

Banco de Dados

Prof. Fernando Rodrigues de Almeida Júnior

UFC – Universidade Federal do Ceará

Curso: Eng. da Computação

e-mail: fernandorodrigues@sobral.ufc.br

Sumário

- ❑ Introdução aos Sistemas de Banco de Dados
- ❑ Projeto de Bancos de Dados Relacionais
- ❑ Bancos de Dados Relacionais
- ❑ Projeto Lógico de Bancos de Dados

➡ Linguagem de Manipulação e Consulta



➡ Visões em SQL

5. SQL

- Visões -

- ❑ Acesso a um banco de dados
 - ↳ Requer conhecimento do esquema
 - ⇒ Indesejável
 - ⇒ Para usuários inexperientes
 - ⇒ Desenvolvedores de aplicativos que acessam o BD
 - ↳ Por questões de segurança e privacidade
 - ⇒ Grupos de usuários devem ter acesso a dados de interesse
 - ⇒ O acesso a todo o banco de dados é perigoso
- ❑ Janelas sobre o banco de dados
 - ↳ Cada janela mostra parte do banco de dados
 - ⇒ Diferentes visões
 - ↳ Visões (views)
 - ⇒ Definidas sobre tabelas do banco de dados
 - ⇒ Tabelas base

5. SQL

- Visões -

❑ Tipos de visões

↳ Visão virtual

- ⇒ A definição da visão é armazenada
 - ⇒ Dados da visão não são persistentes
- ⇒ Sempre que referenciada
 - ⇒ Os dados são materializados
 - ◆ Custo praticamente igual a cada materialização
- ⇒ Quanto ao acesso
 - ⇒ Somente leitura
 - ◆ Visões que só permitem acesso de leitura
 - ⇒ Permitem atualização
 - ◆ Visões que permitem atualizações nas tabelas base

5. SQL

- Visões -

❑ Tipos de visões (cont.)

↳ Visão materializada

- ⇒ Dados e definição são persistentes
- ⇒ Problema de atualização dos dados da visão
 - ⇒ Sempre que há uma atualização nas tabelas base da visão
 - ⇒ Recalculada
 - ⇒ Atualizada
 - Com intervenção humana
 - Automática
- ⇒ Reduz custos de materialização de resultado
- ⇒ Visões somente para leitura
- ⇒ Aplicações
 - ⇒ Implementação Data Warehouse
 - ⇒ Integração de fontes de dados heterogêneas

5. SQL

- Visões -

❑ Definição de visões (SQL Server)

```
CREATE VIEW nome_da_visão [(nome_coluna {, nome_coluna ...})]  
[WITH <característica_da_visão> [ ,...n ] ]  
AS expressão_SQL [WITH CHECK OPTION]
```

```
<view_attribute> ::=  
{[SCHEMABINDING]  
[ENCRYPTION ]  
[ VIEW_METADATA ] }
```

➡ WITH CHECK OPTION

- ⇒ Atualizações (INSERT ou UPDATE) na tabela base através da visão só serão permitidas se
 - ⇒ Resultam em tuplas visíveis para a visão

5. SQL

- Visões -

❑ Definição de visões (SQL Server)

↳ SCHEMABINDING

- ⇒ Associa a visão ao esquema das tabelas base da visão
 - ⇒ Não permite que as tabelas base sejam alteradas (ou removidas), caso a definição da visão seja afetada
 - Remover coluna da tabela base que também aparece na definição da visão
 - ⇒ Para permitir alteração das tabelas base
 - Alterar definição da visão ou remover a visão
- ⇒ Obrigatório para visões materializadas
- ⇒ Na expressão SQL da definição da visão
 - ⇒ Nomes de tabelas, visões, funções têm que ser na forma <schema.objeto>
 - dbo.empregado

5. SQL

- Visões -

❑ Definição de visões (SQL Server)

↳ ENCRYPTION

⇒ Criptografa a definição da visão em sys.syscomments

↳ VIEW_METADATA

⇒ Não permite a visualização do nome das tabelas base da visão

5. SQL

- Visões -

❑ Exemplos

↳ Definindo visões

```
create view V1 (nome_departamento, nome_empregado)
as select d.nome,e.nome
   from Departamento d inner join Empregado e
      on d.cod_depart=e.lotação
```

```
create view V2 (nome_empregado, número_de_dependentes)
as select e.nome, (select count(*) from Dependente where
                    matr_resp=e.matr)
   from Empregado e
```

```
create view V3 (nome,salário)
as select nome,salário
   from Empregado
  where salário<700 with check option
```

5. SQL

- Visões -

❑ Exemplos

→ Acessando o banco de dados através de visões

```
select * from V1
```

```
select * from V2 where número_de_dependentes>2
```

```
select * from V3
```

nome	salário
caio	500.0
rebeca	500.0

→ Atualizando o banco de dados através de visões

```
update v3 set salário=salário+100
```

```
select * from V3
```

nome	salário
caio	600.0
rebeca	600.0

```
update v3 set salário=salário+150
```

Erro, pois as tuplas a serem alteradas vão deixar de ser visíveis para V3

5. SQL

- Visões -

❑ Visão materializada – SQL Server

**Create view V4 (cod_dep,nome, contador_grupos) with schemabinding
As**

```
select d.cod_dep, d.nome, count_big(*)  
from dbo.Empregado e, dbo.Departamento d  
where d.cod_dep=e.lotação  
group by d.cod_dep, d.nome
```

Create unique clustered index I_V4 on V4 (cod_dep)

5. SQL

- Visões -

- ❑ Visões que permitem atualizações apresentam as seguintes restrições na subconsulta
 - A cláusula *FROM* deve possuir apenas uma tabela
 - ⇒ Tabela base
 - ⇒ Visão
 - ⇒ Deve permitir atualização
 - Não estão especificadas as cláusulas *group by* e *having*
 - A palavra reservada *distinct* não está especificada
 - A cláusula *where* não contém subconsulta que referencia qualquer tabela na cláusula *from* diretamente ou indiretamente (via visões)
 - Todas as colunas da subconsulta são colunas simples
 - ⇒ Não são permitidas colunas do tipo *avg(salário)* ou expressões aritméticas
 - Possui a chave primária da relação base, bem como todos os atributos com a restrição *NOT NULL* que não têm default

5. SQL

- Visões -

- ❑ Definição (simplificada) de visões (MySQL):

```
CREATE VIEW view_name [(column_list)]  
    AS select_statement  
[WITH [CASCADED | LOCAL] CHECK OPTION];
```

- ❑ Remoção de visões:

```
DROP VIEW view_name;
```

- ❑ Alteração de visões:

```
ALTER VIEW view_name [(column_list)]  
    AS select_statement
```

5. SQL

- Visões -

❑ *select_statement*

- Cláusula SELECT que fornece a definição da visão;
- Esta cláusula pode selecionar (consultar) a partir de tabelas base ou de outras visões;
- A definição da visão é “congelada” no momento da criação. Mudanças posteriores nas tabelas subjacentes não afetam a definição da visão. Por exemplo, se uma visão é definida como “Select * From Empregado”, e novas colunas são adicionadas posteriormente a esta tabela (Empregado), a visão não é alterada / afetada.

5. SQL

- Visões -

❑ WITH [CASCADED | LOCAL] CHECK OPTION

- Esta cláusula pode ser usada para restringir “*inserts*” ou “*updates*” para linhas das tabelas referenciadas por uma visão atualizável, exceto para aquelas linhas para as quais a cláusula WHERE na sentença SELECT é verdade.
- As palavras reservadas LOCAL e CASCADED determinam o escopo do teste de check quando a visão é definida em termos de outra visão:
 - A palavra LOCAL restringe o CHECK OPTION somente para a visão sendo definida;
 - CASCADED faz com que os checks para as visões subjacentes sejam avaliados também.

5. SQL

- Visões -

❑ Exemplo – Visão (MySQL):

- Cria uma tabela, insere uma tupla, cria uma visão sobre esta tabela e faz uma consulta sobre a mesma:

```
CREATE TABLE t (qty INT, price INT);  
INSERT INTO t VALUES (3, 50);  
CREATE VIEW v AS SELECT qty, price, qty*price AS value FROM t;  
SELECT * FROM v;
```

- Resultado da consulta sobre a visão criada

qty	price	value
3	50	150

5. SQL

- Views -

❑ Exercícios:

- Especificar as seguintes visões:
 - Exibir os nomes dos funcionários juntamente com os nomes dos departamentos em que trabalham e os nomes de seus dependentes. O resultado deve ser exibido ordenado por nome do departamento e nome dos funcionários, ambos em ordem crescente;
 - Exibir os nomes dos departamentos de forma agrupada, mostrando ainda o número de funcionários que trabalha naquele departamento, o total de salários pagos para os funcionários daquele departamento e a média salarial do mesmo;

Dica: Utilize as cláusulas específicas para junções na cláusula 'FROM' do SQL!

Referências



- ❑ Notas de Aula – Prof. Angelo Brayner
- ❑ Manual MySQL 5.7



FIM