

# Cubos OLAP: Definição, Arquitetura e Implementação

## Definição

Cubos OLAP (Online Analytical Processing) são estruturas de dados que permitem análises multidimensionais rápidas e complexas de grandes volumes de dados. Eles são usados principalmente em sistemas de Business Intelligence (BI) para facilitar a análise de dados, como vendas, desempenho financeiro, inventário, etc.

Os cubos OLAP são organizados em dimensões e fatos:

Dimensões: Aspectos pelos quais os dados podem ser categorizados (e.g., tempo, localização, produto).

Fatos: Dados quantitativos que podem ser agregados (e.g., vendas, lucros).

## Arquitetura

A arquitetura de um cubo OLAP geralmente segue uma estrutura de três níveis:

Fonte de Dados: Base de dados transacionais ou sistemas de dados externos.

Processo ETL (Extract, Transform, Load): Extrai dados das fontes, transforma-os em um formato adequado e carrega-os no data warehouse.

Data Warehouse: Armazena dados de forma estruturada, organizados em esquemas (estrela ou floco de neve) para facilitar o acesso e a análise.

OLAP Server: Processa consultas OLAP, calcula agregações e fornece acesso rápido aos dados.

Cliente OLAP: Interfaces de usuário, ferramentas de BI que permitem aos usuários finais interagir com os dados.

## Implementação

Modelagem Dimensional:

Identifique as dimensões e fatos.

Crie o esquema estrela ou floco de neve.

Criação do Cubo OLAP:

Utilize uma ferramenta OLAP ou biblioteca que suporte a criação e manipulação de cubos OLAP.

Exemplos incluem Microsoft SQL Server Analysis Services (SSAS), Pentaho, Apache Kylin.

## Exemplo de Cubo OLAP Implementado com MySQL

Embora MySQL não ofereça suporte nativo para OLAP como alguns bancos de dados especializados, é possível criar um sistema OLAP básico usando tabelas e consultas SQL.

### 1. Estrutura de Tabelas

Tabela de Fatos (fatos\_vendas):

```
CREATE TABLE fatos_vendas (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    data_venda DATE,  
    id_produto INT,  
    id_cliente INT,  
    id_localizacao INT,  
    quantidade INT,  
    valor DECIMAL(10,2)  
);
```

Tabelas de Dimensões:

Dimensão Tempo (dim\_tempo):

```
CREATE TABLE dim_tempo (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    data DATE,  
    ano INT,  
    mes INT,  
    dia INT,  
    trimestre INT  
);
```

Dimensão Produto (dim\_produto):

```
CREATE TABLE dim_produto (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nome_produto VARCHAR(255),  
    categoria VARCHAR(255)  
);
```

Dimensão Cliente (dim\_cliente):

```
CREATE TABLE dim_cliente (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nome_cliente VARCHAR(255),  
    pais VARCHAR(255),  
    cidade VARCHAR(255)  
);
```

Dimensão Localização (dim\_localizacao):

```
CREATE TABLE dim_localizacao (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    pais VARCHAR(255),  
    estado VARCHAR(255),  
    cidade VARCHAR(255)  
);
```

## 2. Popular as Tabelas

Insira dados nas tabelas de dimensões e fatos:

```
INSERT INTO dim_tempo (data, ano, mes, dia, trimestre)  
VALUES ('2023-05-27', 2023, 5, 27, 2);  
  
INSERT INTO dim_produto (nome_produto, categoria)  
VALUES ('Produto A', 'Categoria 1');  
  
INSERT INTO dim_cliente (nome_cliente, pais, cidade)  
VALUES ('Cliente 1', 'Brasil', 'São Paulo');  
  
INSERT INTO dim_localizacao (pais, estado, cidade)  
VALUES ('Brasil', 'SP', 'São Paulo');  
  
INSERT INTO fatos_vendas (data_venda, id_produto, id_cliente, id_localizacao, quantidade, valor)  
VALUES ('2023-05-27', 1, 1, 1, 10, 100.00);
```

### 3. Consultas OLAP

Consulta de Agregação por Mês e Produto:

```
SELECT
    dt.mes,
    dp.nome_produto,
    SUM(fv.quantidade) AS total_quantidade,
    SUM(fv.valor) AS total_valor
FROM
    fatos_vendas fv
JOIN
    dim_tempo dt ON fv.data_venda = dt.data
JOIN
    dim_produto dp ON fv.id_produto = dp.id
GROUP BY
    dt.mes, dp.nome_produto;
```

Consulta de Agregação por Ano e Localização:

```
SELECT
    dt.ano,
    dl.cidade,
    SUM(fv.quantidade) AS total_quantidade,
    SUM(fv.valor) AS total_valor
FROM
    fatos_vendas fv
JOIN
    dim_tempo dt ON fv.data_venda = dt.data
JOIN
    dim_localizacao dl ON fv.id_localizacao = dl.id
GROUP BY
    dt.ano, dl.cidade;
```

### Conclusão

Embora MySQL não seja a escolha ideal para operações OLAP complexas, é possível criar uma solução básica utilizando tabelas relacionais e consultas SQL para realizar análises multidimensionais. Para necessidades OLAP mais avançadas, ferramentas dedicadas como Microsoft SSAS, Apache Kylin ou outras plataformas de BI são recomendadas.