



Módulo 2

SQL Introdução

< thefutureisblue.me />

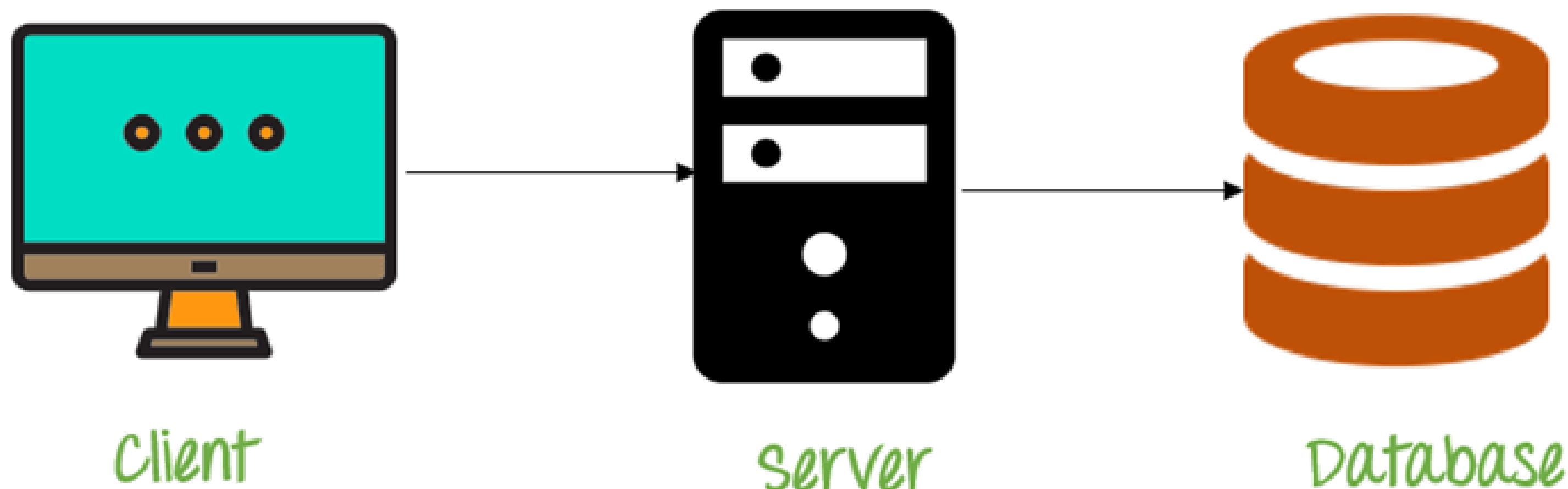


SQL: Introdução

- **SQL** é uma linguagem utilizada nos principais bancos de dados relacionais.
- Em vez de escrever softwares, SQL é utilizada para manipular dados, podendo **CRIAR, LER, EDITAR e REMOVER**.
- Essas ações podem ser traduzidas para o inglês **Create, Read, Update e Delete**, formando o acrônimo **CRUD**.

Arquitetura Client-Server

Three Tier Architecture



© guru99.com

- Em uma arquitetura de software, o usuário (frontend/client) acessa o servidor (backend/server) em busca de informações que estão armazenada no banco de dados (database).

Principais Bancos de Dados Relacionais





Como assim "relacional"?

- Um banco de dados relacional representa informações que estão estruturadas e relacionadas entre si.
- Essas informações estão organizadas em tabelas, sendo que essas tabelas podem estar conectadas ou não.



- Uma tabela de banco de dados é bem semelhante a uma tabela do excel, organizando as informações em linhas e colunas.
- Cada **coluna** representa uma informação. No caso de uma pessoa, as colunas seriam: **nome, sobrenome, idade**, etc.
- Cada **linha** representa um registro. No caso de uma tabela de pessoas, os dados de **cada pessoa** estaria **em uma linha**.



Tabelas

→ Servidor: 127.0.0.1 » Base de Dados: blue-modulo2 » Tabela: pessoas

Procurar

Estrutura

SQL

Pesquisar

Inserir

Exportar

Importar

Privilégios

Operações

Estrutura da tabela

Visão de relação(ões)

	#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Acções
<input type="checkbox"/>	1	id	int(11)			Não	Nenhum			Muda Elimina Mais
<input type="checkbox"/>	2	nome	varchar(255)	utf8mb4_general_ci		Não	Nenhum			Muda Elimina Mais
<input type="checkbox"/>	3	sobrenome	varchar(255)	utf8mb4_general_ci		Não	Nenhum			Muda Elimina Mais
<input type="checkbox"/>	4	idade	varchar(255)	utf8mb4_general_ci		Não	Nenhum			Muda Elimina Mais





Tabelas: Schema

- Em um banco de dados relacionado, as tabelas possuem um Schema.
- Esse Schema contém a estrutura da tabela, indicando quais são as colunas, qual o tipo de dados que cada coluna suporta e qualquer informação extra que defina aquela tabela.



Tipos de dados mais comuns

Tipo	Exemplo de utilização
INT	Números inteiros curtos
BIGINT	Números inteiros longos
DECIMAL	Números com casas decimais
VARCHAR	Textos curtos
TEXT	Textos longos
TINYINT	Geralmente usado para booleanos (0 = false e 1 = true)
DATETIME	Armazena data e hora

Tipos de dados numéricos

Tipo de dados	Intervalo	Armazenamento
bigint	-2^{63} (-9.223.372.036.854.775.808) a $2^{63}-1$ (9.223.372.036.854.775.807)	8 bytes
int	-2^{31} (-2.147.483.648) a $2^{31}-1$ (2.147.483.647)	4 bytes
smallint	-2^{15} (-32.768) a $2^{15}-1$ (32.767)	2 bytes
tinyint	0 a 255	1 byte

- A quantidade de informação por tipo de dados depende da tecnologia de banco de dados.

Tipos de dados para textos

TINYTEXT	256 bytes	
TEXT	65,535 bytes	~64kb
MEDIUMTEXT	16,777,215 bytes	~16MB
LONGTEXT	4,294,967,295 bytes	~4GB

- A quantidade de informação por tipo de dados depende da tecnologia de banco de dados.

Servidor: 127.0.0.1 » Base de Dados: blue-modulo2 » Tabela: pessoas

[Procurar](#)
[Estrutura](#)
[SQL](#)
[Pesquisar](#)
[Inserir](#)
[Exportar](#)
[Importar](#)
[Privilégios](#)
[Operações](#)

[Estrutura da tabela](#)
[Visão de relação\(ões\)](#)

	#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Acções
<input type="checkbox"/>	1	id	int(11)			Não	Nenhum			Muda Elimina Mais
<input type="checkbox"/>	2	nome	varchar(255)	utf8mb4_general_ci		Não	Nenhum			Muda Elimina Mais
<input type="checkbox"/>	3	sobrenome	varchar(255)	utf8mb4_general_ci		Não	Nenhum			Muda Elimina Mais
<input type="checkbox"/>	4	idade	varchar(255)	utf8mb4_general_ci		Não	Nenhum			Muda Elimina Mais

- As tabelas no SQL possuem uma coluna de ID que representa um número sequencial para identificar cada registro. Essa coluna geralmente possui o tipo INT ou BIGINT.

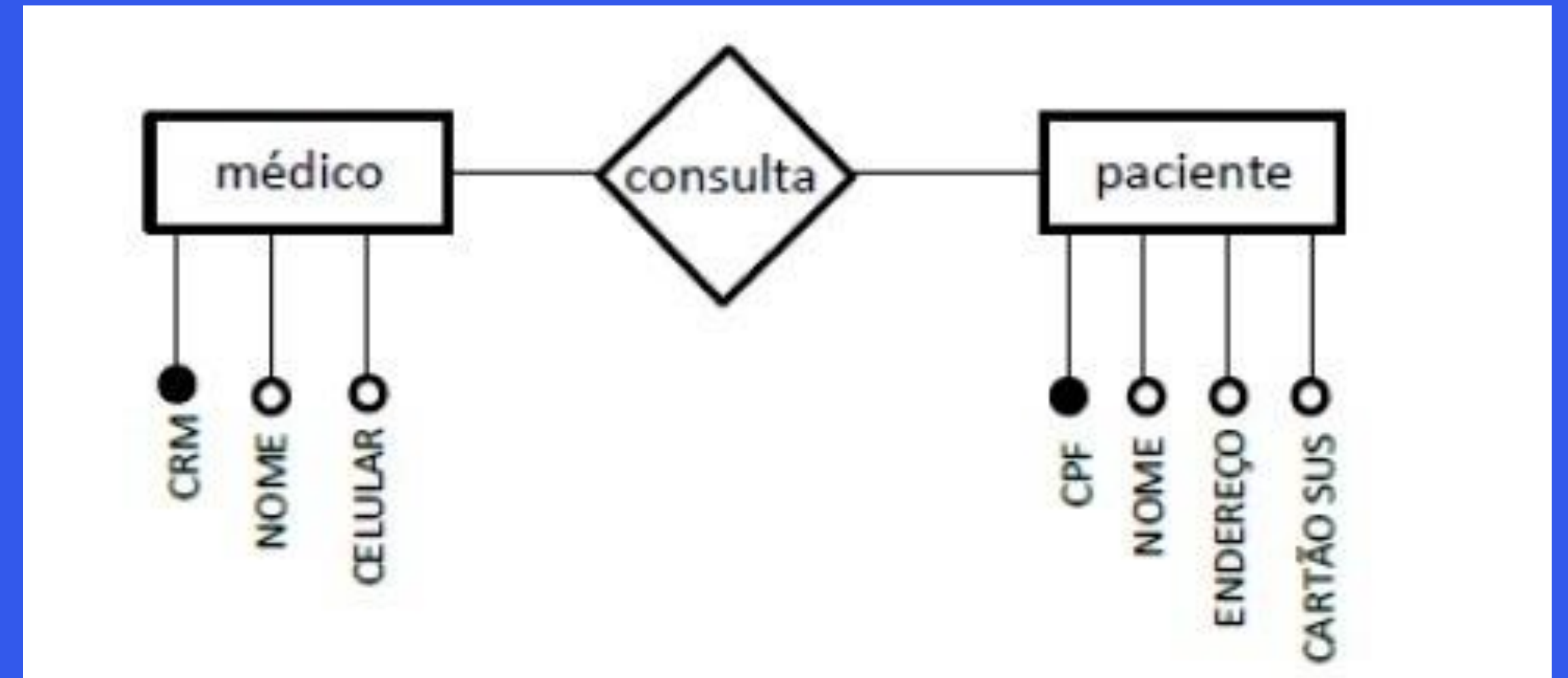


Coluna ID

- Atributo ID é um número sequencial gerado automaticamente pelo AUTO_INCREMENT.
- Tabelas se conectam através do ID.
- Para conectar as tabelas e habilitar o AUTO_INCREMENT, a coluna ID precisa ser uma **Chave Primária**.



O que são chaves primárias?



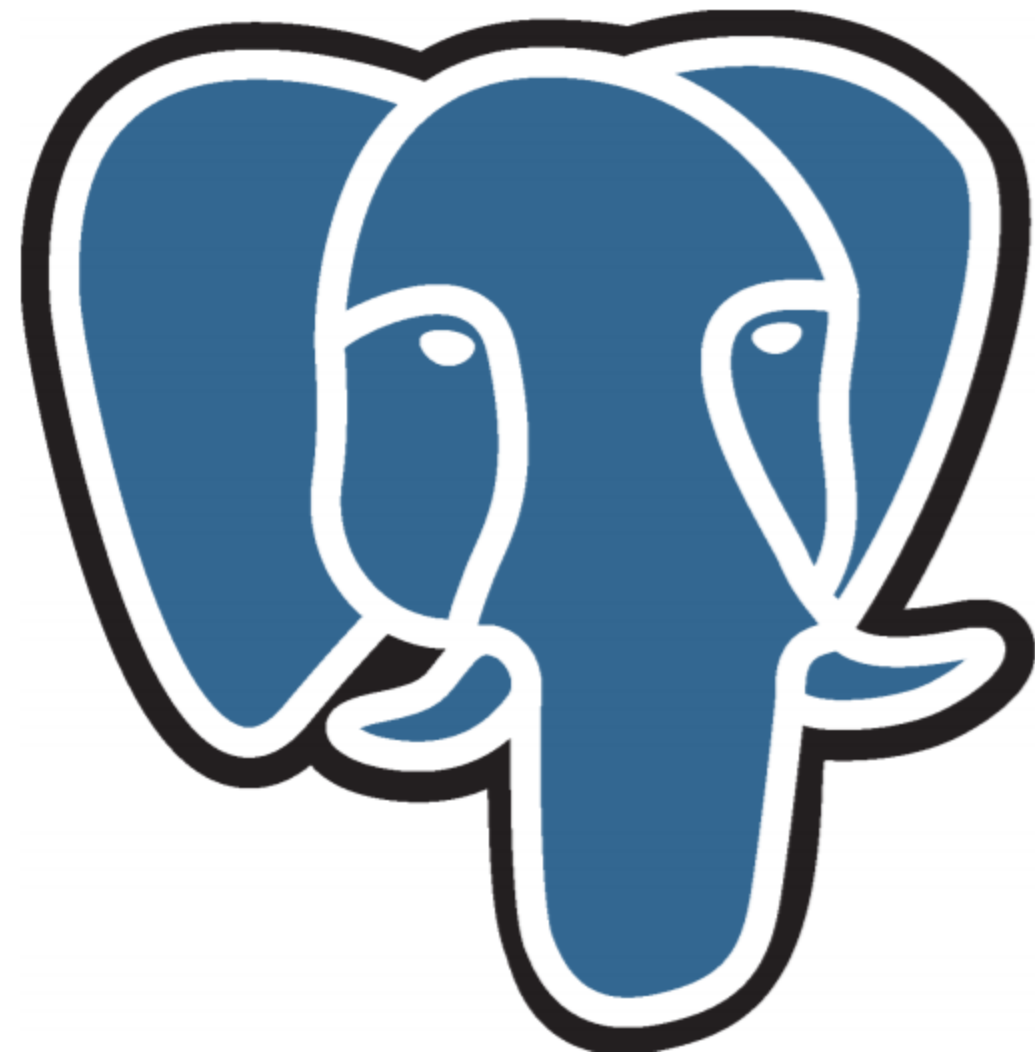
* Uma tabela tem uma coluna ou uma combinação de colunas que contém valores que identificam, exclusivamente, cada linha na tabela.

* Essa coluna (ou colunas) é chamada de chave primária (PRIMARY KEY).



**Botando para
rodar!**

PostgreSQL



























PostgreSQL

the world's most advanced open source database

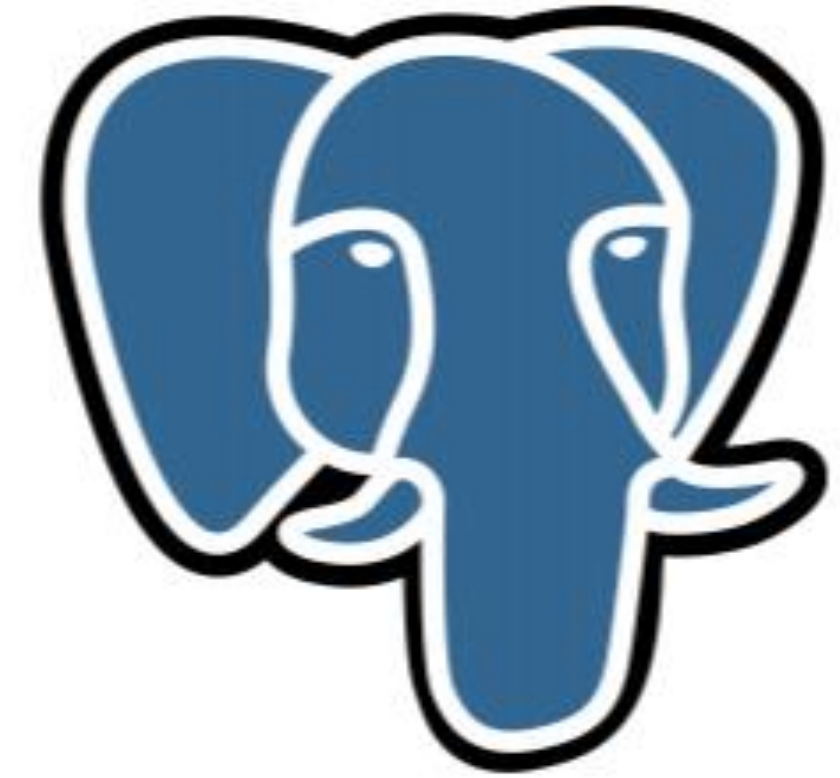
Database Ranking: 4º lugar

371 systems in ranking, June 2021

Rank			DBMS	Database Model	Score		
Jun 2021	May 2021	Jun 2020			Jun 2021	May 2021	Jun 2020
1.	1.	1.	Oracle 	Relational, Multi-model 	1270.94	+1.00	-72.65
2.	2.	2.	MySQL  OpenSource!	Relational, Multi-model 	1227.86	-8.52	-50.03
3.	3.	3.	Microsoft SQL Server 	Relational, Multi-model 	991.07	-1.59	-76.24
4.	4.	4.	PostgreSQL  OpenSource!	Relational, Multi-model 	568.51	+9.26	+45.53
5.	5.	5.	MongoDB  OpenSource!	Document, Multi-model 	488.22	+7.20	+51.14
6.	6.	6.	IBM Db2 	Relational, Multi-model 	167.03	+0.37	+5.23
7.	7.	 8.	Redis  OpenSource!	Key-value, Multi-model 	165.25	+3.08	+19.61
8.	8.	 7.	Elasticsearch  OpenSource!	Search engine, Multi-model 	154.71	-0.65	+5.02
9.	9.	9.	SQLite  OpenSource!	Relational	130.54	+3.84	+5.72
10.	10.	 11.	Microsoft Access	Relational	114.94	-0.46	-2.24
11.	11.	 10.	Cassandra  OpenSource!	Wide column	114.11	+3.18	-4.90
12.	12.	12.	MariaDB  OpenSource!	Relational, Multi-model 	96.79	+0.10	+7.00



Introdução ao PostgreSQL



O servidor de bancos de dados PostgreSQL (SGBD) teve seu projeto iniciado na década de 80 na Universidade de Berkeley com o nome de Ingres e posteriormente teve seu nome alterado para Post Ingres que, por fim foi lançado em sua versão inicial como PostgreSQL em 1996.



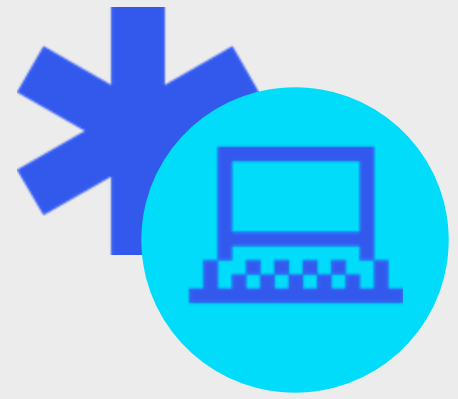


Introdução ao PostgreSQL

O PostgreSQL é um banco de dados open source e foi pioneiro em muitos dos conceitos que se tornaram disponíveis em alguns sistemas de banco de dados comerciais mais tarde.

Usado por milhares de empresas, incluindo:





ElephantSQL: Database as a Service

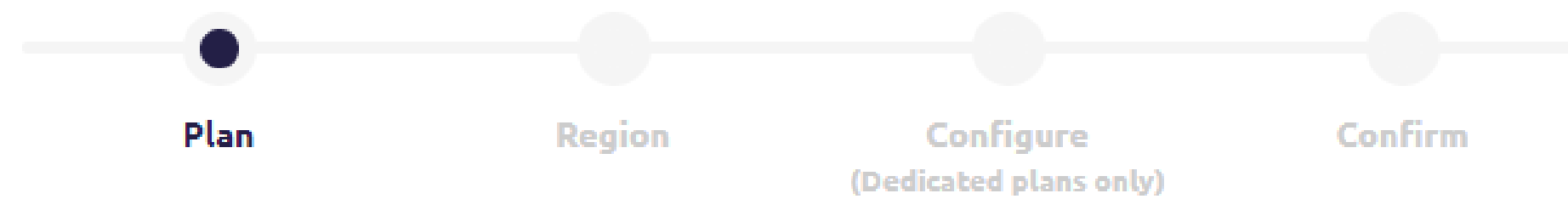
- Serviço que disponibiliza banco de dados PostgreSQL em menos 2 minutos de configuração;
- Banco de dados na nuvem rodando na AWS;
- Possui um Free Tier que disponibiliza 20MB de espaço;
- Acesse em: <https://www.elephantsql.com/>



Create new instance

No credit card Please [add a credit card](#) if you want to subscribe to a paid plan

Missing billing information Please [fill in all required information](#) if you want to subscribe to a paid plan



Select a plan and name - Step 1 of 4

Name	<input type="text" value="Blue-Modulo2"/>
Plan	<input type="text" value="Tiny Turtle (Free)"/>
Tags	<input type="text"/>

Tags are used to separate your instances between projects. This is primarily used in the project listing view for easier navigation and access control.

Tags allow admins to [manage team members access](#) to different groups of instances.

Plan



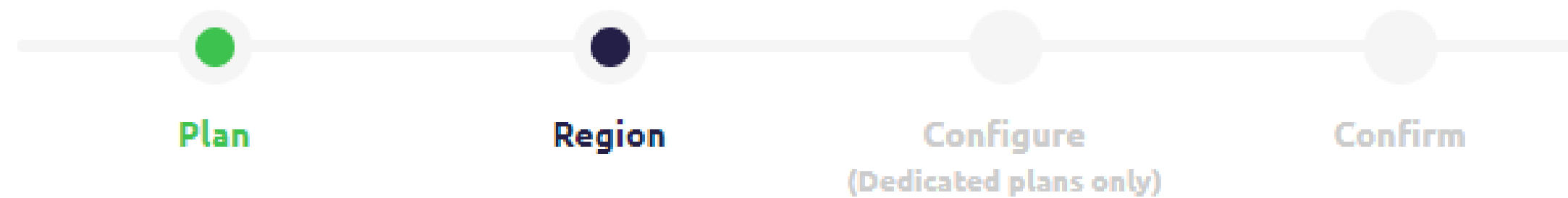
Tiny Turtle

See the [plan page](#) to learn about the different plans.

Cancel

Select Region





Select a region and data center - Step 2 of 4

Data center

US-East-1 (Northern Virginia) ▼

Amazon Web Services

- US-West-2 (Oregon)
- US-East-1 (Northern Virginia)
- US-East-2 (Ohio)
- US-West-1 (Northern California)
- EU-North-1 (Stockholm)
- EU-South-1 (Milan)
- EU-West-1 (Ireland)
- EU-West-2 (London)
- EU-West-3 (Paris)
- EU-Central-1 (Frankfurt)
- CA-Central-1 (Canada)
- AP-SouthEast-1 (Singapore)
- AP-SouthEast-2 (Sydney)
- AP-NorthEast-1 (Tokyo)
- AP-NorthEast-2 (Seoul)
- AP-NorthEast-3 (Osaka)
- AP-South-1 (Mumbai)
- AP-East-1 (Hong Kong)
- SA-East-1 (Sao Paulo)

« Back

Plan



Tiny Turtle

See the [plan page](#) to learn about the different plans.

Cancel

Review

Plan


Region

Configure
(Dedicated plans only)

Confirm

Confirm new instance - Step 4 of 4

Plan



Tiny Turtle

Total: Free

Name:

Provider:

Region:

Blue-Modulo2
Amazon Web Services
SA-East-1 (Sao Paulo)

« Back

Cancel

Create instance ✓

Conectando ao banco na nuvem

Abra o software DBeaver, disponível em <https://dbeaver.io/>

Crie uma nova conexão com o database, selecione o tipo PostgreSQL e preencha os dados de acordo com o site ElephantSQL.

Details

Server tuffi.db.elephantsql.com (tuffi-01)

Region amazon-web-services::sa-east-1

Created at 2021-06-04 05:44 UTC+00:00

User & Default database bmtpripl

Reset

Password YkyqGP...  



URL postgres://bmtpripl:***@tuffi.db.elephantsql.com/bmtpripl  

Current database size


Max database size 20 MB



Connect to a database

Connection Settings

PostgreSQL connection settings

 PostgreSQL

Main

PostgreSQL

Driver properties

SSH

Proxy

SSL

Server

Host: Port:

Database:

Authentication

Authentication:

Database Native

Username:

Password: ☒ Save password locally

Advanced

User role:

Local Client:

PostgreSQL Binaries

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: PostgreSQL

Edit Driver Settings

Test Connection ...

< Back

Next >

Finish

Cancel

Criando a primeira tabela

Com a conexão feita, clique com o botão direito em cima da conexão no Database Navigator, selecione a opção "SQL Editor", clique em "Open SQL Script" e digite o seguinte trecho:

```
CREATE TABLE pessoas (  
    id INT NOT NULL GENERATED ALWAYS AS IDENTITY,  
    nome VARCHAR NOT NULL,  
    sobrenome VARCHAR NOT NULL,  
    idade INT NULL  
);
```


Criando a primeira tabela

Para adicionar dados, clique na aba "Data" e aperte o atalho "Alt + Insert" (ou clique no botão de adicionar dados, na parte inferior da tabela);

Para persistir a informação você deve preenchê-la e clicar em "Save", isso irá executar o comando SQL para inserir o dado.

The screenshot shows the DBeaver Data Editor interface. At the top, there are tabs for 'Properties', 'Data', and 'ER Diagram'. Below the tabs, a filter bar shows 'pessoas' and a SQL expression input. The main area displays a table with one row. The table has columns: 'id', 'nome', 'sobrenome', and 'idade'. The row contains values: '[NULL]', 'Paulo', 'Salvatore', and '27'. The interface includes a left sidebar with 'Grid' and 'Text' views. A bottom toolbar contains 'Save', 'Cancel', 'Script', and other icons. Red boxes and numbers highlight specific elements: '1' points to the 'Save' button, '2' points to the 'id' column header, and '3' points to the 'Record' button.

id	nome	sobrenome	idade
[NULL]	Paulo	Salvatore	27

Pronto! Tabela criada e registro inserido!

Para criar a tabela, executamos um código SQL contendo as colunas e as informações específicas de cada coluna.

Porém, para adicionar um registro, utilizamos o DBeaver, que executou um código SQL para nós.



Agora que criamos uma tabela e temos registros, vamos praticar com novas tabelas?

Crie uma tabela chamada 'tipos_produtos', com as colunas 'id' e 'descricao'.

O campo 'id' deverá ser a chave primária da tabela.

O campo 'descricao' deverá aceitar textos de, no máximo, 30 chars.

Vamos criar uma tabela para os produtos?

Crie uma tabela chamada 'produtos', com as colunas 'id', 'descricao', 'preco' e 'tipo_produto_id', sendo que esse último precisa estar conectado com a tabela anterior, que contém os tipos de produtos.

Como assim conectado?



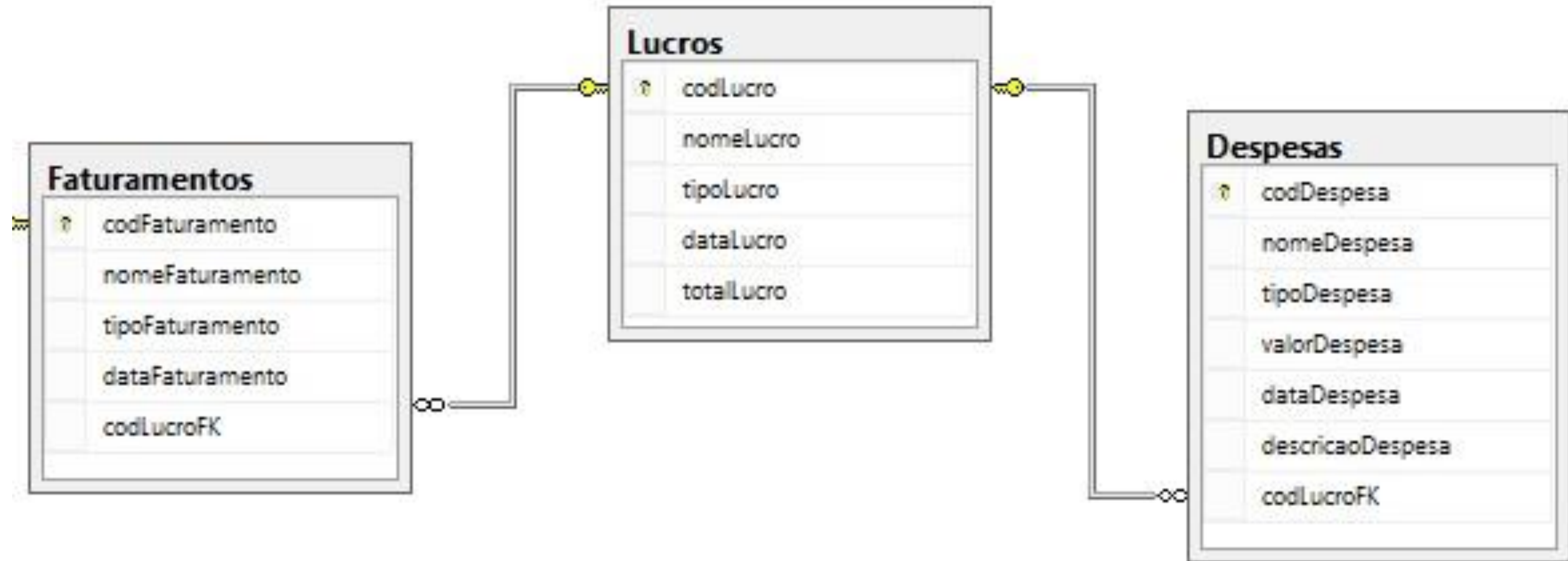
FOREIGN KEYS

- Uma **FOREIGN KEY**, ou chave estrangeira, é o campo que estabelece o **relacionamento entre duas tabelas**.
- Assim, uma coluna corresponde à mesma coluna que é a chave primária de outra tabela.
- Dessa forma, deve-se especificar, na tabela que contém a chave estrangeira, **quais são essas colunas e a qual tabela está relacionada**.





Exemplo



Como criamos uma foreign key?

Em banco de dados, é possível criar chaves estrangeiras de duas maneiras principais:

- Durante a criação da própria tabela, referenciando a qual coluna de qual tabela aquele campo se conecta.
- Após a criação da tabela, alterando-a e adicionando uma referência.



Botando para rodar!

**Criando a tabela de produtos
com uma foreign key.**





Botando para rodar!

Crie alguns registros nas tabelas de 'pessoas', 'produtos_tipos' e 'produtos'.



Por hoje é só!
Obrigado! =)