

Conteúdo 2

Site: [AVA - Acadêmico do ZL/IFRN](#)
Curso: Programação em Bancos de Dados
Livro: Conteúdo 2

Impresso por: Hagliberto Oliveira
Data: sábado, 10 ago. 2024, 13:15



Índice

1. Introdução
2. Preparação do ambiente
3. Vantagens
4. Criação de views
5. Consulta a views
6. Alteração de views
7. Exclusão de views
8. Referências



1. Introdução

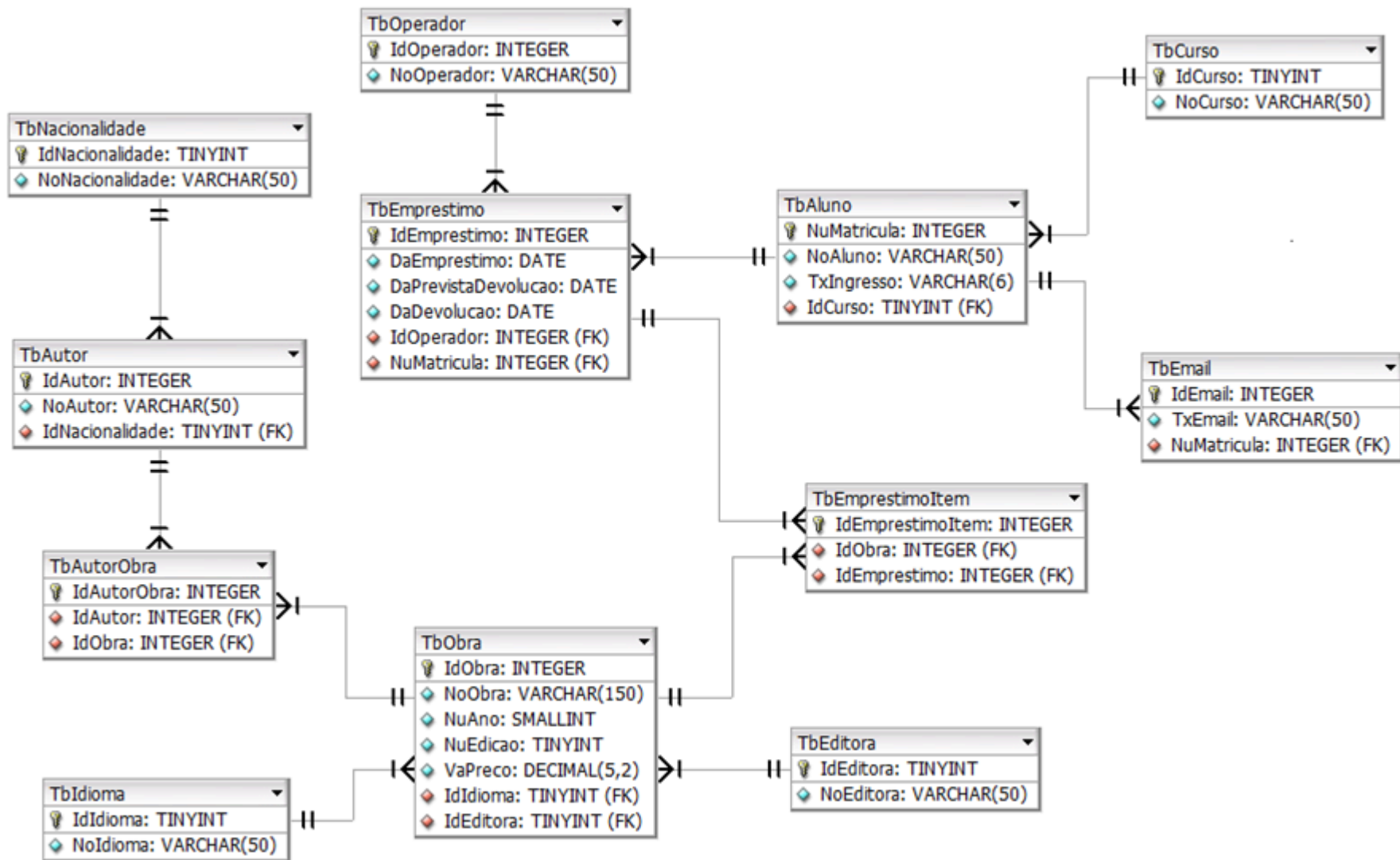
No contexto de sistemas de gerenciamento de banco de dados, uma *view* (ou visão) é uma tabela virtual que consiste no resultado produzido pela execução de uma consulta SQL. O corpo da *query* pode fazer referência a uma tabela (ou a outra *view*) ou a um conjunto de tabelas relacionadas (ou a outras *views*).

Assim como as tabelas de um banco de dados, as *views* possuem linhas e colunas. Por esse motivo, podemos utilizar palavras-chave da linguagem SQL como JOINS, WHERE e diversos tipos de funções. Dependendo de sua estrutura, também podem ser utilizados comandos como INSERT, UPDATE e DELETE. Mas, isso veremos depois.

2. Preparação do ambiente

Antes de começarmos a implementar nossas primeiras visões, vamos baixar um script que é responsável por criar e por popular um banco de dados, o qual chamamos de dbBiblioteca. Para isso, [clique aqui](#).

O modelo de dados lógico de dbBiblioteca segue abaixo:



3. Vantagens

Uma das vantagens em se utilizar visões é o menor retrabalho na escrita de uma consulta SQL que envolve muitas tabelas. Por exemplo, considere uma *query* em que são extraídas informações de 10 (dez) tabelas de um banco de dados. Para escrever essa consulta, seria necessário a utilização de vários JOINS tornando o processo de construção da *query* um trabalho enfadonho.

Outra vantagem é a possibilidade de disponibilizar para os usuários e aplicações apenas as informações que são necessárias para um determinado perfil, isto é, apenas partes do banco de dados. Como exemplo, pensemos em uma pequena empresa em que existem três setores: atendimento ao cliente, compras e vendas. Nesse caso, seria possível criar uma visão para retornar informações sobre os clientes (para o setor de atendimento ao cliente), outra para retornar informações sobre fornecedores e suas promoções (para o setor de compras) e, por fim, uma visão para retornar informações sobre produtos e estoque (para o setor de vendas).

Uma *view* também pode restringir o acesso a linhas e/ou colunas de uma ou mais tabelas. Por exemplo, imagine usuários do setor de gestão de pessoas cujos perfis são “funcionários” e “supervisores”. Nesse caso, o perfil “funcionários” teria acesso apenas para visualizar a matrícula, o nome, a data de admissão e o e-mail dos funcionários da empresa. Por outro lado, as permissões de acesso concedidas a usuários do perfil “supervisores” abrangem, além do perfil “funcionários”, informações sensíveis como CPF, salário base e gratificações.

4. Criação de views

A sintaxe genérica para criação de uma *view* é apresentada a seguir:

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

Agora, retomemos ao modelo lógico do banco de dados dbBiblioteca. Observe que há 02 (dois) campos na tabela TbCurso: IdCurso e NoCurso. Com o script abaixo, podemos criar uma visão para retornar os campos da tabela:

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

Para criar a view, no ambiente do PHPMyAdmin, clique no banco de dados dbBiblioteca. Na sequência, clique na aba SQL. Em seguida, digite o script e pressione o botão “Continuar”.

Servidor: 127.0.0.1 » Banco de dados: dbbiblioteca

Estrutura SQL Pesquisar Pesquisa por formulário Exportar Importar Operações Privilégios Mais

Executa comando(s) SQL na base de dados dbbiblioteca:

```
1 CREATE VIEW vwCurso
2 AS
3 SELECT IdCurso, NoCurso
4 FROM TbCurso;
```

Limpar Formato Obter consulta auto-salva

☐ Unir parâmetros

Marcar esta query SQL:

Delimitador ; ☒ Mostrar de novo aqui este comando ☐ Reter a caixa da consulta (query) ☐ Rollback quando terminado

☒ Ativa a verificação de chaves estrangeiras **Continuar**

Se você seguir o passo a passo conforme orientação, deverá aparecer a seguinte mensagem:

Servidor: 127.0.0.1 » Banco de dados: dbbiblioteca

Estrutura SQL Pesquisar Pesquisa por formulário Exportar Importar Operações Privilégios Mais

Mostrar Caixa do query

✓ MySQL não retornou nenhum registro. (A consulta demorou 0,0006 segundos.)

CREATE VIEW vwCurso AS SELECT IdCurso, NoCurso FROM TbCurso;

[Editar em linha] [Editar] [Criar código PHP]

5. Consulta a views

Como uma visão é uma tabela virtual, então o comando SELECT também é aplicado a ela. Mais uma vez, no ambiente do PHPMyAdmin, clique no banco de dados dbBiblioteca. Na sequência, clique na aba SQL. Em seguida, digite o script abaixo e pressione o botão “Continuar” para exibição do resultado da consulta, conforme figura abaixo.

Servidor: 127.0.0.1 » Banco de dados: dbbiblioteca » Ver: vwCurso

[Procurar](#)
[Estrutura](#)
[SQL](#)
[Pesquisar](#)
[Inserir](#)
[Exportar](#)
[Privilégios](#)
[Operações](#)
[Rastreado](#)

Mostrar Caixa do query

⚠ A seleção atual não contém uma coluna exclusiva. As funções edição em grade, editar, copiar e apagar podem apresentar resultados indesejados.

✓ A mostrar registos de 0 - 3 (4 total, A consulta demorou 0,0006 segundos.)

`SELECT * FROM vwCurso;`

☐ Perfil
 [\[Editar em linha \]](#)
[\[Editar \]](#)
[\[Explicar SQL \]](#)
[\[Criar código PHP \]](#)
[\[Actualizar \]](#)

☐ Mostrar tudo |
 Número de registos:
 Filtrar registos:

Opções extra

	IdCurso	NoCurso
<input type="checkbox"/> ✎ Editar 📋 Copiar 🗑 Apagar	1	Informática
<input type="checkbox"/> ✎ Editar 📋 Copiar 🗑 Apagar	2	Sistemas para Internet
<input type="checkbox"/> ✎ Editar 📋 Copiar 🗑 Apagar	3	Mecatrônica
<input type="checkbox"/> ✎ Editar 📋 Copiar 🗑 Apagar	4	Redes de Computadores

6. Alteração de views

A sintaxe genérica para de alteração de uma *view* é apresentada a seguir:

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

Agora, vamos alterar o conteúdo da visão vwCurso para que a listagem seja ordenada pelo nome do curso. O script abaixo modifica a visão ordenando o resultado de acordo com a coluna informada:

```
ALTER VIEW vwCurso  
AS  
    SELECT IdCurso, NoCurso  
        FROM TbCurso  
    ORDER BY NoCurso;
```

7. Exclusão de views

Por fim, se você deseja excluir uma view que criou, basta utilizar a seguinte sintaxe:

```
DROP VIEW <nome-da-visão>
```

Caso queira testar, você pode excluir uma view qualquer e depois excluí-la. Imagine que você criou uma view chamada vwLixo. Para excluir basta executar o comando:

```
DROP VIEW vwLixo
```

8. Referências

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: 20 mar 2024.

DUCEA. **Dumping MySQL Stored Procedures, Functions and Triggers**. Disponível em: <https://www.ducea.com/2007/07/25/dumping-mysql-stored-procedures-functions-and-triggers/>. Acessado em: 20 mar 2024.

MYSQL. **CREATE VIEW Statement**. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/create-view.html>. Acessado em: 20 mar 2024.

MYSQL. **Dumping Data in SQL Format with mysqldump**. Disponível em: <https://dev.mysql.com/doc/mysql-backup-excerpt/5.7/en/mysqldump-sql-format.html>. Acessado em: 20 mar 2024.

PROCOPIO NA REDE. **Programando no MySQL - Módulo 02: Visões**. YouTube, 28 de fevereiro de 2023. 15min30s. Disponível em: <https://www.youtube.com/playlist?list=PLqsF5rntN2na2N6pmJzFPnznWvpFHEhCn>. Acessado em: 20 mar 2024.