



Universidade do Minho
Escola de Engenharia

ANÁLISE DE DADOS

Data Warehouse para a NorthWind

Mariana Fernandes A81728

Mateus Silva A81952

Maria Pinto PG39292

Diogo Araújo A78485

Conteúdo

1	Introdução	3
2	Base de dados operacional	4
3	Data Warehouse	6
3.1	Arquitetura	6
3.2	Esquema Lógico	6
3.3	Mapa lógico dos dados	8
3.4	ETL	10
3.5	Refrescamento do <i>Data Warehouse</i>	16
4	Indicadores de <i>Business Intelligence</i>	17
4.1	Microsoft PowerBI	17
4.2	Indicadores e sua respectiva análise	17
5	Conclusão	26

Lista de Figuras

1	Modelo lógico da base de dados <i>NorthWind</i>	4
2	Modelo lógico do <i>Data Warehouse</i>	7
3	<i>ETL</i>	11
4	Um exemplo da <i>dashboard</i> principal com alguns indicadores.	17
5	Relação da quantidade de vendas pela empresa de envio.	18
6	Relação da quantidade de vendas pelo fornecedor.	19
7	Relação da quantidade de vendas pelo funcionário.	19
8	Relação da quantidade de vendas por cidade e estado.	20
9	Relação da quantidade de vendas pela categoria do produto.	21
10	Relação da quantidade de vendas por mês.	21
11	Relação da quantidade de vendas por dia da semana.	22
12	Indicadores que nos indicam médias de tempos de Preparação e Quantidade de Produtos.	23
13	Tempo de preparação consoante a categoria dos produtos presentes na venda.	24
14	Indicadores absolutos sobre a informação presente na DW.	24
15	Dashboard otimizada para a visualização mobile.	25

1 Introdução

Estamos perante uma era em que há cada vez mais dados disponíveis e mais fontes de dados, com um reduzido custo de armazenamento, levando isto à necessidade de fazer análises sobre esses dados. A utilização da informação por parte das empresas é um fator fundamental para estas se manterem competitivas e alimentarem os sistemas de suporte à decisão, que ajudam a tomar decisões corretas e rápidas. Para isso, estas necessitam de manter a informação armazenada de forma consolidada, sendo aí que surgem os *data warehouses*, que integram os dados internos e externos de uma organização numa única estrutura, permitindo uma melhor utilização da informação, o aumento da capacidade de resposta e adaptação.

Este projeto, proposto na unidade curricular de Análise de Dados, do 4º ano do Mestrado Integrado em Engenharia Informática, visa pôr em prática as competências adquiridas nas aulas teóricas e práticas. O projeto proposto consiste na análise, planeamento e implementação de um sistema de bases de dados multidimensionais (SBDMD). Para a sua realização foi disponibilizada, pelos docentes, uma base de dados denominada "*North Wind*", que armazena a operacionalidade de uma empresa fictícia que se dedica ao comércio internacional de produtos alimentares.

De forma resumida, será feita a projeção e implementação de um *data warehouse* para a empresa referida anteriormente, tendo por base a base de dados operacional desta empresa. Será feito um o seu povoamento inicial, usando MySQL, e de seguida, após a sua implementação e povoamento, serão criados indicadores de *Business Intelligence*, usando o *Microsoft Power BI*.

2 Base de dados operacional

Para projetar a base de dados multidimensional a construir (*data warehouse*), deve ser feita uma análise detalhada à base de dados a ser utilizada como base da sua construção. Essa base de dados corresponde à *NorthWind*, que foi fornecida pelos docentes da unidade curricular, na forma de *scripts MySQL* para sua criação e povoamento e que suporta a operacionalidade de uma empresa de comércio internacional de produtos alimentares. De notar que esta base de dados foi criada pela *Microsoft* para fins educacionais.

Passando para a análise desta base de dados analítica e explorando o seu conteúdo, esta deve ser feita de forma a identificar quais as tabelas e atributos que devem estar presentes no *data warehouse* a construir, de forma a futuramente fornecer informação consolidada e útil que permita uma melhoria nas decisões a tomar.

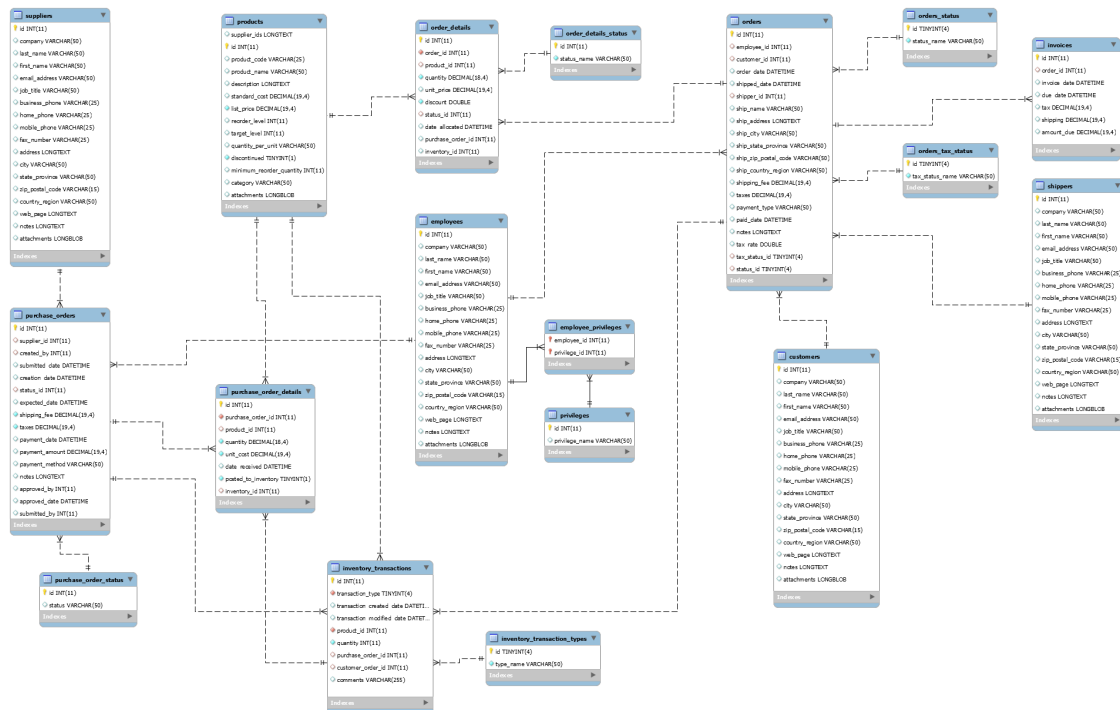


Figura 1: Modelo lógico da base de dados *NorthWind*.

Foi notado que esta base de dados permite a construção de um *data warehouse* com diversos cenários aplicativos dentro da empresa, tais como: análise de vendas, análise dos tempos de preparação e entrega, análise do desempenho dos empregados, etc. Estas são algumas das áreas que podem ser melhoradas através de uma melhor

gestão das vendas, dos funcionários e de vários outros fatores, e que podem levar a um aumento do lucro e satisfação dos clientes.

Tendo em conta a análise feita, decidimos fazer uma análise orientada ao tema das vendas, que permitirá responder a questões tais como:

- Quais os produtos mais vendidos?
- Quais os empregados que mais vendem?
- Qual a localização onde são feitas as encomendas?
- Qual o *shipper* que mais vende?
- Qual o dia da semana que possui mais encomendas realizadas?
- e muitos outros...

Como tal, há um conjunto de tabelas selecionadas para fazer a análise direcionada ao tema escolhido, entre elas podem ser destacadas:

- *Suppliers*: correspondem aos fornecedores de produtos para a empresa
- *Shippers*: transportadores das encomendas feitas pelos clientes
- *Customers*: clientes da empresa que realizam encomendas
- *Products*: conjunto de produtos disponiveis para venda e que compõem as encomendas
- *Orders*: encomendas realizadas pelos clientes à empresa
- *Order_details*: detalhes das encomendas realizadas
- *Employees*: empregados da empresa

Importante notar que, das tabelas mencionadas apenas serão escolhidos alguns atributos como informação para, mais tarde, preencher o *data warehouse*.

3 Data Warehouse

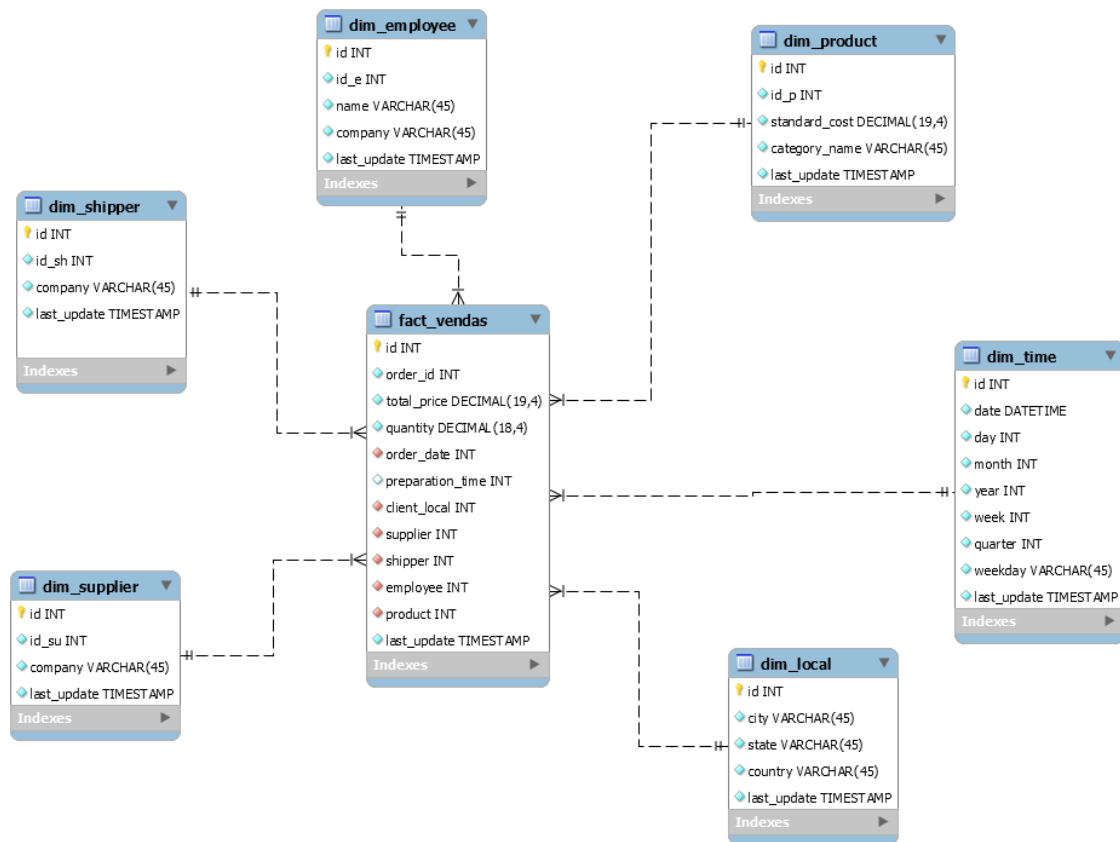
3.1 Arquitetura

Em primeiro lugar, para conceber um *data warehouse* é necessário decidir quais as informações que queremos extrair do mesmo, para que esteja centrado sobre o assunto mais adequado para o negócio em questão. Neste caso, e dada a natureza dos dados, escolhemos as vendas como foco principal do *data warehouse*.

Decidiu-se elaborar o modelo lógico em estrela, por ser mais simples, responder mais rapidamente às *queries* e por o volume de dados disponível não ser grande suficiente que justificasse o uso de um esquema em floco de neve. Este esquema consiste numa tabela de factos central, com diversas dimensões. A tabela de factos encontra-se na 3ª forma normal, quanto que as dimensões se encontram desnormalizadas. A tabela de factos possui apenas valores numéricos.

3.2 Esquema Lógico

Tendo em conta o modelo dimensional escolhido e feita a análise da base de dados de produção elaborou-se o seguinte esquema lógico:

Figura 2: Modelo lógico do *Data Warehouse*

Como é possível observar e como foi já referido anteriormente, este modelo foi desenvolvido segundo a perspetiva das vendas, isto significa que o que se pretende analisar e guardar no *data warehouse* é referente às vendas da empresa e a todos os dados com estas relacionados, nomeadamente, empregados, fornecedores, clientes, preços, quantidades, produtos, etc.

O *data warehouse* projetado possui 6 tabelas de dimensão e a tabela de factos:

- *dim_shipper*: dimensão que representa os expedidores das encomendas da empresa. Contém o nome da empresa que realiza a expedição
- *dim_supplier*: dimensão que representa os fornecedores de produtos para a empresa. Contém o nome da empresa fornecedora.
- *dim_local*: dimensão que representa o local de onde é realizada uma encomenda, ou seja, a morada do cliente. Contém cidade, estado e país.

- *dim_time*: dimensão que contém informação sobre o momento de realização de encomendas. Contém a data, o dia, mês, ano, semana, trimestre e dia da semana.
- *dim_product*: dimensão que representa os produtos vendidos pela empresa. Contém o seu preço e categoria.
- *dim_employee*: dimensão que representa os funcionários da empresa. Contém o seu nome e a empresa para a qual trabalham.
- *fact_vendas*: possui o componente a ser analisado, ou seja, as vendas. Contém o identificador da encomenda, o preço total, a quantidade de produto, a data de realização da encomenda, o tempo de preparação, o cliente, o fornecedor, o expedidor, o funcionário e o produto.

Para além dos atributos referenciados, cada tabela possui um *id* como chave primária, que é *auto-increment*. As tabelas *dim_shipper*, *dim_supplier*, *dim_employee* e *dim_product* possuem ainda os identificadores que possuem na base de dados *NorthWind*.

Para além disso, é importante referir que cada uma das tabelas possui um campo *timestamp* que corresponde ao momento em que foram inseridas no *data warehouse*.

3.3 Mapa lógico dos dados

Tabela *dim_employee*:

Nesta tabela escolheu-se manter a *company*, apesar de, nos dados fornecidos, esta ser igual para todos os *employees*. Futuramente, com a expansão da *NorthWind*, poderão abrir mais lojas, pelo que faz sentido manter este campo para gerar o indicador de vendas por *company*.

Tabela de Origem	Campo de Origem	Campo de Destino
employees	id	id_e
employees	first_name+last_name	name
employees	company	company

Tabela *dim_shipper*:

Tabela de Origem	Campo de Origem	Campo de Destino
shippers	id	id_sh
shippers	company	company

Tabela *dim_supplier*:

Tabela de Origem	Campo de Origem	Campo de Destino
suppliers	id	id_su
suppliers	company	company

Tabela dim_product:

Tabela de Origem	Campo de Origem	Campo de Destino
products	id	id_p
products	standard_cost	standard_cost
products	category	category_name

Tabela dim_local:

Tabela de Origem	Campo de Origem	Campo de Destino
customers	city	city
customers	state_province	state
customers	country_region	country

Tabela dim_time:

Pensamos em guardar a hora como um campo separado, mas não a distribuição de vendas por hora não pareceu ser muito relevante para este negócio em particular. Em vez disso, produzimos o campo *quarters*, que representa o trimestre a que pertence a data, pois vendas por trimestre são muitas vezes é um indicador utilizado frequentemente para analisar um negócio e a sua sazonalidade.

Tabela de Origem	Campo de Origem	Campo de Destino
orders	order_date	date
orders	day(order_date)	day
orders	month(order_date)	month
orders	year(order_date)	year
orders	week(order_date)	week
orders	quarter(order_date)	quarter
orders	weekday(order_date)	weekday

Tabela fact_vendas:

Consideramos que seria útil para análise a base de dados de produção ter o campo *delivery_date* para que fosse possível analisar os tempos de entrega dos *shippers*. Para além dos campos presentes na tabela seguinte, a mesma contém as chaves estrangeiras das tabelas de dimensão.

Tabela de Origem	Campo de Origem	Campo de Destino
order_details	order_id	order_id
order_details	sum(unit_price)	total_price
order_details	quantity	quantity
orders	datediff(shipped_date,order_date)	preparation_time

3.4 ETL

Uma vez que estamos perante uma situação em que apenas existe uma fonte de dados (base de dados da *NorthWind*), decidimos que não seria necessário uma área de retenção, pois esta base de dados auxiliar tem como função agregar, sumariar e armazenar a informação de diferentes fontes de dados, pelo que seria igual ao *data warehouse* a ser construído.

Nesta fase terminada a modelação dimensional, usou-se a ferramenta *Forward Engineer* disponibilizada pelo *MySQL Workbench* para passar o modelo projetada para uma base de dados física, o *data warehouse*, denominado por *dw*. Para além disso, sabemos também a fonte e destino de cada um dos atributos do *data warehouse*.

O passo seguinte corresponde ao processo ETL que consiste num conjunto de processos que inclui a extração dos dados de fontes de informação, a transformação dos dados de acordo com as necessidades da organização e, por fim, o carregamento dos mesmos numa estrutura de dados, como por exemplo um *data warehouse*.

O processo de extração de dados consiste em compreender, selecionar e copiar os dados fonte para a área de retenção, o que no nosso processo de ETL não acontece, uma vez que não há necessidade dessa estrutura intermédia. No nosso caso, durante este processo, foi necessário lidar com valores nulos, mas não foi necessário lidar com datas em diferentes formatos, nem foi necessário fazer integração de dados de fontes heterogéneas, porque apenas fizemos extração de uma fonte, a base de dados operacional da *NorthWind*.

No que toca ao processo de transformação dos dados, este consiste na eliminação de campos inúteis, combinação de dados provenientes de diferentes fontes, criação de chaves primárias independentes dos sistemas fonte e construção de agregados, de modo a acelerar as pesquisas.

O principal objetivo desta etapa é melhorar a qualidade dos dados, de forma a que estes se tornem corretos, claros (possuem apenas um significado) e consistentes. No que toca à limpeza dos dados, este processo foi consideravelmente trivial neste trabalho, pois não existiam valores sem sentido, dados duplicados, nem dados cujo significado não é claro ou é contraditório. Apenas foi necessário tratar alguns atributos em que se notou ausência de valores.

O carregamento dos dados, onde geralmente são carregados muitos registos de uma só vez (*bulk*), tal como acontece no projeto que desenvolvemos, permite a disponibilidade no *data warehouse* dos dados extraídos das fontes operacionais, e que foram corretamente validados na área de retenção, que tal como mencionado anteriormente, na nossa implementação, esta validação ocorre não nas áreas de retenção, mas sim numa transformação instantânea dos dados entre a sua extração das fontes

e o carregamento no *data warehouse*. Tudo o que é feito para otimizar o desempenho de um *data warehouse* tende a atrasar o carregamento, como por exemplo a criação de índices, agregados, particionamento de tabelas, paralelismo e distribuição.

No processo de carregamento, começou-se por elaborar um planeamento que consistiu, basicamente, na escolha da ferramenta para elaborar o processo de ETL, que foi um script SQL. Esta fase faz uso do mapa lógico de dados desenvolvido. O passo seguinte passou pelo carregamento das dimensões. Por fim, foi realizado o carregamento dos factos.

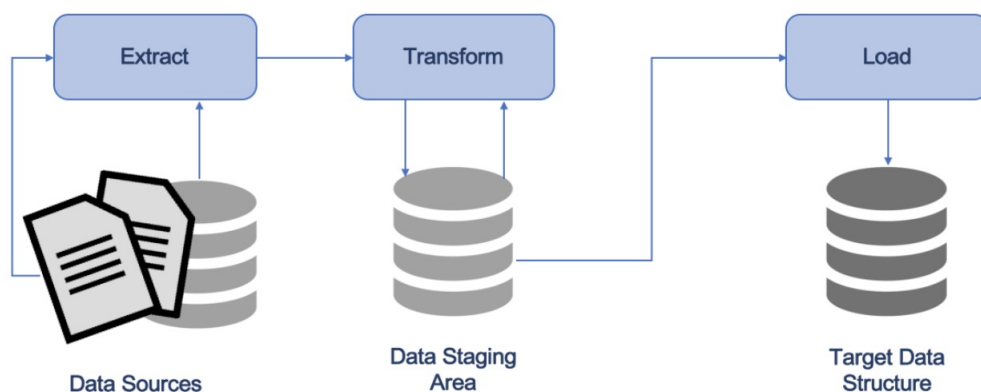


Figura 3: ETL

Como já referido anteriormente, o processo ETL realizado resume-se no carregamento dos dados da fonte diretamente para o *data warehouse*, tendo sido este processo realizado usando *queries* de inserção.

Durante todo este processo foi necessário ter atenção aos diversos valores a *NULL* presentes na base de dados *NorthWind*, pois segundo os nossos conhecimentos em *data warehouses* não devem existir campos a *NULL*.

Para lidar com estas situações, onde encontramos valores a *NULL*, estes foram substituídos por "*Unknown*". Para além disso, incluímos uma nova entrada nas tabelas de dimensão, apenas nas que foi necessário, que corresponde a um campo que é desconhecido, ou seja, é referenciado na base de dados *NorthWind* com *NULL*. Estas entradas possuem o valor -1" como chave primária para não interferir com as restantes entradas, cuja chave primária é incrementada automaticamente.

Como ditam as regras de povoamento de um *data warehouse*, inicialmente foram povoadas as tabelas de dimensão, para futuramente ser possível povoar a tabela de factos.

Usando como exemplo a tabela de dimensão de *shippers*, é apresentado o seguinte código:

```
1 insert into dw.dim_shipper(id_sh, company, last_update)
2   select s.id, s.company, now()
3   from northwind.shippers s;
4
5 insert into dim_shipper (id, id_sh, company, last_update) VALUES
   (-1, -1, 'Unknown', '1975-01-01 00:00:00');
```

Listing 1: Povoamento tabela de dimensão de *shipper*

Esta tabela necessitou do *insert* da entrada com índice -1”, pois há na base de dados *NorthWind*, na tabela de *orders*, valores cuja chave estrangeira de *shipper* é *NULL*, então, quando esses valores são encontrados, a tabela de factos irá conter este valor -1, na dimensão *shipper* como a chave estrangeira do facto.

As restantes tabelas de dimensão do *data warehouse* foram povoadas de forma semelhante, tendo sido necessário apenas a inserção da entrada para lidar com *NULLs* nas dimensões *shipper* e *supplier*. De referir que no povoamento da tabela de dimensão *time* fizemos uso de funções predefinidas do *MySQL* para separar uma data nos diferentes campos, tais como *day*, *month*, *year*, *week*, *quarter* e *weekday*.

Após o povoamento das tabelas de dimensão, foi o momento de realizar o povoamento da tabela de factos. Este povoamento é bastante mais complexo do que as tabelas de factos, pois relaciona os dados existentes na tabela *order_details* da base de dados *NorthWind* com os dados das tabelas de dimensão, contudo é realizado usando uma *query SQL*.

De forma a simplificar a *query* de povoamento da tabela de factos, foram criadas algumas funções:

- Função que calcula o preço total de uma encomenda. Recebe como argumento a o identificador de uma encomenda.

```
1 delimiter $$
2 create function total_price( order_id int )
3 returns decimal(19,4)
4 deterministic
5 begin
6   declare total decimal(19,4);
7
8   set total = (select sum(od.unit_price) from northwind.
   order_details od where od.order_id = order_id);
9
10  return total;
```

```
11 end $$  
12 delimiter ;
```

- Função para retornar o identificador do *supplier* dos produtos presentes numa encomenda, dado o identificador de um *order_details*. Se o atributo *purchase_order_id* estiver *NULL* é retornado -1” que corresponde à entrada ”Unknown” presente na dimensão *supplier* criada para lidar com os valores inexistentes. Caso contrário acedemos à estrada correspondente a esse valor na tabela *purchase_ordes* da *NorthWind* para retirar o valor do *supplier*, tal como pretendido.

```
1 delimiter $$  
2 create function get_supplier( order_details_id int )  
3 returns int  
4 deterministic  
5 begin  
6     declare res int;  
7     declare interm int;  
8     declare res2 int;  
9  
10    set interm = (select od.purchase_order_id from northwind.  
11                  order_details od where od.id = order_details_id);  
12    set res = coalesce(interm,-1);  
13  
14    if res = -1 then  
15        return res;  
16    else  
17        set res2 = (select po.supplier_id from northwind.  
18                    purchase_orders po where po.id = res);  
19    end if;  
20    return res2;  
21 end $$  
22 delimiter ;
```

- Função para retornar o valor do *shipper* de uma encomenda, dado o identificador de um *order_details*. Mais uma vez, se este valor estiver a *NULL* na tabela *order_details*, será retornado -1”, caso contrário será retornado o valor do restepivo *shipper*.

```
1 delimiter $$  
2 create function get_shipper( orders_id int )  
3 returns int  
4 deterministic  
5 begin  
6     declare res int;  
7     declare interm int;  
8
```

```
9   set interm = (select shipper_id from northwind.orders where
    id = orders_id);
10  set res = coalesce(interm,-1);
11
12  return res;
13 end $$
14 delimiter ;
```

- Função para obter o tempo de preparação de em encomenda em dias, dado o identificador de uma *order*. Este campo corresponde à diferença de datas entre o momento em que uma encomenda é realizada e o momento em que é enviada. Para isso, usamos a função predefinida no *MySQL* *datediff* com os parâmetros *shipped_date* e *order_date* presentes na tabela *orders* da base de dados da *NorthWind*.

```
1 delimiter $$
2 create function get_preparation_time( orders_id int )
3 returns int
4 deterministic
5 begin
6     declare res int;
7     declare interm int;
8
9     set interm = datediff((select shipped_date from northwind.
    orders where id = orders_id), (select order_date from
    northwind.orders where id = orders_id));
10    set res = coalesce(interm,-1);
11
12    return res;
13 end $$
14 delimiter ;
```

Por fim, e usando as funções definidas é feito o povoamento da tabela de factos usando a seguinte *query*:

```
1 insert into dw.fact_vendas
2 (order_id,
3  total_price,
4  quantity,
5  order_date,
6  preparation_time,
7  client_local,
8  supplier,
9  shipper,
10 employee,
11 product,
```

```
12     last_update
13     )
14     select
15         od.order_id,
16         total_price(od.order_id),
17         od.quantity,
18         t.id,
19         get_preparation_time(ord.id),
20         l.id,
21         s.id,
22         shi.id,
23         e.id,
24         p.id,
25         now()
26     from
27         northwind.order_details od,
28         northwind.orders ord,
29         dw.dim_time t,
30         dw.dim_local l,
31         northwind.customers c,
32         dw.dim_shipper shi,
33         dw.dim_employee e,
34         dw.dim_product p,
35         dw.dim_supplier s
36     where
37         od.order_id = ord.id and
38         ord.order_date = t.date and
39         ord.customer_id = c.id and
40         c.city = l.city and
41         c.state_province = l.state and
42         c.country_region = l.country and
43         s.id_su = get_supplier(od.id) and
44         shi.id_sh = get_shipper(ord.id) and
45         ord.employee_id = e.id_e and
46         od.product_id = p.id_p;
```

Listing 2: Povoamento tabela de factos do *data warehouse*

Terminado o povoamento do *data warehouse*, estamos agora em condições de criar indicadores de *Business Intelligence*.

3.5 Refrescamento do *Data Warehouse*

Para fazer o refrescamento do *data warehouse*, ou seja, adicionar apenas os novos dados na base de dados de produção que não constam do *data warehouse*. Fundamentalmente, basta fazer um *left join* entre os dados da *Northwind* e do DW.

Por exemplo, para a tabela *dim_shipper*

```
1 insert into dw.dim_shipper
2   (id_sh, company)
3 select northwind.shippers.id, coalesce(northwind.shippers.company, '
4   Unkown')
5 from northwind.shippers
6   left join dw.dim_shipper on northwind.shippers.id = dw.
7   dim_shipper.id_sh and
8   northwind.shippers.company = dw.dim_shipper.company
9 where dw.dim_shipper.id is null;
```

Este procedimento é repetido para as restantes tabelas, no entanto, é preciso atualizar os *ids* da tabela de factos quando uma tabela de dimensão tem um novo registo com a mesma chave natural (chave original da base de dados de produção) que um registo anterior.

```
1 delimiter //
2 create trigger shipper_update
3 after
4 insert
5   on
6   dw.dim_shipper
7 for each row
8 begin
9
10    update dw.fact_vendas
11   inner join dw.dim_shipper
12     on
13     dw.sales_fact.shipper = dw.dim_shipper.id and
14     dw.dim_shipper.id_sh = new.id_sh and dw.dim_shipper.id_sh = new
15     .id_sh
16   set dw
17     .fact_vendas.shipper = new.id;
18 end; //
19 delimiter ;
```

Desta tarefa, concluímos que deveríamos ter utilizado uma área de retenção, apesar de só haver uma fonte de dados, pois teria facilitado muito este processo.

4 Indicadores de *Business Intelligence*

Através do *Business Intelligence* recolhemos, transformamos, organizamos, analisamos e distribuimos os dados. Utilizamos este conceito para melhorar e simplificar o processo de tomada de decisão e desta forma, termos informação útil para uma análise de dados.

4.1 Microsoft PowerBI

De forma a ser possível realizar uma análise de dados mais corrente e útil, desenvolveu-se uma variedade de indicadores que permitissem ao utilizador e visualizador relacionar os dados do *Data Warehousing* para as suas futuras tomadas de decisão. Todos os indicadores desenvolvidos foram agrupados numa *dashboard* denominada de ***Indicadores Principais*** permitindo a sua visualização e interação para o grupo de trabalho e analistas tanto no computador como na sua versão mobile. Estes indicadores serão então analisados mais profundamente na próxima secção.

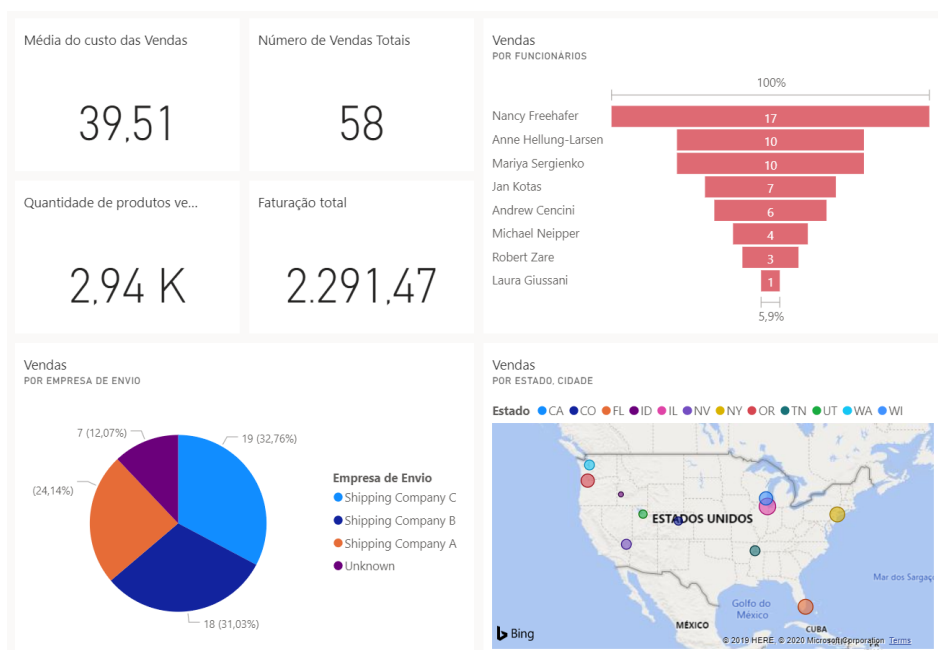


Figura 4: Um exemplo da *dashboard* principal com alguns indicadores.

4.2 Indicadores e sua respetiva análise

É através destes indicadores que realizamos uma análise mais profunda e simplificada dos dados fornecidos. Conseguimos visualizar através destes indicadores, o mês e

dia da semana que se obteve mais vendas, o funcionário que realizou mais vendas, o produto mais vendido, a cidade e estados onde se obtêm mais vendas, etc. Estes indicadores são fundamentais para uma boa análise e tomada de decisão.

Os indicadores e suas respectivas informações são as seguintes:

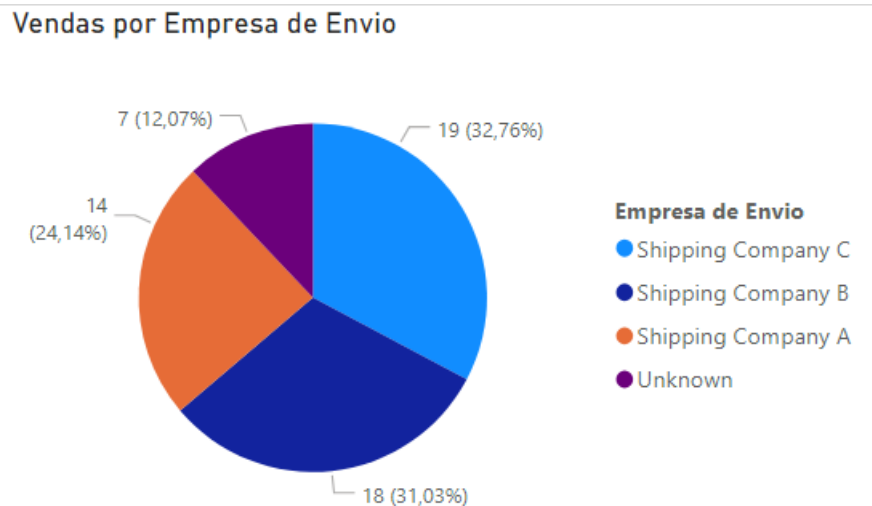


Figura 5: Relação da quantidade de vendas pela empresa de envio.

Através deste indicador (Figura 5), conseguimos concluir que a empresa que enviou mais vendas foi a *Shipping Company C*.

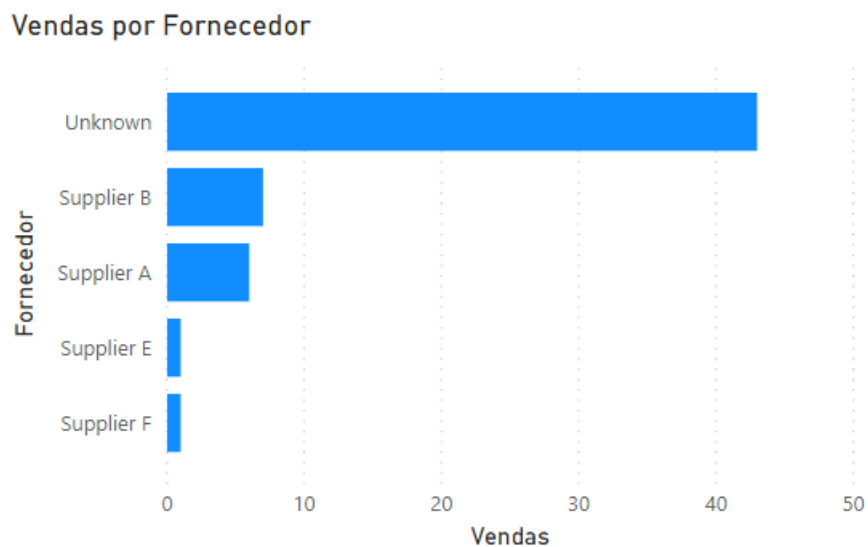


Figura 6: Relação da quantidade de vendas pelo fornecedor.

Através deste indicador (Figura 6), concluímos que o fornecedor que realizou mais vendas era desconhecido. O fornecedor B destacou-se com 7 vendas, seguido pelo fornecedor A com 6 vendas. Seguiu-se o fornecedor E e F.

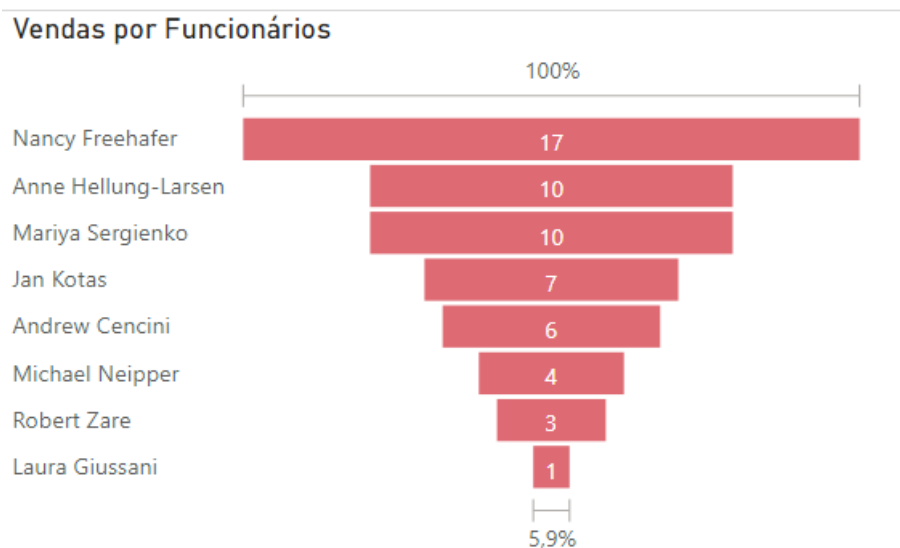


Figura 7: Relação da quantidade de vendas pelo funcionário.

Através deste indicador (Figura 7), pretendíamos visualizar o número de vendas

realizado por cada funcionário. Com a realização deste indicador, podemos concluir que a funcionária Nancy Freehafer teve o mais número de vendas. Seguiu-se a funcionária Anne Hellung-Larsen e a Mariya Sergienko. Esta análise de dados é uma mais valia para a empresa, visto que, a partir desta informação a empresa consegue controlar e recompensar os funcionários que fizeram um bom trabalho.



Figura 8: Relação da quantidade de vendas por cidade e estado.

Através deste indicador (Figura 8), pretendíamos visualizar o número de vendas realizado em cada cidade e estado. Através deste mapa, concluímos que Chicago foi o local com o maior número de vendas, seguido de, New York, Miami e Portland. Esta informação é bastante útil para a empresa, de modo que, a empresa sabendo o local onde as vendas são mais rentáveis pode optar por abrir mais lojas nesses mesmas cidades ou estados.

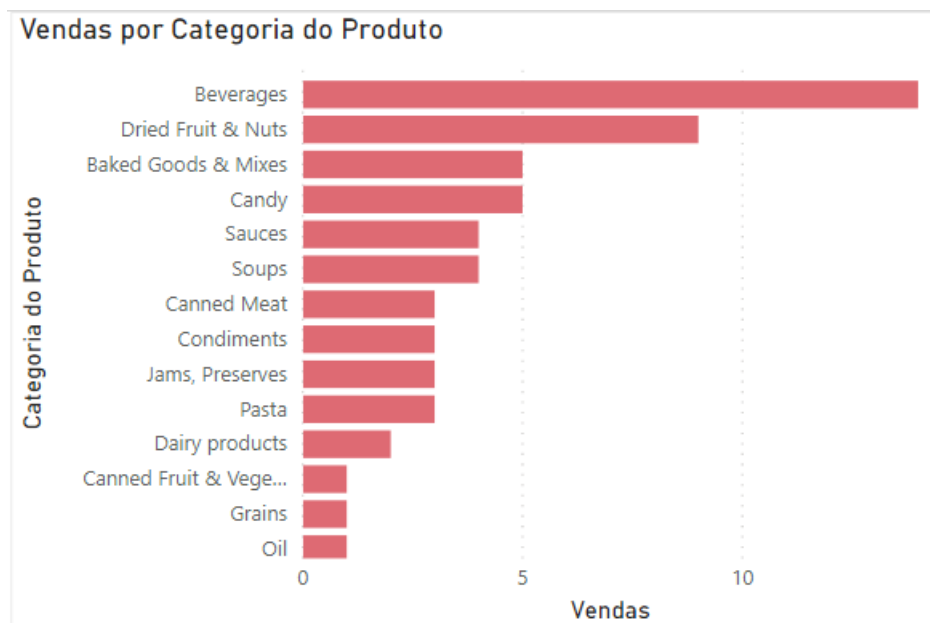


Figura 9: Relação da quantidade de vendas pela categoria do produto.

Através deste indicador (Figura 9), visualizamos que o maior lucro nas vendas é através das bebidas e dos frutos secos. A partir desta informação a empresa pode apostar neste tipo de produtos de modo a aumentar o lucro.

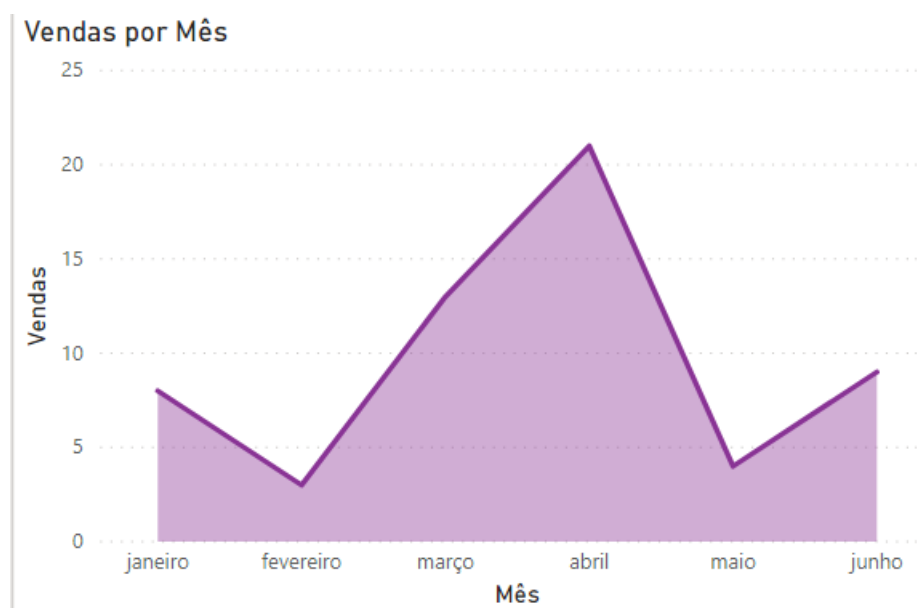


Figura 10: Relação da quantidade de vendas por mês.

Através deste indicador (Figura 10), conseguimos visualizar os meses do ano em que a empresa obteve mais vendas. Com este gráfico, conseguimos concluir que o mês de Abril é o pico de vendas, ou seja, é o mês com o maior número de vendas (21). Seguiu-se o mês de Março e de Janeiro.

Dia de Semana	Vendas
Sexta-Feira	19
Quarta-Feira	15
Segunda-Feira	9
Domingo	5
Terça-Feira	5
Sábado	3
Quinta-Feira	2
Total	58

Figura 11: Relação da quantidade de vendas por dia da semana.

Neste indicador (Figura 11), foi criada a visualização em modo tabela das vendas por dia da semana e depois colocado de forma decrescente para melhor compreensão de que o dia em que houve mais vendas foi efetivamente à sexta-feira com dezanove vendas.

Para acontecer esta visualização de forma mais humana e perceptível teve de existir uma modificação dos dados recebidos pelo *Data Warehouse* que nos fornecia os dias da semana de 0 (zero) a 6 (seis) que correspondiam a Segunda-Feira e Domingo, respetivamente. Com a troca destes dados utilizando a função do *Microsoft PowerBI* intitulada de *Power Query Editor* que nos forneceu uma interface simples para transformar estes dados em algo final para se demonstrar como indicador.



Figura 12: Indicadores que nos indicam médias de tempos de Preparação e Quantidade de Produtos.

Estes dois indicadores que são demonstrados na Figura 12 são especialmente interessantes dado que visualmente nos dão uma informação aparentemente simples mas com bastante relevância.

No primeiro indicador temos a média do tempo que demora a preparar as encomendas, ou seja, desde a sua data de encomenda até à data de envio. Assim conseguimos observar que a média para tal é de cerca de 1,6 dias mas pode ir a um máximo de catorze dias como informa o indicador.

No segundo indicador temos uma informação similar em que nos demonstra que a quantidade média de produtos numa venda é por volta de cinquenta itens, mas chegando a um máximo (até agora) de trezentos produtos.

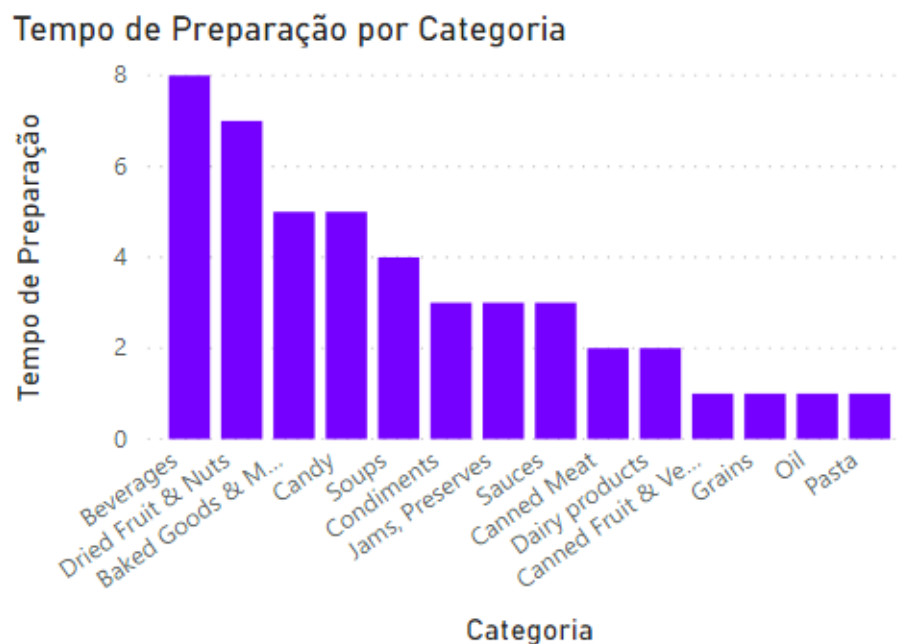


Figura 13: Tempo de preparação consoante a categoria dos produtos presentes na venda.

Na figura 13 temos uma informação útil para uma decisão sobre como é possível otimizar a preparação de certo tipo de produtos dado que nos demonstra o tempo da preparação consoante o tipo de produto, o que nos pode ajudar a melhorar várias secções do *background* pré-envio.



Figura 14: Indicadores absolutos sobre a informação presente na DW.

Por fim, na figura 14 foram adicionadas estas métricas que são sempre importantes

dado que nos demonstram as informações absolutas sobre a faturação, quantidade de produtos vendidos e o número de vendas até ao momento.

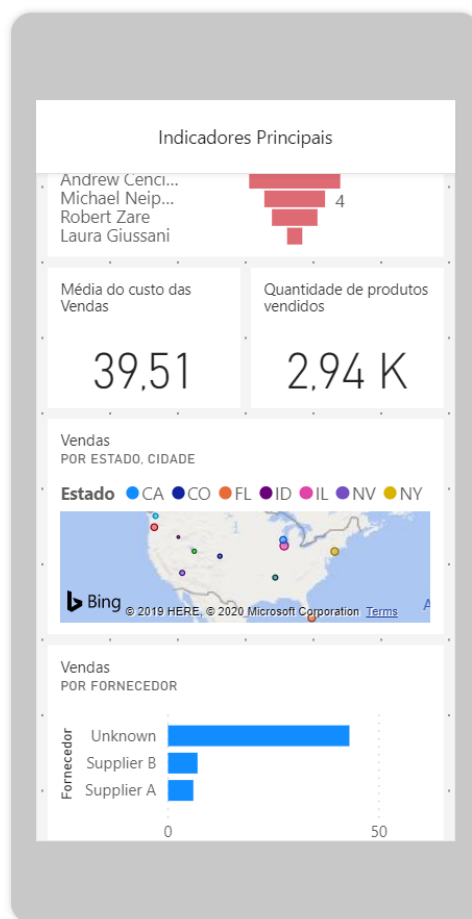


Figura 15: Dashboard otimizada para a visualização mobile.

É de salientar que foram utilizadas as funcionalidades do *Microsoft PowerBI* por forma a ser possível a visualização da **dashboard** explicada acima através das aplicações oficiais presentes nos sistemas operativos líderes no mercado *mobile*, concretamente o Apple iOS e o Google Android, sendo que a disposição dos vários indicadores foi colocada e alterada por forma a se visualizar mais confortavelmente no modo vertical dos ecrãs dos *smartphones*.

5 Conclusão

Com a realização deste trabalho, o grupo conseguiu entender a importância de um Data Warehouse e da sua relação com o conceito de Business Intelligence. É através de uma análise de dados que temos as informações necessárias para termos vantagens face a outras empresas. Para além de termos vantagem competitiva, conseguimos tomar decisões corretas e rápidas, superar as vendas e garantir os resultados pretendidos. Deste modo, a partir da informação útil que obtemos, podemos aumentar os lucros e conseguir os melhores valores possíveis.

Um Data Warehouse permite uma melhor utilização dos dados, na medida em que, integra os dados internos e externos de outra base de dados numa estrutura única. Através de um Data Warehouse e da apresentação da informação, através do POWER BI, podemos obter uma melhor utilização dos dados, e, consequentemente, obtermos melhores e novos produtos/serviços.

Em suma, este trabalho prático foi bastante importante, para melhorar a nossa perspetiva e capacidade de planear e gerir uma base de dados, e também a importância da informação face à competitividade entre empresas.