

Criando Bancos e Tabelas no PostgreSQL

16

Aula 4

...Introdução a linguagem
SQL



DDL, DML, DCL

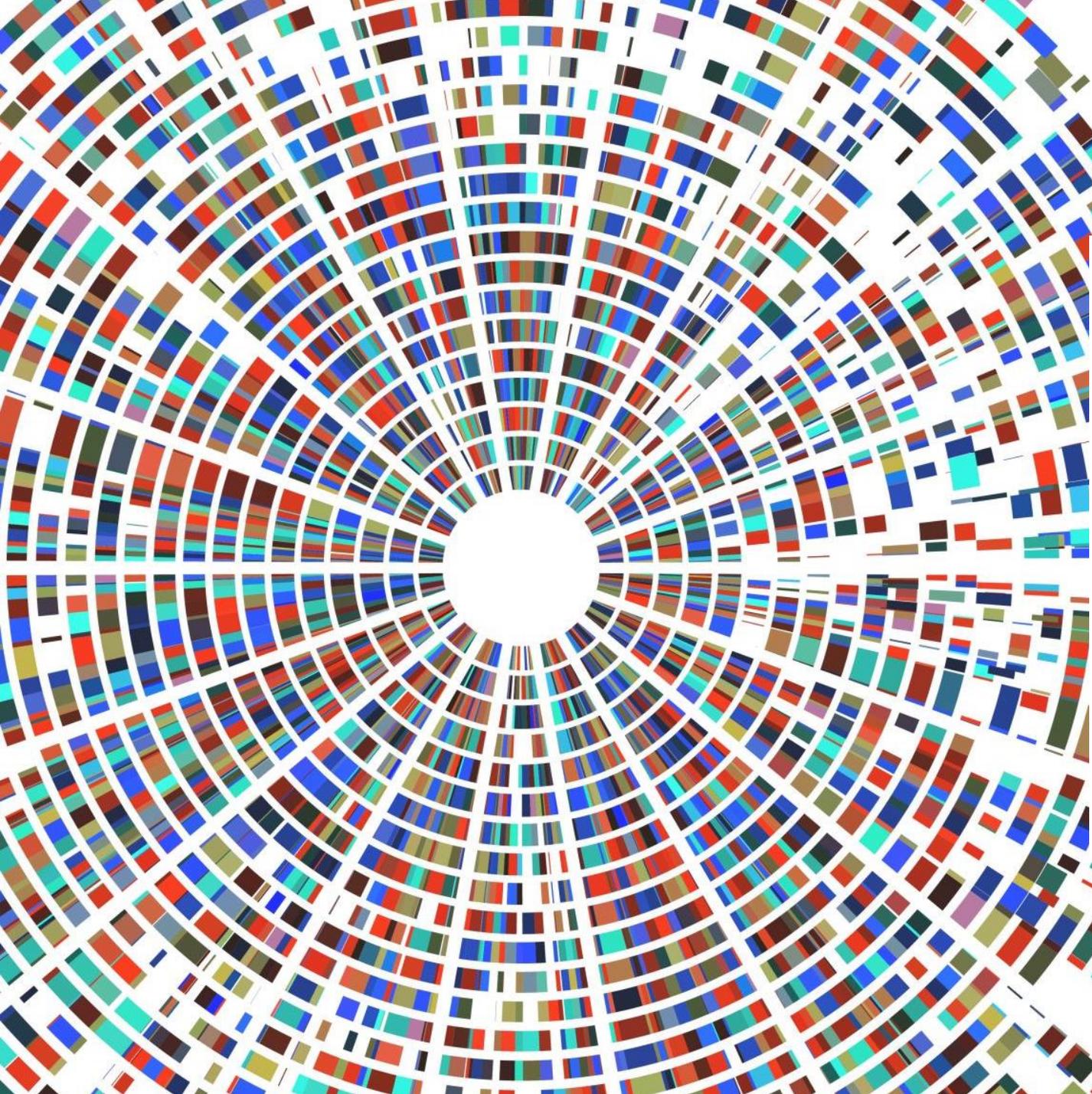
SQL

Linguagem de Definição de Dados - DDL

Objetos
fundamentais
da
linguagem:

DATABASE
USER
ROLE
SCHEMA
TABLESPACE
TABLE
INDEX
DOMAIN
FUNCTION
SEQUENCE
TRIGGER
VIEW

Todos os objetos podem ser criados (CREATE), corrigidos (ALTER) e removidos (DROP), cada um com sintaxe própria.



Recordando



DDL - Comando CREATE TABLE

Criar uma Tabela no Esquema da Aplicação

Sintaxe geral de
um Comando
CREATE TABLE

- CREATE TABLE <nome da tabela> (
 - <definição de Coluna>,...
 - <Restrições de Integridade>,...
 -);

onde <definição
de Coluna> pode
ser:

- <nome atr> <tipo de dado> [NULL | NOT NULL]
- [DEFAULT <value>] — DEFAULT NULL

DDL - Comando CREATE TABLE

As
constraints
possíveis
para
colunas
são:

- **NOT NULL**
- **NULL (padrão)**
- **CHECK**
- **DEFAULT**
- **UNIQUE**
- **PRIMARY KEY**
- **REFERENCES**

DDL - Comando CREATE TABLE

onde
<ações>
pode
ser:

- **ON UPDATE**
- **DELETE CASCADE**
- **RESTRICT**
- **SET NULL**
- **SET DEFAULT**
- **NO ACTION**

Restrições (constraints) de integridade referencial

As restrições de integridade referencial são usadas para garantir a consistência e integridade dos dados em um banco de dados PostgreSQL.

Elas definem regras que impõem relacionamentos entre tabelas, garantindo que os valores das chaves estrangeiras estejam sempre em conformidade com os valores das chaves primárias a que se referem.

Restrições de integridade referencial: Chave primária e chave estrangeira

Antes de entendermos as restrições de integridade referencial, é importante entender o conceito de chave primária e chave estrangeira.

Chave primária

Chave primária:

- Uma chave primária é uma coluna ou conjunto de colunas que identifica exclusivamente cada registro em uma tabela.
- Ela garante a unicidade dos registros e é usada para estabelecer relacionamentos com outras tabelas.

Chave primária

Chave estrangeira:

- Uma chave estrangeira é uma coluna ou conjunto de colunas em uma tabela que faz referência à chave primária de outra tabela.
- Ela estabelece um relacionamento entre duas tabelas, permitindo que os dados estejam relacionados e que as operações de integridade referencial sejam aplicadas.

Criando restrições de integridade referencial

No PostgreSQL, as restrições de integridade referencial podem ser criadas durante a definição das colunas da tabela usando a cláusula **REFERENCES**.

Vamos ver um exemplo:

```
CREATE TABLE Clientes.pedidos (  
    id SERIAL PRIMARY KEY,  
    id_FK INTEGER REFERENCES Clientes.dados(ID),  
    valor NUMERIC(10,2)  
);
```

- ▼ Clientes
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > Aa FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > 1..3 Sequences
 - ▼ Tables (2)
 - > dados
 - ▼ pedidos
 - > Columns
 - ▼ Constraints (2)
 - pedidos_id_fk_fkey
 - pedidos_pkey
 - > Indexes
 - RI S Policies

Exemplo1/postgres@PostgreSQL 16



Query Query History

```
1 CREATE TABLE Clientes.pedidos (  
2     id SERIAL PRIMARY KEY,  
3     id_FK INTEGER REFERENCES Clientes.dados(ID),  
4     valor NUMERIC(10,2)  
5 );  
6
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 45 msec.

Alterando ou excluindo restrições de integridade referencial

É possível alterar ou excluir as restrições de integridade referencial existentes em uma tabela.

Para isso, utilizamos os comandos **ALTER TABLE** e **DROP CONSTRAINT**.

Por exemplo, para alterar uma restrição de integridade referencial, podemos renomeá-la.

- >  Functions
- >  Materialized Views
- >  Operators
- >  Procedures
- > 1.3 Sequences
- ▼  Tables (2)
 - >  dados
 - ▼  pedidos
 - >  Columns
 - ▼  Constraints (2)
 -  pedidos_cliente_id_fkey
 -  pedidos_pkey

```
6 ALTER TABLE Clientes.pedidos
7   RENAME CONSTRAINT pedidos_id_fk_fkey TO pedidos_cliente_id_fkey;
```

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 47 msec.

Índices

Índices

Os índices são estruturas utilizadas para otimizar as consultas em bancos de dados PostgreSQL.

Eles permitem um acesso mais rápido e eficiente aos dados, especialmente em tabelas grandes com muitos registros.

O que são índices?

Um índice é uma estrutura de dados que melhora a velocidade de busca e recuperação de registros em uma tabela.

Ele funciona como um "índice" de um livro, onde você pode encontrar rapidamente informações específicas com base em palavras-chave.

Da mesma forma, um índice em um banco de dados ajuda a localizar registros com base em valores de colunas específicas.

O que são índices?

Os índices desempenham um papel crucial na otimização do desempenho das consultas em bancos de dados PostgreSQL.

Aqui estão algumas razões pelas quais os índices são importantes:

Melhor desempenho das consultas:

- Os índices permitem que o banco de dados localize rapidamente os registros que correspondem aos critérios de busca, resultando em consultas mais rápidas e eficientes.

Redução da carga de trabalho do servidor:

- Com índices adequados, o banco de dados pode evitar a necessidade de ler todas as linhas de uma tabela para executar uma consulta. Isso reduz a carga de trabalho do servidor e melhora o tempo de resposta.

Aqui estão algumas razões pelas quais os índices são importantes:

Suporte a restrições de chave única:

- Os índices podem ser usados para garantir a unicidade dos valores em uma coluna ou conjunto de colunas, permitindo a definição de restrições de chave única.

Ordenação eficiente:

Os índices permitem que os dados sejam armazenados de forma ordenada, facilitando a execução de consultas com base em ordem crescente ou decrescente.

Melhor desempenho de junções:

Os índices também podem melhorar o desempenho de operações de junção entre tabelas, ajudando o banco de dados a encontrar rapidamente os registros relacionados.

Tipos de índices no PostgreSQL

Índice B-Tree:

- É o tipo de índice mais comum no PostgreSQL. Ele funciona bem para colunas com valores repetidos e permite a busca rápida e eficiente.

Índice Hash:

- É adequado para consultas de igualdade (exatamente igual) em colunas. É mais eficiente quando os valores de coluna são distribuídos uniformemente.

Índice GiST (Generalized Search Tree):

- É um tipo de índice genérico que suporta vários tipos de dados e operadores. É útil para consultas espaciais, pesquisa de texto e muito mais.

Índice GIN (Generalized Inverted Index):

- É otimizado para consultas com base em listas de valores. É usado principalmente para pesquisa de texto e pesquisa de listas.

Criando e gerenciando índices

No PostgreSQL, os índices podem ser criados usando o comando CREATE INDEX.

Aqui está a sintaxe básica do comando para criar um índice:

```
CREATE INDEX <nome_do_indice> ON <nome_da_tabela> (coluna1, coluna2, ...);
```

Criando e gerenciando índices

Substitua **<nome_do_indice>** pelo nome que você deseja dar ao índice e **<nome_da_tabela>** pelo nome da tabela na qual você deseja criar o índice.

Em seguida, especifique as colunas que você deseja incluir no índice, separadas por vírgula.

Criando e gerenciando índices

- Por exemplo, vamos criar um índice na tabela "Clientes.dados" para a coluna "primeiro_nome":

```
CREATE INDEX nome_idx ON  
Clientes.dados(primeiro_nome);
```

- > FTS Templates
- > Foreign Tables
- > Functions
- > Materialized Views
- > Operators
- > Procedures
- > 1.3 Sequences
- ▼ Tables (2)
 - ▼ dados
 - ▼ Columns (4)
 - id
 - primeiro_nome
 - sobrenome
 - telefone
 - > Constraints
 - ▼ Indexes (1)
 - nome_idx
 - > RLS Policies
 - > Rules
 - > Triggers
 - > pedidos
- > Trigger Functions
- > Types
- > Views

Query Query History

```
1 CREATE INDEX nome_idx ON clientes.dados (primeiro_nome);  
2
```

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 31 msec.

Popular a base com Dados

Exemplo CVS

https://dados.agricultura.gov.br/sv/dataset

The screenshot shows the web interface of the Portal de Dados Abertos do Ministério da Agricultura e Pecuária. The browser's address bar displays the URL 'dados.agricultura.gov.br/sv/dataset'. The page features a green header with the title 'Portal de Dados Abertos do Ministério da Agricultura e Pecuária' and a 'Logga in' link. Below the header is a navigation menu with 'Dataset', 'Organisationer', 'Grupper', and 'Om' tabs, and a search bar containing the text 'Sök'. The main content area is titled '/ Dataset' and includes a sidebar with filters for 'Organisationer' (Ministério da Agric... (23), Serviço Florestal B... (4)), 'Grupper' (Biblioteca Nacional... (1)), and 'Taggar' (aquicultura (4), AGROECOLOGIA (3), CERTIFICAÇÃO ORGÂNICA (3), SUSTENTABILIDADE (3), Agricultura Familiar (2), pgi (2)). The main content area has a search bar 'Sök dataset...' and a dropdown menu 'Sortera på: Relevans'. Three datasets are listed: 'Sistema de Agrotóxicos Fitossanitários - Agrofit' (with a .CSV tag), 'SISZARC - Sistemas de Zoneamento Agrícola e Risco Climático' (with a CSV tag), and 'Selo Nacional da Agricultura Familiar'.

Exemplo agronegócios

CREATE SCHEMA agronegocios;

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows a tree view of the PostgreSQL 16 instance. Under the 'Databases (4)' folder, the 'Exemplo1' database is expanded, and the 'Schemas (3)' folder is selected. The 'agronegocios' schema is highlighted in blue. The right pane shows the 'Query' editor with the following SQL code:

```
1 CREATE SCHEMA agronegocios;  
2 CREATE TABLE agrofitprodutosformulados (  
3 )
```

Below the query editor, the 'Messages' tab is active, displaying the following output:

```
CREATE SCHEMA  
  
Query returned successfully in 50 msec.
```

Arquivo baixado: X:\2024-1\BD

Para uma • meninas | sumario 1 | O Conceito de | rrr • Sem título • rrr • Diagnóstico BB | agrofita | × +

Arquivo Editar Exibir



NR_REGISTRO;MARCA_COMERCIAL;FORMULACAO;INGREDIENTE_ATIVO;TITULAR_DE_REGISTRO;CLASSE;MODO_DE_ACAO;CULTURA;PRAGA_NOME_CIENTIFICO;PRAGA_NOME_COMUM;EMPRESA_PAIS_TIPO;CLASSE_TOXICOLOGICA;CLASSE_AMBIENTAL;ORGANICOS;SITUACAO

35523; KBR-829M1-02;Nematóides vivos;Heterorhabditis bacteriophora (Nematóides entomopatogênicos) (840 g/kg);Koppert do Brasil Holding S.A. - Piracicaba/SP ;Agente Biológico de Controle;;Todas as culturas;Scaptocoris castanea;Percevejo-castanho; (Koppert do Brasil Holding S.A. - Piracicaba/SP <BRASIL>FORMULADOR) + (Koppert do Brasil Holding S.A. - Piracicaba/SP <BRASIL>MANIPULADOR) + (Koppert do Brasil Holding Ltda.- Charqueada/SP<BRASIL>FORMULADOR) + (Koppert do Brasil Holding Ltda.- Charqueada/SP<BRASIL>MANIPULADOR) + (Koppert do Brasil Macrobiologicos Ltda ☒ Charqueada/SP<BRASIL>FORMULADOR) + (Koppert do Brasil Macrobiologicos Ltda ☒ Charqueada/SP<BRASIL>MANIPULADOR) + (Koppert BV - Holanda<PAÍSES BAIXOS, HOLANDA>FORMULADOR) + (Koppert Biological Systems, Inc.<ESTADOS UNIDOS>FORMULADOR) + (Koppert (Beijing) Agriculture Co., Ltd<CHINA, REPUBLICA POPULAR>FORMULADOR) + (Koppert México S.A. de C.V.<MÉXICO>FORMULADOR) + (Koppert Africa do Sul<ÁFRICA DO SUL>FORMULADOR) + (Nitrasoil - Buenos Aires ☒ Argentina<ARGENTINA>FORMULADOR);Não Classificado - Produto Não Classificado;Produto Pouco Perigoso ao Meio Ambiente;NAO;TRUE

35523; KBR-829M1-02;Nematóides vivos;Heterorhabditis bacteriophora (Nematóides entomopatogênicos) (840 g/kg);Koppert do Brasil Holding S.A. - Piracicaba/SP ;Agente Biológico de Controle;;Todas as culturas;Sphenophorus levis;"Bicudo da cana-de-açúcar; Gorgulho-da-cana"; (Koppert do Brasil Holding S.A. - Piracicaba/SP <BRASIL>FORMULADOR) + (Koppert do Brasil Holding S.A. - Piracicaba/SP <BRASIL>MANIPULADOR) + (Koppert do Brasil Holding Ltda.- Charqueada/SP<BRASIL>FORMULADOR) + (Koppert do Brasil Holding Ltda.- Charqueada/SP<BRASIL>MANIPULADOR) + (Koppert do Brasil Macrobiologicos Ltda ☒ Charqueada/SP<BRASIL>FORMULADOR) + (Koppert do Brasil Macrobiologicos Ltda ☒ Charqueada/SP<BRASIL>MANIPULADOR) + (Koppert BV - Holanda<PAÍSES BAIXOS, HOLANDA>FORMULADOR) + (Koppert Biological Systems, Inc.<ESTADOS UNIDOS>FORMULADOR) + (Koppert (Beijing) Agriculture Co., Ltd<CHINA, REPUBLICA POPULAR>FORMULADOR) + (Koppert México S.A. de C.V.<MÉXICO>FORMULADOR) + (Koppert Africa do Sul<ÁFRICA DO SUL>FORMULADOR) + (Nitrasoil - Buenos Aires ☒ Argentina<ARGENTINA>FORMULADOR);Não Classificado - Produto Não Classificado;Produto Pouco Perigoso ao Meio Ambiente;NAO;TRUE

35523; KBR-829M1-02;Nematóides vivos;Heterorhabditis bacteriophora (Nematóides entomopatogênicos) (840 g/kg);Koppert do Brasil Holding S.A. - Piracicaba/SP ;Agente Biológico de Controle;;Todas as culturas;Spodoptera frugiperda;Lagarta-militar; (Koppert do Brasil Holding S.A. - Piracicaba/SP <BRASIL>FORMULADOR) + (Koppert do Brasil Holding S.A. - Piracicaba/SP <BRASIL>MANIPULADOR) + (Koppert do Brasil Holding Ltda.- Charqueada/SP<BRASIL>FORMULADOR) + (Koppert do Brasil Holding Ltda.- Charqueada/SP<BRASIL>MANIPULADOR) + (Koppert do Brasil Macrobiologicos Ltda ☒ Charqueada/SP<BRASIL>FORMULADOR) + (Koppert do Brasil Macrobiologicos Ltda ☒ Charqueada/SP<BRASIL>MANIPULADOR) + (Koppert BV - Holanda<PAÍSES BAIXOS, HOLANDA>FORMULADOR) + (Koppert Biological Systems, Inc.<ESTADOS UNIDOS>FORMULADOR)

Object Explorer



Servers (1)

PostgreSQL 16

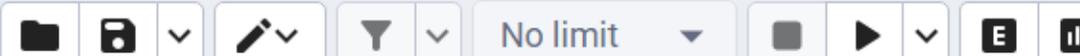
Databases (4)

Exemplo1

- > Casts
- > Catalogs
- > Event Triggers
- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (3)
 - > agronegocios
 - > clientes
 - > public
- > Subscriptions
- > Teste
- > exemplo
- > postgres
- > Login/Group Roles
- > Tablespaces

Dashboard × Properties × SQL × Statistics × Dependencies ×

Exemplo1/postgres@PostgreSQL 16



Query Query History

```
1 CREATE SCHEMA agronegocios;  
2 CREATE TABLE agrofitprodutosformulados (  
3 )
```

Data Output Messages Notifications

CREATE SCHEMA

Query returned successfully in 50 msec.

- ✓ Schemas (3)
 - ✓ agronegocios
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > Aa FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > 1.3 Sequences
 - ✓ Tables (1)
 - ✓ agrofitprodutosformulados
 - Columns (15)
 - nr_registro
 - marca_comercial
 - formulacao
 - ingrediente_ativo
 - titular_de_registro
 - classe
 - modo_de_acao
 - cultura
 - praga_nome_cientifico
 - praga_nome_comum
 - empresa_pais_tipo
 - classe_toxicologica

Exemplo1/postgres@PostgreSQL 16

Query **File** Query History

```
1 CREATE SCHEMA agronegocios;
2 CREATE TABLE agronegocios.agrofitprodutosformulados (
3     NR_REGISTRO DECIMAL(10) NOT NULL PRIMARY KEY,
4     MARCA_COMERCIAL VARCHAR(50),
5     FORMULACAO VARCHAR(50),
6     INGREDIENTE_ATIVO VARCHAR(50),
7     TITULAR_DE_REGISTRO VARCHAR(50),
8     CLASSE VARCHAR(50),
9     MODO_DE_ACAO VARCHAR(50),
10    CULTURA VARCHAR(50),
11    PRAGA_NOME_CIENTIFICO VARCHAR(50),
12    PRAGA_NOME_COMUM VARCHAR(50),
13    EMPRESA_PAIS_TIPO VARCHAR(50),
14    CLASSE_TOXICOLOGICA VARCHAR(50),
15    CLASSE_AMBIENTAL VARCHAR(50),
16    ORGANICOS VARCHAR(50),
17    SITUACAO VARCHAR(50)
18 );
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 30 msec.

Através do comando `psql \copy` é possível importar dados de um arquivo para uma tabela.

COPY: comando utilizado para importar a tabela

O arquivo
pode
estar

`COPY tabela_original_para_a_qual_os_dados_devem_ser_copiados`

`(`

`X,`

nome da tabela que receberá os dados

`(...):` dentro do parenteses colocamos as colunas que receberão os dados. Neste caso, utilizamos os nomes das

DELIMITER: após o delimiter devemos colocar o separador que o arquivo utiliza. O separador poder ser `;"`, `,"`, `"|"` entre outros. Neste caso, o separador do arquivo é `;"`;

maquina
cliente.

`DELIMITER ';' ;`
`CSV HEADER;`

FROM: após o "FROM" devemos colocar o diretório que está localizado o arquivo. O diretório sempre deve usar `"/`, nunca `"\"`, tanto

CSV HEADER: utilize "CSV HEADER" caso não queira importar a primeira linha. Para este arquivo não importamos a primeira linha, pois é o cabeçalho;

Object Explorer



Servers (1)

PostgreSQL 16

Databases (4)

Exemplo1

- > Casts
- > Catalogs
- > Event Triggers
- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (3)
 - > agronegocios
 - > clientes
 - > public
- > Subscriptions
- > Teste
- > exemplo
- > postgres
- > Login/Group Roles
- > Tablespaces

Dashboard × Properties × SQL × Statistics × Dependencies × Dependents ×

Exemplo1/postgres@PostgreSQL 16



Query Query History

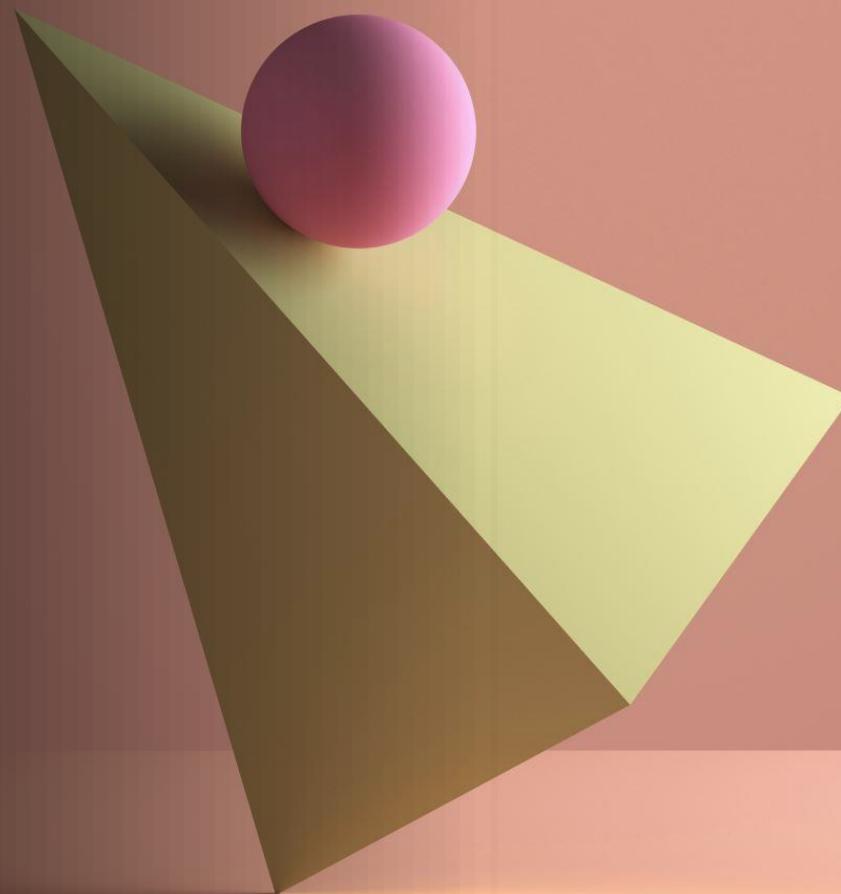
```
1 COPY agronegocios.agrofitprodutosformulados
2 (
3     NR_REGISTRO,
4     MARCA_COMERCIAL,
5     FORMULACAO,
6     INGREDIENTE_ATIVO,
7     TITULAR_DE_REGISTRO,
8     CLASSE,
9     MODO_DE_ACAO,
10    CULTURA,
11    PRAGA_NOME_CIENTIFICO,
12    PRAGA_NOME_COMUM,
13    EMPRESA_PAIS_TIPO,
14    CLASSE_TOXICOLOGICA,
15    CLASSE_AMBIENTAL,
16    ORGANICOS,
17    SITUACAO
18 )
19 FROM 'G:\2024-1\BD\agrofitprodutosformulados.csv'
20 DELIMITER ';'
21 CSV HEADER;
22
```

Data Output Messages Notifications

Query returned successfully in 30 msec.



Outras Fontes



Fontes

Formula 1

- <http://ergast.com/mrd/db/> Last updated: 12/06/2018

Alternativas

- <https://www.kaggle.com/cjgdev/formula-1-race-data-19502017>
- <https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020>

Aeroportos:

- <https://ourairports.com/data/>

GeoCities:

- <https://www.geonames.org/export/>

Not secure | ergast.com/mrd/db/

API do Desenvolvedor Ergast

Banco de Imagens

As imagens completas do banco de dados Ergast são publicadas logo após cada corrida sob a licença Attribution-NonCommercial-ShareAlike 3.0 Unported .



A estrutura do banco de dados é mostrada em um Diagrama de Relacionamento de Entidade e explicada no Guia do Usuário do Banco de Dados . As tabelas foram criadas usando: f1db_tables.sql

As imagens do banco de dados foram atualizadas pela última vez em: 14/11/2022

Imagens do banco de dados MySQL

Estes são despejos de dados do MySQL (5.7). A versão ANSI pode ser mais fácil de importar para outros formatos de banco de dados. A codificação de caracteres é "utf8" em ambas as versões.

[f1db.sql.gz](#) Despejo do banco de dados MySQL 5.7
[f1db_ansi.sql.gz](#) Despejo de banco de dados MySQL "compatível com ANSI"

Tabelas de banco de dados CSV

O banco de dados também pode ser baixado como um conjunto de arquivos CSV que podem ser importados para planilhas e outros tipos de software:

[f1db_csv.zip](#)

Cada arquivo CSV contém uma única tabela de banco de dados. A codificação de texto é UTF-8 e a primeira linha de cada arquivo contém os cabeçalhos das colunas. As tabelas são descritas no Guia do Usuário .

Isenção de responsabilidade

O autor se isenta de qualquer responsabilidade por qualquer perda ou dano decorrente do uso dessas imagens de banco de dados e não pode garantir a precisão dos dados.

Atribuição

A atribuição é opcional, mas apreciada ao abordar um público técnico. Uma referência a "Ergast" com um link para este site é suficiente.

Índice

- Documentação da API
- Lista de Temporadas
- Classificação de Corrida
- Resultados de corrida
- Resultados de qualificação
- Resultados de qualificação de sprint
- Classificação
- Informações do motorista
- Informações do Construtor
- Informações do Circuito
- Status de acatamento
- Tempo de volta
- Pit Stop
- Parâmetros de desenvolvimento
- Banco de dados de consulta
- Banco de Imagens
- Bateria de aplicativos
- Termos e Condições
- Perguntas Frequentes
- Comentários
- Últimas notícias
- Relatório de erros

links

- Contatos
- Web acessível

meta

- Conectar-se
- Entrada RSS
- RSS de comentários
- WordPress.org

Procurar por:



drivers	
driverId	int(11)
driverRef	varchar(255)
number	int(11)
code	varchar(3)

seasons	
year	int(11)
url	varchar(255)

qualifying	
qualifyId	int(11)
raceId	int(11)
driverId	int(11)
constructorId	int(11)
number	int(11)
position	int(11)
q1	varchar(255)
q2	varchar(255)
q3	varchar(255)

results	
resultId	int(11)
raceId	int(11)
driverId	int(11)
constructorId	int(11)
number	int(11)
grid	int(11)
position	int(11)
positionText	varchar(255)
positionOrder	int(11)
points	float
laps	int(11)
time	varchar(255)
milliseconds	int(11)
fastestLap	int(11)
rank	int(11)
fastestLapTime	varchar(255)
fastestLapSpeed	varchar(255)
statusId	int(11)

driverStandings	
driverStandingsId	int(11)
raceId	int(11)
driverId	int(11)
points	float
position	int(11)
positionText	varchar(255)
wins	int(11)

constructorStandings	
constructorStandingsId	int(11)
raceId	int(11)
constructorId	int(11)
points	float
position	int(11)
positionText	varchar(255)
wins	int(11)

constructorResults	
constructorResultsId	int(11)
raceId	int(11)
constructorId	int(11)
points	float
status	varchar(255)

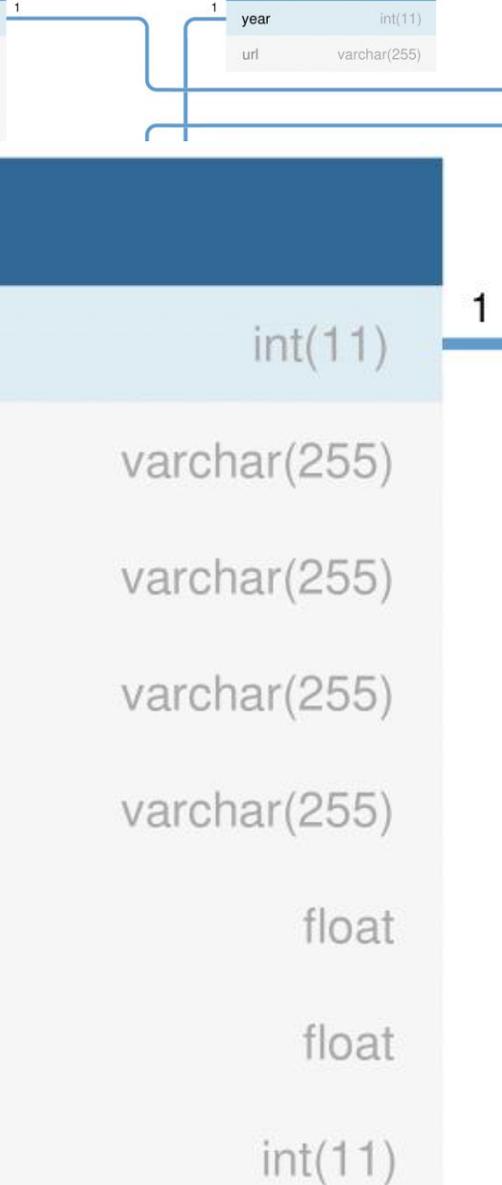
sprintResults	
sprintResultId	int(11)
raceId	int(11)
driverId	int(11)
constructorId	int(11)
number	int(11)
grid	int(11)
position	int(11)
positionText	varchar(255)
positionOrder	int(11)
points	float
laps	int(11)
time	varchar(255)
milliseconds	int(11)
fastestLap	int(11)
fastestLapTime	varchar(255)
statusId	int(11)

pitStops	
raceId	int(11)
driverId	int(11)
stop	int(11)
lap	int(11)
time	time
duration	varchar(255)
milliseconds	int(11)

status	
statusId	int(11)
status	varchar(255)

lapTimes	
raceId	int(11)
driverId	int(11)
lap	int(11)
position	int(11)
time	varchar(255)
milliseconds	int(11)

circuits	
circuitId	int(11)
circuitRef	varchar(255)
name	varchar(255)
location	varchar(255)
country	varchar(255)
lat	float
lng	float
alt	int(11)
url	varchar(255)



List of Tables
circuits
constructorResults
constructorStandings
constructors
driverStandings
drivers
lapTimes
pitStops
qualifying
races
results
seasons
status

General Notes
Dates, times and durations are in ISO 8601 format
Dates and times are UTC
Strings use UTF-8 encoding
Primary keys are for internal use only
Fields ending with "Ref" are unique identifiers for external use
A grid position of '0' is used for starting from the pitlane
Labels used in the positionText fields:
"D" - disqualified
"E" - excluded
"F" - failed to qualify
"N" - not classified
"R" - retired
"W" - withdrew

circuits table

Field	Type	Null	Key	Default	Extra	Description
circuitId	int(11)	NO	PRI	NULL	auto_increment	Primary key
circuitRef	varchar(255)	NO				Unique circuit identifier
name	varchar(255)	NO				Circuit name
location	varchar(255)	YES		NULL		Location name
country	varchar(255)	YES		NULL		Country name
lat	float	YES		NULL		Latitude
lng	float	YES		NULL		Longitude

circuits.csv

Arquivo Editar Formatar Exibir Ajuda

```
circuitId,circuitRef,name,location,country,lat,lng,alt,url
```

```
1,"albert_park","Albert Park Grand Prix Circuit","Melbourne","Australia",-37.8497,144.
2,"sepang","Sepang International Circuit","Kuala Lumpur","Malaysia",2.76083,101.738,18
3,"bahrain","Bahrain International Circuit","Sakhir","Bahrain",26.0325,50.5106,7,"http
4,"catalunya","Circuit de Barcelona-Catalunya","Montmeló","Spain",41.57,2.26111,109,"l
5,"istanbul","Istanbul Park","Istanbul","Turkey",40.9517,29.405,130,"http://en.wikipe
6,"monaco","Circuit de Monaco","Monte-Carlo","Monaco",43.7347,7.42056,7,"http://en.wil
7,"villeneuve","Circuit Gilles Villeneuve","Montreal","Canada",45.5,-73.5228,13,"http
8,"magny_cours","Circuit de Nevers Magny-Cours","Magny Cours","France",46.8642,3.1636:
9,"silverstone","Silverstone Circuit","Silverstone","UK",52.0786,-1.01694,153,"http://
10,"hockenheimring","Hockenheimring","Hockenheim","Germany",49.3278,8.56583,103,"http
11,"hungaroring","Hungaroring","Budapest","Hungary",47.5789,19.2486,264,"http://en.wil
12,"valencia","Valencia Street Circuit","Valencia","Spain",39.4589,-0.331667,4,"http:,
13,"spa","Circuit de Spa-Francorchamps","Spa","Belgium",50.4372,5.97139,401,"http://er
14,"monza","Autodromo Nazionale di Monza","Monza","Italy",45.6156,9.28111,162,"http://
15,"marina_bay","Marina Bay Street Circuit","Marina Bay","Singapore",1.2914,103.864,18
16,"fuji","Fuji Speedway","Oyama","Japan",35.3717,138.927,583,"http://en.wikipedia.org
17,"shanghai","Shanghai International Circuit","Shanghai","China",31.3389,121.22,5,"ht
18,"interlagos","Autódromo José Carlos Pace","São Paulo","Brazil",-23.7036,-46.6997,78
19,"indianapolis","Indianapolis Motor Speedway","Indianapolis","USA",39.795,-86.2347,:
20,"nurburgring","Nürburgring","Nürburg","Germany",50.3356,6.9475,578,"http://en.wikip
21,"imola","Autodromo Enzo e Dino Ferrari","Imola","Italy",44.3439,11.7167,37,"http://
22,"suzuka","Suzuka Circuit","Suzuka","Japan",34.8431,136.541,45,"http://en.wikipedia
23 "osterreichring" "A1-Ring" "Sielhero" "Austria" 47 2197 14 7647 678 "http://en wil
```

Criando a tabela circuitos

The screenshot shows a PostgreSQL browser interface with the following components:

- Browser:** A tree view on the left showing the database structure. The 'public' schema is expanded, and the 'Tables (1)' folder is selected, containing the 'circuitos' table.
- Properties:** The top right pane shows the connection name 'formula1/postgres@PostgreSQL 14*'. Below it, the current database is 'formula1/postgres@PostgreSQL 14'. There are icons for file operations and a 'No limit' dropdown.
- Query:** The main editor pane contains the following SQL code:

```
1 CREATE TABLE circuitos (  
2     circuitId DECIMAL(4) NOT NULL PRIMARY  
3     circuitRef VARCHAR(20) NOT NULL,  
4     name VARCHAR(60) NOT NULL,  
5     location VARCHAR(40) NOT NULL,  
6     country VARCHAR(20) NOT NULL,  
7     lat FLOAT (10) NOT NULL,  
8     lng FLOAT (10) NOT NULL,  
9     alt FLOAT (10) NOT NULL,  
10    url VARCHAR(60) NOT NULL  
11 );  
12
```
- Data output / Messages:** The bottom pane shows the execution result: 'CREATE TABLE' and 'Query returned successfully in 124 msec.'

Circuitos

Script de
criação
da
tabela
circuitos.

circuitos=

{circuitId,circuitRef,name,location,country,lat,lng,alt,url}

```
CREATE TABLE circuitos (  
    circuitId DECIMAL(4) NOT NULL PRIMARY KEY,  
    circuitRef VARCHAR(20) NOT NULL,  
    name VARCHAR(60) NOT NULL,  
    location VARCHAR(40) NOT NULL,  
    country VARCHAR(20) NOT NULL,  
    lat FLOAT (10) NOT NULL,  
    lng FLOAT (10) NOT NULL,  
    alt FLOAT (10) NOT NULL,  
    url VARCHAR(60) NOT NULL  
);
```

COPY circuitos

(

circuitId,

circuitRef,

name,

location,

country,

lat,

lng,

alt,

url,

)

FROM 'F:\2022-2\BD\Kthannus\Formula1\circuits.csv'

DELIMITER ';' ;

CSV HEADER;



Executando o comando

pgAdmin File Object Tools Help

Browser Properties SQL formula1/postgres@PostgreSQL 14*

formulal/postgres@PostgreSQL 14

No limit

Query Query History

```
1 COPY circuitos
2 (
3     circuitId,
4     circuitRef,
5     name,
6     location,
7     country,
8     lat,
9     lng,
10    alt,
11    url
12 )
13 FROM 'F:\2022-2\BD\Kthannus\Formula1\circuits.csv'
14 DELIMITER ','
15 CSV HEADER;
16
```

Data output Messages Notifications

COPY 79

Query returned successfully in 110 msec.

pgAdmin 4

File Object Tools Help

Browser

Publications

Schemas (1)

public

- Aggregates
- Collations
- Domains
- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (1)
 - circuitos
 - Columns (9)
 - circuitid
 - circuitref
 - name
 - location
 - country
 - lat
 - lng
 - alt

formula1/postgres@PostgreSQL 14

No limit

Query

```
1 SELECT name, country FROM circuitos;
```

Data output

	name	country
1	Albert Park Grand Prix Circuit	Australia
2	Sepang International Circuit	Malaysia
3	Bahrain International Circuit	Bahrain
4	Circuit de Barcelona-Catalunya	Spain

Total rows: 79 of 79 Query complete 00:00:00.074

Execício

Criar e
popular as
seguintes
tabelas
tabelas:

- Racers
- Drivers
- Constructors
- ...

