

ETEC – Cidade do Livro

Banco de Dados II

Aula 1



Prof. Me. Marco Aurelio M. Antunes

Marco Aurelio Migliorini Antunes



Formação:

Curso Técnico: Técnico em Contabilidade

Graduação: Análise de Sistemas – USC

Graduação: Matemática - FAEL

Pós Graduação: Recursos Humanos com Sistemas de Informação – FIB

Mestrado: Tv Digital – FAAC – UNESP

Atuação Profissional:

- Lógica de Programação
- Banco de Dados
- Desenvolvimento de Sistemas
- Desenvolvimento Web / Mobile
- Pós Graduação – FIB
- Professor/Coordenador Curso Técnico em Informática – ABFA/INSSC
- Professor/Coordenador Atividade Complementar – FIB
- Professor Curso Técnico - ETEC

Contato:

- prof_antunes@outlook.com
- <https://www.linkedin.com/in/marco-aurelio-m-antunes-85826027/>
- <http://lattes.cnpq.br/0148514838383451>





BANCO DE DADOS

Visões

O que são Visões?

A view pode ser definida como uma tabela virtual composta por linhas e colunas de dados vindos de tabelas relacionadas em uma query (um agrupamento de SELECT's, por exemplo). As linhas e colunas da view são geradas dinamicamente no momento em que é feita uma referência a ela.

Como já dito, a query que determina uma view pode vir de uma ou mais tabelas, ou até mesmo de outras views.

Ao criarmos uma view, podemos filtrar o conteúdo de uma tabela a ser exibida, já que a função da view é exatamente essa: filtrar tabelas, servindo para agrupá-las, protegendo certas colunas e simplificando o código de programação.

É importante salientar que, mesmo após o servidor do SQL Server ser desligado, a view continua “**viva**” no sistema, assim como as tabelas que criamos normalmente. As views não ocupam espaço no banco de dados.

Por que utilizar Visões?

- Restringir acesso a dados, utilizando o comando SELECT, podemos filtrar linhas e colunas que não devam ser mostrados a todos os usuários.
- Buscas complexas tornam-se simples: usuários não precisam realizar complexos comandos SELECTs para localizar as informações de que eles necessitam, assim quando precisarem da informação, farão a busca na visão.
- Independência de dados.
- Dados mostrados de maneira diferente (customizada).

As visões podem se classificadas por: **Simples**

- Os dados são extraídos de uma única tabela
- Não contém funções
- Não possuem dados agrupados

Complexas

- Os dados são extraídos de várias tabelas
- Podem conter funções
- Podem conter dados agrupados



Vantagens das Views

- Reuso:** as views são objetos de caráter permanente. Pensando pelo lado produtivo isso é excelente, já que elas podem ser lidas por vários usuários simultaneamente.
- Segurança:** as views permitem que ocultemos determinadas colunas de uma tabela. Para isso, basta criarmos uma view com as colunas que achamos necessário que sejam exibidas e as disponibilizarmos para o usuário.
- Simplificação do código:** as views nos permitem criar um código de programação muito mais limpo, na medida em que podem conter um SELECT complexo, é uma forma de aumentar a produtividade da equipe de desenvolvimento.



Desvantagens das Views

- Esconde uma complexidade da *query* podendo enganar o desenvolvedor quanto à performance necessária para acessar determinada informação. E pode ser pior quando *views* usam outras *views*. Em alguns casos você pode estar fazendo consultas desnecessárias sem saber de forma muito intensiva.
- Cria uma camada extra. mais objetos para administrar. Algumas pessoas consideram isto um aumento de complexidade. Uma outra forma de ver isto é que uma *view* pode ser mal usada.
- Pode limitar exageradamente o que o usuário pode acessar impedindo certas tarefas.



Criando uma View

Id	Nome	Fabricante	Quantidade	VUnitario	Tipo
1	Playstation 3	Sony	100.00	1999.00	Console
2	Core 2 Duo 4GB Ram 500GB HD	Dell	200.00	1899.00	Notebook
3	Xbox 360 120GB	Microsoft	350.00	1299.00	Console
4	GT-I6220 Quad Band	Samsung	300.00	499.00	Celular
5	iPhone 4 32GB	Apple	50.00	1499.00	Smartphone
6	Playstation 2	Sony	100.00	399.00	Console
7	Sofá Estofado	Coréia	200.00	499.00	Sofá
8	Armário de Serviço	Aracaju	50.00	129.00	Armário
9	Refrigerador 420L	CCE	200.00	1499.00	Refrigerador
10	Wii 120GB	Nintendo	250.00	999.00	Console

```
create table produtos(  
  idprod      int      not null,  
  nome        varchar(45) not null,  
  fabricante  varchar(45) not null,  
  quantidade  decimal(6,2) not null,  
  vlunitario  decimal(7,2) not null,  
  tipo        varchar(30) not null  
);
```

Criando uma View

```
create view vwprodutos
as
    select    id          as Código,
             nome         as Produto,
             fabricante,
             quantidade,
             vlunitario   as Valor Unitário,
             tipo
    from
        produtos;
```

Para consultarmos os dados na view usamos o comando [SELECT](#), da mesma forma que se estivéssemos fazendo uma consulta em uma tabela comum.

```
select * from vwprodutos;
```

Criando uma View

```
create view vwprodutos2
as
select    idprod      'Código do Produto',
          nome        'Nome do Produto',
          fabricante   'Fabricante',
          quantidade   'Saldo em Estoque',
          vlunitario   'Valor Unitário',
          tipo         'Tipo do produto'
from
          produtos;
```

```
select * from vwprodutos2;
```

Alterando uma View

```
alter view vwprodutos
as
    select  id          as código,
            nome        as produto,
            vlunitario  as [valor unitário]
    from
        produtos
    where
        vlunitario > 499.00
```

O comando **ALTER VIEW** é utilizado para atualizar uma view, após ela já ter sido criada e necessitar de alterações.

```
select * from vwprodutos;
```

Excluindo uma View

Para excluirmos uma view é bem simples: é só usar o comando **DROP VIEW**.

A exclusão de uma view implica na exclusão de todas as permissões que tenham sido dadas sobre ela. Dito isso, devemos usar o comando DROP VIEW apenas quando desejamos de fato retirar a view do sistema.

```
drop view vwprodutos;
```



Exercícios

- 1 – Criar 3 views com exemplos de união de tabelas – esses exemplos devem ter pelo menos 4 tabelas envolvidas.
- 2 – Criar 3 views com exemplos de agrupamento.
- 3 – Criar 3 view com funções AVG, COUNT e SUM
- 4 – Criar 3 views utilizando DISTINCT, BETWEEN, LIKE, IN e WHERE