



UNITAU
Universidade de Taubaté





SQL Introdução

The background features abstract blue and white geometric shapes, including a large blue sphere and several white spheres, set against a black background.

Linguagem SQL



Structured Query Language, ou Linguagem de Consulta Estruturada ou SQL, é uma linguagem de pesquisa declarativa para banco de dados relacional. Muitas das características originais do SQL foram inspiradas na álgebra relacional.

A linguagem SQL é um grande padrão de banco de dados. Isto decorre da sua simplicidade e facilidade de uso. Ela se diferencia de outras linguagens de consulta a banco de dados no sentido em que uma consulta SQL especifica a forma do resultado e não o caminho para chegar a ele.

A linguagem SQL foi criada nos anos 70, inicialmente com o nome SEQUEL. A partir de 1986 o ANSI e ISO adotaram um padrão para a linguagem SQL definindo o SQL-86, revisões posteriores da linguagem definiram os padrões SQL-89, SQL-92, SQL:1999 (SQL3), SQL:2003, SQL:2006, SQL:2008, SQL:2011 e mais recentemente SQL:2016.

Linguagem SQL



A linguagem SQL pode ser dividida em 3 grupos de instruções:

Linguagem de Definição de Dados: a **DDL** (Data Definition Language) é utilizada para definição da estrutura da base de dados, a DDL atua sobre o dicionário de dados da base de dados.

Linguagem de Manipulação de Dados: a **DML** (Data Manipulation Language) é utilizada para recuperação, inclusão, alteração e deleção dos dados da base de dados. Pode ser procedural, que especifica como os dados devem ser obtidos do banco ou pode também ser declarativa (não procedural), em que os usuários não necessitam especificar como os dados serão obtidos.

Linguagem de Controle de Dados: a **DCL** (Data Control Language) é utilizada para controlar o acesso aos dados da base de dados.

CREATE USER



O comando **CREATE USER** adiciona um novo usuário ao agrupamento de bancos de dados do PostgreSQL. Apenas os superusuários do banco de dados podem usar este comando.

```
CREATE USER nome [ [ WITH ] opção [ ... ] ]
```

Algumas opções disponíveis:

```
SUPERUSER | NOSUPERUSER  
| CREATEDB | NOCREATEDB  
| CREATEUSER | NOCREATEUSER  
| LOGIN | NOLOGIN  
| [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'senha'
```

Exemplo:

```
CREATE USER aluno WITH PASSWORD 'aluno';
```


CREATE DATABASE



O comando **CREATE DATABASE** cria um banco de dados no PostgreSQL. Para poder criar um banco de dados é necessário ser um superusuário ou possuir o privilégio especial **CREATEDB**.

CREATE DATABASE nome

```
[ [ WITH ] [ OWNER [=] dono_do_banco_de_dados ]  
  [ TEMPLATE [=] modelo ]  
  [ ENCODING [=] codificação ]  
  [ TABLESPACE [=] espaço_de_tabelas ]  
  [ CONNECTION LIMIT [=] connlimit ] ]
```

Normalmente, o criador se torna o dono do novo banco de dados. Por padrão, o novo banco de dados é criado clonando o banco de dados padrão do sistema template1.

Exemplo:

```
CREATE DATABASE aluno OWNER aluno;
```

CREATE TABLE



O comando **CREATE TABLE** cria uma tabela, inicialmente vazia, no banco de dados corrente. O usuário que executa o comando se torna o dono da tabela.

```
CREATE [ [ GLOBAL | LOCAL ] { TEMPORARY | TEMP } ] TABLE [ IF NOT
EXISTS ] nome_tabela
(
    { nome_coluna tipo_dado [ DEFAULT expressão_padrão ]
      [ restrição_coluna [ ... ] ]
      | restrição_de_tabela
      | LIKE tabela_ancestral [ { INCLUDING | EXCLUDING }
DEFAULTS ] }
    [, ... ]
)
[ INHERITS ( tabela_ancestral [, ... ] ) ]
[ WITH ( storage_parameter [ = value ] [,...] ) | WITH OIDS |
WITHOUT OIDS ]
[ ON COMMIT { PRESERVE ROWS | DELETE ROWS | DROP } ]
[ TABLESPACE espaço_de_tabelas ]
```


CREATE TABLE



Exemplo:

```
CREATE TABLE contato (  
    nome          VARCHAR(50) PRIMARY KEY,  
    telefone     VARCHAR(15),  
    email        VARCHAR(50)  
    );
```


Tipos de Dados



Alguns dos tipos de dados da linguagem SQL são:

Numérico: smallint (int2), integer (int, int4) , bigint (int8), numeric (decimal), real (float4), double precision (float8), serial (serial4), bigserial (serial8)

Caractere: character varying (varchar), character (char), text

Binário: bytea

Data e Hora: timestamp [without time zone], timestamp with time zone (timestamptz), interval, date, time [without time zone], time with time zone (timetz)

Lógico: boolean (bool)

Restrições de Coluna



```
[ CONSTRAINT nome_da_restrição ]
{ NOT NULL |
  NULL |
  CHECK ( expression ) [ NO INHERIT ] |
  DEFAULT default_expr |
  GENERATED ALWAYS AS ( generation_expr ) STORED |
  GENERATED { ALWAYS | BY DEFAULT } AS IDENTITY
  [ ( sequence_options ) ] |
  UNIQUE [ NULLS [ NOT ] DISTINCT ] index_parameters |
  PRIMARY KEY index_parameters |
  REFERENCES tabela_referenciada
  [ ( coluna_referenciada ) ]
  [ MATCH FULL | MATCH PARTIAL | MATCH SIMPLE ]
  [ ON DELETE ação ] [ ON UPDATE ação ] }
[ DEFERRABLE | NOT DEFERRABLE ]
[ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]
```


Restrições de Tabela



```
[ CONSTRAINT nome_da_restrição ]  
{ CHECK ( expression ) [ NO INHERIT ] |  
  UNIQUE [ NULLS [ NOT ] DISTINCT ] ( column_name  
    [, ... ] ) index_parameters |  
  PRIMARY KEY ( column_name [, ... ] ) index_parameters |  
  FOREIGN KEY ( column_name [, ... ] )  
    REFERENCES reftable [ ( refcolumn [, ... ] ) ]  
    [ MATCH FULL | MATCH PARTIAL | MATCH SIMPLE ]  
    [ ON DELETE action ] [ ON UPDATE action ] }  
[ DEFERRABLE | NOT DEFERRABLE ]  
[ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]
```


CREATE TABLE



Exemplo:

```
CREATE TABLE produto ( codigo      CHAR(2)  PRIMARY KEY,
                        descricao    CHAR(10) NOT NULL );
CREATE TABLE entrada ( codigo      INTEGER  PRIMARY KEY
                        GENERATED ALWAYS AS IDENTITY,
                        data          DATE     NOT NULL,
                        produto       CHAR(2)  NOT NULL
                        REFERENCES produto,
                        quantidade    INTEGER );
CREATE TABLE saida (   codigo      INTEGER  PRIMARY KEY
                        GENERATED BY DEFAULT AS IDENTITY,
                        data          DATE     NOT NULL,
                        produto       CHAR(2)  NOT NULL
                        REFERENCES produto,
                        quantidade    INTEGER );
```


CREATE INDEX



O comando **CREATE INDEX** constrói um índice na tabela especificada. O PostgreSQL fornece os métodos de índice btree, hash, gist, spgist e gin. O método padrão é o btree.

```
CREATE [ UNIQUE ] INDEX nome_do_índice ON tabela
    [ USING método ]
    ( { coluna | ( expressão ) } [ classe_de_operadores ]
      [, ...] )
    [ ASC | DESC ] [ NULLS { FIRST | LAST } ]
    [ TABLESPACE espaço_de_tabelas ]
    [ WHERE predicado ]
```

Exemplo:

```
CREATE INDEX aluno_nome ON aluno ( nome );
```


Exercícios



Monte o script para criar as seguintes tabelas:

a) Diretoria(codigo, descricao)

- codigo - 3 caracteres
- descricao - 10 caracteres, obrigatório

b) Secao(codigo, descricao, diretoria)

- codigo - 3 caracteres
- descricao - 10 caracteres, obrigatório
- diretoria - chave estrangeira para a tabela de diretorias, obrigatorio

c) Funcao(funcao, descricao)

- funcao - campo com auto incremento
- descricao - 10 caracteres, obrigatório

d) Funcionario(matricula, nome, nascimento, salario, funcao, secao)

- matricula - 5 caracteres
- nome - 15 caracteres, obrigatório
- nascimento - data
- salario - numérico, 7 casas com 2 decimais
- funcao - chave estrangeira para a tabela de funções
- secao - chave estrangeira para a tabela de secoes

Exercícios



e) Banco(codigo, nome)

- codigo - 3 caracteres
- nome - 10 caracteres

f) Conta(codigo, banco, agencia, numero, descricao)

- codigo - 2 caracteres
- banco - chave estrangeira para a tabela de bancos
- agencia - 4 caracteres
- numero - 5 caracteres
- descricao - 10 caracteres, obrigatório

g) GrupoTipo(codigo, descricao)

- codigo - 1 caracter
- descricao - 10 caracteres, obrigatório

h) Grupo(codigo, descricao, tipo)

- codigo - 2 caracteres
- descricao - 15 caracteres
- tipo - chave estrangeira para a tabela grupotipo

i) Lancamento(codigo, conta, grupo, data, valor)

- codigo - campo com auto incremento
- conta - chave estrangeira para a tabela de contas
- grupo - chave estrangeira para a tabela de grupos
- data - data
- valor - numérico com 6 posições e 2 casas decimais

Exercícios



j) Projeto(codigo, descricao, supervisor, horas_supervisao, custo_hora)

- codigo - 2 caracteres
- descricao - 25 caracteres
- supervisor - 10 caracteres
- horas_supervisao - inteiro
- custo_hora - numérico com 5 posições e 2 casas decimais

k) Fase(codigo, descricao)

- codigo - 1 caracteres
- descricao - 15 caracteres, obrigatório

l) Plataforma(codigo, descricao)

- codigo - 1 caracter
- descricao - 10 caracteres, obrigatório

m) Participante(projeto, participante, fase, plataforma, inicio, final, horas, custo_hora)

- projeto - chave estrangeira para a tabela de projetos
- participante - 10 caracteres
- fase - chave estrangeira para a tabela de fases, obrigatório
- plataforma - chave estrangeira para a tabela de plataformas
- inicio - data
- final - data
- horas - inteiro
- custo_hora - numérico com 5 posições e 2 casas decimais

Exercícios



n) Fabricante(codigo, nome)

- codigo - 2 caracteres
- nome - 10 caracteres

o) Revenda(codigo, nome, cidade, estado)

- codigo - 2 caracteres
- nome - 15 caracteres
- cidade - 10 caracteres
- estado - 2 caracteres

p) Cliente(codigo, nome, sobrenome)

- codigo - 2 caracteres
- nome - 10 caracteres
- sobrenome - 10 caracteres

q) Automovel(codigo, fabricante, modelo, ano, pais, preco)

- codigo - 2 caracteres
- fabricante - chave estrangeira para a tabela de fabricantes
- modelo - 15 caracteres
- cor - 10 caracteres
- ano - 4 caracteres
- pais - 9 caracteres
- compra - data
- preco - numérico com 7 posições e 2 casas decimais

Exercícios



- r) Venda(cliente, revenda, automovel, data, valor)
- cliente - chave estrangeira para a tabela de clientes
 - revenda - chave estrangeira para a tabela de revendas
 - automovel - chave estrangeira para a tabela de automoveis
 - data - data
 - valor - numérico com 7 posições e 2 casas decimais

Exercícios



s) Organizacao(nome, vendas, superior)

- **nome - 15 caracteres**
- **vendas - numérico com 9 posições e 2 casas decimais**
- **superior - chave estrangeira para a tabela organizacao**