

Visões e permissões de acesso

Vinicius da Silveira Segalin

vinicius.segalin@posgrad.ufsc.br

Sumário

- **Visões**
- Permissões de acesso
- Exercícios

Visões

- O que é
 - Uma tabela virtual baseada no resultado de uma consulta sobre uma ou mais tabelas
- Para que serve
 - Abstrair a complexidade das tabelas
 - Simplificar (especialmente) para os usuários finais
 - Restringir que o usuário veja apenas o que ele pode ver
- Como funciona
 - SGBD cria uma tabela virtual sobre os dados resultantes da consulta
 - Sem dados redundantes
 - A visão é consultada como uma tabela comum

Visões

- Criando uma visão

CREATE VIEW `view_name` (`col1, col2..., coln`) → Lista opcional para os campos da visão
AS
`consulta`
↓
Nome da visão a ser criada
↓
Consulta que definirá a visão (SELECT FROM WHERE)

- Deletando uma visão

DROP VIEW `view_name`;

Visões

- Exemplo

```
SELECT * FROM tb_funcionario
```

	codfuncionario integer	nome character varying(100)	cargo character varying(50)	salario numeric	email character varying(100)
1	1	João	Gerente	5000	joao@gmail.com
2	2	José	Caixa	2500	jose@gmail.com
3	3	Maria	Caixa	2500	maria@gmail.com
4	4	Carlos	Diretor	10000	carlos@gmail.com
5	5	Ana	Sub-gerente	4000	ana@gmail.com

```
CREATE VIEW v_funcionarios (nome, cargo, contato)  
AS  
SELECT nome, cargo, email  
FROM tb_funcionario  
WHERE cargo <> 'Diretor';
```

```
SELECT * FROM v_funcionarios;
```

	nome character varying(100)	cargo character varying(50)	contato character varying(100)
1	João	Gerente	joao@gmail.com
2	José	Caixa	jose@gmail.com
3	Maria	Caixa	maria@gmail.com
4	Ana	Sub-gerente	ana@gmail.com

Visões

- Exemplo (visão de visão)

```
SELECT * FROM v_funcionarios;
```

	nome character varying(100)	cargo character varying(50)	contato character varying(100)
1	João	Gerente	joao@gmail.com
2	José	Caixa	jose@gmail.com
3	Maria	Caixa	maria@gmail.com
4	Ana	Sub-gerente	ana@gmail.com

```
CREATE VIEW v_funcionarios_2  
AS  
SELECT nome, cargo  
FROM v_funcionarios;
```

```
SELECT * FROM v_funcionarios_2;
```

	nome character varying(100)	cargo character varying(50)
1	João	Gerente
2	José	Caixa
3	Maria	Caixa
4	Ana	Sub-gerente

Sumário

- Visões
- **Permissões de acesso**
- Exercícios

Permissões de acesso

- O que é
 - Processo onde o SGBD determina quais operações e dados o usuário pode acessar
- Para que serve
 - Segurança
 - Proteção contra acessos não autorizados
- Como funciona
 - Privilégios são controlados através de papéis (roles)
 - SGBD verifica se o papel do usuário tem permissão de realizar a operação requisitada

Permissões de acesso

- Todo objeto (tabela, visão, função...) tem um dono
 - Apenas o dono pode realizar operações sobre os objetos
 - Ou papéis com permissão
- Permissão
 - Agente (role)
 - Grânulo (BD, tabela, colunas...)
 - Operação (select, insert, update...)

Permissões de acesso

- Superuser
 - Dono do BD
 - Alguns privilégios exclusivos
 - Concede/retira privilégios de acesso
- Outros papéis
 - Dono dos objetos que criou
 - Pode ter privilégio para conceder/retirar privilégios
 - Obs: papéis são independentes dos usuários do SO

```
CREATE USER vinicius;  
CREATE ROLE vinicius LOGIN;
```

Permissões de acesso

- GRANT – conceder permissões de acesso

```
GRANT privilégios  
ON objeto  
TO papel [WITH GRANT OPTION];
```

- Privilégios: ALL, SELECT, INSERT, UPDATE, DELETE, CREATE, CONNECT, EXECUTE...
- Objeto: tabelas, sequências, BDs, colunas específicas...
- Papel: papel específico ou PUBLIC

Permissões de acesso

- GRANT - exemplos
 - GRANT select, insert, update ON tb_pessoa TO gerente;
 - GRANT select (nome, email) ON tb_pessoa TO funcionario;
 - GRANT ALL ON tb_pessoa TO admin WITH GRANT PERMISSION;
 - GRANT admin TO gerente;
 - GRANT select ON v_pessoa TO funcionario;

Permissões de acesso

- REVOKE – retirar permissões de acesso

```
REVOKE privilégios  
ON objeto  
FROM papel [CASCADE];
```

- Exemplos:
 - REVOKE select ON tb_pessoa FROM funcionario;
 - REVOKE all ON tb_pessoa FROM gerente;
 - REVOKE admin FROM gerente;
 - REVOKE all FROM gerente;

Sumário

- Visões
- Permissões de acesso
- **Exercícios**