|  |  |  |  |
| --- | --- | --- | --- |
| FACULDADE DE TECNOLOGIA SENAC RIO | | | |
| Curso: Análise e Desenvolvimento de Sistemas | | Semestre letivo: 2022.2 | |
| Unidade Curricular: Banco de Dados II | | Módulo: 3 | |
| Professor: Roberto Harkovsky | | Data: | |
| Competências a serem avaliadas:  Administrar Sistemas Gerenciadores de Banco de Dados. | Indicadores de Competência:  Administra Sistemas Gerenciadores de Banco de Dados. | | |
| Aluno: | | | Conceito: |

# Avaliação prática – Construção de Transações

Considere o esquema da base HOSPITAL:

Ambulatorio (numeroAmb, andar, capacidade)

Medicos (codm, CPF, nome, idade, cidade, especialidade, *numeroAmb*)

Pacientes (codp, CPF, nome, sexo, idade, cidade, doença)

Consultas (*codm*, *codp*, data, hora)

Funcionarios (codf, CPF, nome, idade, cidade, salario)

Horário (codm, diasemana)

Pagamentos(especialidade, valorConsulta)

As tabelas MEDICOS, PACIENTES e FUNCIONARIOS descrevem as características destes indivíduos;

CONSULTAS relaciona médicos e pacientes em determinada data para consultas

AMBULATORIO apresenta características do local onde os médicos atendem os pacientes, especialmente a capacidade em leitos, o que se traduz na quantidade máxima de atendimento simultâneo; vários médicos podem atender no mesmo ambulatório, contudo cada medico tem seu ambulatório especifico;

A tabela HORARIO especifica os dias de semana que um médico atende no hospital. Considere os seguintes valores: 2=2ª feira, 3=3ª feira, 4=4ª feira, 5=5ª feira, 6=6ª feira, 7=sábado,1=domingo

PAGAMENTOS especifica o valor da consulta, baseado na especialidade no médico atendente.

Utilize a base *DB\_Medicos v6* disponível no Moodle para esta atividade, e construa as seguintes transações dentro de procedures. Os parâmetros de testes sugeridos ajudam na verificação da funcionalidade:

Grupo 1 e 2: Consultas médicas

1. Inserir uma consulta, passando como parâmetros código do médico, código do paciente, data e hora. A transação deve ser confirmada se não houver erros (marcação duplicada, marcação em horário já utilizado), caso contrário deve ser desfeito.

Nome da transação (procedimento) – T1

Parâmetros de teste –

T1 (49, 1070, “2022/08/01”, “09:00”)

T1 (49, 1030, “2022/08/01”, “09:00”)

T1 (50, 1030, “2022/08/01”, “09:30”)

T1 (49, 1070, “2022/08/01”, “09:00”)

T1 (51, 1030, “2022/08/01”, “09:30”)

CREATE PROCEDURE T1 (@codm int, @codp int, @data VARCHAR(50), @hora time)

AS

BEGIN

BEGIN TRY

IF NOT EXISTS (SELECT \* FROM Consultas WHERE (codm = @codm OR codp = @codp) AND hora = @hora)

BEGIN TRANSACTION

INSERT INTO Consultas(codm, codp, data, hora) VALUES (@codm, @codp, @data, @hora)

PRINT('INSERIDO COM SUCESSO')

COMMIT TRANSACTION

-- SELECT @@IDENTITY FROM Consultas

END TRY

BEGIN CATCH

PRINT('NÃO FOI POSSÍVEL REALIZAR A TRANSAÇÃO')

-- ROLLBACK TRANSACTION

END CATCH

END

EXECUTE T1 51, 1011, "2010-10-04", "21:46:03";

EXECUTE T1 49, 1070, "2022/08/01", "09:00";

EXECUTE T1 49, 1030, "2022/08/01", "09:00";

EXECUTE T1 50, 1030, "2022/08/01", "09:30";

EXECUTE T1 49, 1070, "2022/08/01", "09:00";

EXECUTE T1 51, 1030, "2022/08/01", "09:30";

EXECUTE T1 51, 1030, "2022/08/01", "07:30:00";

SELECT \* FROM Consultas WHERE hora = '21:40:00';

GO

DROP PROCEDURE T1

GO

Grupo 1: Consultas Médicas

1. Inserir uma consulta médica, passando como parâmetros código do médico, código do paciente, data e hora. A transação deve ser confirmada se não houver erros (marcação duplicada, marcação em horário já utilizado), além da restrição de que o ambulatório no qual o médico atende deva ter leito disponível no agendamento proposto. caso contrário deve ser desfeito.

Nome da transação (procedimento) – T2

Parâmetros de teste –

T2 (79, 1030, “2022/08/01”, “09:00”)

T2 (81, 1040, “2022/08/01”, “09:00”)

T2 (97, 1050, “2022/08/01”, “09:00”)

T2 (98, 1060, “2022/08/01”, “09:00”)

T2 (75, 1070, “2022/08/01”, “09:00”)

T2 (76, 1080, “2022/08/01”, “09:00”)

CREATE PROCEDURE T2 (@codm int, @codp int, @data VARCHAR(50), @hora time)

AS

BEGIN

BEGIN TRY

IF NOT EXISTS (

SELECT \* FROM Consultas c WHERE (codm = @codm OR codp = @codp) AND hora = @hora

UNION ALL

SELECT \*, GETDATE() FROM Ambulatorio am WHERE capacidade > 0

)

BEGIN TRANSACTION

INSERT INTO Consultas(codm, codp, data, hora) VALUES (@codm, @codp, @data, @hora)

PRINT('INSERIDO COM SUCESSO')

COMMIT TRANSACTION

END TRY

BEGIN CATCH

PRINT('NÃO FOI POSSÍVEL REALIZAR A TRANSAÇÃO')

-- ROLLBACK TRANSACTION

END CATCH

END

EXECUTE T2 79, 1030, "2022/08/01", "09:00"

EXECUTE T2 81, 1040, "2022/08/01", "09:00"

EXECUTE T2 97, 1050, "2022/08/01", "09:00"

EXECUTE T2 51, 1011, "2016-10-09", "21:40:56"

EXECUTE T2 98, 1060, "2022/08/01", "09:00"

EXECUTE T2 75, 1070, "2022/08/01", "09:00"

EXECUTE T2 76, 1080, "2022/08/01", "09:00"

SELECT \* FROM Consultas;

SELECT \* FROM Ambulatorio;

SELECT \* FROM Horario;

SELECT \* FROM Medicos;

DROP PROCEDURE T2

1. Inserir uma consulta médica, passando como parâmetros código do médico, código do paciente, data e hora. A transação deve ser confirmada se não houver erros (marcação duplicada, marcação em horário já utilizado), além da restrição de que o médico não pode ter mais do que 15 consultas marcadas naquela determinada data. caso contrário deve ser desfeito.

Nome da transação (procedimento) – T3

Parâmetros de teste –

T3 (79, 1030, “2022/08/01”, “09:00”)

T3 (79, 1040, “2022/08/01”, “09:30”)

T3 (79, 1050, “2022/08/01”, “10:30”)

T3 (79, 1050, “2022/08/01”, “10:30”)

T3 (79, 1060, “2022/08/01”, “11:00”)

T3 (79, 1070, “2022/08/01”, “11:30”)

T3 (79, 1080, “2022/08/01”, “11:30”)

CREATE PROCEDURE T3 (@codm int, @codp int, @data VARCHAR(50), @hora time)

AS

BEGIN

BEGIN TRY

IF EXISTS (SELECT \* FROM Consultas c WHERE (codm <> @codm OR codp <> @codp) AND hora <> @hora)

IF EXISTS (SELECT \* FROM Consultas WHERE LEN(codm) = 15 AND data = @data)

BEGIN TRANSACTION

INSERT INTO Consultas(codm, codp, data, hora) VALUES (@codm, @codp, @data, @hora)

PRINT('INSERIDO COM SUCESSO')

COMMIT TRANSACTION

END TRY

BEGIN CATCH

PRINT('NÃO FOI POSSÍVEL REALIZAR A TRANSAÇÃO')

-- ROLLBACK TRANSACTION

END CATCH

END

EXECUTE T2 79, 1030, "2022/04/02", "09:10"

EXECUTE T2 81, 1040, "2022/08/01", "09:00"

EXECUTE T2 97, 1050, "2022/08/01", "09:00"

EXECUTE T2 51, 1011, "2016-10-09", "21:40:56"

EXECUTE T2 98, 1060, "2022/08/01", "09:00"

EXECUTE T2 75, 1070, "2022/08/01", "09:00"

EXECUTE T2 76, 1080, "2022/08/01", "09:00"

SELECT \* FROM Consultas;

SELECT \* FROM Ambulatorio;

SELECT \* FROM Horario;

SELECT \* FROM Medicos;

DROP PROCEDURE T3

Grupo 2

1. Inserir uma consulta médica, passando como parâmetros código do médico, código do paciente, data e hora. A transação deve ser confirmada se não houver erros (marcação duplicada, marcação em horário já utilizado), além da restrição de que o médico deve estar trabalhando naquele dia da semana, conforme a tabela HORARIO. caso contrário deve ser desfeito.

Nome da transação (procedimento) – T4

Parâmetros de teste –

T4 (91, 1030, “2022/08/01”, “09:00”)

T4 (91, 1040, “2022/08/02”, “09:30”)

T4 (90, 1050, “2022/08/01”, “10:30”)

T4 (90, 1060, “2022/08/01”, “11:00”)

T4 (91, 1070, “2022/08/03”, “11:30”)

T4 (91, 1080, “2022/08/04”, “11:30”)

CREATE PROCEDURE T4 (@codm int, @codp int, @data VARCHAR(50), @hora time)

AS

BEGIN

BEGIN TRY

IF EXISTS (SELECT \* FROM Consultas c

INNER JOIN Horario h ON c.codm = h.codm

WHERE (c.codm <> @codm OR c.codp <> @codp) AND c.hora <> @hora

AND h.DiaSemana = SUBSTRING(c.[data],1,2)

)

BEGIN TRANSACTION

INSERT INTO Consultas(codm, codp, data, hora) VALUES (@codm, @codp, @data, @hora)

PRINT('INSERIDO COM SUCESSO')

COMMIT TRANSACTION

END TRY

BEGIN CATCH

PRINT('NÃO FOI POSSÍVEL REALIZAR A TRANSAÇÃO')

-- ROLLBACK TRANSACTION

END CATCH

END

EXECUTE T4 91, 1030, "2022/08/01", "09:00"

EXECUTE T4 91, 1040, "2022/08/02", "09:30"

EXECUTE T4 90, 1050, "2022/08/01", "10:30"

EXECUTE T4 90, 1060, "2022/08/01", "11:00"

EXECUTE T4 91, 1070, "2022/08/03", "11:30"

EXECUTE T4 91, 1080, "2022/08/04", "11:30"

SELECT \* FROM Consultas;

SELECT \* FROM Ambulatorio;

SELECT \* FROM Horario;

SELECT \* FROM Medicos;

DROP PROCEDURE T4

Grupo 1: Cadastrar Novos Médicos a um Ambulatório

1. Associar um ambulatório a um médico recém-contratado, passando como parâmetro o código do médico. A escolha do ambulatório deve ser feita pela transação, baseada na especialidade do médico de acordo com a tabela.

|  |  |
| --- | --- |
| **Especialidade** | **Andar** |
| **Cardiologia** | 2 |
| **Cirurgia** | 5 |
| **Clínica** | 2 |
| **Emergência** | 1 |
| **Ginecologia** | 2 |
| **Oftalmologia** | 3, 4 |
| **Ortopedia** | 3, 4 |
| **Otorrinolaringologia** | 4 |
| **Pediatria** | 2 |
| **Pneumologia** | 3 |
| **Psiquiatria** | 2, 3 |

Nome da transação (procedimento) – T5

Parâmetros de teste –

T5 (30)

T5 (89)

T5 (90)

CREATE PROCEDURE T5 (@codm INT)

AS

BEGIN

BEGIN TRY

IF EXISTS (SELECT a.\*, m.especialidade FROM (SELECT nroa, andar FROM Ambulatorio) AS a

JOIN

(SELECT nroa, especialidade FROM Medicos m) AS m ON a.nroa = m.nroa)

BEGIN TRANSACTION

INSERT INTO Consultas(codm) VALUES (@codm)

PRINT('INSERIDO COM SUCESSO')

COMMIT TRANSACTION

END TRY

BEGIN CATCH

PRINT('NÃO FOI POSSÍVEL REALIZAR A TRANSAÇÃO')

-- ROLLBACK TRANSACTION

END CATCH

END

EXECUTE T5 30

EXECUTE T5 89

EXECUTE T5 90

SELECT \* FROM Consultas;

SELECT \* FROM Ambulatorio;

SELECT \* FROM Horario;

SELECT \* FROM Medicos;

DROP PROCEDURE T5

-- Select p.nome From Pacientes p Where exists

-- (Select \*

-- From Medicos m

-- Where exists

-- (Select \*

-- From Consultas c

-- Where c.codm = m.codm

-- and c.codp = p.codp))

1. Crie uma transação que calcule o valor de uma consulta, a partir do médico (codm) e paciente(codp) que atende aquela consulta médica. Caso o paciente seja funcionário, então aplique um desconto de 20% no valor da consulta

Nome da transação (procedimento) – T7

Parâmetros de teste –

T7 (79, 1030)

T7 (79, 1040)

T7 (79, 1070)

T7 (79, 1080)

CREATE PROCEDURE T6 (@codm INT, @codp INT, @valor INT)

AS

BEGIN

BEGIN TRY

IF EXISTS (SELECT CPF FROM Funcionarios WHERE CPF IN (SELECT CPF FROM Pacientes))

(SELECT (valor \* 0.02) FROM Pagamentos)

BEGIN TRANSACTION

INSERT INTO Consultas(codm, codp) VALUES (@codm, @codp)

INSERT INTO Pagamentos(valor) VALUES (@valor)

PRINT('INSERIDO COM SUCESSO')

COMMIT TRANSACTION

END TRY

BEGIN CATCH

PRINT('NÃO FOI POSSÍVEL REALIZAR A TRANSAÇÃO')

-- ROLLBACK TRANSACTION

END CATCH

END

EXECUTE T6 79, 1030, 20

EXECUTE T6 79, 1040, 20

EXECUTE T6 79, 1070, 20

EXECUTE T6 79, 1080, 20

SELECT \* FROM Consultas;

SELECT \* FROM Ambulatorio;

SELECT \* FROM Horario;

SELECT \* FROM Medicos;

SELECT \* FROM Pacientes;

SELECT \* FROM Funcionarios;

SELECT \* FROM Pagamentos;

DROP PROCEDURE T6

1. Uma nova médica Rachel acaba de chegar ao hospital e precisa poder acessar as consultas médicas. Para isto, crie um login/senha e um usuário na base Médicos acima e forneça acesso de leitura a tabela CONSULTAS

CREATE LOGIN medico WITH PASSWORD = '12345', DEFAULT\_DATABASE=DBMEDICOS , CHECK\_POLICY=OFF;

CREATE USER rachel FOR LOGIN medico;

SELECT \* FROM sys.sql\_logins WHERE name = 'medico';

EXECUTE AS USER = 'rachel';

1. Para melhorar a administração dos usuários médicos do hospital, crie uma ROLE chamada R\_MEDICOS e de acesso de leitura a tabela CONSULTAS. Retire o acesso ao usuário da TRaquel e o adicione esta ROLE.

CREATE ROLE R\_MEDICOS AUTHORIZATION rachel

REVOKE SELECT ON Consultas TO rachel;

REVOKE INSERT ON Consultas TO rachel;

REVOKE UPDATE ON Consultas TO rachel;

REVOKE DELETE ON Consultas TO rachel;

EXEC sp\_addrolemember 'R\_MEDICOS', 'rachel'

REVOKE SELECT ON DATABASE::DBMEDICOS TO R\_MEDICOS;

1. Crie também outra ROLE chamada R\_ADM e forneça acesso total (leitura, atualização, inserção e deleção) a todas as tabelas da base.

CREATE ROLE R\_ADM AUTHORIZATION rachel

GRANT SELECT ON Consultas TO rachel;

GRANT INSERT ON Consultas TO rachel;

GRANT UPDATE ON Consultas TO rachel;

GRANT DELETE ON Consultas TO rachel;

EXEC sp\_addrolemember 'R\_MEDICOS', 'rachel'

GRANT SELECT ON DATABASE::DBMEDICOS TO R\_MEDICOS;

1. Liste os índices existentes da tabela MEDICOS. Supondo uma busca por idade na tabela médico (ex: SELECT nome, especialidade FROM médicos Where idade > 40), sugira um novo índice para melhoria do desempenho desta consulta

-- CREATE CLUSTERED INDEX IDX\_Id ON Medicos (codm);

CREATE NONCLUSTERED INDEX IDX\_Nome ON Medicos (codm)

CREATE NONCLUSTERED INDEX IDX\_Idade ON Medicos (idade)

DROP INDEX IDX\_Idade ON Medicos

SELECT \* FROM Medicos

SELECT nome, especialidade FROM Medicos Where idade > 40