VOCÊ JÁ SE PERGUNTOU COMO SIMPLIFICAR A CONSULTA A DIVERSOS BANCOS DE DADOS E FORMATOS DE ARQUIVOS SEM COMPLICAÇÕES?

O TRINO É A RESPOSTA!

trino

docker

MINIO

CSV

JSON

Parquet

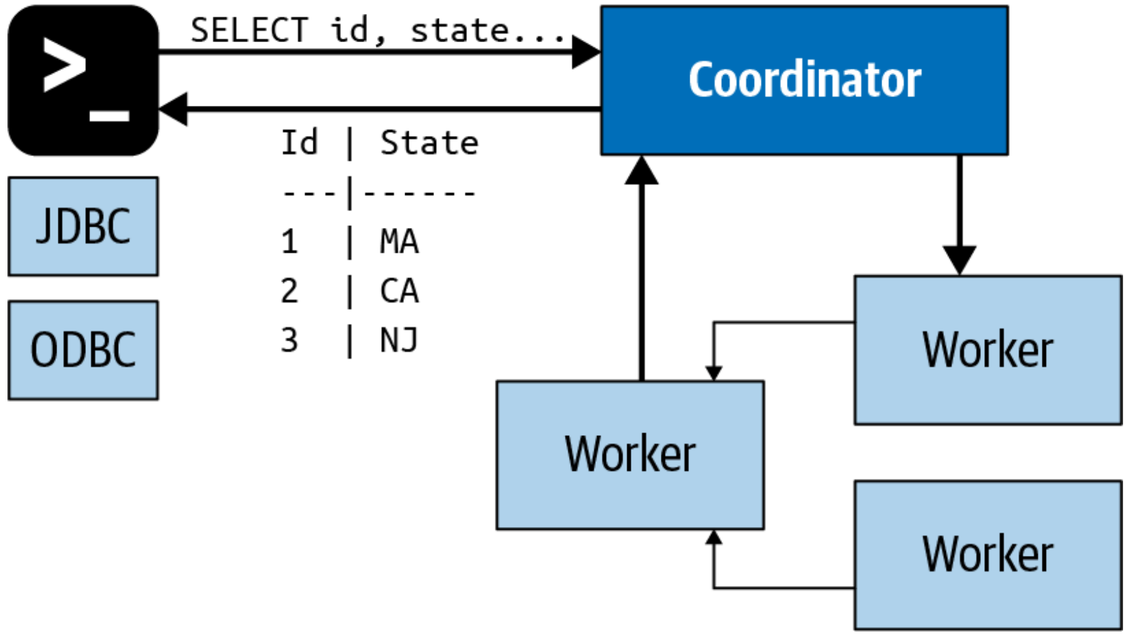
ORACLE DATABASE

PostgreSQL

Microsoft SQL Server

Trino é um mecanismo de consulta SQL distribuído de código aberto, projetado para análise interativa em várias fontes de dados heterogêneas

1. **Velocidade: O Trino é altamente paralelo e distribuído, otimizado para consultas de baixa latência e eficientes.**
2. **Escala: Organizações de grande porte usam o Trino para consultar datalakes com exabytes de dados e enormes datawarehouses.**
3. **Simplicidade: O Trino é compatível com SQL ANSI e funciona com ferramentas de BI como R, Tableau, Power BI e Superset.**
4. **Versatilidade: Suporta diversos casos de uso.**
5. **Análise no local: Você pode consultar dados diretamente em sistemas como Hadoop, S3, Cassandra, MySQL e outros, sem a necessidade de processos complexos e lentos de cópia de dados.**
6. **Execução em qualquer lugar: O Trino é otimizado para ambientes locais e em nuvem, incluindo Amazon, Azure, Google Cloud e outros.**
7. **Confiável: Grandes organizações usam o Trino para operações comerciais críticas, incluindo resultados financeiros para mercados públicos.**



Como Funciona

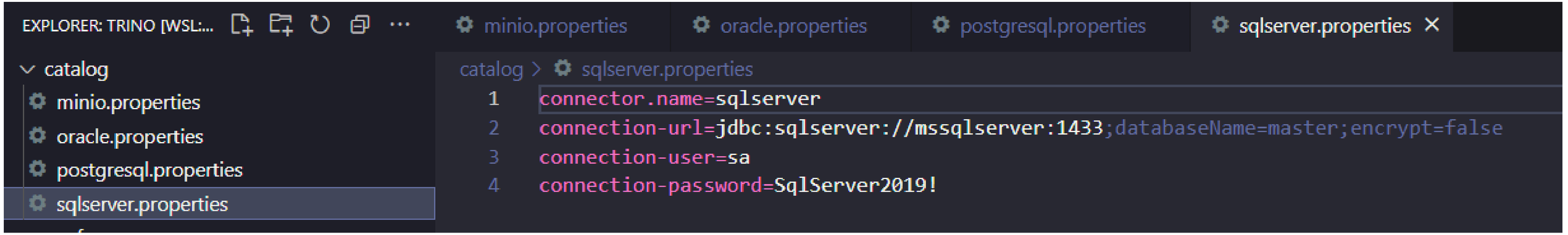
1. **Envio da Consulta: O usuário envia uma consulta SQL para o coordinator.**
2. **Planejamento: O coordinator analisa e planeja a consulta, dividindo-a em várias etapas.**
3. **Distribuição: O coordinator distribui as tarefas entre os workers.**
4. **Execução: Os workers executam as tarefas e processam os dados.**
5. **Coleta de Resultados: O coordinator coleta os resultados dos workers e os retorna ao usuário.**

Essa arquitetura distribuída permite que o Trino execute consultas complexas de forma rápida e eficiente, mesmo em grandes volumes de dados.
Se precisar de mais detalhes ou tiver outras perguntas, estou à disposição!

Arquivo docker-compose com os serviços necessarios

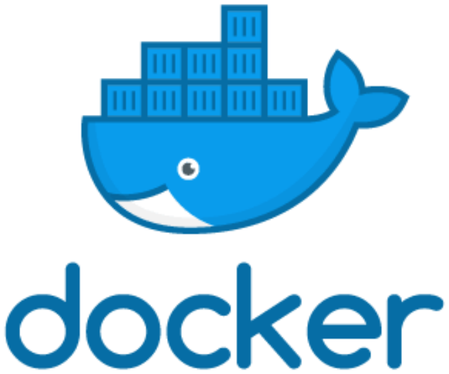
```yaml
  minio:
    ports:
      - "9001:9001"
    command: server /data --console-address
    #Não esquecer de comentar o volume na h
    volumes:
      - /mnt/d/volume/minio:/data
    environment:
      MINIO_ACCESS_KEY: minio
      MINIO_SECRET_KEY: minio123
    networks:
      - roussenq

  trino-coordinator:
    image: trinodb/trino:latest
    container_name: trino-coordinator
    environment:
      - TRINO_ENVIRONMENT=production
    volumes:
      - ./etc:/usr/lib/trino/etc
      - ./catalog:/etc/trino/catalog
    ports:
      - "8080:8080"
    depends_on:
      - roussenq

  trino-worker:
    image: trinodb/trino:latest
    container_name: trino-worker
    environment:
      - TRINO_ENVIRONMENT=production
      - TRINO_DISCOVERY_URI=http://trino-coordinator:8080
    volumes:
      - ./etc:/usr/lib/trino/etc
      - ./catalog:/etc/trino/catalog
    depends_on:
      - trino-coordinator
    networks:
      - roussenq

  hive-metastore:
    image: 'apache/hive:4.0.0'
    hostname: hive-metastore
    container_name: hive
    ports:
      - '9083:9083'
    volumes:
      - ./conf/core-site.xml:/opt/hadoop/etc/hadoop/core-site.xml:ro
      - ./conf/metastore-site.xml:/opt/apache-hive-metastore-4.0.0-bin/conf/meta
    environment:
      SERVICE_NAME: metastore
      METASTORE_DB_HOSTNAME: minio
      HIVE_AUX_JARS_PATH: /opt/hadoop/share/hadoop/tools/lib/hadoop-aws-3.3.6.ja
    depends_on:
      - minio
    networks:
      - roussenq
```

Arquivos de catologo responsaveis pelo conexao do trino com os serviços



Demostração

Software necessários

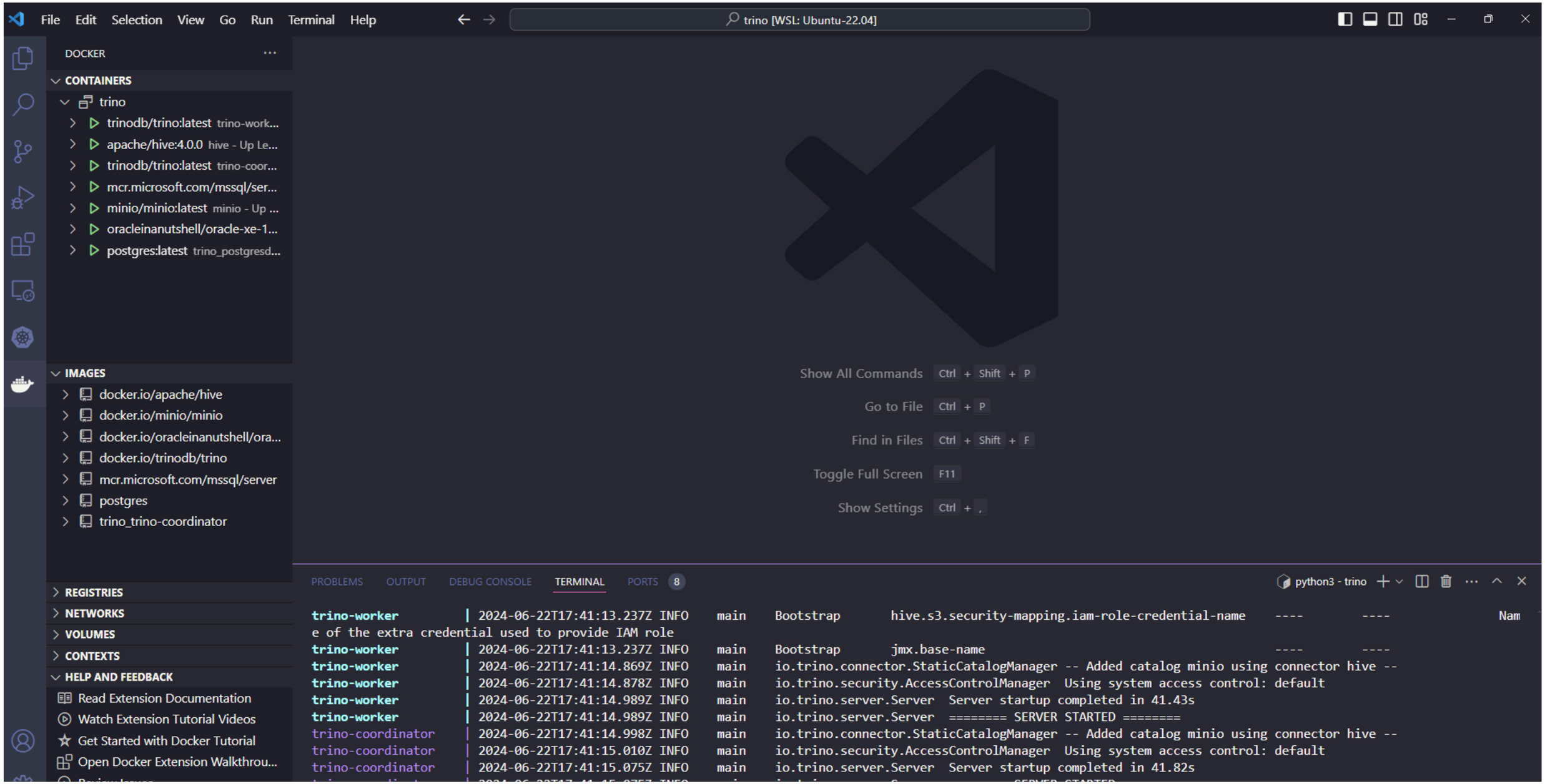


DBeaver

GitHub

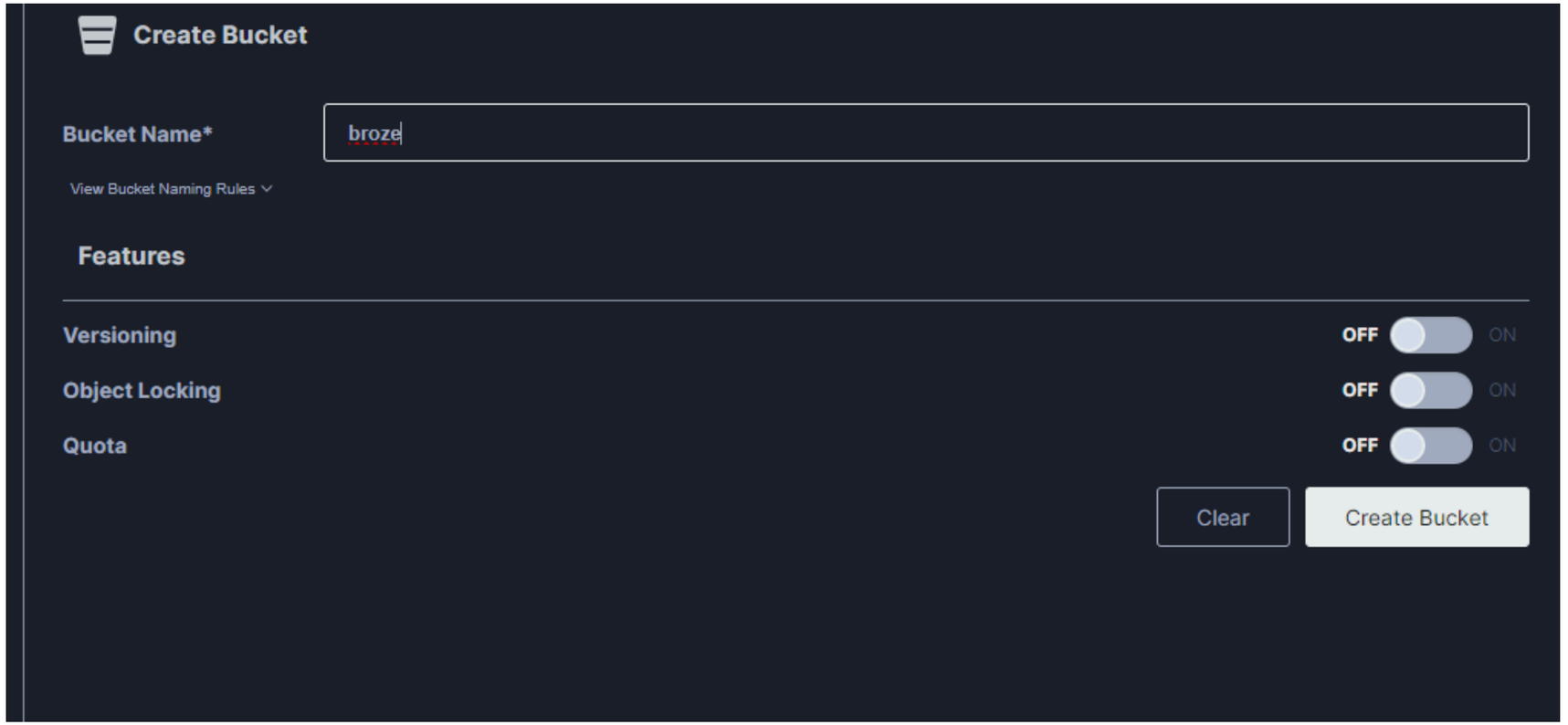
https://github.com/roussenqtb/trino

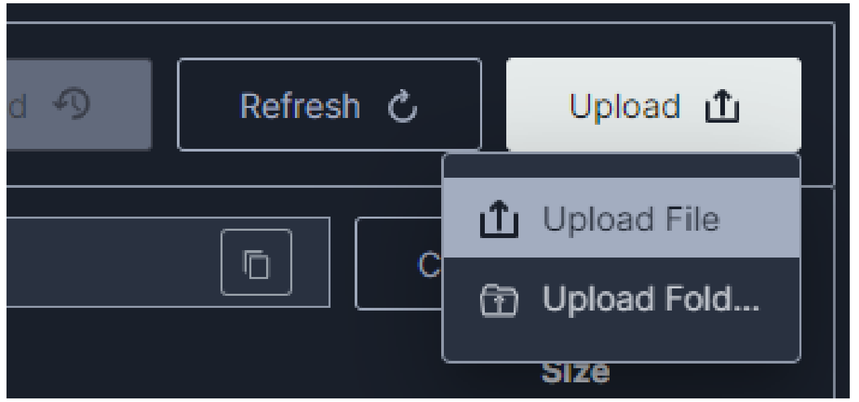
docker-compose up --build



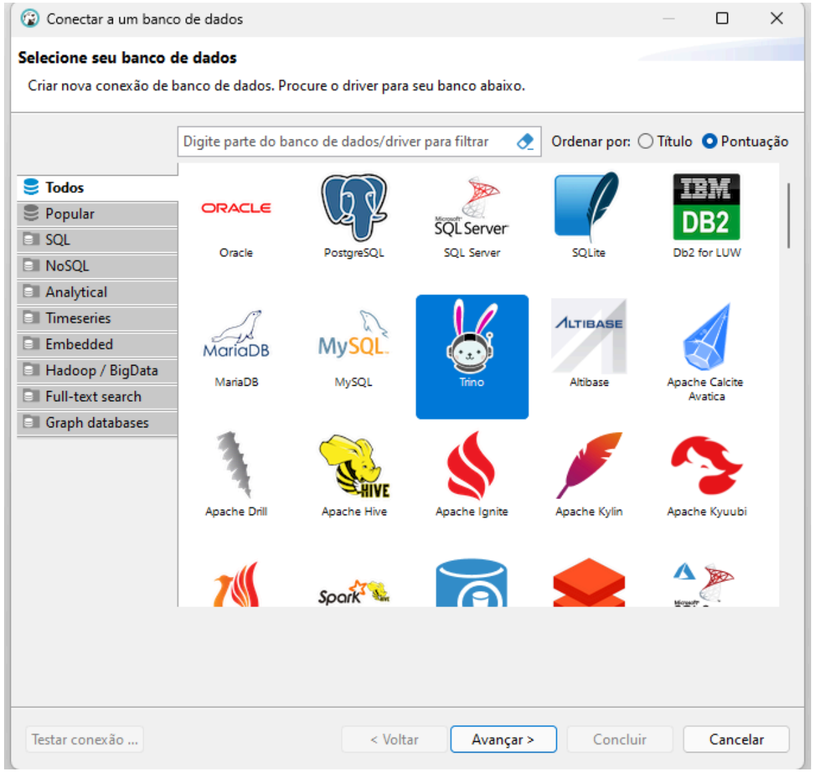
http://localhost:9001/login

usuário = minio
senha = minio123



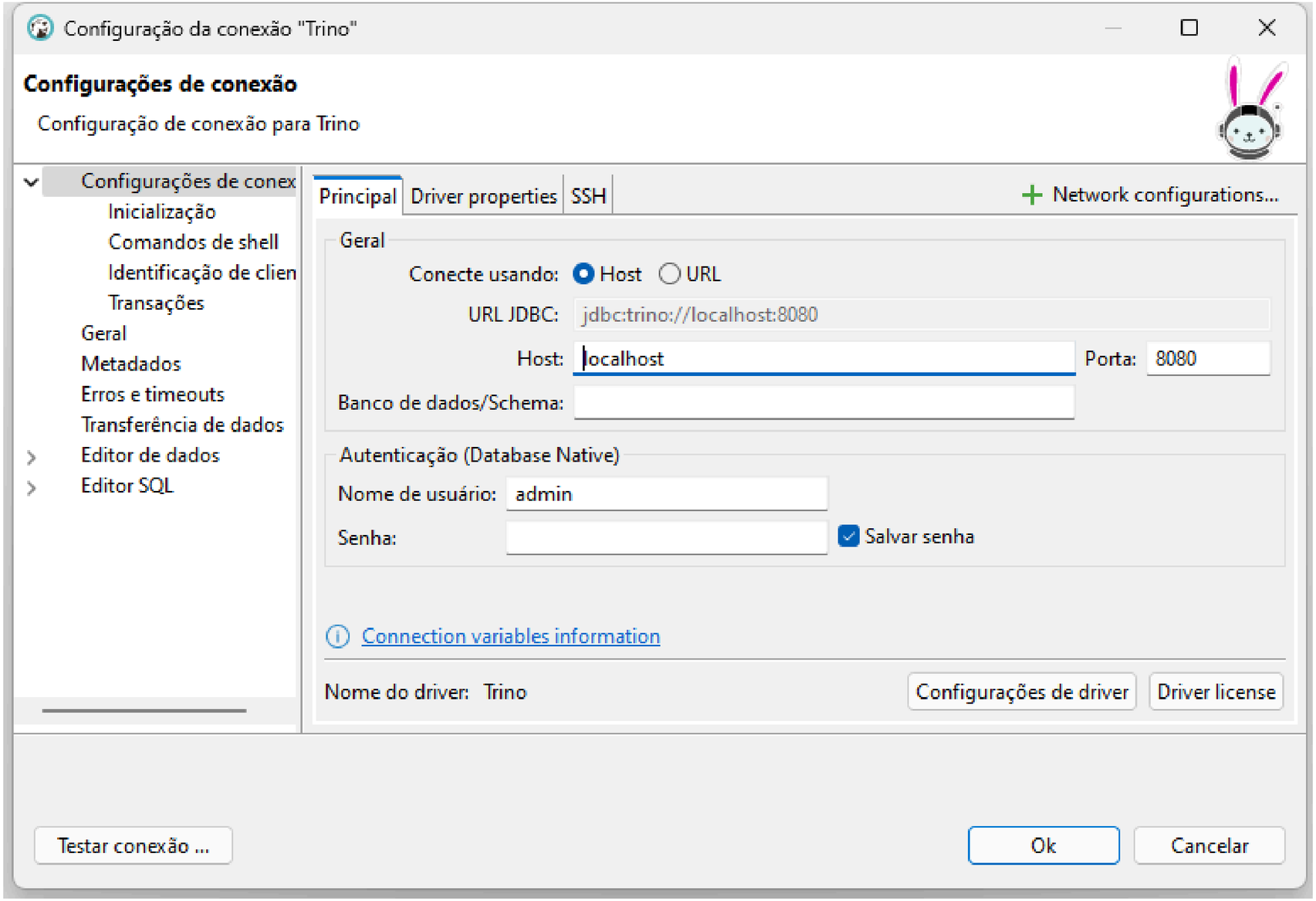


Configure o trino no dbeaver

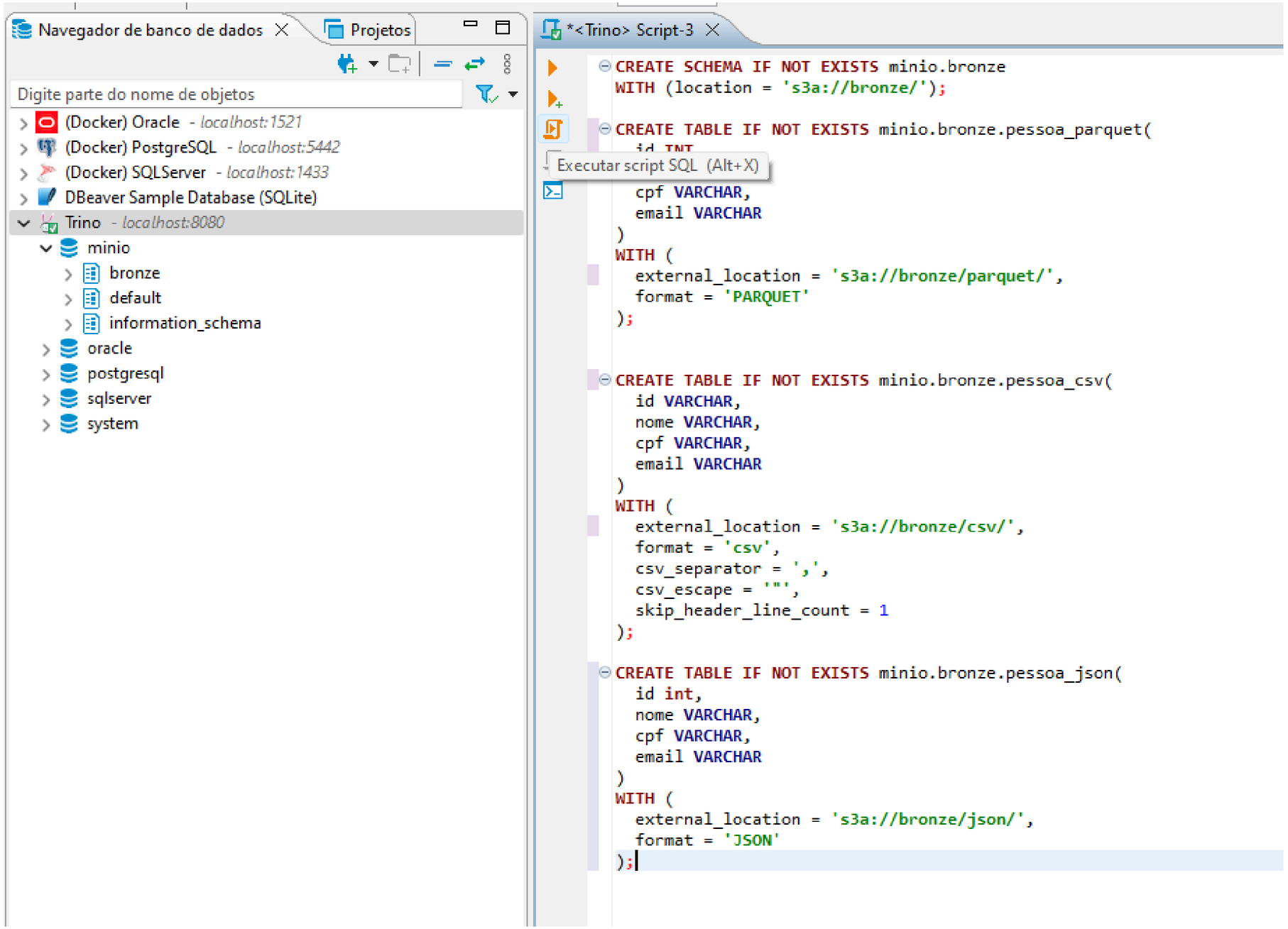


usuário = admin

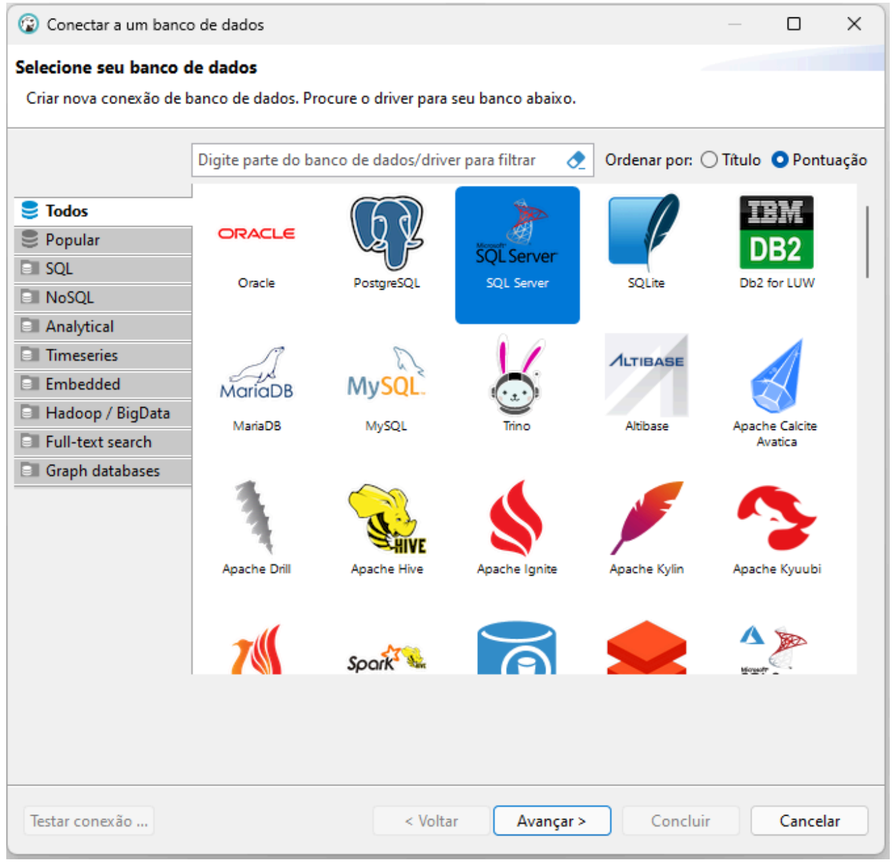
senha =



Execute o script 01 - minio.sql

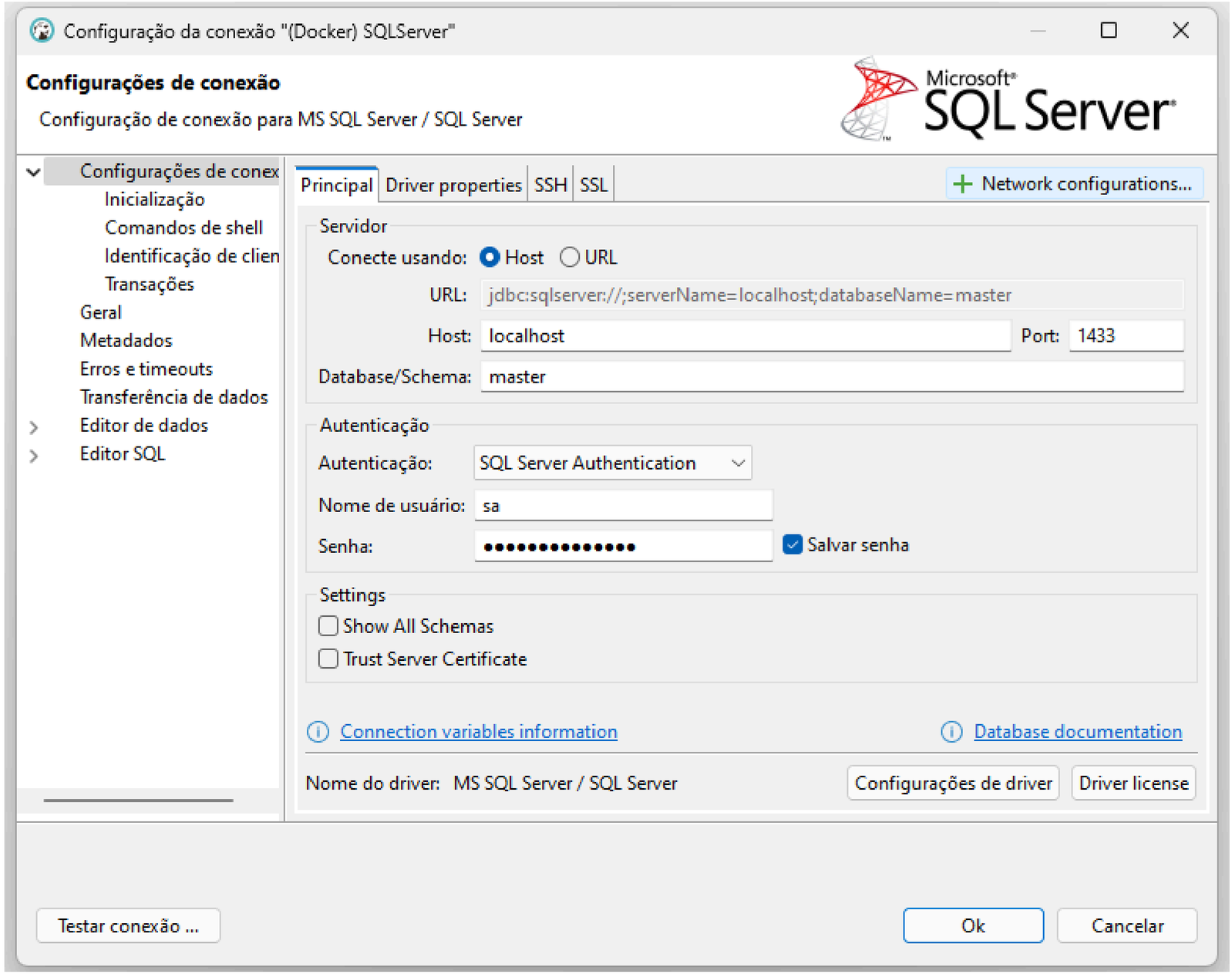


Configure a conexão com banco SQL Server

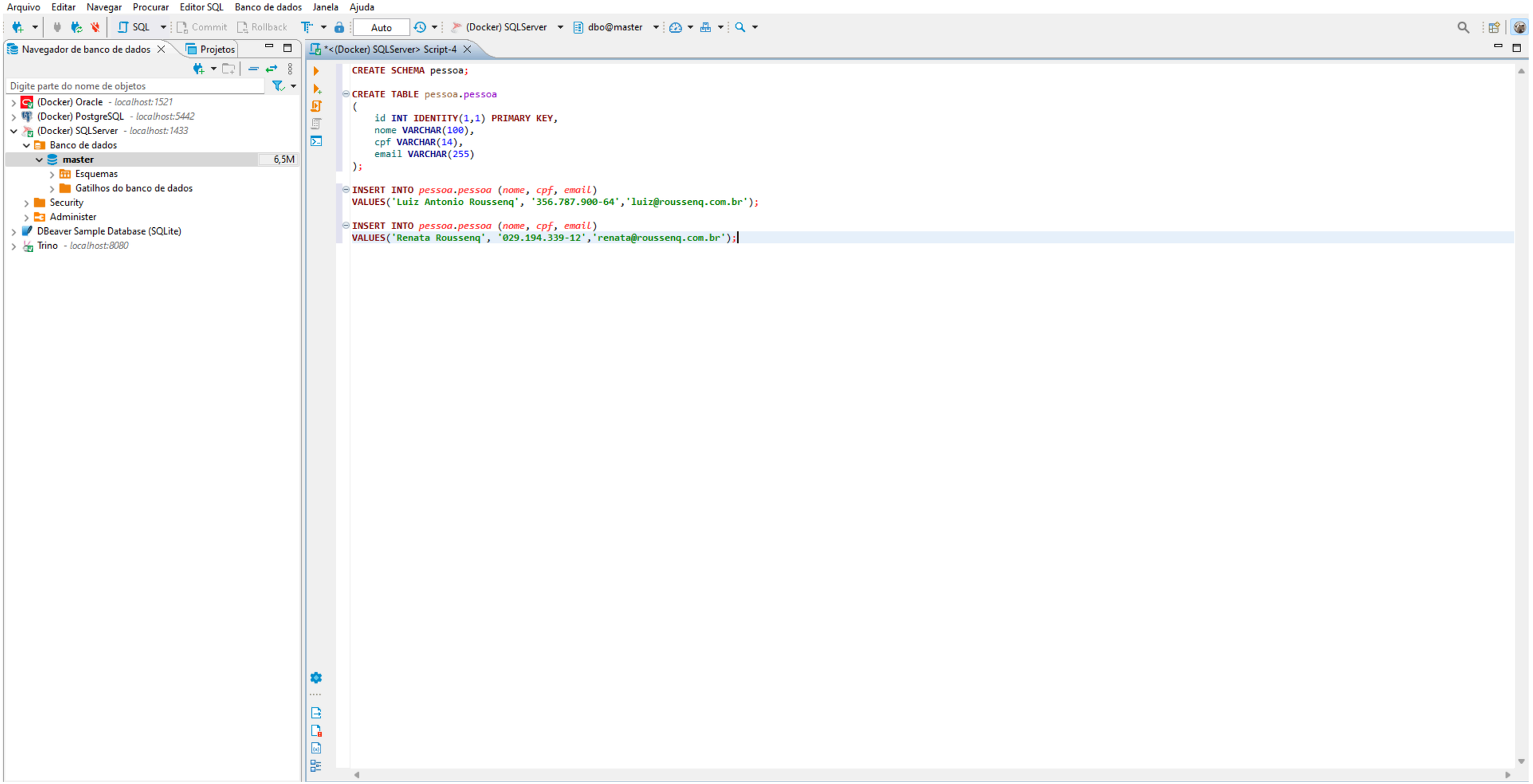


usuário = sa

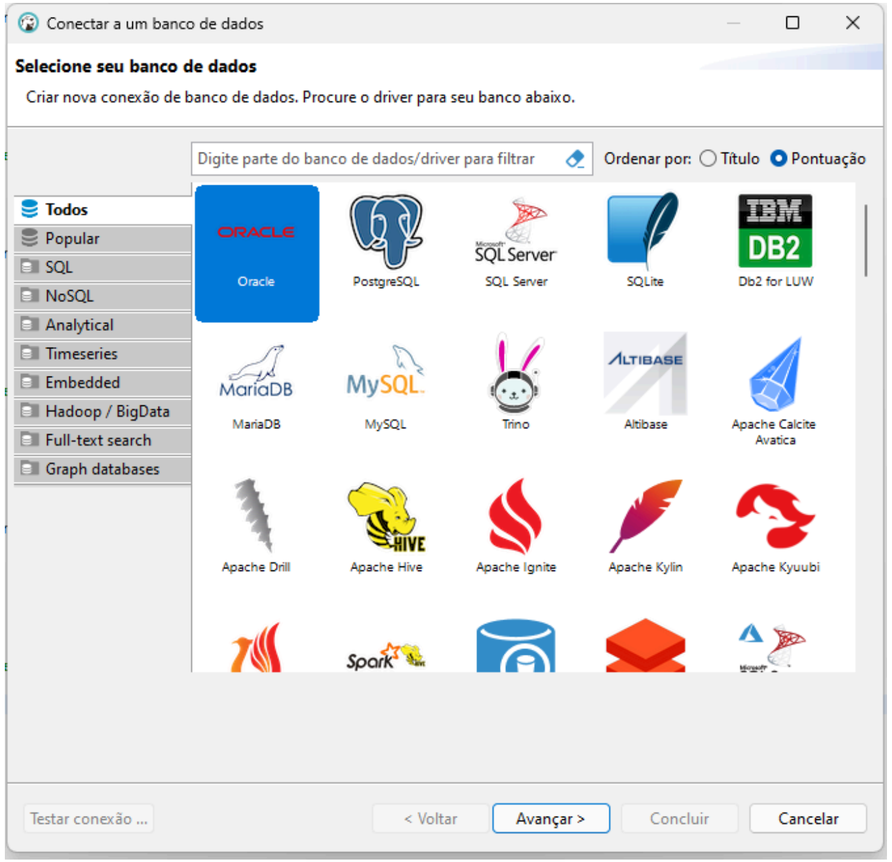
senha = SqlServer2019!



Execute o script 02 - sqlserver.sql

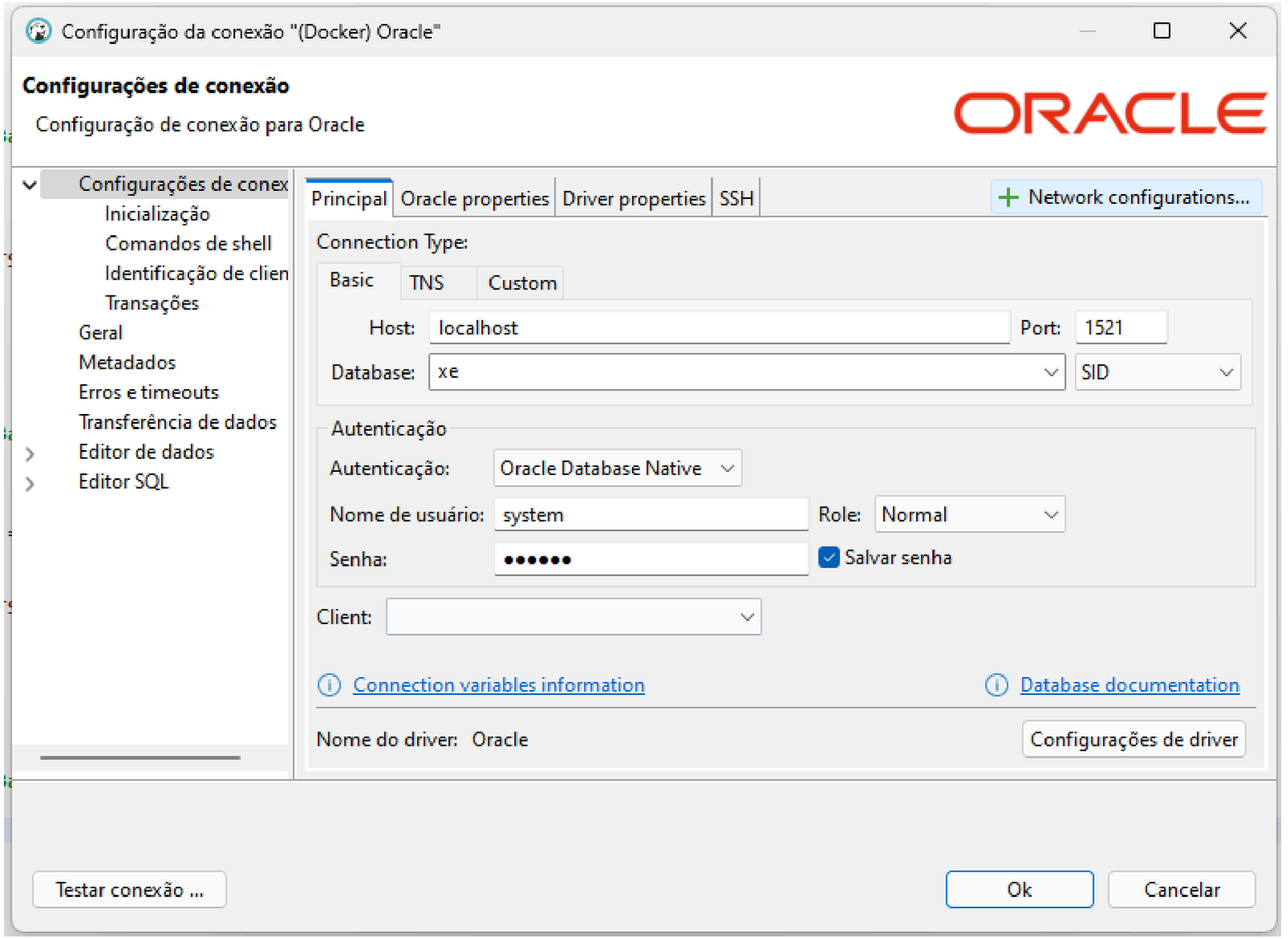


Configure a conexão com banco oracle

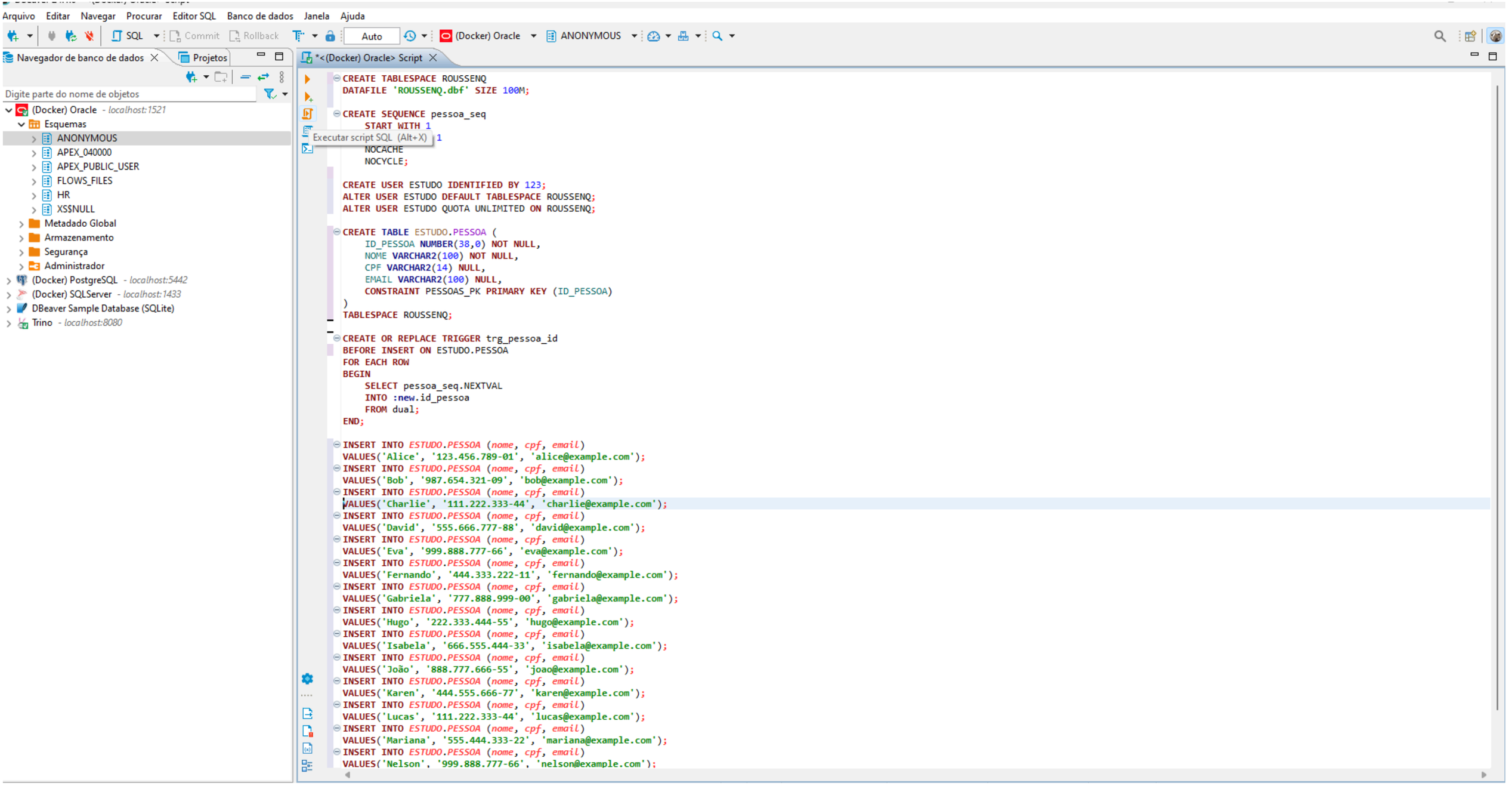


usuário = system

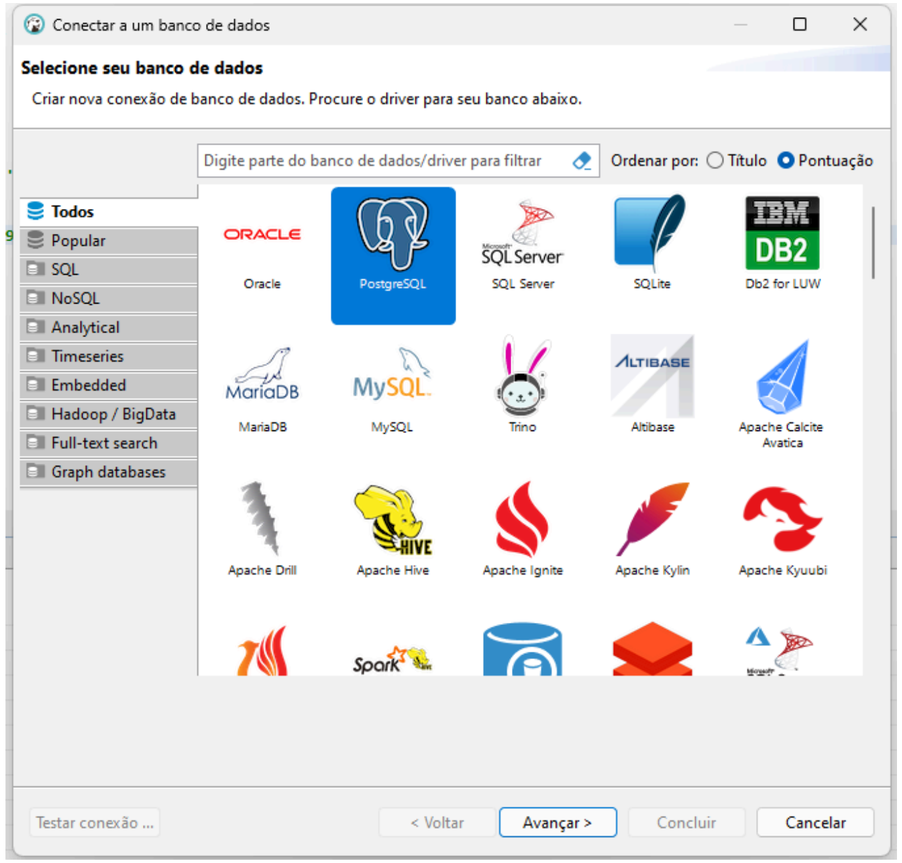
senha = oracle



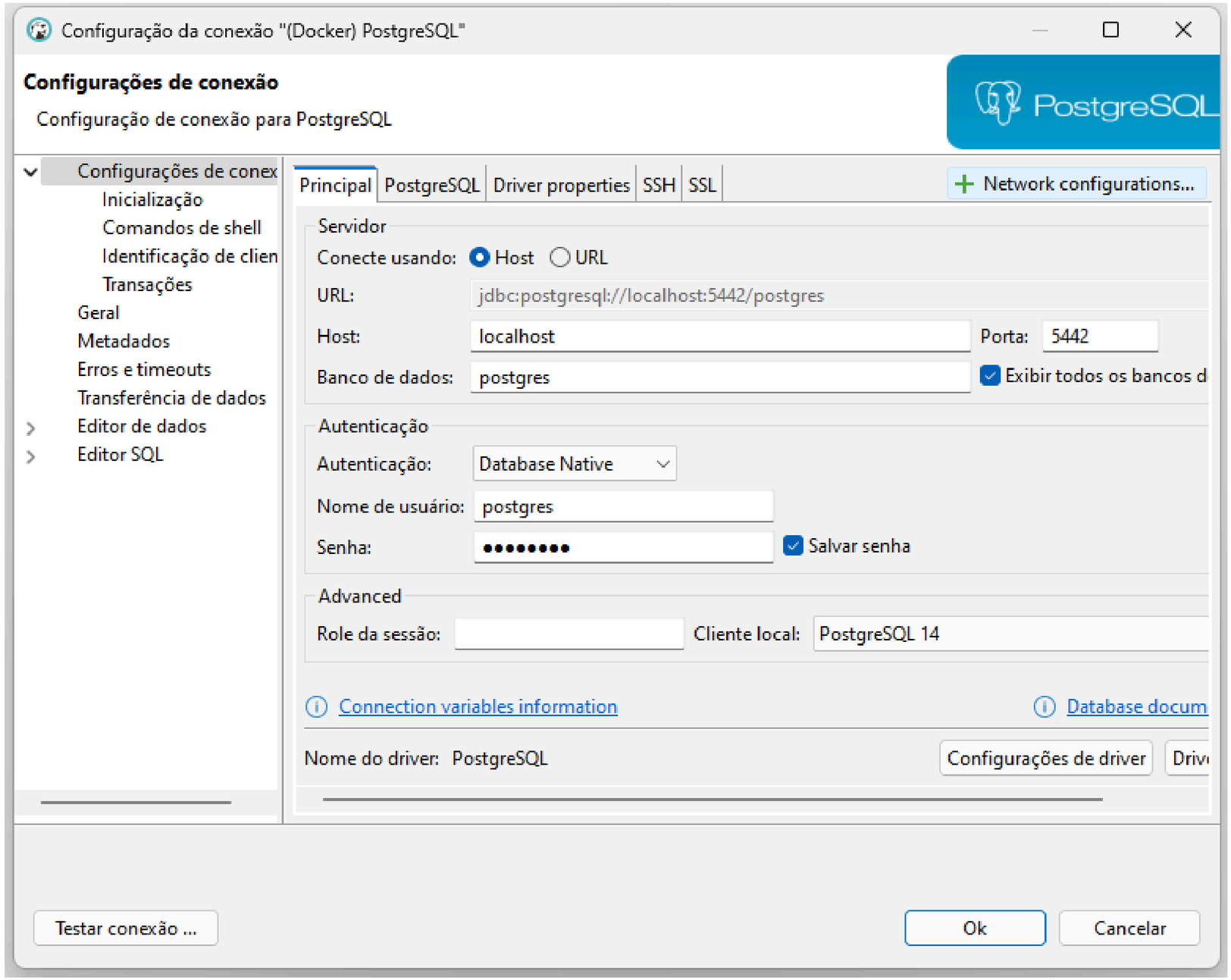
Execute o script 03 - oracle.sql



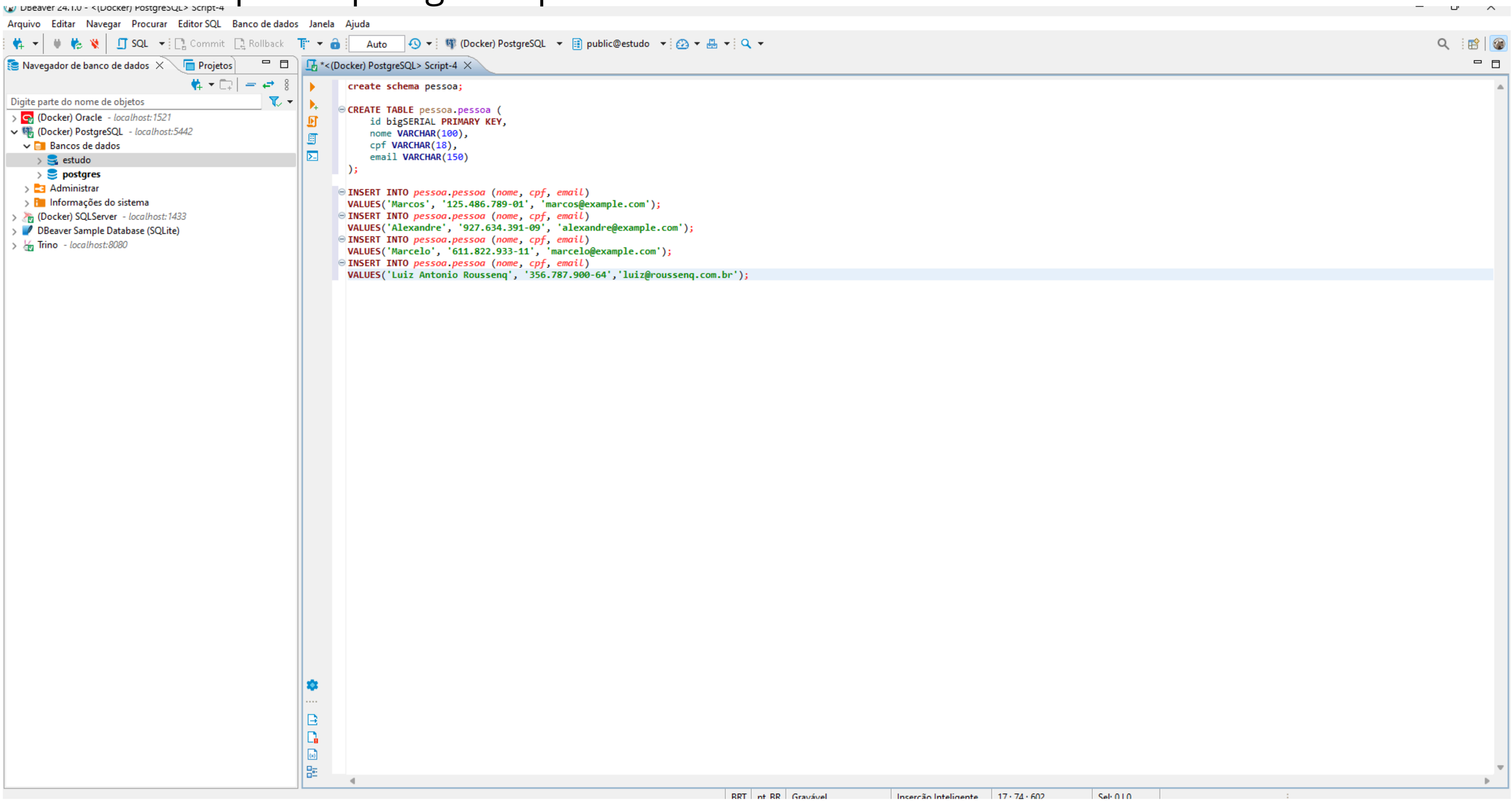
Configure a conexão com banco oracle



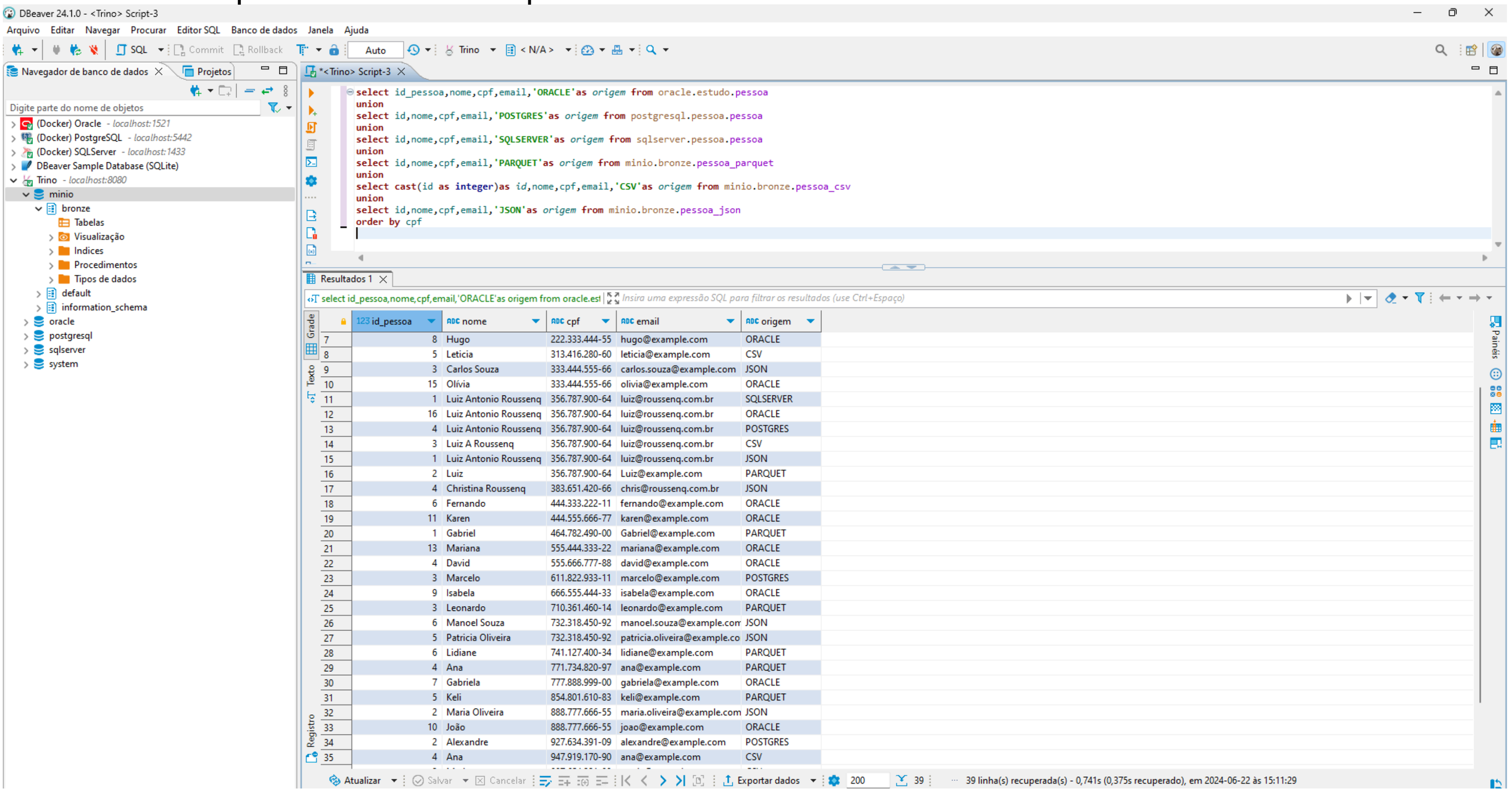
usuário = postgres
senha = postgres



Execute o script 04 - postgres.sql

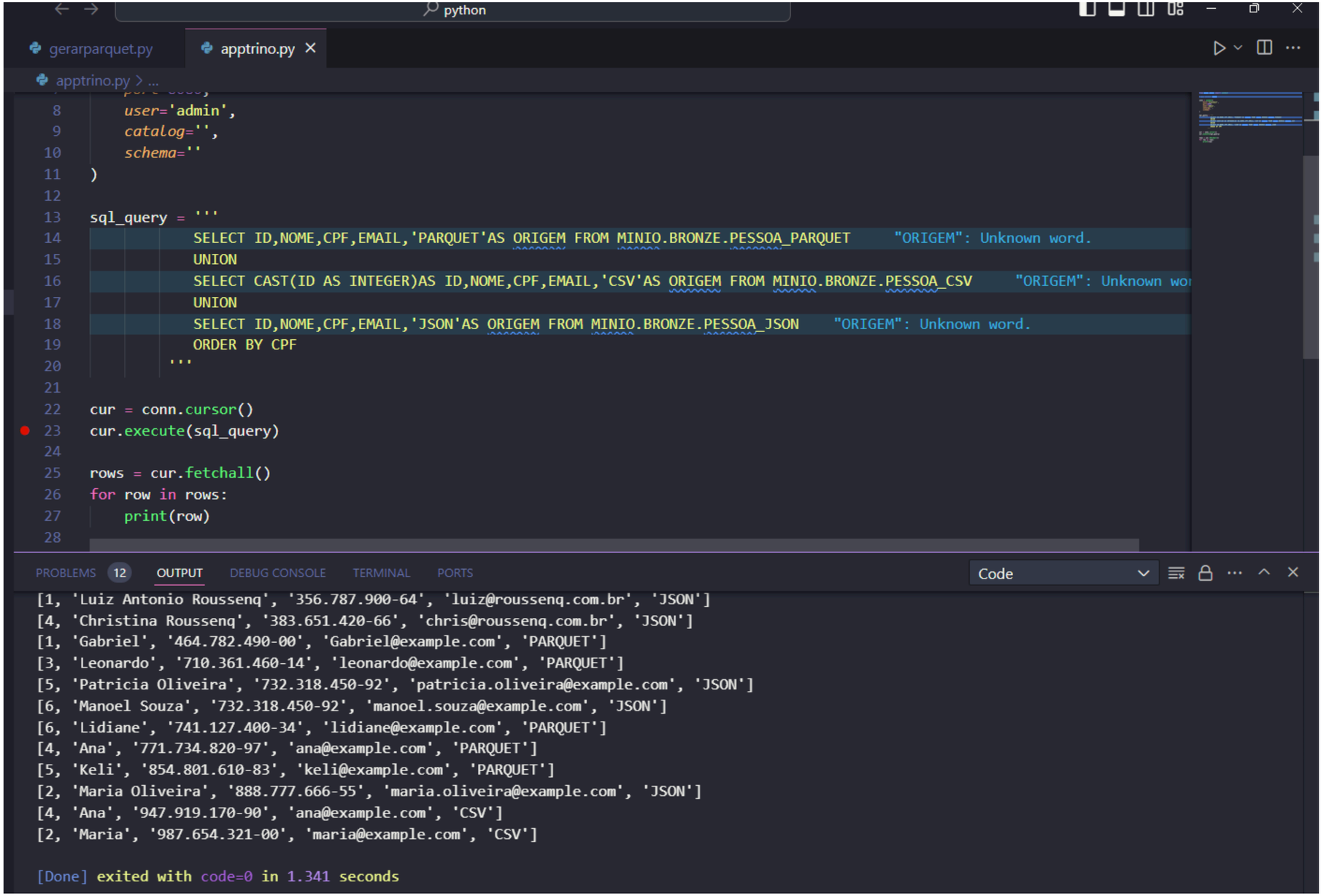


Execute o script 05- consulta.sql



	id_pessoa	nome	cpf	email	origem
7	8	Hugo	222.333.444-55	hugo@example.com	ORACLE
8	5	Leticia	313.416.280-60	leticia@example.com	CSV
9	3	Carlos Souza	333.444.555-66	carlos.souza@example.com	JSON
10	15	Olívia	333.444.555-66	olivia@example.com	ORACLE
11	1	Luiz Antonio Roussenq	356.787.900-64	luiz@roussenq.com.br	SQLSERVER
12	16	Luiz Antonio Roussenq	356.787.900-64	luiz@roussenq.com.br	ORACLE
13	4	Luiz Antonio Roussenq	356.787.900-64	luiz@roussenq.com.br	POSTGRES
14	3	Luiz A Roussenq	356.787.900-64	luiz@roussenq.com.br	CSV
15	1	Luiz Antonio Roussenq	356.787.900-64	luiz@roussenq.com.br	JSON
16	2	Luiz	356.787.900-64	Luiz@example.com	PARQUET
17	4	Christina Roussenq	383.651.420-66	chris@roussenq.com.br	JSON
18	6	Fernando	444.333.222-11	fernando@example.com	ORACLE
19	11	Karen	444.555.666-77	karen@example.com	ORACLE
20	1	Gabriel	464.782.490-00	Gabriel@example.com	PARQUET
21	13	Mariana	555.444.333-22	mariana@example.com	ORACLE
22	4	David	555.666.777-88	david@example.com	ORACLE
23	3	Marcelo	611.822.933-11	marcelo@example.com	POSTGRES
24	9	Isabela	666.555.444-33	isabela@example.com	ORACLE
25	3	Leonardo	710.361.460-14	leonardo@example.com	PARQUET
26	6	Manoel Souza	732.318.450-92	manoel.souza@example.com	JSON
27	5	Patricia Oliveira	732.318.450-92	patricia.oliveira@example.co	JSON
28	6	Lidiane	741.127.400-34	lidiane@example.com	PARQUET
29	4	Ana	771.734.820-97	ana@example.com	PARQUET
30	7	Gabriela	777.888.999-00	gabriela@example.com	ORACLE
31	5	Keli	854.801.610-83	keli@example.com	PARQUET
32	2	Maria Oliveira	888.777.666-55	maria.oliveira@example.com	JSON
33	10	João	888.777.666-55	joao@example.com	ORACLE
34	2	Alexandre	927.634.391-09	alexandre@example.com	POSTGRES
35	4	Ana	947.919.170-90	ana@example.com	CSV

Execute a apptrino



docker-compose down