

Universidade de Araraquara
Bacharel em Engenharia de Computação

Gustavo Santos Voltolino

Projeto de Fundamentos Banco de Dados 2
Database de E-commerce Dona Baunilha

Matão, São Paulo
2025

1. Introdução

O projeto tem como objetivo simular um ambiente de análise de dados para uma cafeteria fictícia chamada “**Dona Baunilha**”, que possui três unidades localizadas nas cidades de **Araraquara**, **Matão** e **São Carlos**. A proposta é criar um banco de dados relacional com tabelas de fato e dimensão, realizar a carga dos dados no Databricks e aplicar análises exploratórias para responder perguntas de negócio.

O ambiente utilizado para este projeto é o **Databricks Community Edition**, uma plataforma gratuita que permite processamento de dados em nuvem com suporte para SQL, Python, Spark, entre outros.

Como bônus, realizei a transferência da plataforma Databricks para um banco de dados SQL local, utilizando SSMS (SQL Server Management Studio) e SQL Server Express, para posteriormente consumir esses dados no Power BI.

Repositório Completo:

<https://github.com/guvoltolino/bd2-projeto-donabaunilha>

2. Ambiente

O projeto foi desenvolvido na plataforma Databricks Community Edition, que permite processar dados em nuvem gratuitamente. O ambiente oferece suporte a linguagens como SQL, Python e Spark, além de notebooks interativos para desenvolvimento.

Foi criado um cluster de processamento para executar os comandos SQL e manipular os dados. Toda a construção do banco de dados — desde a criação das tabelas até as análises — foi realizada dentro de notebooks SQL, permitindo integração entre código, visualização de dados e geração de gráficos de forma prática e eficiente.

O Databricks fornece armazenamento temporário para os dados e permite que o usuário utilize um ambiente escalável, moderno e voltado para análise de dados.

Para criar uma conta gratuita no Databricks Community, basta inscrever-se em: <https://community.cloud.databricks.com>

2.1 Notebooks

O desenvolvimento do projeto foi dividido em dois notebooks principais, cada um com funções específicas e bem definidas para organizar o fluxo de trabalho de forma lógica e eficiente.

00. Criação do Database e Inserção de Dados:

Este notebook é responsável pela estruturação do banco de dados e população inicial dos dados. As principais atividades desenvolvidas foram:

- Criação do Database: Definição do banco chamado **donabaunilha**, que simula os dados operacionais da cafeteria fictícia *Dona Baunilha*.
- Criação das Tabelas: Modelagem de dados com base no conceito de Data Warehouse, utilizando tabelas dimensionais e uma tabela fato:
 - Tabelas dimensionais: **dcliente**, **dproduto**, **dcategoria**, **dbu** (unidade de negócio).
 - Tabela fato: **fvendas**.

- Tabela auxiliar de itens de pedido: `ditemorder`.
- Inserção dos Dados: Geração e inserção de dados simulados para as tabelas, respeitando regras de integridade referencial e coerência com o cenário real de uma cafeteria.
- Validação: Consultas iniciais para garantir que os dados foram inseridos corretamente e que as tabelas estão relacionadas adequadamente.

01. Perguntas (Análises de Dados):

Neste notebook foram desenvolvidas as análises de negócio, com o objetivo de responder perguntas estratégicas. As principais atividades incluem:

- Consultas SQL para Análises: Desenvolvimento de queries para responder às seguintes perguntas:
 1. Quais são os clientes que estão há mais tempo sem comprar?
 2. Quem são os 10 clientes que mais compraram em um período específico?
 3. Quais são os três produtos mais vendidos em um determinado período?
 4. Quais foram os itens vendidos por categoria?
 5. Qual o valor total de vendas por categoria em um período?

3. Desenvolvimento

Primeiramente, criamos o schema/db e demos um drop caso exista algo com o mesmo nome.

Observação: O Databricks Community tem limitação de 2h de cluster ligado (nosso motor) e quando ele desliga perdemos tudo que criamos, por isso coloquei isso no código com objetivo de não ter problemas em rodar o notebook quantas vezes for necessário.

00. Criação do Database e Inserção de Dados

Python

File Edit View Run Help Last edit was 8 hours ago

Run all

Terminated

Share

Publish



8 hours ago (1s)

1

SQL



```
%sql
-- Criando Schema/Database
CREATE SCHEMA IF NOT EXISTS donabaunilha;

-- Apagando Tabela Dimensões e Fato
DROP TABLE IF EXISTS donabaunilha.dendereco;
DROP TABLE IF EXISTS donabaunilha.dcliente;
DROP TABLE IF EXISTS donabaunilha.dbu;
DROP TABLE IF EXISTS donabaunilha.dcategoria;
DROP TABLE IF EXISTS donabaunilha.dproduto;
DROP TABLE IF EXISTS donabaunilha.ditemorder;
DROP TABLE IF EXISTS donabaunilha.fvendas;
```

OK

8 hours ago (2s)

2

```
#Limando os registro delta
dbutils.fs.rm("dbfs:/user/hive/warehouse/donabaunilha/dendereco", True)
dbutils.fs.rm("dbfs:/user/hive/warehouse/donabaunilha/dcliente", True)
dbutils.fs.rm("dbfs:/user/hive/warehouse/donabaunilha/dbu", True)
dbutils.fs.rm("dbfs:/user/hive/warehouse/donabaunilha/dcategoria", True)
dbutils.fs.rm("dbfs:/user/hive/warehouse/donabaunilha/dproduto", True)
dbutils.fs.rm("dbfs:/user/hive/warehouse/donabaunilha/ditemorder", True)
```

Criação da Tabela dendereco e inserção de dados:

00. Criação do Database e Inserção de Dados

Python

File Edit View Run Help Last edit was 8 hours ago

Run all

Terminated

Share

Publish

```
True

%sql

-- Criando a Tabela Dimensão de Endereço
CREATE TABLE IF NOT EXISTS donabaunilha.dendereco (
  id_endereco INT,
  logradouro VARCHAR(50),
  numero INT,
  complemento VARCHAR(50),
  cidade VARCHAR(50),
  estado VARCHAR(50),
  cep VARCHAR(10)
) USING DELTA LOCATION 'dbfs:/user/hive/warehouse/donabaunilha/dendereco';

(3) Spark Jobs
OK

%sql

--Inserindo dados na Tabela Endereco
INSERT INTO donabaunilha.dendereco(id_endereco, logradouro, numero, complemento, cidade, estado, cep)
VALUES
(1, 'Avenida das Flores', 168, 'Casa', 'Matão', 'São Paulo', '15990-000').
```

Criação da Tabela dcliente e inserção de dados:

```
5

%sql

-- Criando a Tabela Dimensão de Cliente
CREATE TABLE IF NOT EXISTS donabaunilha.dcliente (
  id_cliente INT,
  id_endereco INT,
  nome VARCHAR(50),
  email VARCHAR(50),
  cpf VARCHAR(15),
  telefone VARCHAR(50)
) USING DELTA LOCATION 'dbfs:/user/hive/warehouse/donabaunilha/dcliente';

(3) Spark Jobs
OK

6

%sql

--Inserindo dados na Tabela Cliente

INSERT INTO donabaunilha.dcliente (id_cliente, id_endereco, nome, cpf, email, telefone) VALUES
(1, 1, 'Sra. Ana Clara Vieira', '716.305.289-86', 'zda-conceicao@da.br', '+55 (011) 4985-8427'),
(2, 2, 'Eduardo Oliveira', '895.402.361-42', 'souzaisabel@gmail.com', '61 4739-6356'),
(3, 3, 'Isabella da Cunha', '386.294.175-28', 'davi03@uol.com.br', '0500 729 6367'),
(4, 4, 'Ana Livia Araújo', '743.520.816-71', 'heloisaalves@moura.com', '+55 21 6680 6717'),
(5, 5, 'Pietra Castro', '849.150.637-39', 'mariana72@ig.com.br', '(084) 0706-0290'),
(6, 6, 'Sra. Bárbara Rocha', '760.512.348-53', 'yasmin11@teixeira.com', '(081) 3842 7079'),
(7, 7, 'Milena das Neves', '649.087.125-49', 'naquel82@carvalho.com', '+55 84 4860 3117'),
(8, 8, 'João Teixeira', '623.075.894-65', 'igor89@yahoo.com.br', '+55 (041) 6098 0997'),
(9, 9, 'Beatriz Souza', '378.921.045-50', 'tda-paz@hotmail.com', '+55 (051) 8274 1007'),
(10, 10, 'Kaique da Cunha', '104.367.589-20', 'rochavitor@caldeira.br', '+55 21 1026 2100'),
(11, 11, 'Yago Barros', '497.365.802-29', 'bernardocorreia@castro.com', '0500-918-2602'),
(12, 12, 'Dr. João Miguel Dias', '150.293.746-80', 'epires@ig.com.br', '(081) 6103 4451'),
(13, 13, 'Alexia Dias', '460.538.917-20', 'thiago91@dias.org', '61 0738 7553'),
(14, 14, 'João Pedro Nunes', '843.072.916-03', 'ffarias@moura.br', '+55 21 3971 4648'),
(15, 15, 'Srta. Bruna Martins', '874.531.962-64', 'julia00@hotmail.com', '71 8554 3017'),
(16, 16, 'Gustavo Araújo', '380.572.169-21', 'catarinaaraujo@peixoto.br', '(041) 5260 1048'),
(17, 17, 'Raquel Dias', '752.396.814-73', 'arthurda-paz@cardoso.com', '(081) 6075 3258'),
(18, 18, 'Sr. Paulo Carvalho', '318.509.627-40', 'da-motamaria-luiza@carvalho.com', '71 7582 4129'),
```

Criação da Tabela dbu e inserção de dados:

00. Criação do Database e Inserção de Dados Python

File Edit View Run Help Last edit was 8 hours ago

Run all Terminated Share Publish

8 hours ago (2s) 7

```
%sql

-- Criando a Tabela Dimensão de BU
CREATE TABLE IF NOT EXISTS donabaunilha.dbu (
  id_bu INT,
  nome VARCHAR(50),
  logradouro VARCHAR(50),
  numero INT,
  cidade VARCHAR(50),
  estado VARCHAR(50),
  cep VARCHAR(10)
) USING DELTA LOCATION 'dbfs:/user/hive/warehouse/donabaunilha/dbu';
```

(3) Spark Jobs

OK

8 hours ago (3s) 8

```
%sql

--Inserindo dados na Tabela BU

INSERT INTO donabaunilha.dbu (id_bu, nome, logradouro, numero, cidade, estado, cep) VALUES
(1, 'Dona Baunilha Araraquara', 'Av. Principal', 100, 'Araraquara', 'SP', '14800-000'),
(2, 'Dona Baunilha Matão', 'Rua do Café', 200, 'Matão', 'SP', '15990-000'),
(3, 'Dona Baunilha São Carlos', 'Av. Central', 300, 'São Carlos', 'SP', '13560-000');
```

(6) Spark Jobs

Criação da Tabela dcategoria e inserção de dados:

8 hours ago (2s) 9

```
%sql

-- Criando a Tabela Dimensão de Categoria
CREATE TABLE IF NOT EXISTS donabaunilha.dcategoria (
  id_categoria INT,
  nome VARCHAR(50)
) USING DELTA LOCATION 'dbfs:/user/hive/warehouse/donabaunilha/dcategoria';
```

(3) Spark Jobs

OK

8 hours ago (2s) 10

```
%sql

--Inserindo dados na Tabela Dimensão de Categoria

INSERT INTO donabaunilha.dcategoria (id_categoria, nome) VALUES
(1, 'Bebidas Quentes'),
(2, 'Bebidas Geladas'),
(3, 'Doces'),
(4, 'Salgados'),
(5, 'Brunch');
```

Criação da Tabela dproduto e inserção de dados:

```
▶ ✓ 8 hours ago (2s) 11

%sql

-- Criando a Tabela Dimensão de Produto
CREATE TABLE IF NOT EXISTS donabaunilha.dproduto (
  id_produto INT,
  id_categoria INT,
  nome VARCHAR(50),
  preco_unit FLOAT,
  custo_prod FLOAT
) USING DELTA LOCATION 'dbfs:/user/hive/warehouse/donabaunilha/dproduto';

▶ (3) Spark Jobs
OK
```

```
▶ ✓ 8 hours ago (3s) 12

%sql

--Inserindo dados na Tabela Produto

INSERT INTO donabaunilha.dproduto (id_produto, id_categoria, nome, preco_unit, custo_prod) VALUES
(1, 3, 'Torta de Limão 1', 109.78, 36.02),
(2, 6, 'Pão de Queijo Gourmet 2', 46.37, 19.63),
(3, 1, 'Frapuccino Dona Baunilha 3', 57.36, 15.12),
(4, 6, 'Frapuccino Dona Baunilha 4', 19.57, 8.93),
(5, 4, 'Capuccino Cremoso 5', 21.94, 10.92),
```

Criação da Tabela fvendas e inserção de dados:

```
▶ ✓ 8 hours ago (2s) 13

--Criando a Tabela fvendas

CREATE TABLE IF NOT EXISTS donabaunilha.fvendas (
  id_venda INT,
  id_cliente INT,
  id_bu INT,
  valor_total DECIMAL(10,2),
  desconto DECIMAL(10,2),
  status VARCHAR(20),
  data_venda TIMESTAMP
) USING DELTA
LOCATION 'dbfs:/user/hive/warehouse/donabaunilha/fvendas';

▶ (3) Spark Jobs
OK
```

```
▶ ✓ 8 hours ago (3s) 14

%sql

--Inserindo dados na tabela fvendas

INSERT INTO donabaunilha.fvendas (id_venda, id_cliente, id_bu, valor_total, desconto, status, data_venda) VALUES
(1, 47, 3, 428.71, 43.84, 'Pendente', '2024-05-16 22:17:38'),
(2, 15, 2, 320.07, 49.1, 'Cancelada', '2024-05-12 09:31:53'),
```

Criação da Tabela ditemorder e inserção de dados:


```
▶ ✓ 8 hours ago (2s) 15
--Criando a tabela ditemorder

CREATE TABLE IF NOT EXISTS donabaunilha.ditemorder (
  id_item INT,
  id_produto INT,
  id_venda INT,
  qtd_item INT,
  valor_item DECIMAL(10,2)
) USING DELTA
LOCATION 'dbfs:/user/hive/warehouse/donabaunilha/ditemorder';

▶ (3) Spark Jobs
OK

▶ ✓ 8 hours ago (3s) 16

%sql

--Inserindo dados na tabela ditemorder

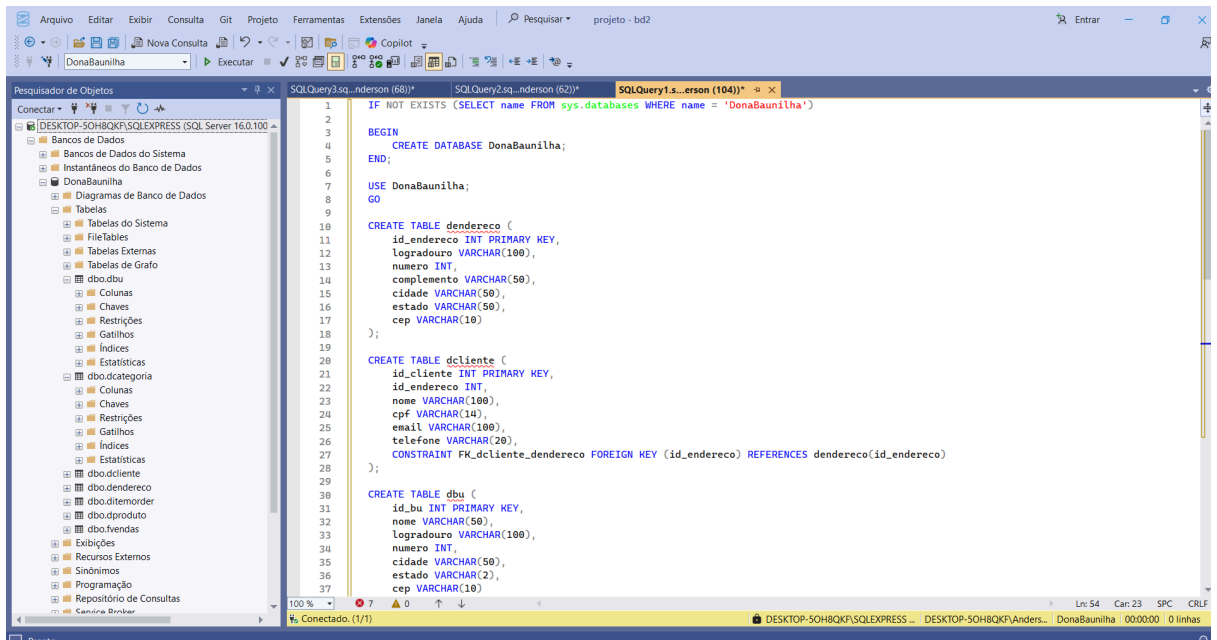
INSERT INTO donabaunilha.ditemorder (id_item, id_produto, id_venda, qtd_item, valor_item) VALUES
(1, 42, 82, 5, 369.15),
(2, 49, 5, 3, 242.82),
(3, 36, 176, 1, 77.76),
(4, 31, 75, 2, 169.66).
```

E após isso foi criado o notebook respondendo as 5 perguntas iniciais. Segue código completo em repositório: <https://github.com/guvoltolino/bd2-projeto-donabaunilha>

4. Banco de Dados SQL

O projeto também foi realizado utilizando SSMS e SQL SERVER, utilizando a linguagem T-SQL, queries completas no repositório. Abaixo seguem anexos:

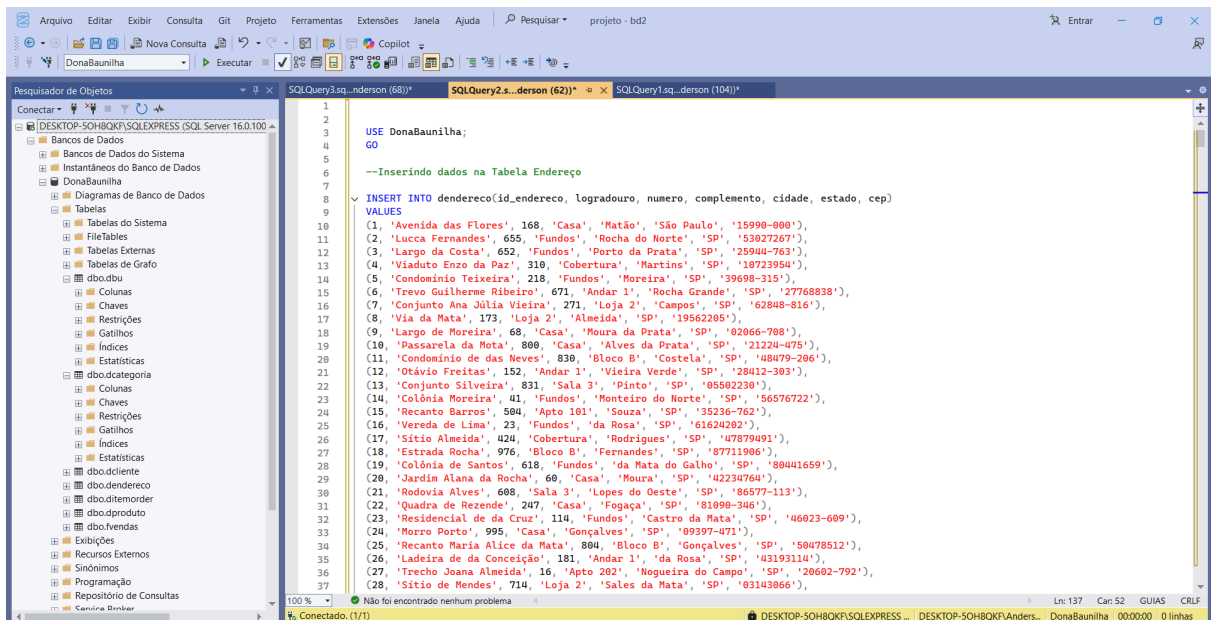
Criação do Database e Tabelas:



The screenshot shows the SQL Server Enterprise Manager interface on the left, displaying the 'DonaBaunilha' database under 'Bancos de Dados'. The right pane shows a SQL Query window with the following T-SQL script:

```
1 IF NOT EXISTS (SELECT name FROM sys.databases WHERE name = 'DonaBaunilha')
2
3 BEGIN
4     CREATE DATABASE DonaBaunilha;
5 END;
6
7 USE DonaBaunilha;
8 GO
9
10 CREATE TABLE dendereco (
11     id_endereco INT PRIMARY KEY,
12     logradouro VARCHAR(100),
13     numero INT,
14     complemento VARCHAR(50),
15     cidade VARCHAR(50),
16     estado VARCHAR(50),
17     cep VARCHAR(10)
18 );
19
20 CREATE TABLE dcliente (
21     id_cliente INT PRIMARY KEY,
22     id_endereco INT,
23     nome VARCHAR(100),
24     cpf VARCHAR(14),
25     email VARCHAR(100),
26     telefone VARCHAR(20),
27     CONSTRAINT FK_dcliente_dendereco FOREIGN KEY (id_endereco) REFERENCES dendereco(id_endereco)
28 );
29
30 CREATE TABLE dbu (
31     id_bu INT PRIMARY KEY,
32     nome VARCHAR(50),
33     logradouro VARCHAR(100),
34     numero INT,
35     cidade VARCHAR(50),
36     estado VARCHAR(2),
37     cep VARCHAR(10)
38 );
```

Inserção de dados nas Tabelas:



The screenshot shows the SQL Query window with the following T-SQL script:

```
1
2
3 USE DonaBaunilha;
4 GO
5
6 --Inserindo dados na Tabela Endereço
7
8 INSERT INTO dendereco(id_endereco, logradouro, numero, complemento, cidade, estado, cep)
9 VALUES
10 (1, 'Avenida das Flores', 168, 'Casa', 'Matão', 'São Paulo', '15998-000'),
11 (2, 'Lucca Fernandes', 655, 'Fundos', 'Rocha do Norte', 'SP', '53027267'),
12 (3, 'Largo da Costa', 652, 'Fundos', 'Porto da Prata', 'SP', '25944-763'),
13 (4, 'Vizinho Enzo da Paz', 310, 'Cobertura', 'Martins', 'SP', '18723954'),
14 (5, 'Condomínio Teixeira', 218, 'Fundos', 'Moreira', 'SP', '39650-315'),
15 (6, 'Trevo Guilherme Ribeiro', 671, 'Andar 1', 'Rocha Grande', 'SP', '27768838'),
16 (7, 'Conjunto Ana Júlia Vieira', 271, 'Loja 2', 'Campos', 'SP', '62848-816'),
17 (8, 'Via da Mata', 173, 'Loja 2', 'Almeida', 'SP', '19562205'),
18 (9, 'Largo de Moreira', 68, 'Casa', 'Moura da Prata', 'SP', '02866-708'),
19 (10, 'Passarela da Mota', 809, 'Casa', 'Alves da Prata', 'SP', '21224-475'),
20 (11, 'Condomínio de das Neves', 830, 'Bloco B', 'Costela', 'SP', '48079-206'),
21 (12, 'Otávio Freitas', 152, 'Andar 1', 'Vieira Verde', 'SP', '28012-303'),
22 (13, 'Conjunto Silveira', 831, 'Sala 3', 'Pinto', 'SP', '05502230'),
23 (14, 'Colônia Moreira', 41, 'Fundos', 'Monteiro do Norte', 'SP', '56576722'),
24 (15, 'Recanto Barros', 504, 'Apto 101', 'Souza', 'SP', '35236-762'),
25 (16, 'Vereda de Lima', 23, 'Fundos', 'da Rosa', 'SP', '61624202'),
26 (17, 'Sítio Almeida', 424, 'Cobertura', 'Rodrigues', 'SP', '47879491'),
27 (18, 'Estrada Rocha', 976, 'Bloco B', 'Fernandes', 'SP', '87711906'),
28 (19, 'Colônia de Santos', 618, 'Fundos', 'da Mata do Galho', 'SP', '80441659'),
29 (20, 'Jardim Alana da Rocha', 60, 'Casa', 'Moura', 'SP', '42234764'),
30 (21, 'Rodovia Alves', 608, 'Sala 3', 'Lopes do Oeste', 'SP', '86577-113'),
31 (22, 'Quadra de Rezende', 247, 'Casa', 'Fogaça', 'SP', '81998-366'),
32 (23, 'Residencial de da Cruz', 114, 'Fundos', 'Castro da Mata', 'SP', '46023-609'),
33 (24, 'Morro Porto', 995, 'Casa', 'Gonçalves', 'SP', '89397-471'),
34 (25, 'Recanto Maria Alice da Mata', 804, 'Bloco B', 'Gonçalves', 'SP', '50478512'),
35 (26, 'Ladeira de da Conceição', 181, 'Andar 1', 'da Rosa', 'SP', '43193114'),
36 (27, 'Trecho Joana Almeida', 16, 'Apto 202', 'Nogueira do Campo', 'SP', '20602-792'),
37 (28, 'Sítio de Mendes', 714, 'Loja 2', 'Sales da Mata', 'SP', '03143066'),
```

Respostas das Perguntas de Negócio:

Arquivo Editar Exibir Consultas Git Projetos Ferramentas Extensões Janela Ajuda Pesquisa projeto - bd2

Entrar

DonaBaunilha

Pesquisador de Objetos

Conectar

DESKTOP-SOH8QKF\SQLEXPRESS (SQL Server 16.0.100)

Bancos de Dados
Bancos de Dados do Sistema
Instantâneos do Banco de Dados
DonaBaunilha
Diagramas de Banco de Dados
Tabelas
Tabelas do Sistema
FileTables
Tabelas Externas
Tabelas de Grafo
dbo.dbo
Colunas
Chaves
Restrições
Gatilhos
Índices
Estatísticas
dbo.dcatlogia
Colunas
Chaves
Restrições
Gatilhos
Índices
Estatísticas
dbo.dcliente
dbo.dendereço
dbo.ditemizador
dbo.dproduto
dbo.dvendas
Exibições
Recursos Externos
Sinônimos
Programação
Repositório de Consultas
Camada Roubler

SQLQuery3.s_derson (68)* SQLQuery2.sq...nderson (62)* SQLQuery1.sq...derson (104)*

```
40 WHERE v.data_venda BETWEEN '2024-01-01' AND '2024-12-31'  
41 AND v.status = 'Concluida'  
42 GROUP BY c.nome  
43 ORDER BY Valor_Total DESC;  
44  
45 --Quais foram os 10 clientes que mais compraram em um dado periodo?  
46  
47 SELECT TOP 10  
48     c.nome AS Cliente,  
49     ROUND(SUM(v.valor_total), 2) AS Total_Gasto,  
50     b.nome AS Unidade_Negocio  
51 FROM fvwendas v  
52 JOIN dcliente c ON v.id_cliente = c.id_cliente  
53 JOIN dbu b ON v.id_bu = b.id_bu  
54 WHERE v.data_venda BETWEEN '2024-01-01' AND '2024-12-31'  
55 AND v.status = 'Concluida'  
56 GROUP BY c.nome, b.nome  
57 ORDER BY Total_Gasto DESC;  
58  
59 --Quais são os clientes qua estão há mais tempo sem comprar nada?
```

100 % Não foi encontrado nenhum problema Ln: 55 Car: 58 SPC CRLF

Resultados Mensagens

	Cliente	Total_Gasto	Unidade_Negocio
1	Luigi Pereira	1202.47	Dona Baunilha Araraquara
2	Emily da Cunha	911.00	Dona Baunilha São Carlos
3	Gabriela Gonçalves	862.39	Dona Baunilha Araraquara
4	Luiz Fernando Campos	778.37	Dona Baunilha Matão
5	Sra. Ana Clara Vieira	769.44	Dona Baunilha Matão
6	Gabriela Gonçalves	544.74	Dona Baunilha Matão
7	Sra. Clara Castro	636.47	Dona Baunilha Araraquara
8	Beatriz Souza	554.94	Dona Baunilha São Carlos

	Cliente	ultima compra
1	Yuri Lopes	2024-05-09
2	Nathan Fogaca	2024-05-10
3	Milena das Neves	2024-05-12

Consulta executada com êxito.

DESKTOP-SOH8QKF\SQLEXPRESS - DESKTOP-SOH8QKF\Anders... DonaBaunilha 00:00:00 57 linhas

5. Conclusão

Este projeto demonstra como estruturar um banco de dados analítico no Databricks Community Edition, desde a modelagem até a geração de insights. Através dele, foi possível responder perguntas de negócio fundamentais para a cafeteria Dona Baunilha, auxiliando na tomada de decisões estratégicas.

Por fim, fizemos a ingestão dos dados do banco de dados do SQL no Power BI, respondendo mais perguntas de negócio. Demonstrado na imagem abaixo:

