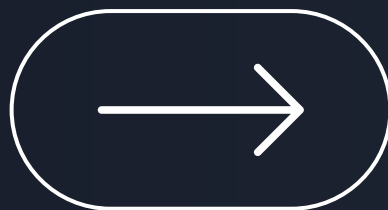


GRUPO DO DRONE
CARLOS, DAVI, GUILHERME E PEDRO

Procedures

(stored procedures)



```
IF EXISTS (SELECT 1 FROM SYS.objects WHERE TYPE = 'P' AND NAME = 'SP_ADD_PACIENTE')
BEGIN
    DROP PROCEDURE SP_ADD_PACIENTE
END
GO

CREATE PROCEDURE SP_ADD_PACIENTE
    @idPaciente INT,
    @CPF VARCHAR(14),
    @email VARCHAR(80),
    @nome VARCHAR(100),
    @dataNasc DATE,
    @sexo BIT,
    @CRM VARCHAR(10),
    @convenio VARCHAR(40) IS NULL
AS
    INSERT INTO Paciente (idPaciente, CPF, email, nome, dataNasc, sexo, CRM, Convenio)
    VALUES (@idPaciente, @CPF, @email, @nome, @dataNasc, @sexo, @CRM, @convenio);

EXEC SP_ADD_PACIENTE
    @idPaciente = 21,
    @CPF = '711.456.123-01',
    @Email = 'grupo9891odrone@gmail.com',
    @Nome = 'Grupo Drone',
    @DataNasc = '2000-12-25',
    @Sexo = 1,
    @CRM = 'CRM12345',
    @Convenio = 'Itau';

SELECT * FROM Paciente
```

PROCEDURES

Explicação.

Uma **procedure** é uma **sub-rotina** de comandos SQL, que é **armazenada** e **executada** no banco de dados.

Uma **sub-rotina** é um **conjunto** sequencial de instruções que pode ser **executado** dentro de um programa

Características:

Importância:

1 Deixar o código mais limpo

3 Melhor Legibilidade e Organização

2 Redução de Código Redundante

4 Reuso e Flexibilidade

Principais Características:

Importância:

Segurança:

A **segurança** em **procedures** no SQL Server é uma das principais **vantagens**, pois permite **controlar** exatamente quem pode acessar ou **modificar** os **dados do banco**.

(continuação)

Síntaxe Básica.

```
IF EXISTS (SELECT 1 FROM SYS.OBJECTS WHERE TYPE = 'P' AND NAME =  
'SP_EXEMPLO_PROCEDURE')
```

```
BEGIN
```

```
    DROP PROCEDURE SP_EXEMPLO_PROCEDURE
```

```
END
```

```
GO
```

```
--STORED PROCEDURE SIMPLES SEM PARAMETROS
```

```
CREATE PROCEDURE SP_EXEMPLO_PROCEDURE
```

```
AS
```

```
    SELECT * FROM [TABELA]
```

```
GO
```

```
--COMANDO PARA EXECUTAR A PROCEDURE
```

```
EXEC SP_EXEMPLO_PROCEDURE
```

Enquanto no MySQL, é só usar o ALTER PROCEDURE

Síntaxe com variável.

```
IF EXISTS (SELECT 1 FROM SYS.OBJECTS WHERE TYPE = 'P' AND NAME =  
'SP_EXEMPLO_PROCEDURE')  
BEGIN  
DROP PROCEDURE SP_EXEMPLO_PROCEDURE  
END  
GO
```

```
--STORED PROCEDURE SIMPLES COM PARAMETROS  
CREATE PROCEDURE SP_EXEMPLO_PROCEDURE  
@ID_CLIENTE INT  
AS  
SELECT * FROM CLIENTES  
WHERE ID_CLIENTE = @ID_CLIENTE  
GO
```

```
--COMANDO PARA EXECUTAR A PROCEDURE  
EXEC SP_EXEMPLO_PROCEDURE 2
```

Exemplos



exemplo:



```
use ClinicaNewSun

IF EXISTS (SELECT 1 FROM SYS.objects WHERE TYPE = 'P' AND NAME = 'SP_DIAGNOSTICO_PACIENTE')
BEGIN
    DROP PROCEDURE SP_DIAGNOSTICO_PACIENTE
END
GO

CREATE PROCEDURE SP_DIAGNOSTICO_PACIENTE
    @idPaciente INT
AS
    SELECT
        paciente.cpf,
        paciente.nome,
        anamnese,
        diagnostico,
        planoTratamento
    FROM Paciente
    inner join Prontuario ON Prontuario.idPaciente = Paciente.idPaciente
    WHERE paciente.idPaciente = @idPaciente
GO

EXEC SP_DIAGNOSTICO_PACIENTE 2
```

Resultado:

	cpf	nome	anamnese	diagnostico	plano Tratamento
1	987.654.321-02	Maria Souza	Visão desfocada e dor ocular	Hipemetropia	Óculos ou lentes de contato corretivas

exemplo:



```
IF EXISTS (SELECT 1 FROM SYS.objects WHERE TYPE = 'P' AND NAME = 'SP_ADD_PACIENTE')
BEGIN
    DROP PROCEDURE SP_ADD_PACIENTE
END
GO

CREATE PROCEDURE SP_ADD_PACIENTE
    @idPaciente INT,
    @CPF VARCHAR(14),
    @email VARCHAR(80),
    @nome VARCHAR(100),
    @dataNasc DATE,
    @sexo BIT,
    @CRM VARCHAR(10),
    @convenio VARCHAR(40)
AS
    INSERT INTO Paciente (idPaciente, CPF, email, nome, dataNasc, sexo, CRM, Convenio)
    VALUES (@idPaciente, @CPF, @email, @nome, @dataNasc, @sexo, @CRM, @convenio);

    SELECT * FROM Paciente
GO

EXEC SP_ADD_PACIENTE
    @idPaciente = 6,
    @CPF = '000.111.222-01',
    @Email = 'procedures@gmail.com',
    @Nome = 'Storage Procedure',
    @DataNasc = '1998-12-25',
    @Sexo = 1,
    @CRM = 'CRM65432',
    @Convenio = 'Bradesco';
```

Antes:

	idPaciente	CPF	email	nome	dataNasc	sexo	CRM	Convenio
1	1	123.456.789-01	joao.silva@example.com	João Silva	1985-03-22	1	CRM12345	Amil
2	2	987.654.321-02	maria.souza@example.com	Maria Souza	1990-07-11	0	CRM54321	SulAmérica
3	3	456.789.123-03	carlos.santos@example.com	Carlos Santos	1978-12-05	1	CRM98765	Unimed
4	4	321.654.987-04	ana.lima@example.com	Ana Lima	2000-02-17	0	CRM12345	NotreDameIntermedica
5	5	654.321.987-05	fernando.alves@example.com	Fernando Alves	1995-05-30	1	CRM98765	NULL

Depois:

	idPaciente	CPF	email	nome	dataNasc	sexo	CRM	Convenio
1	1	123.456.789-01	joao.silva@example.com	João Silva	1985-03-22	1	CRM12345	Amil
2	2	987.654.321-02	maria.souza@example.com	Maria Souza	1990-07-11	0	CRM54321	SulAmérica
3	3	456.789.123-03	carlos.santos@example.com	Carlos Santos	1978-12-05	1	CRM98765	Unimed
4	4	321.654.987-04	ana.lima@example.com	Ana Lima	2000-02-17	0	CRM12345	NotreDameIntermedica
5	5	654.321.987-05	fernando.alves@example.com	Fernando Alves	1995-05-30	1	CRM98765	NULL
6	6	000.111.222-01	procedures@gmail.com	Storage Procedure	1998-12-25	1	CRM65432	Bradesco

exemplo:



```
IF EXISTS (SELECT 1 FROM SYS.objects WHERE TYPE = 'P' AND NAME = 'SP_PESQUISA_MEDICO')
BEGIN
    DROP PROCEDURE SP_PESQUISA_MEDICO
END
GO

CREATE PROCEDURE SP_PESQUISA_MEDICO
    @CRMDoMedico VARCHAR(10)
AS
SELECT
    nome,
    a.CRM,
    STRING_AGG(c.Especialidade, ', ') AS 'Especialidades do Médico',
    email,
    dataNasc,
    CASE
        WHEN sexo = 1 THEN 'Masculino'
        ELSE 'Feminino'
    END AS sexo,
    salarioMensal
FROM Medico AS a
INNER JOIN EspecialidadeMedico AS b
ON a.CRM = b.CRM
INNER JOIN Especialidades AS c
ON b.idEspecialidade = c.idEspecialidade

WHERE a.CRM = @CRMDoMedico
GROUP BY nome, a.CRM, email, dataNasc, sexo, salarioMensal;
GO
```

executando:

```
EXEC SP_PESQUISA_MEDICO @CRMDoMedico = 'CRM12345'
```

Resultado:

	nome	CRM	Especialidades do Médico	email	dataNasc	sexo	salarioMensal
1	Dr. Gustavo da Silva	CRM65432	Oftalmopediatria, Hipemetropia, Miopia	dr.gustavodasilva@gmail.com	1982-06-17	1	10000.00

exemplo:



Antes:

```
IF EXISTS (SELECT 1 FROM SYS.objects WHERE TYPE = 'P' AND NAME = 'SP_DELETAR_LINHAS')
BEGIN
    DROP PROCEDURE SP_DELETAR_LINHAS
END
GO

CREATE PROCEDURE SP_DELETAR_LINHAS
    @nomeTabela VARCHAR(30),
    @coluna VARCHAR(30),
    @idDaColuna INT
AS

    DECLARE @codigoDelete NVARCHAR(MAX)
    SET @codigoDelete = 'DELETE FROM ' + @nomeTabela + ' WHERE ' + @coluna + ' = @idDaColuna'
    EXEC sp_executesql @codigoDelete, N'@idDaColuna INT', @idDaColuna
GO

EXEC SP_DELETAR_LINHAS
    @nomeTabela = 'Paciente',
    @coluna = 'idPaciente',
    @idDaColuna = 6
```

	idPaciente	CPF	email	nome	dataNasc	sexo	CRM	Convenio
1	1	123.456.789-01	joao.silva@example.com	João Silva	1985-03-22	1	CRM12345	Amil
2	2	987.654.321-02	maria.souza@example...	Maria S...	1990-07-11	0	CRM54321	SulAmé...
3	3	456.789.123-03	carlos.santos@exempl...	Carlos S...	1978-12-05	1	CRM98765	Unimed
4	4	321.654.987-04	ana.lima@example.com	Ana Lima	2000-02-17	0	CRM12345	NotreD...
5	5	654.321.987-05	fernando.alves@examp...	Fernand...	1995-05-30	1	CRM98765	NULL
6	6	000.111.222-01	procedures@gmail.com	Storage ...	1998-12-25	1	CRM65432	Bradesco

Depois:

	idPaciente	CPF	email	nome	dataNasc	sexo	CRM	Convenio
1	1	123.456.789-01	joao.silva@example.com	João Silva	1985-03-22	1	CRM12345	Amil
2	2	987.654.321-02	maria.souza@example.com	Maria Souza	1990-07-11	0	CRM54321	SulAmérica
3	3	456.789.123-03	carlos.santos@example.com	Carlos Santos	1978-12-05	1	CRM98765	Unimed
4	4	321.654.987-04	ana.lima@example.com	Ana Lima	2000-02-17	0	CRM12345	NotreDameIntermédica
5	5	654.321.987-05	fernando.alves@example.com	Fernando Alves	1995-05-30	1	CRM98765	NULL

Exemplos Contoso

Exemplo 1.

```
CREATE PROCEDURE AddCustomer
    @CustomerKey NUMERIC,
    @GeographyKey NUMERIC,
    @CustomerLabel NVARCHAR (100),
    @FirstName NVARCHAR (50),
    @MiddleName NVARCHAR (50),
    @LastName NVARCHAR (50),
    @Gender VARCHAR (1)
AS
INSERT INTO DimCustomer
(CustomerKey,GeographyKey,CustomerLabel,FirstName,MiddleName,LastName
,Gender)
VALUES
(@CustomerKey,@GeographyKey,@CustomerLabel,@FirstName,@MiddleName,
@LastName,@Gender)

SET IDENTITY_INSERT DimCustomer ON;
EXEC AddCustomer 19146,736,CS4000,Fernando,Henrique,Cardoso,M
```

Resultado 1.

Antes:

CustomerKey	GeographyKey	CustomerLabel	Title	FirstName	MiddleName
19135	725	CS725	NULL	NULL	NULL
19136	726	CS726	NULL	NULL	NULL
19137	727	CS727	NULL	NULL	NULL
19138	728	CS728	NULL	NULL	NULL
19139	729	CS729	NULL	NULL	NULL
19140	730	CS730	NULL	NULL	NULL
19141	731	CS731	NULL	NULL	NULL
19142	732	CS732	NULL	NULL	NULL
19143	733	CS733	NULL	NULL	NULL
19144	734	CS734	NULL	NULL	NULL
19145	735	CS735	NULL	NULL	NULL

Depois:

CustomerKey	GeographyKey	CustomerLabel	Title	FirstName	MiddleName
19136	726	CS726	NULL	NULL	NULL
19137	727	CS727	NULL	NULL	NULL
19138	728	CS728	NULL	NULL	NULL
19139	729	CS729	NULL	NULL	NULL
19140	730	CS730	NULL	NULL	NULL
19141	731	CS731	NULL	NULL	NULL
19142	732	CS732	NULL	NULL	NULL
19143	733	CS733	NULL	NULL	NULL
19144	734	CS734	NULL	NULL	NULL
19145	735	CS735	NULL	NULL	NULL
19146	736	CS4000	NULL	Fernando	Henrique

Exemplo 2.

```
IF EXISTS (SELECT 1 FROM SYS.OBJECTS WHERE TYPE = 'P' AND NAME =  
'AumentoSalario')  
    BEGIN  
        DROP PROCEDURE AumentoSalario  
    END  
GO
```

```
CREATE PROCEDURE AumentoSalario  
    @percentage DECIMAL (5, 2),  
    @EmployeeKey INT  
AS  
BEGIN  
    UPDATE DimEmployee  
    SET BaseRate = BaseRate * (1 + @percentage / 100)  
    WHERE EmployeeKey = @EmployeeKey  
END;
```

```
EXEC AumentoSalario 20, 1
```

Resultado 2.

Antes:

	ler	PayFrequency	BaseRate
1		2	72.45

Depois:

	ler	PayFrequency	BaseRate
1		2	86.94

Exemplo 3.

```
CREATE PROCEDURE ProcurarProdutos
    @ProductKey INT
AS
BEGIN
    SELECT
        A.SalesKey,
        B.ChannelName,
        C.StoreName,
        D.ProductName,
        A.SalesAmount
    FROM FactSales AS A
        INNER JOIN DimChannel AS B ON A.ChannelKey = B.ChannelKey
        INNER JOIN DimStore AS C ON A.StoreKey = C.StoreKey
        INNER JOIN DimProduct AS D ON A.ProductKey = D.ProductKey
    WHERE D.ProductKey = @ProductKey
    ORDER BY A.SalesAmount DESC;
END;

EXEC ProcurarProdutos 2
```

Fim !!!