

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
NOME DA UNIDADE ACADÊMICA
CAMPUS DE PONTA PORÃ

PEDRO ANTÔNIO PENHA, VINÍCIUS PAZ DE SOUZA

BANCOS DE DADOS RELACIONAIS E NÃO RELACIONAIS:
UMA ANÁLISE COMPARATIVA E GUIA DE INSTALAÇÃO E USO

Ponta Porã
2024

PEDRO ANTÔNIO DA PENHA, VINÍCIUS PAZ DE SOUZA

BANCOS DE DADOS RELACIONAIS E NÃO RELACIONAIS:
UMA ANÁLISE COMPARATIVA E GUIA DE INSTALAÇÃO E USO

Trabalho de Conclusão de Curso
apresentado à Faculdade da Universidade
Federal de Mato Grosso do Sul como
requisito parcial para obtenção do título
de bacharel em Sistemas de Informação

Área de concentração: Bancos de dados

Orientador: Robson Soares da Silva

Ponta Porã
2024

RESUMO

No final do século XX houve um aumento significativo da quantidade de informação sendo gerada a todos os instantes no mundo inteiro, forçando a criação de uma nova forma de armazenar e gerenciar essa informação. Com isso vieram os bancos de dados, que rapidamente se popularizaram devido a sua praticidade, eficácia e adaptabilidade, permitindo que qualquer possa organizar quantias absurdas de informação de forma rápida e fácil. Esse trabalho busca trazer uma explicação sucinta sobre diferentes tipos de bancos de dados e como utilizá-los. A metodologia deste trabalho se baseou no estudo de diversos artigos e periódicos sobre o assunto, além de testes relacionado à implementação tanto do PostgreSQL quanto do MongoDB, dois sistemas de gerenciamento de banco de dados. O estudo indicou que apesar das diversas vantagens dos bancos de dados NoSQL sobre os SQL, ambos tem seus usos e podem coexistir dentro do mercado, havendo até mesmo situações onde eles podem ser utilizados em conjunto, aproveitando ao máximo as forças de cada um.

Palavras-chave: Bancos de dados. Evolução tecnológica. Sistemas de informação. Tutorial.

ABSTRACT

At the end of the 20th century, there was a significant increase in the amount of information constantly being generated worldwide, prompting the creation of new ways to store and manage this information. This led to the development of databases, which quickly became popular due to their practicality, efficiency, and adaptability, enabling anyone to organize massive amounts of data in a quick and easy way. This paper aims to provide a concise explanation of different types of databases and how to use them. The methodology of this work was based on the study of various articles and journals on the topic, as well as testing related to the implementation of both PostgreSQL and MongoDB, two database management systems. The study indicated that despite the various advantages of NoSQL databases over SQL, both have their uses and can coexist in the market, with situations where they may even be used together, maximizing the strengths of each of them.

Keywords: Databases. Technological evolution. Information systems. Tutorial.

1. INTRODUÇÃO

O controle e transmissão de informação sempre foi uma das características mais importantes do ser humano e um dos fatores que nos diferencia dos outros animais e nos permite ser a raça dominante do nosso planeta. Essa informação que antes era transmitida de forma oral e posteriormente por meio da escrita hoje em dia é acumulada das mais diversas formas, sendo necessária uma forma mais eficaz de gerenciar esses dados.

Não só o acesso a essa informação se tornou muito mais importante, mas também a quantidade de informação disponível aumentou de forma exponencial nas últimas décadas, sendo necessário que a nossa forma de acesso e manipulação desses dados evoluísse também. Com essa necessidade, surgiram os bancos de dados, que nos permitem controlar grandes quantidades de informação de forma simples e confiável, além de nos garantir o acesso a elas de forma remota e instantânea.

Inovações como o surgimento e popularização de tecnologias de gravação de vídeos e áudios, as novas formas de coletas de informações em todas as formas de ciências e a da internet fizeram com que seja inviável o uso de armazenamento de informações em mídias físicas, popularizando o uso do banco de dados de uma forma que, hoje em dia, até pequenos estabelecimentos podem precisar de um banco de dados para se organizar.

Portanto, o entendimento e conhecimento sobre como funcionam esses bancos de dados é crucial para qualquer um que queira otimizar suas formas de acesso a qualquer tipo de informação. Além de ser um requerimento para o desenvolvimento profissional daqueles que buscam uma carreira dentro da análise e de dados, uma área que está cada vez mais atrativa dentro do mercado de trabalho.

Esse artigo busca então trazer de forma resumida o funcionamento desses bancos de dados e ensinar a manuseá-los, trazendo o passo a passo da instalação dos softwares necessários e criação de um banco de dados básico em dois programas de sistemas de bancos de dados diferentes, sendo eles o PostgreSQL e o MongoDB.

2. O QUE SÃO BANCOS DE DADOS

Segundo Kosinski (2024), bancos de dados são repositórios digitais com o propósito de armazenar, gerenciar e proteger coleções de dados organizados. A inclusão do termo “digital” é importante por nos permitir traçar uma linha na história da humanidade, pois grandes repositórios de informação sempre existiram, sendo a Biblioteca de Alexandria um dos exemplos mais notórios, mas com o surgimento da computação, a escala de dados sendo gerada propiciou uma evolução no que consideramos um banco de dados.

Os primeiros bancos de dados modernos remontam à década de 60, onde os sistemas de gerenciamento de arquivos eram hierárquicos, ou seja, os dados seguiam um relacionamento “pai-filho” e eram organizados em uma estrutura de árvore. Apesar de inovador para a época, esses modelos eram extremamente inflexíveis devido a sua simplicidade e são considerados obsoletos nos dias de hoje.

Com o passar das décadas e a evolução das capacidades computacionais de nossos aparelhos, foram surgindo outros modelos de bancos de dados, cada um com suas vantagens e desvantagens, além de propósitos e funções específicas a cada um, sendo os dois mais comuns os relacionais e não relacionais, alvos primários desse trabalho.

Qualquer banco de dados precisa de um sistema de gerenciamento conhecido como DBMS (*Data Management System*), uma ferramenta de software que permite criar, manipular e gerenciar esses bancos de dados através de uma interface simples e segura, garantindo a integridade das informações. Exemplos populares desses DBMS são o PostgreSQL e o MongoDB, que serão abordados mais profundamente nesse trabalho.

A adoção dos bancos de dados pelo mundo da computação foi previsivelmente universal, é inegável a praticidade e confiabilidade que esses sistemas trazem para todas as áreas de interesse humano, desde análises e estudos científicos até as tarefas mais mundanas, como o gerenciamento de estoque dentro de um supermercado, o impacto que a invenção dos bancos de dados trouxe para o mundo é sentida por todos.

3. BANCOS DE DADOS RELACIONAIS

Inventado em 1970 pelo matemático britânico Edgar Codd, que publicou o artigo “*A Relational Model of Data for Large Shared Data Banks*”, o modelo relacional revolucionou a forma como os dados eram organizados e acessados, dando início à era dos bancos de dados relacionais dentro das ciências da computação. Codd é amplamente reconhecido como o “pai do SQL”, recebendo até mesmo um prêmio Turing pela sua contribuição.

Bancos de dados relacionais armazenam os dados em tabelas com colunas e linhas, onde as colunas representam os atributos dos dados e as linhas representam uma instância desses dados, permitindo que os dados sejam organizados de maneira lógica, facilitando consultas e a manipulação de grandes volumes de informações. Cada tabela possui uma chave primária que identifica exclusivamente cada registro e pode conter chaves estrangeiras para criar relações com outras tabelas, promovendo a integração entre os dados. Exemplos populares desse modelo atualmente no mercado são Oracle, MySQL, Microsoft SQL e PostgreSQL.

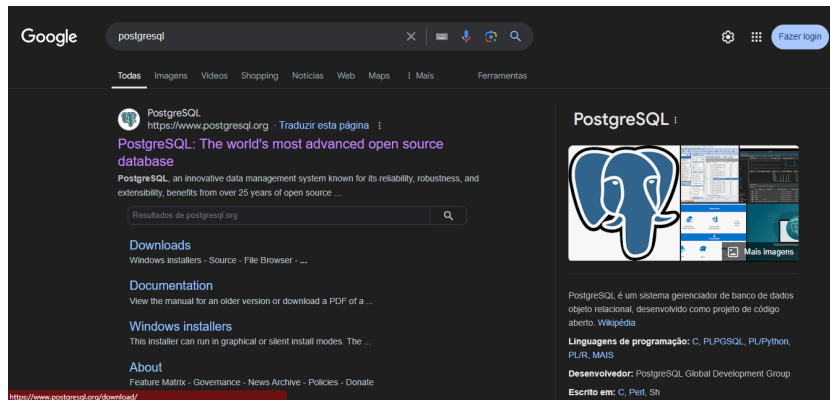
3.1 POSTGRESQL

Originário da Universidade da Califórnia na cidade de Berkeley nos Estados Unidos, o PostgreSQL foi concebido no ano de 1986 pelo professor Michael Stonebreaker e sua equipe. Frustrado com as limitações do banco de dados Ingres, Stonebreaker iniciou o projeto Postgres (*Post* Ingres) com o objetivo de suportar estruturas mais complexas e personalizáveis dentro de seus bancos de dados, como objetos e arrays.

Quase uma década depois, no ano de 1996, o projeto foi adaptado para suportar a linguagem SQL, ganhando o nome atual de PostgreSQL. Esse sistema de código aberto rapidamente se popularizou, pois permite que desenvolvedores e empresas personalizem sua implementação de acordo com necessidades específicas, sendo adotado por grandes empresas como a IBM e a Microsoft e é considerado um dos sistemas de bancos de dados relacionais mais confiáveis do mercado.

3.1.1 INSTALANDO O POSTGRESQL

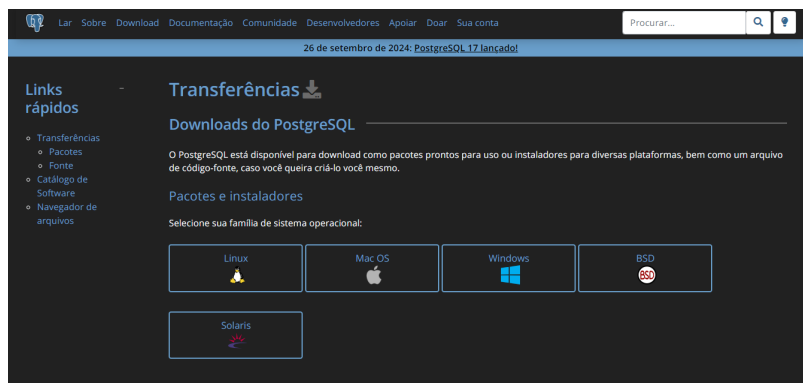
1º Passo: Acessar o site oficial do PostgreSQL



2º Passo: Acessar a página de Download do PostgreSQL



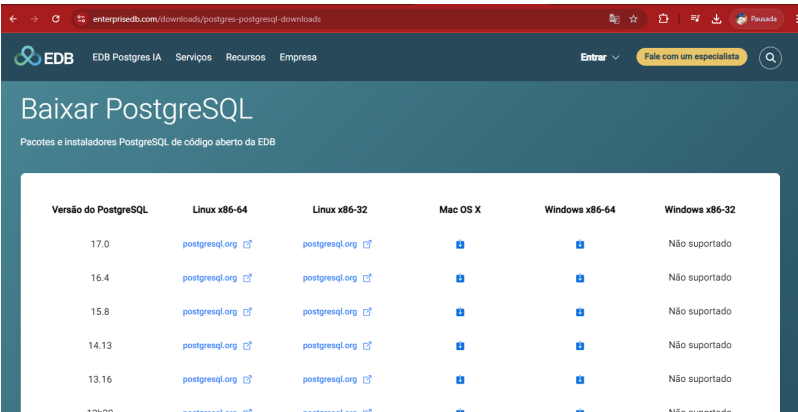
3º passo: Selecionar o seu sistema operacional. Neste caso será usado o Windows



4º passo: Selecionar a opção que permite que todas as versões do PostgreSQL sejam suportadas



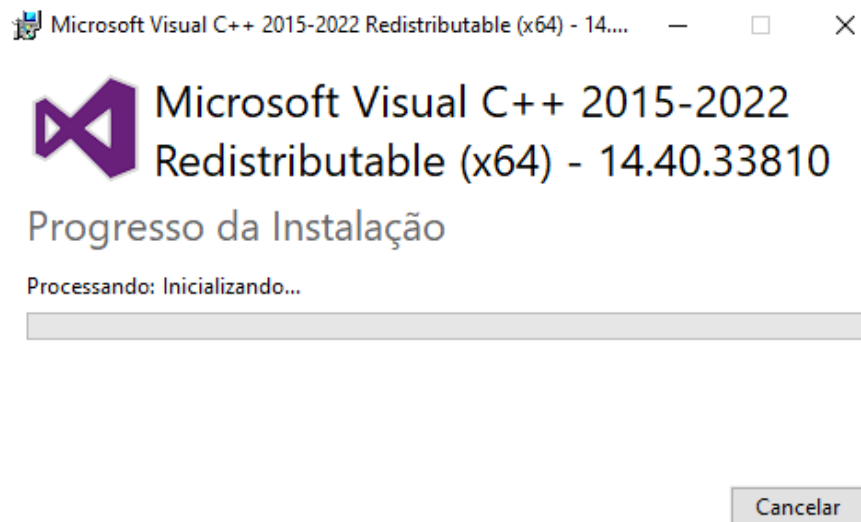
5º Passo: Escolher a versão compatível com o seu dispositivo



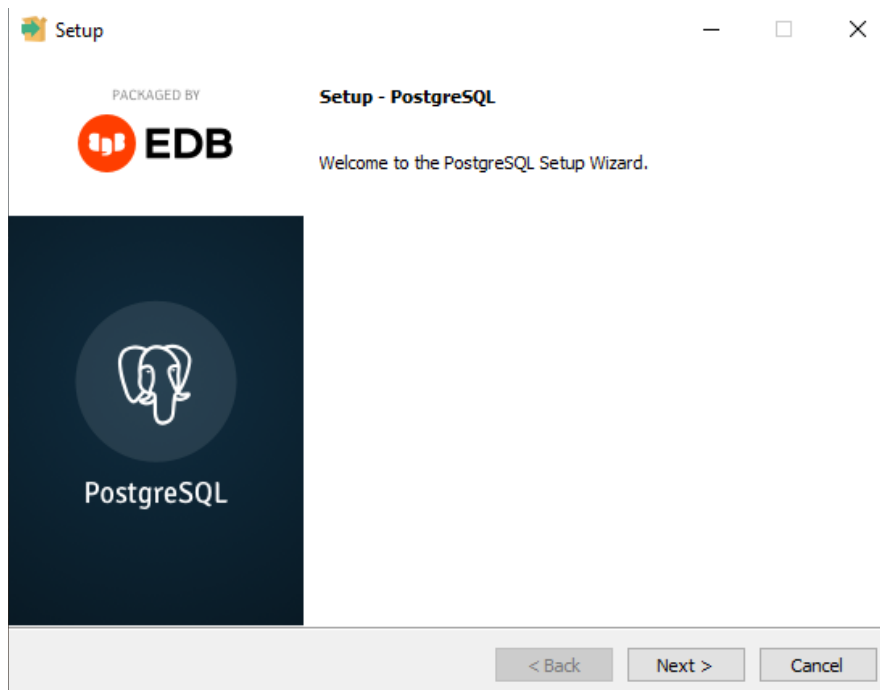
6º passo: Executar o instalador



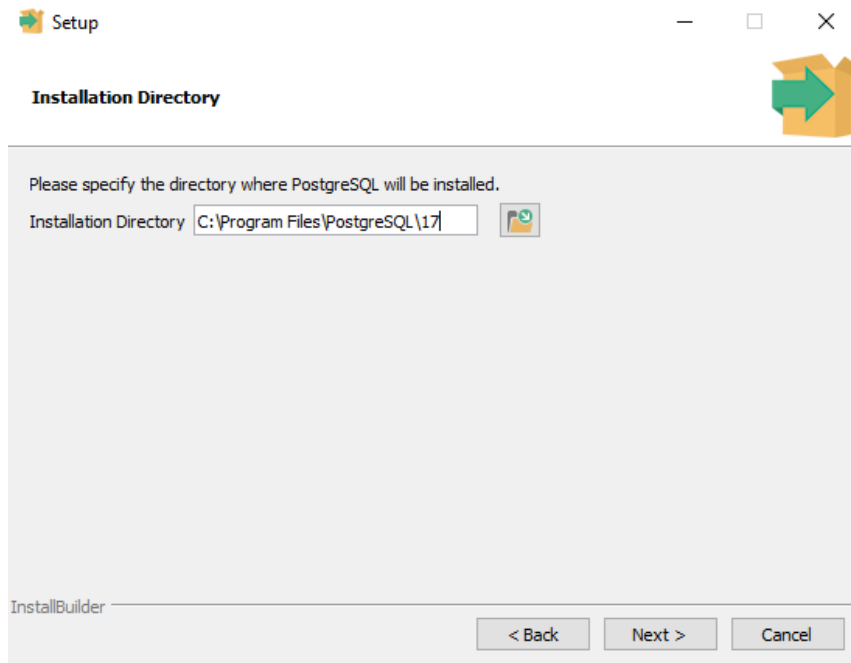
7º passo: Realizar a inicialização de seu instalador



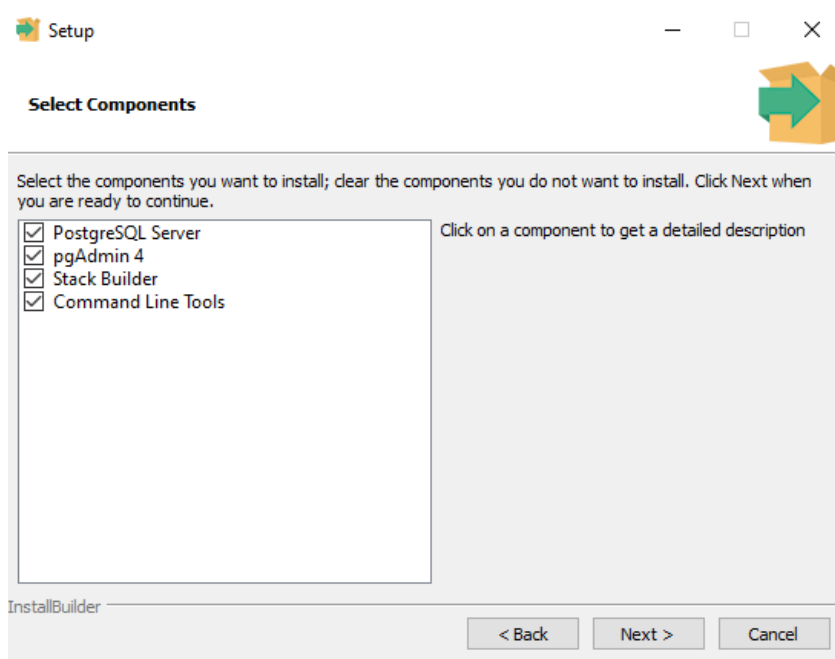
8º passo: Pressionar o botão "Next >", para ir para próxima etapa da instalação



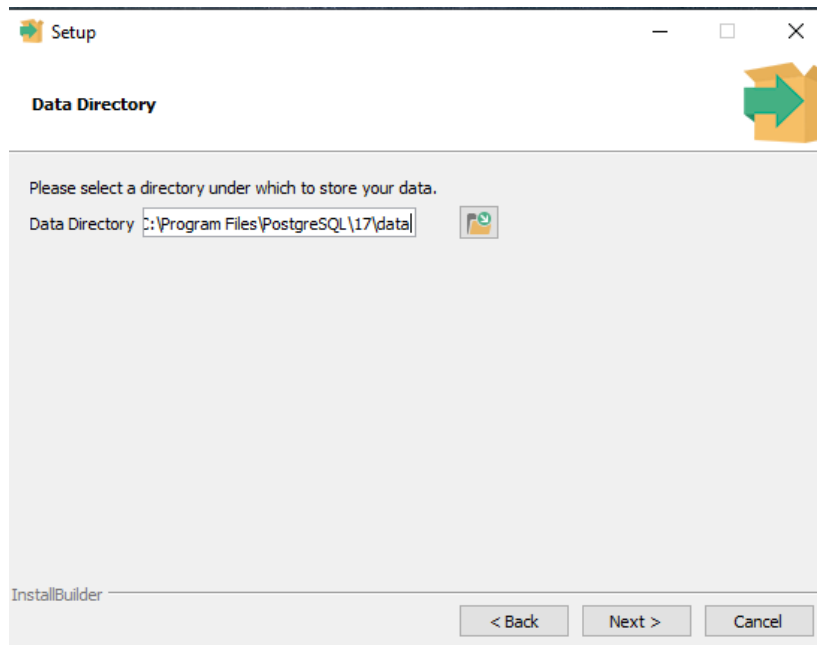
9º Passo: Escolher o diretório de onde será instalado o programa no seu computador



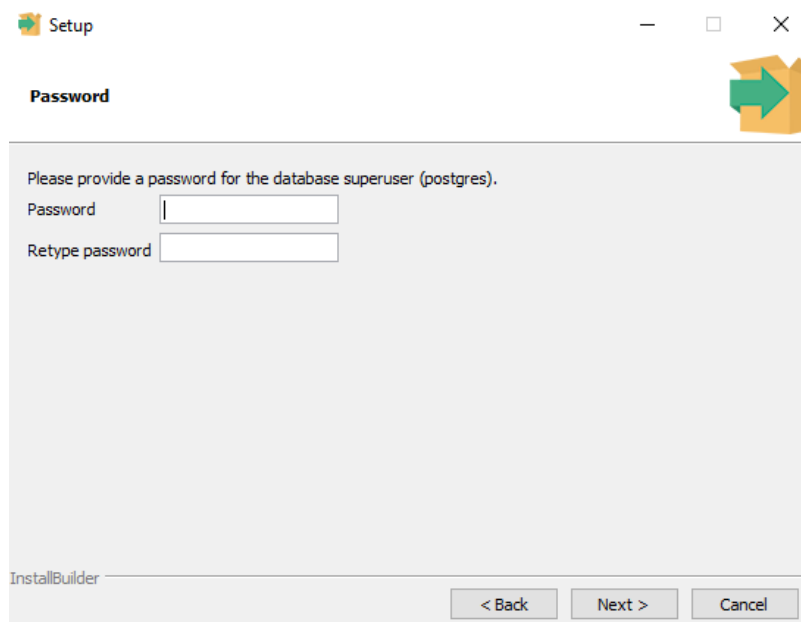
10º Passo: Selecionar todas as opções e pressionar o botão “Next >”



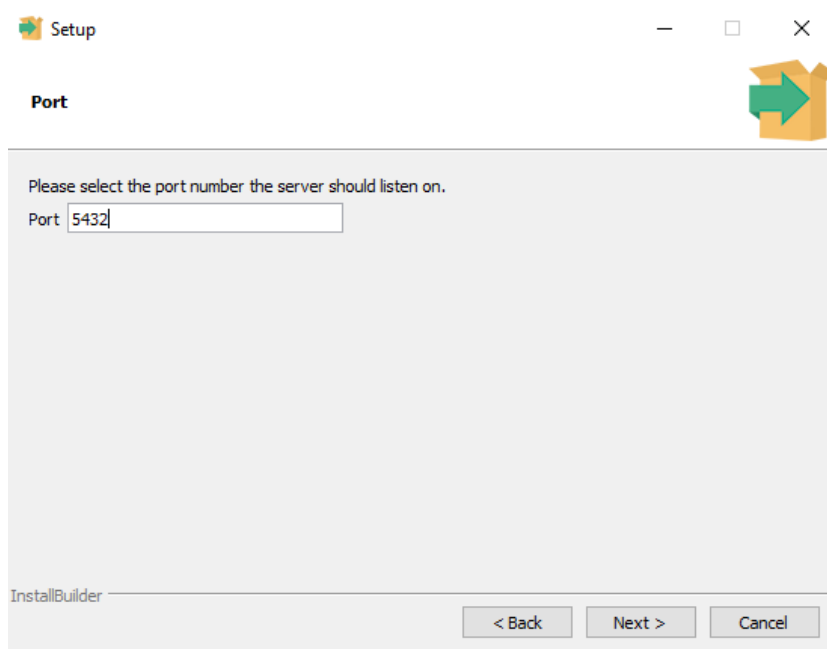
11º Passo: Pressionar o botão “Next” novamente



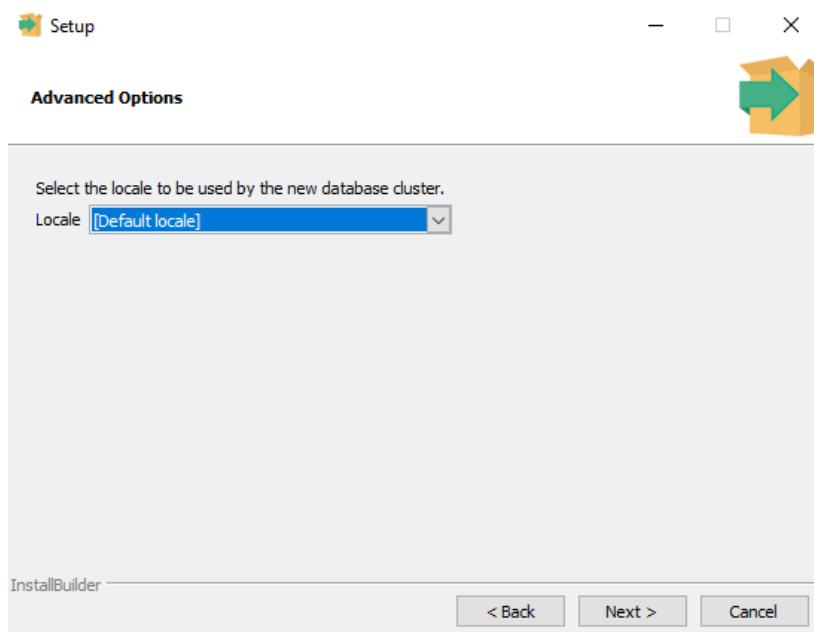
12º Passo: Configurar uma senha de segurança para seu PostgreSQL



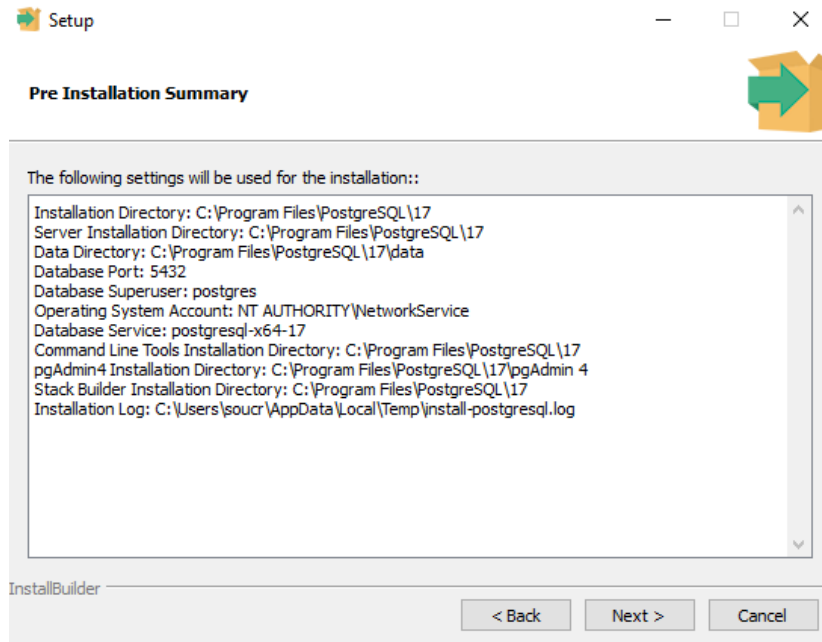
13º Passo: Neste passo ele mostra uma configuração de porta no qual serve para o resumo das informações do PostgreSQL



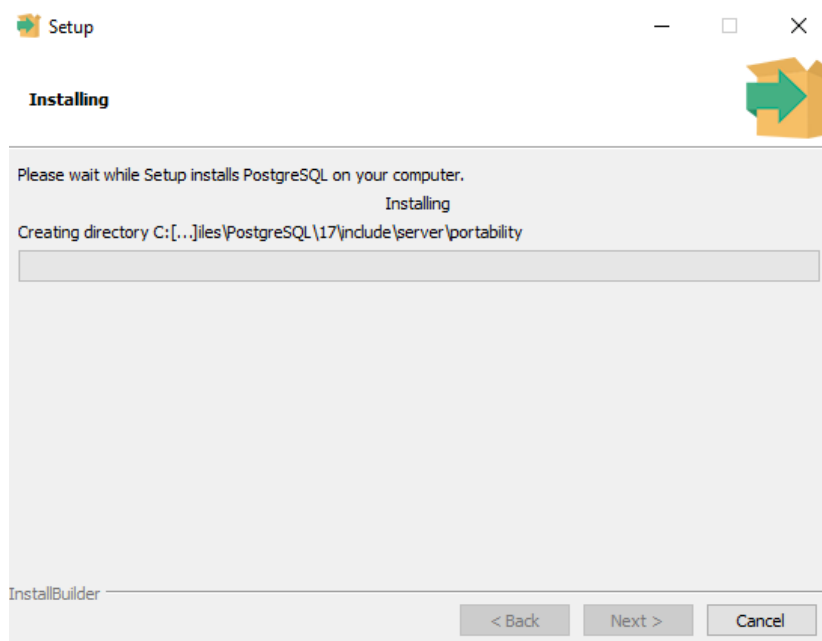
14º Passo: Pressionar o botão “Next >”



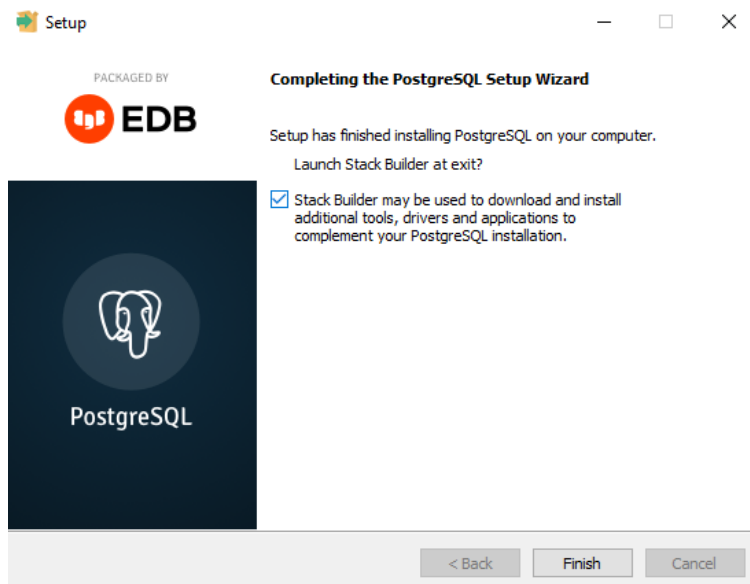
15º passo: Pressionar o botão “Next >” novamente



16º passo: Aguardar a instalação



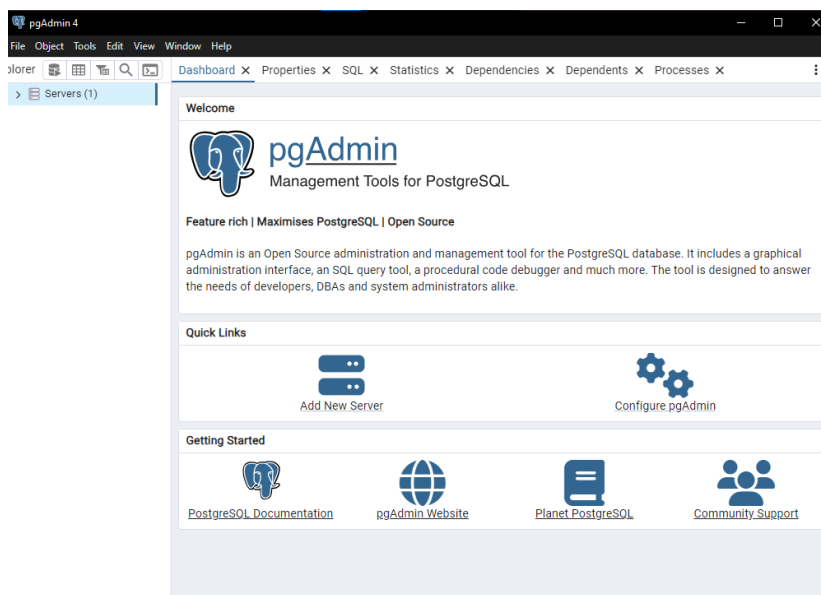
17º passo: Desmarcar a opção disponível e finalizar a instalação



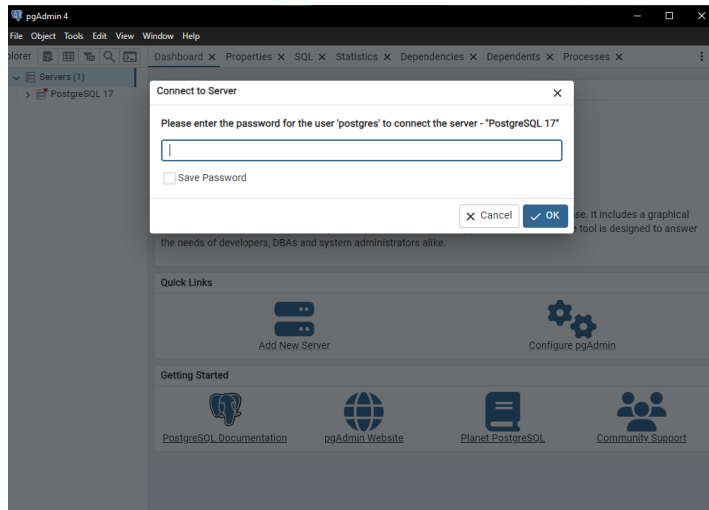
3.1.2 USANDO O POSTGRESQL

A seguir está um tutorial de como editar tabelas dentro do PostgreSQL

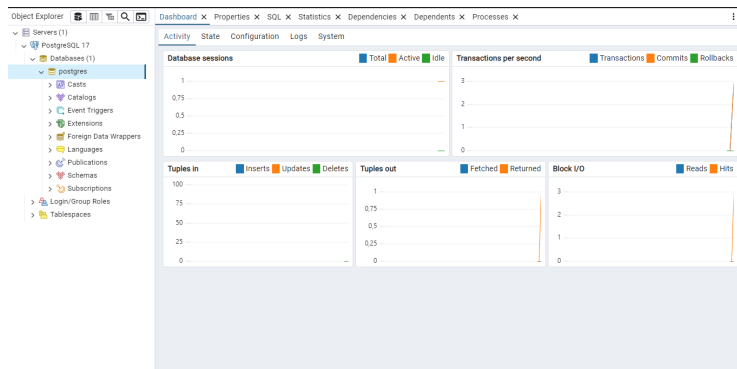
Na tela inicial, clique em “Add New Server”:



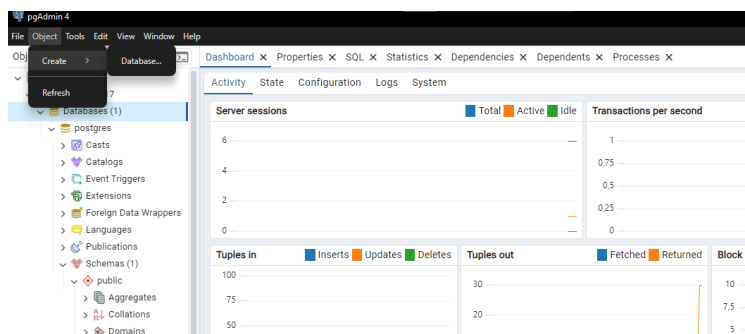
Será pedido para que o usuário crie uma senha:



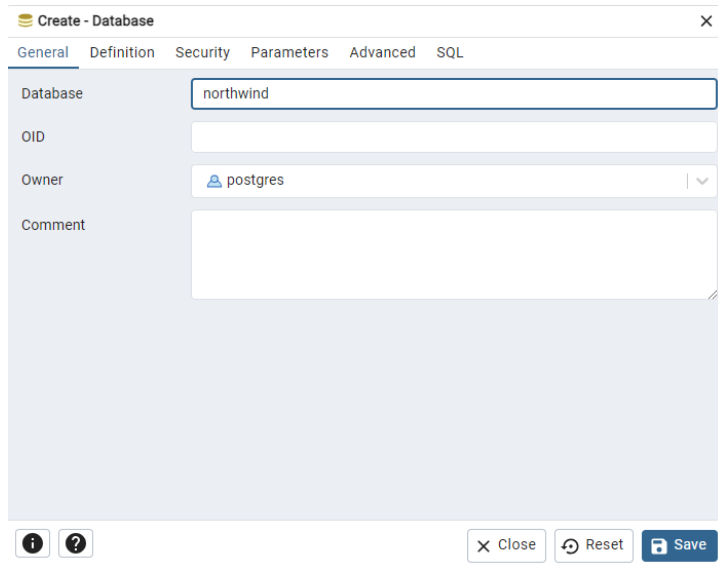
No lado esquerdo da tela, expandir as opções “Postgres”, “Databases” e “Schemas” permitirá ver as tabelas que serão criadas:



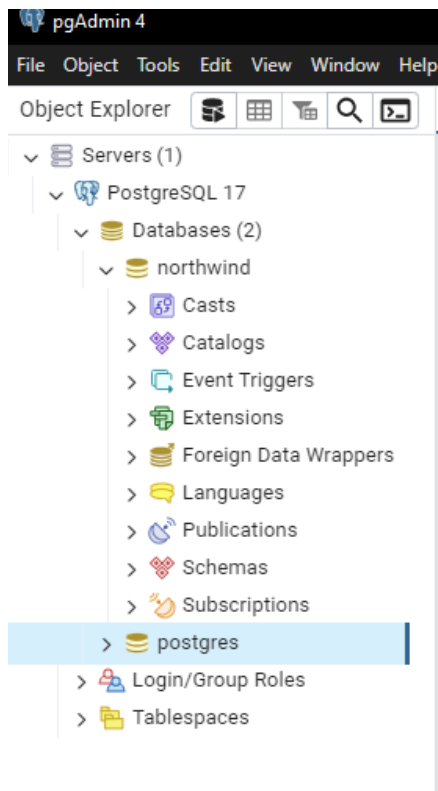
Para criar um banco de dados clique em “Object”, “Create” e depois em “Database”:



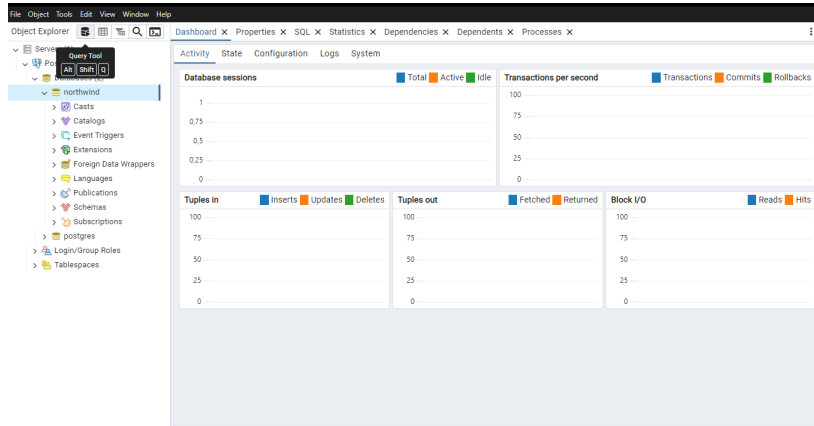
Defina o nome do seu banco de dados e pressione “Save”.



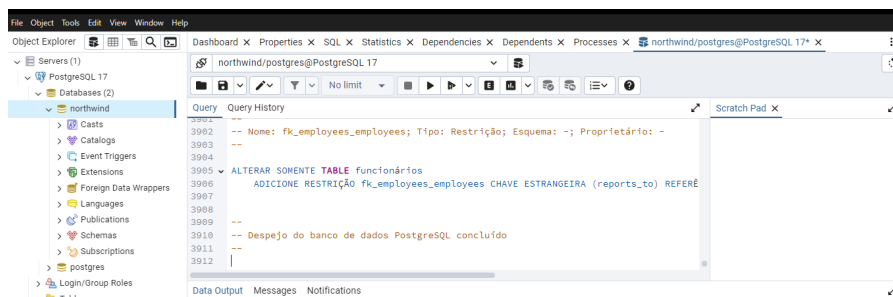
O banco de dados deve aparecer no lado esquerdo da tela:



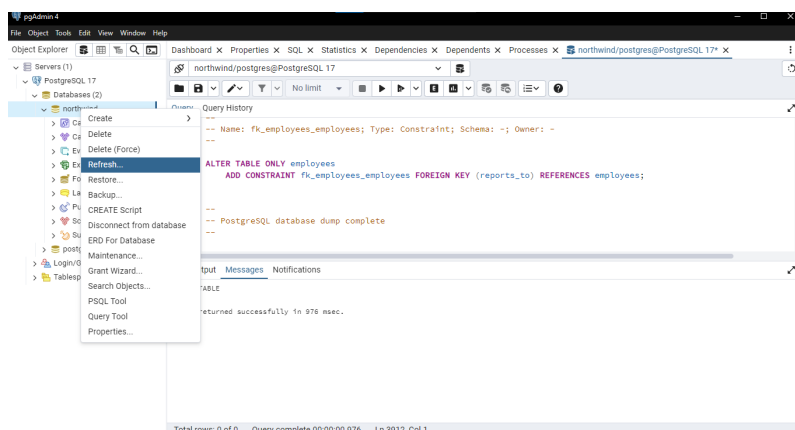
Para criar a tabela, selecione a opção “Query Tool” ou use o comando Alt+Shift+Q:



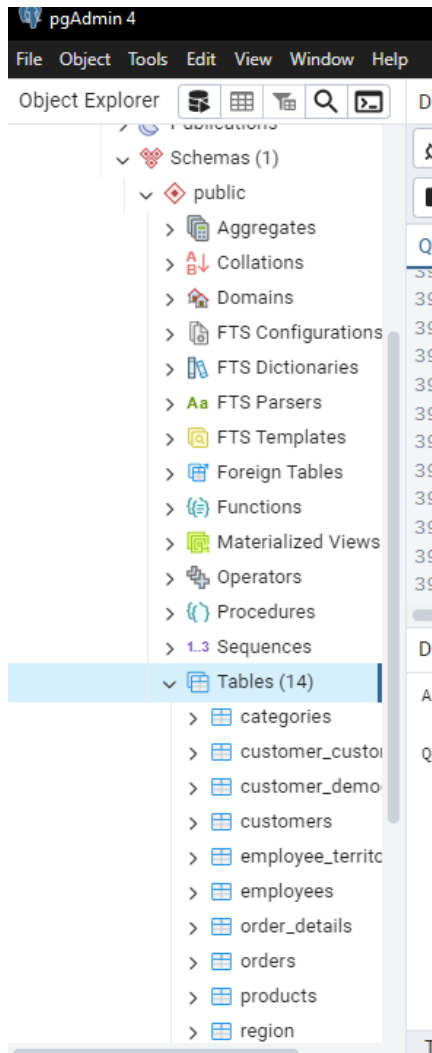
Insira e execute o código desejado, para este exemplo foram utilizadas as tabelas do banco de dados da Northwind que contém 14 tabelas sobre uma empresa fictícia no ramo de vendas de mercadoria no modelo B2B (*Business to Business*). É um banco de dados da Microsoft para ser utilizado no mysql server:



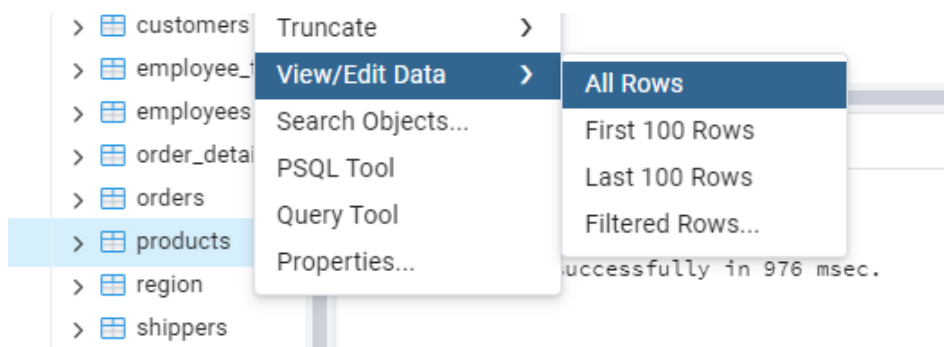
O terminal deverá retornar que o comando foi um sucesso:



Ao olhar no lado esquerdo da tela, agora vemos as tabelas que foram inseridas:



Selecione a tabela que deseja editar, neste exemplo editaremos a tabela de produtos:



Através do Query, é possível realizar os comandos desejados para adicionar, remover ou visualizar os dados contidos na tabela:

The screenshot shows a database query interface. At the top, there's a 'Query' tab with a query history list containing one query: 'SELECT * FROM customers;'. Below this is the 'Data Output' tab, which displays the results of the query in a table. The table has six columns: customer_id, company_name, contact_name, contact_title, address, and city. The results show six rows of customer data. At the bottom of the interface, a status bar indicates 'Total rows: 91 of 91', 'Query complete 00:00:00.384', and 'Ln 1, Col 25'. A green message box at the bottom right states 'Successfully run. Total query runtime: 384 msec. 91 rows affected.'

customer_id	company_name	contact_name	contact_title	address	city
1	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57
2	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222
3	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312
4	AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.
5	BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Berguvsvägen 8
6	BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Carsteadtstr. 57

4. BANCOS DE DADOS NÃO RELACIONAIS

Os bancos de dados não relacionais, também conhecidos como NoSQL, foram inventados pelo engenheiro de software Carlo Strozzi em 1998 para atender a crescente demanda por um sistema de banco de dados mais eficiente e escalável na era do *Big Data*, através de formas alternativas de estruturar os dados dentro do sistema.

Os dados dentro do modelo NoSQL são organizados de forma não-estruturada ou semi-estruturada, podendo funcionar através de vários algoritmos diferentes, sendo as quatro principais de acordo com Katz (2022):

- Baseado em par chave-valor: contém somente duas colunas “chave” e “valor”, usa uma tabela *hash* com uma chave única e um ponteiro;
- Baseado em colunas: usam conjuntos de colunas conhecidos como “famílias” para armazenar os dados;
- Baseado em grafos: cada elemento é armazenado como um nó, sendo cada nó e conexão entre eles identificável;
- Baseado em documentos: idêntico ao modelo chave-valor, mas os pares são armazenados na forma de um documento.

Os exemplos mais populares desse modelo atualmente no mercado são MongoDB, Apache, Redis, Amazon DynamoDB e Couchbase.

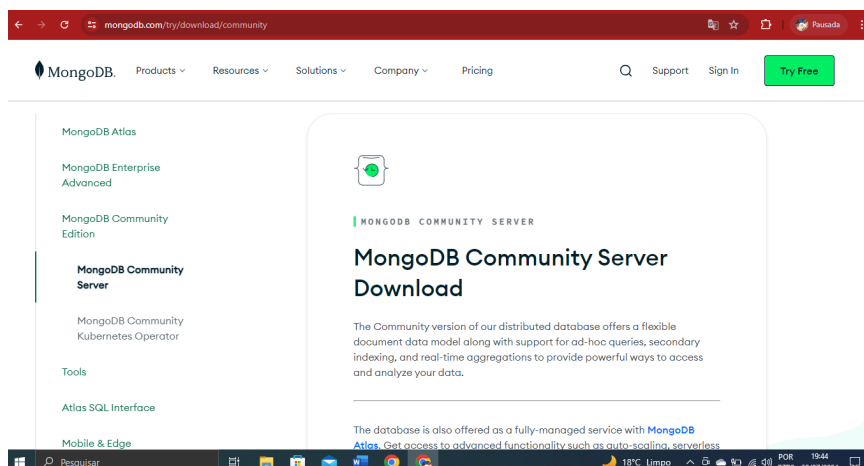
4.1 MONGODB

Em 2007, os desenvolvedores Dwight Merriman, Eliot Horowitz e Kevin Ryan enfrentaram as limitações dos bancos de dados relacionais enquanto tentavam expandir a infraestrutura da DoubleClick, uma empresa de publicidade digital. O objetivo era suportar a exibição de mais de 400.000 anúncios por segundo, o que exigia uma capacidade de processamento extremamente alta e uma arquitetura de banco de dados capaz de escalar com eficiência. Eles então decidiram criar sua própria solução que permitisse maior produtividade e escalabilidade, o resultado sendo a criação da companhia hoje conhecida como MongoDB.

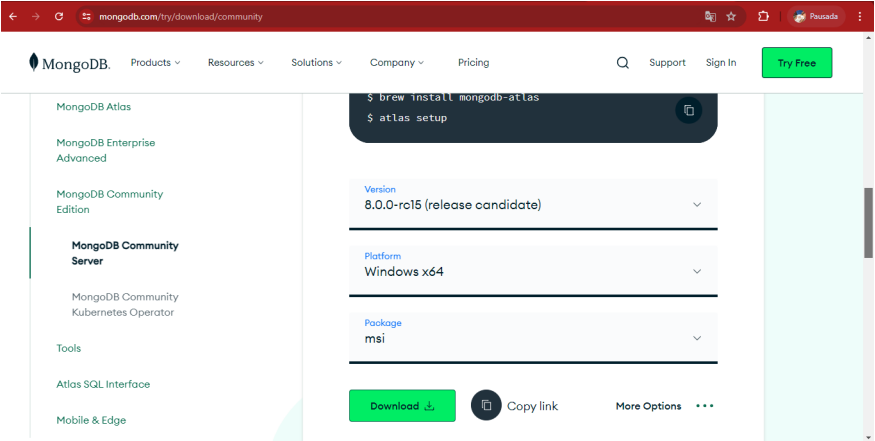
O MongoDB rapidamente se popularizou entre desenvolvedores devido à sua habilidade de lidar com grandes volumes de dados não estruturados, característica ideal para aplicações modernas que exigem alta velocidade e flexibilidade, como redes sociais, comércio eletrônico e serviços em tempo real. se tornando até hoje uma das principais soluções de NoSQL do mercado, principalmente através do MongoDB Atlas, um serviço de banco de dados gerenciado na nuvem.

4.1.1 INSTALAÇÃO DO MONGODB

1º passo: Acessar o site oficial do mongo



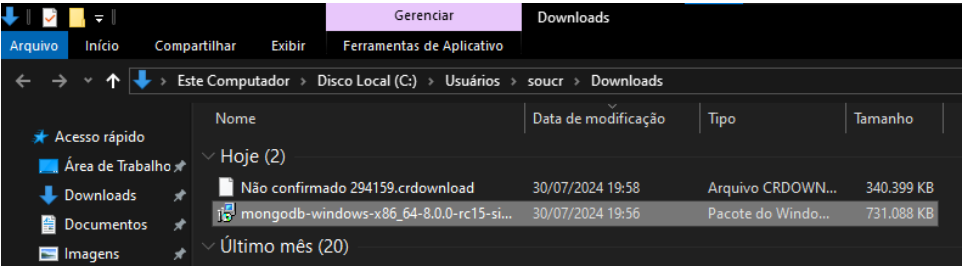
2º passo: Selecionar o sistema operacional e versão adequada



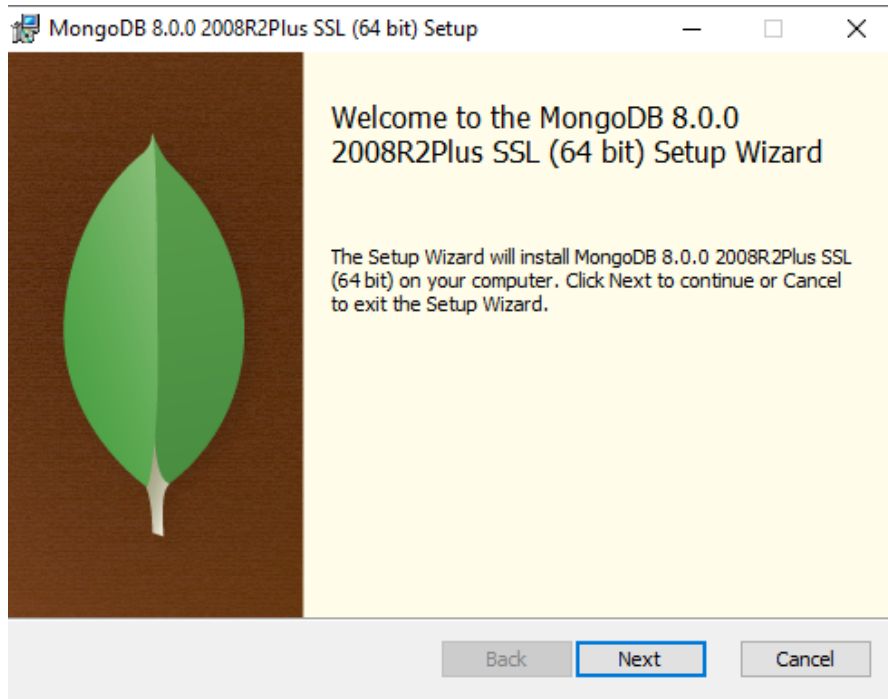
3º passo: Realizar o download do instalador



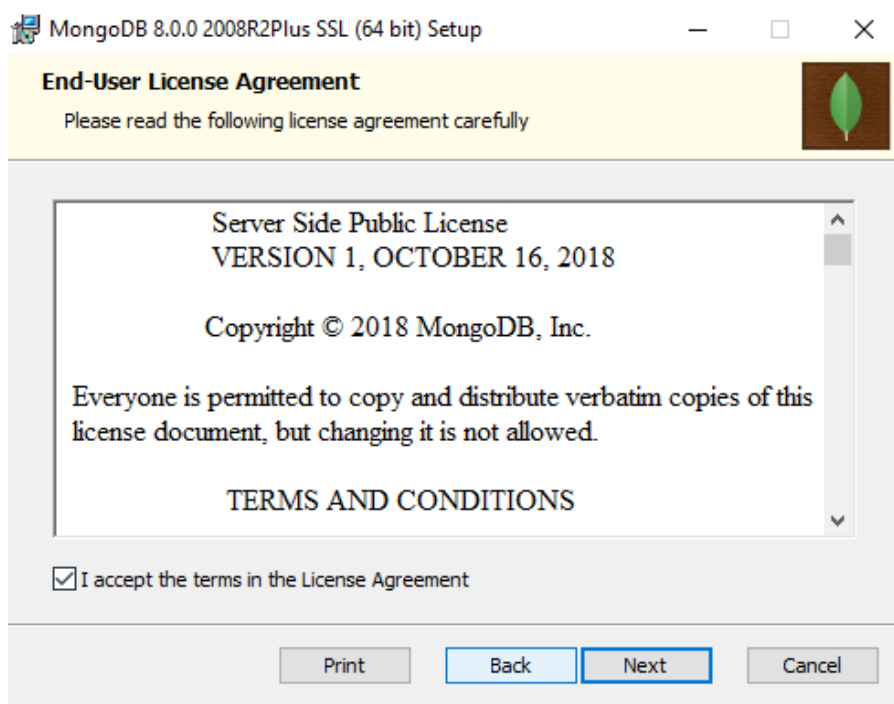
4º passo: Executar o instalador do MongoDB



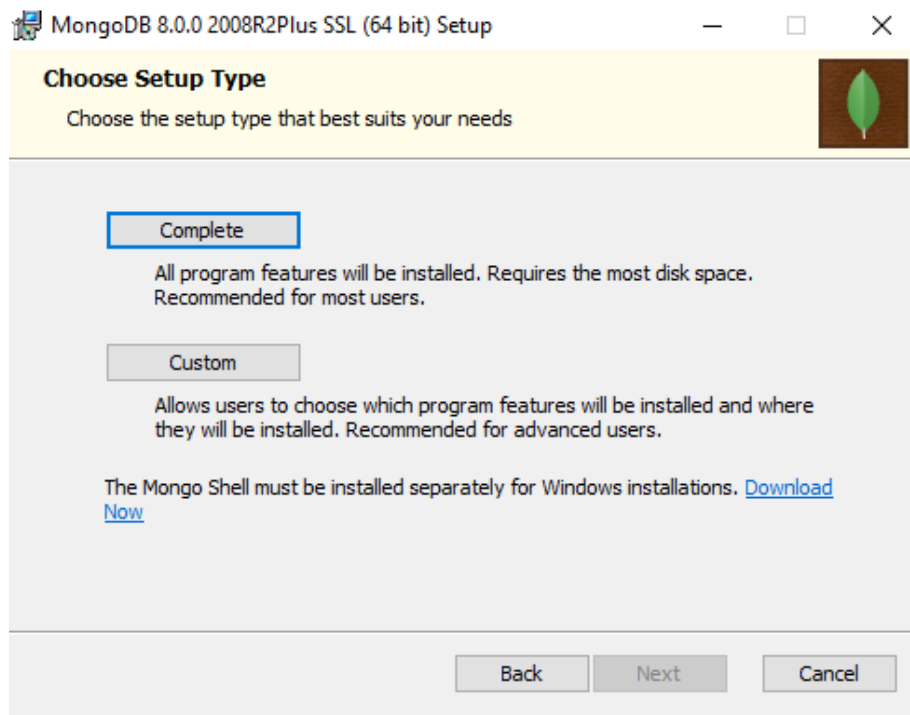
5º passo: Pressionar o botão “Next”



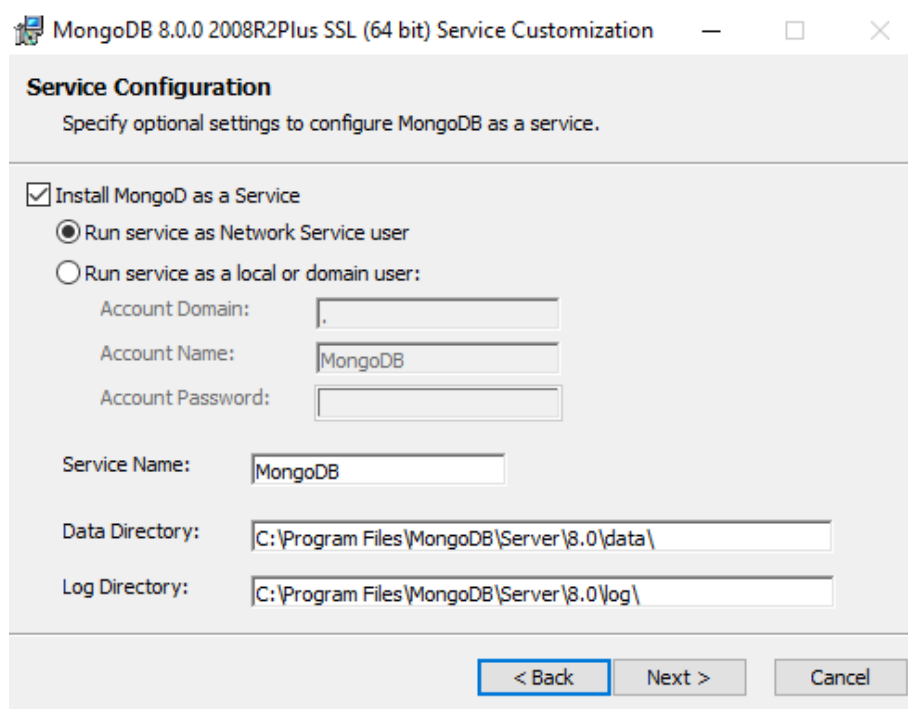
6º passo: Pressionar o botão “Next” novamente



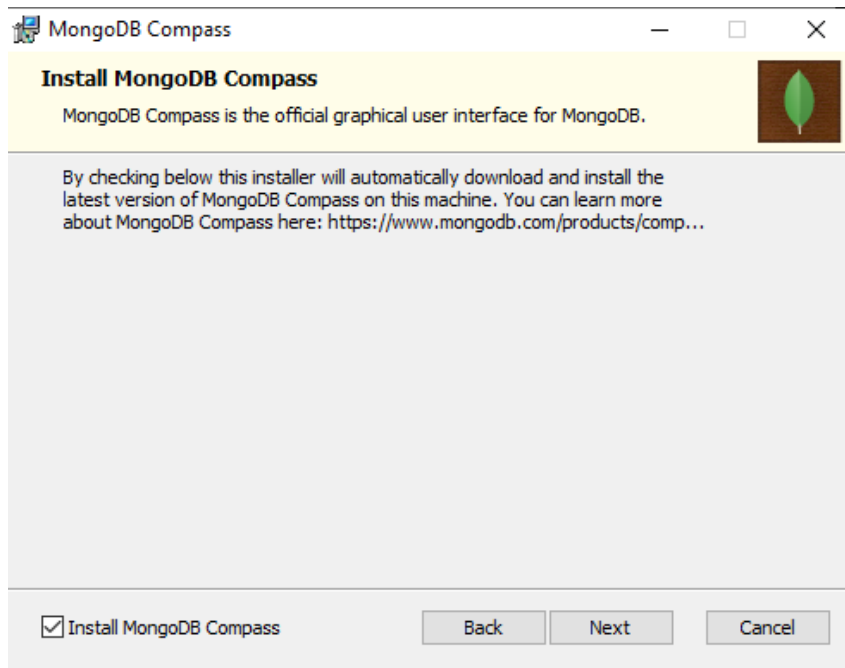
7º passo: Escolher o tipo de Setup



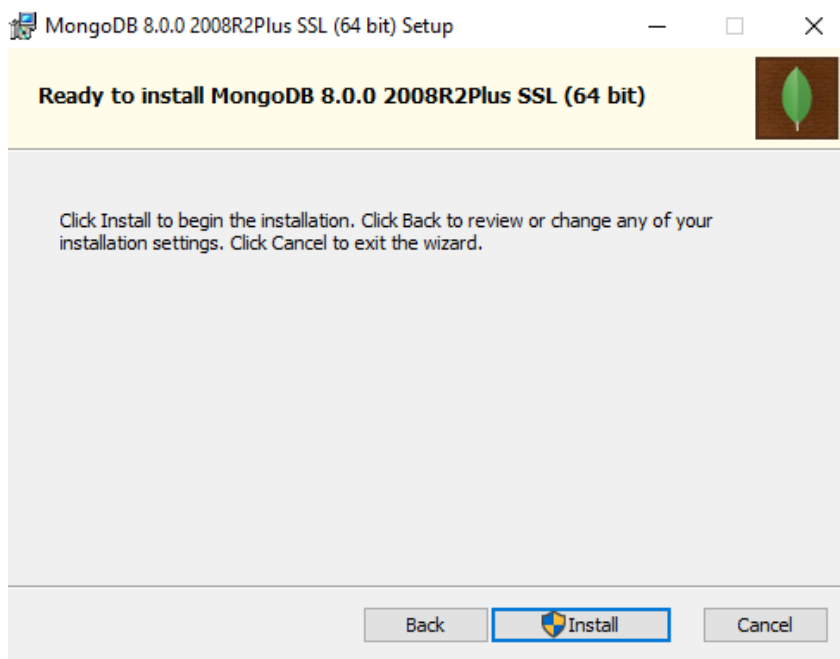
8º passo: Configurar o tipo de serviço que será utilizado no seu MongoDB.



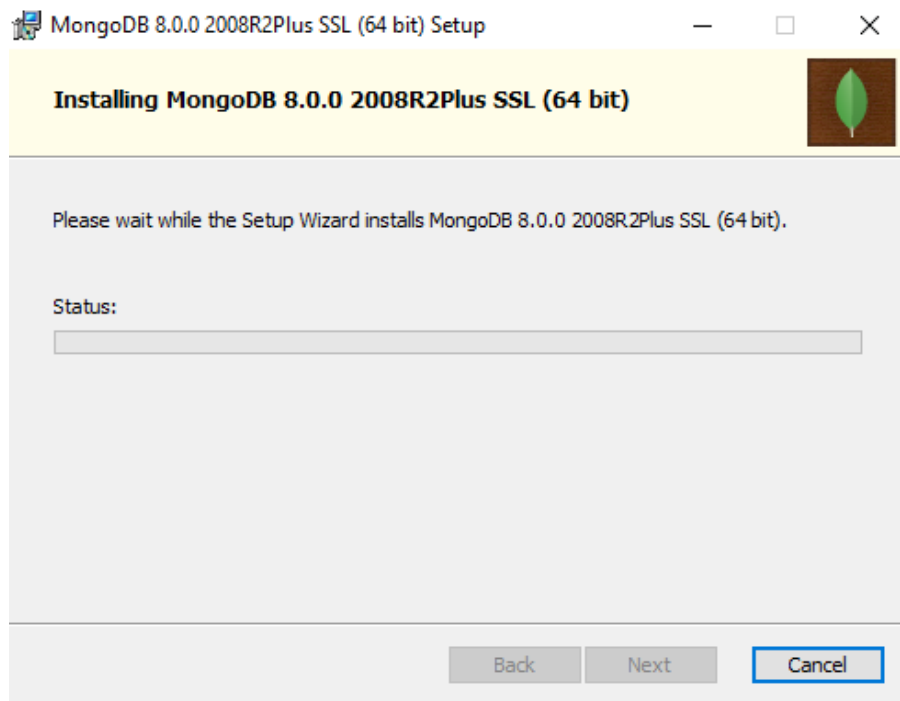
9º passo: Selecionar a opção para instalar o MongoDB Compass e pressione o botão “Next”



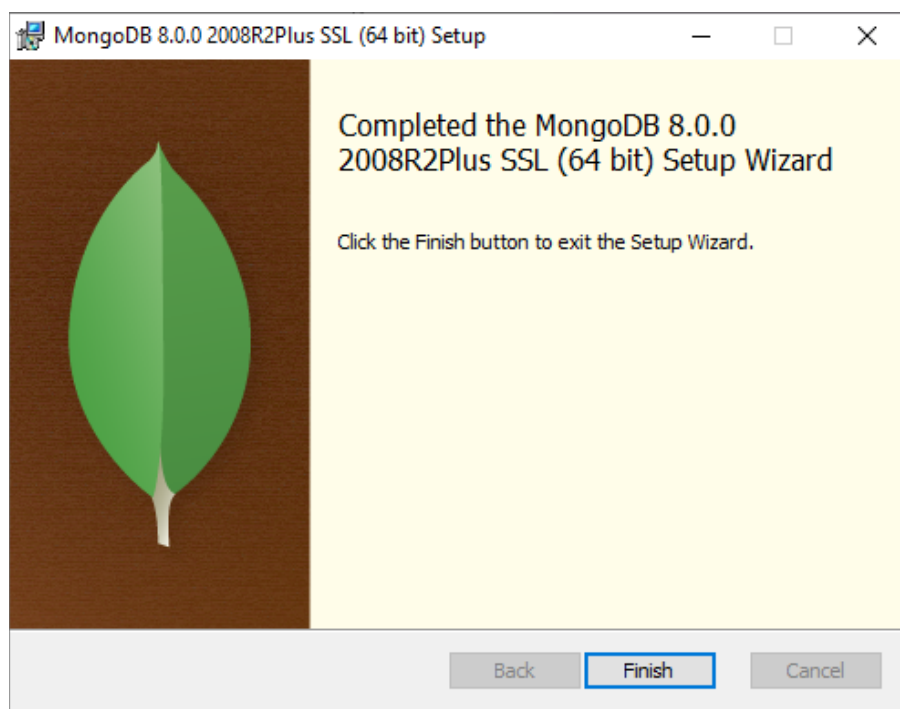
10º passo: pressionar o botão “Install”



11º passo: Aguardar a instalação do software



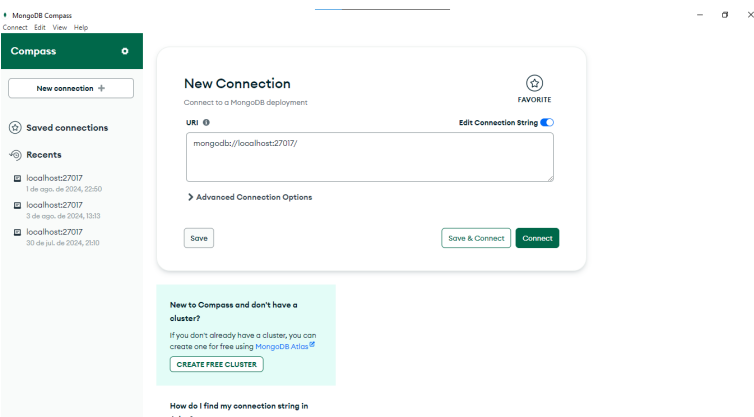
12º passo: pressionar o botão "Finish"



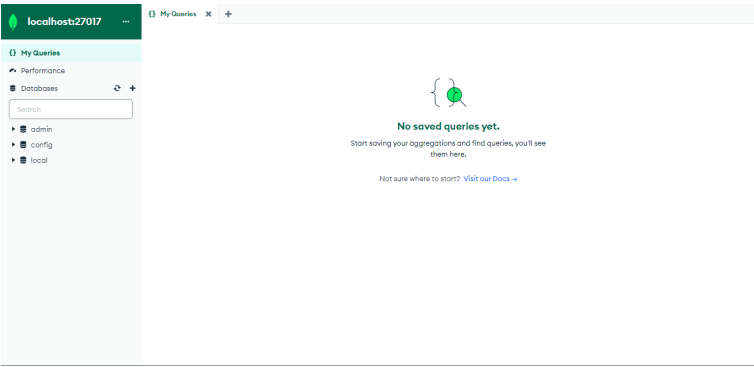
3.1.2 USANDO O MONGODB

A seguir está um tutorial de como usando o MongoDB

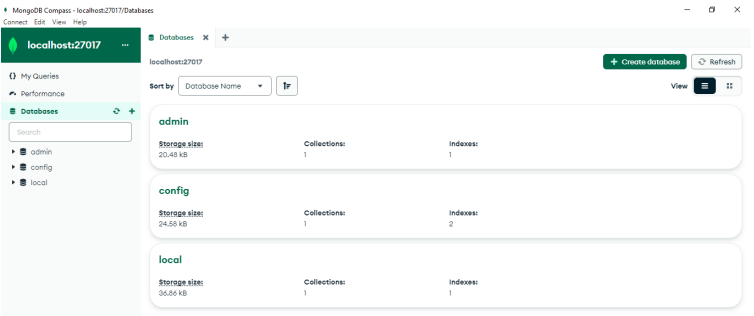
Selecione o “Connect” para iniciar sua conexão:



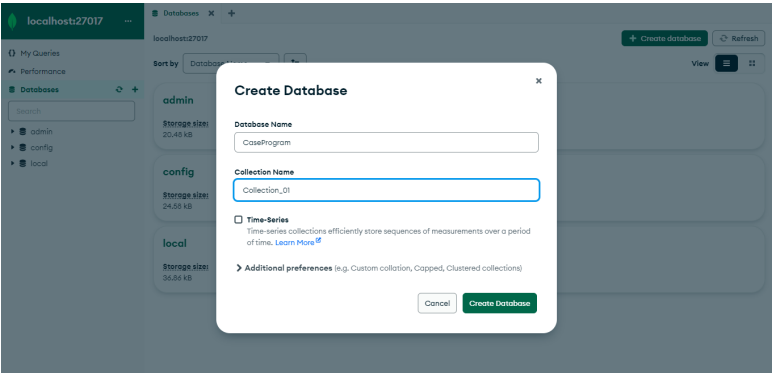
Clique no símbolo de “+” ao lado de “Databases” no lado esquerdo da tela:



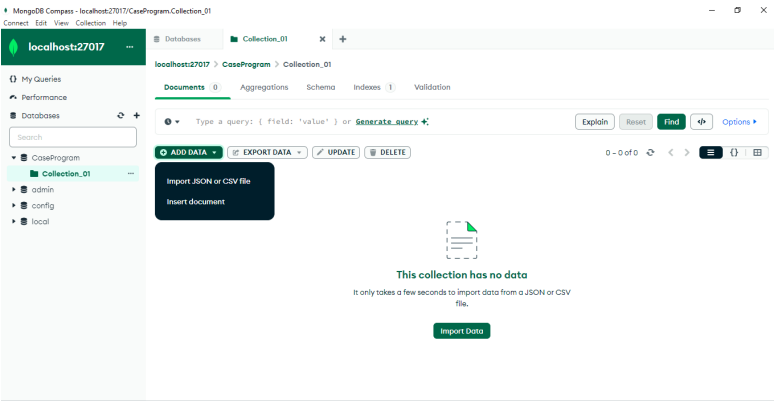
Pressione “+ Create database”:



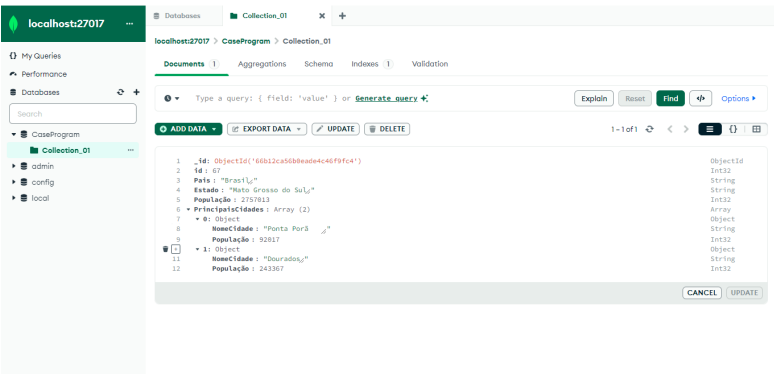
Nomeie seu banco de dados e sua coleção:



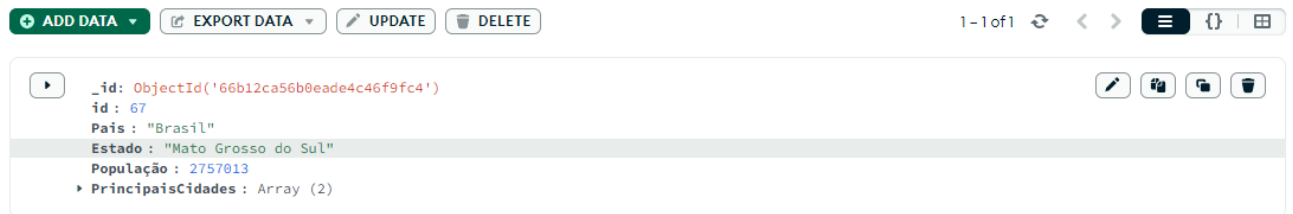
Dentro da opção “+ ADD DATA”, selecione “Import document”:



Escreva o código dos documentos que serão usados para a criação do banco de dados:



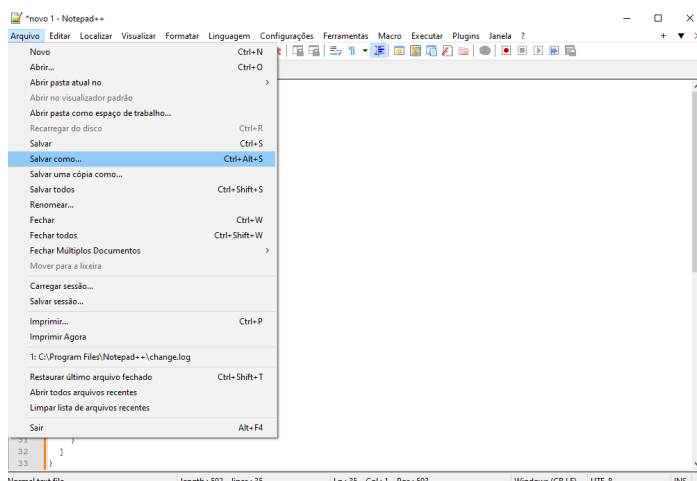
Agora podemos visualizar o documento criado, além de modificá-lo com as opções no lado direito:



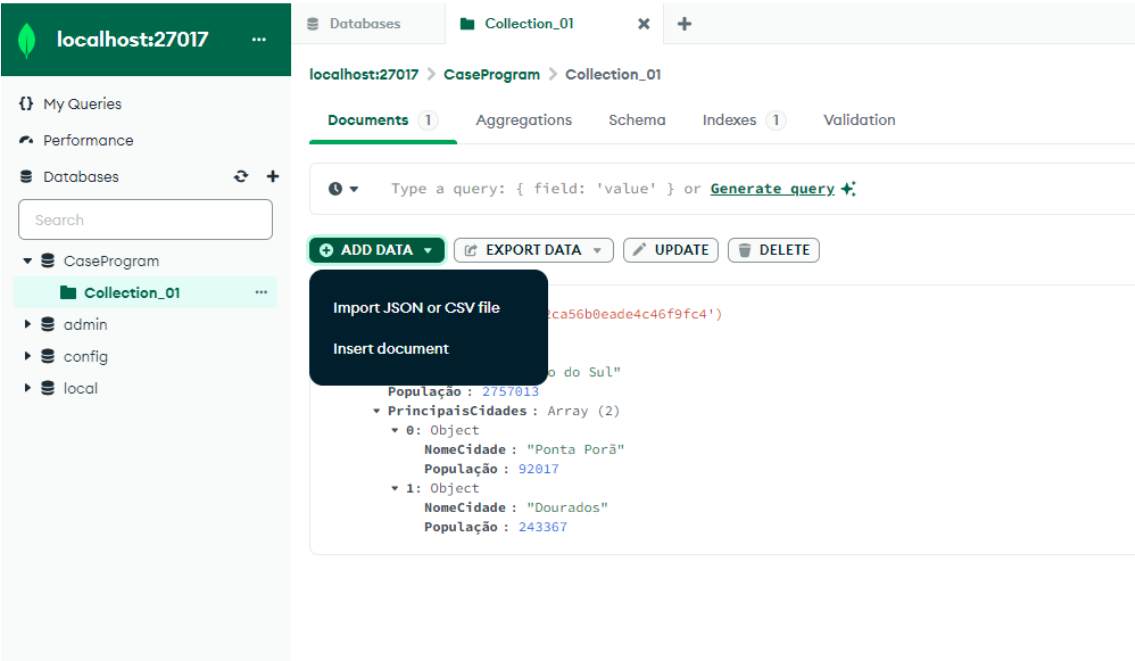
Outra forma de adicionar documentos ao banco de dados é importando um arquivo JSON, neste exemplo foi utilizado o Notepad++ para exportar este arquivo:



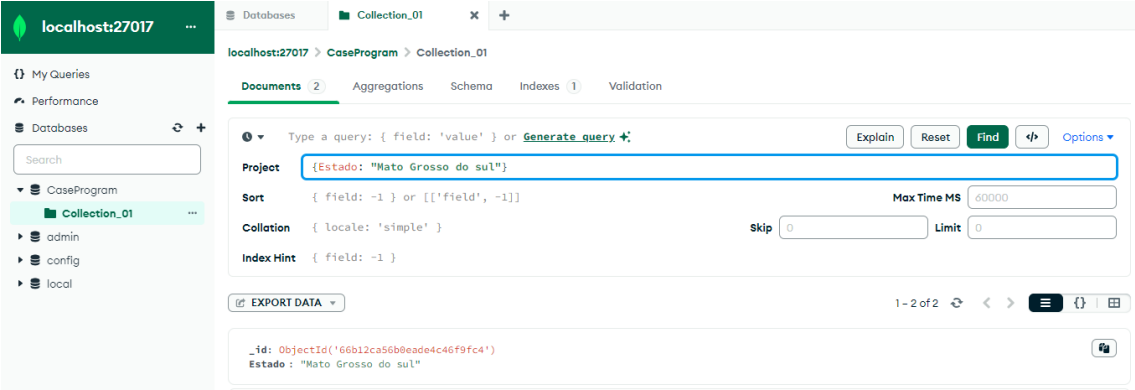
Dentro do Notepad++ escreva o código que será usado para criar o banco de dados e salve como JSON:



Na interface do MongoDB, importe o arquivo JSON através da opção “+ ADD DATA”:



Para visualizar os dados presentes dentro dos documentos, clique em “Options” no canto direito superior e busque pelos dados desejados:



5. COMPARATIVO ENTRE OS TIPOS DE BANCOS DE DADOS

Apesar de o NoSQL ter sido criado com a ideia de ser uma evolução do SQL, ambos apresentam seu próprio conjunto de vantagens e desvantagens, além de características únicas que os diferenciam de outros modelos, sendo as principais de acordo com Souza (2023):

SQL - (Relacional)	NoSQL - (Não Relacional)
Organiza os dados em linhas e colunas, que coletivamente formam uma tabela. E o tipo de banco de dados mais encontrados nos sistemas tradicionais.	Utiliza os tipos de armazenamentos em chave/valor, documentos como o JSON ou até mesmo usando gráficos que consistem em bordas e vértices.
O escalonamento em um modelo relacional, no entanto, é muito complexo. Possui uma natureza estruturada, portanto, a inserção dinâmica e transparente de novos nós a tabela não é realizada naturalmente.	Não possui um esquema pré-definido fazendo com que este tipo de modelo seja flexível o que favorece a inserção de outros elementos.
As suas regras de consistência são bastante rigorosas no que diz respeito à consistência das informações.	Se não houver nenhuma atualização nos dados, todos os acessos aos itens devolverão o último valor que foi atualizado.
Por não conseguir trabalhar de forma eficiente com a distribuição de dados, o modelo relacional acaba não suportando uma demanda muito grande de informações.	Alto nível de distribuição de dados, permitindo assim que seja possível fazer com que um enorme fluxo de solicitações aos dados seja atendido com a vantagem de o sistema ficar indisponível o menor tempo possível.
Depende do subsistema do disco. A otimização de consultas, índices e estrutura de tabela é necessária para alcançar o máximo desempenho.	Função do tamanho do cluster do hardware subjacente, da latência de rede e do aplicativo que faz a chamada.
São projetados para aplicativos transacionais e fortemente consistentes de processamento de transações online (OLTP) e são eficientes para processamento analítico online (OLAP).	São projetados para OLTP para vários padrões de acesso aos dados que incluem aplicativos de baixa latência. Os bancos de dados de pesquisa do NoSQL são projetados para análise de dados semiestruturados.

Podemos ver que o modelo NoSQL se diferencia pela sua estrutura flexível, que facilita a inserção de grandes volumes de dados, e permite escalabilidade horizontal, enquanto o modelo SQL mantém sua vantagem na sua consistência e acessibilidade, sendo mais simples de implementar e manusear.

6. CASOS DE USO

Alguns exemplos de empresas que usam bancos de dados são a Amazon e a NASA, que precisam que seus bancos de dados sejam extremamente confiáveis e estejam sempre disponíveis. Sistemas bancários também costumam usar bancos SQL, não querendo arriscar a perda de qualquer tipo de dado de seus clientes, pois um pequeno erro pode custar centenas de milhões para esses bancos e, mais importante, a perda da confiança que seus clientes têm neles.

As empresas que mais se beneficiam da implementação de bancos de dados não relacionais são aquelas onde a confiabilidade e consistência são os fatores mais importantes, permitindo que a estabilidade dos sistemas relacionais brilhe em comparação aos não relacionais. Outro atrativo dos bancos SQL são os relacionamentos complexos e bem definidos, que os tornam perfeitos para sistemas de cadastros, por exemplo.

Por outro lado, empresas que utilizam bancos de dados No SQL são empresas com muitos dados e variáveis, como por exemplo a Netflix ou o Spotify, onde as preferências de todos os usuários estão sempre se modificando e precisando ser atualizadas. Redes sociais também precisam de bancos de dados NoSQL para manejar a extrema quantidade de informação sendo criada a todo momento.

Podemos então definir que o NoSQL é a melhor opção quando a quantidade de dados a ser acessada é muito grande e precisa ser manipulada de forma ágil e flexível, como em serviços de streaming que precisam armazenar as preferências ou em análises em tempo real, principalmente quando latência baixa é uma das prioridades desses serviços.

Além disso, existem empresas que usam modelos híbridos, usando tanto bancos SQL como NoSQL ao mesmo tempo para diferentes funções, como por exemplo a Uber, que usa bancos de dados SQL para armazenar as informações de seus passageiros e motoristas mas usa bancos de dados NoSQL para armazenar os trajetos de suas corridas e realizar a geolocalização em tempo real que garante a segurança de seus usuários.

7. RESULTADOS E DISCUSSÕES

É impossível imaginar um mundo sem bancos de dados, o fluxo de informação do mundo moderno é grande demais para conseguirmos gerenciar sem a ajuda deles e a velocidade em que todos os sistemas do planeta se movem é muito rápida para acompanharmos de forma analógica. Os bancos de dados estão entre as revoluções mais importantes da história da computação e o estudo deles se torna imperativo quando imaginamos que o fluxo de informações no mundo só vai aumentar no futuro.

Ao estudar a linha do tempo dos bancos de dados e ver o ritmo de evolução das tecnologias relacionadas ao mesmo, é notável o quanto a necessidade gera a inovação, e como só vemos o surgimento de um novo modelo quando os modelos anteriores começam a se mostrar insuficientes. Com base nessa ideia, podemos assumir que talvez o próximo grande modelo de banco de dados esteja próximo, com os avanços que pesquisadores têm feito no ramo da computação é só uma questão de tempo até que os modelos que temos hoje em dia sejam ineficientes demais para suportar o fluxo exigido.

A comparação entre bancos de dados relacionais e não relacionais parece ser um pouco injusta, visto que um foi feito com o propósito de ser a evolução do outro, e apesar de esse ser de fato o caso na maioria das situações, os bancos de dados SQL ainda podem se mostrar úteis devido a sua simplicidade e facilidade de implementação, podendo ser usado até mesmo em conjunto com bancos de dados NoSQL em certos casos, como um restaurante que use um banco de dados SQL para fazer o inventário de seus produtos e um banco de dados NoSQL para armazenar avaliações de clientes.

Sendo assim, é importante para aqueles que desejam trabalhar nessa área, compreender e dominar o uso de ambos os tipos de bancos de dados, estudando quando aplicar cada um deles e os melhores DBMS para gerenciá-los. Tanto o PostgreSQL quanto o MongoDB são excelentes escolhas para quem quer aprender um pouco mais sobre o assunto devido ao quão acessível eles são, podendo ser utilizados até mesmo por quem tem pouca ou nenhuma experiência no ramo sem maiores dificuldades.

REFERÊNCIAS

A Brief History of PostgreSQL. **PostgreSQL**, 2024. Disponível em: <https://www.postgresql.org/docs/current/history.html>. Acesso em: 25 out. 2024.

ARAÚJO, Raquel. Introdução ao NoSQL e MongoDB. **Hashtag Treinamentos**, 12 maio 2023. Disponível em: <https://www.hashtagtreinamentos.com/introducao-ao-nosql-e-mongodb-sql>. Acesso em: 25 out. 2024.

CATUNDA, Heitor. Instalação do PostgreSQL e criação do seu 1º banco de dados! **Hashtag Treinamentos**, 16 abr. 2022. Disponível em: <https://www.hashtagtreinamentos.com/instalacao-do-postgresql-sql>. Acesso em: 25 out. 2024.

KATZ, Eyal. The 4 Types of NoSQL Databases You need to Know. **Spectral**, 22 jun. 2022. Disponível em: <https://spectralops.io/blog/nosql-databases/>. Acesso em: 25 out. 2024.

KOSINSKI, Matthew. What is a database? **IBM**, 30 set. 2024. Disponível em: <https://www.ibm.com/think/topics/database>. Acesso em: 25 out. 2024.

LACEFIELD, Jonathan. The Evolution of NoSQL. **DataStax**, 29 ago. 2018. Disponível em: <https://www.datastax.com/blog/evolution-nosql>. Acesso em: 25 out. 2024.

NoSQL Databases: What Every Data Scientist Needs to Know. **Datacamp**, 24 jun. 2022. Disponível em: <https://www.datacamp.com/blog/nosql-databases-what-every-data-scientist-needs-to-know>. Acesso em: 25 out. 2024.

Our Story. **MongoDB**, 2024. Disponível em: <https://www.mongodb.com/company/our-story>. Acesso em: 25 out. 2024.

SOUZA, Larissa Vital. **Estudo comparativo entre os modelos de banco de dados NOSQL e SQL**. 2023. Dissertação (Graduação em Sistemas para Internet) - Fatec, Faculdade de Tecnologia Prof. José Camargo, Jales, SP, 2023.

THOMET, Pascal. northwind_psql. 2023. Disponível em: https://github.com/pthom/northwind_psql/blob/master/northwind.sql. Acesso em: 25 out. 2024.

What Is a Database? **Oracle**, 24 nov. 2020. Disponível em: <https://www.oracle.com/database/what-is-database/>. Acesso em: 25 out. 2024.

What is NoSQL? **ScyllaDB**, 2024. Disponível em: <https://www.scylladb.com/learn/nosql/>. Acesso em: 25 out. 2024.