisitos:

- O nome do paciente deve ser uma string aleatória de 10 caracteres. Dica: use NEWID().
- A data a ser gerada deve ser qualquer data válida entre 1990 e 2024, exceto 29 de fevereiro. Se a data gerada não for válida, esta deve ser ajustada para a data válida mais próxima.
- Os colesterolis devem ser gerados no intervalo [0,300].

Na sequência, adicione 100 registros na tabela utilizando esta PROCEDURE.

```

CREATE OR ALTER PROCEDURE add_exame
@n INT
AS
BEGIN
    DECLARE
        @i INT = 1;
    WHILE @i <= @n
    BEGIN
        DECLARE
            @nome VARCHAR(100) = LEFT(NEWID(), 10),
            @dat_exame DATE = DATEADD(DAY, ABS(CHECKSUM(NEWID)),
            @ldl DECIMAL(7,2) = RAND() *(300 - 0) + 0, -- Dei
            @hdl DECIMAL(7,2) = RAND() *(300 - 0) + 0; -- Dei
        DECLARE
            @colesterol DECIMAL(7,2) = @ldl + @hdl;
    
```

```

        SET @nome = REPLACE(@nome, '-', 'a')
        SET @nome = REPLACE(@nome, '0', 'b')
        SET @nome = REPLACE(@nome, '1', 'c')
        SET @nome = REPLACE(@nome, '2', 'd')
        SET @nome = REPLACE(@nome, '3', 'e')
        SET @nome = REPLACE(@nome, '4', 'f')
        SET @nome = REPLACE(@nome, '5', 'g')
        SET @nome = REPLACE(@nome, '6', 'h')
        SET @nome = REPLACE(@nome, '7', 'i')
        SET @nome = REPLACE(@nome, '8', 'i')
        SET @nome = REPLACE(@nome, '9', 'j')

        INSERT INTO
            tb_exames
                (nome, data_exame, colesterol_total, LDL,
VALUES
                (@nome, @dat_exame, @colesterol, @ldl, @h

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

-- Executa a procedure
EXEC add_exame 100;

```

2. Adicione uma coluna na tabela acima chamada 'condicao' como sendo um VARCHAR(10).

a. O valor associado a essa coluna deve ser:

- i. 'Controle' se todos os valores de colesterol estiverem dentro das faixas consideradas saudáveis;
- ii. 'Paciente' caso contrário.

```

-- Declaração do cursor
DECLARE cursor_condicao CURSOR
FOR
SELECT id, colesterol_total FROM tb_exames
OPEN cursor_condicao;

```

```

-- Varredura dos colestorais
DECLARE
    @id INT,
    @colesterol_total DECIMAL(7,2),
    @condicao VARCHAR(10),
    @i INT = 1,
    @tamanho INT = (SELECT COUNT(*) FROM tb_exames);
WHILE @i <= @tamanho
BEGIN
    -- Uso do cursor
    FETCH NEXT FROM cursor_condicao INTO @id, @colesterol_tot
    IF @colesterol_total < 200
    BEGIN
        SET @condicao = 'Controle';
    END
    ELSE
    BEGIN
        SET @condicao = 'Paciente';
    END

    UPDATE
        tb_exames
    SET
        condicao = @condicao
    WHERE
        id = @id;

    SET @i = @i + 1;
END;

-- Fechamento
CLOSE cursor_condicao;
-- Deletando o cursor
DEALLOCATE cursor_condicao;

```

3. Exercício 3

- a. Delete aleatoriamente um paciente com id entre 2 e 99.
- b. Qual é o paciente com colesterolis mais semelhantes ao paciente ('observação', '2024-08-28', 50, 50, 50, ' - ')?
 - i. Dica 1: Considere os três colesterolis simultaneamente ao fazer as comparações.
 - ii. Dica 2: Use a distância euclidiana (raiz quadrada da soma das diferenças quadráticas).

```
-- consultar o registro
-- descobrir a distancia entre cada um dos colestoris e o do
-- armazena a distancia em uma variavel e o id do guerreiro
-- no final ele ficar com o id do que tem a menor distancia
DECLARE cursor_distancia CURSOR
FOR
SELECT id, colesterol_total, ldl, hdl FROM tb_examenes
OPEN cursor_distancia;
DECLARE
    @id INT,
    @colesterol_total DECIMAL(7, 2),
    @LDL DECIMAL(7, 2),
    @HDL DECIMAL(7, 2),
    @distancia DECIMAL(7, 2) = 1000,
    @id_distancia INT;
FETCH NEXT FROM cursor_distancia INTO @id, @colesterol_total,
WHILE @@FETCH_STATUS = 0
BEGIN
    -- Calcular a distancia euclidiana entre o registro obtido
    DECLARE
        @distancia_obtida DECIMAL(7, 2) = (SELECT SQRT(POWER(
        IF @distancia_obtida < @distancia
    BEGIN
        SET @distancia = @distancia_obtida;
        SET @id_distancia = @id;
    END;
    -- Avançar para o próximo registro
    FETCH NEXT FROM cursor_distancia INTO @id, @colesterol_to
END;
```

```

PRINT(CONCAT('Guerreiro menos distante ', @id_distancia, ' '),

-- Fechamento
CLOSE cursor_distancia;

-- Deletando o cursor
DEALLOCATE cursor_distancia;

```

4. Escreva uma PROCEDURE que, dado o id de um paciente, atribua a ele a condição do seu paciente mais próximo.

```

CREATE OR ALTER PROCEDURE update_condicao
@id INT
AS
BEGIN
    DECLARE
        @colesterol_total DECIMAL(7, 2),
        @LDL DECIMAL(7, 2),
        @HDL DECIMAL(7, 2);

    -- Traz os dados do guerreiro
    SELECT
        @colesterol_total = colesterol_total,
        @LDL = ldl,
        @HDL = hdl
    FROM
        tb_exames
    WHERE
        id = @id;

    DECLARE cursor_distancia CURSOR
    FOR
    SELECT id, colesterol_total, ldl, hdl FROM tb_exames
    OPEN cursor_distancia;
    DECLARE
        @comp_id INT,
        @comp_colesterol_total DECIMAL(7, 2),
        @comp_LDL DECIMAL(7, 2),

```

```

        @comp_HDL DECIMAL(7, 2),
        @distancia DECIMAL(7, 2) = 1000,
        @id_distancia INT;
    FETCH NEXT FROM cursor_distancia INTO @comp_id, @comp_HDL;
    WHILE @@FETCH_STATUS = 0
    BEGIN
        -- Calcular a distancia euclidiana entre o registro atual e o amigo informado
        DECLARE
            @distancia_obtida DECIMAL(7, 2) = (
                SELECT SQRT(POWER(@comp_cholesterol_total - @comp_HDL, 2) +
                               POWER(@comp_idade - @id_distancia, 2))
                FROM tb_exames
                WHERE id != @id
                AND id = @comp_id
            );
        IF @distancia_obtida < @distancia
        BEGIN
            SET @distancia = @distancia_obtida;
            SET @id_distancia = @comp_id;
        END;

        -- Avançar para o próximo registro
        FETCH NEXT FROM cursor_distancia INTO @comp_id, @comp_HDL;
    END;

    PRINT(CONCAT('Guerreiro menos distante do amigo informado: ', @id_distancia));

    UPDATE
        tb_exames
    SET
        condicao = (SELECT condicao FROM tb_exames WHERE id = @id);

    -- Fechamento
    CLOSE cursor_distancia;

    -- Deletando o cursor
    DEALLOCATE cursor_distancia;

```

```
END;
```

```
SELECT * FROM tb_examenes WHERE id = 1 -- Paciente  
EXEC update_condicao 1; -- Achou o 46 e a distancia era de 25
```

5. Adicione o paciente ('observação', '2024-08-28', 50, 50, 50, ' - ') na tabela e atribua a ele a condição do paciente mais próximo utilizando a PROCEDURE criada no exercício anterior.

```
INSERT INTO  
    tb_examenes  
    (  
        nome,  
        data_exame,  
        colesterol_total,  
        LDL,  
        HDL  
    )  
VALUES  
    (  
        'observação',  
        '2024-08-28',  
        50,  
        50,  
        50  
    );
```

```
EXEC update_condicao 101; -- Achou o 104 e a distancia é de 2
```

6. Exercício 6

a. a) Adicione os seguintes pacientes à tabela:

- (i) ('pac (i)', '2024-08-28', 70, 70, 50, 'Paciente');
- (ii) ('pac (ii)', '2024-08-28', 70, 50, 70, 'Controle');
- (iii) ('pac (iii)', '2024-08-28', 50, 70, 70, 'Controle');


```

INSERT INTO
    tb_exames
        (nome, data_exame, colesterol_total, LDL, HDL)
VALUES
    ('pac (i)', '2024-08-28', 70, 70, 50),
    ('pac (ii)', '2024-08-28', 70, 50, 70),
    ('pac (iii)', '2024-08-28', 50, 70, 50);

```

b. Escreva uma PROCEDURE que, a partir do id de um paciente, apresente quais são os pacientes mais próximos a ele, no caso de haver empate na distância.

```

CREATE OR ALTER PROCEDURE select_proximos
    @id INT
AS
BEGIN
    -- Declaração de variáveis para armazenar os valores d
    DECLARE
        @colesterol_total DECIMAL(7, 2),
        @LDL DECIMAL(7, 2),
        @HDL DECIMAL(7, 2),
        @menor_distancia DECIMAL(7, 2);

    -- Buscar os dados do paciente informado
    SELECT
        @colesterol_total = colesterol_total,
        @LDL = ldl,
        @HDL = hdl
    FROM
        tb_exames
    WHERE
        id = @id;

    -- Calcular a menor distância entre o paciente informa
    SELECT
        @menor_distancia = MIN(SQRT(POWER(colesterol_total
    FROM
        tb_exames

```

```

WHERE
    id != @id; -- Excluir o próprio paciente da comparação

-- Selecionar todos os pacientes que tenham essa menor distância
SELECT
    @menor_distancia
UNION
SELECT
    SQRT(POWER(colesterol_total - @colesterol_total, 2) +
        POWER(70 - @LDL, 2) +
        POWER(70 - @HDL, 2)) AS menor_distancia
FROM
    tb_examenes
WHERE
    id != @id
    AND SQRT(POWER(colesterol_total - @colesterol_total, 2) +
        POWER(70 - @LDL, 2) +
        POWER(70 - @HDL, 2)) < @menor_distancia;
END;

```

c. Quais são os pacientes mais próximos do elemento inserido no Exercício 3?

```
EXEC select_proximos 101;
```

| | |
|---|-------------------------|
| | (Nenhum nome de coluna) |
| 1 | 28,28 |

7. Exercício 7

- a. Adicione o paciente ('observação 2', '2024-08-28', 70, 70, 70, ' - ') na tabela e atribua a ele a condição dos pacientes mais próximos utilizando a PROCEDURE confeccionada no exercício anterior.

```

INSERT INTO
    tb_examenes
    (nome, data_exame, colesterol_total, LDL, HDL)
VALUES

```

```

        ('observação 2', '2024-08-28', 70, 70, 70);

SELECT * FROM tb_exames ORDER BY id DESC -- NULL
EXEC update_condicao 105;                -- Virou controle

```

b. Criar uma trigger que automatiza o processo de condição dos pacientes inseridos

```

CREATE OR ALTER TRIGGER on_insert_tb_exames
ON tb_exames
AFTER INSERT
AS
BEGIN
    DECLARE @id INT;

    -- Pegar o id do registro inserido
    SELECT @id = inserted.id
    FROM inserted;

    -- Chamar a procedure passando o id
    EXEC update_condicao @id;
END;

INSERT INTO
    tb_exames
        (nome, data_exame, colesterol_total, LDL, HDL)
VALUES
    ('observação 3', '2024-08-28', 70, 70, 70);

```

8. Exercício 8

a. Adicione o paciente ('observação 3', '2024-08-28', 140, 70, 70, 'Controle').

```

INSERT INTO
    tb_exames
        (nome, data_exame, colesterol_total, LDL, HDL, con

```

```
VALUES
```

```
('observação 3', '2024-08-28', 140, 70, 70, 'Controle')
```

b. Utilizando cursores, crie duas tabelas: tb_controle e tb_paciente. Estas devem ter a mesma estrutura da tabela tb_exames, exceto pela coluna 'condicao'. Uma tabela será dedicada a armazenar uma cópia dos registros dos indivíduos saudáveis ('Controle') e a outra para os indivíduos afetados ('Paciente').

```
DECLARE cursor_copia CURSOR
FOR
SELECT nome, data_exame, colesterol_total, LDL, HDL, condicao
FROM tb_exames;
OPEN cursor_copia;
DECLARE
    @nome VARCHAR(100),
    @data_exame DATE,
    @colesterol_total DECIMAL(7, 2),
    @LDL DECIMAL(7, 2),
    @HDL DECIMAL(7, 2),
    @condicao VARCHAR(10);
FETCH NEXT FROM cursor_copia INTO @nome, @data_exame, @colesterol_total, @LDL, @HDL, @condicao;
WHILE @@FETCH_STATUS = 0
BEGIN
    IF @condicao = 'Controle'
    BEGIN
        INSERT INTO tb_controle
            (nome, data_exame, colesterol_total, LDL, HDL)
        VALUES
            (@nome, @data_exame, @colesterol_total, @LDL, @HDL);
    END;
    ELSE
    BEGIN
        INSERT INTO tb_paciente
            (nome, data_exame, colesterol_total, LDL, HDL)
        VALUES
            (@nome, @data_exame, @colesterol_total, @LDL, @HDL);
    END;
    FETCH NEXT FROM cursor_copia INTO @nome, @data_exame, @colesterol_total, @LDL, @HDL, @condicao;
END;
```

```

-- Avançar para o próximo registro
FETCH NEXT FROM cursor_copia INTO @nome, @data_exame, (
END;

CLOSE cursor_copia;
DEALLOCATE cursor_copia;

```

9. Exercício 9

- a. Crie uma FUNCTION chamada calc_triglicerideos que, a partir do colesterol total, do HDL e do LDL de um paciente, estime seu triglicérideo usando a fórmula

```

CREATE OR ALTER FUNCTION dbo.func_triglicerideos(@coles
RETURNS DECIMAL(7, 2)
AS
BEGIN
    DECLARE
        @trigliceridio DECIMAL(7, 2) = (5 * (@colester
    RETURN @trigliceridio
END;

```

- b. Nas tabelas criadas no exercício anterior, acrescente a coluna 'razao' que armazena a razão entre colesterol total e triglicérideos dada pela fórmula:

$$\frac{\text{colesterol_total}}{5 * (\text{colesterol_total} - \text{HDL} - \text{LDL})}$$

```

ALTER TABLE tb_exames ADD razao DECIMAL(7, 2);
ALTER TABLE tb_controle ADD razao DECIMAL(7, 2);
ALTER TABLE tb_paciente ADD razao DECIMAL(7, 2);

UPDATE
    tb_exames

```

```
SET
    razao = ISNULL(colesterol_total / NULLIF(dbo.func_t
```

10. Crie uma trigger que garanta que todos os colesterolis estejam no intervalo de 0 a 300

```
CREATE OR ALTER TRIGGER on_insert_paciente_2
ON tb_examenes
INSTEAD OF INSERT
AS
BEGIN
    DECLARE
        @nome VARCHAR(100),
        @data_exame DATE,
        @colesterol_total DECIMAL(7, 2),
        @LDL DECIMAL(7, 2),
        @HDL DECIMAL(7, 2);

    -- Pegar os valores inseridos
    SELECT
        @nome = inserted.nome,
        @data_exame = inserted.data_exame,
        @colesterol_total = inserted.colesterol_total,
        @LDL = inserted.ldl,
        @HDL = inserted.hdl
    FROM
        inserted;

    -- Garante que os colesterolis estejam no intervalo
    INSERT INTO tb_examenes
    (
        nome,
        data_exame,
        colesterol_total,
        LDL,
        HDL
    )
    VALUES
```

```

(
    @nome,
    @data_exame,
    CASE
        WHEN @colesterol_total < 1 THEN 1
        WHEN @colesterol_total > 300 THEN 300
        ELSE @colesterol_total
    END,
    CASE
        WHEN @LDL < 1 THEN 1
        WHEN @LDL > 300 THEN 300
        ELSE @LDL
    END,
    CASE
        WHEN @HDL < 1 THEN 1
        WHEN @HDL > 300 THEN 300
        ELSE @HDL
    END
);

END;

INSERT INTO
    tb_exames
    (nome, data_exame, colesterol_total, LDL, HDL)
VALUES
    ('observação 4', '2024-08-28', 301, 0, 400);

```