CEFET/RJ

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA

CAMPUS MARIA DA GRAÇA

SISTEMAS DE INFORMAÇÃO

BANCO DE DADOS ||

STAR WARS

Rio de Janeiro – RJ

Dezembro-2025

STAR WARS

EQUIPE:

Bianca de Jesus

Jeovanna Picanço

Maria dos Anjos

Matheus Satana

Professor: Diego Cardoso

Sumário

1.Introdução

O trabalho tem como objetivo desenvolver um ambiente completo de banco de dados a partir de uma base pública composta por informações fornecidas por entrevistados sobre a franquia Star Wars. Esses dados abrangem perfis demográficos, opiniões sobre personagens, rankings de filmes e indicadores de preferência, configurando um conjunto amplo e heterogêneo. Assim, a primeira etapa do projeto consistiu em compreender a estrutura original da base, identificar inconsistências e elaborar um dicionário de dados inicial capaz de orientar o processo de modelagem.

Posteriormente, foi realizada uma reestruturação completa do modelo por meio de normalização, definição de entidades e chaves, padronização de tipos e correção de redundâncias. Essa etapa incluiu o desenvolvimento de scripts responsáveis por transportar os dados do modelo original para o modelo ajustado de forma íntegra e confiável. Além disso, foram criados índices destinados a otimizar o desempenho das consultas, especialmente em cenários analíticos e de grande volume de leitura.

O trabalho também contemplou a implementação de automatizações no PostgreSQL como triggers, functions, views e procedures, por fim, a construção de um Data Warehouse baseado em modelagem dimensional. Esse DW foi projetado a partir das principais perguntas de negócio identificadas, possibilitando análises mais robustas e estruturadas. Dessa forma, o projeto integra desde a análise da fonte de dados até a entrega de um ambiente analítico completo, destacando a importância das boas práticas de engenharia de dados.

2.Dicionário de Dados Inicial

Dicionário de dados inicial foi construído a partir da análise direta da base original, sem qualquer modificação estrutural. A tabela principal, denominada star_wars, contém todas as respostas dos entrevistados, reunindo dados demográficos, avaliações e preferências em uma única estrutura. Esse dicionário inicial teve como função compreender completamente o estado bruto da base e servir como ponto de partida para as etapas de normalização e reorganização de dados.

Durante a análise, foram encontradas as seguintes inconsistências estruturais:

- Ausência de chave primária (Primary Key – PK): Não existe uma coluna declarada como identificadora única da tabela. A coluna RespondentID cumpre parcialmente essa função, mas contém valores em ponto flutuante e não está configurada como PK.
- Ausência de chaves estrangeiras (Foreign Keys – FK): Mesmo contendo informações que deveriam ser distribuídas entre diferentes entidades (respondentes, filmes, rankings, personagens), toda a estrutura está agregada em uma única tabela, impossibilitando a existência de relacionamentos formais.
- Colunas sem título: nomeadas automaticamente como Unnamed: X, refletindo falhas no processo de exportação da base.
- Campos agregados: nos quais múltiplas respostas de uma lista são divididas em colunas paralelas, dificultando o tratamento relacional.
- Tipos de dados genéricos: majoritariamente varchar(50), utilizados mesmo quando o conteúdo deveria ser numérico ou categórico bem definido.

A seguir, apresenta-se o detalhamento de cada coluna, com seu nome original, tipo identificado, descrição e observações da tabela star_wars:

Coluna: RespondentID

- Tipo: float4
- Descrição: Identificador numérico do respondente.
- Observações: Não é uma chave primária declarada.

Coluna: Have you seen any of the 6 films in the Star Wars franchise?

- Tipo: varchar(50)

- Descrição: Indica se o respondente já assistiu algum dos seis filmes originais da franquia Star Wars.
- Valores comuns: “Yes”, “No”.
- Observações: Nome da coluna foi renomeado na normalização.

Coluna: Do you consider yourself to be a fan of the Star Wars film franchise?

- Tipo: varchar(50)
- Descrição: Pergunta se o respondente se considera fã da franquia Star Wars.
- Observações: Nome da coluna foi renomeado na normalização.

Coluna: Which of the following Star Wars films have you seen? Please select all that apply.

- Tipo: varchar(50)
- Descrição: Marca se o respondente assistiu determinados filmes específicos da lista.
- Observações: Aqui, apenas o primeiro filme aparece, os demais estão nas colunas Unnamed: 4 a Unnamed: 8.

Colunas: Unnamed: 4, Unnamed: 5, Unnamed: 6, Unnamed: 7, Unnamed: 8

- Tipo: varchar(50)
- Descrição: Cada coluna representa um filme adicional selecionado pelo participante.
- Valores típicos:
  - Episode I – The Phantom Menace
  - Episode II – Attack of the Clones
  - Episode III – Revenge of the Sith
  - Episode IV – A New Hope
  - Episode V – The Empire Strikes Back
  - Episode VI – Return of the Jedi
- Observações: Essas colunas são resultam da lista de filmes selecionados e necessitam ser reorganizadas.

Coluna: Please rank the Star Wars films in order of preference with 1 being your favorite film in the franchise and 6 being your least favorite film.

- Tipo: varchar(50)
- Descrição: Representa a classificação do filme preferido (1) ao menos preferido (6).
- Observações: Como nas colunas anteriores, cada ranking por filme está distribuído nas colunas seguintes: Unnamed: 10 a Unnamed: 14.

Colunas: Unnamed: 10, Unnamed: 11, Unnamed: 12, Unnamed: 13, Unnamed: 14

- Tipo: varchar(50)
- Descrição: Cada coluna corresponde ao ranking atribuído a um filme específico.
- Valores comuns: 1 a 6 (ordem de preferência).
- Observações: Essa identificação está implícita pela posição da coluna.

Colunas: Please state whether you view the following characters favorably, unfavorably, or are unfamiliar with him/her.

- Tipo: varchar(50)
- Descrição: Representam o início da seção sobre avaliação de personagens. O campo registra a opinião do respondente sobre o personagem.
- Valores comuns: Favorable, Unfavorable, Neither favorable nor unfavorable, Unfamiliar (N/A).

Colunas: Unnamed: 16 a Unnamed: 28

- Tipo: varchar(50)
- Descrição: Cada coluna contém a opinião do respondente sobre um personagem específico.
- Exemplos de personagens mapeados:
  - Han Solo
  - Luke Skywalker
  - Princess Leia
  - Anakin Skywalker
  - Obi-Wan Kenobi

- Darth Vader
- Palpatine
- Yoda
- Padmé Amidala
- Jar Jar Binks
- C-3PO
- R2-D2

- Observações: Colunas foram separadas em uma tabela na etapa normalização.
- Valores comuns:
- "Very favorably"
- "Somewhat favorably"
- "Neither favorably nor unfavorably"
- "Unfamiliar (N/A)"

Coluna: Which character shot first?

- Tipo: varchar(50)
- Descrição: Identifica a resposta do participante à pergunta polêmica “Quem atirou primeiro?”
- Valores comuns: “Han”, “Greedo”, “I don't understand this question”.

Coluna: Are you familiar with the Expanded Universe?

- Tipo: varchar(50)
- Descrição: Indica se o respondente conhece o Universo Expandido de Star Wars.

Coluna: Do you consider yourself to be a fan of the Expanded Universe?

- Tipo: varchar(50)
- Descrição: Pergunta se o participante se considera fã do Universo Expandido.

Coluna: Do you consider yourself to be a fan of the Star Trek franchise?

- Tipo: varchar(50)

- Descrição: Pergunta se o participante é fã da franquia Star Trek.

Coluna: Gender

- Tipo: varchar(50)
- Descrição: Gênero do participante.
- Valores comuns: “Male”, “Female”.

Coluna: Age

- Tipo: varchar(50)
- Descrição: Faixa etária do respondente.
- Exemplos: “18–29”, “30–44”, “45–60”.

Coluna: Household Income

- Tipo: varchar(50)
- Descrição: Faixa de renda familiar.
- Exemplos:
  - “$0 – $24,999”
  - “$25,000 – $49,999”
  - “$100,000 – $149,999”

Coluna: Education

- Tipo: varchar(50)
- Descrição: Nível educacional.
- Exemplos:
  - “High school degree”
  - “Some college”
  - “Bachelor degree”
  - “Graduate degree“

Coluna: Location (Census Region)

- Tipo: varchar(50)
- Descrição: Região censitária dos EUA onde o participante reside.
- Exemplos:

- “South Atlantic”
- “Pacific”
- “West North Central”
- “Middle Atlantic”

3.Análise da Base, Ajustes e Indexação

A análise estrutural da tabela star_wars indicou uma série de problemas que comprometiam a integridade lógica do banco e dificultavam consultas e análises. Entre os principais problemas identificados, destacam-se a ausência normalização, colunas nas sem nomenclatura adequada, repetição de grupos de atributos, tipos incorretos, nenhuma chave primária e ausência de relacionamento. Com base nos problemas identificados, foi elaborado um processo de normalização, reorganizando a base em entidades distintas. O objetivo foi corrigir inconsistências, preservar os dados existentes e proporcionar maior integridade e desempenho.

DDL:

Tabela: respondentid

- Descrição Geral: Armazena exclusivamente os identificadores originais dos respondentes, preservando a referência única da base bruta.
- Colunas: respondent_id (BIGINT): corresponde ao identificador do respondente conforme presente na base original. Representa a chave primária da tabela. Antes da normalização, este valor estava armazenado como float, o que justificou sua conversão para BIGINT.
- Chave Primária: respondent_id
- Chaves Estrangeiras: não possui.
- Justificativa: Isolar o identificador original evita perda de rastreabilidade e permite manter o ID bruto como referência.

Tabela: respostas

- Descrição Geral: Armazena os dados demográficos e as respostas individuais fornecidas pelo participante. Cada respondente recebe um identificador interno gerado automaticamente.
- Colunas:
  - id (BIGINT IDENTITY): identificador interno do respondente. Serve como chave primária desta tabela.
  - respondent_id (BIGINT): referencia o ID original do participante na tabela respondentid. É utilizado como chave estrangeira.
  - gender (VARCHAR): gênero informado pelo participante. Exemplos: “Male”, “Female”.

  - age (idade – ENUM): faixa etária do respondente. Valores permitidos: “18-29”, “30-44”, “45-60”, “>60”.
  - household_income (VARCHAR): faixa de renda familiar. Exemplos: “$0-24,999”, “$50,000-99,999”.
  - education (VARCHAR): nível educacional. Exemplos: “High school degree”, “Bachelor degree”, “Graduate degree”.
  - region (VARCHAR): região censitária de residência. Exemplos: “Pacific”, “South Atlantic”.
  - seen_any_star_wars (VARCHAR): indica se o respondente já assistiu algum filme da franquia Star Wars.
  - fan_of_star_wars (VARCHAR): informa se o participante se considera fã da franquia Star Wars.
  - fan_of_startrek (VARCHAR): indica se o participante é fã da franquia Star Trek.
  - fan_of_expanded_universe (VARCHAR): determina se o participante gosta do Universo Expandido.
  - familiar_with_expanded_universe (VARCHAR): indica se o participante conhece o Universo Expandido.
  - who_shot_first (VARCHAR): resposta à pergunta “Quem atirou primeiro?”.
- Justificativa: A criação de um ID interno melhora desempenho em operações relacionais e padroniza vínculos. A separação dos dados demográficos em tabela própria elimina redundâncias e atende às regras da normalização.
- Chaves:
  - PK: id
  - FK: respondent_id → respondentid.respondent_id

Tipo ENUM: idade

- Descrição: Define valores válidos para faixa etária, evitando inconsistências como variações textuais.
- Valores permitidos: “18-29”, “30-44”, “45-60”, “>60”.
- Justificativa: Evita erros de digitação e garante padronização total dos valores atribuídos à idade.

Tabela: film

- Descrição Geral: Contém o catálogo oficial dos filmes referenciados na pesquisa.
- Colunas:
  - filmID (INT IDENTITY): identificador único do filme. É chave primária.
  - film_name (VARCHAR): nome completo do filme. Exemplo: “Star Wars: Episode IV – A New Hope”.
- Chaves: PK: filmID
- Justificativa: Na base original, as informações sobre filmes estavam distribuídas em várias colunas (ex.: Unnamed: 4 a Unnamed: 8). A separação em catálogo elimina redundância e permite relacionamento estruturado.

Tabela: film_seen

- Descrição Geral: Registra quais filmes foram vistos por cada respondente. Substitui as múltiplas colunas de indicação de filmes assistidos da base original.
- Colunas:
- respondent_id (BIGINT): identifica o respondente.
- film_id (INT): identifica o filme.
- seen (VARCHAR): indica se o filme foi visto (“Yes” ou “No”).
- Chaves:
  - PK: film_seed_id
  - FK:
    - respondent_id → respostas.id
    - film_id → film.filmID
- Justificativa: A normalização transforma seis colunas redundantes em uma.

Tabela: film_ranking

- Descrição Geral:Armazena o ranking dado por cada participante aos filmes. Cada ranking antes estava distribuído em colunas distintas.
- Colunas:
- film_ranking_id: chave primária.

- respondent_id (BIGINT): identifica o respondente.
- film_id (INT): identifica o filme ranqueado.
- ranking (INT): nota atribuída, com 1 sendo o filme favorito e 6 o menos apreciado.
- Chaves:
- PK: film_ranking_id
- FK:
    - respondent_id → respostas.id
    - film_id → film.filmID
- Justificativa: Essa estrutura remove a repetição de colunas e possibilita análises como rankings e comparação entre grupos demográficos.

Tabela: character_film

- Descrição Geral: Catálogo de personagens avaliados na pesquisa.
- Colunas:
    - character_id (INT IDENTITY): identificador único do personagem. Chave primária.
    - character_name (VARCHAR): nome do personagem, como “Luke Skywalker”, “Yoda”, “Darth Vader”.
- Justificativa: Na base bruta, cada personagem era representado por uma coluna Unnamed. Criar um catálogo é essencial para manter consistência e facilitar integrações com avaliações.

Tabela: character_opinion

- Descrição Geral: Tabela que registra a opinião do respondente sobre cada personagem do catálogo.
- Colunas:
    - caratecter_opnion_id: chave primaria.
    - respondent_id (BIGINT): identificador do respondente.
    - character_id (INT): personagem avaliado.
    - opinion (VARCHAR): opinião expressa, como “Very favorably”, “Unfavorably”, “Unfamiliar”.

- Justificativa: Transforma 14 colunas de opinião em uma estrutura eficiente, simples e totalmente normalizada.
- Chaves:
- PK: caratecter_opnion_id
- FK:
    - respondent_id → respostas.id
    - character_id → character_film.character_id

DML:

Inserção dos Respondentes (RespondentID)

- Operação: Inserção de todos os identificadores de participantes presentes na tabela bruta star_wars.
- Propósito: Isolar o identificador original do respondente (RespondentID) em uma tabela própria, preservando integridade do dado bruto antes da normalização.
- Descrição: A consulta seleciona todos os valores distintos da coluna "RespondentID" na base original e converte o valor para BIGINT, visto que a origem utiliza tipo float.
- Justificativa: Impede perda de dados causada por arredondamentos ou imprecisão numérica. Permite ligação exata entre o respondente original e o novo identificador interno criado na tabela respostas.

Inserção Respostas (respostas)

- Operação: Carregamento da tabela respostas com dados do respondente, convertendo valores brutos para formatos padronizados.
- Descrição: São preenchidos os atributos como gênero, idade, escolaridade, renda, região e respostas diretas sobre hábitos de consumo e preferências de franquias. A faixa etária passa por conversão para o tipo ENUM idade, garantindo validade.
- Justificativa: Padroniza tipos de dados antes armazenados como texto genérico. Garante consistência, especialmente para campos categóricos como gênero, idade e região. Remove redundância existente na tabela

original, que concentrava múltiplas temáticas em uma única estrutura. Cria um identificador interno mais eficiente para relacionamentos futuros.

Inserção dos Filmes (film)

- Operação: Carregamento da tabela de filmes com todos os títulos presentes na pesquisa.
- Descrição: A operação unifica títulos distribuídos em diversas colunas da base bruta, removendo duplicidades e ordenando conforme a cronologia dos episódios.
- Justificativa: Normaliza dados originalmente separados em colunas distintas (“Which of the following films…”, “Unnamed: 4”, etc.). Evita inconsistência de títulos repetidos ou digitados de forma diferente. Cria uma entidade limpa e padronizada que servirá para vários relacionamentos.

Inserção dos Personagens (character_film)

- Operação: Carga dos nomes dos personagens avaliados na pesquisa.
- Descrição: Os nomes são inseridos manualmente com seus respectivos IDs seguindo a ordem implicada pela base original.
- Justificativa: Substitui colunas sem título (“Unnamed: 16 a Unnamed: 28”). Centraliza e padroniza todos os personagens em um único catálogo. Permite relacionamentos consistentes com opiniões de personagens.

Inserção dos Filmes Assistidos (film_seen)

- Operação: Transformação de múltiplas colunas de filmes vistos em um formato relacional Respondente × Filme.
- Descrição: Para cada respondente, cruza-se todos os filmes cadastrados no catálogo e insere-se o valor correspondente à coluna adequada da tabela original.
- Justificativa: Converte um conjunto de colunas redundantes em uma estrutura normalizada 1:N. Reduz drasticamente duplicidade e inconsistência semântica.

Inserção dos Rankings de Filmes (film_ranking)

- Operação: Carrega as notas atribuídas aos filmes, que estavam distribuídas em várias colunas do tipo "Unnamed".
- Descrição: Os valores são convertidos para inteiro e vinculados ao filme correspondente. Campos vazios ou inconsistentes são descartados.
- Justificativa: Normaliza rankings que estavam divididos em colunas paralelas. Permite cálculos como média de ranking. Evita uso de strings para representar valores numéricos.

Inserção das Opiniões sobre Personagens (character_opinion)

- Operação: Carregamento das avaliações favoráveis, desfavoráveis e neutras para cada personagem.
- Descrição: A operação cruza o catálogo de personagens com cada respondente para inserir a opinião correspondente à coluna original correta. Somente valores não nulos são aplicados.
- Justificativa: Remove 14 colunas redundantes e as converte em relacionamentos 1:N. Padroniza respostas antes espalhadas por múltiplas colunas "Unnamed".

Índices:

Índice: idx_respondentid_pk

- Tabela: RespondentID
- Coluna: respondent_id
- Descrição: Criado sobre o identificador original da pesquisa.
- Justificativa: O índice acelera pesquisas e junções entre a tabela respondentid e a tabela respostas, especialmente porque essa coluna é frequentemente utilizada para rastrear o respondente original. Melhora desempenho em validações, integridade e cruzamentos.

Índice: idx_respostas_id

- Tabela: respostas
- Coluna: id
- Descrição: Chave primária gerada automaticamente pela tabela.

- Justificativa: Garante unicidade e integridade da identificação interna dos respondentes. O índice associado à chave primária facilita consultas, junções e filtragens, sendo fundamental em todas as relações Respondente × Entidade.

Índice: idx_respostas_respondent_id

- Tabela: respostas
- Coluna: respondent_id
- Descrição: Representa o identificador do respondente original vinculado via FK à tabela respondentid.
- Justificativa: Melhora desempenho nas funções de transformação, relações com tabelas derivadas e consultas que precisam localizar rapidamente um participante pela ID original da pesquisa.

Índice: idx_film_filmid

- Tabela: film
- Coluna: filmid
- Descrição: Chave primária da tabela de filmes.
- Justificativa: Garante unicidade do catálogo de filmes. O índice facilita junções da tabela film com film_seen e film_ranking, acelerando consultas comuns como rankings, porcentagem de visualização e filtragens por filme.

Índice: idx_film_film_name

- Tabela: film
- Coluna: film_name
- Descrição: Índice criado sobre os títulos dos filmes.
- Justificativa: Acelera pesquisas por nome do filme, especialmente úteis em consultas descritivas, buscas por texto e filtros administrativos.

Índice: idx_film_seen_id

- Tabela: film_seen
- Colunas: film_seen_id

- Descrição: Identifica de forma única se um respondente viu determinado filme.
- Justificativa: Evita duplicidade e garante integridade da relação Respondente × Filme. O índice composto melhora desempenho de consultas como: "Filmes vistos por um respondente" e "Quantidade de espectadores de um filme específico".

Índice: idx_film_seen_respondent_id

- Tabela: film_seen
- Coluna: respondent_id
- Descrição: Usado para filtrar rapidamente todos os filmes vistos por um respondente.
- Justificativa: Auxilia funções, cálculos estatísticos e consultas frequentes de uso analítico.

Índice: idx_film_seen_film_id

- Tabela: film_seen
- Coluna: film_id
- Descrição: Facilita consultas baseadas no filme.
- Justificativa: Permite identificar, de forma rápida, o total de espectadores por filme, acelerando análises de popularidade.

Índice: idx_film_ranking_id

- Tabela: film_ranking
- Colunas: film_ranking_id
- Descrição: Chave primária assegura que cada respondente fornece apenas um ranking por filme.
- Justificativa: Evita dados duplicados e mantém a integridade da avaliação individual do filme. O índice acelera cálculos como média, mediana e ranking geral dos episódios.

Índice: idx_film_ranking_film_id

- Tabela: film_ranking

- Coluna: film_id
- Descrição: Criado para melhorar desempenho de consultas que agregam ou filtram por filme.
- Justificativa: Funções analíticas como a média de ranking dependem desse índice para execução rápida.

Índice: idx_character_film_id

- Tabela: character_film
- Coluna: character_id
- Descrição: Identificador único de cada personagem.
- Justificativa: Garante armazenamento correto dos personagens e suporta relacionamentos com opiniões dos usuários. Como catálogo é pequeno, o índice é leve mas essencial.

Índice: idx_character_film_name

- Tabela: character_film
- Coluna: character_name
- Descrição: Índice criado para facilitar pesquisas e filtragens por nome dos personagens.
- Justificativa: Permite buscas mais rápidas, especialmente úteis em interfaces administrativas, BI ou consultas exploratórias.

Índice: idx_character_opinion_id

- Tabela: character_opinion
- Colunas: character_opinion_id
- Descrição: Define uma opinião por personagem por respondente.
- Justificativa: Evita duplicidade, garante consistência e torna eficiente a leitura de dados por respondente ou personagem.

Índice: idx_character_opinion_character_id

- Tabela: character_opinion
- Coluna: character_id
- Descrição: Facilita consultas sobre opinião agregada por personagem.

- Justificativa: Usado em análises como "qual personagem tem melhor imagem" ou "qual personagem é mais polarizador".

4.Criação de Automatizações no PostgreSQL

FUNCTIONS:

contar_filmes_vistos()

- Descrição: Retorna o total de filmes vistos por um respondente, tratando múltiplas origens de ID.
- Importância: Permite medir engajamento real do participante e é usada em relatórios e validações internas.

obter_ranking_medio_filme()

- Descrição: Calcula o ranking médio de um filme, ignorando valores nulos.
- Importância: Gera estatísticas fundamentais para dashboards e comparações entre filmes.

eh_fan_star_wars()

- Descrição: Indica se um respondente declarou ser fã da franquia.
- Importância: Usado para segmentar análises entre fãs e não-fãs, crucial para entender padrões de opinião.

PROCEDURES:

inserir_respondente_com_validacao()

- Descrição: Insere o respondente garantindo existência na tabela base e assegurando consistência dos dados.
- Importância: Evita registros órfãos e uniformiza o processo de cadastro.

atualizar_opiniao_personagem_lote()

- Descrição: Atualiza opiniões antigas para um valor novo, útil para correções.
- Importância: Facilita ajustes massivos sem atualizar manualmente cada registro.

limpar_respondente()

- Descrição: Exclui todas as respostas ligadas a um respondente.
- Importância: Atende necessidades de privacidade e auditoria (LGPD), removendo dados sob requisição.

TRIGGERS:

trigger_contar_filme_visto

- Descrição: Executado após inserção em film_seen. Hoje só força atualização, mas serve como mecanismo para manter estatísticas.
- Importância: Base para futuros contadores automáticos de engajamento.

trigger_validar_ranking

- Descrição: Impede inserção de ranking fora de 1 a 6.
- Importância: Garante integridade dos dados de qualidade e evita distorções estatísticas.

trigger_validar_opinion_nao_vazia

- Descrição: Bloqueia opiniões vazias em character_opinion.
- Importância: Evita inconsistências e registros inúteis na base.

VIEWS:

v_respondentes_por_regiao

- Descrição: Apresenta estatísticas de respondentes agrupados por região.
- Importância: Fundamental para estudos de distribuição geográfica e representatividade da pesquisa.

v_ranking_medio_filmes

- Descrição: Combina dados do catálogo com rankings, mostrando média, extremos e volume de avaliações.
- Importância: Permite avaliações comparativas e insights sobre preferências.

v_fans_vs_nao_fans

- Descrição: Compara fãs e não fãs, incluindo correlação com Star Trek e visualização de filmes.
- Importância: Crucial para análises comportamentais do público.

5.Modelagem do Data Warehouse (DW)

Nesta modelagem, os dados são organizados em uma tabela fato, que centraliza os eventos mensuráveis, e em tabelas dimensão, que fornecem contexto descritivo e categórico para esses eventos. O tema do Data Warehouse é a opinião e comportamento dos respondentes em relação a filmes e personagens, além da identificação de fãs de determinados universos, como Star Wars e Star Trek. Assim, o Data Warehouse é capaz de representar e analisar informações relacionadas a preferências, hábitos de consumo de mídia e percepções de personagens.

DDL:

Tipo Enumerado

- Criação de um tipo para registrar operações (inclusão, atualização, exclusão).
- Justificativa: Permite rastrear transformações e controlar a qualidade dos dados carregados.

Dimensão Respondente

- Contém dados demográficos: gênero, faixa etária, renda, escolaridade e região.
- Justificativa: Permite segmentar análises por características sociais, essenciais para interpretar preferências e padrões de consumo.

Dimensão Filme

- Lista de filmes com identificador único e nome.
- Justificativa: Estrutura fundamental para análises de popularidade, consumo e comparação entre títulos.

Dimensão Personagem

- Armazena personagens avaliados pelos respondentes.
- Justificativa: Permite mensurar aprovação, rejeição e padrões de percepção do público sobre personagens específicos.

Tabela Fato Respostas

- Registra eventos entre respondente–filme–personagem: opinião, assistiu, ranking, indicações de fã.
- Justificativa: Consolida todas as interações analisáveis e serve como base central para cruzamento das dimensões.

Índices

- Criados sobre colunas de filtros frequentes (opinião, assistiu, fã).
- Justificativa: Aumentam significativamente o desempenho das consultas analíticas.

Tabela de Controle de ETL

- Registra data e hora da última carga do DW.
- Justificativa: Suporte operacional, garantindo integridade e rastreabilidade das cargas.

DML:

Carga da Dimensão Respondente

- Selecionam-se respondentes únicos do sistema fonte, extraindo dados demográficos. Registros já existentes não são reinseridos.
- Função: Garantir que cada respondente possua descrição única e consistente.

Carga da Dimensão Filme

- São listados todos os filmes do sistema original e inseridos na dimensão, evitando duplicações.
- Função: Padronizar a lista oficial de filmes para referência em análises.

Carga da Dimensão Personagem

- Personagens são extraídos e adicionados à dimensão, seguindo o mesmo critério de unicidade.
- Função: Garantir correspondência correta entre opiniões e personagens.

Carga da Tabela Fato Respostas

- Integra dados das tabelas operacionais: respondentes, filmes assistidos, rankings e opiniões.
- Aplicam-se transformações como: Conversão de respostas textuais em booleanas, associação de ranking ao filme correto e consolidação da opinião sobre personagem.
- Resultado: Cada linha representa uma interação completa entre respondente, filme e personagem, pronta para análises multidimensionais.

Perguntas de Negócio Respondidas pelo DW:

Quais personagens têm as opiniões mais positivas?

- A tabela fato armazena todas as opiniões e permite filtrar avaliações positivas. A dimensão personagem identifica quem recebeu cada avaliação.
- O DW permite:
    - Contabilizar elogios por personagem;
    - Comparar aprovação por faixa etária, gênero ou renda;
    - Identificar personagens com maior potencial comercial e narrativo.

Quais filmes são mais assistidos por faixa etária?

- A tabela fato registra se o filme foi assistido; a dimensão respondente fornece a faixa etária.
- O DW permite:
    - Agrupar assistências por idade;
    - Descobrir quais filmes são mais populares entre jovens, adultos ou idosos;
    - Analisar tendências de consumo e direcionar campanhas segmentadas.

Quantos fãs de Star Wars também são fãs de Star Trek?

- A tabela fato contém indicadores de fã para ambos os universos.
- O DW permite:
    - Classificar respondentes em quatro grupos (fãs de ambos, de um só ou de nenhum);
    - Quantificar sobreposição entre fandoms;

- Auxiliar em estratégias de mercado, conteúdos cruzados e estudos de público-alvo.

TRIGGERS:

trigger_caracter

- Função: fn_dim_character
- Tabela de origem: public.character_film
- Tabela de destino: dw.dim_character
- Operações capturadas: INSERT, UPDATE, DELETE
- Justificativa: Permite acompanhar alterações nos personagens avaliados, mantendo histórico e integridade da dimensão.
- Perguntas de negócio atendidas:
  - Quais personagens têm as opiniões mais positivas?
  - Há alterações na percepção do público sobre determinado personagem ao longo do tempo?

trigger_fato

- Função: fn_fato_respostas
- Tabela de origem: public.respostas
- Tabela de destino: dw.fato_respostas
- Operações capturadas: INSERT, UPDATE, DELETE
- Justificativa: Consolida os dados que representam interações mensuráveis, permitindo análises multidimensionais.
- Perguntas de negócio atendidas:
  - Quais filmes são mais assistidos por faixa etária?
  - Quantos fãs de Star Wars também são fãs de Star Trek?
  - Quais personagens possuem maior aprovação?

trigger_filme

- Função: fn_dim_film
- Tabela de origem: public.film
- Tabela de destino: dw.dim_film
- Operações capturadas: INSERT, UPDATE, DELETE

- Justificativa: Garante que todos os filmes analisados estejam corretamente registrados na dimensão, facilitando comparações e análises de popularidade.
- Perguntas de negócio atendidas:
    - Quais filmes são mais populares entre diferentes faixas etárias?
    - Comparar consumo de filmes ao longo do tempo

trigger_respondent

- Função: fn_dim_respondent
- Tabela de origem: public.respostas
- Tabela de destino: dw.dim_respondent
- Operações capturadas: INSERT, UPDATE, DELETE
- Justificativa: Permite segmentação de análises por características demográficas, essencial para interpretar padrões de comportamento e preferências.
- Perguntas de negócio atendidas:
    - Como o comportamento dos respondentes varia conforme idade, gênero ou região?
    - Existe correlação entre características demográficas e fandom (Star Wars/Star Trek)?

6.Considerações finais

O desenvolvimento do projeto permitiu compreender de forma aprofundada a importância da modelagem correta de um banco de dados e dos riscos associados ao uso de fontes não normalizadas. A organização da base inicial, que apresentava inconsistências e estruturas pouco adequadas, possibilitou a criação de um modelo mais eficiente, coerente e alinhado às boas práticas de sistemas relacionais. Essa transformação garantiu maior clareza sobre o domínio e facilitou a posterior aplicação de rotinas de análise e processamento.

A migração dos dados preservou integralmente as informações relevantes e restabeleceu relações consistentes entre respondentes, filmes e personagens. Ademais, a criação de índices otimizou o desempenho das consultas, especialmente em operações de cruzamentos estatísticos e exploração analítica. O uso de automatizações no PostgreSQL contribuiu para a integridade do sistema, a padronização de operações e a automação de tarefas rotineiras.

De forma geral, o trabalho evidencia a relevância da normalização, da definição rigorosa de chaves, da documentação técnica e da modelagem adequada para assegurar qualidade e eficiência em projetos de dados. Demonstra também como uma base simples, mas mal estruturada, pode ser transformada em um ambiente robusto, preparado para análises avançadas, construção de Data Warehouses e integração com pipelines de ETL. Assim, reforça-se a importância da engenharia de dados como disciplina essencial para a geração de valor por meio da informação.

7.Referências Bibliográficas

KAGGLE. Official Crime Data – São Paulo State (Brazil) – SSP. Disponível em: https://www.kaggle.com/datasets/dbwaller/official-crime-data-sao-paulo-statebrazil-ssp/data. Acesso em: 30 nov. 2025.