

# Visão & Visão Materializada

Laboratório de Bases de Dados

Profa. Dra. Cristina Dutra de Aguiar Ciferri

# Visão

- Em SQL
  - tabela simples que é derivada de outras tabelas
  - não existe necessariamente em sua forma física: **tabela virtual**
- Utilidade
  - forma de se especificar uma tabela que precisa ser acessada frequentemente, embora essa tabela não exista fisicamente
  - facilita a escrita de consultas complexas

# CREATE VIEW

```
CREATE OR REPLACE VIEW nome_visão  
[(nome_atributo [, nome_atributo ...])] AS  
<SELECT>  
[WITH [CHECK OPTION | READ ONLY]]
```

- Especifica uma visão
- Características
  - lista\_de\_atributos: opcional
  - SELECT: especifica o conteúdo da visão

# Exemplo

- Crie a visão `emprega_araucaria`, a qual recupera o primeiro nome e o sobrenome dos funcionários da empresa que contém Araucária em seu nome.

```
CREATE OR REPLACE VIEW emprega_araucaria AS
SELECT primeiro_nome, sobrenome
FROM funcionario, empresa
WHERE funcionario.emp_codigo = empresa.emp_codigo
      AND upper(emp_nome) LIKE upper ('%Araucaria%');

SELECT *
FROM emprega_araucaria;
```

# Operações sobre Visões

- Não atualizáveis: READ-ONLY
  - seleção: SELECT
- Atualizáveis
  - seleção: SELECT
  - inserção: INSERT INTO
  - remoção: DELETE
  - atualização: UPDATE

dados alterados: dados  
das relações base

# Operações sobre Visões

- Visões inerentemente atualizáveis **não** possuem
  - operadores de conjunto
  - DISTINCT
  - funções de agregação
  - GROUP BY
  - ORDER BY
  - subconsulta aninhada
  - JOIN
  - *stored procedures*

# Operações sobre Visões

- Visão “somente leitura”
  - visão que permite somente a realização de operações de seleção
- Visão “atualizável”
  - visão que permite as operações de seleção, inserção, remoção e atualização

# Exemplo

- Insira os seguintes dados na visão `emprega_araucária`:
    - (i) `primeiro_nome`: 'Prim\_Teste';
    - (ii) `sobrenome_nome`: 'SobreNome\_Teste'.
  - `INSERT INTO emprega_araucaria VALUES ('Prim_teste', 'SobreNome_Teste');`
- A inserção é realizada? Por quê? Qual tabela do BD é alterada?

# Exemplo

- Crie a visão cliente\_codigo1, a qual lista o código e o nome dos clientes, para aqueles clientes cujo código se inicia com 1.

```
CREATE OR REPLACE VIEW cliente_codigo1 AS
SELECT cli_codigo, cli_nome
FROM cliente
WHERE cli_codigo LIKE ('1%');

SELECT *
FROM cliente_codigo1;
```

# Exemplo

- Insira os seguintes dados na visão cliente\_codigo1:
    - (i) código: 100;
    - (ii) nome: 'Teste Completo'
  - INSERT INTO cliente\_codigo1 VALUES (100, 'Teste Completo');
- A inserção é realizada? Por quê? Qual tabela do BD é alterada?

# Exemplo

```
CREATE OR REPLACE VIEW cliente_codigo1 AS  
SELECT cli_codigo, cli_nome  
FROM cliente  
WHERE cli_codigo LIKE ('1%')  
WITH READ ONLY;
```

- INSERT INTO cliente\_codigo1 VALUES (100, 'Teste Completo');

➤ A inserção é realizada? Por quê? Qual tabela do BD é alterada?

# WITH CHECK OPTION

- Não permite que as operações violem a condição de seleção que define a visão

```
CREATE OR REPLACE VIEW cliente_codigo1 AS  
SELECT cli_codigo, cli_nome  
FROM cliente  
WHERE cli_codigo LIKE ('1%')  
WITH CHECK OPTION;
```

```
INSERT INTO cliente_codigo1 VALUES (300, 'Teste');
```

➤ A inserção é realizada? Por quê?

# DROP VIEW

```
DROP VIEW nome_visão
```

- Remove a definição de uma visão
- Observação:
  - para se alterar uma visão, ela primeiramente deve ser excluída para depois ser criada novamente

# Visão Materializada

- Em SQL
  - tabela simples que é derivada de outras tabelas
  - existe necessariamente em sua forma física: **não é uma tabela virtual**
- Discussão
  - replicação dos dados
  - armazenamento de dados agregados
  - custo de consultas x custo de atualização

# Visão Materializada

```
CREATE MATERIALIZED VIEW nome_visão  
[BUILD [DEFERRED | IMMEDIATE]]  
[REFRESH [COMPLETE | FAST]]  
    [ON COMMIT| ON DEMAND]]  
[[ENABLE | DISABLE] QUERY REWRITE]  
AS  
<SELECT>
```

+ diversas outras opções

# CREATE MATERIALIZED VIEW

- BUILD
  - *quando a visão materializada é populada*
  - IMMEDIATE: imediatamente
  - DEFERRED: primeiro REFRESH
- REFRESH
  - *como é feita a atualização da visão*
  - COMPLETE: atualiza completamente a visão, executando o comando SELECT
  - FAST: somente considera as alterações realizadas (atualização incremental)

# CREATE MATERIALIZED VIEW

- QUERY REWRITE
  - *se a visão materializada pode ser usada para reescrita de consultas*
  - ENABLE: sim
  - DISABLE: não
- Diversas outras opções ...

# Exemplo

```
CREATE MATERIALIZED VIEW
    conta_empresa (nome_empresa, nro_filiais)
BUILD IMMEDIATE
REFRESH COMPLETE AS
SELECT emp_nome, count(*)
FROM empresa LEFT JOIN filial
    ON empresa.emp_codigo = filial.emp_codigo
GROUP BY emp_nome;
```