

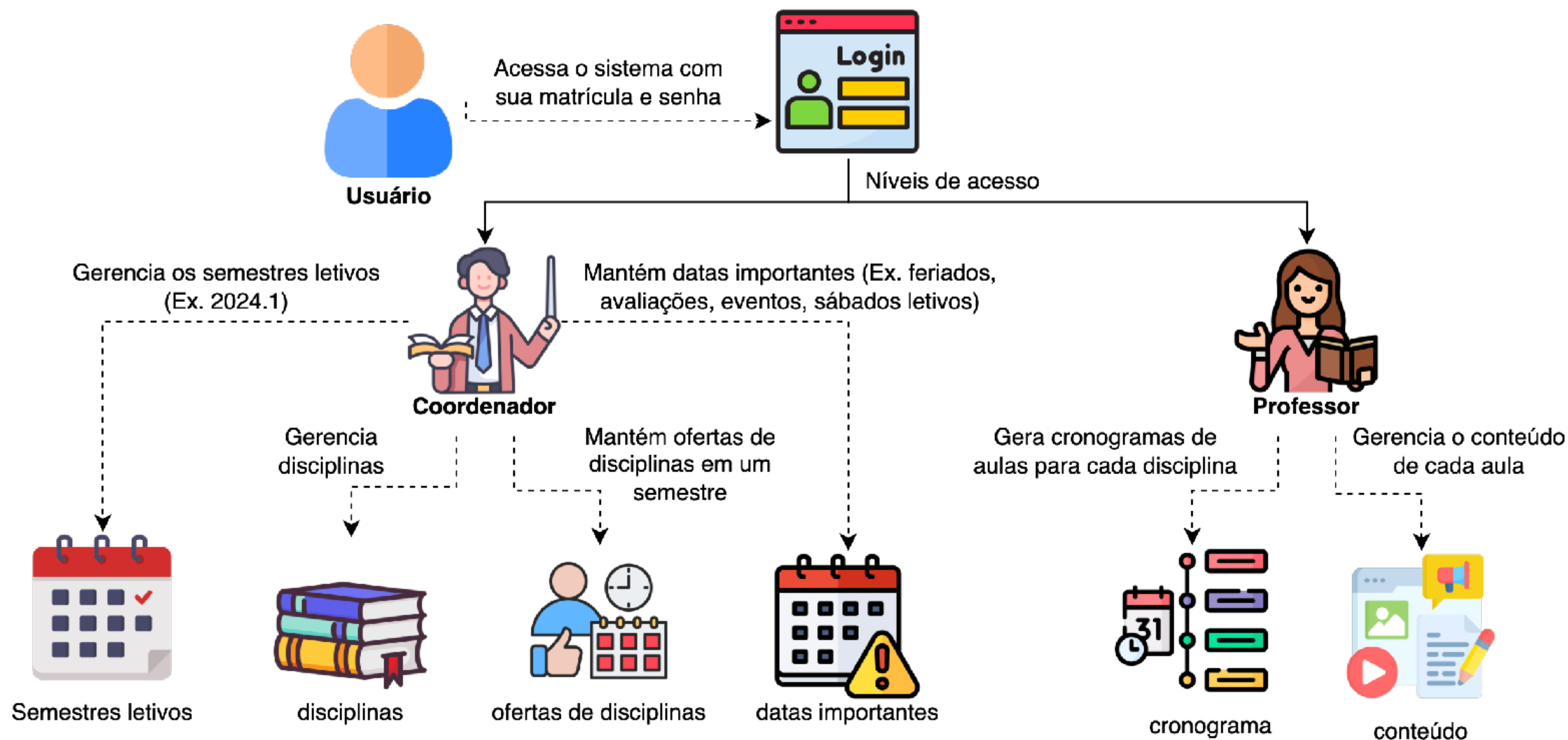
# BANCO DE DADOS II

## CRIAÇÃO DE BANCO DE DADOS E TABELAS

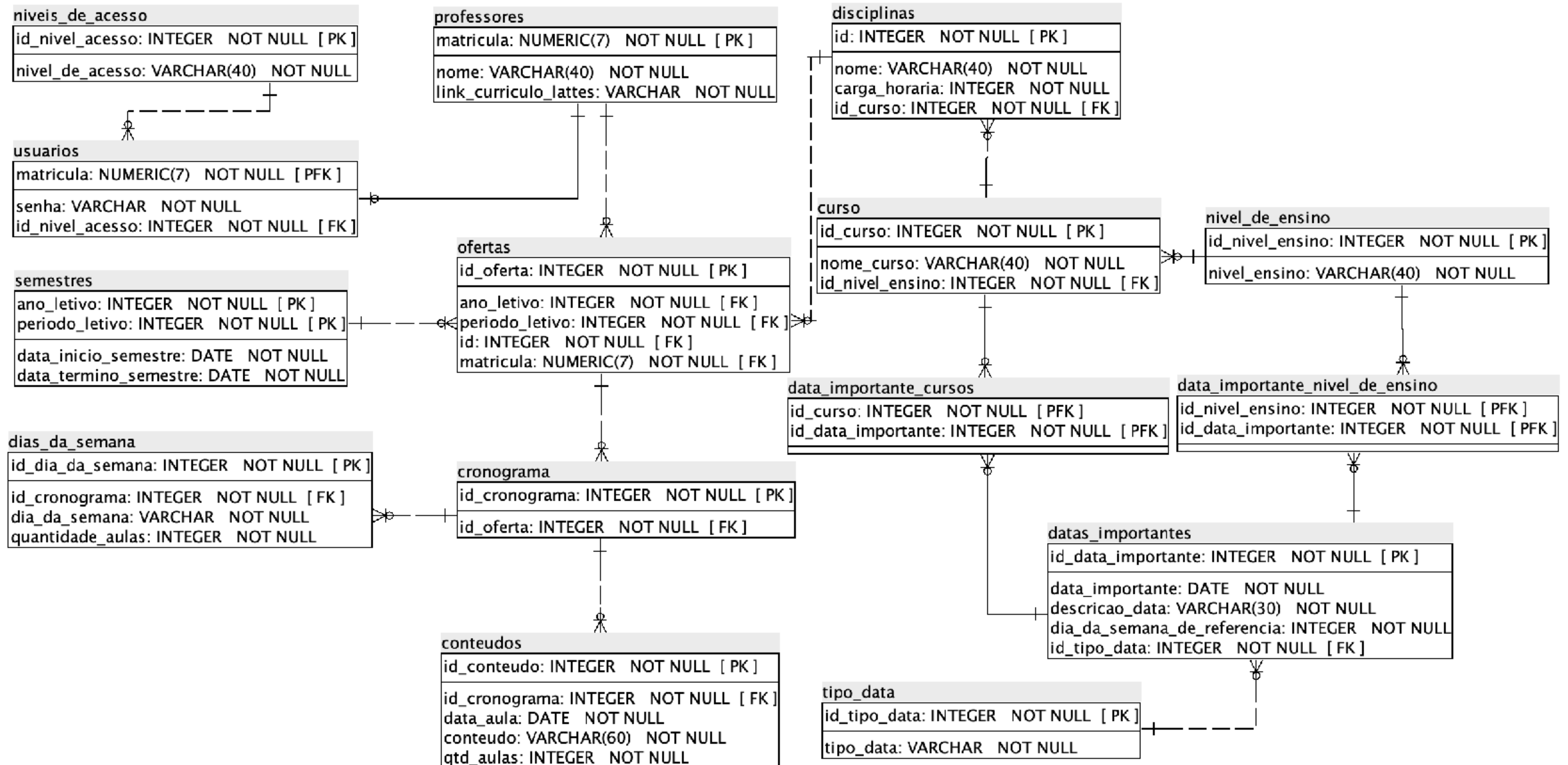


*Prof. Rangel Nunes*

# O projeto...



# Proposta de DE-R para o projeto



# CRIAÇÃO DO BANCO DE DADOS

---

*Comandos DDL*



</SQL CODE >

CREATE DATABASE NOME;

</SQL CODE >

## ALGUNS PARÂMETROS DO COMANDO CREATE DATABASE...

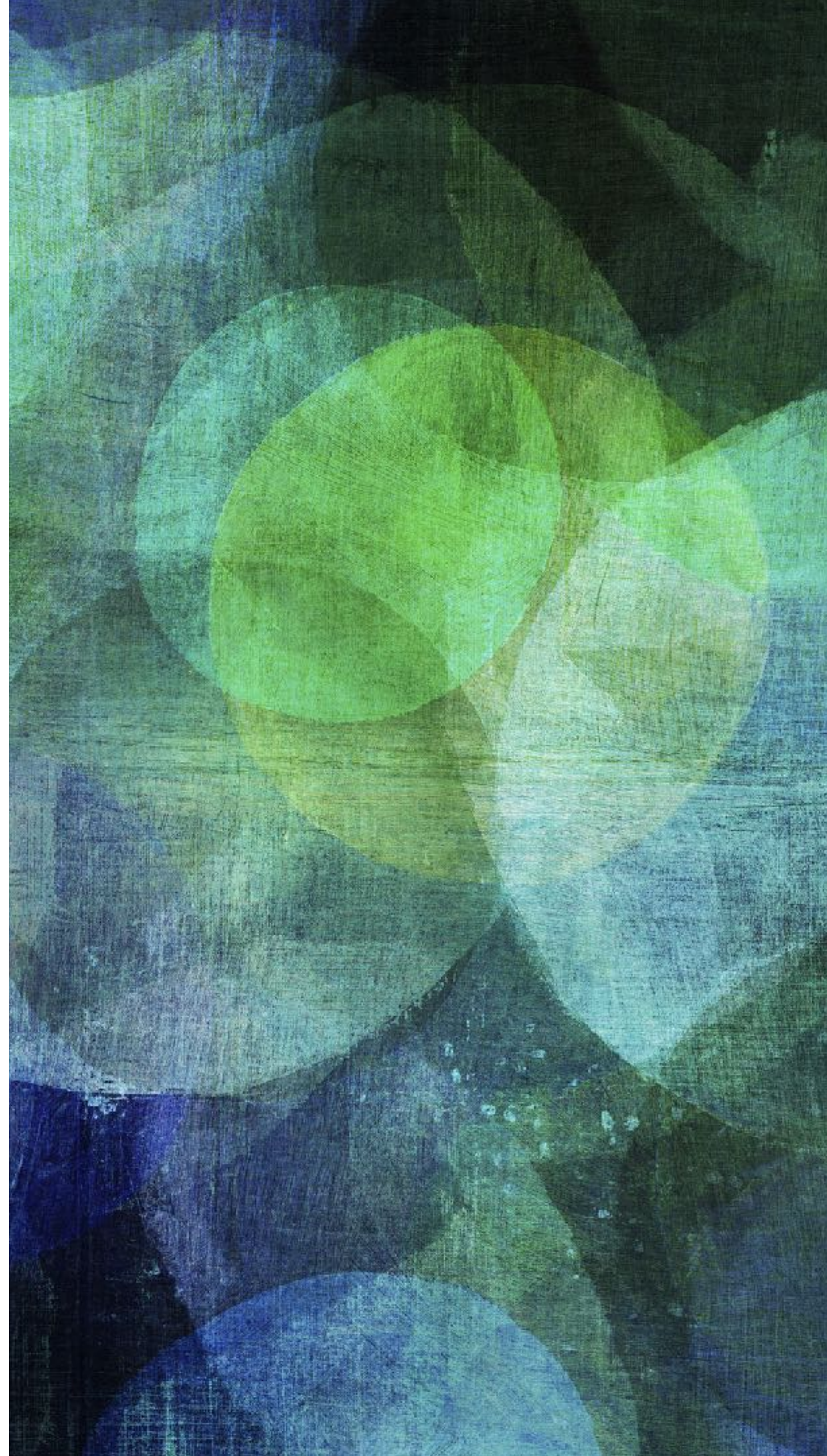


```
CREATE DATABASE database_name
WITH
    [OWNER = role_name]
    [TEMPLATE = template]
    [ENCODING = encoding]
    [LC_COLLATE = collate]
    [LC_CTYPE = ctype]
    [TABLESPACE = tablespace_name]
    [ALLOW_CONNECTIONS = true | false]
    [CONNECTION LIMIT = max_concurrent_connection]
    [IS_TEMPLATE = true | false ];
```

# CRIANDO TABELAS

---

*Mais comandos da DDL*



# SINTAXE DO COMANDO CREATE TABLE

---

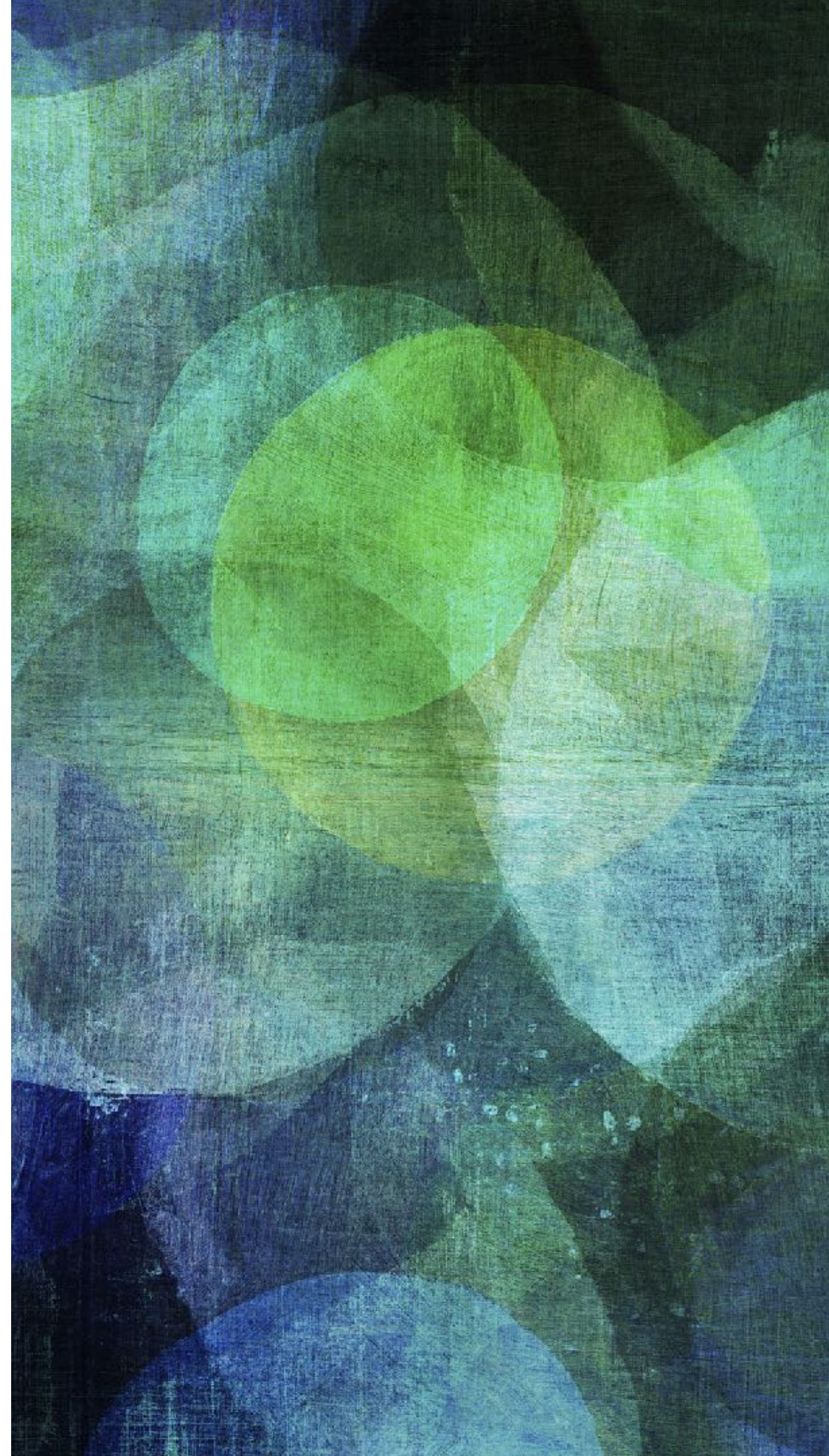
```
create table nome_da_tabela (  
    atributo1 tipo_de_dado restrição_se_tiver,  
    atributo2 tipo_de_dado restrição_se_tiver  
);
```



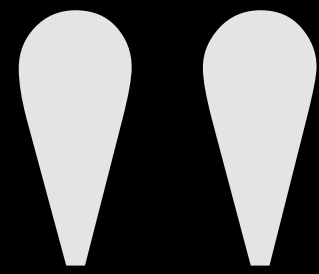
```
create table produto (  
  codigo_produto integer,  
  nome_produto varchar(30)  
);
```

# CRIANDO TABELAS COM RESTRICÇÕES

*Constraints SQL*

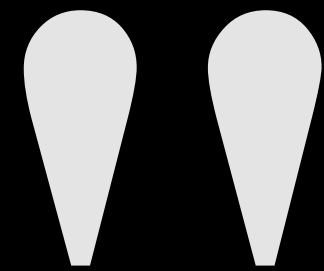


# Qual tipo de dados usar?



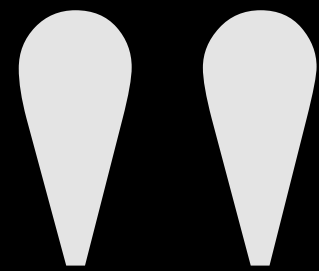
```
CREATE TABLE niveis_de_acesso (  
    id_nivel_acesso ??? primary key,  
    nivel_de_acesso VARCHAR(40) NOT NULL  
);
```

# Usando o pseudotipo **Serial**



Name	Storage Size	Range
SMALLSERIAL	2 bytes	1 to 32,767
SERIAL	4 bytes	1 to 2,147,483,647
BIGSERIAL	8 bytes	1 to 9,223,372,036,854,775,807

# Usando o pseudotipo **Serial**



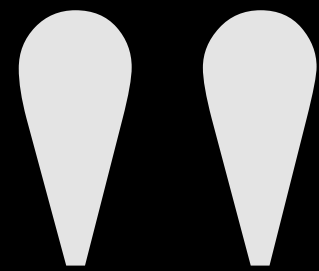
```
CREATE TABLE table_name(  
    id SERIAL  
);
```

O que é equivalente a:

```
CREATE SEQUENCE table_name_id_seq;  
  
CREATE TABLE table_name (  
    id integer NOT NULL DEFAULT nextval('table_name_id_seq')  
);  
  
ALTER SEQUENCE table_name_id_seq  
OWNED BY table_name.id;
```

# Usando o UUID

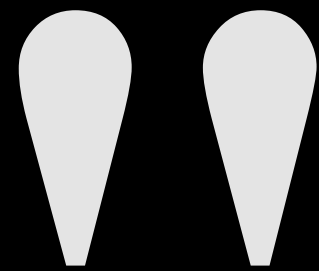
*Universal Unique Identifier*



```
CREATE EXTENSION pgcrypto;
```

# Usando o UUID

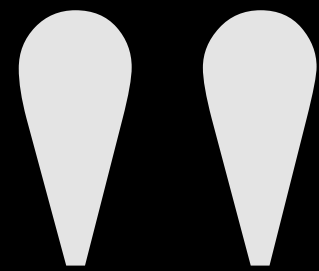
*Universal Unique Identifier*



```
SELECT gen_random_uuid();
```

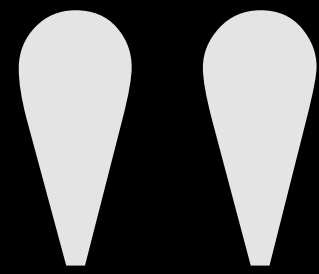
# Usando o UUID

*Universal Unique Identifier*



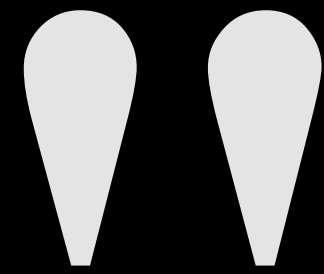
```
CREATE TABLE table_name (  
    table_id uuid DEFAULT gen_random_uuid(),  
    first_name VARCHAR NOT NULL,  
    last_name VARCHAR NOT NULL,  
    PRIMARY KEY (table_id)  
);
```

# Nossa tabela pode ficar assim



```
CREATE TABLE niveis_de_acesso (  
    id_nivel_acesso smallserial primary key,  
    nivel_de_acesso VARCHAR(40) NOT NULL  
);
```

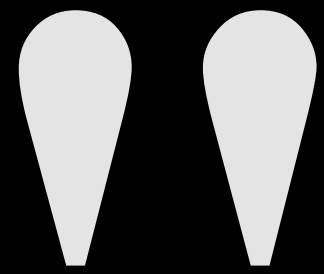
# Criando chaves estrangeiras



## DELETE CASCADE

```
CREATE TABLE parent_table(  
    id SERIAL PRIMARY KEY,  
    ...  
);  
  
CREATE TABLE child_table(  
    id SERIAL PRIMARY KEY,  
    parent_id INT,  
    FOREIGN_KEY(parent_id)  
        REFERENCES parent_table(id)  
        ON DELETE CASCADE  
);
```

# Nossa tabela **usuários** ficará assim



```
create table usuarios(  
  matricula numeric(7) not null,  
  senha text not null,  
  id_nivel_acesso integer not null,  
  constraint professor_fk foreign key (matricula)  
  references professores(matricula) on delete cascade,  
  constraint usuarios_pk primary key(matricula)  
);
```