

Curso de Especialização:  
Engenharia e Administração de Sistemas de Banco de Dados

# Fundamentos de Sistemas de Banco de Dados



## Transact SQL – Visões (*Views*)

Profa. Dra. Gisele Busichia Baioco

[gisele@ft.unicamp.br](mailto:gisele@ft.unicamp.br)



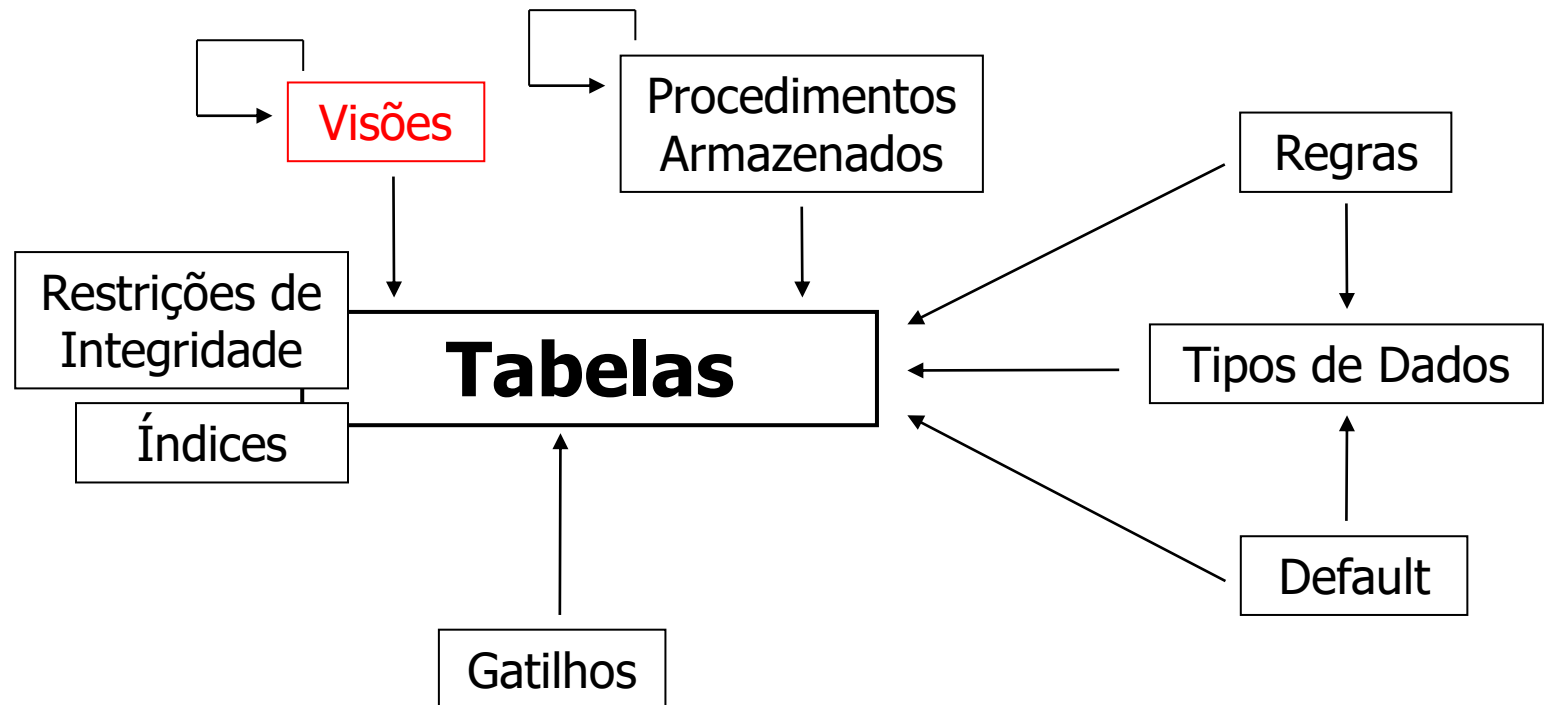
UNICAMP



FACULDADE DE TECNOLOGIA

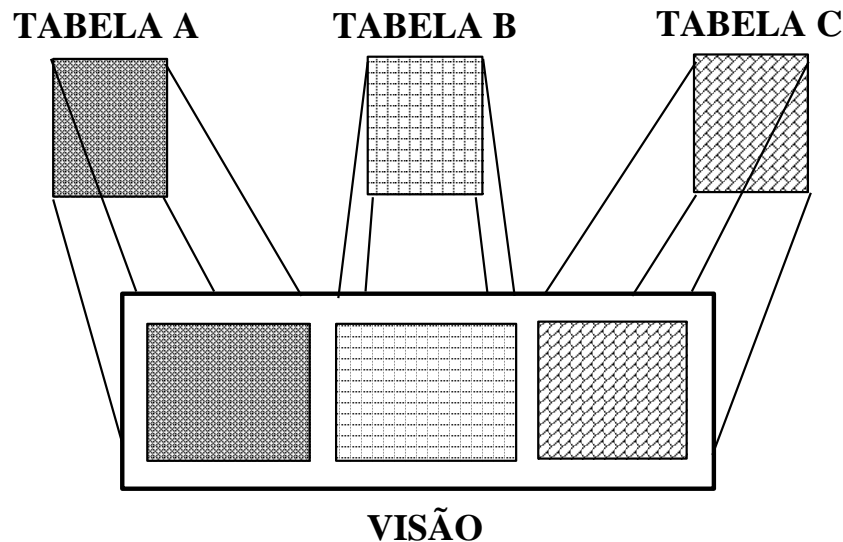
# Visões – *Views*

## Banco de Dados



# Visões – *Views*

- **Definição:** uma visão (*view*) é um recurso disponível nos SGBDs relacionais que possibilita a observação de dados de uma ou mais tabelas ou de outras visões.
  - Pode-se imaginar uma visão como uma moldura pela qual vêem-se apenas dados de interesse.





# Visões – *Views*

## Vantagens

---

- Uma visão pode restringir o acesso a colunas de tabelas (para leitura ou para alteração), o que é útil no caso de controle de acesso (segurança);
- Uma consulta complexa que é usada frequentemente pode ser criada como visão e, a cada vez que ela é necessária, basta selecionar dados da visão;
- Visões podem conter valores calculados ou valores de resumo, o que simplifica a operação.



# Visões – *Views*

## Desvantagens

---

- Desempenho: a questão de desempenho no acesso aos dados através de uma visão está ligada ao desempenho da execução do comando *select* utilizado em sua criação.
  - Se a visão for resultado de junção, mesmo que uma consulta acesse dados contidos em apenas uma das tabelas que compõe a visão, a junção é sempre executada.
- Alterações de dados (*insert*, *update* e *delete*) através de visões têm restrições.



# Visões – *Views*

## Como criar uma visão usando T-SQL?

---

- Comando CREATE VIEW;
- Sintaxe básica:

```
create view nome_visão [(coluna [,...n])]  
as  
comando_select
```

- Deve-se escolher a base de dados onde se deseja criar a visão;
- Após sua criação, uma visão pode ser tratada de forma similar a qualquer tabela da BD. A impressão que se tem ao acessar os dados de uma visão é a de estar acessando os dados de uma tabela;
- **Não há replicação dos dados das tabelas que compõem uma visão – o SGDB armazena apenas a definição da visão.**
  - Os dados acessados através de uma visão são os mesmos que estão armazenados nas tabelas que a compõem, implicando que qualquer modificação feita nos dados diretamente nas tabelas reflita na visão e vice-versa.



# Visões – *Views*

## Como criar uma visão usando T-SQL?

■ Exemplo:

Esquema físico de dados	Visão
<pre>create table produto (   codproduto int not null,   nome varchar(60) not null,   preco numeric(5,2) not null,   primary key (codproduto) ) go create table fornecedor (   codfornecedor int not null,   nome varchar(60) not null,   endereco varchar(70) not null,   primary key (codfornecedor) ) go create table produtofornecedor (   codproduto int not null,   codfornecedor int not null,   primary key (codproduto, codfornecedor),   foreign key (codproduto) references produto,   foreign key (codfornecedor) references fornecedor ) go</pre>	<pre>create view visaofofornecedor as select f.nome nomefornecedor, p.nome nomeproduto from fornecedor f inner join produtofornecedor pf   on f.codfornecedor = pf.codfornecedor   inner join produto p   on pf.codproduto = p.codproduto</pre>



# Visões – *Views*

## Como usar uma visão?

---

- Consulta a dados de visões – é exatamente igual à feita em tabelas:

```
select * from visaofofneoproduto  
select nomefofneedor from visaofofneoproduto
```

- Alteração de dados através de visões – deve obedecer a alguns critérios:

- Uma visão é atualizável se as seguintes restrições forem obedecidas no comando *select* que a origina:

- não pode ter funções de agregação na lista de seleção;
- não pode ter colunas derivadas de outras colunas (por exemplo, pelos operadores de adição ou subtração) na lista de seleção;
- a cláusula FROM deve referenciar pelo menos uma tabela. Por exemplo, a seguinte visão não pode ser alterada:

```
create view semtabela  
as  
select getdate() as DataAtual, @@language as LinguagemAtual
```

- Os comandos de alteração de dados através de visões atualizáveis devem obedecer a algumas restrições:

- *update* e *insert* devem alterar dados de apenas uma tabela referenciada na cláusula FROM do comando *select* que originou a visão;
- a inserção de dados (*insert*) deve respeitar todas as colunas de todas as tabelas que compõem a visão, mesmo que essas colunas não façam parte dela. A solução seria incluir na visão todas as colunas com opção *not null*, mesmo que algumas não sejam necessárias;
- a remoção de registros através de visões (*delete*) não é permitida no caso da visão ser composta por mais de uma tabela.





# Visões – *Views*

## Como alterar uma visão?

---

- Comando ALTER VIEW;
- Sintaxe básica:

```
alter view nome_visão [(coluna [,...n])]  
as  
comando_select
```

- A sintaxe de alteração de visões é similar a de criação;
- O comando ALTER VIEW sobrepõe a definição da visão original, mantendo todas as permissões anteriormente atribuídas;
- O procedimento armazenado do sistema **sp\_helptext** possibilita visualizar o comando de criação de uma visão. Esse procedimento utiliza a tabela *syscomments*.

- Exemplo:

```
sp_helptext visaofofornecproduto
```



# Visões – *Views*

## Como excluir uma visão?

---

- Comando DROP VIEW;

- Sintaxe:

```
drop view nome_visão1[, ..., nome_visãoN]
```

- Exemplo:

```
drop view visaofofneccproduto
```



# Bibliografia

---

ELMASRI, R.; NAVATHE, S. B., Fundamentals of database systems. 7 ed., Pearson, 2016.

Visões (ou Exibições) no SQL Server

<https://docs.microsoft.com/pt-br/sql/relational-databases/views/create-views?view=sql-server-ver15>

<https://docs.microsoft.com/pt-br/sql/t-sql/statements/create-view-transact-sql?view=sql-server-ver15>