

THIAGO ADRIANO

POSTECH

MACHINE LEARNING ENGINEERING

BANCOS DE DADOS

AULA 01

SUMÁRIO

O QUE VEM POR AÍ?	3
HANDS ON	4
SAIBA MAIS.....	5
O QUE VOCÊ VIU NESTA AULA?	16
REFERÊNCIAS.....	17

EMSE

O QUE VEM POR AÍ?

Nesta aula, você aprenderá conhecimentos essenciais sobre desenvolvimento com banco de dados relacionais. Explicaremos o que é o SQL e quais são os benefícios de se utilizar um banco de dados relacional, tanto local, em um servidor na sua empresa, como em uma Cloud, como na AWS. Ao final desta aula, você terá conhecimentos sólidos sobre como estruturar e manipular um banco de dados relacional usando o PostgreSQL.

Para acessar o GitHub desta aula, clique [aqui](#).

HANDS ON

Nessa aula prática, os professores e professoras demonstram como trabalhar com um banco relacional utilizando o PostgreSQL na AWS RDS, iniciando com a configuração de um ambiente com o banco de dados PostgreSQL local com Docker até a criação do banco de dados na AWS RDS.



SAIBA MAIS

A Structured Query Language (SQL) é uma linguagem de consulta estruturada projetada para interagir com sistemas de gerenciamento de bancos de dados relacionais. Ela permite que os usuários criem, modifiquem e consultem dados de maneira eficaz, seguindo um conjunto de regras e comandos bem definidos.

A linguagem SQL é usada para executar diversas tarefas, como criação de tabelas, inserção de dados, recuperação de informações e definição de restrições de integridade.

Para que possamos entender melhor o SQL, confira uma lista contendo seus principais componentes:

- **DDL (Data Definition Language):** essa parte do SQL é usada para definir e gerenciar estruturas de bancos de dados, incluindo a criação, alteração e exclusão de tabelas, índices e outros objetos do banco de dados.
- **DML (Data Manipulation Language):** o DML lida com a manipulação de dados, permitindo que os usuários insiram, atualizem e excluam registros em tabelas.
- **DQL (Data Query Language):** o DQL é usado para consultar dados de tabelas, permitindo que os usuários recuperem informações específicas com base em critérios de pesquisa.
- **DCL (Data Control Language):** esta parte do SQL lida com permissões e privilégios de acesso aos dados, definindo quem pode realizar determinadas operações no banco de dados.

A figura 1 demonstra esses pontos:

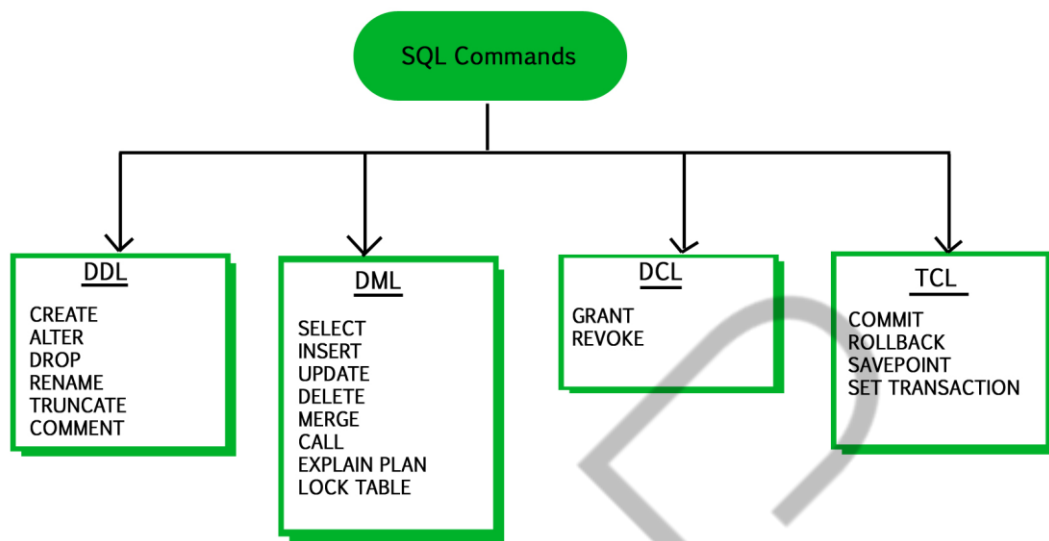


Figura 1 - Estrutura SQL
Fonte: Geeks for Geeks (2022)

Agora que sabemos o que é o SQL, vamos a uma introdução sobre o PostgreSQL.

O PostgreSQL é um sistema de gerenciamento de banco de dados relacional de código aberto, amplamente respeitado por sua robustez e recursos avançados. Ele é conhecido por sua confiabilidade, escalabilidade e riqueza de recursos, tornando-se uma escolha para diversas aplicações, desde pequenos projetos até grandes empresas.

Características do PostgreSQL

- Conformidade com padrões: o PostgreSQL segue rigorosamente os padrões SQL, garantindo que pessoas desenvolvedoras possam escrever consultas SQL compatíveis com a maioria dos sistemas de gerenciamento de banco de dados relacionais.
- Extensibilidade: uma das características mais notáveis do PostgreSQL é sua capacidade de ser estendido. Desenvolvedores(as) podem criar

funções personalizadas, tipos de dados, operadores e muito mais, o que o torna altamente adaptável às necessidades específicas do projeto.

- Suporte a dados não-estruturados: além de dados tabulares tradicionais, o PostgreSQL oferece suporte a dados semi-estruturados e JSON, sendo adequado para aplicativos web modernos e serviços de API.
- Transações ACID: o PostgreSQL garante transações ACID (Atomicidade, Consistência, Isolamento e Durabilidade); portanto, é confiável para aplicações que exigem alta integridade e segurança dos dados.
- Replicação e alta disponibilidade: o PostgreSQL oferece opções de replicação, permitindo a criação de ambientes de alta disponibilidade para minimizar o tempo de inatividade.
- Desempenho: o PostgreSQL é conhecido por seu desempenho sólido e por sua otimização de consultas, tornando-o adequado para ambientes de alto tráfego e grandes volumes de dados.

Por que escolher o PostgreSQL?

- Código aberto: o PostgreSQL é uma solução de código aberto, o que significa que é gratuito para uso e sua comunidade ativa garante desenvolvimento e suporte constantes.
- Robustez e confiabilidade: a longa história do PostgreSQL no mercado é um testemunho de sua robustez e confiabilidade, sendo uma escolha sólida para empresas que exigem dados seguros.
- Flexibilidade: a capacidade de personalizar e estender o PostgreSQL o torna adequado para uma variedade de casos de uso e setores.
- Comunidade ativa: a comunidade do PostgreSQL é conhecida por seu apoio e colaboração, tornando a resolução de problemas e o desenvolvimento de soluções mais fáceis.
- Ecossistema de ferramentas: o PostgreSQL possui um ecossistema robusto de ferramentas de terceiros que o complementam, facilitando o gerenciamento e o desenvolvimento de aplicativos.

Agora que conhecemos o SQL e sabemos o que é o PostgreSQL, que tal criarmos o nosso primeiro banco de dados em uma cloud gratuita? Para isso,

utilizaremos o Supabase neste material de apoio, para que, na nossa aula prática, possamos criá-lo na AWS, utilizando o RDS.

Para quem não conhece o Supabase, ele é um projeto 100% open-source que utiliza outras ferramentas open-source para fornecer serviços de autenticação, banco de dados e armazenamento de arquivos. Nós utilizaremos essa plataforma para criar um banco de dados PostgreSQL e, depois, acessar esse banco utilizando o pgadmin 4.

Nosso primeiro passo será [acessar a página do Supabase](#). Autentique com a sua conta do GitHub e em seguida clique em “New project”:

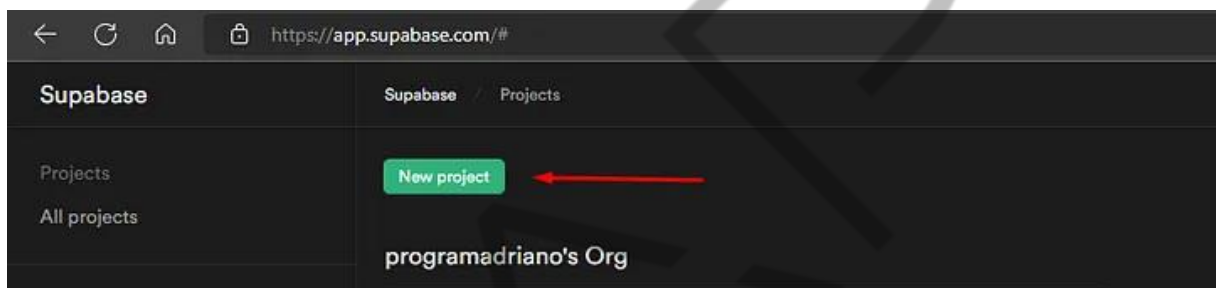
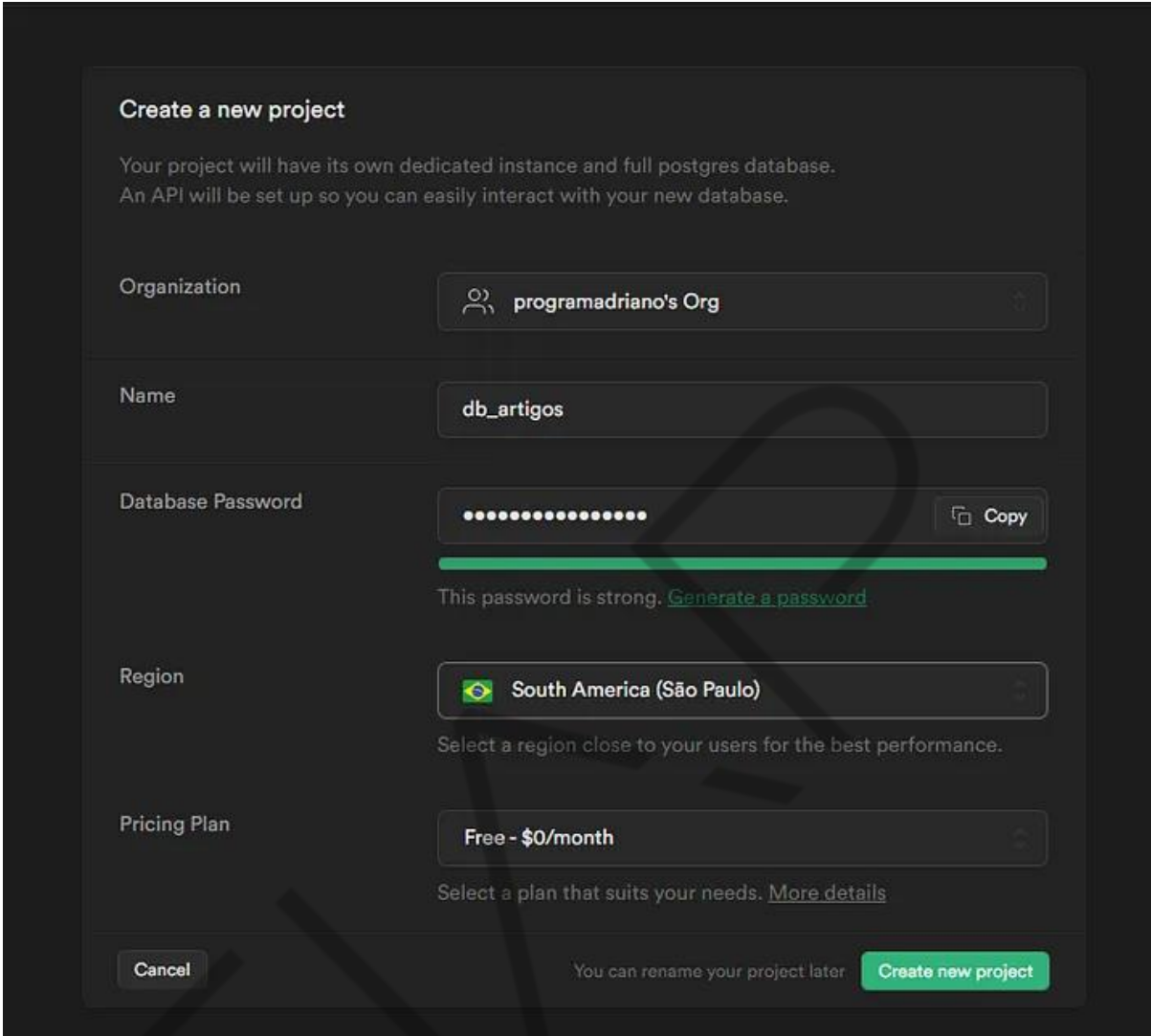


Figura 2 - Criando projeto no supabase
Fonte: Elaborado pelo autor (2024)

O próximo passo será preencher o form com as informações do seu projeto. Para este exemplo, eu preenchi o nome com **db_artigos** e, em Region, eu deixei **Brasil**, conforme a figura 3.



The screenshot shows the 'Create a new project' form in Supabase. The form is dark-themed and contains the following fields and options:

- Organization:** A dropdown menu showing 'programadriano's Org' with a user icon and a star icon.
- Name:** A text input field containing 'db_artigos'.
- Database Password:** A password input field with a strength indicator (a green bar) and a 'Copy' button. Below the field, it says 'This password is strong. [Generate a password](#)'.
- Region:** A dropdown menu showing 'South America (São Paulo)' with a Brazilian flag icon. Below the field, it says 'Select a region close to your users for the best performance.'
- Pricing Plan:** A dropdown menu showing 'Free - \$0/month'. Below the field, it says 'Select a plan that suits your needs. [More details](#)'.

At the bottom of the form, there is a 'Cancel' button, a note 'You can rename your project later', and a green 'Create new project' button.

Figura 3 - Criando conta no Supabase
Fonte: Elaborado pelo autor (2024)

Clique em **Create new project** e espere alguns minutos para que o seu projeto seja criado. Em seguida, clique em **Settings** > **Database** para copiar os dados de conexão da sua base de dados:

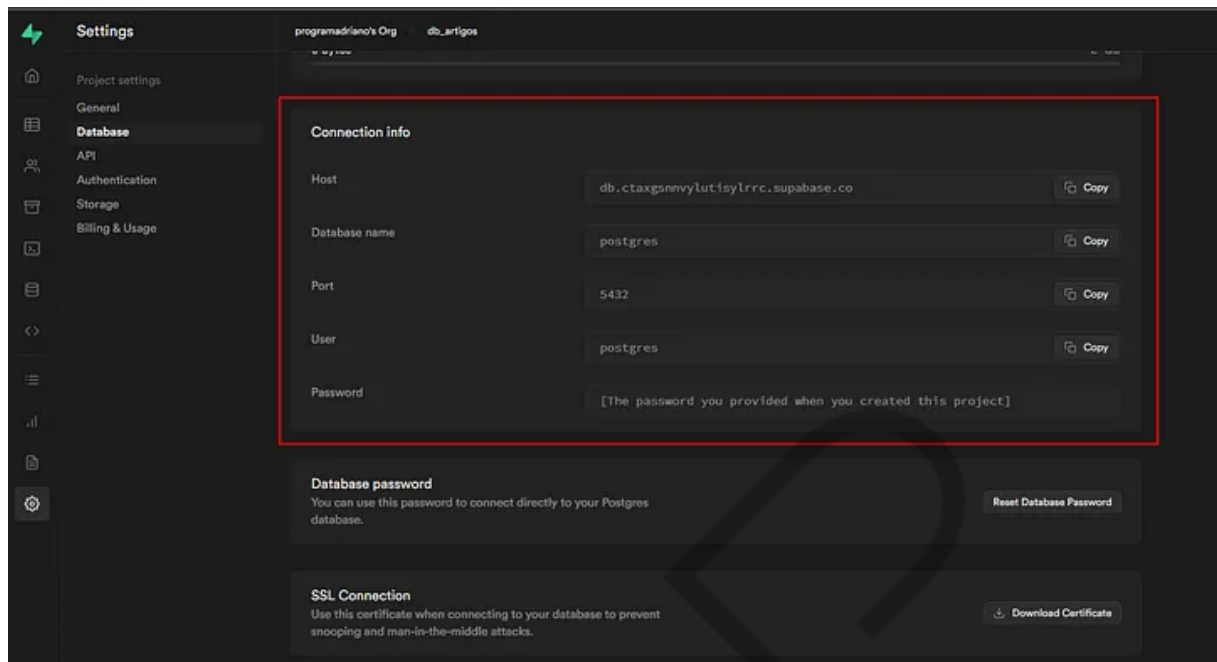


Figura 4 - Finalização da criação da conta
Fonte: Elaborado pelo autor (2024)

Com o projeto criado, o próximo passo será a instalação do pgAdmin4. Para isso, [acesse a página de download do pgAdmin](#), escolha um instalador, conforme o seu sistema operacional, e em seguida selecione padrão Windows, next, next e finish.

Antes de finalizarmos esta aula, vamos a uma rápida introdução sobre o que é o banco de dados RDS da AWS.

O Amazon Relational Database Service (RDS) é um serviço de banco de dados gerenciado pela AWS. Ele oferece uma solução fácil de usar para configurar, operar e escalar bancos de dados relacionais na nuvem, sem a necessidade de gerenciar a infraestrutura subjacente.

Principais Características do Amazon RDS:

- **Multi-Engine Support:** o RDS suporta vários motores de banco de dados relacionais, incluindo MySQL, PostgreSQL, MariaDB, Oracle e Microsoft SQL Server.
- **Gerenciamento automático:** o RDS gerencia automaticamente tarefas operacionais, como provisionamento de hardware, aplicação de patches de software, backups regulares e monitoramento de desempenho.

- Backup e recuperação automáticos: o RDS oferece backups automáticos diários e retenção configurável, permitindo restaurar facilmente o banco de dados para um ponto específico no tempo.
- Alta disponibilidade: configurações de alta disponibilidade, como Multi-AZ deployments, estão disponíveis para melhorar a resiliência do sistema. Isso envolve a replicação automática de dados para uma instância secundária em uma zona de disponibilidade diferente.
- Escalabilidade vertical e horizontal: é possível ajustar a capacidade computacional da instância do banco de dados (escalabilidade vertical) ou adicionar réplicas de leitura para distribuir a carga (escalabilidade horizontal).
- Segurança: o RDS oferece recursos de segurança robustos, incluindo criptografia em repouso e em trânsito, controle de acesso baseado em IAM (Identity and Access Management) e a capacidade de implantar bancos de dados em uma VPC (Virtual Private Cloud) para maior isolamento.
- Monitoramento e métricas: o RDS fornece métricas detalhadas de desempenho e eventos por meio do Amazon CloudWatch, permitindo monitorar e ajustar a configuração conforme necessário.

Depois desta rápida introdução sobre o Amazon RDS, vamos criar o nosso banco de dados PostgreSQL.

O primeiro passo será acessar a [página do nível gratuito do Amazon RDS](#) e criar uma conta na AWS, caso ainda não tenha uma. Se já possuir uma conta, pesquise na tela principal por RDS, como mostra a figura 5.

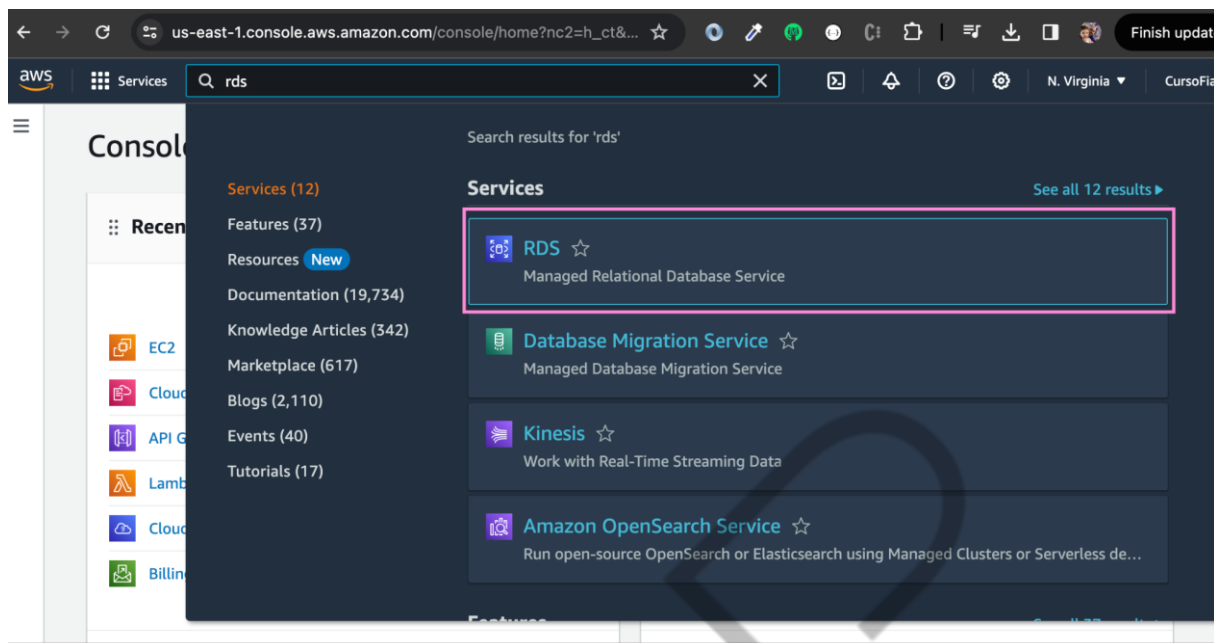


Figura 5 - Pesquisa por RDS
Fonte: Elaborado pelo autor (2024)

Em seguida, clique em **Create Database**, conforme destacado na figura 6.

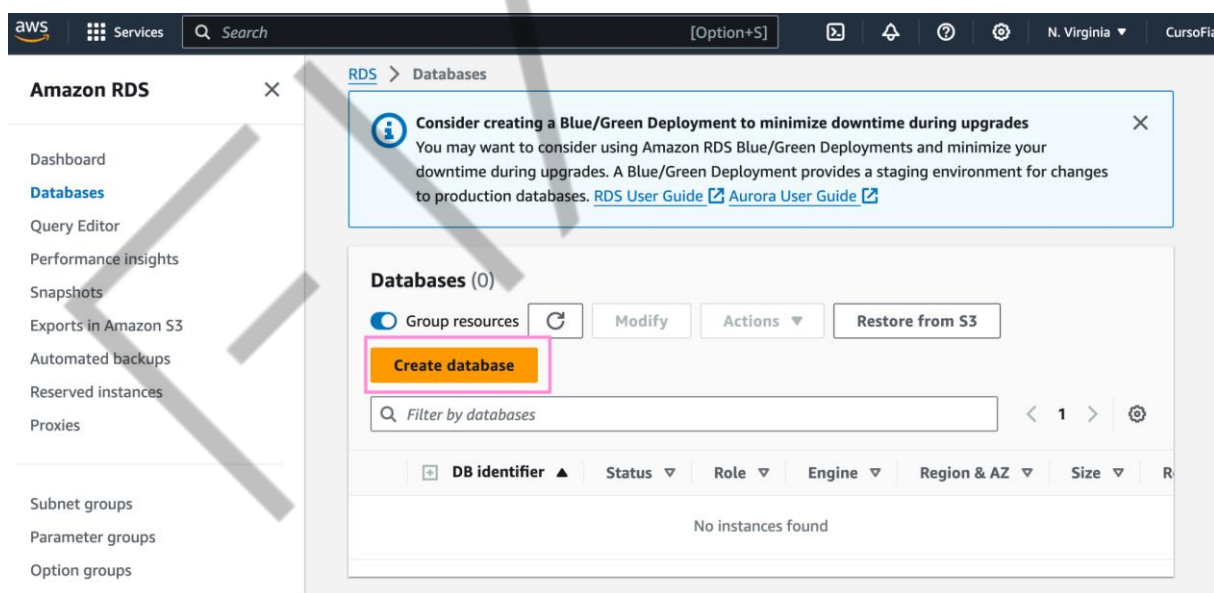


Figura 6 - Criando RDS
Fonte: Elaborado pelo autor (2024)

Na próxima tela, selecione:

- Easy Create: esta opção utiliza configurações recomendadas. Você poderá fazer alterações depois de criar o banco de dados.

- Selecione o banco **PostgreSQL** e depois a camada **free-tier**, para não ter custo com essa criação.
- Depois, preencha com os dados que você deseja para o seu banco de dados, como nome do banco, usuário, etc.

A figura 7 demonstra a minha configuração para essa etapa:

Region.

curso-fiap

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Master username [Info](#)

Type a login ID for the master user of your DB instance.

tadriano

1 to 16 alphanumeric characters. The first character must be a letter.

☐ Auto generate a password

Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

.....

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm master password [Info](#)

.....

Figura 7 - Criando banco de dados PostgreSQL
Fonte: Elaborado pelo autor (2024)

Por fim, clique em **Create Database** para criar a sua instância. Esse processo pode demorar alguns minutos.

Com a sua instância criada, clique em **View connection details**, conforme mostra a figura 8.

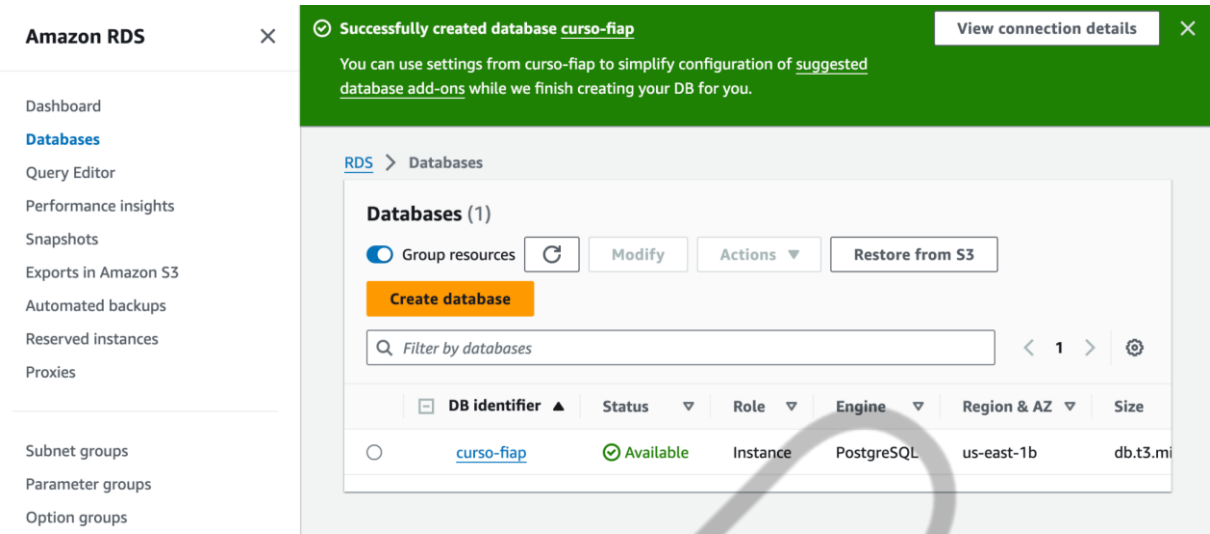


Figura 8 - Acessando banco de dados PostgreSQL
Fonte: Elaborado pelo autor (2024)

Esse passo vai abrir uma modal com as informações de conexão do seu banco de dados.

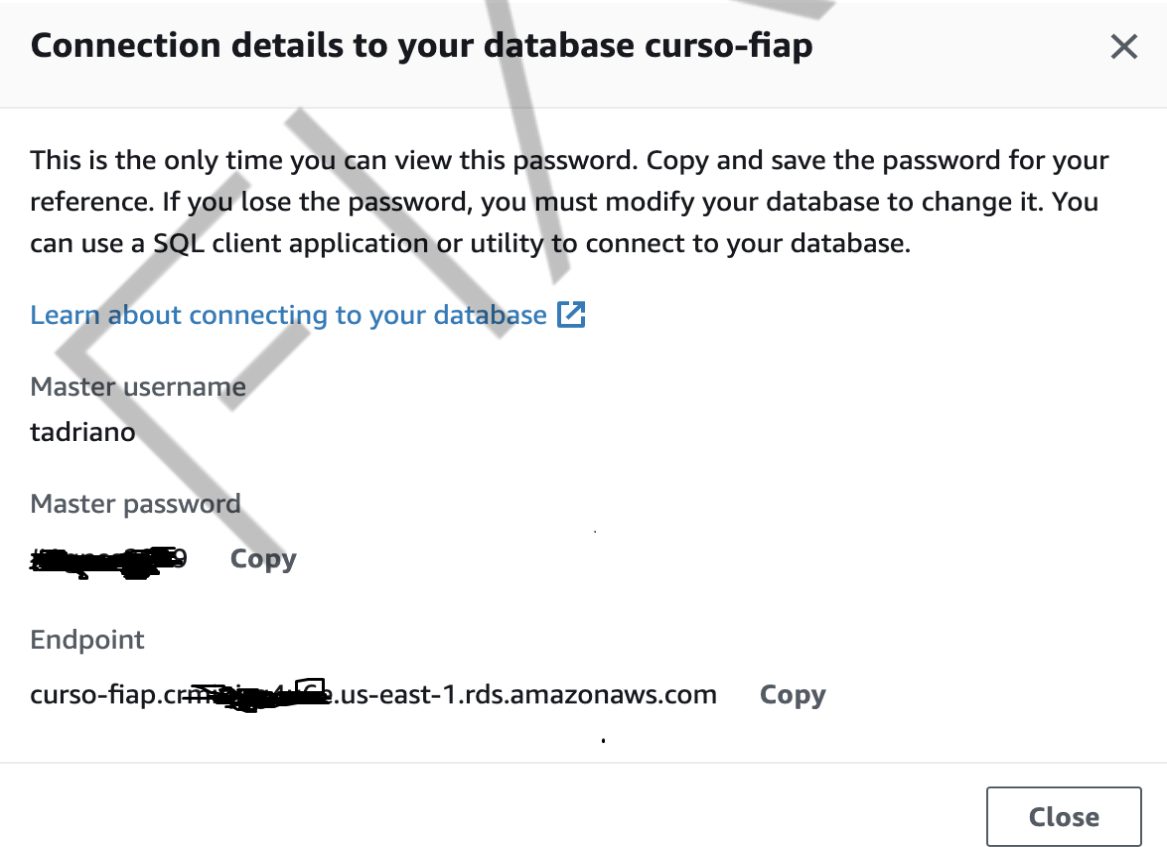


Figura 9 — Dados de conexão com banco de dados PostgreSQL
Fonte: Elaborado pelo autor (2024)

Agora podemos nos conectar ao nosso banco de dados utilizando o pgAdmin4. Caso não tenha ele instalado, você pode [baixar uma versão conforme o seu sistema operacional](#).

EXEMPLO

O QUE VOCÊ VIU NESTA AULA?

Durante essa aula, você ganhou habilidades práticas que o(a) capacitam a criar e gerenciar eficientemente bancos de dados relacionais usando o PostgreSQL. Esses conhecimentos são essenciais para pessoas desenvolvedoras e profissionais de banco de dados que precisam lidar com sistemas dependentes de dados estruturados.

EMANSP

REFERÊNCIAS

AMAZON AWS. **O que é o SQL (linguagem de consulta estruturada)?** [s.d.]. Disponível em: <https://aws.amazon.com/pt/what-is/sql/>. Acesso em: 08 mar. 2024.

MASTER DAWEB. **O que é PostgreSQL?** 2023. Disponível em: <https://masterdaweb.com/blog/o-que-e-postgresql/>. Acesso em: 18 mar. 2024.

DIGITALOCEAN. **Como Instalar e Configurar o pgAdmin 4 no Modo Servidor.** 2018. Disponível em: <https://www.digitalocean.com/community/tutorials/como-instalar-e-configurar-o-pgadmin-4-no-modo-servidor-pt>. Acesso em: 18 mar. 2024.

ADRIANO, T. S. **[Dica rápida]Conhecendo o Relational Migrator.** 2024. Disponível em: <https://programadriano.medium.com/dica-r%C3%A1pida-conhecendo-o-relational-migrator-956dddc7e5e>. Acesso em: 14 jun. 2024.

ADRIANO, T. S. **Comandos básicos Docker.** 2017. Disponível em: <https://programadriano.medium.com/principais-comandos-docker-f9b02e6944cd>. Acesso em: 14 jun. 2024.

ADRIANO, T. S. **Comparando os termos utilizados no NoSQL com SQL.** 2024. Disponível em: <https://medium.com/xp-inc/comparando-os-termos-utilizados-no-nosql-com-sql-e862788e2374>. Acesso em: 14 jun. 2024.

ADRIANO, T. S. **Trabalhando com AWS em um ambiente localhost.** 2024. Disponível em: <https://dev.to/programadriano/trabalhando-com-aws-em-um-ambiente-localhost-20i8>. Acesso em: 14 jun. 2024.

AMAZON. **Download NoSQL Workbench for DynamoDB.** 2024. Disponível em: <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/workbench.settingup.html>. Acesso em: 14 jun. 2024.

AMAZON.COM. **Nível gratuito do Amazon RDS.** 2024. Disponível em: <https://aws.amazon.com/pt/rds/free/>. Acesso em: 14 jun. 2024.

DIAS, L. A. Grafos, teoria e aplicações. 2019. Disponível em: <https://medium.com/xp-inc/grafos-teoria-e-aplica%C3%A7%C3%B5es-2a87444df855>. Acesso em: 14 jun. 2024.

GITHUB. **Criação do container postgres.** 2024. Disponível em: <https://gist.github.com/programadriano/2171fe63e6a2c1b08985bccf263fe980>. Acesso em: 14 jun. 2024.

GITHUB. **Docker + Neo4J.** 2024. Disponível em: <https://gist.github.com/programadriano/0224c1a1d389af25177eb397d912dbe7>. Acesso em: 14 jun. 2024.

GITHUB. **Estrutura**. 2024. Disponível em: https://app.diagrams.net/#G1_-jdpuomwUwU3KD01rSRGepNwA3kYA9X%7B%22pageld%22%3A%22apt8INFUySjbP2altylb%22%7D. Acesso em: 14 jun. 2024.

GITHUB. **Exemplo prático Neo4j (1)**. 2024. Disponível em: <https://gist.github.com/programadriano/01508490e0f53745f2778a045e1cc070>. Acesso em: 14 jun. 2024.

GITHUB. **Exemplo prático Neo4j (2)**. 2024. Disponível em: <https://gist.github.com/programadriano/e22931ff2bc032dcc68498bc3597d1a7>. Acesso em: 14 jun. 2024.

GITHUB. **Exemplo prático Neo4j (3)**. 2024. Disponível em: <https://gist.github.com/programadriano/8c8e7ea613acc4e854a7f2ffe9f44111>. Acesso em: 14 jun. 2024.

GITHUB. **Query X Scan**. 2024. Disponível em: <https://gist.github.com/programadriano/8169d065da7e0cf8494435ca8caca70>. Acesso em: 14 jun. 2024.

GITHUB.COM. **Carga e Fluxo Neo4J (1)**. 2024. Disponível em: <https://gist.github.com/programadriano/e7c314539571c811a48940deca8c8491>. Acesso em: 14 jun. 2024.

GITHUB.COM. **Carga e Fluxo Neo4J (2)**. 2024. Disponível em: <https://gist.github.com/programadriano/c70281293a5720f1ab8719d65fe5e410>. Acesso em: 14 jun. 2024.

IBM. **O que é o Docker?** 2024. Disponível em: <https://www.ibm.com/br-pt/topics/docker>. Acesso em: 14 jun. 2024.

LOCALSTACK.CLOUD. **Abrir e mostrar o ambiente rodando**. 2024. Disponível em: <https://app.localstack.cloud/dashboard>. Acesso em: 14 jun. 2024.

LOCALSTACK.CLOUD. **LocalStack Desktop**. 2024. Disponível em: <https://app.localstack.cloud/download>. Acesso em: 14 jun. 2024.

MIRO. **Fluxo Miro**. 2024. Disponível em: https://miro.com/welcomeonboard/Mzh1bEhNczFWeWtFeFpUS0tXeIJIQ1RjZnJjZHhmbmhFU0kyd3RjenpjZVhTdVZqajJPN3NxdFpuR3lkMWc5NHwzNDU4NzY0NTY0Ng5NTc4MjlfDI=?share_link_id=113865211181. Acesso em: 14 jun. 2024.

MONGODB. **MongoDB Relational Migrator Download**. 2024. Disponível em: <https://www.mongodb.com/try/download/relational-migrator>. Acesso em: 14 jun. 2024.

NEO4J. **Exploring LinkedIn in Neo4j**. 2013. Disponível em: <https://neo4j.com/blog/exploring-linkedin-in-neo4j/>. Acesso em: 14 jun. 2024.

NEO4J. **Use Cases: Social Media and Social Network Graphs**. 2024. Disponível em: <https://neo4j.com/use-cases/social-network/>. Acesso em: 14 jun. 2024.

PGADMIN.ORG. **PgAdmin4.** 2024. Disponível em:
<https://www.pgadmin.org/download/pgadmin-4-windows/>. Acesso em: 14 jun. 2024.

EMSE

PALAVRAS-CHAVE

Palavras-chave: pgAdmin. SQL. PostgreSQL. RDS. AWS.

EMSE



POSTECH