

Banco de Dados II



Técnicas de Programação em Banco de Dados Views

Prof. Tadeu Classe
tadeu.classe@uniriotec.br

- Views ou visões são formas alternativas de se olhar dados contidos em uma ou mais tabela.
- Com as views, é possível tratar resultados de uma consulta por meio de uma “tabela”;
- É uma excelente maneira de transformar consultas complexas a dados, em consultas simples e mais rápidas.

- Views em SQL são consultas armazenadas em uma estrutura de fácil acesso baseada em comandos SELECT;
- Essa consulta é então armazenada em uma tabela virtual, com o comportamento similar ao de uma tabela real;
- Porém, sem armazenamento de dados;
- Os dados exibidos pelas views são gerados dinamicamente toda vez que a mesma é referenciada;

- Um SGBD armazena somente a definição de uma view;
 - Nome da View;
 - Comandos SELECT;
- Quando a view é chamada o banco associa os dados atribuídos a ela;
- A view apresenta o resultado final do processo de consulta, ocultando os comandos para a sua criação;

- A criação da View é bem simples e sua sintaxe pode ser vista abaixo:

```
create view <Nome_View>  
as <Comando_Select>
```

- Na expressão:
 - <nome_view>: nome para o qual deseja-se atribuir;
 - <comando_select>: execução de um comando select válido;

```
create view <Nome_View>  
as <Comando_Select>
```

- Exercitando
 - View Estática: esta view contem dados estáticos estipulados pelos usuário;

```
CREATE VIEW visao_estatica AS  
    SELECT 'Brasil' as Pais,  
           'Minas Gerais' as estado,  
           'Juiz de Fora' as cidade
```

- Exercitando
 - View Estática: para exibir os valores da view que acabamos de criar, basta realizar um SELECT simples como se fosse uma tabela;

```
SELECT *  
FROM visao_estatica;
```

| | pais | estado | cidade |
|---|--------|--------------|--------------|
| ▶ | Brasil | Minas Gerais | Juiz de Fora |

- Exercitando:
 - View Dinâmica: Como dito anteriormente, uma view executa um SELECT e então o exibe ao ser executada;
 - Portanto é possível criamos selects (mesmo os mais complexos) e criar views a partir deles
 - Isso evita que o usuário execute diversas vezes um comando complexo, simplesmente executando um SELECT simples.

- Exercitando:
 - View Dinâmica (Usando o banco World)
 - Exemplo 1: Retornar todas as cidades da tabela city;
 - 1º. Crie o select para retornar todas as cidades;

```
SELECT *  
FROM city
```

| | ID | Name | CountryCode | District | Population |
|--|----|------------|-------------|--------------|------------|
| | 1 | Kabul | AFG | Kabol | 1780000 |
| | 2 | Qandahar | AFG | Qandahar | 237500 |
| | 3 | Herat | AFG | Herat | 186800 |
| | 4 | Mazar-e... | AFG | Balkh | 127800 |
| | 5 | Amster... | NLD | Noord-H... | 731200 |
| | 6 | Rotterdam | NLD | Zuid-Holl... | 593321 |
| | 7 | Haag | NLD | Zuid-Holl... | 440900 |
| | 8 | Utrecht | NLD | Utrecht | 234323 |
| | 9 | Eindhoven | NLD | Noord-Br... | 201843 |
| | 10 | Tilburg | NLD | Noord-Br... | 193238 |
| | 11 | Groningen | NLD | Groningen | 172701 |

- Exercitando:
 - View Dinâmica (Usando o banco World)
 - Exemplo 1: Retornar todas as cidades da tabela city;
 - 2º. Crie a view para este select;

```
CREATE view cidades AS  
SELECT *  
FROM city
```

- Exercitando:
 - View Dinâmica (Usando o banco World)
 - Exemplo 1: Retornar todas as cidades da tabela city;
 - 0. Execute a view que acabou de ser criada

```
SELECT *  
FROM cidades
```

| | ID | Name | CountryCode | District | Population |
|--|----|------------|-------------|--------------|------------|
| | 1 | Kabul | AFG | Kabol | 1780000 |
| | 2 | Qandahar | AFG | Qandahar | 237500 |
| | 3 | Herat | AFG | Herat | 186800 |
| | 4 | Mazar-e... | AFG | Balkh | 127800 |
| | 5 | Amster... | NLD | Noord-H... | 731200 |
| | 6 | Rotterdam | NLD | Zuid-Holl... | 593321 |
| | 7 | Haag | NLD | Zuid-Holl... | 440900 |
| | 8 | Utrecht | NLD | Utrecht | 234323 |
| | 9 | Eindhoven | NLD | Noord-Br... | 201843 |
| | 10 | Tilburg | NLD | Noord-Br... | 193238 |
| | 11 | Groningen | NLD | Groningen | 172701 |

Criação da View

- Exercitando:
 - View Dinâmica (Usando o banco World)
 - Exemplo 2: Retornar todas as cidades da tabela city ordenadas alfabeticamente, que tenham população maior que 450.000 e que esteja em Minas Gerais;
 - 1º. Crie o select que se pede;

```
SELECT *  
FROM city c  
WHERE c.population > 450000  
      AND c.district = 'Minas Gerais'  
ORDER by c.name
```

| | ID | Name | Count | District | Population |
|---|-----|----------------|-------|--------------|------------|
| ▶ | 209 | Belo Horizonte | BRA | Minas Gerais | 2139125 |
| | 233 | Contagem | BRA | Minas Gerais | 520801 |
| | 241 | Juiz de Fora | BRA | Minas Gerais | 450288 |
| | 235 | Uberlândia | BRA | Minas Gerais | 487222 |

- Exercitando:
 - View Dinâmica (Usando o banco World)
 - Exemplo 2: Retornar todas as cidades da tabela city ordenadas alfabeticamente, que tenham população maior que 450.000 e que esteja em Minas Gerais;
 - 2º. Crie o view com o Select;

```
CREATE VIEW cidade_mg_ord as
SELECT *
FROM city c
WHERE c.population > 450000
      AND c.district = 'Minas Gerais'
ORDER by c.name
```

Criação da View

- Exercitando:
 - View Dinâmica (Usando o banco World)
 - Exemplo 2: Retornar todas as cidades da tabela city ordenadas alfabeticamente, que tenham população maior que 450.000 e que esteja em Minas Gerais;
 - 0. Selecione a View Criada

```
SELECT *  
FROM cidade_mg_ord
```

| | ID | Name | Count | District | Population |
|---|-----|----------------|-------|--------------|------------|
| ▶ | 209 | Belo Horizonte | BRA | Minas Gerais | 2139125 |
| | 233 | Contagem | BRA | Minas Gerais | 520801 |
| | 241 | Juiz de Fora | BRA | Minas Gerais | 450288 |
| | 235 | Uberlândia | BRA | Minas Gerais | 487222 |

- Exercitando:
 - View Dinâmica (Usando o banco World)
 - Como dito antes as views armazenam o seu nome e o comando SELECT para sua execução. Então, TEORICAMENTE quando se alterar dados no banco de dados a view acompanha as alterações.
 - Vejamos o exemplo 2, que retorna cidades mineiras com mais de 450 mil habitantes, só foram listadas: BH, CONT, JF e Uberlandia.

- Exercitando:
 - View Dinâmica (Usando o banco World)
 - Vamos testar: Altere no banco de dados qualquer uma das cidades mineiras, que já não forem listadas para 500 mil habitantes:

```
UPDATE city  
SET Population='500000'  
WHERE ID='266';
```

- EX: Betim
- Após isso, execute a view do exmplo 2 e veja o resultado

Criação da View

- Exercitando:
 - View Dinâmica (Usando o banco World)

```
SELECT *  
FROM cidade_mg_ord
```

| | ID | Name | Country | District | Population |
|---|-----|----------------|---------|--------------|------------|
| ▶ | 209 | Belo Horizo... | BRA | Minas Gerais | 2139125 |
| | 266 | Betim | BRA | Minas Gerais | 500000 |
| | 233 | Contagem | BRA | Minas Gerais | 520801 |
| | 241 | Juiz de Fora | BRA | Minas Gerais | 450288 |
| | 235 | Uberlândia | BRA | Minas Gerais | 487222 |

- A alteração do banco não influencia na View, pois a mesma por executar um select, pega tudo que foi modificado e exibe após a execução.

- Exercitando:
 - View Dinâmica (Usando o banco World)
 - Exemplo 3: Crie uma view que recupere do banco a quantidade de população de todos os estados brasileiros;

- Exercitando:
 - View Dinâmica (Usando o banco World)
 - Exemplo 3: Crie uma view que recupere do banco a quantidade de população de todos os estados brasileiros;

```
CREATE view populacao_estados AS
select c.district as estado, sum(c.population) as habitantes
from city c
where c.countrycode = 'BRA'
group by 1
order by 1
```

Criação da View

- Exercitando:
 - View Dinâmica (Usando o banco World)
 - Exemplo 3: Crie uma view que recupere do banco a quantidade de população de todos os estados brasileiros;

```
select *  
From populacao_estados;
```

| estado | habitantes |
|--------------------|------------|
| Acre | 259537 |
| Alagoas | 965276 |
| Amapá | 256033 |
| Amazonas | 1255049 |
| Bahia | 4765206 |
| Ceará | 2943123 |
| Distrito Federal | 1969868 |
| Espírito Santo | 1579739 |
| Goiás | 1985741 |
| Maranhão | 1616536 |
| Mato Grosso | 823363 |
| Mato Grosso do Sul | 904420 |
| Minas Gerais | 7347952 |
| Paraná | 1140180 |

- Exercitando:
 - View Dinâmica (Usando o banco World)
 - Exemplo 4: Crie uma view que retorne os países que tenha como idioma o Portugues, que tenha o idioma como oficial ou que tenham pelo menos 10% de falantes. Mostre o Pais, a porcentagem de falantes, o continente e a forma de governo;

- Exercitando:
 - View Dinâmica (Usando o banco World)
 - Exemplo 4: Crie uma view que retorne os países que tenha como idioma o Portugues, que tenha o idioma como oficial ou que tenham pelo menos 10% de falantes. Mostre o Pais, a porcentagem de falantes, o continente e a forma de governo;

```
Create view paises_portugues as
select c.Name as pais, l.percentage, c.Continent, c.GovernmentForm
from countrylanguage l
join country c
on c.Code = l.CountryCode
where l.Language = 'portuguese'
and (l.isofficial = 'T' or l.Percentage > 10)
```

Criação da View

- Exercitando:
 - View Dinâmica (Usando o banco World)
 - Exemplo 4:

```
select *  
from paisess_portugues
```

| | pais | percentage | Continent | GovernmentForm |
|---|---------------|------------|---------------|--|
| | Andorra | 10.8 | Europe | Parliamentary Coprincipality |
| ▶ | Brazil | 97.5 | South America | Federal Republic |
| | Cape Verde | 0.0 | Africa | Republic |
| | Guinea-Bissau | 8.1 | Africa | Republic |
| | Luxembourg | 13.0 | Europe | Constitutional Monarchy |
| | Macao | 2.3 | Asia | Special Administrative Region of China |
| | Portugal | 99.0 | Europe | Republic |
| | East Timor | 0.0 | Asia | Administrated by the UN |

Consultas a Views

- Assim como tabelas de banco de dados, é possível realizar qualquer filtro como WHERE, group e etc.
- Ex (View exemplo 4): Filtrar países europeus;

```
select *  
from paises_portugues p  
where p.Continent = 'Europe'
```

| | pais | percentage | Continent | GovernmentForm |
|---|----------|------------|-----------|------------------------------|
| ► | Andorra | 10.8 | Europe | Parliamentary Coprincipality |
| | Luxem... | 13.0 | Europe | Constitutional Monarchy |
| | Portugal | 99.0 | Europe | Republic |

- Assim como tabelas de banco de dados, é possível realizar qualquer filtro como WHERE, group e etc.
- Ex (View exemplo 4): Contar a quantidade de países falantes de português por cada continente;

```
select continent, count(pais)
from pais_portugues p
group by continent
```

| Continent | count(pais) |
|---------------|-------------|
| Asia | 2 |
| Europe | 3 |
| Africa | 2 |
| South America | 1 |

- DROP: Exclui uma view

```
DROP VIEW <nome_view>
```

- As views são visões de consultas ao banco de dados utilizados pelos usuários;
- Não é possível incluir (INSERT), alterar (UPDATE) ou excluir (DELETE) informações de uma view.
- É possível realizar qualquer um filtro de informação das views da mesma maneira que se faz com uma tabela de banco;
- A view simplifica o acesso a determinados dados, não necessitando SQLs complexos;
- Veremos que as views são úteis para segurança.

Obrigado e Até a Próxima Aula!

Contato:

Prof. Tadeu Classe
Bacharelado em Sistemas de
Informação (BSI)
Universidade Federal do Estado do
Rio de Janeiro (UNIRIO)

tadeu.classe@uniriotec.br



O trabalho **Banco de Dados II - Universidade Federal do Estado do Rio de Janeiro (UNIRIO)** de Tadeu Moreira de Classe está licenciado com uma Licença [Creative Commons - Atribuição-NãoComercial-CompartilhaIgual 4.0 Internacional](#).

