

# Views

Banco de Dados Relacional e NoSQL

Prof. Moisés Olímpio

# Tópicos

- Views
- Importância do SQL
- Organização e Manutenção
- Controle de Acesso
- Desempenho das Views
- Materialized Views

# Views (Visualizações)

- Uma view é uma consulta SQL nomeada armazenada no banco de dados.
- Na prática criamos a estrutura da consulta e geramos a view, toda vez que precisar usar a consulta faz o uso da view diretamente.
- Views não **\*\*armazenam dados\*\***, toda vez que é usada faz a busca das informações.

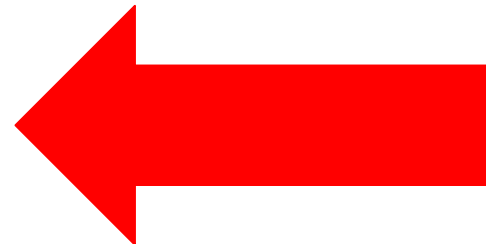
# Exemplo

```
CREATE VIEW vw_alunos_por_turma AS
SELECT
    at.matricula_aluno,
    at.cod_turma,
    a.nome AS aluno_nome,
    t.cod_curso,
    c.nome AS curso_nome
FROM alunos_turmas at
JOIN alunos a ON at.matricula_aluno = a.matricula
JOIN turmas t ON at.cod_turma = t.cod_turma
JOIN cursos c ON t.cod_curso = c.cod_curso;
```

```
select * from vw_alunos_por_turma;
```

Dentro do sistema  
faz uso somente da  
view ao invés de  
colocar todo  
comando SQL no  
código fonte.

Comando usado  
para chamar a  
view



# Importância do SQL

- As views permitem criar consultas uma única vez e reutilizar sempre que necessário.
- No desenvolvimento de sistemas é comum que a mesma informação seja solicitada em diferentes partes do projeto, views ajudam a centralizar a busca da informação.

# Organização e Manutenção

- As views centralizam a lógica da consulta facilitando a organização do projeto.
- Imagine um cenário onde dentro do projeto foi escrito comandos SQL em várias partes e esse comando faz a mesma função, exemplo: buscar uma lista de produtos. Por algum motivo a estrutura da tabela foi modificada, sendo assim, será necessário atualizar os comandos SQL em todo sistema.

# Controle de acesso

- Alguns sistemas trabalham com informações sensíveis, por exemplo folha de pagamento, ao usar views é possível ocultar possíveis colunas do banco não deixando claro as informações existentes.

```
select
| nome, salario_base, descontos, total
from
| colaboradores;
```

Campos visíveis

```
CREATE VIEW vw_total_salario AS
SELECT
| c.nome,
| c.descontos,
| c.total
FROM colaboradores;

select * from vw_total_salario;
```

Campos escondidos

# Desempenho das Views

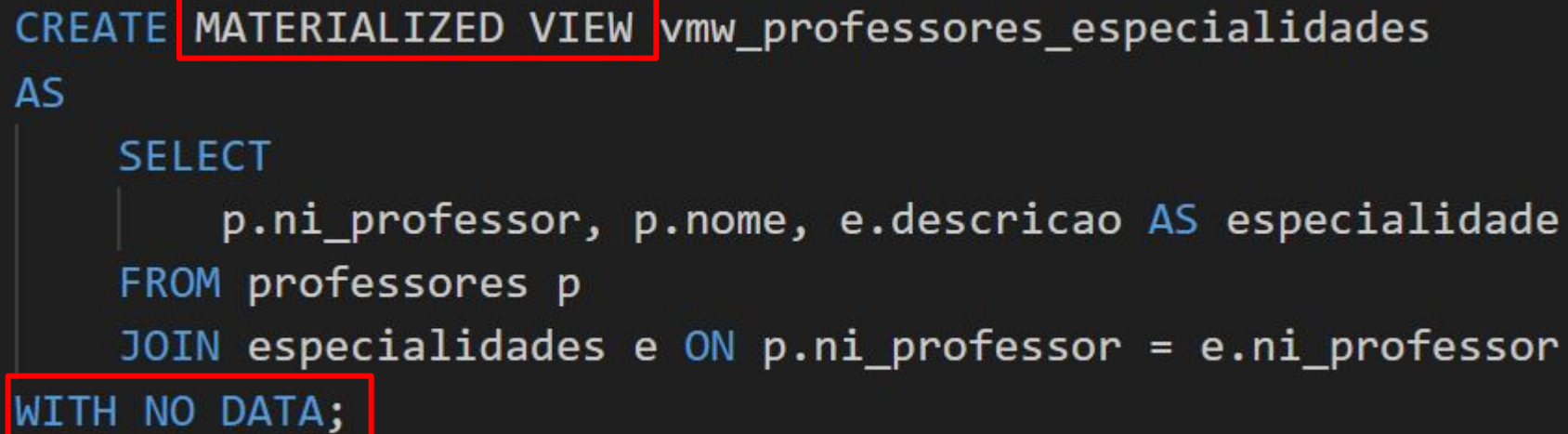
- Até o momento entendemos que as Views ajudam em organização e gerenciamento de consultas SQL, em relação ao desempenho não tem nenhuma diferença.
- Dentro do universo das Views existem as “Materialized Views” ou seja, views materializadas que consegue armazenar permanentemente as informações consultadas.
- Casos onde o acesso a informação é constante as materialized views é uma excelente opção.



# Materialized Views

## Sintaxe para criar uma materialized view

```
CREATE MATERIALIZED VIEW vmw_professores_especialidades
AS
    SELECT
        p.ni_professor, p.nome, e.descricao AS especialidade
    FROM professores p
    JOIN especialidades e ON p.ni_professor = e.ni_professor
WITH NO DATA;
```



Indica que os dados não serão consultados ao criar a view, se remover “WITH NO DATA” a view é preenchida no começo.

# Materialized Views

```
17 select * from vmw_professores_especialidades;
```

```
18
```

Data Output Messages Notifications

ERROR: visão materializada "vmw\_professores\_especialidades" não foi preenchida

HINT: Use o comando REFRESH MATERIALIZED VIEW.

ERRO: visão materializada "vmw\_professores\_especialidades" não foi preenchida

SQL state: 55000

# Materialized Views

```
17 REFRESH MATERIALIZED VIEW vmw_professores_especialidades;  
18  
19 select * from vmw_professores_especialidades;  
20
```

Data Output Messages Notifications

	ni_professor character varying (10) 🔒	nome character varying (255) 🔒	especialidade character varying (255) 🔒
1	P001	Carlos Silva	Programação Web
2	P002	Maria Oliveira	Banco de Dados
3	P003	José Santos	Redes de Computadores
4	P004	Fernanda Costa	Desenvolvimento de Software
5	P005	Ricardo Almeida	Engenharia de Software

Obrigado pela atenção!  
Dúvidas ou considerações?