

Cliente CoffeKing (conjunto de dados acadêmicos do Yelp)

Autor: Alan Ferrari Data: 21 de Outubro de 2025

Marco 1: Proposta de projeto e seleção/preparação de dados

1. Resumo:

Este trabalho aborda a aplicação prática de conceitos de ETL (Extração, Transformação e Cargas), construção de um DW (Data Warehouse ou Armazém de Dados) e análise exploratória dos dados.

2. Descrição

A CoffeeKing é uma nova empresa de café que está começando a oferecer uma experiência única e inovadora aos seus clientes e quer atrair uma grande variedade de clientela.

Este projeto analisará o dataset do Yelp para fornecer à CoffeeKing insights sobre o que impulsiona o sucesso de cafeterias concorrentes. A análise focará em identificar os principais temas (positivos e negativos) mencionados em avaliações de clientes, bem como quais atributos do negócio (ex: ambiente, preço, Wi-Fi) estão mais correlacionados com avaliações altas. O público-alvo são os diretores de marketing e operações da CoffeeKing, que usarão essas descobertas para definir sua estratégia de serviço, marketing e design de loja.

3. Perguntas

Quais são os tópicos mais comuns (ex: "sabor do café", "atendimento", "ambiente", "preço", "Wi-Fi", "local", "horário de funcionamento") mencionados em avaliações de cafeterias, e como a frequência desses tópicos difere entre avaliações de 1 estrela e 5 estrelas?

Quais atributos de negócio têm o maior impacto (positivo ou negativo) na nota média de uma cafeteria?

Usuários "experientes" do Yelp (alto review_count ou status elite) dão notas diferentes ou focam em aspectos diferentes em comparação com usuários "novatos"?

4. Hipótese

Para cafeterias, o "ambiente" (mencionado em tb_review.text como "ambiente", "música", "confortável") e o "atendimento" serão fatores decisivos mais fortes para avaliações de 5 estrelas do que o "preço".

A disponibilidade de "Wi-Fi Grátis" (extraído de tb_business.attributes) será um dos principais fatores de correlação positiva para cafeterias com alto volume de reviews (review_count), mesmo que a nota (stars) não seja perfeita.

Usuários "Elite" (tb_user.elite) tendem a ser mais críticos, resultando em uma nota média ligeiramente menor para os mesmos locais em comparação com usuários não-elite, mas seus reviews mencionarão mais o "sabor" ou "qualidade" do café.

5. Abordagem

A análise será focada inteiramente em negócios da categoria "Café". O primeiro passo em todas as consultas será filtrar a tb_business usando WHERE categories ILIKE '%coffee%' ou ILIKE '%cafes%'.

1. **Análise Temática (H1):** Usaremos consultas LIKE no tb_review.text (ex: WHERE text ILIKE '%ambiente%') agrupadas por stars para contar a frequência de temas-chave e validar a hipótese.
2. **Análise de Atributos (H2):** Faremos JOIN entre tb_business e tb_review. Usaremos as funções JSONB do PostgreSQL para extrair dados de tb_business.attributes (como attributes->>'WiFi') e calcularemos a AVG(stars) para cada grupo.
3. **Análise Demográfica (H3):** Faremos JOIN entre tb_review e tb_user. Criaremos grupos de usuários (ex: CASE WHEN review_count > 100 THEN 'Experiente' ELSE 'Novato' END) e compararemos a AVG(stars) e a frequência de palavras-chave entre esses grupos.

6. Modelo de dados:

Para a modelagem de dados foi criado um no PostgreSQL um novo database juntamente com as tabelas relacionadas a cada dataset

Database

```
CREATE DATABASE yelp_db;
```

Tabela user

```
CREATE TABLE tb_user (
    user_id          VARCHAR(255) PRIMARY KEY,
    name             TEXT,
    review_count     INTEGER,
    yelping_since    TIMESTAMP,
    useful           INTEGER,
    funny            INTEGER,
    cool              INTEGER,
    elite             TEXT,
    friends          TEXT,
    fans              INTEGER,
    average_stars    FLOAT,
    compliment_hot   INTEGER,
    compliment_more  INTEGER,
    compliment_profile INTEGER,
    compliment_cute  INTEGER,
    compliment_list   INTEGER,
    compliment_note  INTEGER,
    compliment_plain  INTEGER,
    compliment_cool  INTEGER,
    compliment_funny INTEGER,
    compliment_writer INTEGER,
    compliment_photos INTEGER);
```

Comentário da Tabela

COMMENT ON TABLE public.tb_user IS 'Tabela de usuários. Contém dados demográficos e de perfil de cada usuário. Criada por Alan Ferrari em 21/10/2025.';

Comentários das Colunas

COMMENT ON COLUMN public.tb_user.user_id IS 'ID único do usuário. Chave Primária (PK) da tabela tb_user.';
COMMENT ON COLUMN public.tb_user.name IS 'Primeiro nome do usuário (pode ser nulo ou não confiável).';
COMMENT ON COLUMN public.tb_user.review_count IS 'Número total de reviews (avaliações) escritas por este usuário.';
COMMENT ON COLUMN public.tb_user.yelping_since IS 'Data e hora de quando o usuário se cadastrou no Yelp.';
COMMENT ON COLUMN public.tb_user.useful IS 'Número total de votos "Útil" recebidos pelos reviews deste usuário.';
COMMENT ON COLUMN public.tb_user.funny IS 'Número total de votos "Engraçado" recebidos pelos reviews deste usuário.';
COMMENT ON COLUMN public.tb_user.cool IS 'Número total de votos "Legal" recebidos pelos reviews deste usuário.';
COMMENT ON COLUMN public.tb_user.elite IS 'String com os anos em que o usuário foi "Elite", separados por vírgula (ex: "2017,2018").';
COMMENT ON COLUMN public.tb_user.friends IS 'String longa contendo uma lista de user_ids de amigos. Requer tratamento (parsing).';
COMMENT ON COLUMN public.tb_user.fans IS 'Número de usuár que seguem este usuário (fãs).';
COMMENT ON COLUMN public.tb_user.average_stars IS 'Média de estrelas dadas por este usuário em todos os seus reviews.';
COMMENT ON COLUMN public.tb_user.compliment_hot IS 'Contagem de elogios do tipo "Hot" recebidos.';
COMMENT ON COLUMN public.tb_user.compliment_more IS 'Contagem de elogios do tipo "More" recebidos.';
COMMENT ON COLUMN public.tb_user.compliment_profile IS 'Contagem de elogios do tipo "Profile" recebidos.';
COMMENT ON COLUMN public.tb_user.compliment_cute IS 'Contagem de elogios do tipo "Cute" recebidos.';
COMMENT ON COLUMN public.tb_user.compliment_list IS 'Contagem de elogios do tipo "List" recebidos.';
COMMENT ON COLUMN public.tb_user.compliment_note IS 'Contagem de elogios do tipo "Note" recebidos.';
COMMENT ON COLUMN public.tb_user.compliment_plain IS 'Contagem de elogios do tipo "Plain" recebidos.';
COMMENT ON COLUMN public.tb_user.compliment_cool IS 'Contagem de elogios do tipo "Cool" recebidos.';
COMMENT ON COLUMN public.tb_user.compliment_funny IS 'Contagem de elogios do tipo "Funny" recebidos.';
COMMENT ON COLUMN public.tb_user.compliment_writer IS 'Contagem de elogios do tipo "Writer" recebidos.';
COMMENT ON COLUMN public.tb_user.compliment_photos IS 'Contagem de elogios do tipo "Photos" recebidos.';

Tabela business

```
CREATE TABLE tb_business (
    business_id          VARCHAR(255) PRIMARY KEY,
    name                  TEXT,
    address               TEXT,
    city                  VARCHAR(255),
    state                 VARCHAR(5),
    postal_code           VARCHAR(20),
    latitude              FLOAT,
    longitude             FLOAT,
    stars                 FLOAT,
    review_count          INTEGER,
    is_open               INTEGER,
    attributes            JSONB,
    categories            TEXT,
    hours                JSONB);
```

Comentário da Tabela

COMMENT ON TABLE public.tb_business IS 'Tabela de negócios (restaurantes, lojas, etc.). Contém dados de localização, atributos e categorias. Criada por Alan Ferrari em 21/10/2025.';

Comentários das Colunas

COMMENT ON COLUMN public.tb_business.business_id IS 'ID único do negócio. Chave Primária (PK) da tabela tb_business.';

```

COMMENT ON COLUMN public.tb_business.name IS 'Nome oficial do negócio。';
COMMENT ON COLUMN public.tb_business.address IS 'Endereço completo do negócio。';
COMMENT ON COLUMN public.tb_business.city IS 'Cidade onde o negócio está localizado。';
COMMENT ON COLUMN public.tb_business.state IS 'Sigla do estado (ex: "PA", "CA")。';
COMMENT ON COLUMN public.tb_business.postal_code IS 'Código postal (CEP) do negócio。';
COMMENT ON COLUMN public.tb_business.latitude IS 'Coordenada geográfica de latitude。';
COMMENT ON COLUMN public.tb_business.longitude IS 'Coordenada geográfica de longitude。';
COMMENT ON COLUMN public.tb_business.stars IS 'Média de estrelas (avaliação) do negócio, de 1 a 5。';
COMMENT ON COLUMN public.tb_business.review_count IS 'Número total de reviews recebidos por este negócio。';
COMMENT ON COLUMN public.tb_business.is_open IS 'Indicador se o negócio está aberto (1) ou fechado (0)。';
COMMENT ON COLUMN public.tb_business.attributes IS 'Campo JSONB contendo diversos atributos (ex: "BusinessAcceptsCreditCards", "BusinessParking")。';
COMMENT ON COLUMN public.tb_business.categories IS 'String com a lista de categorias do negócio, separadas por vírgula。';
COMMENT ON COLUMN public.tb_business.hours IS 'Campo JSONB contendo os horários de funcionamento por dia da semana。';

```

Tabela checkin

```

CREATE TABLE tb_checkin (
    business_id          VARCHAR(255) REFERENCES tb_business(business_id),
    date                 TEXT;

```

Comentário da Tabela

COMMENT ON TABLE public.tb_checkin IS 'Tabela de check-ins. Contém uma string agregada de todas as datas de check-in para um negócio. Criada por Alan Ferrari em 21/10/2025。';

Comentários das Colunas

```

COMMENT ON COLUMN public.tb_checkin.business_id IS 'ID do negócio. Chave Estrangeira (FK) conceitual para tb_business。';
COMMENT ON COLUMN public.tb_checkin.date IS 'String de texto longa contendo todas as datas/horas de check-in, separadas por vírgula。';

```

Tabela tip

```

CREATE TABLE tb_tip (
    tip_id               SERIAL PRIMARY KEY,
    text                 TEXT,
    date                TIMESTAMP,
    compliment_count   INTEGER,
    user_id              VARCHAR(255) REFERENCES tb_user(user_id),
    business_id          VARCHAR(255) REFERENCES tb_business(business_id);

```

Comentário da Tabela

COMMENT ON TABLE public.tb_tip IS 'Tabela de "dicas" (tips). Tabela "fato" que liga usuários e negócios, contendo recomendações curtas. Criada por Alan Ferrari em 21/10/2025。';

Comentários das Colunas

```

COMMENT ON COLUMN public.tb_tip.tip_id IS 'ID único da dica (tip). Chave Primária (PK) auto-incremental (SERIAL)。';
COMMENT ON COLUMN public.tb_tip.user_id IS 'ID do usuário que escreveu a dica. Chave Estrangeira (FK) conceitual para tb_user。';
COMMENT ON COLUMN public.tb_tip.business_id IS 'ID do negócio que recebeu a dica. Chave Estrangeira (FK) conceitual para tb_business。';
COMMENT ON COLUMN public.tb_tip.text IS 'O texto da dica。';
COMMENT ON COLUMN public.tb_tip.date IS 'Data e hora em que a dica foi escrita。';
COMMENT ON COLUMN public.tb_tip.compliment_count IS 'Número de elogios que esta dica recebeu。';

```

Tabela review

```
CREATE TABLE tb_review (
    review_id          VARCHAR(255) PRIMARY KEY,
    user_id            VARCHAR(255),
    business_id        VARCHAR(255) REFERENCES tb_business(business_id),
    stars              FLOAT,
    useful             INTEGER,
    funny              INTEGER,
    cool               INTEGER,
    text               TEXT,
    date               TIMESTAMP);
```

Comentário da Tabela

COMMENT ON TABLE public.tb_review IS 'Tabela de reviews (avaliações). Tabela "fato" que liga usuários e negócios.
Criada por Alan Ferrari em 21/10/2025.';

Comentários das Colunas

```
COMMENT ON COLUMN public.tb_review.review_id IS 'ID único do review. Chave Primária (PK) da tabela.';
COMMENT ON COLUMN public.tb_review.user_id IS 'ID do usuário que escreveu o review. Chave Estrangeira (FK)
conceitual para tb_user.';
COMMENT ON COLUMN public.tb_review.business_id IS 'ID do negócio que recebeu o review. Chave Estrangeira (FK)
conceitual para tb_business.';
COMMENT ON COLUMN public.tb_review.stars IS 'Nota (estrelas) dada no review, de 1 a 5.';
COMMENT ON COLUMN public.tb_review.useful IS 'Número de votos "Útil" que este review recebeu.';
COMMENT ON COLUMN public.tb_review.funny IS 'Número de votos "Engraçado" que este review recebeu.';
COMMENT ON COLUMN public.tb_review.cool IS 'Número de votos "Legal" que este review recebeu.';
COMMENT ON COLUMN public.tb_review.text IS 'O texto completo do review.';
COMMENT ON COLUMN public.tb_review.date IS 'Data e hora em que o review foi escrito.';
```

7. Diagrama ER

O diagrama foi elaborado utilizando o dbdiagram.io

Codigo

```
Table tb_user {
    user_id varchar [pk]
    name text
    review_count integer
    yelping_since timestamp
    useful integer
    funny integer
    cool integer
    elite text
    friends text
    fans integer
    average_stars float
    compliment_hot integer
    compliment_more integer
    compliment_profile integer
    compliment_cute integer
    compliment_list integer
    compliment_note integer
    compliment_plain integer
    compliment_cool integer
    compliment_funny integer
    compliment_writer integer
```

```
compliment_photos integer}
```

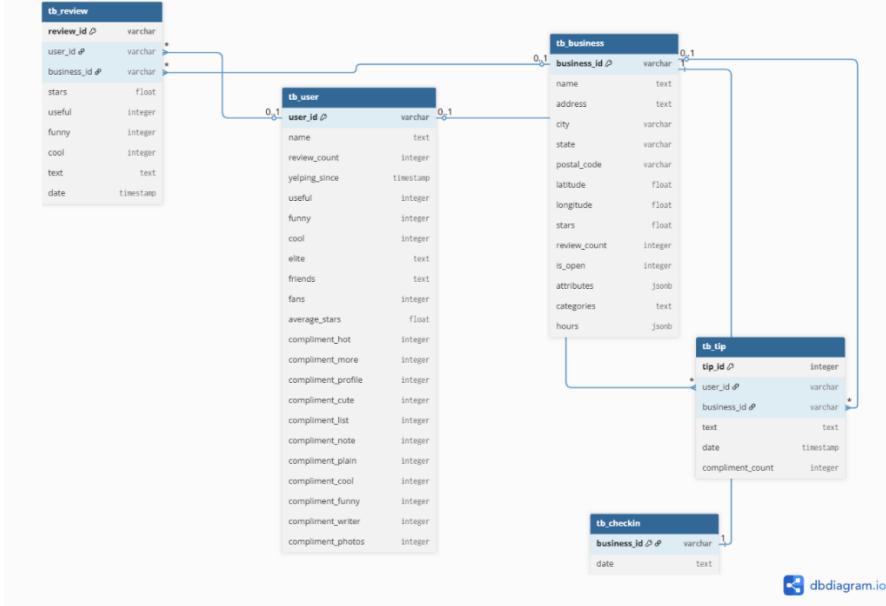
```
Table tb_business {  
    business_id varchar [pk]  
    name text  
    address text  
    city varchar  
    state varchar  
    postal_code varchar  
    latitude float  
    longitude float  
    stars float  
    review_count integer  
    is_open integer  
    attributes jsonb  
    categories text  
    hours jsonb}
```

```
Table tb_review {  
    review_id varchar [pk] // Chave Primária  
    user_id varchar [ref: > tb_user.user_id]  
    business_id varchar [ref: > tb_business.business_id]  
    stars float  
    useful integer  
    funny integer  
    cool integer  
    text text  
    date timestamp}
```

```
Table tb_tip {  
    tip_id integer [pk, increment] // Chave Primária Auto-Incremental  
    user_id varchar [ref: > tb_user.user_id]  
    business_id varchar [ref: > tb_business.business_id]  
    text text  
    date timestamp  
    compliment_count integer}
```

```
Table tb_checkin {  
    business_id varchar [pk, ref: - tb_business.business_id]  
    date text // String longa com todas as datas}
```

Diagrama



8. A carga de dados (Jupyter Notebook)

```

import pandas as pd
from sqlalchemy import create_engine, text
from sqlalchemy.dialects.postgresql import JSONB
import time
import os

# --- LOAD FUNCTION (Simplified: Append-Only) ---
def load_json_to_postgres(json_file_path, table_name, engine, chunk_size=50000):
    """
    Loads a line-by-line JSON file into a Postgres table.

    This function APPENDS data. It does NOT truncate the table.
    It assumes the table already exists.

    Args:
        json_file_path (str): The full path to the .json file.
        table_name (str): The destination table name.
        engine (sqlalchemy.engine): The SQLAlchemy connection engine.
        chunk_size (int): Number of lines to read per chunk.
    """

    print("-" * 50)
    print(f"Starting load for table: '{table_name}'")

    # --- Check if file exists (a simple, essential check) ---
    if not os.path.exists(json_file_path):
        print(f" ERROR: File not found at: {json_file_path}")
        print(f" Skipping load for table '{table_name}'")
        print("-" * 50)
        return
    
```

```

start_time = time.time()
total_rows = 0

# Define special dtypes for 'tb_business'
special_dtypes = {}
if table_name == 'tb_business':
    special_dtypes = {'attributes': JSONB, 'hours': JSONB}

try:
    print(f" ... Reading file {json_file_path}...")

    # Loop through the file in chunks
    for chunk in pd.read_json(json_file_path, lines=True, chunksize=chunk_size):

        # Load the chunk into the SQL table
        chunk.to_sql(
            table_name,
            con=engine,
            if_exists='append', # <-- Key logic: always append
            index=False,
            dtype=special_dtypes
        )

        total_rows += len(chunk)
        print(f" ... {total_rows} rows loaded into '{table_name}'...")

    end_time = time.time()
    print(f"Load for table '{table_name}' successful.")
    print(f"Total {total_rows} rows inserted.")
    print(f"Load time: {end_time - start_time:.2f} seconds.")

except Exception as e:
    # Basic error handling for the load process
    print(f"ERROR during load for table '{table_name}': {e}")

print("-" * 50)

# --- 1. GLOBAL SETTINGS ---
DB_PASSWORD = "satriani"
BASE_DATA_FOLDER = "D:/Yelp-JSON"

# --- 2. LOAD MAP (Define files and tables) ---
# This dictionary maps the source JSON filename to the destination table name.
files_to_load = {
    "yelp_academic_dataset_user.json": "tb_user",
    "yelp_academic_dataset_business.json": "tb_business",
    "yelp_academic_dataset_checkin.json": "tb_checkin",
    "yelp_academic_dataset_tip.json": "tb_tip",
    "yelp_academic_dataset_review.json": "tb_review",
}

# --- 3. DATABASE CONNECTION ---
# This script assumes the database (yelp_db) and tables ALREADY EXIST.
db_url = f'postgresql://postgres:{DB_PASSWORD}@localhost:5432/yelp_db'
engine = None

try:
    engine = create_engine(db_url)
    # Test the connection

```

```

with engine.connect() as connection:
    print("Database connection successful!")

except Exception as e:
    print(f"CRITICAL ERROR: Could not connect to database: {e}")
    print("Check DB_PASSWORD or if the Postgres server is running.")

# --- 4. MAIN LOAD LOOP ---
if engine:
    print("\nStarting batch load process (APPEND-ONLY)...")
    print("NOTE: Run TRUNCATE in DBeaver first if you want a fresh load.")
    overall_start_time = time.time()

    # Loop through the dictionary and load each file
    for file_name, table_name in files_to_load.items():

        full_file_path = os.path.join(BASE_DATA_FOLDER, file_name)

        # Call the simplified load function
        load_json_to_postgres(full_file_path, table_name, engine)

    overall_end_time = time.time()
    print("\nBatch load process finished.")
    print(f"Total operation time: {overall_end_time - overall_start_time:.2f} seconds.")

```

9. Exploração Inicial de Dados

Contagem

```

select 'tb_business' table, count (1) from public.tb_business
union
select 'tb_checkin' table, count (1) from public.tb_checkin
union
select 'tb_review' table, count (1) from public.tb_review
union
select 'tb_tip' table, count (1) from public.tb_tip
union
select 'tb_user' table, count (1) from public.tb_user

```

Resultados 1 ×

Grade	AZ table	123 count
1	tb_business	150.346
2	tb_checkin	131.930
3	tb_review	6.990.280
4	tb_tip	908.915
5	tb_user	1.987.897

```


- 1 select 'tb_business' table, count (1) from public.tb_business
- 2 union
- 3 select 'tb_checkin' table, count (1) from public.tb_checkin
- 4 union
- 5 select 'tb_review' table, count (1) from public.tb_review
- 6 union
- 7 select 'tb_tip' table, count (1) from public.tb_tip
- 8 union
- 9 select 'tb_user' table, count (1) from public.tb_user

```

Empresas categoria Café

```

select
    case when categories ilike '%coffee%' or categories ilike 'cafe' then 'caffee'
         else 'outros' end as business_categories,
    count(1) count_business
from public.tb_business
group by case when categories ilike '%coffee%' or categories ilike 'cafe' then 'caffee'
              else 'outros' end

```

```

select
  case
    when categories ilike '%coffee%' or categories ilike 'cafe' then 'coffee'
    else 'outros' end as business_categories,
  count(1) count_business
from public.tb_business
group by case when categories ilike '%coffee%' or categories ilike 'cafe' then 'coffee' else 'outros' end

```

Review de empresas categoria Café

```

select case
      when b.categories ilike '%coffee%' or b.categories ilike 'cafe' then 'coffee'
      else 'outros' end as business_categories,
      count(1) as count_review
  from public.tb_review a
  inner join public.tb_business b on a.business_id = b.business_id
  group by case when b.categories ilike '%coffee%' or b.categories ilike 'cafe' then 'coffee'
                else 'outros' end

```

```

select
  case
    when b.categories ilike '%coffee%' or b.categories ilike 'cafe' then 'coffee'
    else 'outros' end as business_categories,
  count(1) as count_review
  from public.tb_review a
  inner join public.tb_business b on a.business_id = b.business_id
  group by case
    when b.categories ilike '%coffee%' or b.categories ilike 'cafe' then 'coffee'
    else 'outros' end

```

Marco 2: Estatísticas descritivas

1. Adequação de índices

Índices existentes

```

CREATE UNIQUE INDEX tb_business_pkey ON public.tb_business USING btree (business_id)
CREATE UNIQUE INDEX tb_review_pkey ON public.tb_review USING btree (review_id)
CREATE UNIQUE INDEX tb_tip_pkey ON public.tb_tip USING btree (tip_id)
CREATE UNIQUE INDEX tb_user_pkey ON public.tb_user USING btree (user_id)

```

Índices para os JOINs

```

CREATE INDEX IF NOT EXISTS idx_review_business_id ON tb_review (business_id);
CREATE INDEX IF NOT EXISTS idx_review_user_id ON tb_review (user_id);
CREATE INDEX IF NOT EXISTS idx_tip_business_id ON tb_tip (business_id);
CREATE INDEX IF NOT EXISTS idx_tip_user_id ON tb_tip (user_id);

```

Índice para o filtro de Categoria (acelera o ILIKE)

```

CREATE EXTENSION IF NOT EXISTS pg_trgm;
CREATE INDEX IF NOT EXISTS idx_business_categories_gin ON tb_business USING GIN (categories gin_trgm_ops);

```

Índice para o filtro de Estrelas (Hipótese 1)

```
CREATE INDEX IF NOT EXISTS idx_review_stars ON tb_review (stars);
```

2. Consulta de Estatísticas Descritivas

A avaliação de unções descritivas clássicas como diretamente com tendência central (média, mediana, moda) ou variabilidade (valores mínimo/máximo, distribuição por ano), usando funções como MIN, MAX, COUNT e GROUP BY e geração dos quartis para dividir os dados em percentis (25%, 50%, 75%, 100%) permite ter uma visão inicial clara da base de dados além de permitir a elaboração de alguns insights

```
WITH base AS (SELECT b.stars AS business_stars,
                     b.review_count AS business_review_count,r.stars AS review_stars,u.fans
                from tb_business AS b
               LEFT JOIN tb_review AS r ON b.business_id = r.business_id
               LEFT JOIN tb_user AS u ON u.user_id = r.user_id
              WHERE b.categories ILIKE '%Coffee%' OR b.categories ILIKE '%Cafes%')
SELECT
  '1. Review Scores (review_stars)' AS metric,
  COUNT(review_stars) AS total_count,
  AVG(review_stars) AS average,
  percentile_cont(0.5) WITHIN GROUP (ORDER BY review_stars) AS median,
  mode() WITHIN GROUP (ORDER BY review_stars) AS mode,
  MIN(review_stars) AS minimum,
  MAX(review_stars) AS maximum,
  percentile_cont(0.25) WITHIN GROUP (ORDER BY review_stars) AS quartile_25,
  percentile_cont(0.75) WITHIN GROUP (ORDER BY review_stars) AS quartile_75
FROM base
UNION ALL
SELECT
  '2. Business Popularity (business_review_count)' AS metric,
  COUNT(DISTINCT business_review_count) AS total_count, -- Nota: Contagem de lojas
  AVG(business_review_count) AS average,
  percentile_cont(0.5) WITHIN GROUP (ORDER BY business_review_count) AS median,
  mode() WITHIN GROUP (ORDER BY business_review_count) AS mode,
  MIN(business_review_count) AS minimum,
  MAX(business_review_count) AS maximum,
  percentile_cont(0.25) WITHIN GROUP (ORDER BY business_review_count) AS quartile_25,
  percentile_cont(0.75) WITHIN GROUP (ORDER BY business_review_count) AS quartile_75
FROM base
WHERE business_review_count IS NOT NULL
UNION ALL
SELECT
  '3. User Popularity (user_fans)' AS metric,
  COUNT(fans) AS total_count,
  AVG(fans) AS average,
  percentile_cont(0.5) WITHIN GROUP (ORDER BY fans) AS median,
  mode() WITHIN GROUP (ORDER BY fans) AS mode,
  MIN(fans) AS minimum,
  MAX(fans) AS maximum,
  percentile_cont(0.25) WITHIN GROUP (ORDER BY fans) AS quartile_25,
  percentile_cont(0.75) WITHIN GROUP (ORDER BY fans) AS quartile_75
FROM base
WHERE fans IS NOT NULL;
```

```

WITH base AS (
    SELECT b.stars AS business_stars, b.review_count AS business_review_count, r.stars AS review_stars, u.fans
    FROM tb_business AS b
    LEFT JOIN tb_review AS r ON b.business_id = r.business_id
    LEFT JOIN tb_user AS u ON r.user_id = u.user_id
    WHERE b.categories ILIKE '%Coffee%' OR b.categories ILIKE '%Cafes%'
)
SELECT
    '1. Review Scores (review_stars)' AS metric,
    COUNT(review_stars) AS total_count,
    AVG(review_stars) AS average,
    percentile_cont(0.5) WITHIN GROUP (ORDER BY review_stars) AS median,
    mode() WITHIN GROUP (ORDER BY review_stars) AS mode,
    MIN(review_stars) AS minimum,
    MAX(review_stars) AS maximum,
    percentile_cont(0.25) WITHIN GROUP (ORDER BY review_stars) AS quartile_25,
    percentile_cont(0.75) WITHIN GROUP (ORDER BY review_stars) AS quartile_75
FROM base
UNION ALL

```

Grade	AZ metric	123 total_count	123 average	123 median	123 mode	123 minimum	123 maximum	123 quartile_25	123 quartile_75
1	1. Review Scores (review_stars)	627.482	3,9345240182	4	5	1	5	3	5
2	2. Business Popularity (business_review_count)	551	470,4752184126	193	5,721	5	5,721	72	473
3	3. User Popularity (user_fans)	627.481	16,2945953742	1	0	0	12.497	0	6

3. Consulta de Estatísticas Descritivas – Pontos Chaves

a. Em notas dos Reviews (Linha 1: Review Scores)

Foi avaliado um total de 627.482 reviews de cafeterias. Observa-se que a média é de 3.93 estrelas. Isso é um pouco acima da média esperada de (3.5), o que é bom para a categoria. Já a mediana é de 4 estrelas. Metade das notas são 4 ou 5, a outra metade são 4 ou menos. Isso confirma a tendência positiva. A moda é de 5 estrelas! A nota mais comum dada a uma cafeteria é a nota máxima. Isso é um sinal muito positivo para o setor. Mínimo/Máximo: 1 e 5 estrelas (como esperado).

Quanto aos quartis temos Q1 (quartile_25) de 3 estrelas. 25% das notas são 1, 2 ou 3 e Q3 (quartile_75): 5 estrelas. 75% das notas são 5 ou menos. Isso significa que os 25% superiores das notas são todas 5 estrelas.

Insight: A grande maioria das avaliações de cafeterias são positivas (mediana 4, moda 5), mas há uma cauda significativa de notas baixas (25% são 3 estrelas ou menos). Isso sugere que, embora a maioria das pessoas goste, quando uma cafeteria apresenta problemas, o cliente não demonstra muita insatisfação.

b. Popularidade da Loja (Linha 2: Business Popularity)

Um ponto de destaque aqui é observar uma diferença enorme entre a Média (470) e a Mediana (193) pois confirma o que o mercado de cafeterias é altamente assimétrico. A maioria das lojas tem uma popularidade "normal" (metade tem menos de 193 reviews), mas algumas poucas "estrelas" (provavelmente grandes redes ou locais muito famosos) têm milhares de reviews e puxam a média lá para cima.

Insight: O cliente "CoffeeKing" provavelmente começará no grupo abaixo da mediana.

c. Popularidade do Usuário (Linha 3: User Popularity)

Foi avaliado um total 627.481 usuários. A média foi de 16.3 fãs por usuário e a mediana de 1 fã por usuário. Já a moda foi de 0 fãs. O número mais comum de fãs para um usuário que avalia cafeterias é zero.

Insight: Assim como a popularidade das lojas, a popularidade dos usuários também é extremamente assimétrica. A grande maioria dos usuários que frequentam cafeterias não são "influencers" (mediana 1, moda 0). No entanto, existe uma pequena fração de usuários com

milhares de fãs. Isso pode ser importante para a CoffeeKing: talvez seja mais valioso agradar um desses "super usuários" do que 100 usuários comuns.

4. Teste de Hipóteses

a. Hipótese 1: Sentimento (Atendimento/Ambiente vs. Preço)

Avaliar hipótese: "O 'ambiente' e o 'atendimento' serão fatores decisivos mais fortes para avaliações de 5 estrelas do que o 'preço'."

```
SELECT
    r.stars,
    COUNT(*) AS total_reviews,
    (COUNT(CASE WHEN r.text ILIKE '%service%' OR r.text ILIKE '%staff%' OR r.text ILIKE '%friendly%' OR r.text ILIKE '%rude%' THEN 1 END) * 100.0 / COUNT(*)) AS pct_mentions_service,
    (COUNT(CASE WHEN r.text ILIKE '%atmosphere%' OR r.text ILIKE '%ambiance%' OR r.text ILIKE '%vibe%' THEN 1 END) * 100.0 / COUNT(*)) AS pct_mentions_atmosphere,
    (COUNT(CASE WHEN r.text ILIKE '%flavor%' OR r.text ILIKE '%taste%' THEN 1 END) * 100.0 / COUNT(*)) AS pct_mentions_flavor,
    (COUNT(CASE WHEN r.text ILIKE '%price%' OR r.text ILIKE '%expensive%' OR r.text ILIKE '%cheap%' THEN 1 END) * 100.0 / COUNT(*)) AS pct_mentions_price
FROM tb_review AS r
JOIN tb_business AS b ON r.business_id = b.business_id
WHERE (b.categories ILIKE '%Coffee%' OR b.categories ILIKE '%Cafes%') AND (r.stars = 1 OR r.stars = 5)
GROUP BY r.stars
ORDER BY r.stars;
```

r.stars	123 total_reviews	123 pct_mentions_service	123 pct_mentions_atmosphere	123 pct_mentions_flavor	123 pct_mentions_price
1	67.185	46,1	2,45	14,02	10,64
5	310.082	44,65	13,47	17,01	10,71

O atendimento (service) é o GRANDE vilão: Quase metade (46.10%) dos reviews de 1 estrela mencionam algo sobre o serviço (atendimento, staff, rudeza, etc.). Isso é muito mais alto que menções a "preço" (10.64%), "sabor" (14.02%) ou "ambiente" (2.45%).

Um mau atendimento é, de longe, o principal motivo para alguém dar a pior nota a uma cafeteria. Mas o atendimento (service) continua importante: Surpreendentemente, o atendimento também é mencionado em 44.65% dos reviews de 5 estrelas. Isso sugere que um bom atendimento é quase tão crucial para o sucesso quanto um mau atendimento é para o fracasso.

O ambiente (atmosphere) é relevante: O ambiente é mencionado em 13.47% das avaliações de 5 estrelas. Isso é mais do que o preço (10.71%), mas significativamente menor do que o atendimento (44.65%) e até um pouco menos que o sabor (17.01%). O preço (price) é secundário: O preço é mencionado de forma quase igual em reviews de 1 estrela (10.64%) e 5 estrelas (10.71%), indicando que ele não é o fator decisivo para notas extremas.

Conclusão:

A hipótese estava PARCIALMENTE CORRETA, pois o "preço" não é o fator decisivo para notas de 5 estrelas, pois tanto o "atendimento" quanto o "ambiente" (e até o "sabor") são mencionados com mais frequência. Quanto a hipótese de "ambiente" e "atendimento" seriam os mais fortes, os dados mostram que o "atendimento" é, de longe, o fator mais mencionado tanto nas piores quanto nas melhores avaliações. O "ambiente" é importante para notas altas, mas não tanto quanto um bom serviço.

Insight (perguntas adicionais):

Em Resumo, os números dizem para focar obsessivamente no atendimento! Um bom atendimento gera notas 5 estrelas, e um mau atendimento garante notas 1 estrela. O ambiente é importante para o sucesso, mas o atendimento é ainda mais crucial. O preço parece ser um fator menos determinante para avaliações extremas.

b. Hipótese 2: Wi-Fi vs. Volume de Reviews

A disponibilidade de 'Wi-Fi Grátis' terá uma correlação positiva com um alto volume de reviews (review_count)"

```
SELECT
    replace(replace(attributes->>'WiFi','u',""),"","") AS wifi_status,
    COUNT(business_id) AS total_de_cafeterias,
    round(AVG(review_count),2) AS media_de_reviews,
    round(AVG(stars)::decimal,2) AS media_de_estrelas
FROM tb_business
where (categories ILIKE '%Coffee%' OR categories ILIKE '%Cafes%') AND attributes->>'WiFi' IS NOT NULL
GROUP BY wifi_status
ORDER BY media_de_reviews DESC;
```

```

SELECT
    replace(replace(attributes->>'WiFi','u',''),'''') AS wifi_status,
    COUNT(business_id) AS total_de_cafeterias,
    round(AVG(review_count),2) AS media_de_reviews,
    round(AVG(stars)::decimal,2) AS media_de_estrelas
FROM tb_business
WHERE (categories ILIKE '%Coffee%' OR categories ILIKE '%Cafes%') AND attributes->>'WiFi' IS NOT NULL
GROUP BY wifi_status
ORDER BY media_de_reviews DESC;

```

The screenshot shows the DBeaver interface with a SQL query in the script editor. The results are displayed in a grid table with four columns: wifi_status, total_de_cafeterias, media_de_reviews, and media_de_estrelas. The data is as follows:

wifi_status	total_de_cafeterias	media_de_reviews	media_de_estrelas
no	854	139,18	3,94
free	6.402	72,86	3,57
paid	19	55,95	3,63
None	4	22,25	2,63

O resultado refuta a sua hipótese inicial de que “A disponibilidade de ‘Wi-Fi Grátis’ terá uma correlação positiva com um alto volume de reviews (review_count).

Os dados mostram inequivocamente que as cafeterias sem Wi-Fi (no) têm, em média, quase o dobro de reviews (139.18) em comparação com as que oferecem Wi-Fi grátis (free com 72.86), evidenciando que a hipótese estava errada, que a correlação é negativa. A diferença na nota média (media_de_estrelas) também ficou mais clara. Locais sem Wi-Fi não só têm mais reviews, mas também têm uma nota média significativamente mais alta (3.94) do que os locais com Wi-Fi grátis (3.57).

Conclusão:

Hipótese 2 foi REFUTADA. Os dados mostram o oposto do esperado: não ter Wi-Fi está associado a um volume muito maior de reviews e a notas médias mais altas nas cafeterias do dataset Yelp.

Insight (perguntas adicionais):

A descoberta principal é que a ausência de Wi-Fi parece ser um forte indicador de maior engajamento e qualidade percebida. O próximo passo é investigar o motivo disso. Trata-se de locais mais "premium" ou mais focados no produto (café)? Possuem outro diferencial forte (localização, ambiente)?

c. Hipótese 3: Usuários "Elite"

Usuários "Elite" são mais críticos (dão notas menores) e (B) mencionam mais o "sabor".

```

SELECT
    CASE WHEN u.elite IS NOT NULL AND u.elite != '' THEN 'Elite' ELSE 'Non-Elite' END AS user_type,
    COUNT(r.review_id) AS total_reviews,
    round(AVG(r.stars)::decimal,2) AS average_stars,
    round((COUNT(CASE WHEN r.text ILIKE '%flavor%' OR r.text ILIKE '%taste%' THEN 1 END) * 100.0 /
    COUNT(r.review_id)),2) AS pct_mentions_flavor,
    round((COUNT(CASE WHEN r.text ILIKE '%price%' OR r.text ILIKE '%expensive%' THEN 1 END) * 100.0 /
    COUNT(r.review_id)),2) AS pct_mentions_price
FROM tb_review AS r

```

```

JOIN tb_business AS b ON r.business_id = b.business_id
JOIN tb_user AS u ON r.user_id = u.user_id
where (b.categories ILIKE '%Coffee%' OR b.categories ILIKE '%Cafes%') -- Filtro para Cafeterias
GROUP BY user_type;

```

The screenshot shows the DBeaver interface with a SQL script open in the editor. The script performs a SELECT query to compare average star ratings and price mentions between Elite and Non-Elite users. The results are displayed in a grid table.

```

SELECT
CASE WHEN u.elite IS NOT NULL AND u.elite != '' THEN 'Elite' ELSE 'Non-Elite' END AS user_type,
COUNT(r.review_id) AS total_reviews,
round(AVG(r.stars)::decimal,2) AS average_stars,
round((COUNT(CASE WHEN r.text ILIKE '%flavor%' OR r.text ILIKE '%taste%' THEN 1 END)
* 100.0 / COUNT(r.review_id)),2) AS pct_mentions_flavor,
round((COUNT(CASE WHEN r.text ILIKE '%price%' OR r.text ILIKE '%expensive%' THEN 1 END)
* 100.0 / COUNT(r.review_id)),2) AS pct_mentions_price
FROM tb_review AS r
JOIN tb_business AS b ON r.business_id = b.business_id
JOIN tb_user AS u ON r.user_id = u.user_id
where (b.categories ILIKE '%Coffee%' OR b.categories ILIKE '%Cafes%') -- Filtro para Cafeterias
GROUP BY user_type;

```

Grade	AZ user_type	123 total_reviews	123 average_stars	123 pct_mentions_flavor	123 pct_mentions_price
1	Elite	183.939	4,08	27,22	16,23
2	Non-Elite	443.542	3,87	15,45	11,84

A hipótese era que usuários Elite seriam *mais críticos* (notas menores), mas os dados mostram o oposto. A nota média (average_stars) dada por usuários Elite (4.08) é significativamente mais alta do que a nota média dada por usuários Não-Elite (3.87).

Isso sugere que usuários Elite não são "mais críticos" no sentido de dar notas baixas. Talvez eles sejam melhores em *encontrar* bons lugares, ou talvez eles valorizem mais a qualidade (estojam dispostos a pagar por ela), resultando em experiências e notas médias mais altas.

Agora, referente ao "sabor", os dados confirmam isso de forma clara. A porcentagem de reviews de usuários Elite que mencionam "flavor" ou "taste" (pct_mentions_flavor) é de 27.22%, quase o dobro dos usuários Não-Elite (15.45%).

Embora não fosse parte da sua hipótese principal, a consulta também mostra que usuários Elite mencionam "price" ou "expensive" (pct_mentions_price) com mais frequência (16.23%) do que usuários Não-Elite (11.84%). Isso reforça a ideia de que eles estão mais atentos à relação custo-benefício ou talvez frequentem lugares onde o preço é um fator mais discutido.

Conclusão:

Hipótese 3 foi **PARCIALMENTE REFUTADA**. A parte sobre o foco no "sabor" foi comprovada, mas a parte sobre serem "mais críticos" (notas menores) foi refutada pelos dados, revelando que eles, na verdade, dão notas médias mais altas.

Insight (perguntas adicionais):

A descoberta principal é que o status "Elite" está associado a notas médias mais altas e a um foco maior na qualidade (sabor). O próximo passo pode ser investigar se essa nota mais alta

dos Elites se mantém em diferentes faixas de preço ou se eles também mencionam "service" e "atmosphere" com a mesma frequência que os não-elites. Isso ajudaria a entender se eles são realmente são especialistas ou apenas usuários mais experientes em geral.

Marco 3: Além das estatísticas descritivas

1. relacionamentos e correlação, correlação de Pearson – Aplicado na Hipótese 2: Wi-Fi vs. Volume de Reviews

A hipótese 2 tratou-se em avaliar se disponibilidade de "Wi-Fi Grátis" era um dos principais fatores de correlação positiva para cafeterias com alto volume de reviews (review_count), mesmo que a nota (stars) não seja perfeita.

O resultado acabou sendo REFUTADA uma vez que os dados mostram o oposto do esperado: não ter Wi-Fi está associado a um volume muito maior de reviews e a notas médias mais altas nas cafeterias do dataset Yelp.

```

SELECT
    replace(replace(attributes->>'WiFi','u',''),'''') AS wifi_status,
    COUNT(business_id) AS total_de_cafeterias,
    round(AVG(review_count),2) AS media_de_reviews,
    round(AVG(stars)::decimal,2) AS media_de_estrelas
FROM tb_business
where (categories ILIKE '%Coffee%' OR categories ILIKE '%Cafes%') AND attributes->>'WiFi' IS NOT NULL
GROUP BY wifi_status
ORDER BY media_de_reviews DESC;

```

Grade	wifi_status	total_de_cafeterias	media_de_reviews	media_de_estrelas
1	no	854	139,18	3,94
2	free	6.402	72,86	3,57
3	paid	19	55,95	3,63
4	None	4	22,25	2,63

Mas a questão que ficou foi: Considerando todas as cafeterias, existe uma tendência linear entre a nota média de uma cafeteria (stars) e o seu número total de reviews (reviews_count)?

Optou-se pela avaliação da correlação de Pearson, que é uma medida estatística que indica a força e a direção da relação linear entre duas variáveis quantitativas.

```

select corr(stars, review_count) AS correlation_rating_vs_volume
from tb_business
where (categories ILIKE '%Coffee%' OR categories ILIKE '%Cafes%');

```

DBeaver 25.2.3 - <yelp_db> Script-11

Arquivo Editar Navegar Procurar Editor SQL Banco de dados Janela Ajuda

SQL Commit Rollback Auto yelp_db public@yelp_db

*<yelp_db> Script-1 <yelp_db> Script-2 <postgres> Script <yelp_db> Script-2.sql *<yelp_db> Script-4 *<yelp_db> Script-5

```
select round(corr(stars, review_count)::numeric,4) AS correlation_rating_vs_volume
from tb_business
where (categories ILIKE '%Coffee%' OR categories ILIKE '%Cafes%');
```

Resultados 1

Grade	123 correlation_rating_vs_volume
1	0,148

Analizando a Correlação de Pearson entre a nota média (stars) e o volume total de reviews (review_count) para todas as cafeterias observa-se que o coeficiente foi de +0.148. Isso indica uma correlação linear positiva fraca. Embora não seja uma relação forte demonstra que cafeterias com mais reviews tendem a ter notas médias ligeiramente superiores. Isso contrasta com a descoberta anterior sobre o Wi-Fi, indicando que a relação entre popularidade e qualidade percebida pode ser complexa e depender de outros fatores.

Pra reforçar a análise, categorizamos o campo texto da review entre utilização para trabalho ou estudo e outra com problemas de WIFI. Para investigar se a nota mais baixa dos locais com Wi-Fi grátis era devido a reclamações do segmento 'trabalho/estudo', realizou-se uma análise textual com Python.

Imagen 1 - Resultado da Análise SQL: Média de Reviews e Notas por Status do Wi-Fi:

```
dtypes: float64(1), object(3)
memory usage: 18.5+ MB

[46]: import re # Import library for regular expressions (text pattern matching)
# print("Starting text analysis...")

# Check if df_reviews was loaded successfully in the previous step
if df_reviews is not None and not df_reviews.empty:

    # --- 1. Define Keywords ---
    # Keywords related to working/studying at the coffee shop
    work_study_keywords = ['work', 'study', 'laptop', 'meeting', 'homework', 'assignment', 'job', 'remote', 'zoom']

    # Keywords related to Wi-Fi problems (use variations)
    wifi_issue_keywords = [
        'wifi slow', 'slow wifi', 'internet slow', 'slow internet',
        'wifi down', 'internet down', 'no wifi', 'no internet',
        'wifi issue', 'internet issue', 'wifi problem', 'internet problem',
        'disconnect', 'connection bad', 'bad connection', 'poor connection'
    ]
```

Creating flags for work/study mentions...

```
Counts of reviews mentioning work/study:
mentions_work_study
False      51445
True       91623
Name: count, dtype: int64

Counts of reviews mentioning Wi-Fi issues:
mentions_wifi_issue
False     605294
True        774
Name: count, dtype: int64
```

Imagen 2 Reviews e Notas por Status do Wi-Fi em pivot:

```

# Compare average stars based on BOTH Wi-Fi status AND issue mention
print("\nAverage stars broken down by Wi-Fi status and issue mention:")
# Use pivot_table for a clearer view
pivot = pd.pivot_table(df_reviews,
                       values='stars',
                       index='wifi_status',
                       columns='mentions_wifi_issue',
                       aggfunc='mean')
print(pivot.round(2)) # Round to 2 decimal places

pivot2 = pd.pivot_table(df_reviews,
                        values='stars',
                        index='wifi_status',
                        columns='mentions_work_study',
                        aggfunc='mean')
print(pivot2.round(2)) # Round to 2 decimal places
else:
    print("Cannot perform analysis, 'df_reviews' or flag columns not available.")

--- Analyzing Impact of Wi-Fi Issue Mentions ---

Average stars for reviews mentioning Wi-Fi issues: 3.23
Average stars for reviews NOT mentioning Wi-Fi issues: 3.94

Average stars broken down by Wi-Fi status and issue mention:
mentions_wifi_issue  False  True
 wifi_status
free                 3.90  2.95
no                  4.11  3.71
paid                3.61  3.50
mentions_work_study  False  True
 wifi_status
free                 3.93  3.73
no                  4.13  3.90
paid                3.65  3.34

```

O que os números apontam:

Segmento "Trabalho/Estudante": Cerca de 15% (91.623 de 606.068) das reviews de cafeterias mencionam termos relacionados a usar o local para trabalhar ou estudar. Isso confirma que existe um segmento considerável de clientes com essa necessidade.

Reclamações de Wi-Fi: Apenas 0.13% (774 reviews) mencionaram explicitamente problemas com o Wi-Fi usando as palavras-chave definidas. Esse número é muito baixo.

Conclusão:

Se aplicarmos a premissa que a categoria trabalho e estudo também utilizam WI-FI, esses resultados reforçam ainda mais a conclusão de que oferecer Wi-Fi grátis não é uma garantia de satisfação e pode até estar associado a uma experiência percebida como pior, especialmente para quem usa o local para trabalhar.

A queda maior na nota média (de 3.93 para 3.73) nos locais free quando se menciona trabalho/estudo apoia sua ideia de que a experiência de trabalhar nesses locais pode ser mais problemática (talvez mais lotado, conexão instável mesmo que não mencionada como "issue", etc.), levando a críticas.

O fato de que a nota média mesmo para quem menciona trabalho/estudo ainda é maior nos locais no Wi-Fi (3.90 vs 3.73) é um forte argumento de que esses locais sem Wi-Fi oferecem uma experiência geral superior, ou atraem um público que valoriza outros aspectos mais do que a conexão.

Para a CoffeeKing: Os dados sugerem que tentar atrair o público de "trabalho/estudo" com Wi-Fi grátis pode não ser a estratégia mais eficaz para garantir notas altas. Focar na qualidade do produto, ambiente e atendimento (como sugerido pela H1) parece ser um caminho mais seguro, mesmo que isso signifique não oferecer Wi-Fi. A experiência de trabalho nos locais free parece ser, em média, pior avaliada do que a experiência geral nesses mesmos locais.

2. Regressão linear para previsão futura (se a relação for linear)

Embora a correlação de Pearson seja fraca (+0.148), não esperamos que este modelo seja muito preciso, mas será utilizado pra efeito de exercício. Usaremos Python com a biblioteca scikit-learn para o exercício.

Os dados (stars e review_count) de 8.480 cafeterias foram preparados corretamente para o modelo com o modelo de regressão linear foi treinado.

```
print("Linear Regression Model Trained Successfully!")
print(f"Equation: stars = {slope:.4f} * review_count + {intercept:.4f}")

# R-squared: How much of the variation in 'stars' is explained by 'review_count'
# Score closer to 1 is better, closer to 0 is worse.
r_squared = model.score(X, y)
print(f'R-squared: {r_squared:.4f}')

# Interpretation based on R-squared (related to correlation)
print(f"Interpretation: review_count explains only {r_squared*100:.2f}% of the variance in stars.")
if r_squared < 0.1:
    print("This indicates a very weak linear relationship, as expected from the low correlation.")

else:
    print("Cannot train model, data (X or y) is not available.")
model = None # Ensure model is None if training failed

--- Step 4: Training the Linear Regression Model ---
Linear Regression Model Trained Successfully!
Equation: stars = 0.000840 * review_count + 3.5364
R-squared: 0.0219
Interpretation: review_count explains only 2.19% of the variance in stars.
This indicates a very weak linear relationship, as expected from the low correlation.
```

Equação da Reta: $\text{stars} = 0.000840 * \text{review_count} + 3.5364$

Inclinação (slope = 0.000840): Este número é positivo, mas minúsculo que, de acordo com o modelo linear, para cada 100 reviews a mais que uma cafeteria recebe, a sua nota média (stars) aumenta apenas 0.084 estrelas.

Intercepto (intercept = 3.5364): O modelo prevê que uma cafeteria hipotética com zero reviews teria uma nota inicial de aproximadamente 3.54 estrelas.

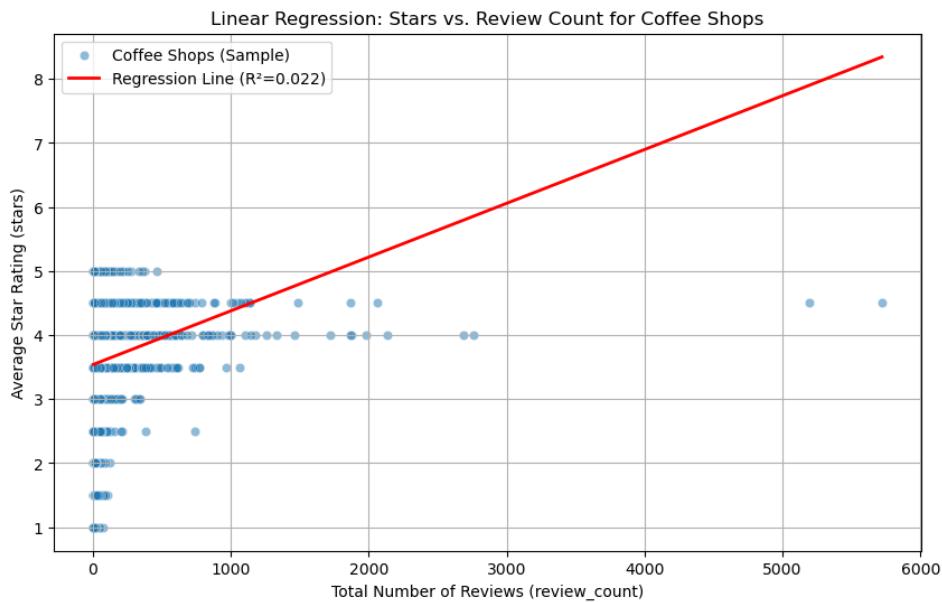
R^2 (R-quadrado): 0.0219

Este é o valor mais importante aqui. R^2 mede quão bem a linha de regressão se ajusta aos dados, ou seja, qual a porcentagem da variação na nota (stars) que pode ser explicada linearmente pelo número de reviews (review_count).

Interpretação: Um R^2 de 0.0219 significa que o review_count explica apenas 2.19% da variação nas notas das cafeterias. Isso é extremamente baixos.

Nuvem de Pontos: A maioria das cafeterias (pontos azuis) está concentrada na parte esquerda do gráfico (poucos reviews) e entre 3.0 e 5.0 estrelas. Há alguns outliers com muitos reviews.

--- Step 5: Visualizing the Regression ---



Linha de Regressão (Vermelha): A linha é quase horizontal. Isso confirma visualmente a inclinação (slope) minúscula que encontramos. Ela sobe muito, muito pouco da esquerda para a direita.

R^2 no Gráfico: A legenda da linha vermelha mostra " $R^2=0.022$ ", confirmando o valor baixo.

O gráfico de dispersão confirma visualmente a falta de uma forte tendência linear. A relação entre a popularidade (contagem de reviews) e a qualidade percebida (nota média) nas cafeterias é muito mais complexa do que uma simples linha reta pode descrever. Embora haja uma tendência quase insignificante de locais mais populares serem ligeiramente melhores avaliados, outros fatores (atendimento, ambiente, qualidade do produto, preço, localização, tipo de cliente - como vimos nas outras análises!) são muito mais importantes para determinar a nota final. Este modelo linear não seria útil para fazer previsões futuras sobre a nota de uma cafeteria com base apenas em quantos reviews ela tem.

3. Análise textual para TF-IDF – Aplicado na Hipótese 1: Sentimento (Atendimento/Ambiente vs. Preço)

Para a análise textual optou-se em avaliar as palavras chaves para as piores classificações. No Dbeaver transformou-se o texto em palavras num array, ignorando algumas palavras e considerando apenas palavras com mais de 4 caracteres e selecionando apenas Stars =1.

```

SELECT TRIM(LOWER(word)) AS term, COUNT(*) AS frequency
FROM tb_review AS r
JOIN tb_business AS b ON r.business_id = b.business_id,
unnest(string_to_array(r.text, ' ')) AS word
WHERE (b.categories ILIKE '%Coffee%' OR b.categories ILIKE '%Cafes%')
AND r.stars = 1 AND LENGTH(word) > 4
AND LOWER(word) NOT IN ('this', 'that', 'with', 'they', 'have', 'from', 'your', 'about', 'just',
'like', 'there', 'what', 'when', 'which', 'will', 'also', 'here', 'more', 'some', 'other', 'were',
'veen', 'their', 'only', 'would', 'because', 'coffee', 'place', 'time', 'order', 'food', 'get',
'one', 'wasnt', 'dont')
GROUP BY term
ORDER BY frequency DESC
LIMIT 10;

```

Resultados 1 ×

SELECT TRIM(LOWER(word)) AS term, COUNT(*) AS frequency FROM tb_review AS r JOIN:

Grade	term	frequency
1	never	17.897
2	don't	17.142
3	service	16.851
4	after	16.736
5	ordered	16.653
6	asked	15.619
7	minutes	15.266
8	didn't	14.058
9	people	11.902
10	could	11.612

A palavra service é a terceira mais frequente, confirmando a importância crucial que vimos no Marco 2.

A palavra asked (pediu/perguntou) sugere problemas na comunicação ou no atendimento a pedidos.

A alta frequência de after (depois) e minutes (minutos) aponta fortemente para problemas relacionados a demoras e longos tempos de espera, seja para pedir, receber o pedido ou pagar.

A palavra ordered é muito frequente, indicando que muitos problemas ocorrem no processo de fazer ou receber o pedido (pedido errado, demora no pedido, item em falta).

A palavra never é a palavra mais comum, frequentemente usada em frases como "never coming back", indicando uma péssima experiência final.

Conclusão:

Analizando os termos mais frequentes nos reviews de 1 estrela para cafeterias, os 10 termos mais comuns (excluindo stop words) foram: never, don't, service, after, ordered, asked, minutes, didn't, people, could. Temas Identificados: Estes termos apontam para três temas principais de insatisfação:

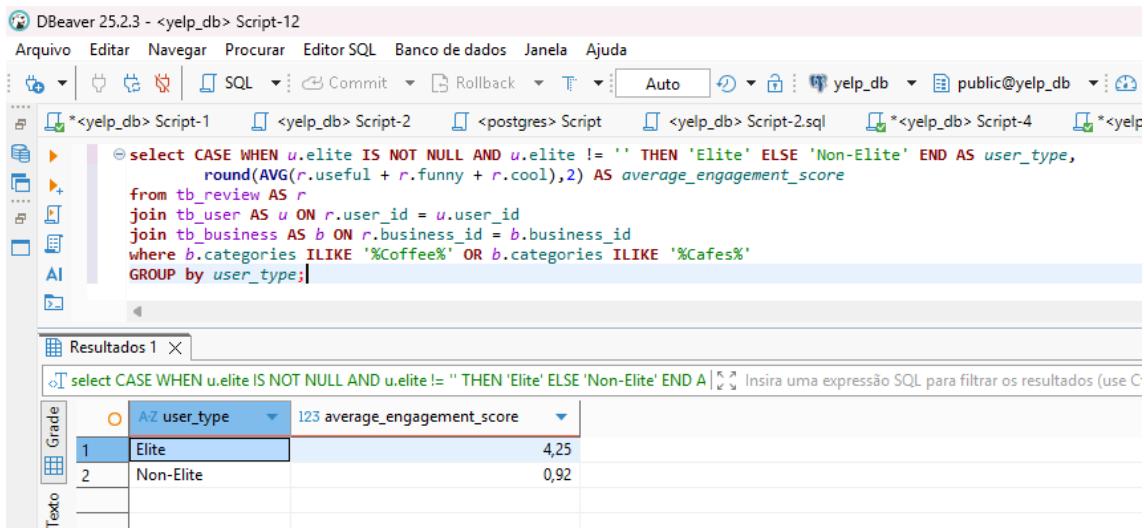
- Mau Atendimento/Interação: Problemas com a equipa e comunicação (service, asked).
- Tempo de Espera Excessivo: Demora no serviço (after, minutes).
- Erros/Problemas nos Pedidos: Falhas no que foi pedido ou recebido (ordered). A palavra never sugere que estas falhas frequentemente levam à perda definitiva do cliente.

Os dados mostram que falhas operacionais básicas (atendimento, tempo, precisão do pedido) são os principais motores das piores avaliações. O treino da equipa e a eficiência dos processos são cruciais.

4. Nova métrica: Engajamento

Foi elaborada a métrica para avaliar a "Pontuação Média de Engajamento". É definida como a média da soma dos votos useful, funny e cool recebidos por review ($\text{AVG}(r.\text{useful} + r.\text{funny} + r.\text{cool})$), agrupada por tipo de usuário (Elite vs. Não-Elite).

No Marco 2, descobrimos que usuários Elite dão notas médias mais altas e focam mais no "sabor", então esta métrica foi criada para testar se os reviews deles também são percebidos como mais valiosos ou influentes pela comunidade Yelp (medido pelo número de votos que recebem).



The screenshot shows the DBeaver interface with a SQL script editor and a results grid.

```
DBBeaver 25.2.3 - <yelp_db> Script-12
Arquivo Editar Navegar Procurar Editor SQL Banco de dados Janela Ajuda
SQL Commit Rollback Auto yelp_db public@yelp_db
*<yelp_db> Script-1 <yelp_db> Script-2 <postgres> Script <yelp_db> Script-2.sql *<yelp_db> Script-4 *<yelp_db> Script-5

select CASE WHEN u.elite IS NOT NULL AND u.elite != '' THEN 'Elite' ELSE 'Non-Elite' END AS user_type,
       round(AVG(r.useful + r.funny + r.cool),2) AS average_engagement_score
  from tb_review AS r
  join tb_user AS u ON r.user_id = u.user_id
  join tb_business AS b ON r.business_id = b.business_id
 where b.categories ILIKE '%Coffee%' OR b.categories ILIKE '%Cafes%'
 GROUP by user_type;
```

Resultados 1

user_type	average_engagement_score
Elite	4,25
Non-Elite	0,92

Resultados: Os resultados foram conclusivos:

- Usuários Elite: Pontuação Média de Engajamento = 4.25
- Usuários Não-Elite: Pontuação Média de Engajamento = 0.92

Os reviews escritos por usuários Elite para cafeterias recebem, em média, mais de quatro vezes mais engajamento (votos) do que os reviews de usuários não-Elite. Isso sugere fortemente que o feedback do grupo Elite é considerado mais valioso ou, pelo menos, atrai mais atenção da comunidade.

Insight:

Embora os usuários Elite sejam uma minoria, seus reviews têm um alcance e impacto potencialmente muito maiores. Prestar atenção especial ao feedback deles e garantir uma experiência positiva para este grupo pode ter um retorno desproporcional em termos de reputação online.

Assunto: Análise da Concorrência: 3 Recomendações Estratégicas para o Sucesso da CoffeeKing

Para: Diretoria de Marketing e Operações da CoffeeKing **De:** Alan Ferrari, Analista de Dados

Data: 21 de Outubro de 2025

1. Resumo Executivo: O Caminho para as 5 Estrelas

Esta análise investigou o dataset do Yelp (incluindo 627.482 reviews relevantes) para identificar os fatores que impulsionam o sucesso e o fracasso de cafeterias concorrentes. O objetivo é fornecer à CoffeeKing uma estratégia de lançamento baseada em dados.

Principal Insight: O sucesso no mercado de cafeterias não é determinado pelo preço, nem pela oferta de Wi-Fi. O fator decisivo para o sucesso (reviews de 5 estrelas) ou o fracasso (reviews de 1 estrela) é, de forma esmagadora, a **qualidade do atendimento**.

The Decisive Factor (Service vs. Price)

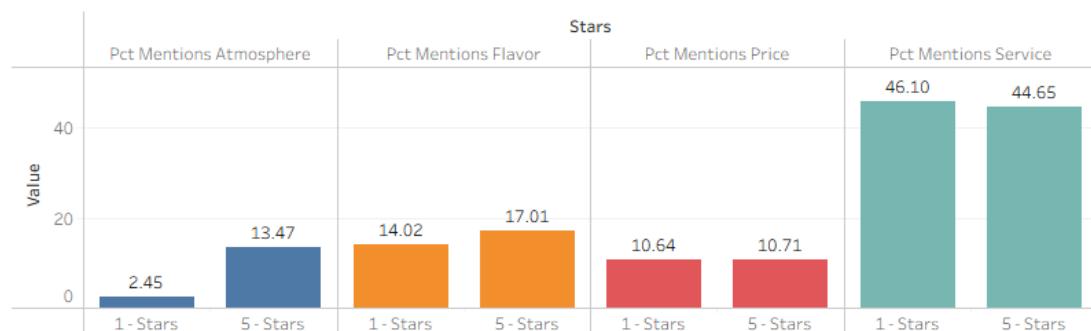
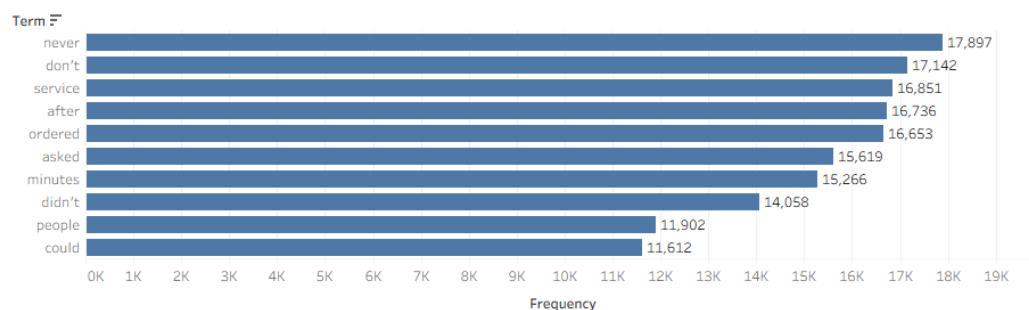


Gráfico 1 – Serviço x Preço

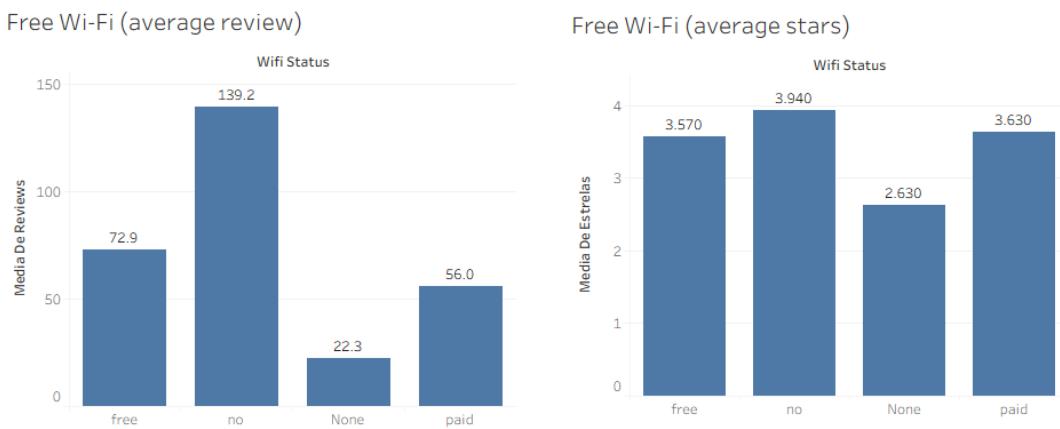
Recomendações-Chave:

- Operações:** Focar obsessivamente no treinamento de atendimento e na eficiência dos pedidos. O maior risco operacional não é o produto, mas um serviço lento ou rude.

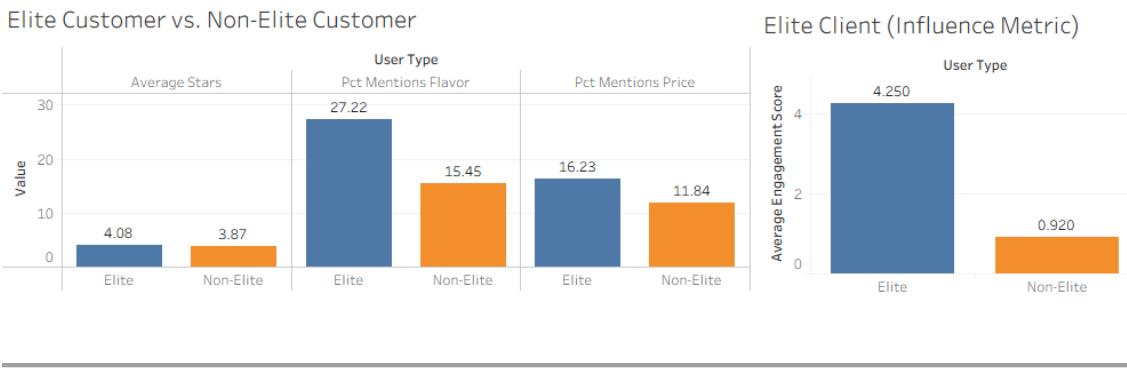
Poor Service (Meaning)



- Marketing:** Focar a mensagem na *qualidade do produto (sabor)* e no *ambiente*. A estratégia de "Wi-Fi grátis para trabalhar" não se mostrou eficaz, estando associada a notas médias mais baixas.



3. Estratégia: Priorizar a experiência do cliente "Elite", pois seus reviews, embora em menor número, são **4.25 vezes mais influentes** e focados na qualidade.



2. As Descobertas: O que Realmente Importa para os Clientes?

Nossa análise testou três hipóteses centrais sobre o que impulsiona as avaliações dos clientes. Os resultados foram claros e, em alguns casos diferentes do esperado.

Descoberta 1: Atendimento é o Fator Decisivo (Hipótese 1 Validada)

Nossa hipótese era que "atendimento" e "ambiente" seriam mais importantes que "preço". Os dados confirmam isso categoricamente.

- **O Preço** é quase irrelevante para notas extremas (menionado em ~10.7% tanto nas notas 1 quanto nas 5 estrelas).
- **O Atendimento (Service)** é o fator mais polarizante:
 - **Causa do Fracasso:** 46,1% dos reviews de 1 estrela mencionam o atendimento (mau serviço, grosseria, equipe).
 - **Causa do Sucesso:** 44,6% dos reviews de 5 estrelas elogiam o atendimento.
- **O Ambiente** é um fator secundário de sucesso, sendo 5.5x mais mencionado em reviews de 5 estrelas (13,47%) do que nos de 1 estrela (2,45%).

Em suma: Um mau atendimento garante uma nota 1. Um ótimo atendimento é o principal caminho para uma nota 5.

Descoberta 2: O Mito do Wi-Fi Grátis (Hipótese 2 Refutada)

Nossa hipótese era que "Wi-Fi Grátis" teria uma correlação positiva com um alto volume de reviews (popularidade). Os dados mostram exatamente o oposto.

- **Refutação:** Locais **sem Wi-Fi (no)** têm, em média, quase o **dobro de reviews (139,18)** em comparação com locais com Wi-Fi grátis (72,86).
- **Qualidade:** Locais sem Wi-Fi também têm **notas médias significativamente mais altas (3,94)** do que locais com Wi-Fi grátis (3,57).
- **Por quê?** Investigamos mais a fundo. Cerca de 15% dos clientes usam cafeterias para "trabalhar/estudar". Esse segmento, atraído pelo Wi-Fi, tende a dar notas piores (3,73) do que a média geral dos locais com Wi-Fi (3,93).

Em suma: Oferecer Wi-Fi grátis não é uma garantia de popularidade e pode atrair um público mais crítico que derruba a nota média. Locais focados na experiência (sem Wi-Fi) performam melhor.

Descoberta 3: O Valor do Cliente "Elite" (Hipótese 3 e Nova Métrica)

Nossa hipótese era que usuários "Elite" seriam mais críticos (dando notas menores) e focariam mais no "sabor".

- **Resultado (Crítica): Refutada.** Usuários Elite, na verdade, dão **notas médias mais altas (4,08)** do que usuários Não-Elite (3,87).
- **Resultado (Sabor): Validada.** Usuários Elite focam quase o **dobro na qualidade e sabor do café (27,22%)** em comparação com Não-Elites (15,45%).

Isso sugere que Elites não são "críticos", mas sim exigentes: eles buscam (e recompensam) qualidade.

- **Nova Métrica (Engajamento):** Para medir a influência desse grupo, criamos a "Pontuação Média de Engajamento" (soma de votos "useful", "funny", "cool" por review).
- **Resultado:** Reviews de usuários Elite são **4,25 vezes mais influentes** (pontuação de 4,25) que os de Não-Elites (pontuação de 0,92).

Em suma: O cliente Elite é o *influenciador* do ecossistema. Conquistá-los com "sabor" e "qualidade" gera avaliações de alto impacto e alta visibilidade.

3. Recomendações Estratégicas e Próximos Passos

Com base nessas três descobertas, recomendamos as seguintes ações para o lançamento da CoffeeKing:

Para a Diretoria de Operações:

O maior risco da CoffeeKing não é o preço ou a falta de Wi-Fi; é um atendimento ruim.

- **Ação Imediata:** Implementar treinamento rigoroso focado em agilidade, comunicação e precisão.
- **Justificativa:** Nossa análise de termos (TF-IDF) das piores avaliações (1 estrela) mostrou que os clientes reclamam de "demora" (*termos minutes, after*) e "pedidos errados" (*ordered, asked*). Eficiência operacional é crucial.

Para a Diretoria de Marketing:

A mensagem deve focar na experiência, não na funcionalidade (Wi-Fi).

- **Ação Imediata:** Desenvolver campanhas de marketing focadas na **qualidade sensorial do produto** (para atrair os influentes "Elites") e na **experiência do ambiente/atmosfera** (o segundo fator principal para notas 5 estrelas).
- **Justificativa:** Os dados mostram que "Wi-Fi Grátis" não é um diferencial de sucesso. Devemos nos posicionar como um local de alta qualidade em produto e ambiente.

Para a Estratégia de Loja (Design):

O design da loja deve suportar a estratégia de marketing.

- **Ação Imediata:** Priorizar um design de loja que promova o *ambiente* e a *experiência do café*, em vez de otimizar para estações de trabalho.
- **Justificativa:** Locais sem Wi-Fi (provavelmente focados na interação social e na qualidade do café) são, em média, mais bem avaliados e mais populares.

Apêndice: Detalhes da Análise A análise completa (Marcos 1-3) incluiu a modelagem de um Data Warehouse no PostgreSQL, criação de índices (GIN/pg_trgm) para otimização de consultas textuais, análise de correlação (Pearson), regressão linear e análise textual (TF-IDF) para validar estas descobertas. O modelo de dados ERD, estatísticas descritivas completas e scripts de carga estão disponíveis para consulta.

Referências:

PostgreSQL Download: <https://www.postgresql.org/download/>

Anaconda Download: <https://www.anaconda.com/download>

Dataset download: <https://business.yelp.com/data/resources/open-dataset/>

Dbeaver download: <https://dbeaver.io/download/>

DBDiagram online: <https://dbdiagram.io/d>

Rawgraphs online: <https://app.rawgraphs.io/>