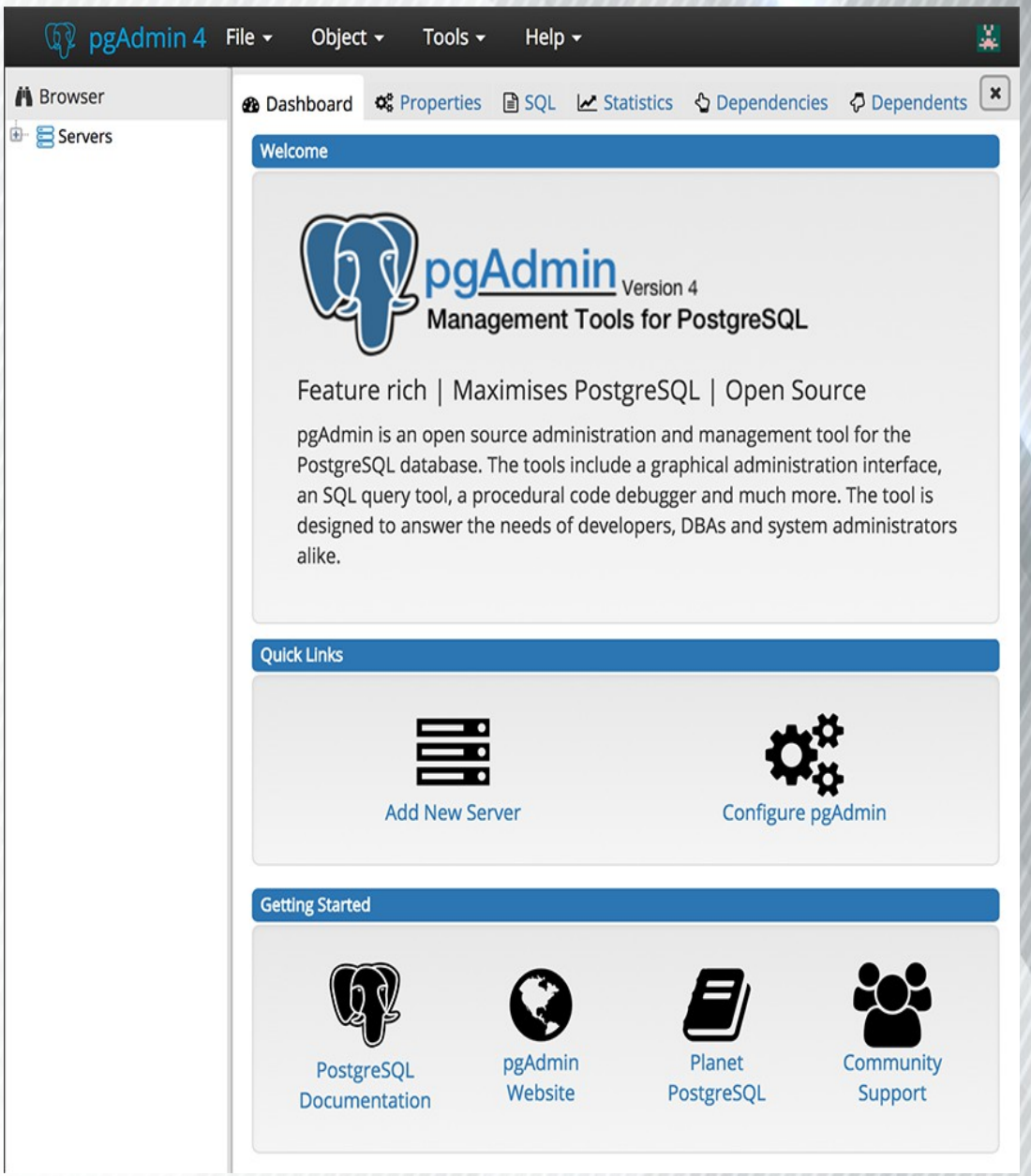
POSTGRES



PostgreSQL

tiagodev.valerio@gmail.com

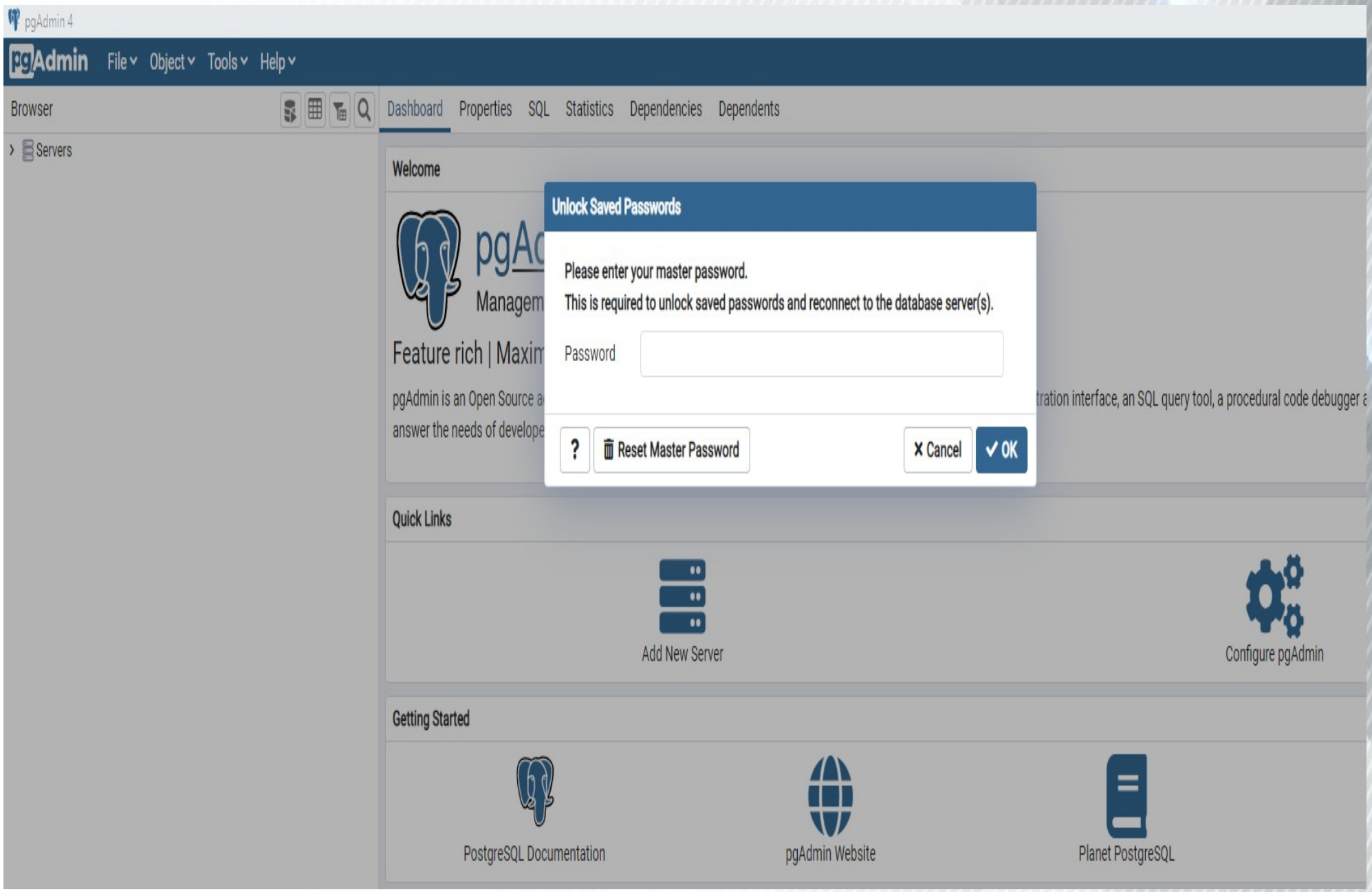
NOMENCLATURAS

PostgreSQL

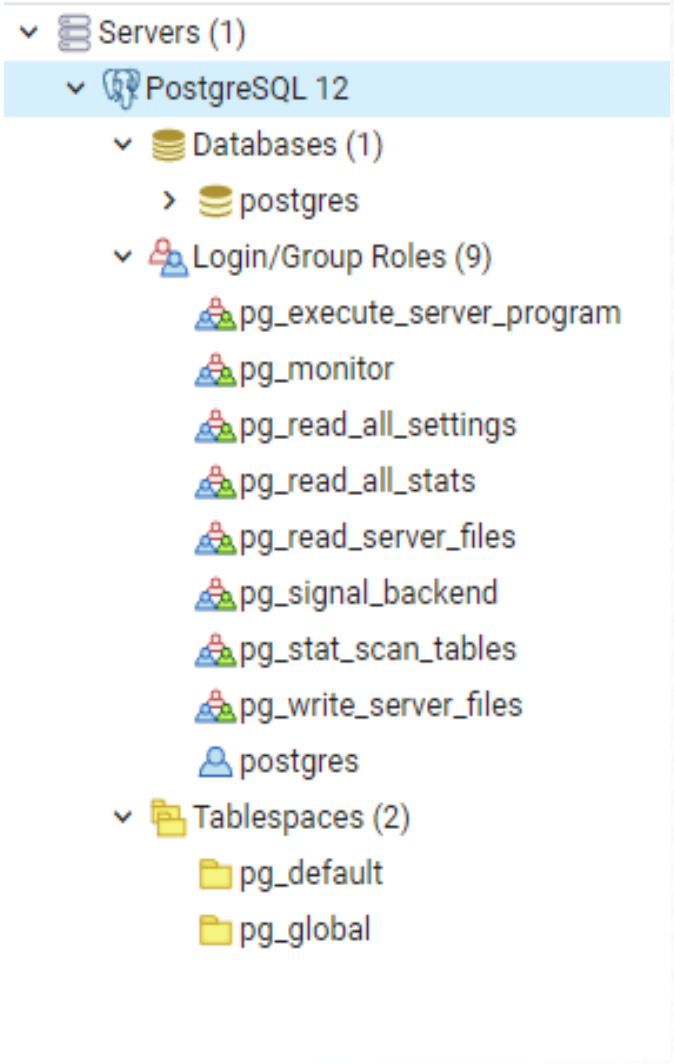
Comandos – Letra Maiuscula – Exemplo (CREATE, ALTER, SELECT, UPDATE).

Objetos – Letra Minúscula – Exemplo (tabelas, Colunas, indices).

ACESSO AO POSTGRES

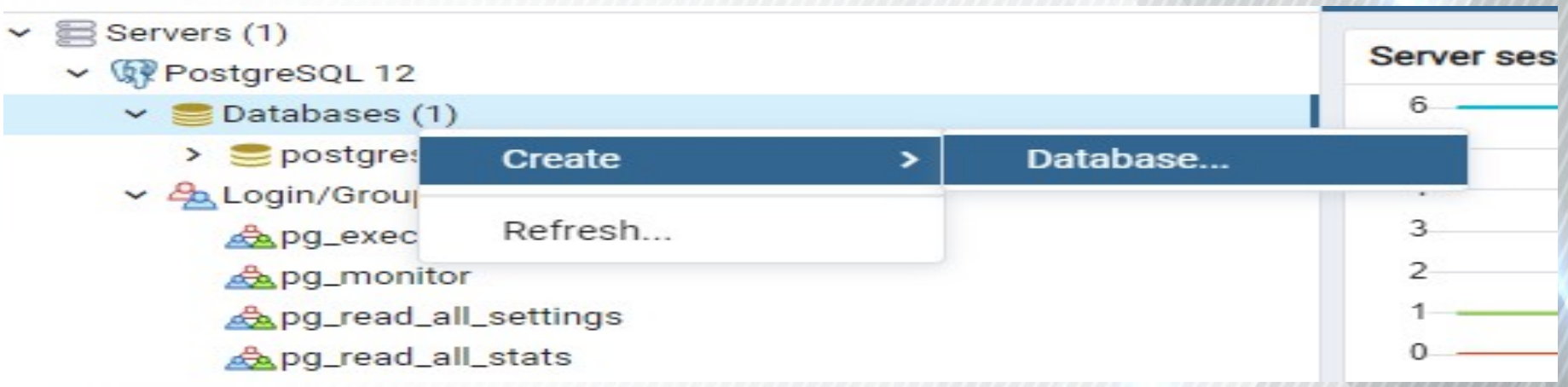


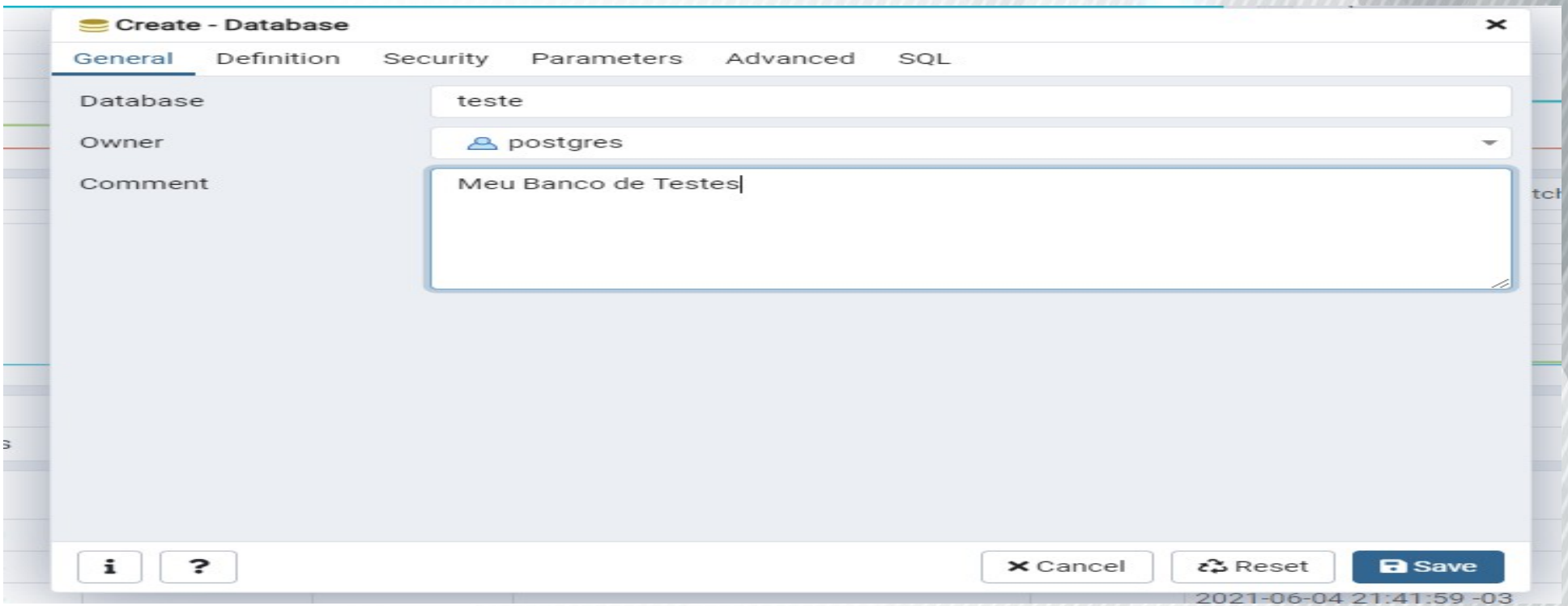
O QUE TEMOS ?



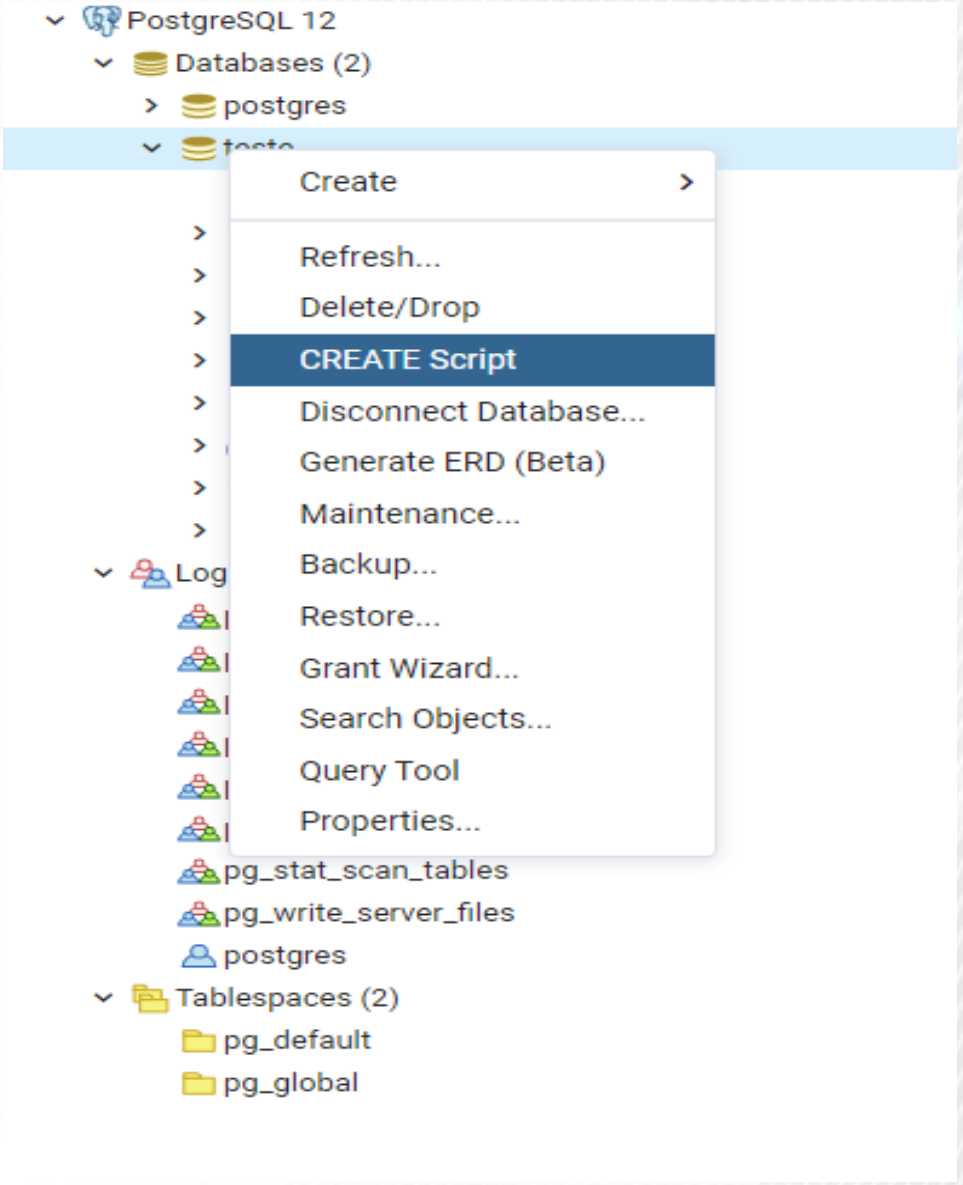
Será listado:
- databases (banco de dados)
- users (Usuários)
- tablespaces (onde os objetos ficam)

CRIAR DATABASE





NEW LIFE - SQL


PostgreSQL 12
Databases (2)
postgres
Create
Refresh...
Delete/Drop
CREATE Script
Disconnect Database...
Generate ERD (Beta)
Maintenance...
Backup...
Restore...
Grant Wizard...
Search Objects...
Query Tool
Properties...
pg_stat_scan_tables
pg_write_server_files
postgres
Tablespaces (2)
pg_default
pg_global

TIPOS DE DADOS - POSTGRES

Name	Storage Size	Description	Range
smallint	2 bytes	small-range integer	-32768 to +32767
integer	4 bytes	typical choice for integer	-2147483648 to +2147483647
bigint	8 bytes	large-range integer	-9223372036854775808 to 9223372036854775807
decimal / numeric	variable	user-specified precision, exact	up to 131072 digits before the decimal point; up to 16383 digits after the decimal point
real	4 bytes	variable-precision, inexact	6 decimal digits precision
double precision	8 bytes	variable-precision, inexact	15 decimal digits precision
serial	4 bytes	autoincrementing integer	1 to 2147483647
bigserial	8 bytes	large autoincrementing integer	1 to 9223372036854775807

TIPOS DE DADOS - POSTGRES

Name	Storage Size	Description
character varying(n), varchar(n)	variable(can store n chars)	variable-length with limit
character(n), char(n)	n chars	fixed-length, blank padded
text	variable	variable unlimited length
"char"	1 byte	single-byte internal type
name	64 bytes	internal type for object names

TIPOS DE DADOS - POSTGRES

Name	Storage Size	Description	Low Value	High Value	Resolution
timestamp [(p)] [without time zone]	8 bytes	both date and time (no time zone)	4713 BC	294276 AD	1 microsecond / 14 digits
timestamp [(p)] with time zone	8 bytes	both date and time, with time zone	4713 BC	294276 AD	1 microsecond / 14 digits
date	4 bytes	date (no time of day)	4713 BC	5874897 AD	1 day
time [(p)] [without time zone]	8 bytes	time of day (no date)	00:00:00	24:00:00	1 microsecond / 14 digits
time [(p)] with time zone	12 bytes	times of day only, with time zone	00:00:00+1459	24:00:00-1459	1 microsecond / 14 digits
interval [fields] [(p)]	12 bytes	time interval	-178000000 years	178000000 years	1 microsecond / 14 digits

CRIANDO TABELA

```
CREATE TABLE alunosBetha(
    i_alunos INTEGER,
    nome   VARCHAR(60)
);
```

ADICIONANDO DADOS

```
INSERT INTO alunosBetha (i_alunos, nome)
VALUES (1, 'Tiago da Rosa Valério');
```

DELETANDO DADOS

```
DELETE FROM alunosBetha;
```

EXCLUINDO DO POSTGRES

```
DROP TABLE alunosBetha;
```

COLUNA NOT NULL

```
CREATE TABLE alunosBetha(
i_alunos   INTEGER,
nome    VARCHAR(60) NOT NULL,
cidade  VARCHAR(60));
```

VALOR PADRÃO

```
CREATE TABLE alunosBetha(
i_alunos   INTEGER,
nome    VARCHAR(60) NOT NULL,
cidade  VARCHAR(60),
Estado VACHAR(2) DEFAULT 'SC' );
```

RESTRIÇÃO DE UNICIDADE

```
CREATE TABLE alunosBetha(
i_alunos   INTEGER,
nome    VARCHAR(60) NOT NULL UNIQUE,
cidade  VARCHAR(60),
Estado VACHAR(2) DEFAULT 'SC' );
```

RESTRIÇÃO DE VERIFICAÇÃO

```
CREATE TABLE alunosBetha(
i_alunos   INTEGER,
nome    VARCHAR(60) NOT NULL UNIQUE,
cidade  VARCHAR(60),
estado VARCHAR(2) DEFAULT 'SC',
ativo VARCHAR(1) CHECK(ATIVO in ('S','N')) );
```

CHAVE PRIMÁRIA

```
CREATE TABLE alunosBetha(
i_alunos   INTEGER,
nome    VARCHAR(60) NOT NULL UNIQUE,
cidade  VARCHAR(60),
PRIMARY KEY(i_alunos) );
```

EXEMPLO

PESSOAS

(pk)i_pessoas	serial,	
nome	varchar(60)	not null,
tipo_pessoa	varchar(1)	'F','J' valor padrão F
cpf_cnpj	varchar(14)	not null único,
email	varchar(100),	
telefone	varchar(14),	
celular	varchar(14)	

EXERCÍCIO

ATIVIDADES

(pk)i_atividades	serial,		
descricao	varchar(60)	not null	unico,
tipo	varchar(1)	not null	'A','S'
classificacao	varchar(1)	not null	'A','I','C','S'
aliquota	decimal(7,4)	not null	> 0,

CHAVE ESTRANGEIRA

```
CREATE TABLE responsaveis_alunosBetha(
i_responsaveis INTEGER,
i_alunos     INTEGER REFERENCES alunosBetha(i_alunos),
nome     VARCHAR(60) NOT NULL,
telefone  VARCHAR(14),
email  VARCHAR(60),
PRIMARY KEY(i_responsaveis) );
```

EXEMPLO

PESSOAS

(pk)i_pessoas	serial,	
nome	varchar(60)	not null,
tipo_pessoa	varchar(1)	'F','J' valor padrão F
cpf_cnpj	varchar(14)	not null único,
email	varchar(100),	
telefone	varchar(14),	
celular	varchar(14)	

ENDERECOS_PESSOAS

(pk)i_pessoas	serial	fk de pessoas,
(pk)tipo_endereco	varchar(1)	'P','S','C' valor padrão P
bairro	varchar(60),	
rua	varchar(60),	
numero	varchar(10,	
cidade	varchar(60),	
uf	varchar(2),	valor padrão SC

EXERCÍCIO

ATIVIDADES

(pk)i_atividades	serial,		
descricao	varchar(60)	not null	unico,
tipo	varchar(1)	not null	'A','S'
classificacao	varchar(1)	not null	'A','I','C','S'
aliquota	decimal(7,4)	not null	> 0,

EMPRESAS

(pk)i_empresas	serial,		
nome	varchar(100)	not null,	
nome_fantasia	varchar(100)	not null,	
funcionarios	integer	> 0	
(fk)i_atividades	integer	Fk de Atividades,	

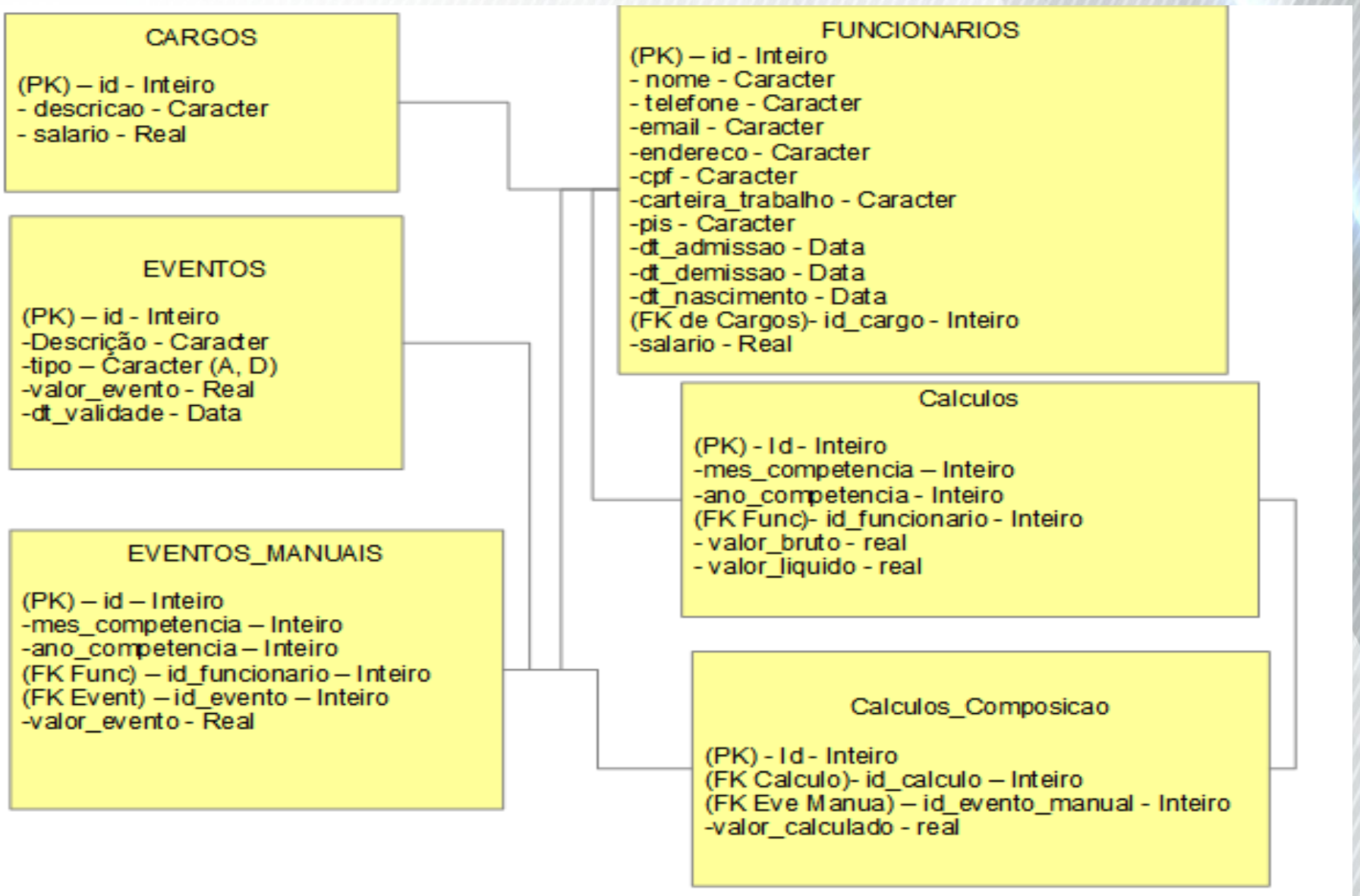
ALTERANDO TABELAS

```
ALTER TABLE alunosBetha ADD estado VARCHAR(2);

ALTER TABLE alunosBetha DROP estado;

ALTER TABLE alunosBetha
ALTER COLUMN cidade TYPE VARCHAR(100);
```

Exercício – Criação de Estrutura



Exercício – Tarefa

Criar Database/Estrutura no modelo criado no trabalho.