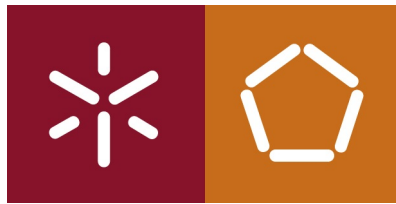


UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA



NorthWind - SDBMD

Engenharia do Conhecimento
Análise de Dados

Henrique Pereira (a80261)
Sarah Tiffany (a76867)

Abril 2020

Conteúdo

1	Introdução	3
2	Análise da Base de Dados <i>NorthWind</i>	4
2.1	Modelo Lógico	4
2.2	Qualidade dos Dados	5
2.2.1	Produtos	5
2.2.2	Clientes	6
2.2.3	Fornecedores	6
2.2.4	Transportadoras	7
2.2.5	Funcionários	7
2.2.6	Vendas	7
2.2.7	Compras	8
2.2.8	Inventário	9
3	Planeamento do Data Warehouse	10
3.1	Métricas e requisitos	10
3.2	Modelo do Data Warehouse	10
4	Implementação: ETL	13
4.1	Mapas Lógicos de Dados	13
4.2	Povoamento do Data Warehouse	15
4.3	Mecanismos de atualização do Data Warehouse	18
5	Dashboards de Business Intelligence	24
5.1	Compras - <i>purchases</i>	24
5.1.1	Quantidade total e quantidade média das compras por fornecedor	24
5.1.2	Valor total e valor médio das compras por fornecedor	25
5.1.3	Quantidade de cada produto vendido por fornecedor	26
5.1.4	Produtos mais comprados	27
5.2	Vendas - <i>sales</i>	28
5.2.1	Quantidade total e quantidade média das compras por cliente	28
5.2.2	Valor total e valor médio das compras por cliente	29
5.2.3	Número total e médio de compras por país e cidade	30
5.2.4	Produtos mais vendidos	32
5.3	Comparação entre Compras e Vendas	33
5.3.1	Quantidade e valor de compras e vendas por mês	33
5.3.2	Quantidade e valor de compras e vendas por categoria de produto	34
5.3.3	Quantidade e valor de compras e vendas por mês e categoria de produto	35

5.3.4	Número de compras e vendas por dia da semana	37
5.4	Funcionários	38
5.4.1	Funcionários que mais venderam	38
6	Conclusão	39

1 Introdução

No âmbito da Unidade Curricular Análise de Dados, do perfil de Engenharia do Conhecimento do Mestrado Integrado em Engenharia Informática, foi-nos proposta o planeamento e implementação de um Sistema de Bases de Dados Multidimensionais (SBDMD), além da construção de plataformas para suporte à decisão. Isto é, foi-nos pedido que arquitetássemos um Data Warehouse (DW) e que, partindo deste, gerássemos *dashboards* para facilitar a visualização dos dados.

Para tal, a base de dados (BD) *NorthWind* foi selecionada como fonte de dados para o nosso SBDMD. Como referido no enunciado deste trabalho prático, a *NorthWind* trata-se de uma empresa fictícia cuja atividade reside no comércio internacional de produtos alimentares. É uma base de dados amplamente utilizada, desenhada pela Microsoft, para âmbitos académicos e de teste. A BD da *NorthWind*, como será explicitado na secção seguinte, contém todas as transações entre fornecedores e NorthWind e entre NorthWind e clientes. Ora, uma empresa que funciona como um intermediário neste gigantesco processo, se não organizar os dados de maneira a permitir uma fácil visualização dos mesmos, terá pela frente um grande desafio, daí a necessidade da criação de um SBDMD.

Esta base de dados será construída com o MySQL, utilizando os *scripts* existentes no repositório do GitHub fornecido no enunciado.

Seguindo os conteúdos lecionados nas aulas teóricas, este projeto foi dividido em várias fases:

1. Análise da BD
2. Planeamento do DW
3. ETL e Implementação do DW
4. Criação de *dashboards*
5. Análise dos resultados

Conforme sugerido pelos docentes, a implementação do DW e processos de ETL serão realizados utilizando também *scripts* de MySQL. Para a visualização de dados, será utilizado o *software* da Microsoft Power BI.

2 Análise da Base de Dados *NorthWind*

O primeiro passo na construção do SDBMD, como já foi referido, passa pela análise das suas fontes, neste caso a base de dados *NorthWind*. No repositório do *GitHub* facultado pelos docentes existem *scripts* SQL e outros documentos que nos facultam esta análise. Um destes documentos é o diagrama ER, ou seja, o modelo lógico, da base de dados. Este diagrama pode ser consultado na figura 1 (apesar da sua reduzida dimensão neste relatório, é possível consultá-lo com maior clareza no repositório supramencionado).

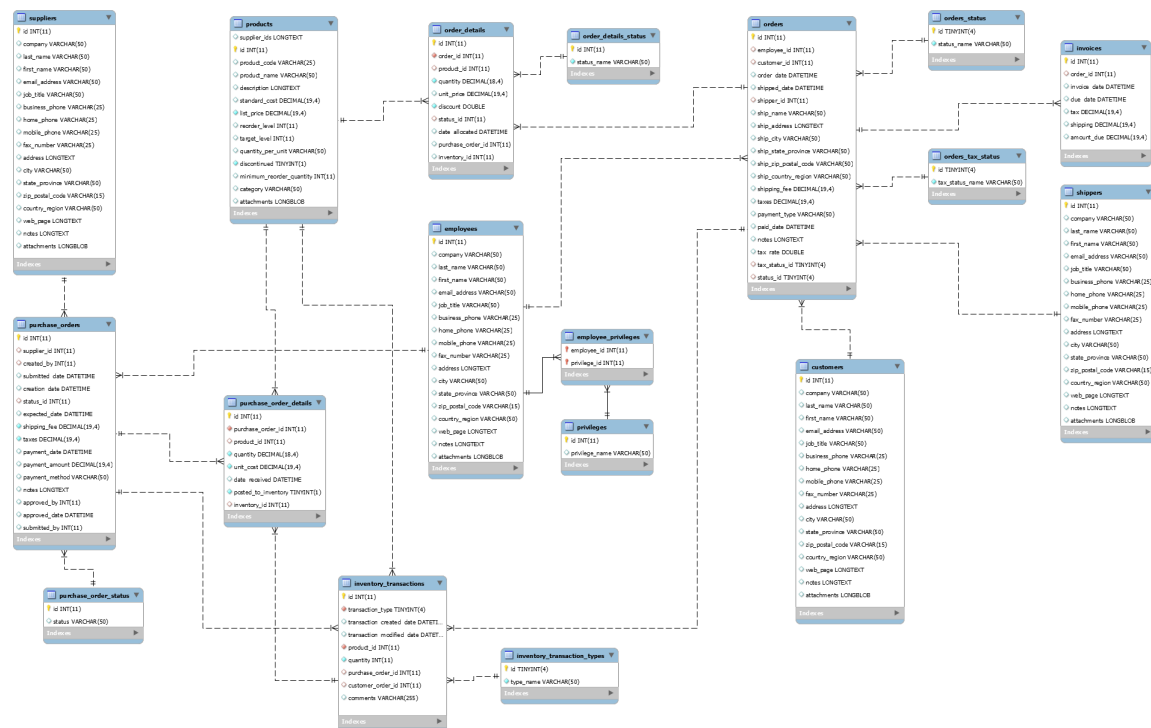


Figura 1: NorthWind - Diagrama ER

2.1 Modelo Lógico

Observando atentamente e analisando o diagrama, podemos retirar sumariamente as seguintes informações principais, que nos ajudaram imenso a compreender o modelo:

- A base de dados é composta por dezoito tabelas que, traduzindo, referem-se ao seguinte: Encomendas, Clientes, Funcionários, Transportadoras, Faturas, Transações de Inventário, Produtos, Fornecedores e Compras. Além destas, existem outras que complementam as já referidas, com informações mais detalhadas (como por exemplo estado das encomendas, quantidades e preços

dos produtos de cada encomenda ou até privilégios dos trabalhadores). Algumas delas representam tabelas auxiliares entre entidades com relações N-N (muitos-para-muitos).

- A base de dados foca-se sobretudo nas encomendas, quer entre cliente e *NorthWind* quer entre fornecedor e *NorthWind*.
- Temos, então, presentes quatro intervenientes: clientes (que compram os produtos à *NorthWind*), fornecedores (que vendem produtos à *NorthWind*), funcionários (responsáveis pelas compras/vendas) e transportadoras (que realizam o envio dos produtos vendidos aos clientes).
- Assim, temos dois tipos de troca diferentes: *Purchase Orders* (compras da *NorthWind* aos fornecedores) e *Sales Orders* (vendas da *NorthWind* aos clientes). Ora, além destas, a base de dados regista também as transações realizadas no inventário dos produtos, ou seja, as entradas e saídas de cada um.

Cada uma das entidades referidas possui um vasto número de atributos, que podem ser observados no modelo lógico e cujo nome é bastante descritivo do significado do mesmo. Resumindo, as entidades (clientes, fornecedores, transportadoras e funcionários) são caracterizadas com o nome, contactos e localização, os produtos são caracterizadas pelo seu preço, categoria e descrição, as compras com o produto, quantidade e fornecedor, as vendas com o produto, quantidade e cliente e transações de inventário com o produto e quantidade deste. Além destes, existem outros atributos, mas os principais são os que foram já referidos.

2.2 Qualidade dos Dados

Após a análise do modelo lógico da base de dados, construímos a base de dados com os *scripts* fornecidos e procedemos ao seu povoamento. Tendo já a *NorthWind* alguns valores, pudemos observar os dados e verificar qual a sua qualidade, isto é, se os dados estão com formatos diferentes em várias tabelas, se estão corrompidos ou desformatados e se são inexistentes (*null*). Desta observação retiramos o seguinte:

2.2.1 Produtos

supplier_ids	id	product_code	product_name	description	standard_cost	list_price	reorder_level	target_level	quantity_per_unit	discontinued	minimum_reorder_quantity	category	attachments
4	1	NWTR-1	Northwind Traders Chai	CHAI	13.5000	18.0000	10	40	10 boxes x 20 bags	0	10	Beverages	Beverages
10	3	NWTCO-3	Northwind Traders Syrup	SYRUP	7.5000	10.0000	25	100	12 - 550 ml bottles	0	25	Condiments	Condiments
10	4	NWTCO-4	Northwind Traders Cajun Seasoning	CAJUN	16.5000	22.0000	10	40	48 - 6 oz jars	0	10	Condiments	Condiments
10	5	NWTO-5	Northwind Traders Olive Oil	OLIVE	16.0125	21.3500	10	40	36 boxes	0	10	Oil	Oil
2/6	6	NWTP-6	Northwind Traders Boysenberry Spread	BOYSEN	18.7500	25.0000	25	100	12 - 8 oz jars	0	25	Jams, Preserves	Jams, Preserves
2	7	NWTDN-7	Northwind Traders Dried Pears	PEARS	22.5000	30.0000	10	40	12 - 1 lb pkgs.	0	10	Dried Fruit & Nuts	Dried Fruit & Nuts
8	8	NWTS-8	Northwind Traders Curry Sauce	CURRY	30.0000	40.0000	10	40	12 - 12 oz jars	0	10	Sauces	Sauces
2/6	14	NWTDN-14	Northwind Traders Walnuts	WALNUT	17.4375	23.2500	10	40	40 - 100 g pkgs.	0	10	Dried Fruit & Nuts	Dried Fruit & Nuts
6	17	NWTCV-17	Northwind Traders Fruit Cocktail	COCKTAIL	29.2500	39.0000	10	40	15.25 OZ	0	10	Canned Fruit & Vegetables	Canned Fruit & Vegetables
1	19	NWTRGM-19	Northwind Traders Chocolate Biscuits Mix	BISCUITS	6.9000	9.2000	5	20	10 boxes x 12 pieces	0	5	Baked Goods & Mixes	Baked Goods & Mixes
2/6	20	NWTP-20	Northwind Traders Marmalade	MARMALADE	60.7500	81.0000	10	40	30 gift boxes	0	10	Jams, Preserves	Jams, Preserves
1	21	NWTRGM-21	Northwind Traders Scones	SCONES	7.5000	10.0000	5	20	24 pkgs. x 4 pieces	0	5	Baked Goods & Mixes	Baked Goods & Mixes

Figura 2: Dados da tabela dos Produtos (1)

3	81	NWTB-81	Northwind Traders Green Tea	NULL	2.0000	2.9900	100	125	20 bags per box	0	25	Beverages	\$1.00
1	82	NWTC-82	Northwind Traders Granola	NULL	2.0000	4.0000	20	100	NULL	0	NULL	Cereal	\$1.00
9	83	NWTC-83	Northwind Traders Potato Chips	NULL	0.5000	1.8000	30	200	NULL	0	NULL	Chips, Snacks	\$1.00
1	85	NWTC-85	Northwind Traders Brownie Mix	NULL	9.0000	12.4900	10	20	3 boxes	0	5	Baked Goods & Mixes	\$1.00
1	86	NWTC-86	Northwind Traders Cake Mix	NULL	10.5000	15.9900	10	20	4 boxes	0	5	Baked Goods & Mixes	\$1.00
7	87	NWTC-87	Northwind Traders Tea	NULL	2.0000	4.0000	20	50	100 count per box	0	NULL	Beverages	\$1.00
6	88	NWTCFV-88	Northwind Traders Pears	NULL	1.0000	1.3000	10	40	15.25 OZ	0	NULL	Canned Fruit & Vegetables	\$1.00
6	89	NWTCFV-89	Northwind Traders Peaches	NULL	1.0000	1.5000	10	40	15.25 OZ	0	NULL	Canned Fruit & Vegetables	\$1.00
6	90	NWTCFV-90	Northwind Traders Pineapple	NULL	1.0000	1.8000	10	40	15.25 OZ	0	NULL	Canned Fruit & Vegetables	\$1.00
6	91	NWTCFV-91	Northwind Traders Cherry Pie Filling	NULL	1.0000	2.0000	10	40	15.25 OZ	0	NULL	Canned Fruit & Vegetables	\$1.00
6	92	NWTCFV-92	Northwind Traders Green Beans	NULL	1.0000	1.2000	10	40	14.5 OZ	0	NULL	Canned Fruit & Vegetables	\$1.00
6	93	NWTCFV-93	Northwind Traders Corn	NULL	1.0000	1.2000	10	40	14.5 OZ	0	NULL	Canned Fruit & Vegetables	\$1.00
6	94	NWTCFV-94	Northwind Traders Peas	NULL	1.0000	1.5000	10	40	14.5 OZ	0	NULL	Canned Fruit & Vegetables	\$1.00
7	95	NWTCM-95	Northwind Traders Tuna Fish	NULL	0.5000	2.0000	30	50	5 oz	0	NULL	Canned Meat	\$1.00
7	96	NWTCM-96	Northwind Traders Smoked Salmon	NULL	2.0000	4.0000	30	50	5 oz	0	NULL	Canned Meat	\$1.00
1	97	NWTC-82	Northwind Traders Hot Cereal	NULL	3.0000	5.0000	50	200	NULL	0	NULL	Cereal	\$1.00
6	98	NWTSO-98	Northwind Traders Vegetable Soup	NULL	1.0000	1.8900	100	200	NULL	0	NULL	Soups	\$1.00

Figura 3: Dados da tabela dos Produtos (2)

Observando os dados relativos aos Produtos, podemos verificar que estes são legíveis e não se encontram corrompidos. Podemos verificar também que alguns valores são inexistentes, faltando a descrição do produto em todos eles, a quantidade mínima para re-encomenda e a quantidade por unidade de produto.

2.2.2 Clientes

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	not
Bedece	Anna	NULL	Owner	(123)555-0100	NULL	NULL	(123)555-0101	123 1st Street	Seattle	WA	99999	USA	NULL	NULL
Gratacos Solsona	Antonio	NULL	Owner	(123)555-0100	NULL	NULL	(123)555-0101	123 2nd Street	Boston	MA	99999	USA	NULL	NULL
Axen	Thomas	NULL	Purchasing Representative	(123)555-0100	NULL	NULL	(123)555-0101	123 3rd Street	Los Angeles	CA	99999	USA	NULL	NULL
Lee	Christina	NULL	Purchasing Manager	(123)555-0100	NULL	NULL	(123)555-0101	123 4th Street	New York	NY	99999	USA	NULL	NULL
O'Donnell	Martin	NULL	Owner	(123)555-0100	NULL	NULL	(123)555-0101	123 5th Street	Minneapolis	MN	99999	USA	NULL	NULL
Pérez-Olaeta	Francisco	NULL	Purchasing Manager	(123)555-0100	NULL	NULL	(123)555-0101	123 6th Street	Milwaukee	WI	99999	USA	NULL	NULL
Xie	Ming-Yang	NULL	Owner	(123)555-0100	NULL	NULL	(123)555-0101	123 7th Street	Boise	ID	99999	USA	NULL	NULL
Andersen	Elizabeth	NULL	Purchasing Representative	(123)555-0100	NULL	NULL	(123)555-0101	123 8th Street	Portland	OR	99999	USA	NULL	NULL
Mortensen	Sven	NULL	Purchasing Manager	(123)555-0100	NULL	NULL	(123)555-0101	123 9th Street	Salt Lake City	UT	99999	USA	NULL	NULL
Wickler	Roland	NULL	Purchasing Manager	(123)555-0100	NULL	NULL	(123)555-0101	123 10th Street	Chicago	IL	99999	USA	NULL	NULL
Krschne	Peter	NULL	Purchasing Manager	(123)555-0100	NULL	NULL	(123)555-0101	123 11th Street	Miami	FL	99999	USA	NULL	NULL
Edwards	John	NULL	Purchasing Manager	(123)555-0100	NULL	NULL	(123)555-0101	123 12th Street	Las Vegas	NV	99999	USA	NULL	NULL
Ludick	Andre	NULL	Purchasing Representative	(123)555-0100	NULL	NULL	(123)555-0101	456 13th Street	Memphis	TN	99999	USA	NULL	NULL
Grlo	Carlos	NULL	Purchasing Representative	(123)555-0100	NULL	NULL	(123)555-0101	456 14th Street	Denver	CO	99999	USA	NULL	NULL
Kupkova	Helena	NULL	Purchasing Manager	(123)555-0100	NULL	NULL	(123)555-0101	456 15th Street	Honolulu	HI	99999	USA	NULL	NULL
Goldschmidt	Daniel	NULL	Purchasing Representative	(123)555-0100	NULL	NULL	(123)555-0101	456 16th Street	San Francisco	CA	99999	USA	NULL	NULL
Bagel	Jean Phill...	NULL	Owner	(123)555-0100	NULL	NULL	(123)555-0101	456 17th Street	Seattle	WA	99999	USA	NULL	NULL
Autier Miconi	Catherine	NULL	Purchasing Representative	(123)555-0100	NULL	NULL	(123)555-0101	456 18th Street	Boston	MA	99999	USA	NULL	NULL
Eggerer	Alexander	NULL	Accounting Assistant	(123)555-0100	NULL	NULL	(123)555-0101	789 19th Street	Los Angeles	CA	99999	USA	NULL	NULL

Figura 4: Dados da tabela dos Clientes

Tal como na tabela dos Produtos, os dados apresentados na dos Clientes são legíveis e não apresentam corrupção, existindo também colunas com valores inexistentes, neste caso o endereço de email, os contactos telefónicos, a página *web* e as notas.

2.2.3 Fornecedores

id	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	note
1	Supplier A	Andersen	Elizabeth A.	NULL	Sales Manager	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2	Supplier B	Weiler	Cornelia	NULL	Sales Manager	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
3	Supplier C	Kelley	Madeleine	NULL	Sales Representative	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
4	Supplier D	Sato	Naoki	NULL	Marketing Manager	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
5	Supplier E	Hernandez-Echevarria	Amaya	NULL	Sales Manager	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
6	Supplier F	Hayakawa	Satomi	NULL	Marketing Assistant	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
7	Supplier G	Glasson	Stuart	NULL	Marketing Manager	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
8	Supplier H	Dunton	Bryn Paul	NULL	Sales Representative	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
9	Supplier I	Sandberg	Mikael	NULL	Sales Manager	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10	Supplier J	Sousa	Luis	NULL	Sales Manager	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 5: Dados da tabela dos Fornecedores

Passando para a tabela dos fornecedores, podemos constatar que maior parte das colunas possuem apenas valores inexistentes, exceto o nome da empresa, o primeiro e último nome do responsável pela mesma, assim como a sua função dentro desta.

2.2.4 Transportadoras

id	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
1	Shipping Company A	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	123 Any Street	Memphis	TN	99999	USA	NULL	NULL
2	Shipping Company B	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	123 Any Street	Memphis	TN	99999	USA	NULL	NULL
3	Shipping Company C	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	123 Any Street	Memphis	TN	99999	USA	NULL	NULL

Figura 6: Dados da tabela das Transportadoras

O comportamento dos dados da tabela das Transportadoras é muito semelhante ao da tabela dos Fornecedores, isto é, os dados são legíveis, mas com uma quantidade substancial de colunas com valores nulos, como podemos verificar na figura acima. Neste caso, apenas o nome da empresa e os dados relativos à sua localização estão disponíveis.

2.2.5 Funcionários

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
Nancy	nancy@northwindtraders.com	Sales Representative	(123)555-0100	(123)555-0102	NULL	(123)555-0103	123 1st Avenue	Seattle	WA	99999	USA	http://northwindtraders.com#	NULL
Andrew	andrew@northwindtraders.com	Vice President, Sales	(123)555-0100	(123)555-0102	NULL	(123)555-0103	123 2nd Avenue	Bellevue	WA	99999	USA	http://northwindtraders.com#http://northwindt...	Joined th
Jan	jan@northwindtraders.com	Sales Representative	(123)555-0100	(123)555-0102	NULL	(123)555-0103	123 3rd Avenue	Redmond	WA	99999	USA	http://northwindtraders.com#http://northwindt...	Was hirec
Mariya	mariya@northwindtraders.com	Sales Representative	(123)555-0100	(123)555-0102	NULL	(123)555-0103	123 4th Avenue	Kirkland	WA	99999	USA	http://northwindtraders.com#http://northwindt...	NULL
Steven	steven@northwindtraders.com	Sales Manager	(123)555-0100	(123)555-0102	NULL	(123)555-0103	123 5th Avenue	Seattle	WA	99999	USA	http://northwindtraders.com#http://northwindt...	Joined th
Michael	michael@northwindtraders.com	Sales Representative	(123)555-0100	(123)555-0102	NULL	(123)555-0103	123 6th Avenue	Redmond	WA	99999	USA	http://northwindtraders.com#http://northwindt...	Fluent in
Robert	robert@northwindtraders.com	Sales Representative	(123)555-0100	(123)555-0102	NULL	(123)555-0103	123 7th Avenue	Seattle	WA	99999	USA	http://northwindtraders.com#http://northwindt...	NULL
Laura	laura@northwindtraders.com	Sales Coordinator	(123)555-0100	(123)555-0102	NULL	(123)555-0103	123 8th Avenue	Redmond	WA	99999	USA	http://northwindtraders.com#http://northwindt...	Reads an
Anne	anne@northwindtraders.com	Sales Representative	(123)555-0100	(123)555-0102	NULL	(123)555-0103	123 9th Avenue	Seattle	WA	99999	USA	http://northwindtraders.com#http://northwindt...	Fluent in

Figura 7: Dados da tabela dos Funcionários

A tabela dos Funcionários está bastante completa, com apenas uma coluna só com valores nulos (número telefónico) e outra com alguns (notas). Os dados não aparentam estar corrompidos e são facilmente legíveis.

2.2.6 Vendas

te	shipped_date	shipper_id	ship_name	ship_address	ship_city	ship_state_province	ship_zip_postal_code	ship_country_region	shipping_fee	taxes	payment_type	paid_date	notes	tax_rate	tax_status_id	status
10 00:00:00	2006-01-31 00:00:00	3	Elizabeth Andersen	123 8th Street	Portland	OR	99999	USA	50.0000	0.0000	Credit Card	2006-01-30 00:00:00	NULL	0	NULL	3
16 00:00:00	2006-02-07 00:00:00	3	Christa Lee	123 4th Street	New York	NY	99999	USA	4.0000	0.0000	Check	2006-02-06 00:00:00	NULL	0	NULL	3
10 00:00:00	2006-02-12 00:00:00	2	Soo Jung Lee	789 29th Street	Denver	CO	99999	USA	7.0000	0.0000	Check	2006-02-10 00:00:00	NULL	0	NULL	3
13 00:00:00	2006-02-25 00:00:00	2	Thomas Aven	123 3rd Street	Los Angeles	CA	99999	USA	7.0000	0.0000	Cash	2006-02-23 00:00:00	NULL	0	NULL	3
16 00:00:00	2006-03-09 00:00:00	2	Francisco Pérez-Oleata	123 6th Street	Milwaukee	WI	99999	USA	12.0000	0.0000	Credit Card	2006-03-06 00:00:00	NULL	0	NULL	3
10 00:00:00	2006-03-11 00:00:00	3	Amritansh Raghav	789 28th Street	Memphis	TN	99999	USA	10.0000	0.0000	Check	2006-03-10 00:00:00	NULL	0	NULL	3
12 00:00:00	2006-03-24 00:00:00	3	Elizabeth Andersen	123 8th Street	Portland	OR	99999	USA	5.0000	0.0000	Check	2006-03-22 00:00:00	NULL	0	NULL	3
14 00:00:00	2006-03-24 00:00:00	2	Roland Wacker	123 10th Street	Chicago	IL	99999	USA	9.0000	0.0000	Credit Card	2006-03-24 00:00:00	NULL	0	NULL	3
14 00:00:00	2006-03-24 00:00:00	2	Wing-Feng Lie	123 7th Street	Boise	ID	99999	USA	0.0000	0.0000	NULL	NULL	NULL	0	NULL	0
14 00:00:00	2006-04-07 00:00:00	1	Roland Wacker	123 10th Street	Chicago	IL	99999	USA	0.0000	0.0000	NULL	NULL	NULL	0	NULL	2
14 00:00:00	2006-04-07 00:00:00	3	Peter Kirschne	123 11th Street	Miami	FL	99999	USA	0.0000	0.0000	NULL	NULL	NULL	0	NULL	0
14 00:00:00	2006-04-07 00:00:00	3	Anna Bedecs	123 1st Street	Seattle	WA	99999	USA	0.0000	0.0000	NULL	NULL	NULL	0	NULL	0
17 00:00:00	2006-04-07 00:00:00	3	Amritansh Raghav	789 28th Street	Memphis	TN	99999	USA	40.0000	0.0000	Credit Card	2006-04-07 00:00:00	NULL	0	NULL	3
15 00:00:00	2006-04-05 00:00:00	1	Sven Mortensen	123 9th Street	Salt Lake City	UT	99999	USA	100.0000	0.0000	Check	2006-04-05 00:00:00	NULL	0	NULL	3
18 00:00:00	2006-04-08 00:00:00	2	Francisco Pérez-Oleata	123 6th Street	Milwaukee	WI	99999	USA	300.0000	0.0000	Credit Card	2006-04-08 00:00:00	NULL	0	NULL	3
15 00:00:00	2006-04-05 00:00:00	2	Elizabeth Andersen	123 8th Street	Portland	OR	99999	USA	50.0000	0.0000	Check	2006-04-05 00:00:00	NULL	0	NULL	3
15 00:00:00	2006-04-05 00:00:00	1	John Rodman	789 25th Street	Chicago	IL	99999	USA	5.0000	0.0000	Cash	2006-04-05 00:00:00	NULL	0	NULL	3
15 00:00:00	2006-04-05 00:00:00	3	Run Liu	789 26th Street	Miami	FL	99999	USA	60.0000	0.0000	Credit Card	2006-04-05 00:00:00	NULL	0	NULL	3
15 00:00:00	2006-04-05 00:00:00	2	Soo Jung Lee	789 29th Street	Denver	CO	99999	USA	200.0000	0.0000	Check	2006-04-05 00:00:00	NULL	0	NULL	3
13 00:00:00	2006-04-03 00:00:00	3	Francisco Pérez-Oleata	123 6th Street	Milwaukee	WI	99999	USA	0.0000	0.0000	Check	2006-04-03 00:00:00	NULL	0	NULL	3

Figura 8: Dados da tabela das Vendas

Analisando os dados da tabela das Vendas, podemos verificar que já existe mais alguma complexidade: existem colunas com valores nulos dependendo do estado das mesmas, ou seja, uma venda que não tenha data de expedição é porque ainda não foi efetuada e, portanto, não apresenta também informações acerca do pagamento e o seu estado é igual a 0 (*none*); porém, quando o estado é igual a 2 (*invoiced*, ou seja, faturada), a venda tem data de envio mas não tem informação acerca do pagamento.

Além destes campos vazios, existem também duas colunas que apresentam valores nulos para todas as encomendas: notas e estado do imposto.

2.2.7 Compras

id	supplier_id	created_by	submitted_date	creation_date	status_id	expected_date	shipping_fee	taxes	payment_date	payment_amount	payment_method	notes	approved_by	approved_date
100	2	9	2006-01-14 00:00:00	2006-01-22 00:00:00	2	NULL	0.0000	0.0000	NULL	0.0000	NULL	Purchase generated based on Order #39	2	2006-01-22 00:00:00
101	1	2	2006-01-14 00:00:00	2006-01-22 00:00:00	2	NULL	0.0000	0.0000	NULL	0.0000	NULL	Purchase generated based on Order #40	2	2006-01-22 00:00:00
102	1	1	2006-03-24 00:00:00	2006-03-24 00:00:00	2	NULL	0.0000	0.0000	NULL	0.0000	NULL	Purchase generated based on Order #41	2	2006-04-04 00:00:00
103	2	1	2006-03-24 00:00:00	2006-03-24 00:00:00	2	NULL	0.0000	0.0000	NULL	0.0000	NULL	Purchase generated based on Order #42	2	2006-04-04 00:00:00
104	2	1	2006-03-24 00:00:00	2006-03-24 00:00:00	2	NULL	0.0000	0.0000	NULL	0.0000	NULL	Purchase generated based on Order #45	2	2006-04-04 00:00:00
105	5	7	2006-03-24 00:00:00	2006-03-24 00:00:00	2	NULL	0.0000	0.0000	NULL	0.0000	Check	Purchase generated based on Order #46	2	2006-04-04 00:00:00
106	6	7	2006-03-24 00:00:00	2006-03-24 00:00:00	2	NULL	0.0000	0.0000	NULL	0.0000	NULL	Purchase generated based on Order #46	2	2006-04-04 00:00:00
107	1	6	2006-03-24 00:00:00	2006-03-24 00:00:00	2	NULL	0.0000	0.0000	NULL	0.0000	NULL	Purchase generated based on Order #47	2	2006-04-04 00:00:00
108	2	4	2006-03-24 00:00:00	2006-03-24 00:00:00	2	NULL	0.0000	0.0000	NULL	0.0000	NULL	Purchase generated based on Order #48	2	2006-04-04 00:00:00
109	2	4	2006-03-24 00:00:00	2006-03-24 00:00:00	2	NULL	0.0000	0.0000	NULL	0.0000	NULL	Purchase generated based on Order #48	2	2006-04-04 00:00:00
110	1	3	2006-03-24 00:00:00	2006-03-24 00:00:00	2	NULL	0.0000	0.0000	NULL	0.0000	NULL	Purchase generated based on Order #49	2	2006-04-04 00:00:00
111	1	2	2006-03-31 00:00:00	2006-03-31 00:00:00	2	NULL	0.0000	0.0000	NULL	0.0000	NULL	Purchase generated based on Order #56	2	2006-04-04 00:00:00
140	6	NULL	2006-04-25 00:00:00	2006-04-25 16:40:51	2	NULL	0.0000	0.0000	NULL	0.0000	NULL	NULL	2	2006-04-25 16:40:51
141	8	NULL	2006-04-25 00:00:00	2006-04-25 17:10:35	2	NULL	0.0000	0.0000	NULL	0.0000	NULL	NULL	2	2006-04-25 17:10:35
142	8	NULL	2006-04-25 00:00:00	2006-04-25 17:18:29	2	NULL	0.0000	0.0000	NULL	0.0000	Check	NULL	2	2006-04-25 17:18:29
146	2	2	2006-04-26 18:26:37	2006-04-26 18:26:37	1	NULL	0.0000	0.0000	NULL	0.0000	NULL	NULL	NULL	NULL
147	7	2	2006-04-26 18:33:28	2006-04-26 18:33:28	1	NULL	0.0000	0.0000	NULL	0.0000	NULL	NULL	NULL	NULL
148	5	2	2006-04-26 18:33:52	2006-04-26 18:33:52	1	NULL	0.0000	0.0000	NULL	0.0000	NULL	NULL	NULL	NULL

Figura 9: Dados da tabela das Compras

Os dados da tabela das Compras possuem um comportamento semelhante ao da tabela dos Funcionários: dados legíveis, mas apresentando colunas com valores nulos predominantes e colunas onde estes estão em minoria. Um exemplo destes dados pode ser observado na figura acima.

2.2.8 Inventário

id	transaction_type	transaction_created_date	transaction_modified_date	product_id	quantity	purchase_order_id	customer_order_id	comments
68	2	2006-03-22 16:10:06	2006-03-22 16:10:27	43	20	NULL	NULL	NULL
69	2	2006-03-22 16:11:39	2006-03-24 11:00:55	19	20	NULL	NULL	NULL
70	2	2006-03-22 16:11:56	2006-03-24 10:59:41	48	10	NULL	NULL	NULL
71	2	2006-03-22 16:12:29	2006-03-24 10:57:38	8	17	NULL	NULL	NULL
72	1	2006-03-24 10:41:30	2006-03-24 10:41:30	81	200	NULL	NULL	NULL
73	2	2006-03-24 10:41:33	2006-03-24 10:41:42	81	200	NULL	NULL	Fill Back Ordered pro...
74	1	2006-03-24 10:53:13	2006-03-24 10:53:13	48	100	NULL	NULL	NULL
75	2	2006-03-24 10:53:16	2006-03-24 10:55:46	48	100	NULL	NULL	Fill Back Ordered pro...
76	1	2006-03-24 10:53:36	2006-03-24 10:53:36	43	300	NULL	NULL	NULL
77	2	2006-03-24 10:53:39	2006-03-24 10:56:57	43	300	NULL	NULL	Fill Back Ordered pro...
78	1	2006-03-24 10:54:04	2006-03-24 10:54:04	41	200	NULL	NULL	NULL
79	2	2006-03-24 10:54:07	2006-03-24 10:58:40	41	200	NULL	NULL	Fill Back Ordered pro...
80	1	2006-03-24 10:54:33	2006-03-24 10:54:33	19	30	NULL	NULL	NULL
81	2	2006-03-24 10:54:35	2006-03-24 11:02:02	19	30	NULL	NULL	Fill Back Ordered pro...
82	1	2006-03-24 10:54:58	2006-03-24 10:54:58	34	100	NULL	NULL	NULL
83	2	2006-03-24 10:55:02	2006-03-24 11:03:00	34	100	NULL	NULL	Fill Back Ordered pro...
84	2	2006-03-24 14:48:15	2006-04-04 11:41:14	6	10	NULL	NULL	NULL
85	2	2006-03-24 14:48:23	2006-04-04 11:41:14	4	10	NULL	NULL	NULL
86	3	2006-03-24 14:49:16	2006-03-24 14:49:16	80	20	NULL	NULL	NULL
87	3	2006-03-24 14:49:20	2006-03-24 14:49:20	81	50	NULL	NULL	NULL

Figura 10: Dados da tabela de Transações de Inventário

Por fim, olhando para a tabela relativa às Transações de Inventário, podemos verificar que apenas duas colunas são compostas exclusivamente por valores nulos (identificadores de compra e venda). Porém, a coluna dos comentários apresenta maioritariamente *nulls*.

3 Planeamento do Data Warehouse

Com estas informações, podemos começar a pensar quais as métricas que podemos retirar do SDBMD para os gestores da *NorthWind* e em que poderá ajudá-los na tomada de decisões.

3.1 Métricas e requisitos

Discutindo em grupo e pondo-nos na perspetiva dos gestores, definimos os seguintes requisitos a serem respondidos:

- Quantidade total e quantidade média das compras por fornecedor
- Valor total e valor médio das compras por fornecedor
- Quantidade de cada produto vendido por fornecedor
- Produtos mais comprados
- Quantidade total e quantidade média das compras por cliente
- Valor total e valor médio das compras por cliente
- Número total e médio de vendas por país e cidade
- Produtos mais vendidos
- Quantidade e valor de compras e vendas por mês
- Quantidade e valor de compras e vendas por categoria de produto
- Quantidade e valor de compras e vendas por mês e categoria de produto
- Número de compras e vendas por dia da semana
- Funcionários que mais venderam

3.2 Modelo do Data Warehouse

De maneira a podermos responder a cada um dos requisitos supramencionados, desenvolvemos o seguinte modelo para o Data Warehouse:

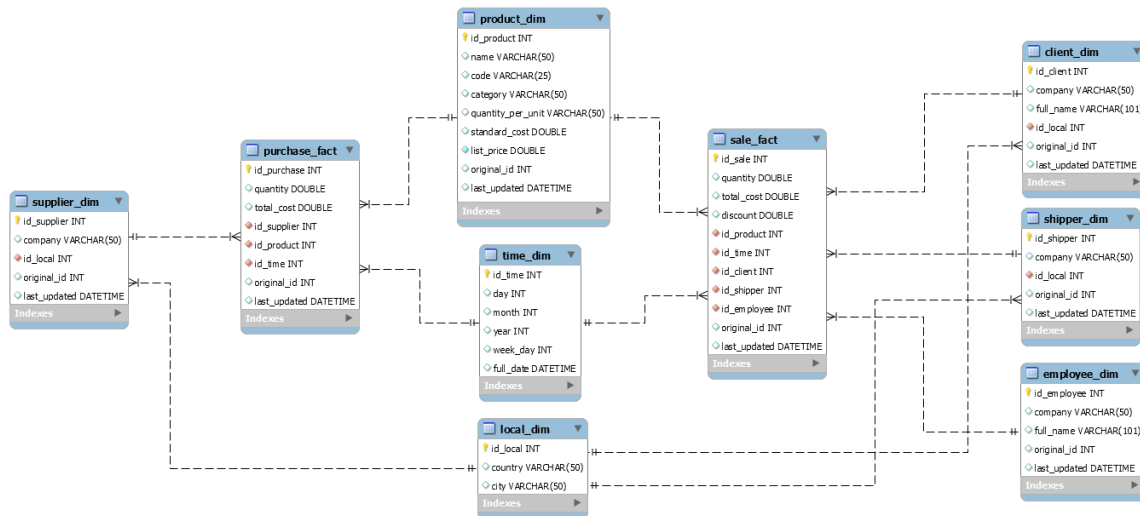


Figura 11: Modelo Lógico do Data Warehouse

Como é possível observar, este segue o esquema de Constelação de Factos, onde constam os seguintes factos e dimensões:

Factos:

- *Purchase*: compras aos fornecedores
- *Sale*: vendas aos clientes

Dimensões:

- *Client*: clientes que efetuam compras à empresa
- *Supplier*: fornecedores que vendem produtos à empresa
- *Shipper*: transportadores que entregam os produtos aos clientes
- *Employee*: funcionários que executam as vendas
- *Product*: produtos comercializados
- *Time*: data das compras/vendas, por dia, mês, ano e dia da semana
- *Local*: localidade (país e cidade) dos fornecedores, clientes e transportadoras

Caso quisessemos tratar apenas as vendas aos clientes, um esquema em Estrela seria suficiente, mas usando uma constelação de factos permite-nos ter a flexibilidade de tratar também as compras aos fornecedores, apesar de trazer maior complexidade ao sistema.

Além disso, adicionamos um *timestamp* relativo à inserção/última atualização no DW e o ID original de cada *row* nas tabelas de fornecedor, compras e vendas, clientes, produtos, funcionários e transportadoras.

Para facilitar o povoamento do Data Warehouse, utilizamos uma *staging area*, que tem um modelo muito semelhante ao do DW. É nesta *staging area* (SA) que iremos também tratar os valores em falta (null) existentes na base de dados original e substituí-los por valores legíveis, como por exemplo "Unknown".

O modelo que desenvolvemos para a *staging area* é o seguinte:

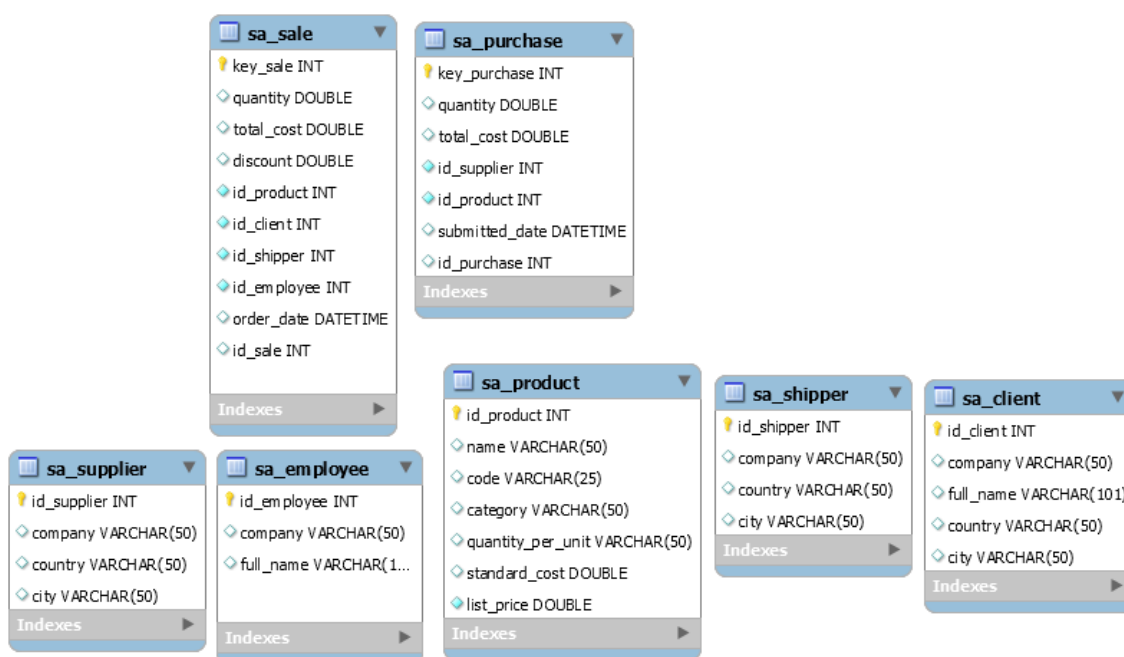


Figura 12: Modelo Lógico da Staging Area

Ou seja, existe uma tabela para cada entidade interveniente e uma para as compras e as vendas. Estas duas últimas têm como chave primária um valor autoincremental, enquanto as restantes tabelas das entidades possuem como chave primária o seu ID natural da tabela original da *Northwind*. Isto porque desta forma conseguimos tratar mais facilmente o povoamento do Data Warehouse. Além disso, como algumas vendas não possuem um identificador de Transportadora, este foi adicionado como sendo "0", logo teremos que adicionar um novo registo na tabela das Transportadoras, onde o seu ID é 0 e o resto dos campos é "Unknown".

4 Implementação: ETL

4.1 Mapas Lógicos de Dados

Para facilitar o povoamento do Data Warehouse e da Staging Area, definimos os seguintes mapas lógicos de dados, apresentados em formato vertical nas páginas que se seguem:

Tabella	Destino			Origem			Transformação	
Tabella	Coluna	Tipo de Dados	Tipo de Tabella	SCD	BD	Tabella	Coluna	Tipo de Dados
sa.supplier	id_supplier	INT	Staging Area	1	Northwind	suppliers	id	INT
sa.supplier	company	VARCHAR(50)	Staging Area	1	Northwind	suppliers	company	VARCHAR(50)
sa.supplier	country	INT	Staging Area	1	Northwind	suppliers	country_region	VARCHAR(50)
sa.supplier	city	INT	Staging Area	1	Northwind	suppliers	city	VARCHAR(50)
sa.purchase	key_purchase	INT	Staging Area	1	Northwind	-	-	-
sa.purchase	id_purchase	INT	Staging Area	1	Northwind	purchase_orders	id	INT
sa.purchase	quantity	DOUBLE	Staging Area	1	Northwind	purchase_order_details	quantity	DECIMAL(18,4)
sa.purchase	total_cost	DOUBLE	Staging Area	1	Northwind	purchase_order_details	quantity, unit_cost	DOUBLE
sa.purchase	id_supplier	INT	Staging Area	1	Northwind	purchase_orders	supplier_id	INT
sa.purchase	id_product	INT	Staging Area	1	Northwind	purchase_order_details	product_id	INT
sa.purchase	submitted_date	DATE/TIME	Staging Area	1	Northwind	purchase_orders	submitted_date	-
sa.product	id_product	INT	Staging Area	1	Northwind	products	id_product	INT
sa.product	name	VARCHAR(50)	Staging Area	1	Northwind	products	product_name	VARCHAR(50)
sa.product	code	VARCHAR(25)	Staging Area	1	Northwind	products	product_code	VARCHAR(50)
sa.product	category	VARCHAR(50)	Staging Area	1	Northwind	products	category	VARCHAR(50)
sa.product	quantity_per_unit	VARCHAR(50)	Staging Area	1	Northwind	products	quantity_per_unit	VARCHAR(50)
sa.product	standard_cost	DOUBLE	Staging Area	1	Northwind	products	standard_cost	DECIMAL(19,4)
sa.product	list_price	DOUBLE	Staging Area	1	Northwind	products	list_price	DECIMAL(19,4)
sa.client	id_client	INT	Staging Area	1	Northwind	customers	id	INT
sa.client	company	VARCHAR(50)	Staging Area	1	Northwind	customers	company	VARCHAR(50)
sa.client	full_name	VARCHAR(101)	Staging Area	1	Northwind	customers	first_name, last_name	VARCHAR(50)
sa.client	country_region	VARCHAR(50)	Staging Area	1	Northwind	customers	country_region	VARCHAR(50)
sa.shipper	city	VARCHAR(50)	Staging Area	1	Northwind	shippers	city	VARCHAR(50)
sa.shipper	id_shipper	INT	Staging Area	1	Northwind	shippers	id	INT
sa.shipper	company	VARCHAR(50)	Staging Area	1	Northwind	shippers	company	VARCHAR(50)
sa.shipper	country_region	VARCHAR(50)	Staging Area	1	Northwind	shippers	country_region	VARCHAR(50)
sa.shipper	city	VARCHAR(50)	Staging Area	1	Northwind	shippers	city	VARCHAR(50)
sa.sale	key_sale	INT	Staging Area	1	Northwind	-	-	-
sa.sale	quantity	DOUBLE	Staging Area	1	Northwind	order_details	quantity	DECIMAL(19,4)
sa.sale	total_cost	DOUBLE	Staging Area	1	Northwind	order_details	quantity, unit_price	DECIMAL(18,4)
sa.sale	discount	DOUBLE	Staging Area	1	Northwind	order_details	discount	DOUBLE
sa.sale	id_product	INT	Staging Area	1	Northwind	order_details	product_id	INT
sa.sale	order_date	DATE/TIME	Staging Area	1	Northwind	orders	order_date	DATE/TIME
sa.sale	id_client	INT	Staging Area	1	Northwind	orders	customer_id	INT
sa.sale	id_shipper	INT	Staging Area	1	Northwind	orders	shipper_id	INT
sa.sale	id_employee	INT	Staging Area	1	Northwind	orders	employee_id	INT
sa.sale	original_id	INT	Staging Area	1	Northwind	orders	id	INT
sa.employee	id_employee	INT	Staging Area	1	Northwind	employees	id	INT
sa.employee	company	VARCHAR(50)	Staging Area	1	Northwind	employees	company	VARCHAR(50)
sa.employee	full_name	VARCHAR(101)	Staging Area	1	Northwind	employees	first_name, last_name	VARCHAR(50)

Tabela 1: Mapa Lógico de Dados da Staging Area

Destino					Origem				Transformação
Tabela	Coluna	Tipo de Dados	Tipo de Tabela	SCD	BD	Tabela	Coluna	Tipo de Dados	
supplier_dim	key_supplier	INT	Dimensão	1	-	-	-	-	Surrogate Key
supplier_dim	id_supplier	INT	Dimensão	1	Staging Area	sa_supplier	id_supplier	INT	Natural Key
supplier_dim	company	VARCHAR(50)	Dimensão	1	Staging Area	sa_supplier	company	VARCHAR(50)	select company from sa_supplier
supplier_dim	key_local	INT	Dimensão	1	-	-	-	-	select l.key_local from local_dim l, sa_supplier s where l.country = s.country and l.city = s.city
supplier_dim	last_updated	DATETIME	Dimensão	1	-	-	-	-	now()
purchase_fact	key_purchase	INT	Factos	1	-	-	-	-	Surrogate Key
purchase_fact	id_purchase	INT	Factos	1	Staging Area	sa_purchase	id_purchase	INT	Natural Key
purchase_fact	quantity	DECIMAL(18,4)	Factos	1	Staging Area	sa_purchase	quantity	DECIMAL(18,4)	select quantity from sa_purchase
purchase_fact	total_cost	DOUBLE	Factos	1	Staging Area	sa_purchase	total_cost	DOUBLE	select total_cost from sa_purchase
purchase_fact	key_supplier	INT	Factos	1	Staging Area	sa_purchase	id_supplier	INT	select sd.key_supplier from supplier_dim sd, sa_purchase p where sd.id_supplier=p.id_supplier
purchase_fact	key_product	INT	Factos	1	Staging Area	sa_purchase	id_product	INT	select pd.key_product from product_dim pd, sa_purchase p where pd.id_product=p.id_product
purchase_fact	key_time	INT	Factos	1	-	-	-	-	select t.key_time from time_dim t, sa_purchase p where t.full_date = p.submitted_date
purchase_fact	last_updated	DATETIME	Factos	1	-	-	-	-	now()
product_dim	key_product	INT	Dimensão	1	-	-	-	-	Surrogate Key
product_dim	name	VARCHAR(50)	Dimensão	1	Staging Area	sa_product	name	VARCHAR(50)	select name from sa_product
product_dim	code	VARCHAR(50)	Dimensão	1	Staging Area	sa_product	code	VARCHAR(50)	select code from sa_product
product_dim	category	VARCHAR(50)	Dimensão	1	Staging Area	sa_product	category	VARCHAR(50)	select category from sa_product
product_dim	quantity_per_unit	VARCHAR(50)	Dimensão	1	Staging Area	sa_product	quantity_per_unit	VARCHAR(50)	select quantity per unit from sa_product
product_dim	standard_cost	DOUBLE	Dimensão	1	Staging Area	sa_product	standard_cost	DOUBLE	select standard_cost from sa_product
product_dim	list_price	DOUBLE	Dimensão	1	Staging Area	sa_product	list_price	DOUBLE	select list_price from sa_product
product_dim	id_product	INT	Dimensão	1	Staging Area	sa_product	id_product	INT	select id_product from sa_product
product_dim	last_updated	DATETIME	Dimensão	1	-	-	-	-	now()
client_dim	key_client	INT	Dimensão	1	-	-	-	-	Surrogate Key
client_dim	company	VARCHAR(50)	Dimensão	1	Staging Area	sa_client	company	VARCHAR(50)	select company from sa_client
client_dim	full_name	VARCHAR(101)	Dimensão	1	Staging Area	sa_client	full_name	VARCHAR(101)	select full_name from sa_client
client_dim	key_local	INT	Dimensão	1	-	-	-	-	select l.key_local from local_dim l, sa_client c where l.country = c.country and l.city = c.city
client_dim	id_client	INT	Dimensão	1	Staging Area	sa_client	id_client	INT	select id_client from sa_client
client_dim	last_updated	DATETIME	Dimensão	1	-	-	-	-	now()
time_dim	key_time	INT	Dimensão	1	-	-	-	-	Surrogate Key
time_dim	full_date	DATETIME	Dimensão	1	Staging Area	sa_purchase, sa_order	-	-	select submitted_date from sa_purchases union distinct select order_date from sa_orders
time_dim	day	INT	Dimensão	1	Staging Area	-	-	-	select day(full_time) from time_dim
time_dim	month	INT	Dimensão	1	Staging Area	-	-	-	select month(full_time) from time_dim
time_dim	year	INT	Dimensão	1	Staging Area	-	-	-	select year(full_time) from time_dim
time_dim	week_day	INT	Dimensão	1	Staging Area	-	-	-	select weekday(full_time) from time_dim
local_dim	key_local	INT	Dimensão	1	-	-	-	-	Surrogate Key
local_dim	country	VARCHAR(50)	Dimensão	1	Staging Area	sa_supplier, sa_cliente, sa_shipper	country	VARCHAR(50)	select country_region from sa_supplier union distinct select country_region from sa_client union distinct select country_region from sa_shipper
local_dim	city	VARCHAR(50)	Dimensão	1	Staging Area	sa_supplier, sa_cliente, sa_shipper	city	VARCHAR(50)	select city from sa_supplier union distinct select city from sa_client union distinct select city from sa_shipper
shipper_dim	key_shipper	INT	Dimensão	1	-	-	-	-	Surrogate Key
shipper_dim	company	VARCHAR(50)	Dimensão	1	Staging Area	sa_shipper	company	VARCHAR(50)	select company from sa_shipper
shipper_dim	key_local	INT	Dimensão	1	Staging Area	sa_shipper	id_local	INT	select l.key_local from local_dim l, sa_shipper s where l.country = s.country and l.city = s.city
shipper_dim	id_shipper	INT	Dimensão	1	Staging Area	sa_shipper	id_shipper	INT	select id_shipper from sa_shipper
shipper_dim	last_updated	DATETIME	Dimensão	1	-	-	-	-	now()
sale_fact	key_sale	INT	Factos	1	-	-	-	-	Surrogate Key
sale_fact	quantity	DOUBLE	Factos	1	Staging Area	sa_sale	quantity	DOUBLE	select quantity from sa_sale
sale_fact	total_cost	DOUBLE	Factos	1	Staging Area	sa_sale	total_cost	DOUBLE	select total_cost from sa_sale
sale_fact	discount	DOUBLE	Factos	1	Staging Area	sa_sale	discount	DOUBLE	select discount from sa_sale
sale_fact	key_product	INT	Factos	1	Staging Area	sa_sale	id_product	INT	select pd.key_product from product_dim pd, sa_sale s where pd.id_product = s.id_product
sale_fact	key_time	INT	Factos	1	Staging Area	sa_sale	id_time	INT	select t.key_time from time_dim t, sa_sale s where t.full_date = s.order_date
sale_fact	key_client	INT	Factos	1	Staging Area	sa_sale	id_client	INT	select key_client from client_dim cd, sa_sale s where cd.id_client = s.id_client
sale_fact	key_shipper	INT	Factos	1	Staging Area	sa_sale	id_shipper	INT	select key_shipper from shipper_dim sd, sa_sale s where sd.id_shipper = s.id_shipper
sale_fact	key_employee	INT	Factos	1	Staging Area	sa_sale	id_employee	INT	select key_employee from employee_dim ed, sa_sale s where ed.id_employee = s.id_employee
sale_fact	id_sale	INT	Factos	1	Staging Area	sa_sale	id_sale	INT	select id_sale from sa_sale
sale_fact	last_updated	DATETIME	Factos	1	-	-	-	-	now()
employee_dim	id_employee	INT	Dimensão	1	-	-	-	-	Surrogate Key
employee_dim	company	VARCHAR(50)	Dimensão	1	Staging Area	sa_employee	company	VARCHAR(50)	select company from sa_employee
employee_dim	full_name	VARCHAR(101)	Dimensão	1	Staging Area	sa_employee	full_name	VARCHAR(101)	select full_name from sa_employee
employee_dim	id_employee	INT	Dimensão	1	Staging Area	sa_employee	id_employee	INT	select id_employee from sa_employee
employee_dim	last_updated	DATETIME	Dimensão	1	-	-	-	-	now()

Tabela 2: Mapa Lógico de Dados do Data Warehouse

4.2 Povoamento do Data Warehouse

Para o povoamento da Staging Area e do Data Warehouse, desenvolvemos dois *scripts* SQL que permitem ir à base de dados *Northwind* e seleccionarmos os campos pretendidos. Estes scripts são corridos após a criação das bases de dados da *staging area* e do *data warehouse*, cujos nomes foram definidos como **sa_northwind**

e **dw_northwind**.

O script que desenvolvemos para povoar a Staging Area com os dados das tabelas originais foi o seguinte:

- **load_sa.sql**

```
USE sa_northwind;

INSERT INTO sa_client(company, full_name, country, city, id_client)
SELECT company, (CONCAT(first_name, ' ', last_name)), country_region,
city, id
FROM northwind.customers;

INSERT INTO sa_employee(company, full_name, id_employee)
SELECT company, (CONCAT(first_name, ' ', last_name)), id
FROM northwind.employees;

INSERT INTO sa_product(name, code, category, quantity_per_unit,
standard_cost, list_price, id_product)
SELECT product_name, product_code, category, COALESCE(quantity_per_unit,
'Unknown'), standard_cost, list_price, id
FROM northwind.products;

INSERT INTO sa_purchase(quantity, total_cost, id_supplier, id_product,
submitted_date, id_purchase, id_purchase_details)
SELECT pod.quantity, pod.quantity*pod.unit_cost, po.supplier_id, pod.
product_id, po.submitted_date, po.id, pod.id
FROM northwind.purchase_orders po, northwind.purchase_order_details pod
WHERE po.id = pod.purchase_order_id;

INSERT INTO sa_sale(quantity, total_cost, discount, id_product,
id_client, id_shipper, id_employee, order_date, id_sale,
id_sale_details)
SELECT od.quantity, od.quantity * od.unit_price, od.discount, od.
product_id, o.customer_id, COALESCE(o.shipper_id, 0), o.employee_id,
o.order_date, o.id, od.id
FROM northwind.orders o, northwind.order_details od
WHERE o.id = od.order_id;

INSERT INTO sa_shipper(company, country, city, id_shipper)
SELECT company, country_region, city, id
FROM northwind.shippers;
```

```
INSERT INTO sa_supplier(company, country, city, id_supplier)
SELECT company, COALESCE(country_region, 'Unknown'), COALESCE(city, '
    Unknown'), id
FROM northwind.suppliers;
```

Por sua vez, o script desenvolvido para povoar o Data Warehouse, partindo da Staging Area, é o que segue:

- **load_dw.sql**

```
USE dw_northwind;

INSERT INTO local_dim(country, city)
SELECT country, city FROM sa_northwind.sa_supplier
UNION DISTINCT SELECT country, city FROM sa_northwind.sa_client
UNION DISTINCT SELECT country, city FROM sa_northwind.sa_shipper;

INSERT INTO time_dim(full_date, day, month, year, week_day)
SELECT submitted_date, DAY(submitted_date), MONTH(submitted_date), YEAR
    (submitted_date), WEEKDAY(submitted_date) FROM sa_northwind.
    sa_purchase
UNION DISTINCT SELECT order_date, DAY(order_date), MONTH(order_date),
    YEAR(order_date), WEEKDAY(order_date) FROM sa_northwind.sa_sale;

INSERT INTO client_dim(company, full_name, key_local, id_client)
SELECT c.company, c.full_name, l.key_local, c.id_client
FROM sa_northwind.sa_client c, local_dim l
WHERE l.country = c.country AND l.city = c.city;

INSERT INTO employee_dim(id_employee, company, full_name)
SELECT * FROM sa_northwind.sa_employee;

INSERT INTO product_dim(id_product, name, code, category,
    quantity_per_unit, standard_cost, list_price)
SELECT * FROM sa_northwind.sa_product;

INSERT INTO shipper_dim(company, key_local, id_shipper)
VALUES ('Unknown', 1, 0);

INSERT INTO shipper_dim(company, key_local, id_shipper)
SELECT s.company, l.key_local, s.id_shipper
FROM sa_northwind.sa_shipper s, local_dim l
WHERE s.city = l.city AND s.country = l.country;
```

```
INSERT INTO supplier_dim(company, key_local, id_supplier)
SELECT s.company, l.key_local, s.id_supplier
FROM sa_northwind.sa_supplier s, local_dim l
WHERE s.city = l.city AND s.country = l.country;

INSERT INTO purchase_fact(quantity, total_cost, key_supplier,
    key_product, key_time, id_purchase, id_purchase_details)
SELECT p.quantity, p.total_cost, sd.key_supplier, pd.key_product, td.
    key_time, p.id_purchase, p.id_purchase_details
FROM sa_northwind.sa_purchase p, supplier_dim sd, product_dim pd,
    time_dim td
WHERE p.id_supplier = sd.id_supplier AND p.id_product = pd.id_product
    AND p.submitted_date = td.full_date;

INSERT INTO sale_fact(quantity, total_cost, discount, key_product,
    key_time, key_client, key_shipper, key_employee, id_sale,
    id_sale_details)
SELECT s.quantity, s.total_cost, s.discount, pd.key_product, td.
    key_time, cd.key_client, sd.key_shipper, ed.key_employee, s.id_sale
    , s.id_sale_details
FROM sa_northwind.sa_sale s, product_dim pd, time_dim td, client_dim cd
    , shipper_dim sd, employee_dim ed
WHERE s.id_product = pd.id_product AND s.order_date = td.full_date AND
    s.id_client = cd.id_client AND s.id_shipper = sd.id_shipper AND s.
    id_employee = ed.id_employee;
```

4.3 Mecanismos de atualização do Data Warehouse

Após o povoamento do Data Warehouse, tivemos que decidir como o iríamos atualizar. De entre as várias opções colocadas em cima da mesa aquando da discussão, destacaram-se as seguintes: atualizar o Data Warehouse de cada vez que houvesse uma atualização na base de dados original *Northwind* ou atualizar o Data Warehouse periodicamente através de um script que fará a atualização dos campos.

Sendo a *Northwind* uma empresa de comércio internacional, com grande volume de transações, achamos melhor não sobrecarregar a base de dados com a atualização do Data Warehouse, pelo que optámos por seguir a segunda abordagem. Isto permite-nos agendar as atualizações e corrê-las, por exemplo, durante a noite, quando a empresa não tiver tanto movimento e, assim, não se corre o risco de afetar o desempenho dos seus negócios.

Assim sendo, cada vez que se corre a atualização, temos que, primeiro, limpar a *Staging Area* e só de seguida proceder ao processo de ETL, passando os dados da *Northwind* para a *Staging Area* e de seguida passá-los para o *Data Warehouse*.

Neste processo surgiram algumas alterações que tivemos que executar no modelo do Data Warehouse: como não tínhamos adicionado um *UNIQUE INDEX* em

nenhuma coluna, as operações de verificar se deveria inserir ou atualizar, modificando o campo *last_updated* apenas quando há, de facto, uma atualização, todo o processo estava a ser bastante dificultado. Assim, adicionamos um novo campo às tabelas *sa_purchase* e *purchase_dim* (*id_purchase_details*, que nos indica o ID dos detalhes de cada compra) e às tabelas *sa_sale* e *sale_dim* (*id_sale_details*, semelhante ao anterior mas para as vendas). Estas novas colunas, juntamente com o ID da compra/venda, foram utilizadas para criar o *UNIQUE INDEX* relativo à respetiva tabela. Em suma, criamos os seguintes índices para cada tabela:

- *local_dim*: (country, city)
- *time_dim*: full_date
- *client_dim*: id_client
- *employee_dim*: id_employee
- *supplier_dim*: id_supplier
- *shipper_dim*: id_shipper
- *product_dim*: id_product
- *purchase_fact*: (id_purchase, id_purchase_details)
- *sale_fact*: (id_sale, id_sale_details)

Com isto, a atualização do Data Warehouse tornou-se bastante simples: para as duas primeiras tabelas (*local_dim* e *time_dim*) basta correr uma *query* *INSERT IGNORE INTO*, para adicionar novas linhas caso não existam com os valores indicados como únicos; para as restantes, basta correr *INSERT INTO (...) SELECT (...) FROM (...) ON DUPLICATE KEY UPDATE (...)*

Com isto em mente, o script por nós desenvolvido para a atualização dos dados é o que segue:

- **update_dw.sql**

```
USE dw_northwind;

INSERT IGNORE INTO local_dim(country , city)
SELECT country , city FROM sa_northwind.sa_supplier
UNION DISTINCT SELECT country , city FROM sa_northwind.sa_client
UNION DISTINCT SELECT country , city FROM sa_northwind.sa_shipper;

INSERT IGNORE INTO time_dim(full_date , day , month , year , week_day)
```

```
SELECT submitted_date , DAY(submitted_date) , MONTH(submitted_date) , YEAR
(submitted_date) , WEEKDAY(submitted_date) FROM sa_northwind.
sa_purchase
UNION DISTINCT SELECT order_date , DAY(order_date) , MONTH(order_date) ,
YEAR(order_date) , WEEKDAY(order_date) FROM sa_northwind.sa_sale;

INSERT INTO client_dim(company, full_name, key_local, id_client)
SELECT c.company, c.full_name, l.key_local, c.id_client
FROM sa_northwind.sa_client c, local_dim l
WHERE l.country = c.country AND l.city = c.city
ON DUPLICATE KEY UPDATE
    company = c.company,
    full_name = c.full_name,
    key_local = l.key_local,
    id_client = c.id_client;

INSERT INTO employee_dim(id_employee, company, full_name)
SELECT * FROM sa_northwind.sa_employee e
ON DUPLICATE KEY UPDATE
    id_employee = e.id_employee,
    company = e.company,
    full_name = e.full_name;

INSERT INTO product_dim(id_product, name, code, category,
quantity_per_unit, standard_cost, list_price)
SELECT * FROM sa_northwind.sa_product p
ON DUPLICATE KEY UPDATE
    id_product = p.id_product,
    name = p.name,
    code = p.code,
    category = p.category,
    quantity_per_unit = p.quantity_per_unit,
    standard_cost = p.standard_cost,
    list_price = p.list_price;

INSERT INTO shipper_dim(company, key_local, id_shipper)
SELECT s.company, l.key_local, s.id_shipper
FROM sa_northwind.sa_shipper s, local_dim l
WHERE l.country = s.country AND l.city = s.city
ON DUPLICATE KEY UPDATE
    company = s.company,
    key_local = l.key_local,
    id_shipper = s.id_shipper;

INSERT INTO supplier_dim(company, key_local, id_supplier)
```

```
SELECT s.company, l.key_local, s.id_supplier
FROM sa_northwind.sa_supplier s, local_dim l
WHERE l.country = s.country AND l.city = s.city
ON DUPLICATE KEY UPDATE
    company = s.company,
    key_local = l.key_local,
    id_supplier = s.id_supplier;

INSERT INTO purchase_fact(quantity, total_cost, key_supplier,
    key_product, key_time, id_purchase, id_purchase_details)
SELECT p.quantity, p.total_cost, sd.key_supplier, pd.key_product, td.
    key_time, p.id_purchase, p.id_purchase_details
FROM sa_northwind.sa_purchase p, supplier_dim sd, product_dim pd,
    time_dim td
WHERE p.id_supplier = sd.id_supplier AND p.id_product = pd.id_product
    AND p.submitted_date = td.full_date
ON DUPLICATE KEY UPDATE
    id_purchase = p.id_purchase,
    id_purchase_details = p.id_purchase_details,
    quantity = p.quantity,
    total_cost = p.total_cost,
    key_supplier = sd.key_supplier,
    key_product = pd.key_product,
    key_time = td.key_time;

INSERT INTO sale_fact(quantity, total_cost, discount, key_product,
    key_time, key_client, key_shipper, key_employee, id_sale,
    id_sale_details)
SELECT s.quantity, s.total_cost, s.discount, pd.key_product, td.
    key_time, cd.key_client, sd.key_shipper, ed.key_employee, s.id_sale
    , s.id_sale_details
FROM sa_northwind.sa_sale s, product_dim pd, time_dim td, client_dim cd
    , shipper_dim sd, employee_dim ed
WHERE s.id_product = pd.id_product AND s.order_date = td.full_date AND
    s.id_client = cd.id_client AND s.id_shipper = sd.id_shipper AND s.
    id_employee = ed.id_employee
ON DUPLICATE KEY UPDATE
    quantity = s.quantity,
    total_cost = s.total_cost,
    discount = s.discount,
    key_product = pd.key_product,
    key_time = td.key_time,
    key_client = cd.key_client,
    key_shipper = sd.key_shipper,
    key_employee = ed.key_employee,
    id_sale = s.id_sale,
    id_sale_details = s.id_sale_details;
```

Por outro lado, como foi já referido, a parte da atualização do *last_updated* ainda não estava resolvida: de cada vez que se fazia uma atualização, este campo mudava. Para tal, removemos as atribuições a este campo nos *updates* e modificamos o script de criação do Data Warehouse, de modo a que a coluna *last_updated* fosse definida da seguinte forma:

```
'last_updated' TIMESTAMP NOT NULL DEFAULT NOW() ON UPDATE NOW()
```

Ou seja, a coluna passou a ser gerada automaticamente: quando é inserida uma linha na tabela, o seu valor será o timestamp do momento e, quando há um *update*, o seu valor muda para o *timestamp* do momento do *update*.

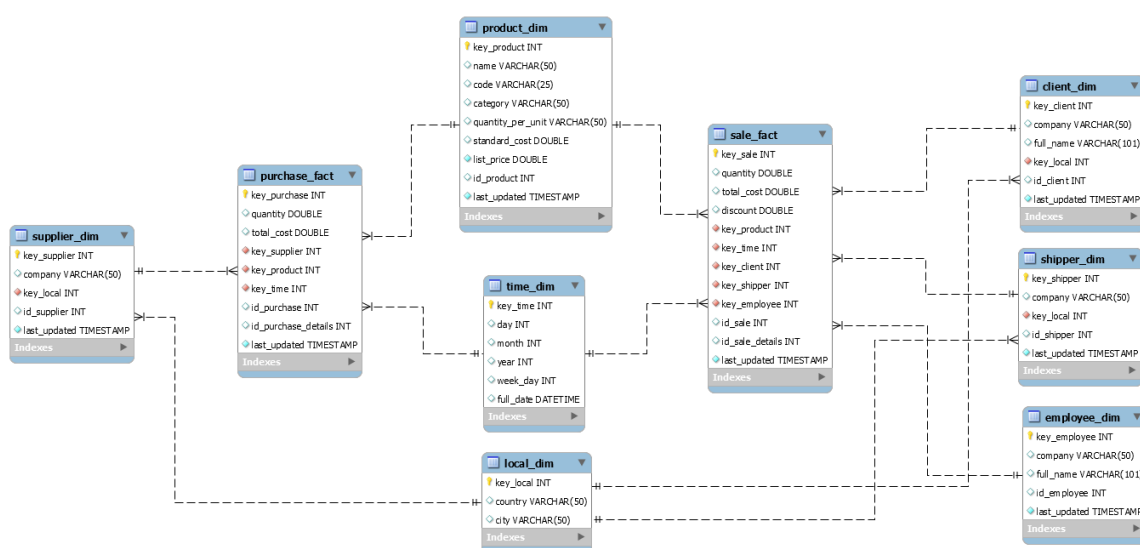


Figura 13: Modelo Lógico do Data Warehouse Atualizado

O diagrama abaixo representa visualmente o processo de ETL, quer na fase de povoamento inicial, quer na fase de atualização, e quais os *scripts* responsáveis por cada tarefa.

POVOAMENTO**ATUALIZAÇÃO**

Figura 14: Diagrama ETL

Finda esta parte, pudemos finalmente passar para a interpretação dos dados do DW e analisar os indicadores de Business Intelligence propostos numa das secções anteriores.

5 Dashboards de Business Intelligence

Ora, partindo dos dados do Data Warehouse, conseguimos desenvolver os seguintes *dashboards* para analisarmos os dados, tendo em atenção os requisitos mencionados na secção 3.1. De forma a facilitar a leitura, dividimos os *dashboards* em partes: uma relativa às compras, outra às vendas, outra comparando as duas anteriores e ainda outra relativa aos funcionários.

5.1 Compras - *purchases*

5.1.1 Quantidade total e quantidade média das compras por fornecedor

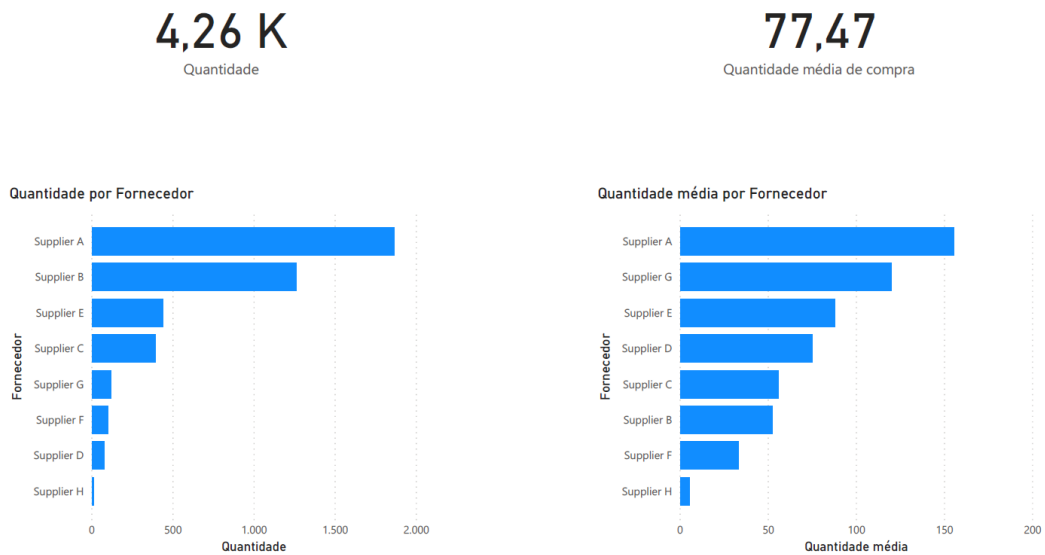


Figura 15: Quantidade total e quantidade média das compras por fornecedor

Observando o gráfico, podemos constatar que a *NorthWind* encomendou mais de quatro mil unidades de produtos, com uma média de 77 unidades por compra. Além disso, é notável a preferência pelos produtos vendidos pelo fornecedor Supplier A, talvez por estes serem também dos mais vendidos. Podemos também ver a insignificância do Supplier H comparativamente aos restantes.

5.1.2 Valor total e valor médio das compras por fornecedor

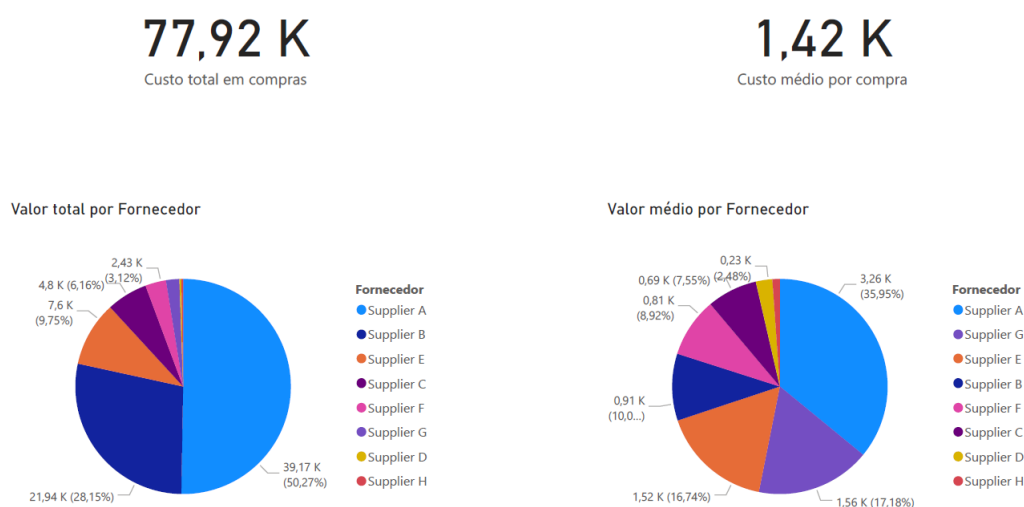


Figura 16: Valor total e valor médio das compras por fornecedor

Com estes gráficos é possível verificarmos, como seria de esperar, que os maiores gastos na compra de produtos são com o fornecedor Supplier A, uma vez que é a este que se fazem o maior número de encomendas. Por outro lado, podemos constatar que se gastou cerca de 78 mil unidades monetárias com as compras, sendo que cada uma tem um custo médio de 1420 unidades monetárias.

5.1.3 Quantidade de cada produto vendido por fornecedor

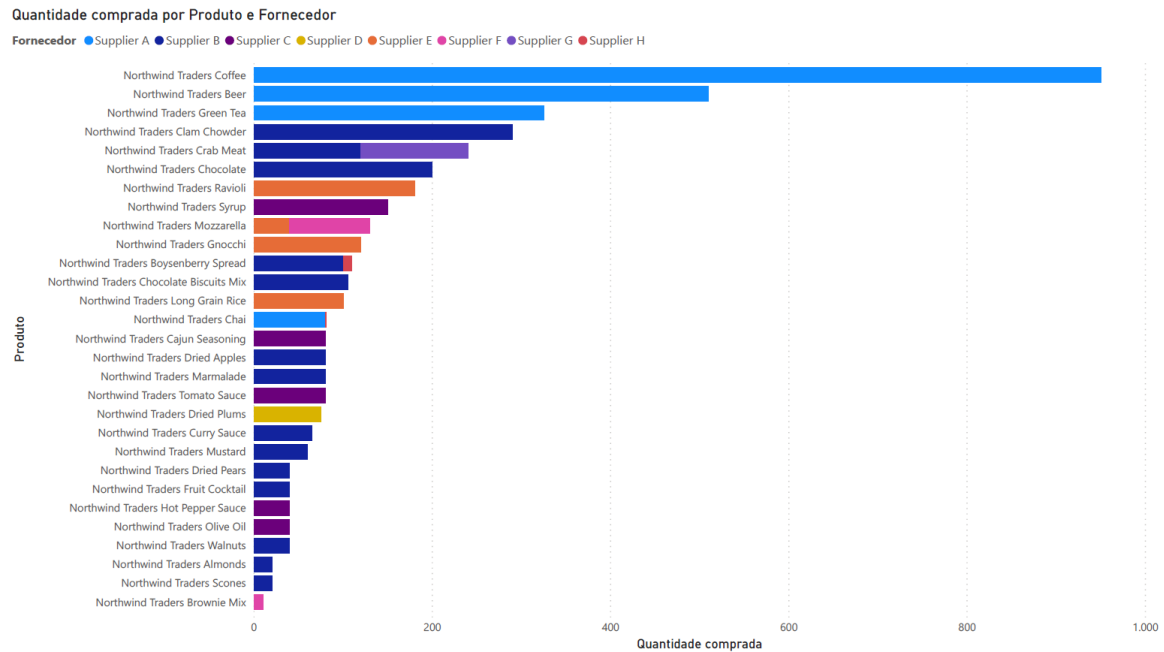


Figura 17: Quantidade de cada produto vendido por fornecedor

Observando este gráfico, conseguimos perceber o porquê da relevância do Supplier A nos gráficos anteriores: esta empresa comercializa em exclusivo com a *Northwind* os três produtos mais encomendados (café, cerveja e chá verde). Por outro lado, são comprados 14 produtos ao Supplier B, apesar de em quantidade muito inferior aos produtos comprados ao Supplier A.

5.1.4 Produtos mais comprados

Quantidade comprada por Produto

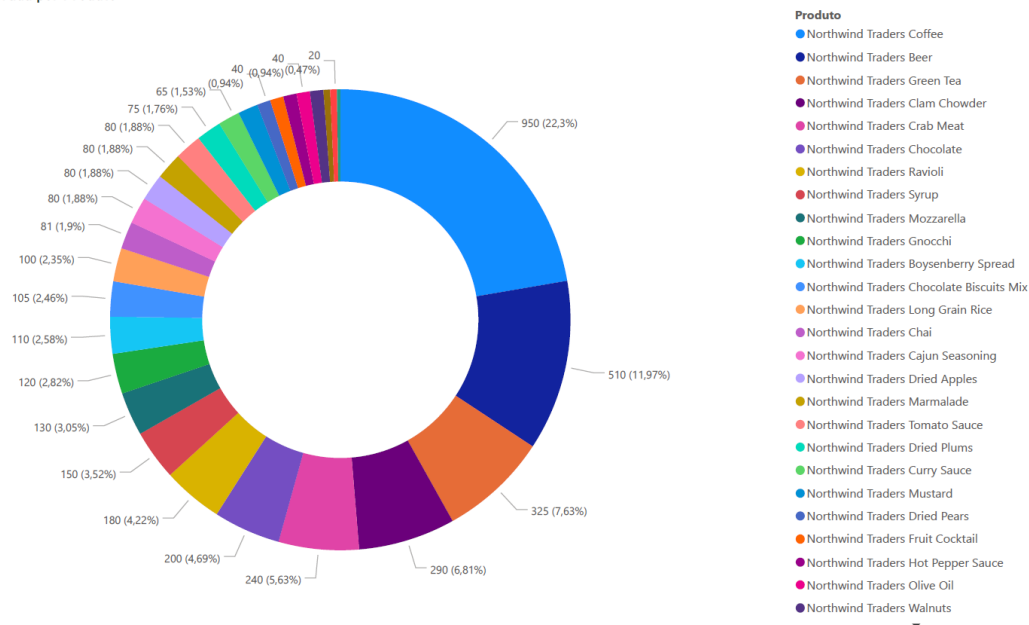


Figura 18: Produtos mais comprados

O que observamos no gráfico anterior, também se consegue perceber neste: o café e a cerveja são os produtos mais comprados aos fornecedores, talvez por serem aqueles que possuem maior procura por parte dos clientes. Só a cerveja e o café representam mais de 30% das compras da *NorthWind*.

5.2 Vendas - sales

5.2.1 Quantidade total e quantidade média das compras por cliente

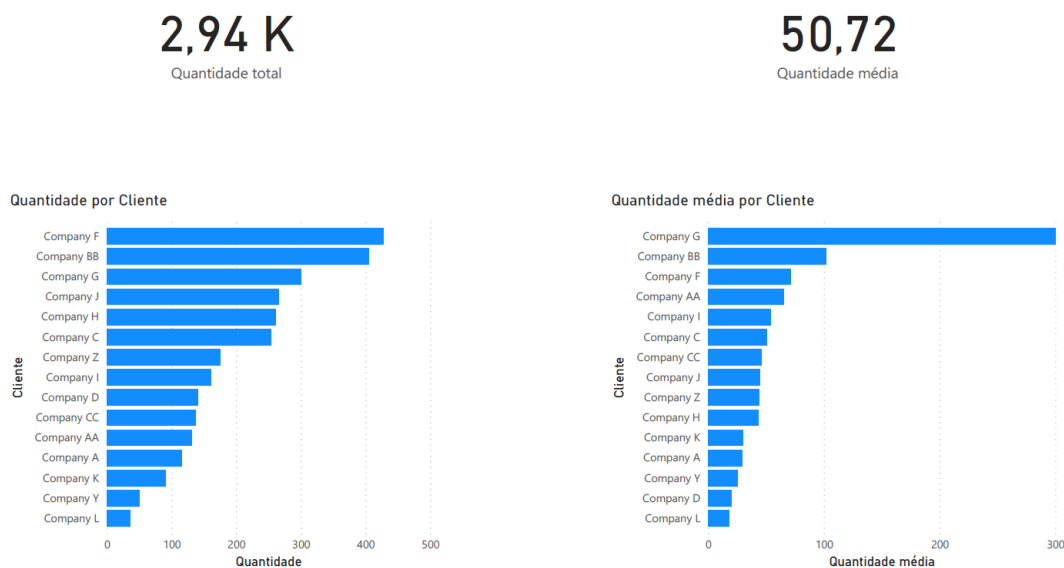


Figura 19: Quantidade total e quantidade média das compras por cliente

Analisando agora as vendas da empresa, é possível verificar que a quantidade dos produtos vendida em cada encomenda é menor àquela comprada (2940 unidades, com 50 unidades em média por encomenda). Além disso, verificamos que os clientes que mais compram são a Company F e a Company BB, por relativa margem comparando à terceira empresa mais relevante (Company G). Porém, esta última é a que apresenta maior média de unidades de produto por encomenda, com cerca de 300, face à segunda mais relevante neste aspeto (Company BB), que apresenta cerca de 100 unidades por encomenda.

5.2.2 Valor total e valor médio das compras por cliente

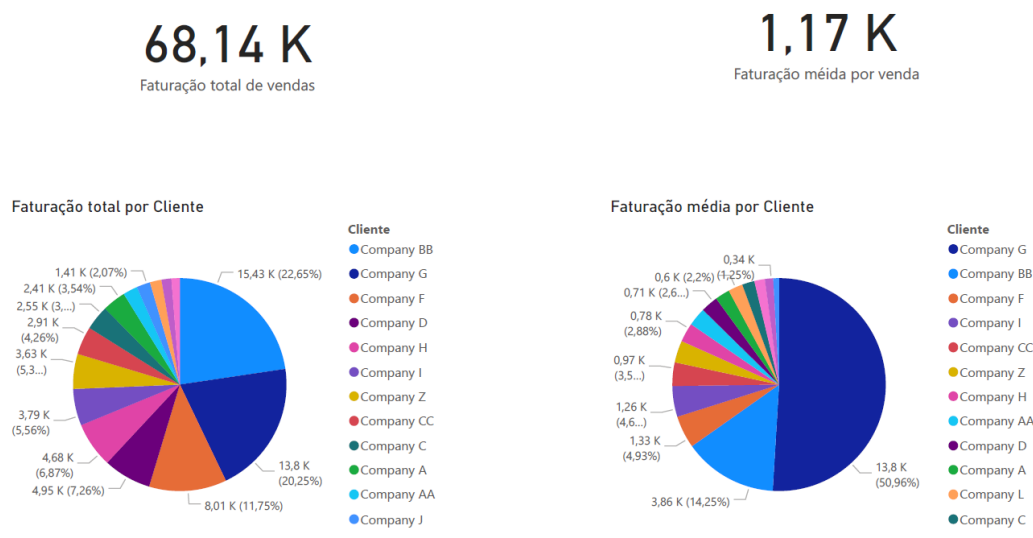


Figura 20: Valor total e valor médio das compras por cliente

Observando desta vez os valores faturados com as vendas, constatamos que o valor total é de 68140 unidades monetárias, quase o mesmo que o dinheiro gasto em compras aos fornecedores. Por outro lado, este dinheiro será certamente reavido, pois a *NorthWind* possui ainda produtos para venda (sendo que não foram todos vendidos). Um facto interessante que podemos tirar deste gráfico é o facto de a empresa que mais quantidade encomendou foi apenas a 3^a que mais gastou, sendo ultrapassada pelas Company BB e Company G. Calculando os gastos médios dos clientes por encomenda, é possível observar que a Company G ultrapassa largamente as restantes empresas, com uma diferença de cerca de dez mil unidades monetárias.

5.2.3 Número total e médio de compras por país e cidade

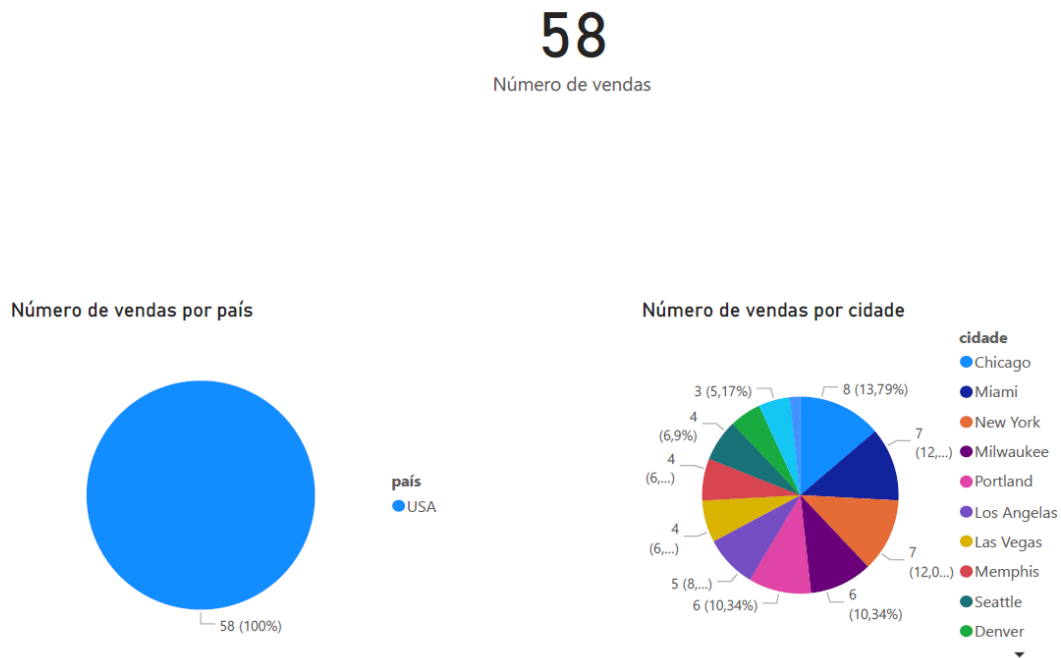


Figura 21: Gráfico - Número total e médio de compras por país e cidade

Analisando agora as vendas da *NorthWind* em termos geográficos, podemos constatar que, no total das 58 vendas, todas foram realizadas nos Estados Unidos da América, em diversas cidades.

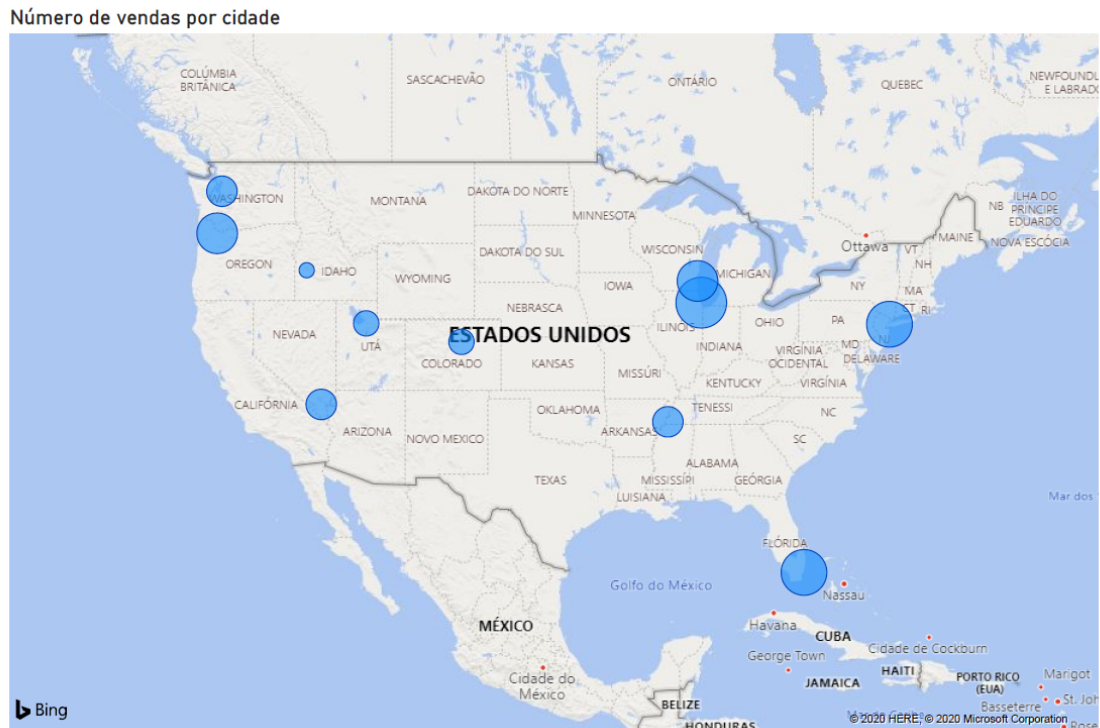


Figura 22: Mapa - Número total e médio de compras por país e cidade

Esta distribuição por cidade dos EUA está graficamente representada no mapa acima, onde é possível observar que a maior frequência de vendas foi feita para Chicago.

5.2.4 Produtos mais vendidos

Quantidade vendida por Produto

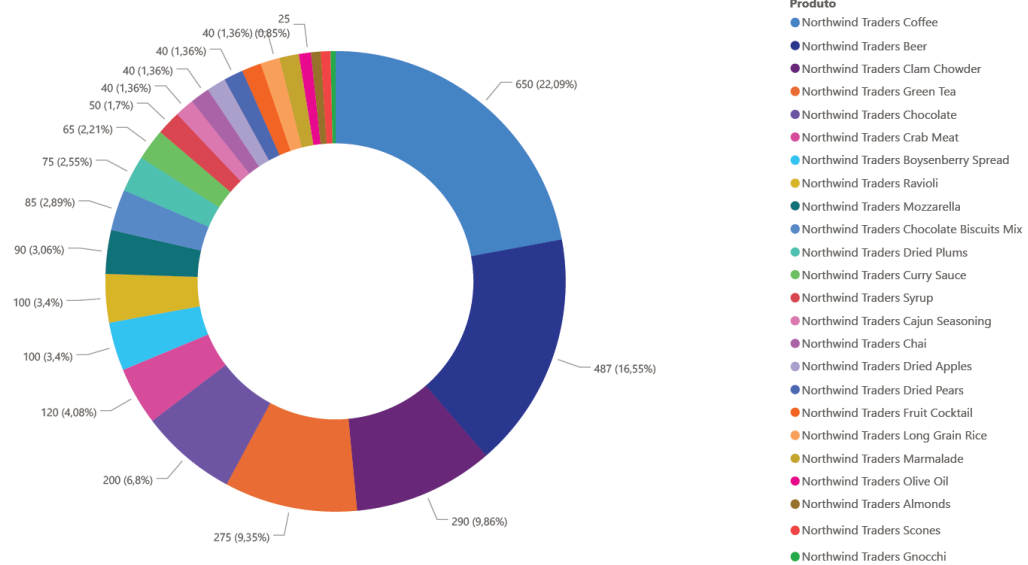


Figura 23: Produtos mais vendidos

Comparando os resultados apresentados neste gráfico com os resultados apresentados na secção 5.1.4, podemos verificar que também o café, a cerveja e o chá verde são dos produtos mais procurados por clientes. Porém, conseguimos perceber que a quantidade de café vendida foi bastante inferior àquela comprada, representando cerca de 2/3 desta última. Além disso, a quantidade de sopa de amêijoia (*Clam Chowder*) vendida foi igual à comprada, ou seja, esgotou o *stock*, pelo que seria interessante para a *NorthWind* talvez comprar mais para satisfazer a procura. Além disso, talvez pudesse ser também benéfica a compra de mais cerveja, uma vez que se venderam quase todas as unidades.

5.3 Comparação entre Compras e Vendas

5.3.1 Quantidade e valor de compras e vendas por mês

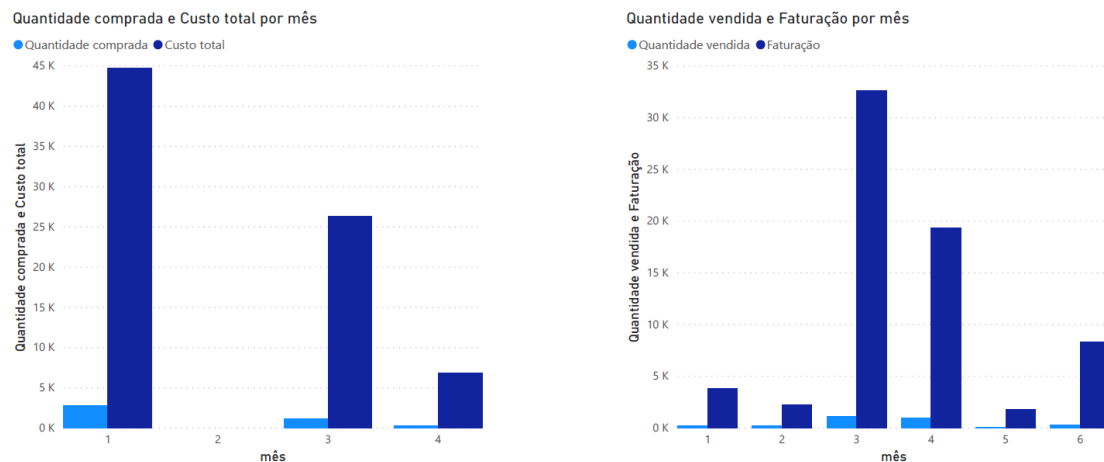


Figura 24: Quantidade e valor de compras e vendas por mês

Comparando os valores das compras e vendas por mês, podemos verificar que grande parte das compras foi efetuada no mês de janeiro, talvez para assegurar produtos disponíveis para os restantes meses. Como é possível verificar, houve um grande pico de vendas no mês de março, pelo que neste mês houve também um número grande de quantidade de produtos comprada, mas ainda assim quase inferior à quantidade do mês de janeiro. Em abril, a quantidade de produtos vendida foi também relevante.

5.3.2 Quantidade e valor de compras e vendas por categoria de produto

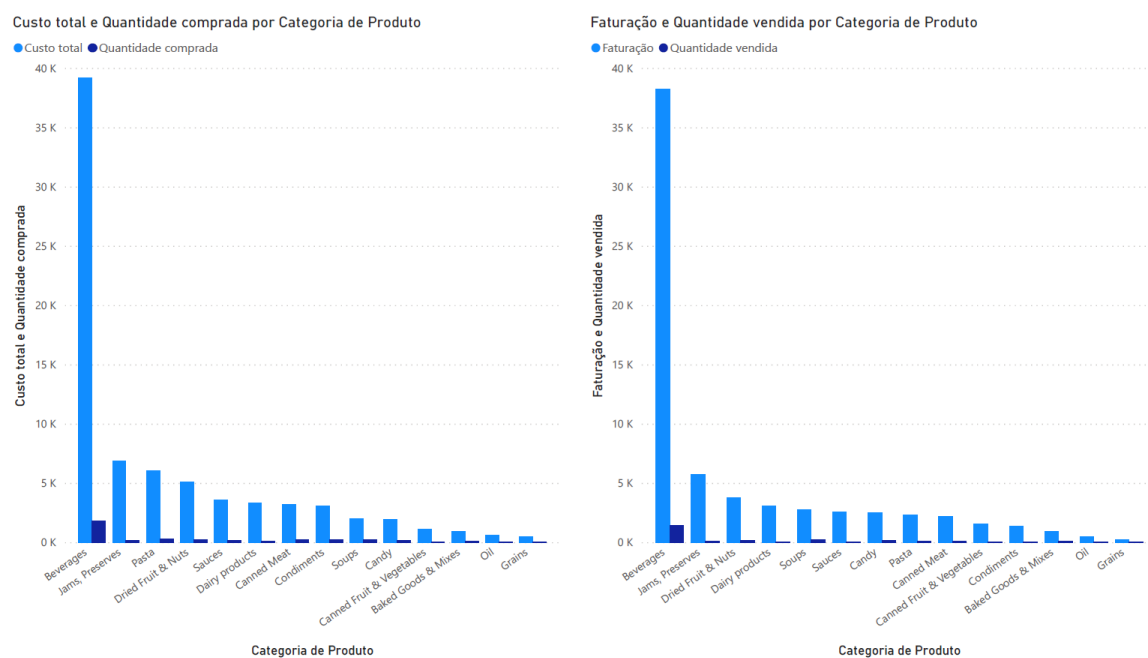


Figura 25: Quantidade e valor de compras e vendas por categoria de produto

Como o gráfico acima ilustra, e algo que já tínhamos verificado anteriormente, a diferença nas quantidades compradas e vendidas de bebidas para as restantes categorias de produtos é abismal. Desta categoria principal fazem parte o café, a cerveja e o chá verde, que fazem parte dos principais produtos comercializados. A diferença é de tal ordem que ronda as trinta mil unidades compradas e trinta mil unidades vendidas.

5.3.3 Quantidade e valor de compras e vendas por mês e categoria de produto

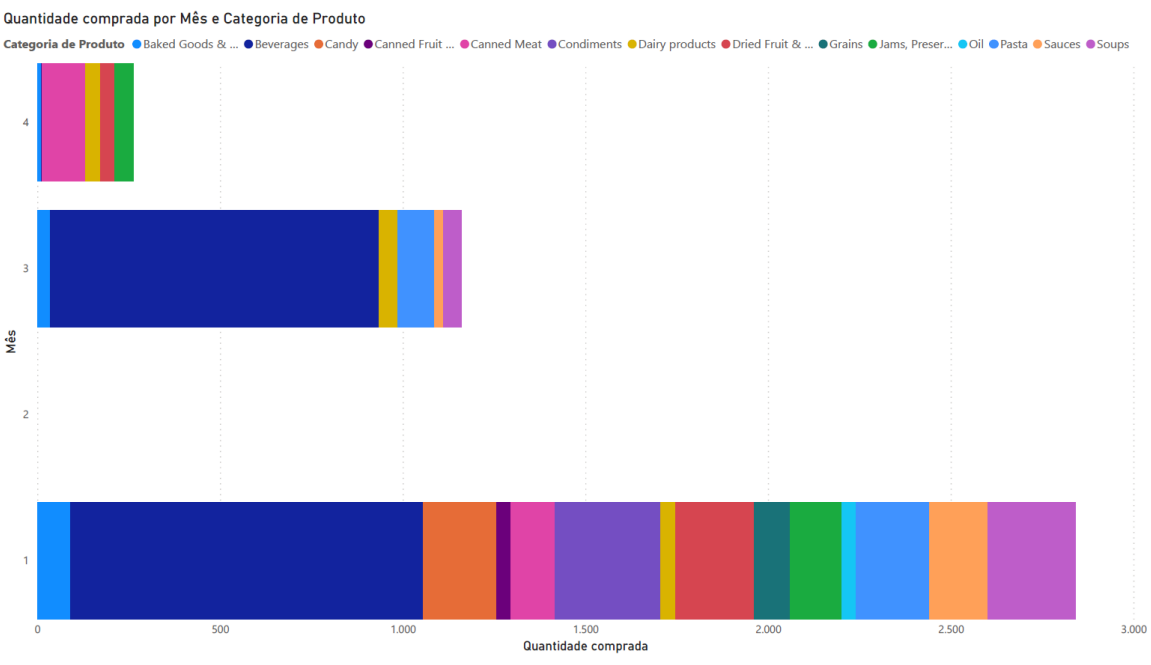


Figura 26: Quantidade e valor de compras por mês e categoria de produto

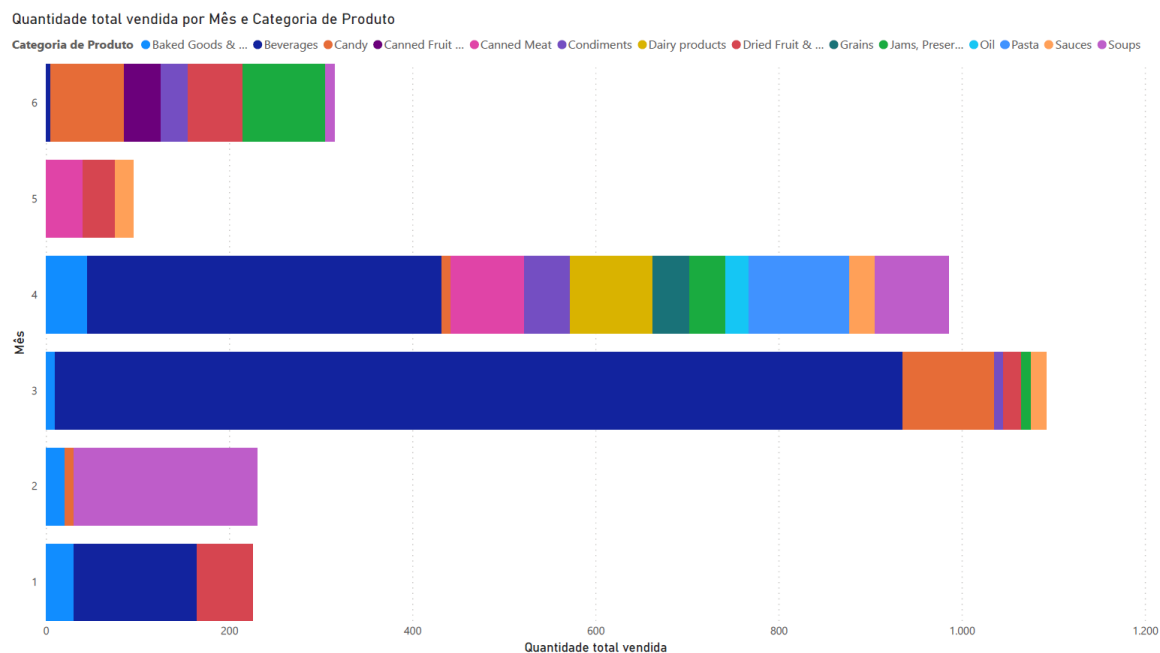


Figura 27: Quantidade e valor de vendas por mês e categoria de produto

Comparando os dois gráficos acima representados, conseguimos perceber que o pico de vendas no mês de março se deveu a um aumento exponencial no consumo de bebidas, o que levou a uma nova encomenda deste produto, no mesmo mês, aos fornecedores por parte da *NorthWind*. O consumo de bebidas permaneceu considerável no mês de abril, apesar de ser cerca de metade do de março. Porém, a empresa conseguiu assegurar a existência de stock. Analisando apenas estes gráficos não se consegue perceber o porquê deste aumento significativo de consumo de bebidas neste período, mas talvez se deva à celebração da Páscoa e das possíveis férias associadas, ou talvez para os estabelecimentos restabelecerem os stocks de bebidas após o Carnaval.

5.3.4 Número de compras e vendas por dia da semana

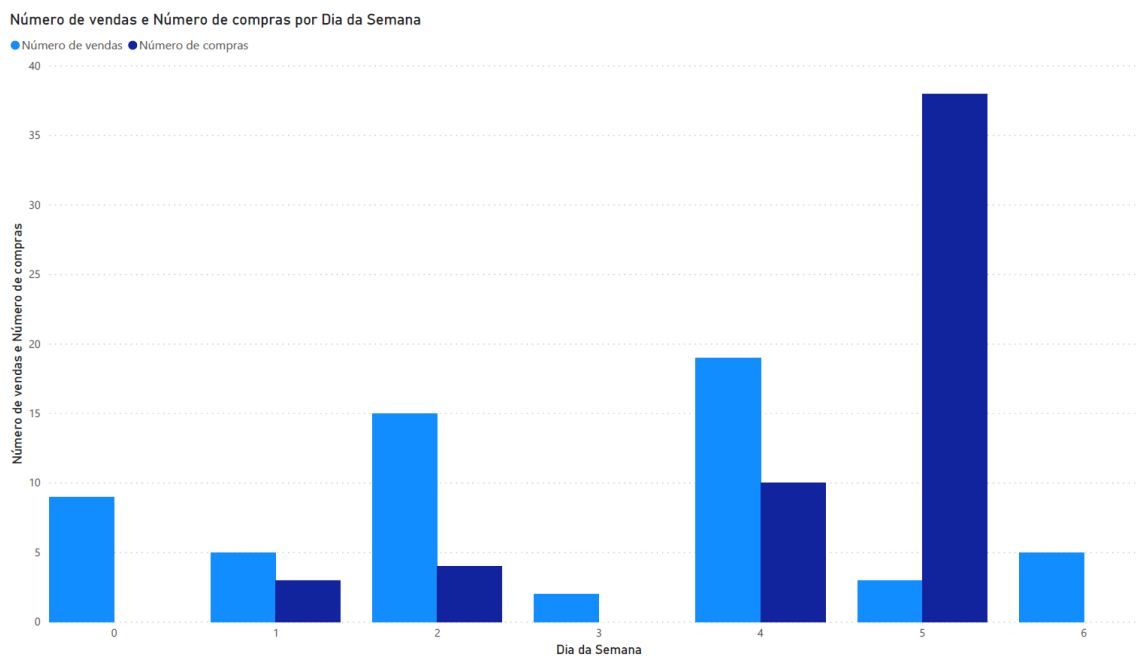


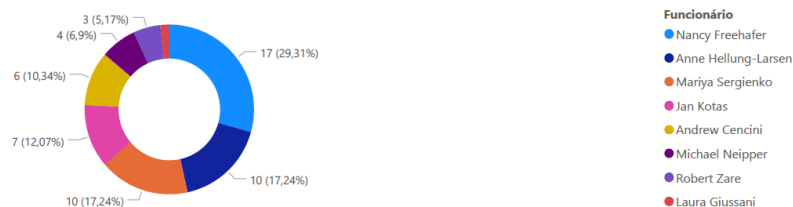
Figura 28: Número de compras e vendas por dia da semana

Com este gráfico, conseguimos verificar que houve vendas em todos os dias da semana. Pelo outro lado, é notável o número de compras no sábado (dia 5) comparando com os restantes dias em que as houve (terça, quarta e sexta). Esta diferença é significativa, sendo de mais de 30 compras entre sábado e os restantes dias referidos.

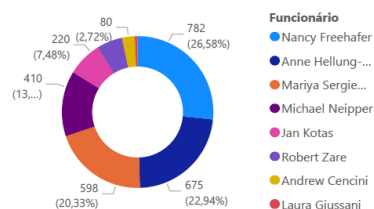
5.4 Funcionários

5.4.1 Funcionários que mais venderam

Número de vendas por Funcionário



Quantidade vendida por Funcionário



Total faturado por Funcionário

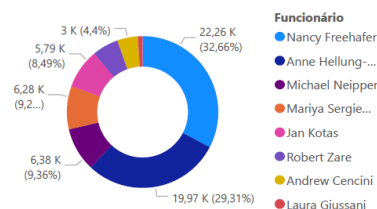


Figura 29: Funcionários que mais venderam

Observando o gráfico relativo aos funcionários e respectivas vendas, podemos verificar que a mais bem sucedida é a Nancy Freehafer. Porém, a Anne Hellung-Larsen e a Mariya Sergie não lhe ficam muito atrás. Apesar da Nancy ter vendido maior quantidade de produtos que a Anne, com mais 7 vendas que esta, a Anne conseguiu vendas com o valor total de cerca de 20 mil unidades monetárias, muito perto do valor conseguido pela Nancy (cerca de 22 mil).

6 Conclusão

Olhando em retrospectiva para todo o trabalho realizado, podemos afirmar que este projeto nos permitiu pôr em prática todos os conteúdos lecionados na unidade curricular Análise de Dados. As várias fases que compuseram o trabalho prático passaram por analisar e compreender uma Base de Dados desconhecida, com relativa complexidade, modelar um Data Warehouse e proceder à sua concretização e atualização, com recurso a uma *Staging Area*, e por fim à análise dos dados presentes no Data Warehouse, criando indicadores de Business Intelligence de modo a permitir a leitura e facilidade na compreensão dos valores presentes no DW, de modo a que os decisores da empresa em questão *NorthWind* possam agir de acordo com os melhores interesses da empresa e analisando os dados passados.

Achamos que fizemos um bom trabalho e que conseguimos cumprir todos os requisitos, apresentando no final um Data Warehouse funcional e com as informações que consideramos mais relevantes. Apresentamos também um conjunto variado de gráficos e *dashboards*, gerados com a ferramenta PowerBI, que nos permitiram fazer uma análise dos dados presentes na base de dados da *NorthWind* e até fazer alguns raciocínios sobre os resultados obtidos e ponderar algumas decisões da empresa.

Em suma, achamos que o trabalho foi bastante enriquecedor e que nos permitiu consolidar a matéria teórica lecionada na UC em que se insere este projeto: Análise de Dados.