



Visões (Views)

Prof. Fábio Procópio
fabio.procopio@ifrn.edu.br



2

Relembrando...

- Na [aula passada](#), mostramos como gerar e restaurar um *backup full* de um banco de dados;
- Nesta aula, falaremos sobre um elemento comum em SGBD conhecido como visões, que é uma maneira alternativa de visualizar os dados. Vamos nessa?!





Introdução

- Em um sistema de gerenciamento de banco de dados, visões são
 - **instruções SELECT** armazenadas no servidor de BD que, quando executadas, apresentam resultados como se uma tabela tivesse sido consultada
 - podem reunir dados de **várias tabelas** e apresentá-los como se fossem uma única tabela
 - consideradas uma **alternativa** de “observar” os dados de uma ou mais tabelas
 - consideradas por alguns autores como “tabelas virtuais”



Vantagens

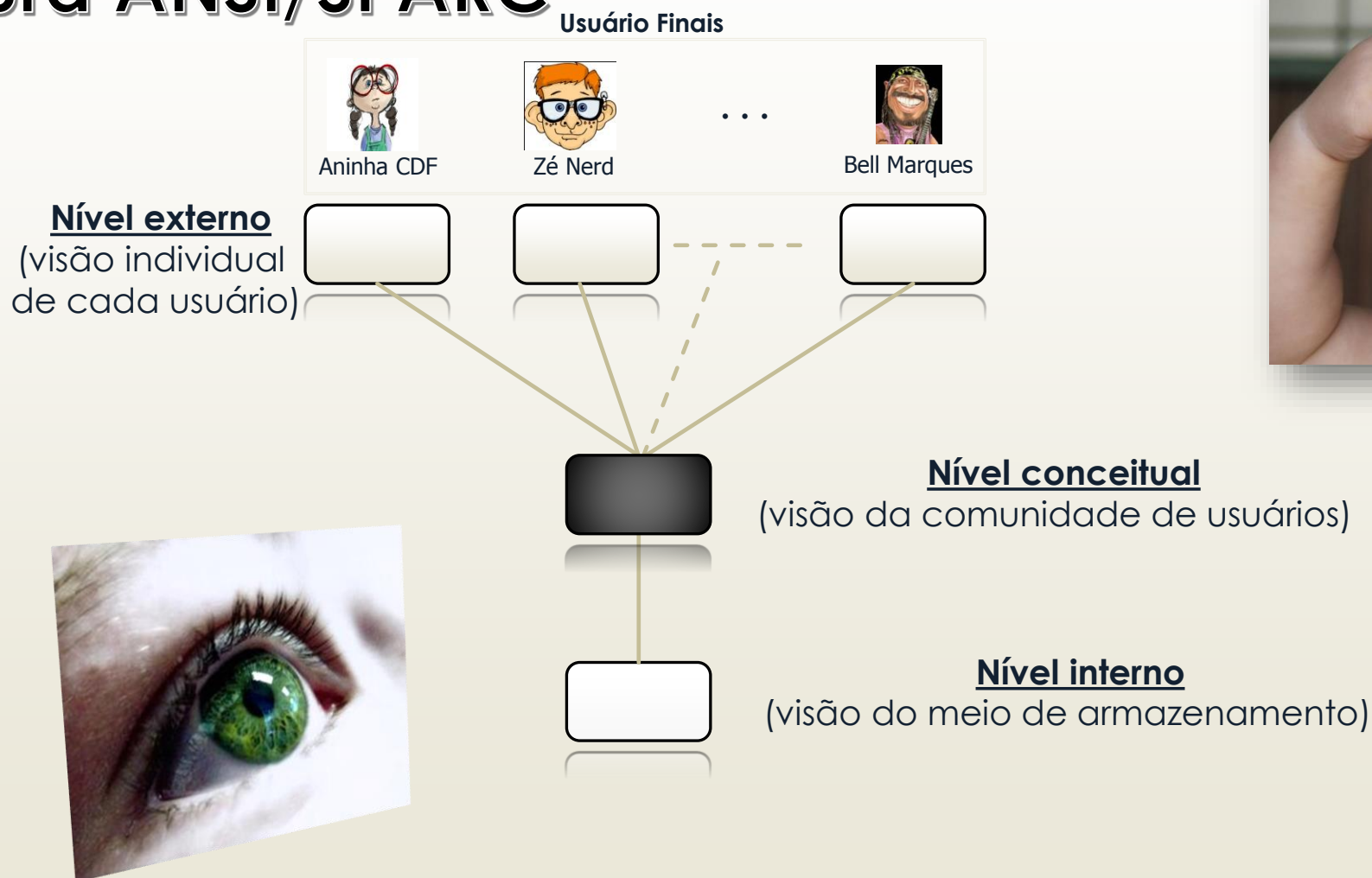
- Menos retrabalho e maior velocidade de acesso aos dados;
- São bastante úteis quando usuários precisam visualizar apenas partes de um banco de dados. Por exemplo:
 - Usuários do setor de atendimento acessam apenas dados de clientes
 - Já os usuários do setor de compras acessam dados de fornecedores
 - Enquanto os usuários de marketing acessam os dados de clientes com poder de compra acima de R\$ 10.000,00
- A utilização de visões permite restringir o acesso a linhas e a colunas específicas de uma ou mais tabelas. Por exemplo:
 - Estagiários do setor de RH podem acessar apenas nome e setor de estagiários
 - Mas, o supervisor pode ver CPF, nome, setor e salário de estagiários funcionários



5

Relembrando...

Arquitetura ANSI/SPARC





Visões atualizáveis

- É possível executar o comando INSERT, UPDATE ou DELETE sobre uma visão, a menos que
 - as colunas referenciadas sejam de apenas uma tabela
 - as colunas não sejam derivadas de AVG, COUNT, SUM, MIN, MAX, GROUPING, STDEV, STDEVP, VAR e VARP
 - não seja o resultado de uma expressão envolvendo outras colunas ou dos operadores UNION, UNION ALL, CROSSJOIN, EXCEPT e INTERSECT
 - não seja usado TOP e nem GROUP BY, HAVING ou DISTINCT



Sintaxes básicas – 1 de 2

- Para criar uma visão, usa-se o comando **CREATE VIEW**, cuja sintaxe básica é:

```
CREATE VIEW <nome_da_visão>  
AS  
  <instrução SELECT>
```

- Para alterar uma visão, usa-se o comando **ALTER VIEW**, cuja sintaxe básica é:

```
ALTER VIEW <nome_da_visão>  
AS  
  <nova instrução SELECT>
```

- Para excluir uma visão, usa-se o comando **DROP VIEW** <nome_da_visão>;
- Para consultar uma visão, a sintaxe é a mesma de uma tabela do BD:

```
SELECT * FROM <nome_da_visão> ORDER BY <campo(s)> [ASC | DESC]
```



Sintaxes básicas – 2 de 2

- Se você colocar a cláusula ORDER BY no corpo de uma visão, é possível que seja gerado o seguinte erro:

A cláusula ORDER BY é inválida em visões, funções embutidas, tabelas derivadas, subconsultas e expressões de tabela comuns, a menos que TOP, OFFSET ou FOR XML também esteja especificado.

- Então, por exemplo, isto não é permitido:

```
CREATE VIEW vwExemplo  
AS  
    SELECT Campo1, Campo2 FROM Tabela ORDER BY Campo1
```

- Mas, isto é:

```
CREATE VIEW vwExemplo  
AS  
    SELECT TOP 2 Campo1, Campo2 FROM Tabela ORDER BY Campo1
```




Visualização de conteúdo

- Para visualizar o código de uma visão, basta executar a *stored procedure* (SP) de sistema **sp_helptext** e enviar, como parâmetro, o nome da visão;
- Veja a sintaxe básica para executar a SP `sp_helptext`:

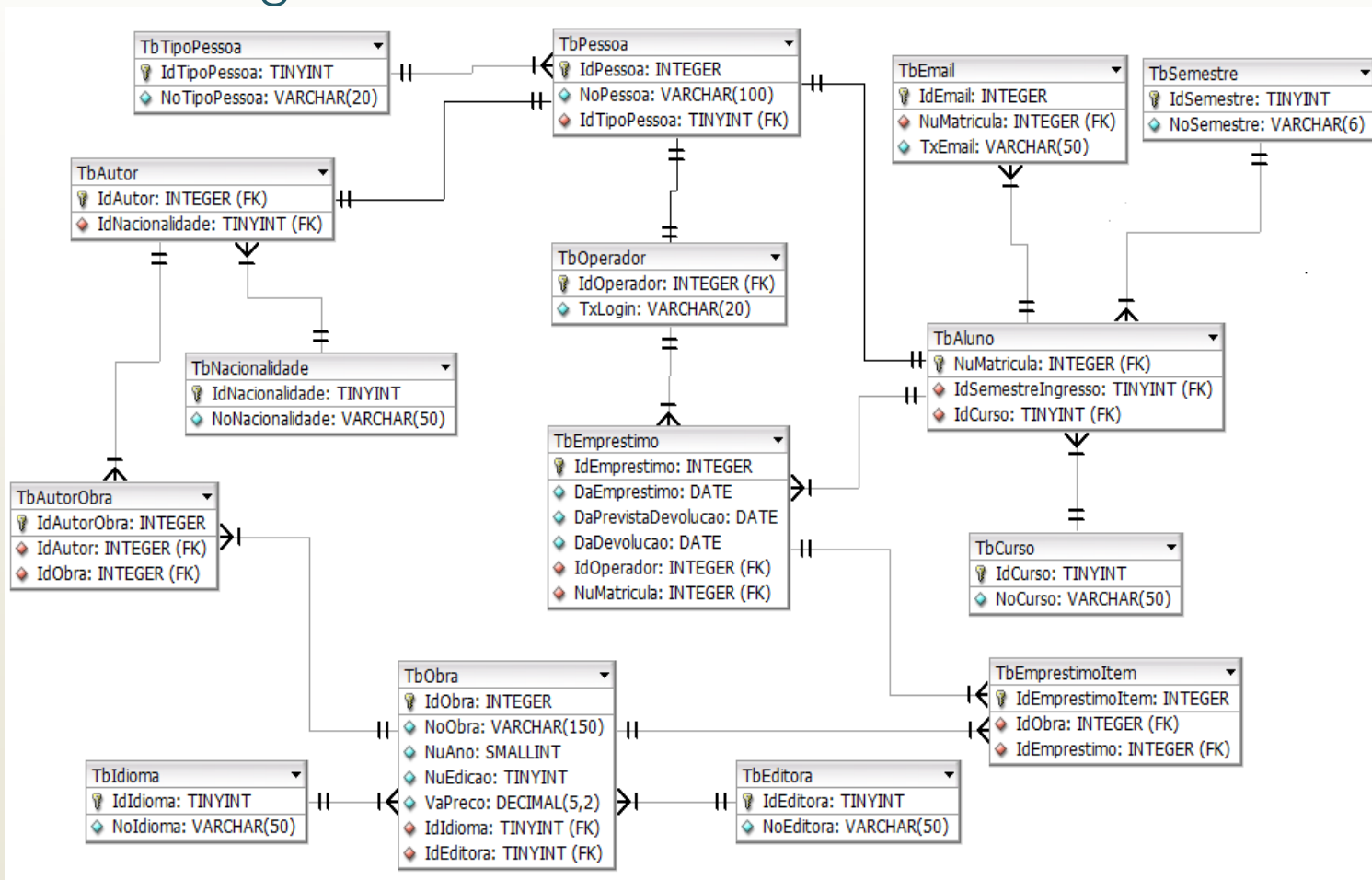
```
EXEC sp_helptext <nome_da_visão>
```

- Muitas vezes, essa SP é combinada com o comando ALTER VIEW da seguinte forma:
 - Executa a SP para obter o comando SQL da visão
 - Altera o código original da visão
 - Executa o comando ALTER VIEW



dbBiblioteca

- Para acompanhar o conteúdo desta aula, [baixe aqui](#) o script do BD dbBiblioteca, cujo modelo de dados segue abaixo:



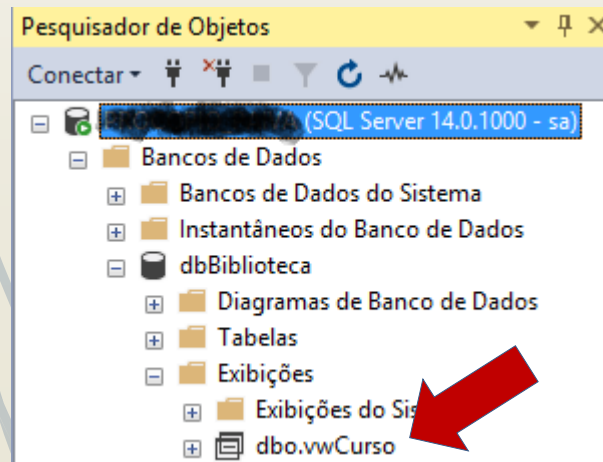


Exemplo 01

- Vamos criar uma visão para retornar as colunas da tabela TbCurso (fique atento(a) porque você deve estar conectado(a) no banco de dados dbBiblioteca)

```
USE dbBiblioteca;  
  
CREATE VIEW vwCurso  
AS  
    SELECT IdCurso, NoCurso  
    FROM TbCurso
```

- Para confirmar a criação da nova visão, conecte no servidor SQL Server usando o SSMS. Em seguida: Banco de Dados → dbBiblioteca → Exibições. Veja:





Exemplo 02

► `vwCurso` é uma visão **atualizável** e, assim, é possível modificar dados. Vamos testar?

```
INSERT INTO vwCurso(IdCurso, NoCurso)
VALUES (5, 'Ensino de Ciências');

/*Para checar a operação, faça uma consulta em TbCurso*/
SELECT * FROM TbCurso;

UPDATE vwCurso
    SET NoCurso = 'Ciências da Natureza'
    WHERE IdCurso = 5;

/*Consulte novamente*/
SELECT * FROM TbCurso;

DELETE FROM vwCurso WHERE IdCurso = 5;

/*Mais uma vez*/
SELECT * FROM TbCurso;
```

Lembra das condições
para uma visão ser
atualizável?

Se não lembrar,
clique [aqui](#).



Exemplo 03

- ➔ Vamos criar uma visão para retornar o nome do aluno, semestre de ingresso e curso.

```
use dbBiblioteca;

CREATE VIEW vwAlunoListagem
AS
    SELECT NoPessoa AS Nome, c.NoCurso AS Curso,
           s.NoSemestre AS Ingresso
    FROM TbPessoa AS p
    INNER JOIN TbAluno AS a
        ON p.IdPessoa = a.NuMatricula
    INNER JOIN TbSemestre AS s
        ON a.IdSemestreIngresso = s.IdSemestre
    INNER JOIN vwCurso AS c
        ON a.IdCurso = c.IdCurso
```

Veja que é possível
realizar JOINS entre
tabelas e
visões.





Exemplo 04

- ➔ Vamos modificar `vwAlunoListagem` para retornar também o e-mail do aluno.

```
use dbBiblioteca;

ALTER VIEW vwAlunoListagem
AS
SELECT NoPessoa AS Nome, c.NoCurso AS Curso,
       s.NoSemestre AS Ingresso, e.TxEmail AS Email
FROM TbPessoa AS p
INNER JOIN TbAluno AS a
    ON p.IdPessoa = a.NuMatricula
INNER JOIN TbSemestre AS s
    ON a.IdSemestreIngresso = s.IdSemestre
INNER JOIN vwCurso AS c
    ON a.IdCurso = c.IdCurso
INNER JOIN TbEmail AS e
    ON a.NuMatricula = e.NuMatricula
```

Neste exemplo, usei
`sp_helptext` para
obter o script da
visão.





15

Exemplo 05

- Agora, vamos testar a visão, ordenando o resultado pelo nome do aluno e pelo curso.

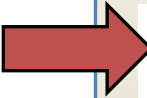
```
use dbBiblioteca;  
  
SELECT *  
  FROM vwAlunoListagem  
 ORDER BY Nome, Curso
```

- Por fim, vamos destruir a visão vwAlunoListagem.

```
DROP VIEW vwAlunoListagem;
```

- Agora, tente consultar a visão recém-destruída.

```
SELECT * FROM vwAlunoListagem;
```



Mensagem 208, Nível 16, Estado 1, Linha 16
Nome de objeto 'vwAlunoListagem' inválido.



Referências

1. MICROSOFT. **CREATE VIEW (Transact-SQL)**. Disponível em: <https://docs.microsoft.com/pt-br/sql/t-sql/statements/create-view-transact-sql?view=sql-server-2017>. Acessado em: 23 jul 2019.
2. DEVMEDIA. **Conceitos e Criação da View – Views no SQL Server – Parte 1**. Disponível em: <http://www.devmedia.com.br/conceitos-e-criacao-da-view-views-no-sql-server-parte-1/22390>. Acessado em: 23 jul 2019.