

# MODELAGEM DIMENSIONAL

Livya Gomes de Carvalho - 01645272  
Engenharia da Computação

## SUMÁRIO

<b>1 STAR SCHEMA.....</b>	<b>1</b>
1.1 O que é.....	1
1.2 Principais características.....	1
1.3 Casos de uso.....	1
<b>2 SNOWFLAKE SCHEMA.....</b>	<b>2</b>
2.1 O que é.....	2
2.2 Principais Características.....	2
2.3 Casos de uso.....	2
<b>3 PRINCIPAIS DIFERENÇAS ENTRE O SISTEMAS STAR E SNOWFLAKE.....</b>	<b>3</b>
3.1 Arquitetura.....	3
3.2 Desempenho.....	3
3.3 Manutenção.....	3
<b>4 IMPLEMENTAÇÃO DE UM EXEMPLO DE MODELO STAR SCHEMA.....</b>	<b>4</b>
4.1 Modelo Lógico.....	4
4.2 Modelo Físico.....	5
<b>5 CUBOS OLAP.....</b>	<b>6</b>
5.1 Definições.....	6
5.2 Arquitetura.....	6
5.2.1 Fonte de Dados.....	6
5.2.2 Dimensões.....	6
5.2.3 Fatos ou Medidas.....	7
5.2.4 Ferramentas de Consulta e Visualização.....	7
5.3 Implementação.....	7
5.3.1 Casos de uso do Cubo OLAP.....	8
<b>6 IMPLEMENTAÇÃO DE UM EXEMPLO DE CUBO OLAP NO MYSQL.....</b>	<b>9</b>
6.1 Dicionário de dados.....	10

# 1 STAR SCHEMA

## 1.1 O que é

Esquema em Estrela é um tipo de esquema de data warehouse e consiste em uma tabela central chamada Tabela de fatos contém referências diretas de chaves estrangeiras a múltiplas tabelas de dimensões. Esse design é chamado de “Star” porque a tabela de fatos está no centro, com tabelas de dimensão que se estendem dela como raios de uma estrela e que não fazem ligações com outras tabelas de dimensão.

## 1.2 Principais características

- Tabela de fatos central - representa atividades, eventos e transações comerciais;
- Tabela de dimensões - representa aspectos específicos e contém atributos descritivos;
- Não há conexão entre tabelas de dimensões;
- Estrutura desnormalizada;
- Dimensionalidade fixa.

## 1.3 Casos de uso

Análise de Negócios: É ideal para ambientes de data warehousing onde a análise de negócios é o principal objetivo, pois o modelo Star fornece uma estrutura simplificada e otimizada para consultas analíticas. Uma empresa de varejo pode usar o star schema para modelar dados de vendas.

Consultas de análises: É altamente otimizado e rápido para consultas de análise que envolvem agregação de dados. Isso ocorre porque as consultas normalmente envolvem menos junções em comparação com esquemas mais normalizados. Instituições educacionais podem usar o star schema para analisar dados de alunos, cursos, instrutores e tempo, permitindo relatórios rápidos sobre desempenho acadêmico, frequência e eficácia dos cursos.

Dashboards e Visualizações: Quando há necessidade de alimentar dashboards e visualizações interativas que requerem respostas rápidas e acessos frequentes aos dados agregados.

## 2 SNOWFLAKE SCHEMA

### 2.1 O que é

O esquema de floco de neve é uma extensão do modelo de esquema em estrela. Nele, há uma tabela de fatos central que também está diretamente ligada a tabelas de dimensões, porém essas tabelas estão normalizadas em subdimensões que contêm atributos específicos de uma dimensão. Essa normalização resulta em uma estrutura de banco de dados semelhante a um floco de neve.

### 2.2 Principais Características

- Normalização - redução de redundância e melhor integridade dos dados.
- Relacionamento entre tabelas - possui relacionamentos de junção adicionais entre as tabelas normalizadas que aumentam a complexidade das consultas.
- Integridade de dados - o esquema Snowflake reduz a redundância e eliminam anomalias, o que garante que os dados sejam armazenados de maneira consistente.
- Flexibilidade - fornece flexibilidade na organização e gerenciamento de dados, o que pode ser útil em determinados cenários de modelagem de dados.

### 2.3 Casos de uso

Grande Volume de Dados: a estrutura normalizada de um esquema Snowflake pode ajudar a otimizar a eficiência do armazenamento. O Snowflake pode ser uma opção mais econômica para gerenciar grandes volumes de dados.

Hierarquia de negócios: a hierarquia dos negócios e suas dimensões são preservadas no modelo de dados por meio do relacionamento de chave externa e chave primária entre as várias tabelas de dimensões. Uma empresa de e-commerce pode usar o snowflake schema para organizar dados de vendas. As tabelas de produtos, categorias e subcategorias podem ser separadas, permitindo uma análise detalhada de desempenho de vendas em diferentes níveis de produto.

Cenários de Relatórios Específicos: para relatórios que exigem análises detalhadas e granularidade nas dimensões, permitindo uma análise mais profunda dos dados. Empresas de telecom podem usar o snowflake schema para gerenciar dados de clientes, planos de serviço, chamadas e tarifas em várias tabelas relacionadas, permitindo análises detalhadas sobre o uso e comportamento dos clientes.

### **3 PRINCIPAIS DIFERENÇAS ENTRE O SISTEMAS STAR E SNOWFLAKE**

Ambos os esquemas têm seus usos e vantagens, e a escolha entre eles geralmente depende das necessidades específicas do projeto, como requisitos de consulta, volume de dados e preferências de modelagem.

#### **3.1 Arquitetura**

- Star - os dados são estruturados em torno de uma tabela de fatos central, com tabelas de dimensão que contêm informações adicionais. Esse modelo usa uma arquitetura desnormalizada, resultando em consultas mais rápidas, mas potencialmente mais redundância de dados.
- Snowflake - emprega uma abordagem de estrela expandida, onde as dimensões são normalizadas em várias tabelas. Isso reduz a redundância de dados, mas pode tornar as consultas mais lentas devido à necessidade de realizar mais operações de junção.

#### **3.2 Desempenho**

- Star - tende a ser mais rápido para consultas que envolvem a tabela de fatos e algumas dimensões relacionadas.
- Snowflake - pode ser mais escalável para ambientes com grandes volumes de dados e consultas complexas.

#### **3.3 Manutenção**

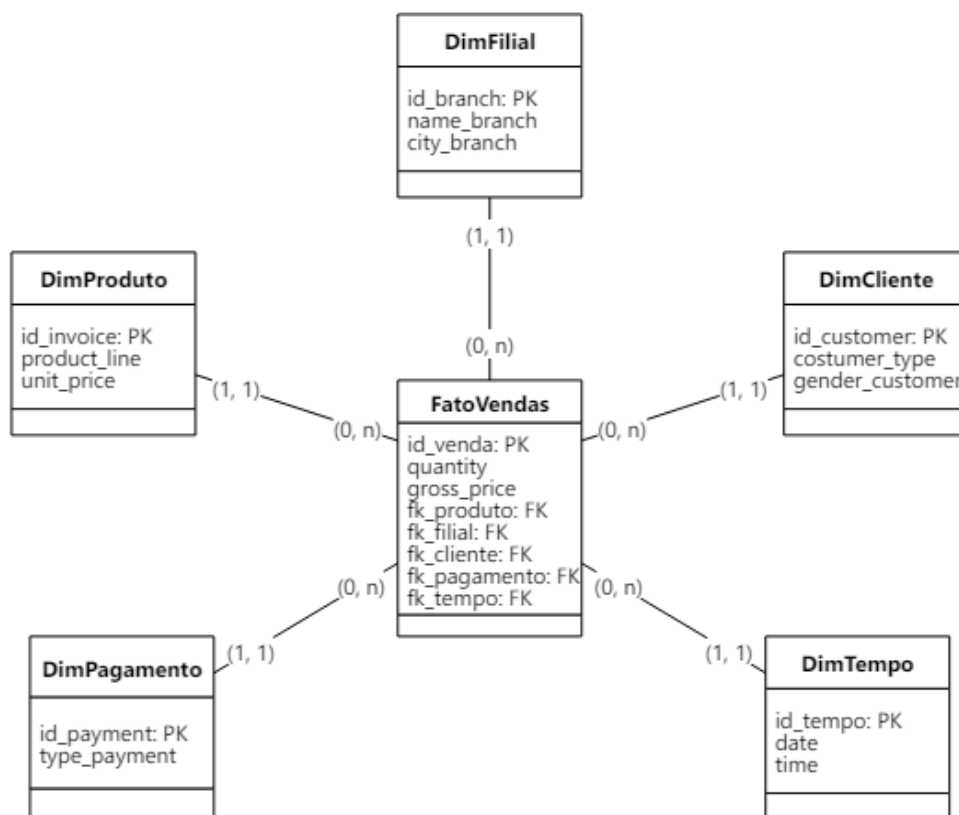
- Star - mais fácil de entender e manter, especialmente para usuários menos experientes.
- Snowflake - oferece maior flexibilidade e capacidade de expansão, sendo mais adequado para ambientes com requisitos de dados sujeitos a mudanças frequentes.

Por fim, enquanto o esquema estrela é mais adequado para consultas analíticas simples e diretas, o esquema floco de neve é mais flexível e eficaz em ambientes onde a normalização dos dados e a integridade são prioridades. A escolha entre os dois depende das necessidades específicas do projeto e das características dos dados envolvidos.

## 4 IMPLEMENTAÇÃO DE UM EXEMPLO DE MODELO STAR SCHEMA

Utilizando como base um dataset sobre [Vendas de um Mercado](#), que pode ser encontrado na plataforma Kaggle, é possível criar um modelo Star Schema como o que está representado abaixo. Após a escolha do dataset, foi feita a identificação da tabela fato, das tabelas dimensões e seus respectivos atributos. Em seguida, foi realizada a construção dos modelos lógico e físico com as devidas relações entre as tabelas.

### 4.1 Modelo Lógico



## 4.2 Modelo Físico

```
CREATE DATABASE mercado;
USE mercado;

-- Criação das tabelas

CREATE TABLE DimProduto (
  id_invoice INT PRIMARY KEY,
  product_line VARCHAR(30) NOT NULL,
  unit_price FLOAT NOT NULL
);

CREATE TABLE DimFilial (
  id_branch INT AUTO_INCREMENT PRIMARY KEY,
  name_branch VARCHAR(5) NOT NULL,
  city_branch VARCHAR(10) NOT NULL
);

CREATE TABLE DimCliente (
  id_customer INT AUTO_INCREMENT PRIMARY KEY,
  costumer_type VARCHAR(10) NOT NULL,
  gender_costumer VARCHAR(10) NOT NULL
);

CREATE TABLE DimPagamento (
  id_payment INT AUTO_INCREMENT PRIMARY KEY,
  type_payment VARCHAR(20) NOT NULL
);

CREATE TABLE DimTempo (
  id_tempo INT AUTO_INCREMENT PRIMARY KEY,
  date DATE NOT NULL,
  time TIME NOT NULL
);

CREATE TABLE FatoVendas (
  id_venda INT PRIMARY KEY,
  quantity INT NOT NULL,
  gross_price FLOAT NOT NULL,
  fk_produto INT NOT NULL,
  fk_filial INT NOT NULL,
  fk_cliente INT NOT NULL,
  fk_pagamento INT NOT NULL,
  fk_tempo INT NOT NULL,
  FOREIGN KEY (fk_produto) REFERENCES DimProduto(id_invoice),
  FOREIGN KEY (fk_filial) REFERENCES DimFilial(id_branch),
  FOREIGN KEY (fk_cliente) REFERENCES DimCliente(id_customer),
  FOREIGN KEY (fk_pagamento) REFERENCES DimPagamento(id_payment),
  FOREIGN KEY (fk_tempo) REFERENCES DimTempo(id_tempo)
);
```

## **5 CUBOS OLAP**

### **5.1 Definições**

Cubos OLAP, também conhecidos como cubos multidimensionais ou hipercubos, são estruturas de dados fundamentais no mundo da análise de dados. Essas estruturas organizam informações de maneira multidimensional, tornando-as facilmente analisáveis a partir de diferentes perspectivas. Contendo medidas (como vendas, lucros, etc.) e dimensões (como tempo, produto, localização, etc.), os cubos OLAP permitem aos usuários explorar e analisar dados de forma eficiente. Operações como drill-down, roll-up, slice, dice e pivot facilitam a exploração dos dados.

Os cubos OLAP são construídos utilizando bancos de dados OLAP, o que permite uma análise quase instantânea dos dados. Essas estruturas superam as limitações dos bancos de dados relacionais, proporcionando uma análise rápida e eficaz. Os bancos de dados OLAP são especializados em extrair informações de business intelligence dos dados, permitindo aos usuários acesso pesquisável a grandes quantidades de dados e a capacidade de acumular, dividir e cortar os dados conforme necessário para responder a uma variedade de perguntas relevantes para suas áreas de interesse.

### **5.2 Arquitetura**

Um cubo OLAP é uma ferramenta poderosa para análise de dados multidimensionais, e entender sua arquitetura é crucial para sua implementação eficaz. A arquitetura do cubo OLAP proporciona uma estrutura eficiente e flexível para análise de dados multidimensionais em tempo real, permitindo que os usuários explorem e analisem grandes volumes de dados de forma intuitiva. Principais componentes da arquitetura do cubo OLAP:

#### **5.2.1 Fonte de Dados**

É a origem dos dados contidos no cubo OLAP, que lê e processa dados brutos a fim de executar cálculos e agregações. Os dados são pré-agregados e armazenados nos cubos para melhorar o desempenho das consultas. Isso significa que as operações analíticas podem ser executadas rapidamente, pois os resultados já estão calculados e armazenados.

#### **5.2.2 Dimensões**

As dimensões permitem filtrar, agrupar e rotular os dados. Elas representam as características pelas quais os dados são analisados. Por exemplo, em um cubo de vendas, as dimensões podem incluir tempo, produto,



localização e cliente. Cada dimensão pode ter várias hierarquias, permitindo uma hierarquia mais detalhada.

### **5.2.3 Fatos ou Medidas**

São valores numéricos que os usuários desejam analisar, segmentar e agregar, geralmente mapeados para colunas numéricas em uma tabela de fatos, mas também podem ser criadas em atributos de dimensão. O desenvolvimento de cubos OLAP envolve determinar e definir quais medidas serão exibidas e como serão calculadas, pois são os valores mais importantes para análise e o principal interesse dos usuários finais. Em um cubo de vendas, os fatos ou medidas podem incluir o total de vendas, quantidades de produtos vendidos e receita.

### **5.2.4 Ferramentas de Consulta e Visualização**

Os cubos OLAP são acessados por ferramentas de BI que permitem aos usuários consultar e visualizar os dados de forma interativa. Isso inclui dashboards, relatórios e ferramentas de análise.

## **5.3 Implementação**

Um cubo OLAP pode ser implementado em uma variedade de casos onde a análise multidimensional dos dados é essencial. A sua implementação envolve várias etapas:

- Modelagem Dimensional - Identificar as dimensões relevantes para a análise e projetar um esquema dimensional. Isso envolve a criação de tabelas de dimensão e tabelas de fatos;
- Extração, Transformação e Carregamento (ETL) - Extrair dados das fontes operacionais, transformá-los em um formato adequado para análise e carregá-los no local desejado;
- Construção do Cubo - Usando ferramentas OLAP, como Microsoft Analysis Services ou Oracle OLAP, construir o cubo com base nos dados. Isso envolve a definição de hierarquias de dimensões, medidas e agregações;
- Publicação e Acesso - Após a construção do cubo, ele é publicado no servidor OLAP para que os usuários finais possam acessá-lo. Os usuários podem então usar ferramentas de análise, como Microsoft Excel ou Tableau, para explorar os dados do cubo e realizar análises multidimensionais.

### 5.3.1 Casos de uso do Cubo OLAP

- Análise de Vendas - pode ser usado para analisar vendas por produto, região, período de tempo e outras dimensões relevantes. Isso permite aos usuários explorar os dados de vendas de várias maneiras para identificar tendências, padrões e oportunidades.
- Gestão de Recursos Humanos - para empresas com uma grande força de trabalho, pode ser usado para analisar dados relacionados a funcionários, como desempenho, histórico de emprego, treinamento, benefícios, etc. Isso pode ajudar na tomada de decisões relacionadas ao RH, como alocação de recursos, planejamento de sucessão e desenvolvimento de talentos.
- Análise Financeira - pode ser usado para analisar dados contábeis, como receitas, despesas, lucros e fluxo de caixa. Os usuários podem explorar esses dados de várias perspectivas, como por região, segmento de mercado ou linha de produtos, para obter insights sobre o desempenho financeiro da organização.
- Gestão de Estoques e Cadeia de Suprimentos - para empresas que lidam com estoques e cadeias de suprimentos complexas, pode ser usado para analisar dados relacionados a inventário, pedidos, fornecedores, tempo de entrega, etc. Isso pode ajudar na otimização de estoques, previsão de demanda e gestão de fornecedores.
- Análise de Marketing - pode ser usado para analisar dados de marketing, como campanhas publicitárias, desempenho de canais de marketing, engajamento do cliente e retorno sobre o investimento em marketing. Isso pode ajudar os profissionais de marketing a entender melhor o impacto de suas atividades e alocar recursos de forma mais eficaz.

## 6 IMPLEMENTAÇÃO DE UM EXEMPLO DE CUBO OLAP NO MYSQL

A partir de um dataset sobre [Vendas de um Mercado](#), já citado anteriormente, em formato .CSV e inserido no MySQL Workbench, é possível fazer DQLs que permitem ao MySQL simular um cubo OLAP.

- Consulta:

```
-- Cria views de cubos OLAP
CREATE VIEW Cubo_Mercado AS
SELECT
    `Invoice ID`,
    `Product line`,
    `Unit price`,
    `Branch`,
    `Customer type`,
    `Gender`,
    `Payment`,
    Date,
    Time,
    SUM(Quantity) AS Total_Quantity,
    SUM(cogs) AS Total_Cogs
FROM
    `supermarket_sales - sheet1`
GROUP BY
    `Invoice ID`,
    `Product line`,
    `Unit price`,
    `Branch`,
    `Customer type`,
    Gender,
    Payment,
    Date,
    Time;

SELECT * FROM Cubo_MERCADO;
```

- Resultado:

83 • **SELECT \* FROM Cubo\_MERCADO;**

84

result Grid    Filter Rows:   Exports:  Wrap Cell Contents:  Fetch rows:										
Invoice ID	Product line	Unit price	Branch	Customer type	Gender	Payment	Date	Time	Total_Quantity	
101-17-6199	Food and beverages	45.79	A	Normal	Male	Credit card	3/13/2019	19:44	7	3
101-81-4070	Health and beauty	62.82	C	Member	Female	Ewallet	1/17/2019	12:36	2	1
102-06-2002	Sports and travel	25.25	C	Member	Male	Cash	3/20/2019	17:52	5	1
102-77-2261	Health and beauty	65.31	C	Member	Male	Credit card	3/5/2019	18:02	7	4

- Consulta:

```
-- Consulta para obter a soma total de cogs por product line e branch
SELECT
    `Product line`,
    `Branch`,
    SUM(Total_Cogs) AS Total_Cogs_Por_Product_Line_Branch
FROM
    Cubo_Mercado
GROUP BY
    `Product line`,
    `Branch`;
```

- Resultado:

```
--
85 -- Consulta para obter a soma total de cogs por product line e branch
86 SELECT
87     `Product line`,
88     `Branch`,
89     SUM(Total_Cogs) AS Total_Cogs_Por_Product_Line_Branch
```

Product line	Branch	Total_Cogs_Por_Product_Line_Branch
Electronic accessories	A	17444.87
Electronic accessories	B	16239.469999999998
Electronic accessories	C	18065.690000000002
Fashion accessories	A	15554.769999999997
Fashion accessories	B	15631.730000000003
Fashion accessories	C	20533.399999999998

## 6.1 Dicionário de dados

Coluna	Significado
Invoice ID	Código do produto
Branch	Filial (A, B, C)
City	Cidade
Gender	Gênero do comprador
Product Line	Linha do produto
Unit Price	Preço unitário
Quantity	Quantidade
Tax 5%	Taxa de 5% na compra
Total	Total da compra
Date	Data da compra
Time	Horário da compra
Payment	Tipo do pagamento
Cogs	Custo bruto no valor total da venda
Gross margin percentage	Porcentagem em cima do valor da venda
Gross income	Lucro Bruto
Rating	Nota do cliente em relação a compra (1 até 10)