



UFRJ – Universidade Federal do Rio de Janeiro

IM – Instituto de Matemática

DMA – Departamento de Matemática Aplicada

EEL890 - Big Data (turma SIGA 16338)

Avaliação 02: Modelagem de Data Warehouse

PARTE II - Modelagem DW

Alunos:

Gabriel Rodrigues da Silva - 121044858

Giovanni Paes Leme da Gama Rodrigues - 117054744

Gabriel Brígido Pinheiro da Silva - 120056519

Nicolas Viana do Espírito Santo - 121042953

1. Introdução

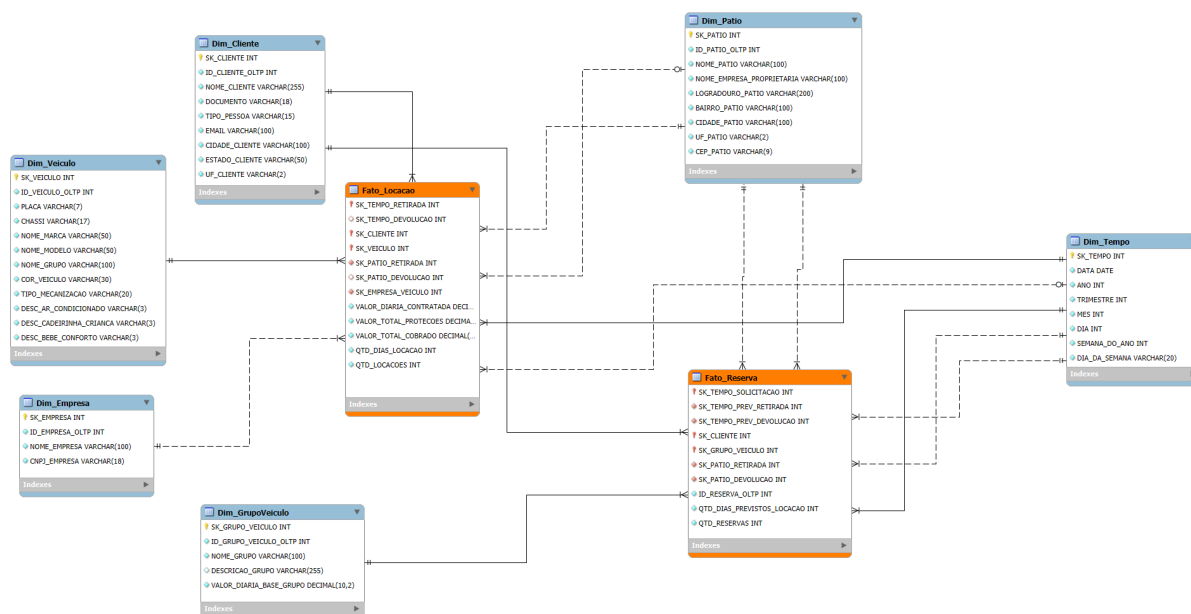
Este documento detalha o processo de desenvolvimento e as escolhas de projeto na construção de uma solução de Data Warehouse (DW) para um conjunto de empresas independentes de locação de veículos. O objetivo principal é integrar dados operacionais de múltiplos sistemas transacionais (OLTP) heterogêneos em uma única plataforma analítica, permitindo a geração de relatórios gerenciais unificados e a realização de análises estratégicas. A solução visa fornecer uma visão ampla do negócio, cobrindo as áreas de clientes, frota, pátios, reservas e locações.

2. Proposta do Modelo Dimensional

Para atender aos requisitos de análise do negócio, optou-se por um modelo dimensional, que é otimizado para consultas e geração de relatórios, em contraste com o modelo transacional focado em operações. A arquitetura do Data Warehouse foi desenvolvida seguindo o princípio do Esquema **Estrela** (Star Schema), que é otimizado para consultas e geração de relatórios. Esta abordagem foi escolhida por sua simplicidade e alta performance, facilitando a compreensão dos dados pelos usuários finais. Como o projeto precisava medir múltiplos processos de negócio — especificamente **Locação** e **Reserva** — a implementação final se materializa como uma **Constelação de Fatos** (Fact Constellation), que é, na essência, uma **coleção de esquemas estrela individuais**.

O pilar desta arquitetura é o uso de **Dimensões Conformadas** (como *Dim_Cliente*, *Dim_Patio* e *Dim_Tempo*) que são compartilhadas entre todas as tabelas de fatos. Isso garante a consistência dos dados em toda a organização e permite a análise cruzada entre os diferentes processos, como comparar o volume de reservas com as locações efetivamente realizadas.

Abaixo, está disponível a figura do Modelo Dimensional citado.



Fonte: https://github.com/gbrods/EEL890_BigData/blob/main/P2_Parte2/Modelo%20Dimensional/modeloDimensional.png

3. Dicionário de Dados do Modelo Dimensional

A seguir, é apresentado o dicionário de dados detalhado para cada tabela do modelo dimensional proposto, descrevendo os atributos, tipos de dados e restrições. Cada campo foi definido para atender a um requisito específico de análise ou para enriquecer o contexto dos dados.

Tabela: **Dim_Cliente**

Armazena os dados descritivos e consolidados dos clientes da locadora.

Atributo	Tipo	PK/FK	Nulo?	Descrição
SK_CLIENTE	INT	PK	Não	Identificador único do cliente no Data Warehouse (chave primária).
ID_CLIENTE_OLTP	INT		Não	ID original do cliente no sistema transacional. Usado para ligar o dado à origem.
NOME_CLIENTE	VARCHAR(255)		Não	Nome completo ou Razão Social do cliente.
DOCUMENTO	VARCHAR(18)		Não	CPF ou CNPJ do cliente.
TIPO_PESSOA	VARCHAR(15)		Não	Define se o cliente é 'Pessoa Física' ou 'Pessoa Jurídica'.

EMAIL	VARCHAR(100)		Não	Endereço de e-mail de contato do cliente.
CIDADE_CLIENTE	VARCHAR(100)		Não	Cidade de residência do cliente, obtida do seu endereço.
ESTADO_CLIENTE	VARCHAR(50)		Não	Estado de residência do cliente, obtido do seu endereço.
UF_CLIENTE	VARCHAR(2)		Não	Sigla do estado (UF) de residência do cliente, obtida do seu endereço.

Tabela: Dim_Veiculo

Armazena os dados consolidados dos veículos da frota.

Atributo	Tipo	PK/FK	Nulo?	Descrição
SK_VEICULO	INT	PK	Não	Identificador único do veículo no Data Warehouse (chave primária).
ID_VEICULO_OLTP	INT		Não	ID original do veículo no sistema transacional. Usado para ligar o dado à origem.
PLACA	VARCHAR(7)		Não	Placa de identificação do veículo.
CHASSI	VARCHAR(17)		Não	Número do chassi do veículo.
NOME_MARCA	VARCHAR(50)		Não	Nome da marca do veículo, como Fiat ou Chevrolet. Originado de MarcaVeiculo.
NOME_MODELO	VARCHAR(50)		Não	Nome do modelo do veículo, como Mobi ou Onix. Originado de ModeloVeiculo.
NOME_GRUPO	VARCHAR(100)		Não	Nome do grupo/categoria ao qual o veículo pertence, como 'Econômico' ou 'SUV'. Originado de GrupoVeiculo.
COR_VEICULO	VARCHAR(30)		Não	Cor predominante do veículo.
TIPO_MECANIZACAO	VARCHAR(20)		Não	Tipo de mecanização do veículo (ex: Manual, Automática).
DESC_AR_CONDICIONADO	VARCHAR(3)		Não	Valor textual ('Sim'/'Não') para clareza nos relatórios. Transformado do campo POSSUI_AR_CONDICIONADO do sistema original.
DESC_CADEIRINHA_CRIANCA	VARCHAR(3)		Não	Valor textual ('Sim'/'Não') para clareza nos relatórios. Transformado do campo POSSUI_CADEIRINHA_CRIANCA do sistema original.

DESC_BEBE_CONFORTO	VARCHAR(3)		Não	Valor textual ('Sim'/'Não') para clareza nos relatórios. Transformado do campo POSSUI_BEBE_CONFORTO do sistema original.
--------------------	------------	--	-----	---

Tabela: Dim_Patio

Armazena os dados descritivos e consolidados dos pátios, permitindo identificar a localização física onde os veículos são retirados e devolvidos, e realizar análises geográficas e de desempenho por localidade.

Atributo	Tipo	PK/FK	Nulo?	Descrição
SK_PATIO	INT	PK	Não	Identificador único do pátio no Data Warehouse (chave primária).
ID_PATIO_OLTP	INT		Não	ID original do pátio no sistema transacional. Usado para ligar o dado à origem.
NOME_PATIO	VARCHAR(100)		Não	Nome do pátio (ex: Galeão, Santos Dumont, Barra Shopping).
NOME_EMPRESA_PROPRIETARIA	VARCHAR(100)		Não	Nome da empresa que administra o pátio. Originado da tabela Empresa via FK_ID_EMPRESA.
LOGRADOURO_PATIO	VARCHAR(200)		Não	Logradouro (rua, avenida) do endereço do pátio.
BAIRRO_PATIO	VARCHAR(100)		Não	Bairro onde o pátio está localizado.
CIDADE_PATIO	VARCHAR(100)		Não	Cidade onde o pátio está localizado.
UF_PATIO	VARCHAR(2)		Não	Sigla do estado (UF) onde o pátio está localizado.
CEP_PATIO	VARCHAR(9)		Não	CEP do endereço do pátio.

Tabela: Dim_Tempo

Armazena uma linha para cada dia de um determinado período. Fornece os atributos básicos para permitir que os fatos sejam analisados por dia, mês, ano, trimestre e semana.

Atributo	Tipo	PK/FK	Nulo?	Descrição
SK_TEMPO	INT	PK	Não	Chave primária da tabela, geralmente no formato AAAAMMDD (ex: 20250621).
DATA	DATE		Não	A data completa no formato AAAA-MM-DD.
ANO	INT		Não	Ano correspondente à data.
TRIMESTRE	INT		Não	Número do trimestre do ano (1 a 4).
MES	INT		Não	Número do mês no ano (1 a 12).
DIA	INT		Não	Número do dia no mês (1 a 31).
SEMANA_DO_ANO	INT		Não	Número da semana no ano (1 a 53).
DIA_DA_SEMANA	VARCHAR(20)		Não	Nome do dia da semana (ex: 'Segunda-feira').

Tabela: Dim_Empresa

Armazena os dados das seis empresas de locação de veículos que formam o grupo. Permite diferenciar a propriedade de pátios e veículos nas análises.

Atributo	Tipo	PK/FK	Nulo?	Descrição
SK_EMPRESA	INT	PK	Não	Identificador único da empresa no Data Warehouse (chave primária).
ID_EMPRESA_OLTP	INT		Não	ID original da empresa no sistema transacional. Usado para ligar o dado à origem.
NOME_EMPRESA	VARCHAR(100)		Não	Nome da empresa de locação de veículos.
CNPJ_EMPRESA	VARCHAR(18)		Não	CNPJ (Cadastro Nacional da Pessoa Jurídica) da empresa.

Tabela: Dim_GrupoVeiculo

Armazena os dados descritivos dos grupos ou categorias de veículos. Permite a análise da demanda e o desempenho por diferentes tipos de carro.

Atributo	Tipo	PK/FK	Nulo?	Descrição
SK_GRUPO_VEICULO	INT	PK	Não	Identificador único do grupo de veículo no Data Warehouse (chave primária).
ID_GRUPO_VEICULO_OLTP	INT		Não	ID original do grupo no sistema transacional. Usado para ligar o dado à origem.
NOME_GRUPO	VARCHAR(100)		Não	Nome do grupo ou categoria do veículo (ex: 'Econômico Básico', 'SUV Compacto').
DESCRICAO_GRUPO	VARCHAR(255)		Sim	Descrição detalhada com as características do grupo de veículo.
VALOR_DIARIA_BASE_GRUPO	DECIMAL(10,2)		Não	Valor de referência da diária para este grupo, usado como base de preço.

Tabela: Fato_Locacao

Armazena as métricas de cada evento de locação. O grão desta tabela é uma locação individual. Cada linha representa um aluguel de veículo, desde a sua retirada até a sua devolução, e contém os valores e quantidades associadas a ele.

Atributo	Tipo	PK/FK	Nulo?	Descrição
SK_TEMPO_RETIRADA	INT	FK	Não	Chave estrangeira para a Dim_Tempo, ligada à data em que o veículo foi retirado.
SK_TEMPO_DEVOLUCAO	INT	FK	Sim	Chave estrangeira para a Dim_Tempo, ligada à data em que o veículo foi devolvido. Pode ser nula se a locação estiver em andamento.
SK_CLIENTE	INT	FK	Não	Chave estrangeira para a Dim_Cliente, identificando quem alugou o veículo.
SK_VEICULO	INT	FK	Não	Chave estrangeira para a Dim_Veiculo, identificando qual veículo foi alugado.
SK_PATIO_RETIRADA	INT	FK	Não	Chave estrangeira para a Dim_Patio, identificando de onde o veículo foi retirado.

SK_PATIO_DEVOLUCAO	INT	FK	Sim	Chave estrangeira para a Dim_Patio, identificando onde o veículo foi devolvido. Pode ser nula se a locação estiver em andamento.
SK_EMPRESA_VEICULO	INT	FK	Não	Chave estrangeira para a Dim_Empresa, identificando a empresa dona do veículo.
VALOR_DIARIA_CONTRATADA	DECIMAL(10,2)		Não	Valor da diária acordado para a locação, conforme registrado na tabela Locacao.
VALOR_TOTAL_PROTECOES	DECIMAL(10,2)		Não	Valor total das proteções adicionais contratadas. É a soma dos valores da tabela LocacaoProtecao para esta locação.
VALOR_TOTAL_COBRADO	DECIMAL(10,2)		Não	Valor total efetivamente cobrado do cliente. É a soma dos valores da tabela Cobranca para esta locação.
QTD_DIAS_LOCAÇÃO	INT		Não	Quantidade de dias que a locação durou. Calculado a partir da diferença entre a data de devolução e a de retirada.
QTD_LOCACOES	INT		Não	Contador para facilitar a contagem de locações. Terá sempre o valor 1.

Tabela: **Fato_Reserva**

Armazena as métricas de cada evento de reserva de veículo. O grão desta tabela é uma reserva individual. Cada linha representa um pedido de reserva feito por um cliente.

Atributo	Tipo	PK/FK	Nulo?	Descrição
SK_TEMPO_SOLICITACAO	INT	FK, PK	Não	Chave para Dim_Tempo, da data em que a reserva foi solicitada.
SK_TEMPO_PREV_RETIRADA	INT	FK	Não	Chave para Dim_Tempo, da data prevista para a retirada do veículo.
SK_TEMPO_PREV_DEVOLUCAO	INT	FK	Não	Chave para Dim_Tempo, da data prevista para a devolução do veículo.
SK_CLIENTE	INT	FK, PK	Não	Chave para Dim_Cliente, do cliente que fez a reserva.
SK_GRUPO_VEICULO	INT	FK, PK	Não	Chave para Dim_GrupoVeiculo, do grupo de veículo reservado.

SK_PATIO_RETIRADA	INT	FK	Não	Chave para Dim_Patio, do pátio previsto para a retirada.
SK_PATIO_DEVOLUCAO	INT	FK	Não	Chave para Dim_Patio, do pátio previsto para a devolução.
ID_RESERVA_OLTP	INT		Não	ID original da reserva no sistema transacional, para rastreabilidade.
QTD_DIAS_PREVISTOS_LOCACAO	INT		Não	Duração prevista da locação em dias.
QTD_RESERVAS	INT		Não	Contador para facilitar a soma de reservas (valor sempre 1).

4. Processo ETL

O processo de ETL (Extração, Transformação e Carga) foi o mecanismo utilizado para popular o Data Warehouse a partir das múltiplas fontes de dados operacionais. O fluxo foi dividido em três fases, operando sobre três ambientes distintos: os Bancos de Dados de Origem (OLTP), a Staging Area (área de preparação) e o Data Warehouse (DW) final.

Para gerenciar a complexidade de integrar sistemas heterogêneos, a primeira decisão de design foi implementar a **Área de Staging** (staging_area). Este é um banco de dados intermediário projetado para receber os dados brutos de todas as fontes sem qualquer tratamento inicial. As tabelas na Staging Area foram criadas sem chaves estrangeiras ou restrições complexas para evitar falhas de carga.

Para garantir a rastreabilidade, duas colunas de metadados foram adicionadas a cada tabela:

1. **FONTE_DADOS**: Uma coluna de texto para identificar de qual empresa (grupo) o registro foi extraído.
2. **DATA_CARGA**: Um timestamp que registra a data e a hora exatas da extração.

O arquivo **staging.sql** está disponível no seguinte link:

https://github.com/gbrods/EEL890_BigData/blob/main/P2_Parte2/%C3%81rea%20Staging%20e%20Processo%20ETL/staging.sql

4.1. Etapa de Extração

A fase de Extração, detalhada no script *extracao_etl.sql*, foi responsável por conectar-se aos múltiplos sistemas transacionais de origem e copiar os dados brutos para a Staging Area. Para cada fonte, foram desenvolvidos scripts *INSERT INTO ... SELECT* que mapeiam as tabelas e colunas originais para as tabelas de staging padronizadas.

O processo de integração abrangeu, além do sistema desenvolvido por nosso grupo, os sistemas de outros três grupos da turma, cada um com sua própria modelagem de banco de dados. As fontes de dados extraídas foram:

- **GRUPO 1:**

Alunos: Fernanda Franco Bottecchia, Jéssica Martins de Oliveira, Kaway Henrique da Rocha Marinho, Rafael Cardim dos Santos, Thiago Dias da Costa.

GitHub: <https://github.com/fernandabottechia/big-data-p2>

- **GRUPO 2:**

Alunos: Alunos: Alice Duarte Faria Ribeiro, Beatriz Farias do Nascimento, Gustavo do Amaral Roxo Pereira.

GitHub: <https://github.com/alicedfr/Big-Data-P2>

- **GRUPO 3:**

Alunos: Guilherme Oliveira Rolim Silva, Ricardo Lorente Kauer, Vinícius Alcântara Gomes Reis de Souza.

GitHub: <https://github.com/rickauer/datawarehouse>

Novamente, um ponto importante nesta etapa foi a adição de uma coluna *FONTE_DADOS* em cada registro para rastrear a origem de cada dado, e uma coluna *DATA_CARGA* com o timestamp da execução. Este processo foi projetado para ser agendado e executado de forma completa e periódica (diariamente), garantindo que o Data Warehouse seja sempre alimentado com as informações mais recentes das operações. O arquivo **extracao_etl.sql** está disponível no seguinte link:

https://github.com/gbrods/EEL890_BigData/blob/main/P2_Parte2/%C3%81rea%20Staging%20e%20Processo%20ETL/extracao_etl.sql

4.2. Etapa de Transformação

A fase de Transformação, executada pelo script *transformacao_etl.sql*, é a etapa onde a inteligência de negócio é aplicada para converter os dados heterogêneos da Staging Area em um conjunto de informações consistentes e de alta qualidade. Nesta fase, foram criadas tabelas transformadas (ex: *stg_dim_cliente*, *stg_fato_locacao*) que unificam os dados de todas as fontes.

Para isso, diversas técnicas foram empregadas, como:

- Limpeza e padronização de campos textuais com *UPPER* e *TRIM*;
- Unificação de registros duplicados através do agrupamento por chaves de negócio (como um CPF/CNPJ padronizado);
- Enriquecimento dos dados, seja pelo cálculo de novas métricas (como a duração de uma locação) ou pela junção de várias tabelas de staging para criar uma visão completa, como no caso da consolidação das informações de veículos, modelos e marcas.

O arquivo **transformacao_etl.sql** está disponível no seguinte link:

https://github.com/gbrods/EEL890_BigData/blob/main/P2_Parte2/%C3%81rea%20Staging%20e%20Processo%20ETL/transformacao_etl.sql

4.3. Etapa de Carga

Finalmente, a etapa de Carga, executada pelo script *carga_etl.sql*, é o passo que move os dados já limpos e integrados da Staging Area para o Data Warehouse de produção (*dw_locadora*). O processo foi implementado como uma carga completa (Full Load), onde as tabelas de fatos e dimensões são reiniciadas com o comando *TRUNCATE* a cada execução, garantindo que o DW reflita o estado mais recente dos dados extraídos.

A carga ocorre em duas fases: primeiro, as tabelas de dimensão são populadas com os registros mestres únicos. Em seguida, as tabelas de fatos são carregadas através de um processo de lookup, onde as chaves de negócio dos eventos transformados são usadas para consultar as dimensões e obter as chaves

substitutas (SK_...) corretas, conectando assim os fatos ao seu respectivo contexto e materializando o modelo estrela.

O arquivo **carga_etl.sql** está disponível no seguinte link:

https://github.com/gbrods/EEL890_BigData/blob/main/P2_Parte2/%C3%81rea%20S tagging%20e%20Processo%20ETL/carga_etl.sql

4.4. Principais desafios

A integração dos múltiplos sistemas revelou desafios práticos que foram solucionados no script de transformação. Alguns exemplos são:

- **Modelos de Clientes Distintos:** Cada grupo modelou clientes de uma forma: um com um campo de documento único, outro com colunas separadas para CPF e CNPJ, e um terceiro com especialização em tabelas *Pessoa_Fisica* e *Pessoa_Juridica*.

Solução: Foi utilizada a função *COALESCE* para unificar as colunas de documento e *LEFT JOINS* para agregar os dados das tabelas especializadas. A limpeza final com *REGEXP_REPLACE* no script de transformação criou uma chave de negócio (Business Key) confiável para cada cliente.

- **Formatos de Endereço Inconsistentes:** Alguns sistemas possuíam endereços totalmente normalizados (com tabelas para rua, bairro, cidade, etc.), enquanto outros continham o endereço completo em um único campo de texto.

Solução: O script de transformação foi projetado para usar *JOINS* para as fontes normalizadas e, para as demais, extrair o campo de texto como está, deixando a possibilidade de parse futuro ou tratando-o como um atributo único.

- **Modelagem de Atributos de Veículos:** A forma de armazenar acessórios variava: colunas booleanas na tabela principal (*possui_ar_condicionado*) ou uma tabela de junção (*VEICULO_ACESSORIO*).

Solução: O script de extração já previa essa diferença, utilizando JOINS ou subconsultas para extrair a informação de forma padronizada para a Staging Area, simplificando a transformação subsequente.

5. Resultados Finais

A implementação do Data Warehouse e do processo ETL se destaca na capacidade de extrair inteligência de negócio através de relatórios e análises antes inviáveis. As consultas SQL desenvolvidas demonstram como o modelo dimensional unificado permite responder a perguntas complexas de forma eficiente. Foram criados relatórios para o controle de locações por grupo, análise de demanda futura de reservas por pátio e perfil de cliente, e o ranking de grupos de veículos mais alugados. Os respectivos links dos scripts são:

- **Controle de pátio:**
https://github.com/gbrods/EEL890_BigData/blob/main/P2_Parte2/Relat%C3%B3rios%20Gerenciais%20%2B%20Markov/controle_patio.sql
- **Controle das locações:**
https://github.com/gbrods/EEL890_BigData/blob/main/P2_Parte2/Relat%C3%B3rios%20Gerenciais%20%2B%20Markov/controle_locacoes.sql
- **Controle de reservas:**
https://github.com/gbrods/EEL890_BigData/blob/main/P2_Parte2/Relat%C3%B3rios%20Gerenciais%20%2B%20Markov/controle_reservas.sql
- **“Grupos” de veículos mais alugados:**
[https://github.com/gbrods/EEL890_BigData/blob/main/P2_Parte2/Relat%C3%B3rios%20Gerenciais%20%2B%20Markov/grupos_veiculos_mais_alugados.s
ql](https://github.com/gbrods/EEL890_BigData/blob/main/P2_Parte2/Relat%C3%B3rios%20Gerenciais%20%2B%20Markov/grupos_veiculos_mais_alugados.sql)

Destaca-se também a análise de controle de pátio, que detalha as locações por origem da frota (própria ou parceira), e, principalmente, a geração da matriz de movimentação entre pátios, que serve como base para a modelagem de previsão de ocupação por cadeias de Markov. Veremos mais sobre essa questão no próximo tópico.

6. Matriz de Movimentação para Cadeias de Markov

Além de viabilizar relatórios gerenciais sobre o estado passado e presente do negócio, um dos objetivos estratégicos do Data Warehouse é fornecer uma base de dados para análises preditivas, especificamente para a previsão de ocupação de pátio utilizando Cadeias de Markov.

Para construir um modelo de Markov, é necessária uma matriz de transição estocástica. Em nosso contexto, isso se traduz em conhecer a probabilidade de um veículo, alugado em um Pátio A, ser devolvido no mesmo Pátio A, ou no Pátio B, Pátio C, e assim por diante. Obter esses dados históricos de forma consolidada e confiável a partir dos sistemas OLTP isolados seria uma tarefa complexa e manual.

Nesse sentido, o modelo dimensional implementado, em particular a tabela *Fato_Locacao*, foi projetado para resolver este problema de forma nativa. Ao registrar cada locação com chaves para o pátio de retirada (*SK_PATIO_RETIRADA*) e o pátio de devolução (*SK_PATIO_DEVOLUCAO*), o DW criou um repositório histórico completo de toda a movimentação da frota entre os pátios.

Para extrair os dados necessários para o modelo, foi construído o script *previsao_ocupacao_patio_markov.sql*. Utilizando uma consulta com Funções de Janela (*SUM(...) OVER (PARTITION BY ...)*), o script calcula:

- O número total de viagens de cada pátio de origem para cada pátio de destino;
- O percentual que cada rota representa em relação ao total de saídas de seu pátio de origem.

Por fim, o resultado é uma matriz de probabilidades de transição que serve como input direto para a modelagem por Cadeias de Markov, permitindo à empresa prever com maior acurácia as futuras necessidades de vagas e a distribuição da frota, otimizando assim a logística e a disponibilidade de veículos.

O arquivo ***previsao_ocupacao_patio_markov.sql*** está disponível no seguinte link:

https://github.com/gbrods/EEL890_BigData/blob/main/P2_Parte2/Relat%C3%B3rios%20Gerenciais%20%2B%20Markov/previsao_ocupacao_patio_markov.sql

7. Conclusão

O projeto demonstrou com sucesso a viabilidade e a eficácia de uma solução de Data Warehouse para o desafio de integração de dados enfrentado pelas empresas independentes de locação de veículos. Através do desenho de um modelo dimensional em constelação de estrelas e da construção dos scripts que compõem um processo ETL completo (da extração de fontes heterogêneas à transformação e carga dos dados), foi estabelecida uma arquitetura teórica robusta e funcional. Além disso, os scripts de consulta desenvolvidos para os relatórios e para a matriz de Markov comprovam que o modelo proposto é capaz de atender a todas as necessidades analíticas especificadas, desde o controle operacional até a análise preditiva.

Conclui-se, portanto, que o trabalho não apenas cumpriu os requisitos técnicos da modelagem, mas também entregou uma arquitetura de referência completa que, se implementada, forneceria à gestão uma base de dados sólida e unificada, essencial para uma tomada de decisão mais informada e ágil.