

# Herança

## Banco de Dados

Prof. Igor Avila Pereira  
igor.pereira@riogrande.ifrs.edu.br

Divisão de Computação  
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)  
Câmpus Rio Grande

# Agenda

## 1 Herança

## Herança

Considere o banco de dados chamado *vendas*.

Suponha que você tenha duas tabelas: *person* (pessoa) e *seller* (vendedor).

person	
id	
name	

seller	
id	
name	
commission	

# Herança

Então fazemos:

```
$ sudo su - postgres  
$ createdb vendas  
$ psql vendas
```

```
CREATE TABLE person (  
    id SERIAL PRIMARY KEY,  
    name TEXT  
);  
CREATE TABLE seller (  
    id SERIAL PRIMARY KEY,  
    name TEXT,  
    commission DECIMAL(6,2)  
);  
INSERT INTO person (name) VALUES ('Paulo');  
INSERT INTO seller (name,commission) VALUES ('Roberto',149.99);
```

E agora?!

Como faço para retornar **TODAS** as pessoas do B.D?

Observação

**VENDEDORES SÃO PESSOAS.**

- Que não estão armazenadas na tabela *person*.

## Herança

## Alternativa 1:

Criar uma **VIEW**.

```
CREATE VIEW people AS
  SELECT name FROM person
  UNION
  SELECT name FROM seller;

SELECT * FROM people ;
```

Que retorna:

```
name
-----
Paulo
Roberto
(2 rows)
```

'Paulo' pertence a *person* e 'Roberto' pertence a *seller*.

# Herança

**VIEW** não é a melhor solução.

Alternativa 2:

Usando herança faríamos da seguinte forma:

```
DROP VIEW people ;
DROP TABLE person, seller;

CREATE TABLE person (
    id SERIAL PRIMARY KEY,
    name VARCHAR(50)
);
CREATE TABLE seller (
    commission DECIMAL(6,2)
) INHERITS (person);
```

# Herança

## Fazendo

```
\d person
```

Column	Type	Table "public.person"	Modifiers
id	integer	not null default	nextval('person_id_seq'::regclass)
name	character varying(50)		

Indexes:  
 "person\_pkey" PRIMARY KEY, btree (id)  
 Number of child tables: 1 (Use \d+ to list them.)

## E

```
\d seller
```

Column	Type	Table "public.seller"	Modifiers
id	integer	not null default	nextval('person_id_seq'::regclass)
name	character varying(50)		
commission	numeric(6,2)		

Inherits: person



## Herança

A diferença é que com menos código criamos as duas tabelas e não precisamos criar uma **VIEW**.

- Mas a tabela *seller* depende da tabela *person*.

### Vantagens:

- a associação é do tipo **one-to-one**
- o esquema é extensível
- evita duplicação de tabelas com campos semelhantes
- a relação de dependência é do tipo pai e filho
- podemos consultar o modelo pai e o modelo filho

### Desvantagens:

- adiciona sobrecarga substancial, uma vez que cada consulta em uma tabela filho requer um *join* com todas as tabelas pai.

**Vamos inserir alguns dados:**

```
INSERT INTO person (name) VALUES ('Paulo'),('Fernando');  
INSERT INTO seller (name,commission) VALUES  
    ('Roberto',149.99),  
    ('Rubens',85.01);
```

**Fazendo:**

```
SELECT name FROM person;
```

```
      name
```

```
-----
```

```
Paulo
```

```
Fernando
```

```
Roberto
```

```
Rubens
```

```
(4 rows)
```

## Herança

Fazendo:

```
SELECT name FROM person;
```

```
      name
```

```
-----
```

```
Paulo
```

```
Fernando
```

```
Roberto
```

```
Rubens
```

```
(4 rows)
```

Obtemos todos os nomes porque na verdade um ***seller*** também é um ***person***.

## Herança

Somente os registros de  
*person*.

```
SELECT name FROM ONLY person;
```

```
  name
-----
Paulo
Fernando
(2 rows)
```

Somente os registros de *seller*.

```
SELECT name FROM seller;
```

```
  name
-----
Roberto
Rubens
(2 rows)
```

# Herança

## Banco de Dados

Prof. Igor Avila Pereira  
igor.pereira@riogrande.ifrs.edu.br

Divisão de Computação  
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)  
Câmpus Rio Grande