

Aplicação de banco de dados



SQL DDL - Integridade

Prof. Alexandre Vitoreti de Oliveira

Autor: Prof. Flavio Ceci



SEQUENCE

- Os campos auto-incremento são geralmente utilizados para definir chaves primárias;
- O PostgreSQL oferece um recurso chamado sequence para solucionar este problema;
- O valor de uma sequência é incrementado em 1 no momento de recuperar o seu valor, ou seja, cada vez que você pede qual o valor atual da sequência seu valor é incrementado em 1.



SEQUENCE

- Criar uma sequência:

```
CREATE SEQUENCE exemplo_sequencia;
```

- Definir o valor inicial da sequência:

```
CREATE SEQUENCE exemplo_sequencia START 250;
```

- Alterando o valor da sequência:

```
SELECT SETVAL('exemplo_sequencia', 10);
```

- Recuperando o valor atual da sequência:

```
SELECT NEXTVAL('exemplo_sequencia');
```



SEQUENCE

- Utilizando uma sequence no insert:

```
CREATE SEQUENCE exemplo_sequencia START  
101;
```

```
INSERT INTO clientes VALUES  
(nextval('exemplo_sequencia'), 'Nome Cliente');
```



SEQUENCE

- Vinculando uma sequence na criação de uma tabela:

```
CREATE SEQUENCE exemplo_sequence;
```

```
CREATE TABLE Cliente (  
  id integer DEFAULT nextval('exemplo_sequence'),  
  nome VARCHAR  
  Primary key(id) );
```



SERIAL

Campo serial

- O campo serial é uma alternativa para a criação de campos auto-incrementáveis:

```
CREATE TABLE usuarios (  
    id SERIAL PRIMARY KEY,  
    nome VARCHAR,  
    age integer);
```



SERIAL

Campo serial

- Para inserir valores na tabela criada no slide anterior:

```
INSERT INTO usuarios (name, age) VALUES  
('Mozart', 20);
```

OU

```
INSERT INTO usuarios (name, age, id) VALUES  
('Mozart', 20, DEFAULT);
```

Exercício – Database Livros

Implementar o modelo físico a partir do modelo lógico abaixo:

