

Aula 6 - Realizando Consultas

Docupedia Export

Author:Goncalves Donathan (SO/OPM-TS21-BR)

Date:17-Jul-2024 13:52

Table of Contents

1	O que é o SELECT	3
2	Excluindo dados e entidades	4
2.1	DELETE - Usado junto com Where para excluir um registro em específico.	4
2.2	TRUNCATE - Exclui todos os registros de uma tabela, mas não a tabela.	4
2.3	DROP - Exclui a entidade e seus dados.	4
3	A falta de um “where”	5
3.1	Como contornar dados possíveis erros de digitação?	5
3.2	Sem esquecer de passar todos os parâmetros necessários.	5
4	TRANSACTION	6
4.1	Mas o que isso quer dizer?	6
4.2	Objetivo	6
4.3	Voltando na Situação Problema da Primeira Aula	8
5	INSERÇÃO DE NOVOS DADOS	10
5.1	Para os próximos exercícios será utilizados dados gerados aleatoriamente.	10
5.2	1. Preciso controlar qual sala está sendo utilizada.	10
5.3	CONVERT para Data e Hora	11
5.4	UPDATE - Atualiza os referidos dados na tabela	12
5.5	ALTER TABLE - Altera as propriedade da entidade como adicionar, excluir, renomear colunas.	13
6	O Funcionamento dos JOINS	14
6.1	Utilizados para agrupar duas ou mais tabelas em uma busca	14
6.1.1	Tabela da esquerda é a que foi declarada primeiro, enquanto a que foi declarada depois, é consideradas a tabela da direita.	14
6.1.2	Tabelas que estão sendo referenciadas não podem ser excluídas.	15
6.2	2. Preciso controlar quem reservou cada sala, pois ele é responsável por deixar tudo no lugar ao sair.	15
6.3	3. Preciso controlar qual dia e por quanto tempo cada sala foi utilizada por cada reunião.	16

1

O que é o SELECT

Consultas (**queries**) realizadas ao banco, retornando os dados solicitados.
Será a base da aula de hoje.

Consulta
<pre>-- "As" é utilizado para criar um "apelido" para a coluna -- Sem o "AS" também funciona SELECT Nome AS Equipamento, Tipo Categoria FROM Equipamento</pre>

	Equipamento	Categoria
1	Caixa de Som JBL	Multimídia

WHERE - Limitar/Restringir os dados, seguido de uma lógica

<pre>-- Recebendo todos que são da categoria "aluno" SELECT * FROM Pessoa WHERE Categoria = 'Aluno' -- Recebendo todas as salas que possuem uma capacidade entre 10 e 20 SELECT * FROM SALA WHERE Capacidade > 10 AND Capacidade < 20</pre>
--

É possível usar vários **operadores lógicos** com o Where.

=	Igual a	>	Maior do que	AND
<> !=	Diferente de	>=	Maior ou igual a	OR
<=	Menor ou igual a	<	Menor do que	NOT

2 Excluindo dados e entidades

2.1 DELETE - Usado junto com Where para excluir um registro em específico.

Obs: Se não houver um Where, todos os dados serão excluídos.

```
DELETE FROM Sala WHERE IDSala = 3;
```

2.2 TRUNCATE - Exclui todos os registros de uma tabela, mas não a tabela.

Obs: Mais rápido que o delete.

```
TRUNCATE TABLE Sala;
```

2.3 DROP - Exclui a entidade e seus dados.

```
DROP TABLE Sala;
```

	DELETE	TRUNCATE	DROP
Tipo	DML	DDL	DDL
Rollback	Sim	Não	Não
Permanente	Não	Sim	Sim
Trigger	Ativada	Não ativa	Não ativa

3 **A falta de um “where”**

Falha em atualização de banco sueco deixa contas de clientes com saldo negativo nesta sexta-feira: diversos usuários compartilharam que suas contas amanheceram todas com os mesmos dois débitos, 724 mil e 514 mil coroas suecas (cerca de 600 mil reais no total). O problema também derrubou o sistema de pagamentos do país. O banco Swedbank afirma que não se trata de um ataque hacker e que irá restaurar os saldos ainda hoje, após uma nova atualização. As informações são do site de notícias sueco SVT Nyheter.

3.1 Como contornar dados possíveis erros de digitação?

3.2 Sem esquecer de passar todos os parâmetros necessários.

4

TRANSACTION

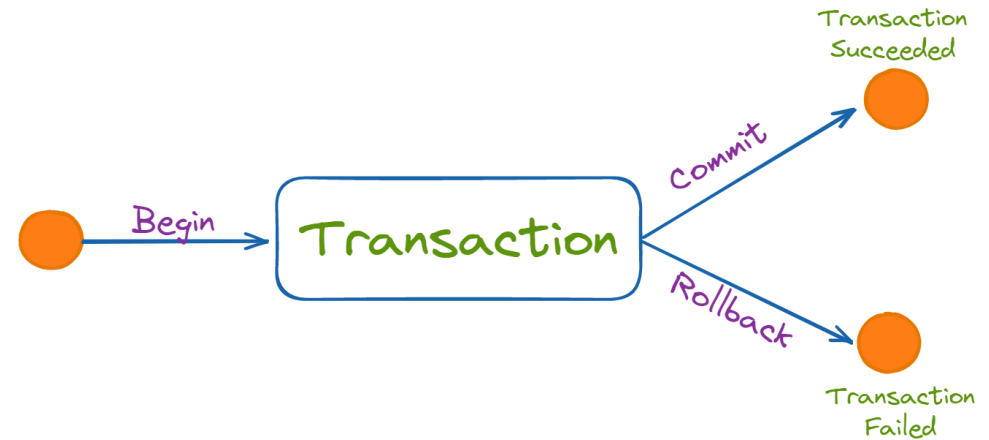
Transação ou Transaction é uma **única unidade de trabalho** processada pelo Sistema de Gerenciamento de Banco de Dados (SGBD).

4.1 Mas o que isso quer dizer?

Significa que uma transação **agrupa** um **conjunto de comandos**, sendo que este conjunto de instruções **só** serão **efetivados** caso **todos sejam executados**.

4.2 Objetivo

Utilizar Transaction **garante** com que haja uma **conformidade** com os **dados reais** para **só então salvá-los** no **BD**.



Transações Implícitas

Cada comando executado já é uma transação isolada de maneira implícita.

Ex: Um comando **INSERT** em uma tabela **só** é realmente **executado**, se todo o comando **estiver correto**, caso algum dos campos ou valores **não esteja** exatamente **como esperado**, **nada é armazenado**.

Transações Explícitas

Desenvolvedor delimita um conjunto de operações, **só** se tornaram **permanente** se todas forem **executadas** com **sucesso**.

Necessário indicar o **começo** e o que será feito ao **final** da **execução** do **código**.

Palavras reservadas que são utilizadas:

BEGIN TRANSACTION – Inicia a Transação.

COMMIT – Efetiva a execução dos comandos.

ROLLBACK – Retorna os dados ao estado anterior.

Podemos fazer de forma manual e verificar se ocorreu tudo da maneira correta.

```
BEGIN TRANSACTION
```

```
INSERT INTO Sala Values ('abcdf1234567', 'Sala de Programação',  
20)
```

```
COMMIT -- Efetivar inserção permanente
```

Podemos executar de maneira automática, garantindo que nenhum comando posterior seja executado.

```
BEGIN TRANSACTION
```

```
DELETE FROM Sala WHERE IDSala = 'abcdf1234567'
```

```
IF @@ERROR <> 0
```

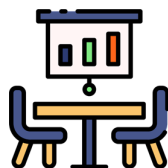
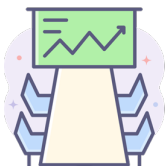
```
    ROLLBACK -- Voltar na situação anterior
```

```
ELSE
```

```
    COMMIT -- Efetivar comando
```

4.3

Voltando na Situação Problema da Primeira Aula



No meu escritório temos 3 salas de reuniões, porém não temos nenhuma forma de **controlar a utilização** destas salas. Sendo assim, gostaria que fosse desenvolvido um **banco de dados**, para na sequência criarmos um aplicativo para facilitar este controle.

Requisitos:

1. Preciso controlar **qual** sala está sendo utilizada;
2. Preciso controlar **quem reservou** cada sala, pois ele é responsável por deixar tudo no lugar ao sair;
3. Preciso controlar **qual dia** e por **quanto tempo** cada sala foi utilizada;
4. Preciso controlar **quem participou** de cada reunião;
5. Preciso controlar se as reuniões estão sendo realizadas nas salas com a **capacidade correta**;
6. Preciso controlar **quais equipamentos** que estão disponíveis em cada sala (projektor, computador, câmera, etc.);

5 INSERÇÃO DE NOVOS DADOS

5.1 Para os próximos exercícios será utilizados dados gerados aleatoriamente.

Acesse e **rode** o arquivo **script** disponibilizado.
Não se esqueça de **alterar** o **caminho** de cada **Json**.

5.2 1. Preciso controlar qual sala está sendo utilizada.

As salas que estão sendo utilizadas estão na entidade Evento

```
SELECT * FROM Evento
```

IDEvento	IDSala	Descricao	DtHrInicio	DtHrFim
1	1	Aula de IoT	2022-03-26T19:00:00Z	2022-03-26T22:30:00Z
2	1	Aula de Banco de Dados	2022-03-27T19:00:00Z	2022-03-27T22:30:00Z
4	1	Aula de Estrutura de Dados	2022-03-28T21:15:00Z	2022-03-26T22:30:00Z

É possível ver as sala usadas, mas **não** é muito **didático**, pois **apenas** foi apresentado o **ID** das salas.

Vamos pedir ao SQL que mostre o **nome** dessas salas.

Left Join

```
SELECT * FROM Evento E
LEFT JOIN Sala S
ON E.IDSala = S.IDSala
```

Evento					Sala		
IDEvento	IDSala	Descricao	DtHrInicio	DtHrFim	IDSala	Nome	Capacidade
1	1	Aula de IoT	2022-03-26T19:00:00Z	2022-03-26T22:30:00Z	1	Sala de Reunião 1	10
2	1	Aula de Banco de Dados	2022-03-27T19:00:00Z	2022-03-27T22:30:00Z	1	Sala de Reunião 1	10
4	1	Aula de Estrutura de Dados	2022-03-28T21:15:00Z	2022-03-26T22:30:00Z	1	Sala de Reunião 1	10

5.3 CONVERT para Data e Hora

Uma **função** que pode ser utilizada para **alterar** campos do tipo **data** ou **hora** para uma **String**

```
SELECT
  CONVERT(VARCHAR(10), GETDATE(), 103) AS Data,
  CONVERT(VARCHAR(5), GETDATE(), 108) AS Hora
```

Agora que você está familiarizado com a função **CONVERT**, crie um **SELECT** que irá retornar os campos abaixo.

IDEvento	Descrição do Evento	Data Início	Hora Início	Data Fim	Hora Fim	Nome da Sala
1	Aula de IoT	26/03/2022	19:00	26/03/2022	22:30	Sala de Reunião 1
2	Aula de Banco de Dados	27/03/2022	19:00	27/03/2022	22:30	Sala de Reunião 1
4	Aula de Estrutura de Dados	28/03/2022	21:15	26/03/2022	22:30	Sala de Reunião 1

Resolução

```
-- Eventos e Salas
SELECT
```

```
E.IDEvento,  
E.Descricao AS "Descrição do Evento",  
CONVERT(VARCHAR(10), E.DtHrInicio, 103) AS "Data Início",  
CONVERT(VARCHAR(5), E.DtHrInicio, 108) AS "Hora Início",  
CONVERT(VARCHAR(10), E.DtHrFim, 103) AS "Data Fim",  
CONVERT(VARCHAR(5), E.DtHrFim, 108) AS "Hora Fim",  
S.Nome AS "Nome da Sala"  
FROM Evento E  
LEFT JOIN Sala S  
ON E.IDSala = S.IDSala
```

Mas ainda é preciso verificar as **salas** que estão sendo utilizadas **nesse momento**.

Adicione ao final do último código

```
WHERE GetDate() BETWEEN DtHrInicio AND DtHrFim
```

5.4 UPDATE - Atualiza os referidos dados na tabela

Nenhuma sala está sendo usada agora, é possível **alterar** algum **registro** já existente.

Update

```
-- Alterar um evento existente  
UPDATE Evento  
SET DtHrInicio = '30/01/2023 19:00', --Coloque a data e a hora de hoje  
    DtHrFim = '30/01/2023 22:30'  
WHERE IDEvento = '414BDC9NB11N'
```

Teste novamente para ver se alguma sala está sendo usada nesse momento.

Encontrar as 3 salas com maior capacidade.

Sala e Equipamento

```
SELECT TOP 3
    Nome 'Nome da Sala',
    Capacidade
FROM Sala
ORDER BY Capacidade DESC
```

5.5 ALTER TABLE - Altera as propriedades da entidade como adicionar, excluir, renomear colunas.

Alter Table

```
-- Adicionando coluna
ALTER TABLE NomeTabela
ADD NomeColuna Tipo;

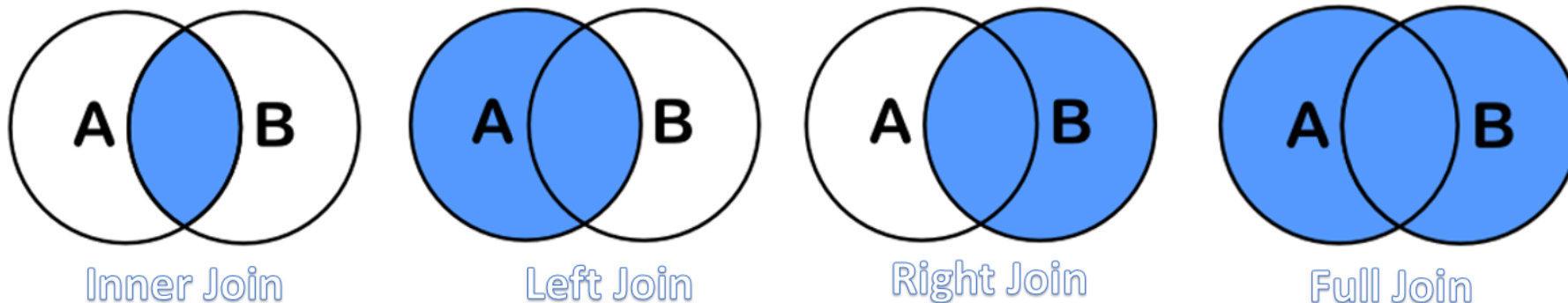
-- Excluindo coluna
ALTER TABLE NomeTabela
DROP COLUMN NomeColuna;

-- Renomeando coluna
ALTER TABLE NomeTabela
RENAME COLUMN AntigoNome to NovoNome;

-- Alterando tipo da coluna
ALTER TABLE NomeTabela
ALTER COLUMN NomeColuna Tipo;
```

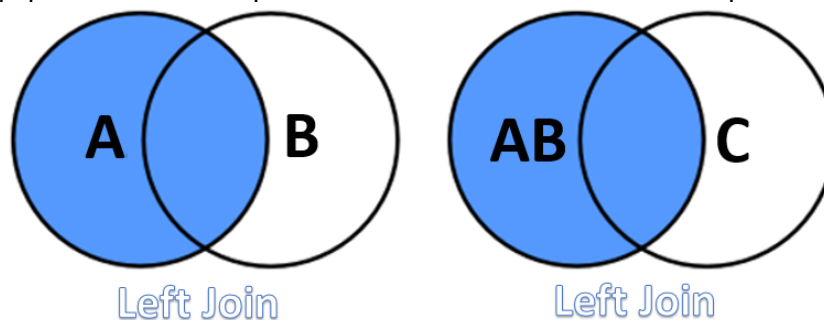
6 O Funcionamento dos JOINS

6.1 Utilizados para agrupar duas ou mais tabelas em uma busca



6.1.1 Tabela da esquerda é a que foi declarada primeiro, enquanto a que foi declarada depois, é considerada a tabela da direita.

É possível fazer a junção de várias tabelas, e o papel de cada tabela pode ser invertido de acordo com essa quantidade de tabelas.



Exemplo

```
SELECT * FROM Tabela1 t1
LEFT JOIN Tabela2 t2
  ON t1.x = t2.x
LEFT JOIN Tabela3 t3
```

```
ON t2.y = t3.y
```

O mais utilizado dos quatro é o Inner Join

6.1.2 Tabelas que estão sendo referenciadas não podem ser excluídas.

Deve-se excluir a tabela que está referenciando com o FK antes de excluir a que está sendo referenciada.

6.2 2. Preciso controlar quem reservou cada sala, pois ele é responsável por deixar tudo no lugar ao sair.

Encontre:

Nome dos participantes responsáveis.

Descrição do Evento.

Data do Evento.

Nome da sala.

Personalize a saída, assim como foi feito anteriormente.

Responsáveis pelas salas

```
SELECT P.Nome          AS "Responsável",
       E.Descricao      AS "Descrição do Evento",
       Convert(Varchar(10), E.DtHrInicio, 103) AS "Data de Início",
       Convert(Varchar(10), E.DtHrInicio, 108) AS "Hora de Início",
       Convert(Varchar(10), E.DtHrFim, 103)   AS "Data de Fim",
       Convert(Varchar(10), E.DtHrFim, 108)   AS "Hora de Fim",
       S.Nome          AS "Nome da Sala"
FROM Evento E

INNER JOIN Sala S
  ON E.IDSala = S.IDSala
INNER JOIN EventoPessoa EP
  ON E.IDEvento = EP.IDEvento
INNER JOIN Pessoa P
  ON EP.IDPessoa = P.IDPessoa

WHERE EP.PapelEvento = 'Responsavel'
```

6.3 3. Preciso controlar qual dia e por quanto tempo cada sala foi utilizada por cada reunião.

Dica: Use **DateDiff**

Tempo usado em cada sala

```
SELECT
    S.Nome          AS "Nome da Sala",
    E.Descricao AS "Descrição do Evento" ,
    DateDiff(MINUTE, E.DtHrInicio, E.DtHrFim) AS "Utilização (minutos)",
    Convert(Varchar(10), E.DtHrInicio, 103) AS "Data de Início",
    Convert(Varchar(10), E.DtHrFim, 103) AS "Data de Fim"

FROM Sala S

INNER JOIN Evento E
    ON S.IDSala = E.IDSala
```