

UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

ANÁLISE DE DADOS

---

**Planeamento e Execução de Projetos de Sistemas de Bases  
de Dados Multidimensionais**

---

Bárbara Cardoso (A80453)

João Vilaça (A82339)

5 de Março de 2021

# 1 Resumo

Atualmente, a manipulação e exploração de dados é uma das mais importantes metodologias à disposição das empresas para poderem suportar decisões e melhorar os processo de negócio.

Os agentes de decisão empresariais procuram agora ferramentas com capacidade de manipular e apresentar dados nas várias perspectivas de negócio, com análise atômica e/ou agregada de espaços multidimensionais.

Neste âmbito, destacam-se as ferramentas de Business Intelligence e Analytics que permitem a automatização e otimização dos processos de análise de negócio e, consequentemente, das ações de tomada de decisão.

# Conteúdo

<b>1</b>	<b>Resumo</b>	<b>1</b>
<b>2</b>	<b>Introdução</b>	<b>3</b>
<b>3</b>	<b>Configuração Inicial</b>	<b>4</b>
<b>4</b>	<b>Arquitetura</b>	<b>5</b>
4.1	Four-Step Dimensional Design Process . . . . .	5
4.1.1	Passo 1: Selecionar Processo de Negócio . . . . .	5
4.1.2	Passo 2: Definir a granularidade . . . . .	5
4.1.3	Passo 3: Identificar as dimensões . . . . .	6
4.1.4	Passo 4: Identificar os factos . . . . .	7
4.2	Criar schema . . . . .	8
<b>5</b>	<b>ETL - Extract, Transformation, and Load System</b>	<b>9</b>
5.1	Desenvolver um plano para ETL . . . . .	9
5.1.1	Passo 1: Concepção de um plano de alto nível . . . . .	9
5.1.2	Passo 2: Escolha de uma ferramenta de ETL . . . . .	9
5.1.3	Passo 3: Aplicação de estratégias . . . . .	9
5.1.4	Passo 4: Drill Down por Target Table . . . . .	10
5.2	Execução da fase ETL . . . . .	10
5.2.1	Extração de dados . . . . .	10
5.2.2	Processar dimensões, factos e carregamento de dados . . . . .	12
<b>6</b>	<b>Business Intelligence Applications</b>	<b>13</b>
<b>7</b>	<b>Conclusão</b>	<b>16</b>
<b>8</b>	<b>Bibliografia</b>	<b>17</b>
<b>9</b>	<b>Anexos</b>	<b>18</b>
9.1	Dimensões . . . . .	18
9.1.1	Data . . . . .	18
9.1.2	Produto . . . . .	18
9.1.3	Expedidor . . . . .	18
9.1.4	Métodos de Pagamento . . . . .	19
9.1.5	Clientes . . . . .	19
9.1.6	Colaboradores . . . . .	19
9.2	Tabela de Factos de Encomendas . . . . .	20
9.3	Índices . . . . .	21
9.3.1	Criar . . . . .	21
9.3.2	Apagar . . . . .	21

## 2 Introdução

Ao longo deste projeto, pretende-se estudar e aplicar técnicas de business intelligence e analytics procurando descobrir padrões de dados e seus relacionamentos entre as várias entidades que compõe o sistema no processo de negócio da empresa *Northwind Traders*, cuja base de dados captura todas as transações de vendas que ocorrem entre a empresa, ou seja, os comerciantes da *Northwind* e seus clientes, bem como as transações de compra entre a *Northwind* e seus fornecedores.

Com o desenvolvimento desta plataforma de suporte à decisão, procura-se explicar os resultados de vendas da empresa, fazer previsões de resultados futuros para determinadas variações dos elementos que constituem o sistema, como por exemplo, a aplicação um desconto sobre um determinado produto.

### 3 Configuração Inicial

Em primeiro lugar, para simular um ambiente de produção sobre o qual serão recolhidos os dados para a alimentação da Data Warehouse, foi lançado uma instância do MySQL em Docker.

Foi então efetuado o download dos modelos e das scripts disponibilizadas em [GitHub/MyWind](#) para recriar dados de um negócio. Em seguida, recorrendo ao MySQL Workbench foi criado na base de dados o schema do 'northwind' com o ficheiro 'northwind-default-current-timestamp.sql'. Por fim, foram carregados os dados da 'northwind-data.sql'.

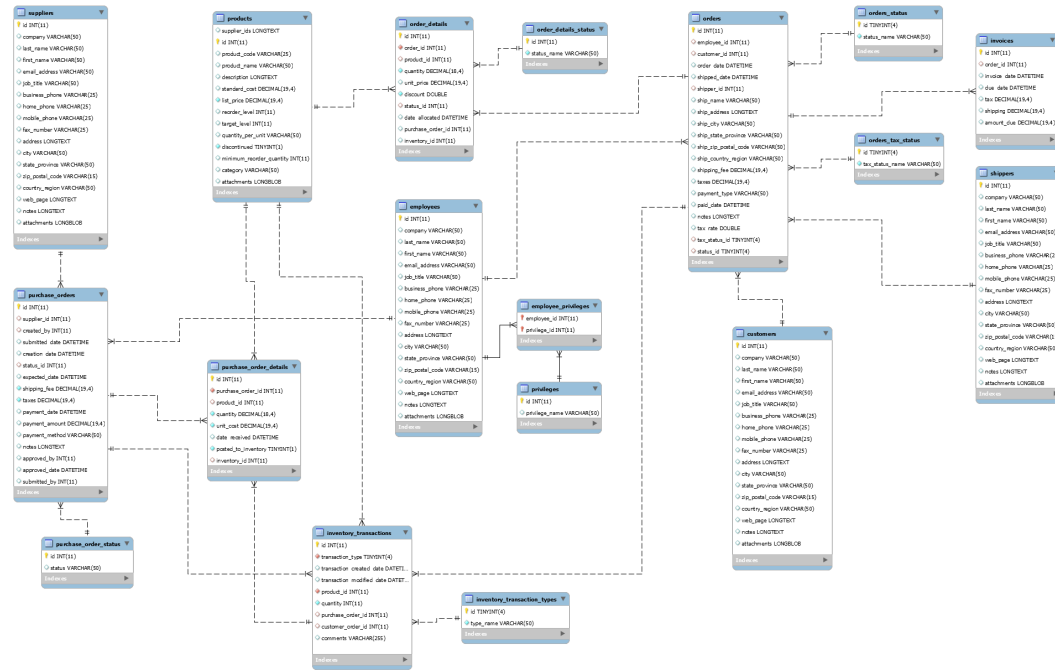


Figura 1: Modelo ERD da Base de Dados

## 4 Arquitetura

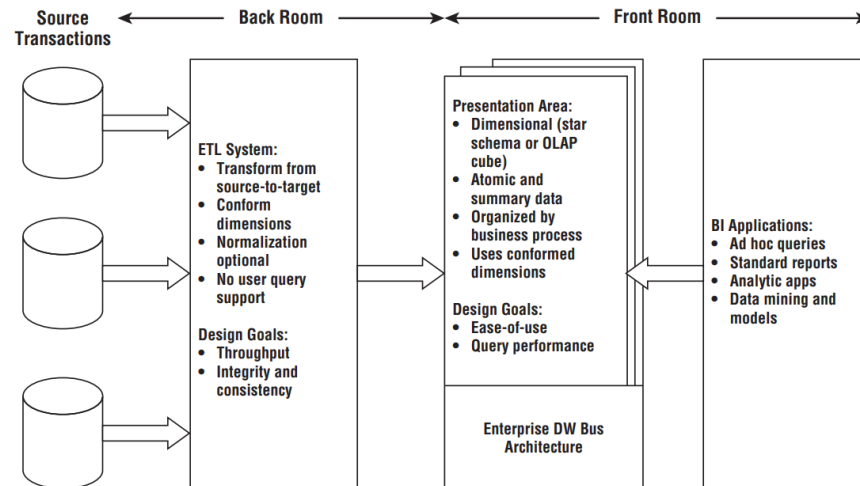


Figura 2: Arquitetura DW/BI seguida (fonte: The Data Warehouse Toolkit, Ralph Kimball)

### 4.1 Four-Step Dimensional Design Process

#### 4.1.1 Passo 1: Selecionar Processo de Negócio

O primeiro projeto do Data Warehouse deve ter como foco o processo de negócio mais crítico e útil para o cliente do DW que pretende ter maior introspeção sobre os aspectos mais importantes do negócio.

Neste caso de estudo, imaginemos que a gerência deseja entender melhor as compras dos clientes conforme capturadas pelo sistema. Portanto, o processo de negócios a modelar será as **encomendas**.

Esses dados permitem que os clientes do DW analisem quais os produtos mais vendidos, as condições promocionais em que foram adquiridos e os detalhes dos clientes.

#### 4.1.2 Passo 2: Definir a granularidade

No contexto de encomendas, e como temos uma baixa quantidade de dados, é preferível escolher o nível mais atômico possível. Esta escolha do uso de dados atômicos fornece a maior flexibilidade analítica possível porque, deste modo, os dados do Data Warehouse podem ser limitados e acumulados livremente e permite que as consultas detalhem os dados de maneiras muito precisas.

#### 4.1.3 Passo 3: Identificar as dimensões

Depois de definida a granularidade, é fácil escolher quais as dimensões que fazem sentido analisar. Aplicam-se então a 'data', 'produto', 'colaborador', 'expedidor', 'método de pagamento' e 'cliente'.

Date Dimension
Date Key
Date
Full Date Description
Day of Week
Calendar Month
Calendar Quarter
Calendar Year
Holiday Indicator
Weekday Indicator

(a)

Product Dimension
Product Key
Product Code
Product Name
Description
Quantity per Unit
Discontinued
Category

(b)

Payment Method Dimension
Payment Method Key
Payment type
Paid date

(c)

Employee Dimension
Employee Key
Company
Product Name
Last Name
First Name
Email address
Job title
Business phone
Home phone
Mobile phone
Fax number
Address
City
State province
Zip postal code
Country region
Web page
Notes
Active

(d)

Shipper Dimension
Shipper Key
Company
Last Name
First Name
Email address
Job title
Business phone
Home phone
Mobile phone
Fax number
Address
City
State province
Zip postal code
Country region
Web page
Notes
Active

(e)

Customer Dimension
Customer Key
Company
Last Name
First Name
Email address
Job title
Business phone
Home phone
Mobile phone
Fax number
Address
City
State province
Zip postal code
Country region
Web page
Notes
Active
Ship address
Ship city
Ship state province
Ship zip postal code
Ship country region

(f)

## Detalhes das Tabelas de Dimensões

### 4.1.4 Passo 4: Identificar os factos

Por fim, resta apenas identificar os factos que farão parte da tabela de factos.

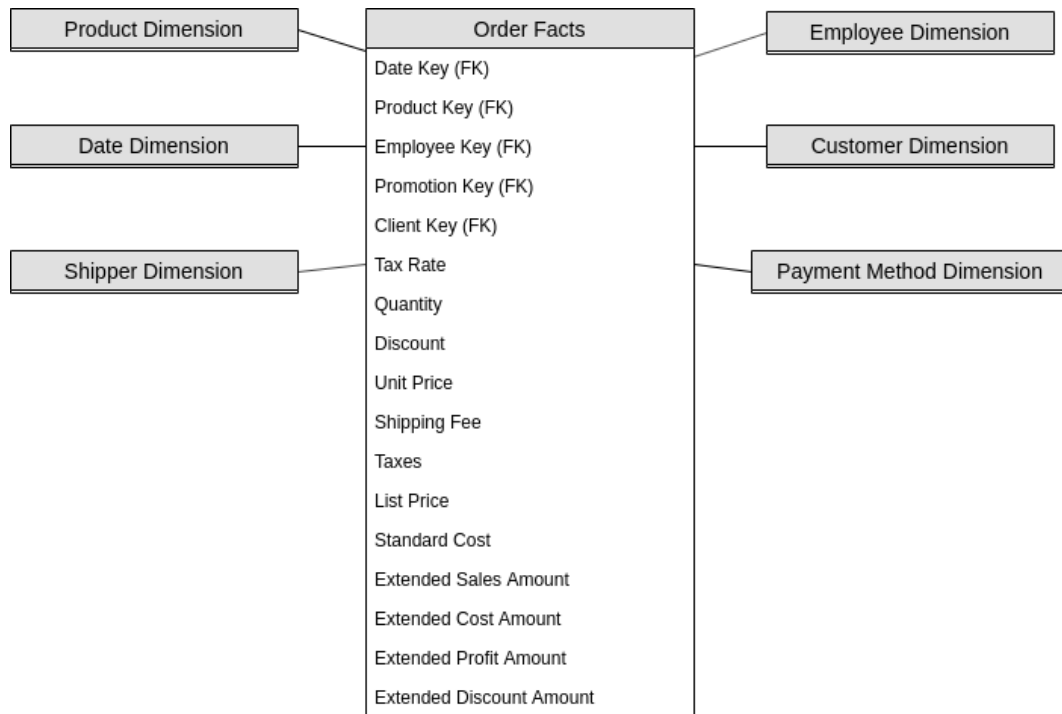


Figura 3: Tabela de Factos de Encomendas e dimensões relacionadas



## 4.2 Criar schema

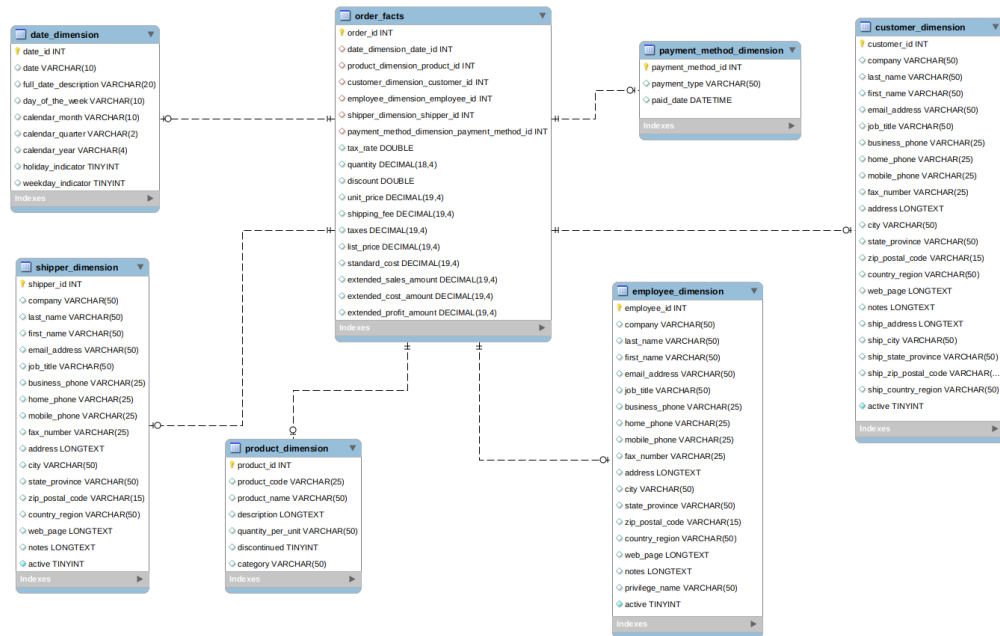


Figura 4: Modelo em Estrela

Em seguida, aplicou-se o modelo acima referido numa instância MySQL onde ficará alojado o Data Warehouse. Depois de criado um novo schema "northwind.dw" foram aplicados os scripts para a criação. Em primeiro lugar, foram criadas as tabelas das dimensões (anexo: 9.1) e, em seguida, a tabela de factos (anexo: 9.2).

## 5 ETL - Extract, Transformation, and Load System

### 5.1 Desenvolver um plano para ETL

#### 5.1.1 Passo 1: Concepção de um plano de alto nível

Nesta fase, começou-se por desenhar um processo simples de todas as partes conhecidas, fontes de dados e destinos, e questões por resolver.

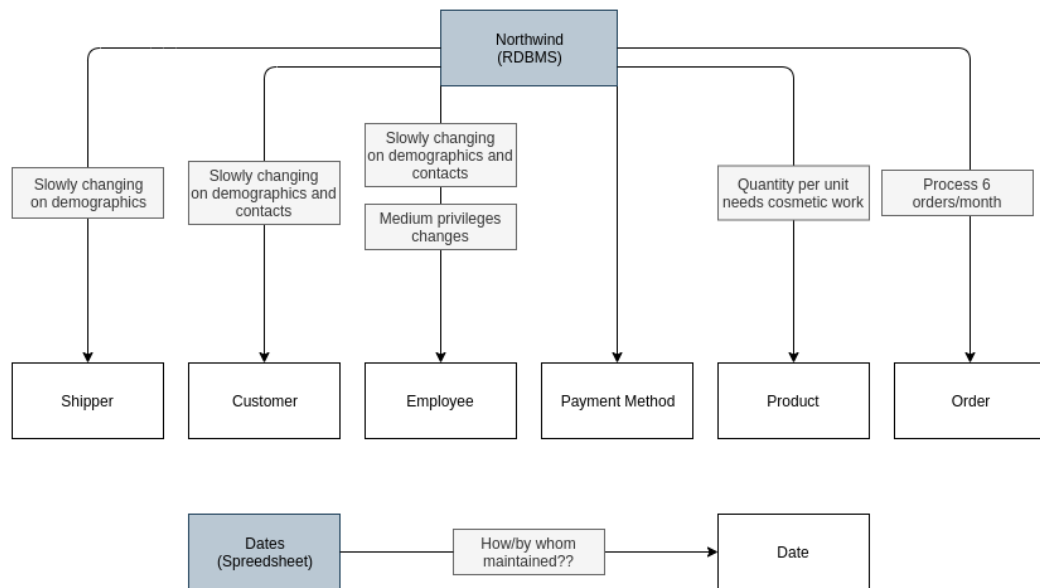


Figura 5: Plano de alto nível para *Staging Area*

#### 5.1.2 Passo 2: Escolha de uma ferramenta de ETL

Como as nossas fontes de dados requerem pouco processamento, a equipa opta por não usar nenhuma ferramenta paga de mercado para ETL e desenvolver código para tratar a transformação dos dados (analisado com mais cuidado em 5.2), e assim diminuir os custos da implementação desta solução.

#### 5.1.3 Passo 3: Aplicação de estratégias

Tal como sugerido por Ralph Kimball em "The Data Warehouse Toolkit", foram aplicadas uma série de processos durante a fase de ETL.

Foram aplicadas as seguintes estratégias:

- **Extração dos dados** - Guardar os dados das fontes de dados num arquivo simples
- **Arquivo dos Dados** - O arquivo de dados do passo anterior será armazenado em *Cloud Storage* durante 1 mês
- **Assegurar a qualidade dos dados** - Foi efetuada uma verificação dos dados recolhidos para as dimensões mas sobretudo para os factos
- **Assegurar requisitos de disponibilidade do sistema** - Foram efetuadas as *queries* mais simples possíveis às fontes de dados
- **Garantir possibilidade de alterações nas tabelas de dimensão** - Reescrita de dados quando é efetuada uma alteração de contacto e criação de uma nova entrada quando é efetuada uma mudança de residência (controlado pela coluna *active*)

#### 5.1.4 Passo 4: Drill Down por Target Table

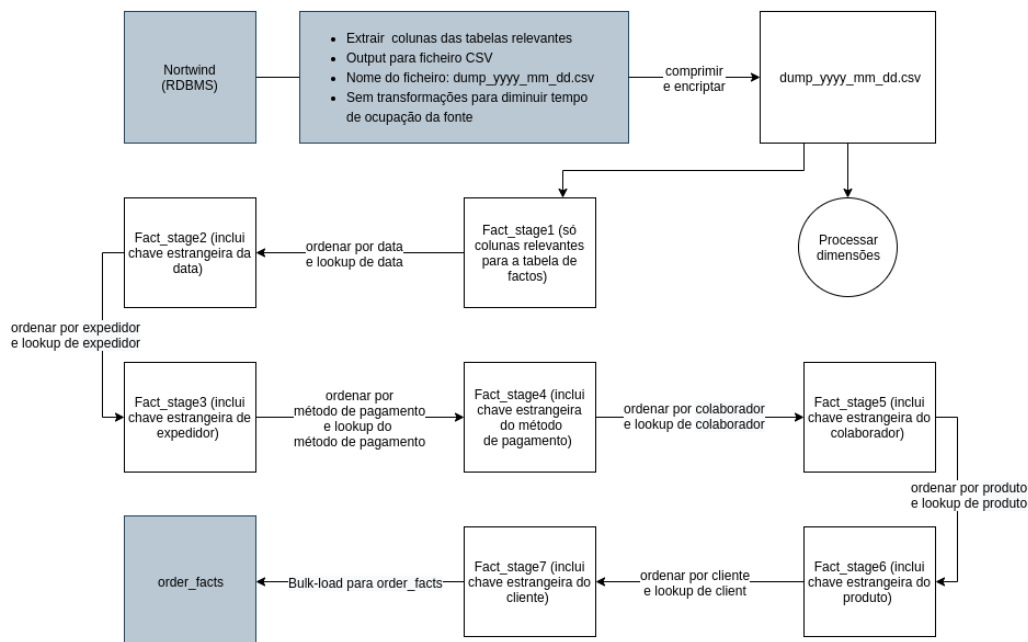


Figura 6: Plano de carregamento de dados para a tabela de Factos

## 5.2 Execução da fase ETL

### 5.2.1 Extração de dados

Com a utilização da dependência Hibernate do Java, geraram-se classes que mapeiam o schema da base de dados *northwind*.

Por exemplo:

```
@Entity
@Table(name = "purchase_order_status", schema = "northwind", catalog = "")
public class PurchaseOrderStatus {
    private int id;
    private String status;

    @Id
    @Column(name = "id")
    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    @Basic
    @Column(name = "status")
    public String getStatus() {
        return status;
    }
}
```

```

    }

    public void setStatus(String status) {
        this.status = status;
    }
}

```

Em seguida, foi criada uma anotação *DWEntity* que ao ser adicionada em qualquer uma destas classes a "marca" como sendo uma das entidades cujos dados serão exportados para CSV.

```

@Entity @DWEntity
@Table(name = "order_details", schema = "northwind", catalog = "")
public class OrderDetails {
    private int id;
    private int orderId;
    private Integer productId;
    private BigDecimal quantity;
    private BigDecimal unitPrice;
    private double discount;
    private Integer statusId;
    private Timestamp dateAllocated;
    private Integer purchaseOrderId;
    private Integer inventoryId;

    @Id
    @Column(name = "id")
    public int getId() {
        return id;
    }

    ...
}

```

Deste modo, então todas as classes com esta anotação, quando é invocado o método *dumpAllTo* serão de maneira simples e direta consultadas as tabelas correspondente e os seus dados escritos para ficheiro em formato CSV.

```

public void dumpAllTo(String filenamePrefix, String suffixPrefix) {
    Reflections reflections = new Reflections("northwind_entities");
    Set<Class<?>> annotated = reflections.getTypesAnnotatedWith(DWEntity.class);

    annotated.forEach(c ->
        dumpOneTo(c, filenamePrefix + c.getSimpleName() + suffixPrefix)
    );
}

public <T> List<T> loadAllData(Class<T> type) {
    Session session = sessionFactory.openSession();
    CriteriaBuilder builder = session.getCriteriaBuilder();
    CriteriaQuery<T> criteria = builder.createQuery(type);
    criteria.from(type);
    List<T> r = session.createQuery(criteria).getResultList();
    session.close();
    return r;
}

```

```
private void dumpOneTo(Class<?> c, String filename) {
    List<?> raw = loader.loadAllData(c);
    print(raw, filename);
}
```

### 5.2.2 Processar dimensões, factos e carregamento de dados

Nesta fase, são carregadas as várias dimensões, a partir da análise dos dados exportados para o CSV, quer seja de maneira direta ou através da conjugação de dados provenientes de diferente tabelas.

Por fim, são analisados e carregados os dados que correspondem aos factos. A esses resultados são depois adicionados os identificadores das dimensões através da procura ordenada da sua correspondências nas mesmas, tal como mostrado no diagram em [5.1.4](#).

```
saver.dumpData(converter.getEmployees());
saver.dumpData(converter.getCustomers());
saver.dumpData(converter.getShippers());
saver.dumpData(converter.getPaymentMethods());
saver.dumpData(converter.getProducts());
saver.dumpData(converter.getDates());

saver.dumpData(converter.loadOrderFacts());

---

public void dumpData(List<?> data) {
    Session session = sessionFactory.openSession();

    session.beginTransaction();

    data.forEach(session::save);

    session.getTransaction().commit();
    session.close();
}
```

## 6 Business Intelligence Applications

Por fim, iremos apresentar alguns exemplos de indicadores clínicos, criados na ferramenta *Power BI*. Este software é um meio de análise de negócios que permite a visualização e estatística dos dados de forma gráfica, carregados a partir da "Presentation Area" da Datawarehouse desenvolvida até aqui.

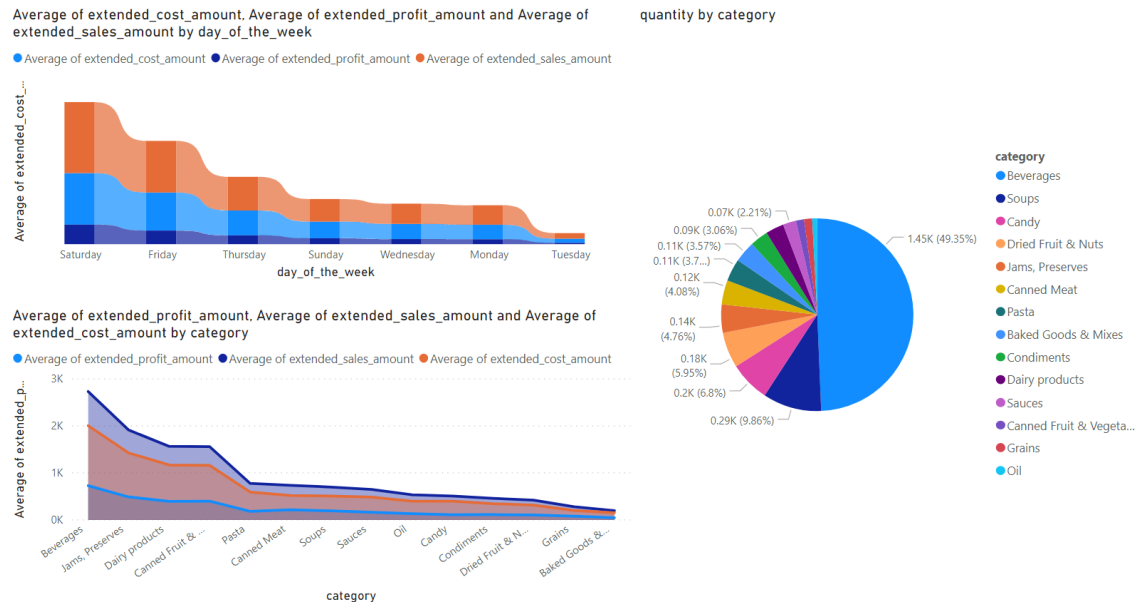


Figura 7: Dashboard 1

Na Dashboard 1 são apresentados 3 indicadores relacionados com a venda de artigos. O primeiro (superior do lado esquerdo) permite observar os valores médios de vendas, de custos, e de lucro gerados por dia da semana.

O segundo (inferior do lado esquerdo) mostra quais as categorias de produto que em média representam maior impacto nas contas da empresa, por exemplo, "Beverages" é a categoria que mais receitas gera para a Northwind.

Por fim, o último gráfico, mostra a quantidade "real" vendida para cada umas das categorias anteriores.



Figura 8: Dashboard 2

Na dashboard 2, podemos observar alguns indicadores que dão algum "insight" sobre o negócio, por exemplo, o lucro gerado pela venda de produtos em cada cidade, os métodos de pagamento mais usados pelos clientes, o peso dos custo de transporte no total do custo do produto e também, a evolução, em termos de sazonais, das vendas de produtos.

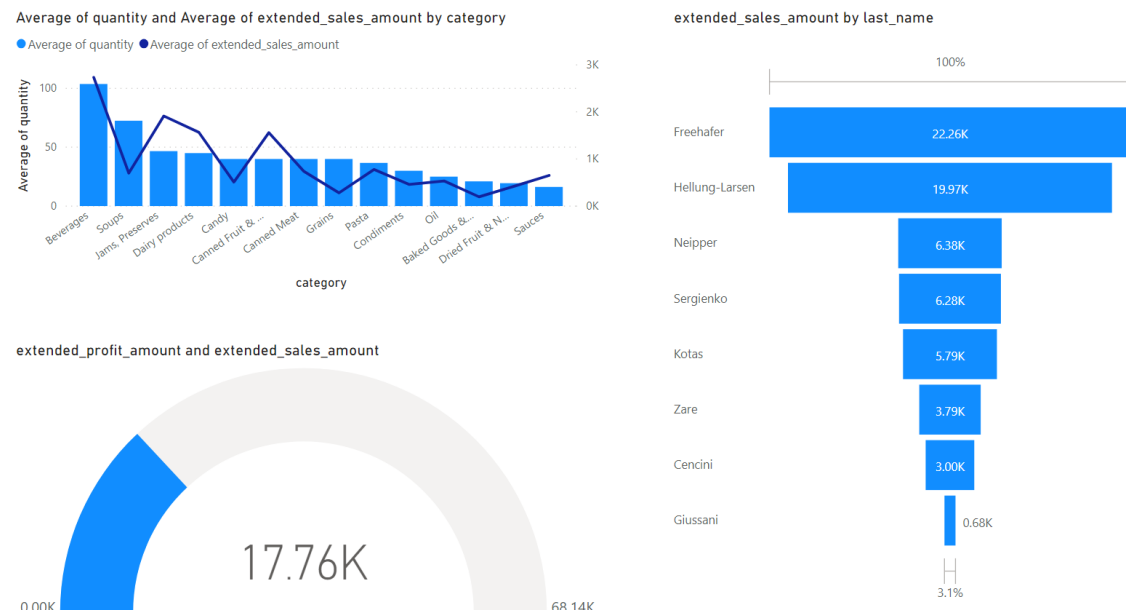


Figura 9: Dashboard 3

Por fim, neste último exemplo, estão apresentados possíveis indicadores para melhorar a gestão da empresa, nomeadamente os produtos mais rentáveis em termos de valores totais de venda vs quantidades vendidas, no qual podemos assumir que "Soups" não é uma categoria muito rentável quando comparada com "Jam, Preserves". A percentagem de lucro que a Northwind faz no valor total de vendas (17.76K/68.14K). E ainda o valor total de vendas de cada um dos seus funcionários, onde se destacam Freehafer e Hellung-Larsen.



## 7 Conclusão

Após o estudo do processo de negócio da empresa *Northwind Traders*, e definidos os padrões de dados e seus relacionamentos a serem analisados pela solução de business intelligence e analytics desenvolvida, com base na metodologia sugerida por Ralph Kimball em *The Data Warehouse Toolkit*, foi desenhada a arquitetura para a Datawarehouse (segundo o *Four-Step Dimensional Design Process*) e desenvolvido e implementado um processo para a extração, transformação e carregamento dos dados (ETL).

Por fim, criada a *Presentation Area* foram, através da ferramenta *Microsoft PowerBI*, criados vários gráficos e dashboards para suporte à decisão, que explicam os resultados de vendas da empresa e possibilitam previsões de resultados futuros.

## 8 Bibliografia

### Referências

- [1] **The Data Warehouse Toolkit:** The definitive guide to dimensional modeling  
*Ralph Kimball, Margy Ross.* Wiley; 3rd edition (July 1, 2013)
- [2] **The Data Warehouse ETL Toolkit:** Practical Techniques for Extracting, Cleaning, Con-  
forming, and Delivering Data  
*Ralph Kimball, Margy Ross.* Wiley; 1 edition (October 1, 2004)
- [3] **Applied Microsoft Power BI:** Bring your data to life!  
*Teo Lachev.* Prologika (January 1, 2019)
- [4] ETL (Extract, Transform, and Load) Process  
<https://www.guru99.com/etl-extract-load-process.html>

## 9 Anexos

### 9.1 Dimensões

#### 9.1.1 Data

```
CREATE TABLE IF NOT EXISTS 'northwind_dw'.'date_dimension' (  
  'date_id' INT NOT NULL,  
  'date' VARCHAR(10) NULL,  
  'full_date_description' VARCHAR(20) NULL,  
  'day_of_the_week' VARCHAR(10) NULL,  
  'calendar_month' VARCHAR(10) NULL,  
  'calendar_quarter' VARCHAR(2) NULL,  
  'calendar_year' VARCHAR(4) NULL,  
  'holiday_indicator' TINYINT NULL,  
  'weekday_indicator' TINYINT NULL,  
  PRIMARY KEY ('date_id'))  
ENGINE = InnoDB;
```

#### 9.1.2 Produto

```
CREATE TABLE IF NOT EXISTS 'northwind_dw'.'product_dimension' (  
  'product_id' INT NOT NULL,  
  'product_code' VARCHAR(25) NULL,  
  'product_name' VARCHAR(50) NULL,  
  'description' LONGTEXT NULL,  
  'quantity_per_unit' VARCHAR(50) NULL,  
  'discontinued' TINYINT NULL,  
  'category' VARCHAR(50) NULL,  
  PRIMARY KEY ('product_id'))  
ENGINE = InnoDB;
```

#### 9.1.3 Expedidor

```
CREATE TABLE IF NOT EXISTS 'northwind_dw'.'shipper_dimension' (  
  'shipper_id' INT NOT NULL,  
  'company' VARCHAR(50) NULL,  
  'last_name' VARCHAR(50) NULL,  
  'first_name' VARCHAR(50) NULL,  
  'email_address' VARCHAR(50) NULL,  
  'job_title' VARCHAR(50) NULL,  
  'business_phone' VARCHAR(25) NULL,  
  'home_phone' VARCHAR(25) NULL,  
  'mobile_phone' VARCHAR(25) NULL,  
  'fax_number' VARCHAR(25) NULL,  
  'address' LONGTEXT NULL,  
  'city' VARCHAR(50) NULL,  
  'state_province' VARCHAR(50) NULL,  
  'zip_postal_code' VARCHAR(15) NULL,  
  'country_region' VARCHAR(50) NULL,  
  'web_page' LONGTEXT NULL,  
  'notes' LONGTEXT NULL,  
  'active' TINYINT NOT NULL,  
  PRIMARY KEY ('shipper_id'))  
ENGINE = InnoDB;
```

#### 9.1.4 Métodos de Pagamento

```
CREATE TABLE IF NOT EXISTS 'northwind_dw'.'payment_method_dimension' (  
    'payment_method_id' INT NOT NULL,  
    'payment_type' VARCHAR(50) NULL,  
    'paid_date' DATETIME NULL,  
    PRIMARY KEY ('payment_method_id'))  
ENGINE = InnoDB;
```

#### 9.1.5 Clientes

```
CREATE TABLE IF NOT EXISTS 'northwind_dw'.'customer_dimension' (  
    'customer_id' INT NOT NULL,  
    'company' VARCHAR(50) NULL,  
    'last_name' VARCHAR(50) NULL,  
    'first_name' VARCHAR(50) NULL,  
    'email_address' VARCHAR(50) NULL,  
    'job_title' VARCHAR(50) NULL,  
    'business_phone' VARCHAR(25) NULL,  
    'home_phone' VARCHAR(25) NULL,  
    'mobile_phone' VARCHAR(25) NULL,  
    'fax_number' VARCHAR(25) NULL,  
    'address' LONGTEXT NULL,  
    'city' VARCHAR(50) NULL,  
    'state_province' VARCHAR(50) NULL,  
    'zip_postal_code' VARCHAR(15) NULL,  
    'country_region' VARCHAR(50) NULL,  
    'web_page' LONGTEXT NULL,  
    'notes' LONGTEXT NULL,  
    'active' TINYINT NOT NULL,  
    'ship_address' LONGTEXT NULL,  
    'ship_city' VARCHAR(50) NULL,  
    'ship_state_province' VARCHAR(50) NULL,  
    'ship_zip_postal_code' VARCHAR(50) NULL,  
    'ship_country_region' VARCHAR(50) NULL,  
    PRIMARY KEY ('customer_id'))  
ENGINE = InnoDB;
```

#### 9.1.6 Colaboradores

```
CREATE TABLE IF NOT EXISTS 'northwind_dw'.'employee_dimension' (  
    'employee_id' INT NOT NULL,  
    'company' VARCHAR(50) NULL,  
    'last_name' VARCHAR(50) NULL,  
    'first_name' VARCHAR(50) NULL,  
    'email_address' VARCHAR(50) NULL,  
    'job_title' VARCHAR(50) NULL,  
    'business_phone' VARCHAR(25) NULL,  
    'home_phone' VARCHAR(25) NULL,  
    'mobile_phone' VARCHAR(25) NULL,  
    'fax_number' VARCHAR(25) NULL,  
    'address' LONGTEXT NULL,  
    'city' VARCHAR(50) NULL,  
    'state_province' VARCHAR(50) NULL,  
    'zip_postal_code' VARCHAR(15) NULL,
```

```

'country_region' VARCHAR(50) NULL,
'web_page' LONGTEXT NULL,
'notes' LONGTEXT NULL,
'privilege_name' VARCHAR(50) NULL,
'active' TINYINT NOT NULL,
PRIMARY KEY ('employee_id'))
ENGINE = InnoDB;

```

## 9.2 Tabela de Factos de Encomendas

```

CREATE TABLE IF NOT EXISTS 'northwind_dw'. 'order_facts' (
  'order_id' INT NOT NULL,
  'date_dimension_date_id' INT NULL,
  'product_dimension_product_id' INT NULL,
  'customer_dimension_customer_id' INT NULL,
  'employee_dimension_employee_id' INT NULL,
  'shipper_dimension_shipper_id' INT NULL,
  'payment_method_dimension_payment_method_id' INT NULL,
  'tax_rate' DOUBLE NULL,
  'quantity' DECIMAL(18,4) NULL,
  'discount' VARCHAR(45) NULL,
  'unit_price' DECIMAL(19,4) NULL,
  'shipping_fee' DECIMAL(19,4) NULL,
  'taxes' DECIMAL(19,4) NULL,
  'list_price' DECIMAL(19,4) NULL,
  'standard_cost' DECIMAL(19,4) NULL,
  'extended_sales_amount' DECIMAL(19,4) NULL,
  'extended_cost_amount' DECIMAL(19,4) NULL,
  'extended_profit_amount' DECIMAL(19,4) NULL,
  PRIMARY KEY ('order_id'),
  CONSTRAINT 'fk_order_facts_date_dimension'
    FOREIGN KEY ('date_dimension_date_id')
    REFERENCES 'northwind_dw'. 'date_dimension' ('date_id')
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT 'fk_order_facts_product_dimension1'
    FOREIGN KEY ('product_dimension_product_id')
    REFERENCES 'northwind_dw'. 'product_dimension' ('product_id')
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT 'fk_order_facts_shipper_dimension1'
    FOREIGN KEY ('shipper_dimension_shipper_id')
    REFERENCES 'northwind_dw'. 'shipper_dimension' ('shipper_id')
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT 'fk_order_facts_payment_method_dimension1'
    FOREIGN KEY ('payment_method_dimension_payment_method_id')
    REFERENCES 'northwind_dw'. 'payment_method_dimension' ('payment_method_id')
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT 'fk_order_facts_customer_dimension1'
    FOREIGN KEY ('customer_dimension_customer_id')
    REFERENCES 'northwind_dw'. 'customer_dimension' ('customer_id')
    ON DELETE NO ACTION

```

```

        ON UPDATE NO ACTION,
    CONSTRAINT 'fk_order_facts_employee_dimension1'
        FOREIGN KEY ('employee_dimension_employee_id')
        REFERENCES 'northwind_dw'.'employee_dimension' ('employee_id')
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

## 9.3 Índices

### 9.3.1 Criar

```

CREATE INDEX 'fk_order_facts_date_dimension_idx'
ON 'northwind_dw'.'order_facts'
('date_dimension_date_id' ASC) VISIBLE;

CREATE INDEX 'fk_order_facts_product_dimension1_idx'
ON 'northwind_dw'.'order_facts'
('product_dimension_product_id' ASC) VISIBLE;

CREATE INDEX 'fk_order_facts_shipper_dimension1_idx'
ON 'northwind_dw'.'order_facts'
('shipper_dimension_shipper_id' ASC) VISIBLE;

CREATE INDEX 'fk_order_facts_payment_method_dimension1_idx'
ON 'northwind_dw'.'order_facts'
('payment_method_dimension_payment_method_id' ASC) VISIBLE;

CREATE INDEX 'fk_order_facts_customer_dimension1_idx'
ON 'northwind_dw'.'order_facts'
('customer_dimension_customer_id' ASC) VISIBLE;

CREATE INDEX 'fk_order_facts_employee_dimension1_idx'
ON 'northwind_dw'.'order_facts'
('employee_dimension_employee_id' ASC) VISIBLE;

```

### 9.3.2 Apagar

```

DROP INDEX 'fk_order_facts_date_dimension_idx'
ON 'northwind_dw'.'order_facts';

DROP INDEX 'fk_order_facts_product_dimension1_idx'
ON 'northwind_dw'.'order_facts';

DROP INDEX 'fk_order_facts_shipper_dimension1_idx'
ON 'northwind_dw'.'order_facts';

DROP INDEX 'fk_order_facts_payment_method_dimension1_idx'
ON 'northwind_dw'.'order_facts';

DROP INDEX 'fk_order_facts_customer_dimension1_idx'
ON 'northwind_dw'.'order_facts';

DROP INDEX 'fk_order_facts_employee_dimension1_idx'
ON 'northwind_dw'.'order_facts';

```