



Universidade do Minho
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Lectivo de 2022/2023

Serviço Logístico de uma Empresa de Entrega

Hugo Costeira(a87976), Tiago Rodrigues(a87952)

Janeiro, 2023

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

LCCEntrega

Hugo Costeira(a87976), Tiago Rodrigues(a87952)

Janeiro,2023

Resumo

Este trabalho é realizado no âmbito de desenvolver um sistema de gestão de bases de dados relacional (SGBDR) destinado a permitir a gestão adequada de dados relacionados a encomendas realizadas numa loja/estabelecimento. Desde muito cedo que tentamos definir bem os métodos que iríamos adotar para resolver cada parte do projeto, de modo a desenvolver o mesmo de forma organizada e progressiva. Para que tal fosse possível, a escolha do tema foi demorada no intuito que o tema fosse do agrado de todos para que assim durante a realização do projeto nos sentíssemos igualmente confortáveis e motivados para a resolução do mesmo. Consequentemente, durante esta fase realizamos as etapas de contextualização e viabilidade do projeto, ponderando se cada tema em questão seria justificável neste sentido. De seguida, avançamos para o levantamento de requisitos onde o grande desafio foi saber o que seria informação importante e qual seria a informação que poderíamos descartar, tendo sido os requisitos alterados, revistos e pensados várias vezes. A fase seguinte foi o desenho do modelo conceptual o qual fizemos sem grandes dificuldades visto que tínhamos bem definido o que queríamos através dos requisitos, seguindo-se o desenho do modelo lógico e por fim a realização do modelo físico. Entre estas etapas realizamos várias etapas secundárias, como verificar se o modelo produzido era o correto e assim proceder à sua validação ou estimar o espaço gasto em disco pela base de dados produzida e a taxa de crescimento anual, por exemplo.

Índice

1. Introdução	1
1.1. Contextualização de aplicação do Sistema	1
1.2. Motivação e objetivos do trabalho	1
1.3. Análise da viabilidade do processo	2
1.4. Recursos e Equipa de Trabalho	2
1.5. Plano de execução do Projeto	2
2. Levantamento e Análise de Requisitos	3
2.1. Método de levantamento e de análise de requisitos adotado	3
2.2. Organização dos requisitos levantados	3
2.2.1 Requisitos de descrição	3
2.2.2 Requisitos de exploração	4
2.2.3 Requisitos de controlo	4
2.3. Análise e validação geral dos requisitos	4
3. Modelação Conceptual	5
3.1 Apresentação da abordagem de modelo realizada	5
3.2 Identificação e caracterização das entidades	5
3.3 Identificação e caracterização dos relacionamentos	5
3.4 Identificação da associação dos atributos com as entidades e relacionamentos	6
3.5 Apresentação e explicação do diagrama ER	6
4. Modelação Lógica	7
4.1 Construção e validação do modelo lógico	7
4.2 Desenho do modelo com interrogações do utilizador	11
4.3 Validação do modelo com interrogações do utilizador	11
5. Implementação Física	16
5.1 Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL(alguns exemplos)	16
5.2 Tradução das interrogações do utilizador para SQL (alguns exemplos)	19
5.3 Cálculo do espaço da bases de dados (inicial e taxa de crescimento anual)	21
5.4 Plano de segurança e recuperação de dados	23
6. Conclusões e Trabalho Futuro	23
Referências	24

Índice de Figuras

Figura 1 -Representação do modelo conceptual.	6
Figura 2 -Conversão de entidade.	7
Figura 3 -Conversão de entidade.	7
Figura 4 - Conversão de entidade	8
Figura 5 -Conversão de entidade	8
Figura 6 -Conversão de entidade	9
Figura 7 - Conversão de relação	9
Figura 8 -Conversão de relação	9
Figura 9 -Conversão de relação.	10
Figura 10 - Conversão de relação	10
Figura 11 - Modelo Lógico.	11
Figura 12 - Árvore de Álgebra relacional.	12
Figura 13 - Árvore de Álgebra relacional.	12
Figura 14 - Árvore de Álgebra relacional.	13
Figura 15 - Árvore de Álgebra relacional.	13
Figura 16 -Árvore de Álgebra relacional.	14
Figura 17 -Árvore de Álgebra relacional.	14
Figura 18 - Árvore de Álgebra relacional.	15
Figura 19 - Árvore de Álgebra relacional.	15
Figura 20 - Árvore de Álgebra relacional.	16
Figura 21 - Conversão da Tabela	16
Figura 22 -Conversão da Tabela	17
Figura 23 -Conversão da Tabela	17
Figura 24 -Conversão da Tabela	17.
Figura 25 -Conversão da Tabela	18
Figura 26 - Conversão da Tabela	18
Figura 27 - Interrogação SQL	19
Figura 28 - Interrogação SQL	19
Figura 29 - Interrogação SQL	20
Figura 30 -Interrogação SQL	20

Índice de Tabelas

Tabela 1 -Dicionário de dados das entidades.	5
Tabela 2 -Caracterização dos relacionamentos.	5
Tabela 3 - Tabela estimativa de tamanho	22
Tabela 4 - Tabela estimativa de tamanho	22
Tabela 5 -Tabela estimativa de tamanho	22
Tabela 6 -Tabela estimativa de tamanho	22
Tabela 7 -Tabela estimativa de tamanho	22

1. Introdução

1.1. Contextualização de aplicação do Sistema

A loja LCCEntrega, funcional há poucos anos, foi criada por dois amigos que decidiram começar um negócio conjunto. Esta loja tem como objetivo encomendar produtos requisitados por clientes. Assim sendo, cada funcionário da loja contacta um determinado fornecedor para obter os produtos requisitados pelos clientes, isto é, o funcionário que atender o cliente fica responsável por efetuar a encomenda. Este processo baseia-se, resumidamente, num pedido que um cliente efetua na loja a um determinado funcionário que, por sua vez, contactará o fornecedor para tratar da encomenda. O funcionário terá ainda que anotar todas as informações relativas ao pedido do cliente e, para tal, necessitará dos dados do cliente, isto é, o nome do cliente, email, a sua morada (rua, código postal, número da casa/prédio, andar), o seu contacto (telemóvel, e endereço eletrónico). Posteriormente, o cliente efetuará o pagamento e a encomenda será recebida na loja, que, por sua vez, a encaminhará para a morada do cliente.

1.2. Motivação e objetivos do trabalho

Os dois amigos entenderam que o método que tinham atualmente como habitual estaria a dificultar o processo das encomendas, na medida em que ocorriam erros e dúvidas relativamente às informações e dados respetivos a clientes e encomendas escritos manualmente e atrasos em atendimentos/encomendas.

Assim, o grupo tornou definitiva esta decisão e decidiram avançar com a ideia de passarem a utilizar um sistema que facilitasse o serviço da loja. Para tal, começou a desenvolver-se um sistema de gestão de bases de dados relacional que lhes possibilitaria:

- Uma melhor gestão dos dados respetivos a cada encomenda
- Uma redução dos equívocos/imprecisões relativamente aos dados apontados

1.3. Análise da viabilidade do processo

A necessidade de uma base de dados surgiu após a empresa decidir mudar a forma como o processo das encomendas era feito, possibilitando assim o crescimento da empresa. Para além disto, a criação de uma base de dados seria a melhor opção devido ao aumento da informação. Com tanta informação para ser guardada e posteriormente consultada, a criação da base de dados é completamente justificada.

1.4. Recursos e Equipa de Trabalho

Os donos da empresa já tinham inicialmente pensado em criar um site, por isso, os recursos necessários para a criação de uma base de dados já estavam assegurados. Um desses recursos era a compra de servidores para hospedar a base de dados. Para a construção da base de dados foi fornecida uma sala com tudo o necessário, como por exemplo, computadores e rápido acesso a internet. Como todo o trabalho foi feito na sala fornecida permitiu uma melhor comunicação entre nós e os proprietários o que facilitou imenso a construção da base de dados de acordo com os seus requisitos.

1.5. Plano de Execução do Projeto

Para a construção da base de dados foi necessário construir também um plano de execução. O plano consistiu em dividir o projeto em 5 partes: Levantamento e Análise de Requisitos, Modelagem Conceitual, Modelagem Lógica e por fim Implementação Física. Na parte de método de levantamento e de análise de requisitos adotado vamos falar de que forma conseguimos perceber o que a empresa necessitava de acordo com os planos para a empresa. Assim foi possível construir uma base de dados de acordo com os seus requisitos. De seguida, tivemos de organizar os requisitos pedidos pela empresa. Separamos os requisitos em 3 temas: descrição, exploração e por fim controlo. Depois de elaborar cada tema foi feita uma análise para avaliar se estava tudo feito conforme o pedido pela empresa. Após a confirmação que estava tudo em ordem, dedicamo-nos à realização do modelo conceptual da base de dados fazendo uma identificação e caracterização das entidades, caracterização dos relacionamentos e devida associação dos atributos com as entidades e relacionamentos. Por fim, foi feita uma apresentação e explicação do diagrama ER.

2. Levantamento e Análise de Requisitos

2.1. Método de levantamento e de análise de requisitos adotado

A forma usada para o nosso grupo levantar os requisitos que adotamos para a realização deste sistema de base de dados foi comunicar com os donos da empresa posteriormente à nossa leitura do pedido inicial deles.

Este diálogo foi essencial para delimitarmos concretamente o que deveria ou não ser feito. Nós próprios sugerimos as nossas próprias ideias com base nos nossos conhecimentos de sistemas de bases de dados, o que foi do agrado dos donos da empresa..

Depois de estabelecermos e clarificarmos todos os pormenores, chegamos aos requisitos apresentados em seguida.

2.2. Organização dos requisitos levantados

2.2.1 Requisitos de descrição

- Todos os clientes devem ser registados no sistema;
- Os clientes devem ser caracterizados por: nome, telefone, email, morada;
- Todos os funcionários devem ser registados no sistema;
- Os funcionários devem ser caracterizados por: nome ,telemóvel, email;
- O fornecedor deve ser registado no sistema;
- O fornecedor deve ser caracterizado por: nome e morada;
- Todas as encomendas devem ser registadas no sistema;
- Cada encomenda deve ser caracterizada por: data e preço;
- Todos os itens disponíveis para venda devem ser registados no sistema;
- Cada item deve ser caracterizado através de nome e peso;

2.2.2 Requisitos de exploração

- O sistema deverá ser capaz de identificar todos os clientes;
- O sistema deverá ser capaz de identificar todos os itens disponíveis para venda;
- O sistema deverá ser capaz de identificar todos os seus funcionários;
- O sistema deverá ser capaz de identificar todos os seus fornecedores;
- O sistema deverá ser capaz de identificar todos os itens disponíveis de um fornecedor;
- O sistema deverá ser capaz de identificar as encomendas feitas pelos clientes;
- O sistema deverá ser capaz de identificar o funcionário responsável pela encomenda;
- O sistema deverá ser capaz de identificar o funcionário com mais vendas;
- O sistema deverá ser capaz de identificar o item mais comprado;
- O sistema deverá ser capaz de identificar os itens comprados durante um período de tempo.

2.2.3 Requisitos de controlo

- Os donos têm controlo total sobre a base de dados;
- Os funcionários apenas podem consultar a base de dados;

2.3. Análise e validação geral dos requisitos

Após a nossa consideração de todos os tipos de requisitos, chegamos à conclusão que correspondem exatamente ao que nos foi pedido pelos donos da empresa, o que nos deixou mais confortável para prosseguir no nosso projeto.

3. Modelação Conceptual

3.1 Apresentação da abordagem de modelação realizada

Na criação e desenvolvimento do modelo conceptual utilizamos Entidades, que representam objetos independentes e reais, Atributos, que fazem a caracterização das Entidades, e Relacionamentos, que fazem a associação entre Entidades. Este modelo foi desenhado e representado através da ferramenta BrModelo.

3.2 Identificação e caracterização das entidades

Entidade	Descrição	Atributos
Cliente	Entidade que requisita a encomenda de um determinado produto;	Nome,Telemóvel,Email,Morada
Encomenda	Produto requisitado por um cliente	Data, Preço,idCliente,idFuncionario
Funcionário	Entidade que realiza a encomenda	Nome, telemóvel,Email
Fornecedor	Entidade que possui os itens para venda	Nome,morada
Item	Entidade que possui a informação dos itens vendidos	Nome,peso,idFornecedor

Tabela 1-Dicionário de dados das entidades

3.3 Identificação e caracterização dos relacionamentos

- Cliente faz uma encomenda;
- Funcionário é responsável por uma encomenda
- Encomenda possui item
- O item tem um fornecedor

Entidade	Multiplicidade	Relação	Entidade	Multiplicidade
Cliente	(1,1)	Faz	Encomenda	(0,n)
Encomenda	(1,n)	Tem	Funcionário	(1,1)
Encomenda	(n,m)	Tem	Item	(n,m)
Item	(1,n)	Pertence	Fornecedor	(1,n)

Tabela 2 - Caracterização dos relacionamentos

3.4 Identificação e caracterização da associação dos atributos com as entidades e relacionamentos.

À partida foi unanimemente concordado que todas as entidades teriam um ID como chave primária.

Cliente foi a primeira entidade a surgir e desde logo consideramos que informações básicas do cliente deveriam ser contidas em atributos: nome, telemóvel, email e morada.

A entidade Encomenda teria que ter como atributos o preço total da encomenda, a data que foi realizada, o ID do item e o ID do funcionário responsável pela encomenda.

Funcionário, tal como o cliente, teria que ter as informações básicas como atributos tais como: nome, telemóvel e email.

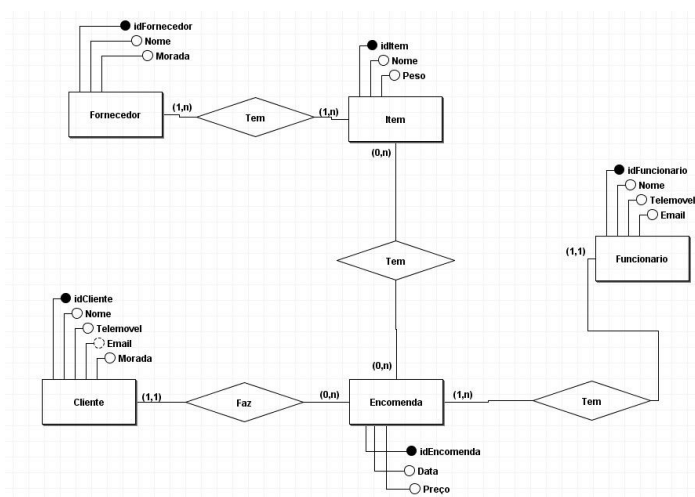
Item contém o seu peso para questões de logística para quando do transporte, o nome e o ID do seu fornecedor.

Por fim, o Fornecedor terá que ter o seu nome e morada.

3.5 Apresentação e explicação do diagrama ER

Atendendo aos requisitos indicados anteriormente, decidimos que deveríamos desenvolver o nosso modelo a partir da entidade Cliente, uma vez que o mesmo terá um pedido(Encomenda) a efetuar. Assim, intuitivamente, a entidade a seguir criada foi a Encomenda. Estas duas entidades teriam que ter uma conexão entre elas no qual a entidade Encomenda teria que conter a informação referente ao cliente, o ID. De seguida foi feita a entidade Funcionário e está, tal como Cliente, também teria uma conexão com Encomenda contendo a informação do funcionário responsável pela mesma. Após estas 3 entidades feitas faltava o conteúdo da Encomenda, ou seja, a entidade Item. Item também teria um relacionamento com Item para poder identificar o conteúdo da Encomenda. Por fim, faltava apenas saber o fornecedor dos itens disponíveis para venda. Para tal, foi feita a entidade Fornecedor com informações básicas sobre o mesmo. Foi feito um relacionamento entre estas duas últimas entidades para poder saber com facilidade o fornecedor de cada item.

Figura 1- Representação do modelo conceptual



4. Modelação Lógica

4.1 Construção e validação do modelo de dados lógico

De forma a construir o modelo lógico, usamos como única base o modelo conceptual apresentado anteriormente. Usamos o MySQL para a sua construção.

Apresentamos em seguida a estratégia usada para para converter cada entidade e relacionamento.

Entidades:

Para criar a tabela Cliente, foram usadas as variáveis da entidade Cliente, adicionando os tipos de dados que achamos mais relevantes. O atributo idCliente foi escolhido como chave primária como previsto no desenvolvimento do modelo conceptual. O único elemento que pode ter valores nulos é o Email do cliente pois o número de telemóvel é obrigatório e é suficiente para contactar o mesmo.

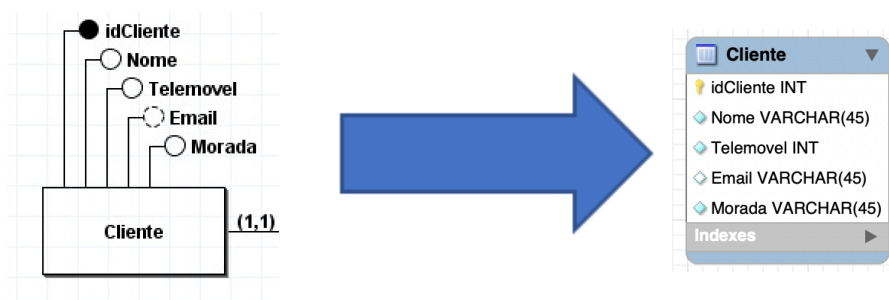


Figura 2- Conversão da entidade

Para criar a tabela Encomenda idEncomenda toma o papel de chave primária. Consideramos que nenhum atributo pode ter valor nulos pois todos os elementos são essenciais para o funcionamento da base de dados.

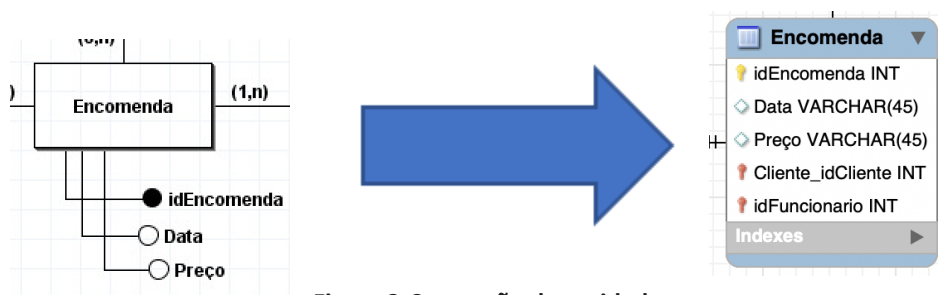


Figura 3- Conversão da entidade

Na tabela Funcionário tem a estrutura idêntica à de Cliente, apenas com menos um atributo, a morada.

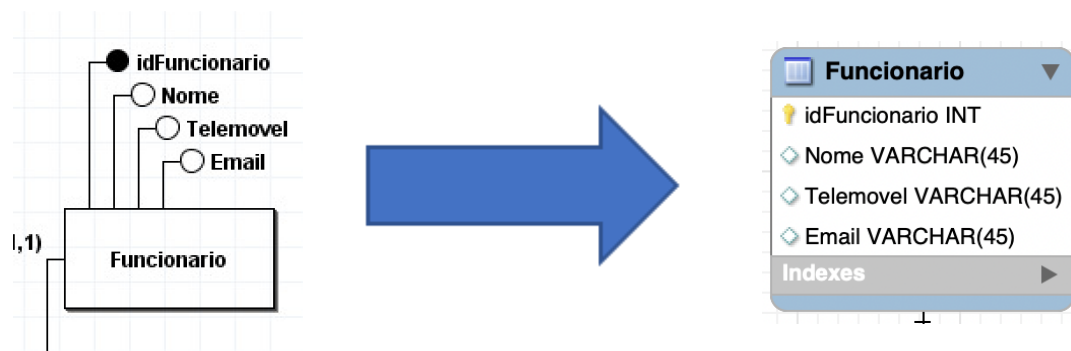


Figura 4-Conversão da entidade

Na tabela fornecedor foi usado a idFornecedor como chave primaria e todos os seus atributos não podem ser nulos.

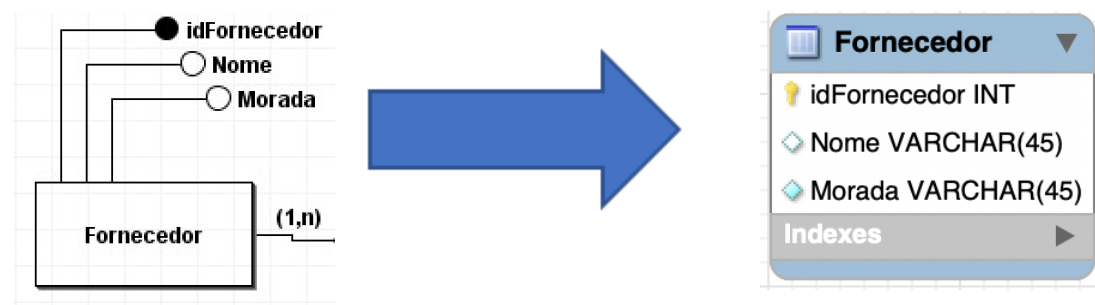


Figura 5-Conversão da entidade

Por fim, na tabela item apenas contém a sua chave primária, id Item, o nome e o seu peso. Para logisticas de entrega, o atributo peso não pode ser nulo.

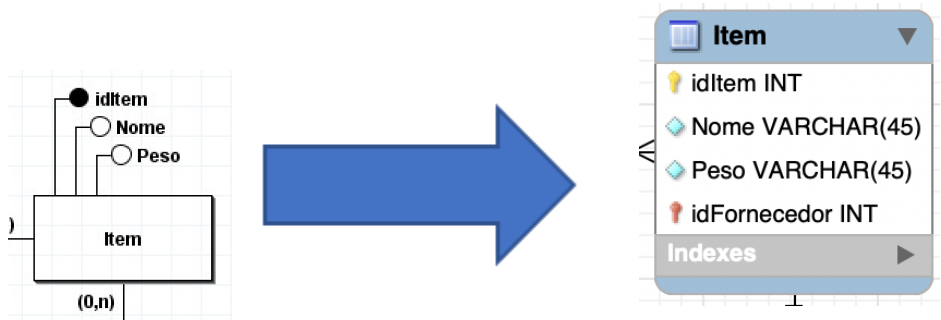


Figura 6- Conversão da entidade

Relacionamentos:

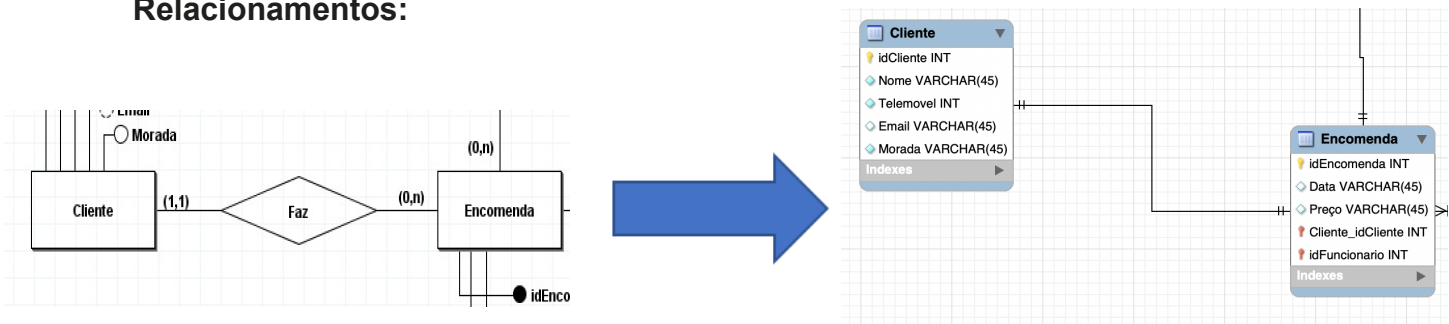


Figura 7- Conversão da relação

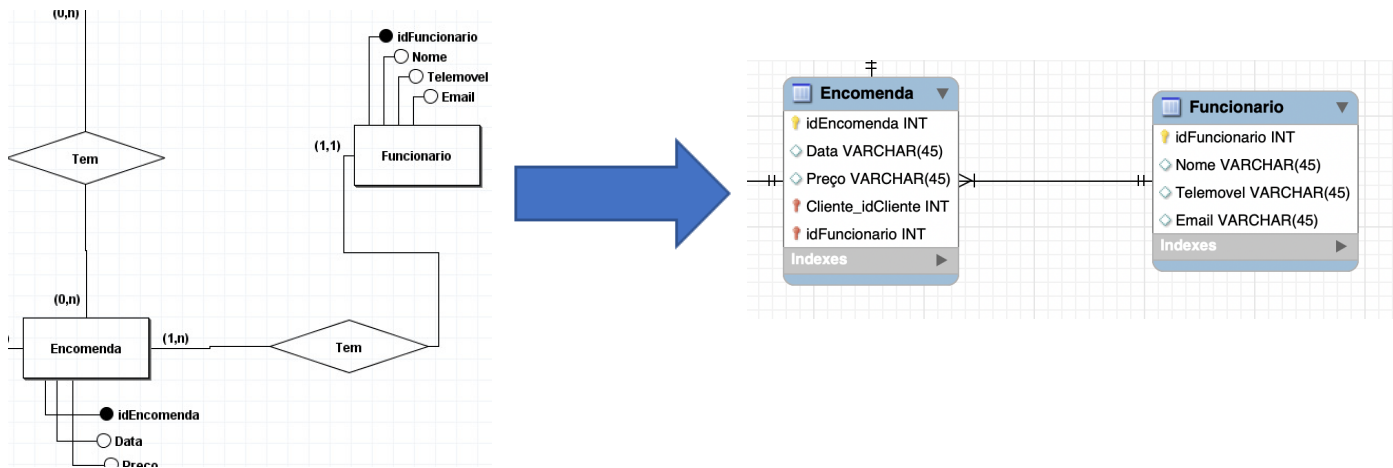


Figura 8- Conversão da relação

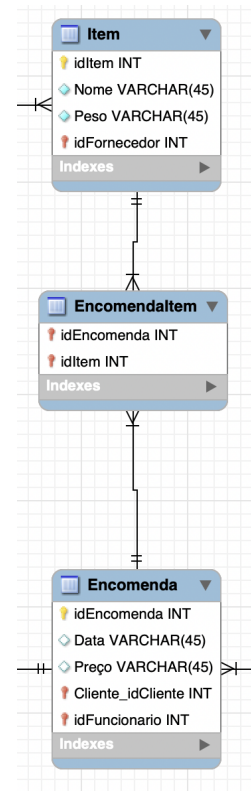
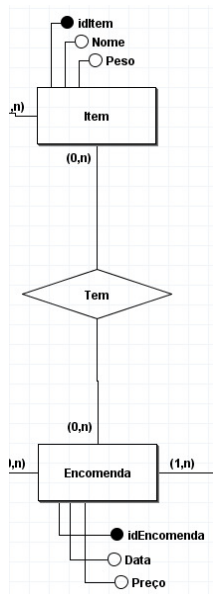


Figura 9- Conversão da relação

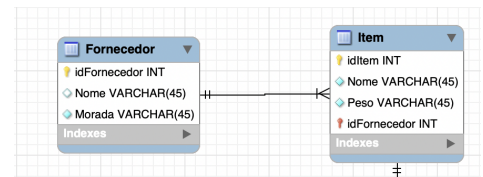
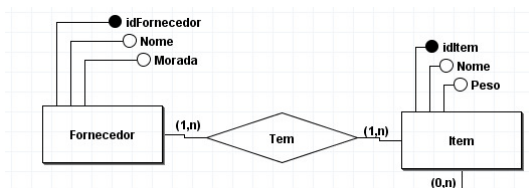


Figura 10- Conversão da relação

4.2 Desenho do modelo lógico

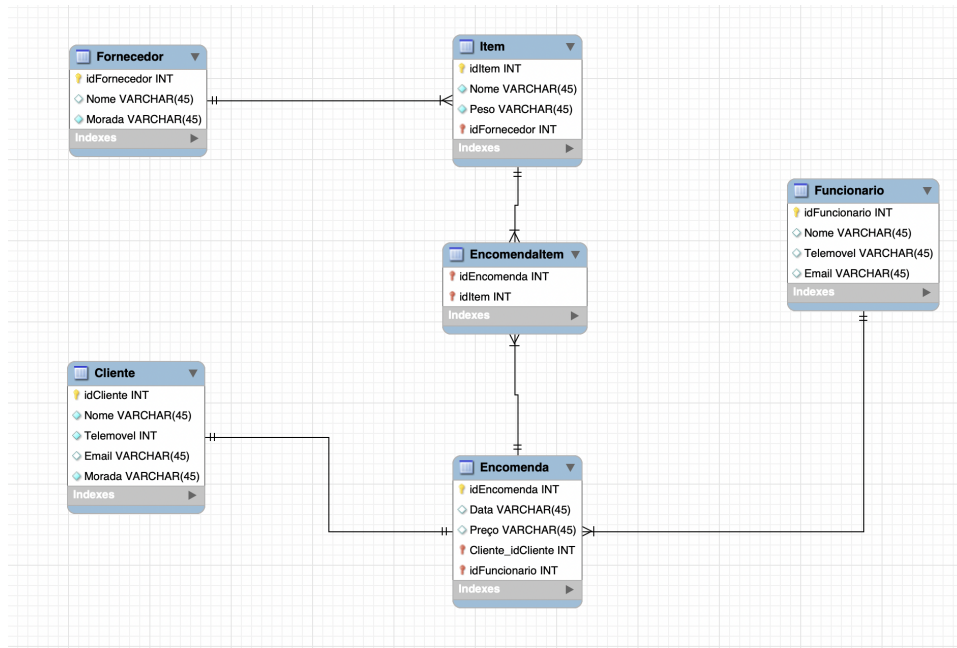


Figura 11-Modelo Lógico

4.3 Validação do modelo com interrogações do utilizador

Interrogação 1:

Mostrar todos os clientes

T_{nome}

$\Pi_{nome \rightarrow nome, telemovel \rightarrow telemovel, email \rightarrow email, morada \rightarrow morada} cliente$

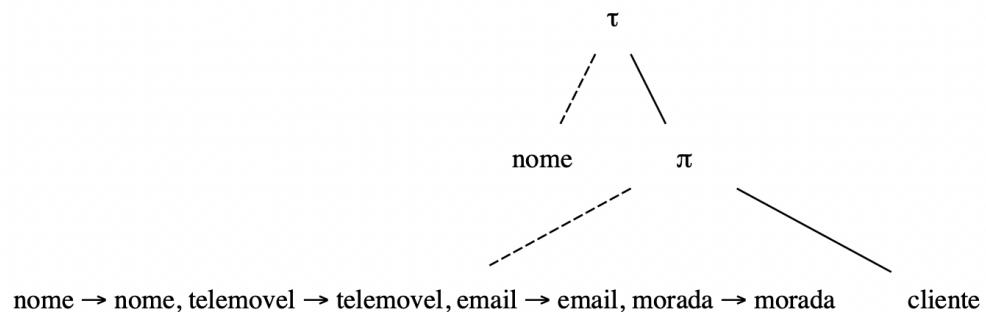


Figura 12- Árvore da Álgebra relacional

Interrogação 2:

Mostrar os itens disponíveis para venda como os seus respetivos fornecedores

$\pi_{\text{item} . \text{nome} \rightarrow \text{nome}, \text{item} . \text{peso} \rightarrow \text{peso}, \text{fornecedor} . \text{nome} \rightarrow \text{fornecedor}} (\text{item} \bowtie_{\text{fornecedor} . \text{idfornecedor} = \text{item} . \text{idfornecedor}} \text{fornecedor})$

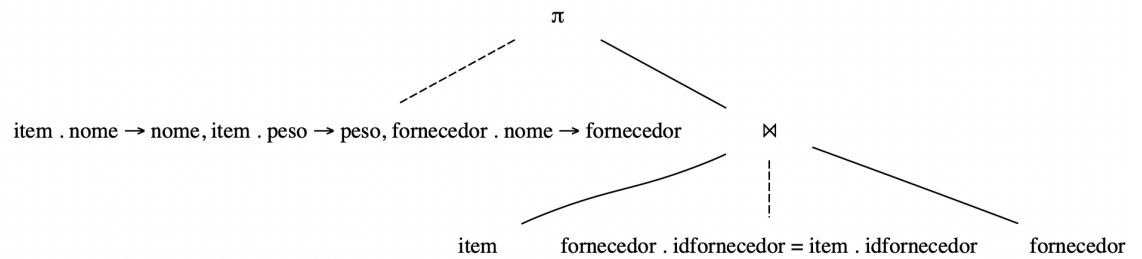


Figura 13- Árvore da Álgebra relacional

Interrogação 3:

Mostrar as informações dos Funcionários

$\pi_{\text{nome} \rightarrow \text{nome}, \text{telemovel} \rightarrow \text{telemovel}, \text{email} \rightarrow \text{email}} \text{funcionario}$

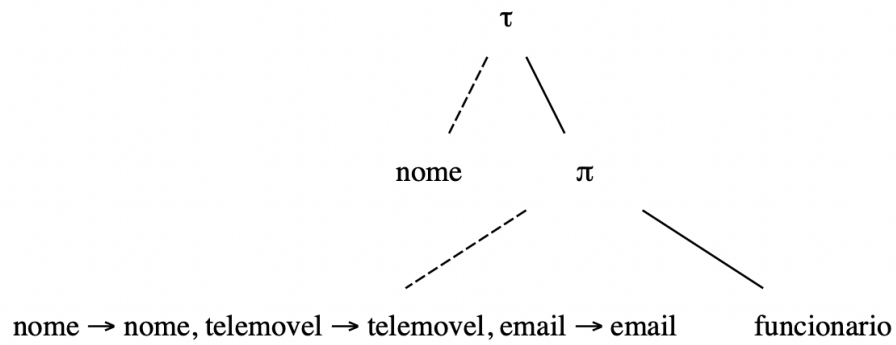


Figura 14- Árvore da Álgebra relacional

Interrogação 4:

Mostrar os Fornecedores disponíveis

$\Pi_{\text{nome} \rightarrow \text{nome}, \text{morada} \rightarrow \text{morada}} \text{fornecedor}$

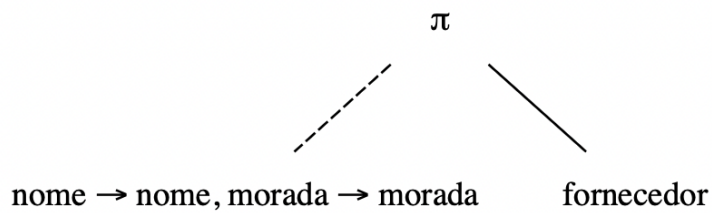


Figura 15- Árvore da Álgebra relacional

Interrogação 5:

Mostrar o que um certo fornecedor tem disponível

$\Pi_{\text{fornecedor} . \text{nome} \rightarrow \text{nome}, \text{item} . \text{nome} \rightarrow \text{item disponível}}$

$\sigma_{\text{fornecedor} . \text{nome} = \text{"Taberna Belga"}} (\text{item} \bowtie_{\text{fornecedor} . \text{idfornecedor} = \text{item} . \text{idfornecedor}} \text{fornecedor})$

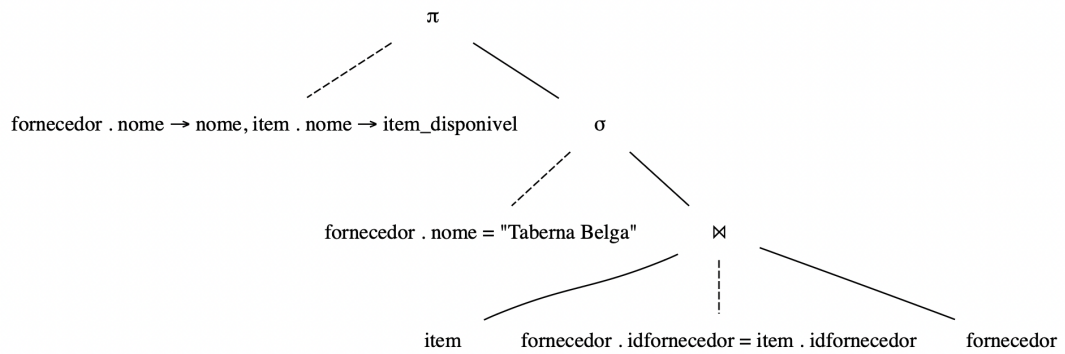


Figura 16- Árvore da Álgebra relacional

Interrogação 6:

Mostrar as encomendas de todos os Clientes

$\Pi_{\text{cliente . nome} \rightarrow \text{nome}, \text{encomenda . data} \rightarrow \text{data}, \text{encomenda . preço} \rightarrow \text{preço}, \text{item . nome} \rightarrow \text{item}} (\text{encomendaitem} \bowtie \text{encomendaitem} .$
 $\text{iditem} = \text{item} . \text{iditem} \text{ item} \bowtie \text{encomendaitem} . \text{idencomenda} = \text{encomenda} . \text{idencomenda} \text{ encomenda} \bowtie \text{cliente} . \text{idcliente} = \text{encomenda} .$
 $\text{cliente_idcliente} \text{ cliente})$

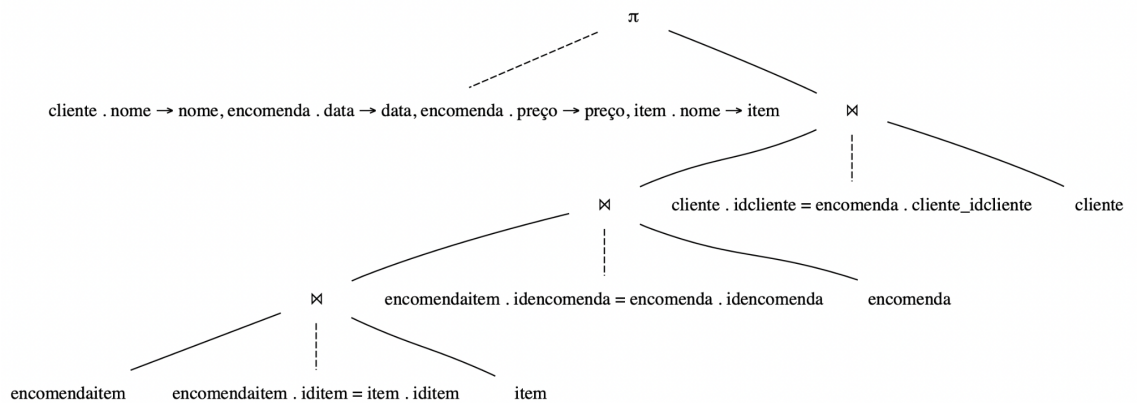


Figura 17- Árvore da Álgebra relacional

Interrogação 7:

Mostrar o funcionário responsável por cada encomenda

$\Pi_{\text{funcionario . nome} \rightarrow \text{funcionario}, \text{cliente . nome} \rightarrow \text{cliente}, \text{item . nome} \rightarrow \text{item}, \text{encomenda . preço} \rightarrow \text{preço}} (\text{encomendaitem} \bowtie$
 $\text{encomendaitem} . \text{iditem} = \text{item} . \text{iditem} \text{ item} \bowtie \text{encomendaitem} . \text{idencomenda} = \text{encomenda} . \text{idencomenda} \text{ encomenda} \bowtie \text{encomenda} .$
 $\text{idfuncionario} = \text{funcionario} . \text{idfuncionario} \text{ funcionario} \bowtie \text{cliente} . \text{idcliente} = \text{encomenda} . \text{cliente_idcliente} \text{ cliente})$

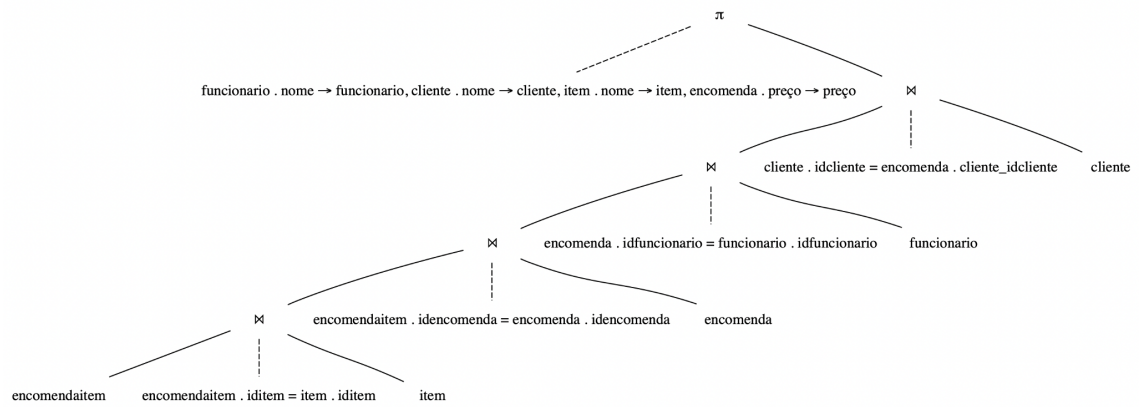


Figura 18- Árvore da Álgebra relacional

Interrogação 8:

Mostrar o funcionário com mais vendas

$\Pi_{funcionario . nome \rightarrow funcionario, COUNT(*)}$

$\gamma_{funcionario, COUNT(*)}(encomenda \bowtie_{encomenda . idfuncionario = funcionario . idfuncionario} funcionario)$

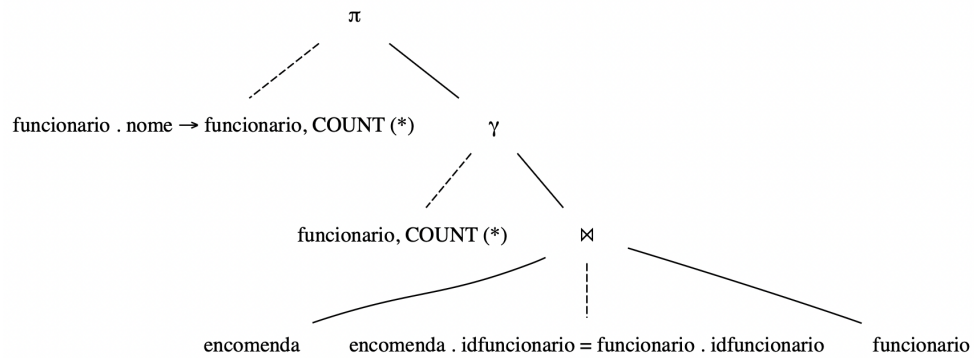


Figura 19- Árvore da Álgebra relacional

Interrogação 9:

Mostrar o item mais comprado

$\Pi_{item . nome \rightarrow item, COUNT(*)}$

$\gamma_{item, COUNT(*)}(encomendaitem \bowtie_{encomendaitem . iditem = item . iditem} item)$

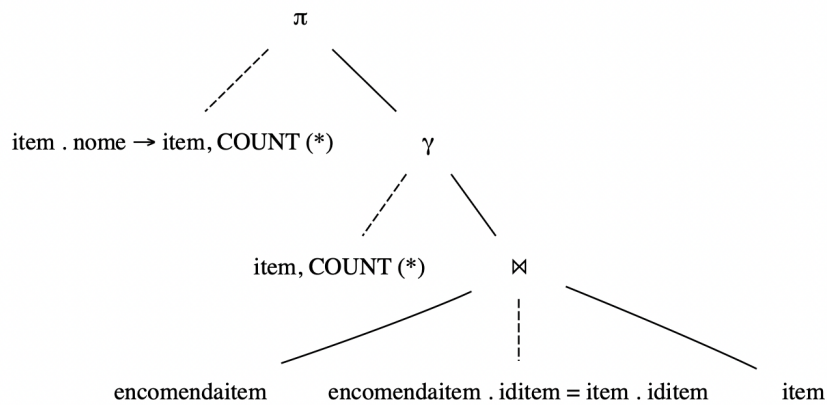


Figura 20- Árvore da Álgebra relacional

5. Implementação Física

Para construir o modelo físico recorreremos à ferramenta disponibilizada pelo MySQL, tal como nas aulas, tendo sido todo o código referente a este modelo gerado automaticamente através do comando Forward Engineer. De seguida iremos mostrar, tal como na secção 4.1, um conjunto de imagens que mostram esta transição de modelos para cada tabela do modelo lógico.

5.1 Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL (alguns exemplos)

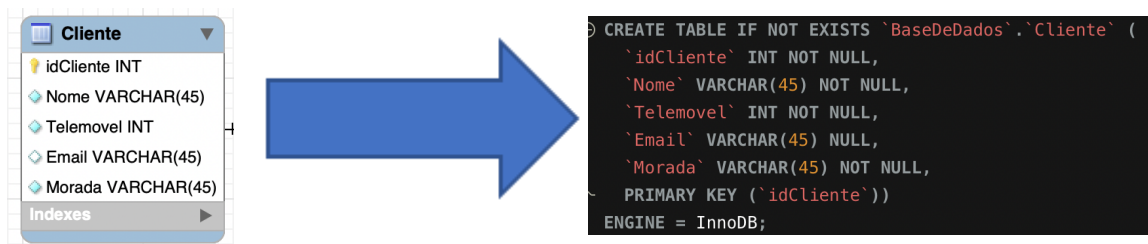
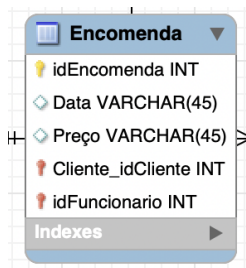
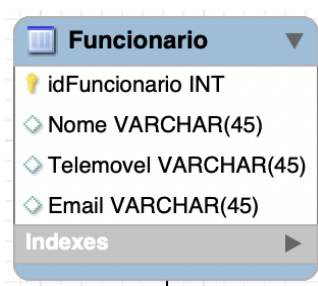


Figura 21- Conversão da Tabela



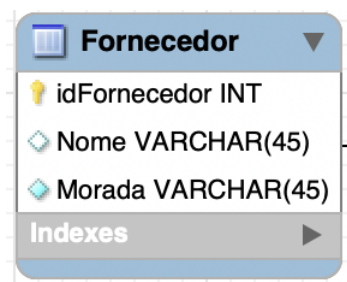
```
CREATE TABLE IF NOT EXISTS `BaseDeDados`.`Encomenda` (
  `idEncomenda` INT NOT NULL,
  `Data` VARCHAR(45) NULL,
  `Preço` VARCHAR(45) NULL,
  `Cliente_idCliente` INT NOT NULL,
  `idFuncionario` INT NOT NULL,
  PRIMARY KEY (`idEncomenda`, `Cliente_idCliente`, `idFuncionario`),
  INDEX `fk_Encomenda_Cliente1_idx` (`Cliente_idCliente` ASC) VISIBLE,
  INDEX `fk_Encomenda_Funcionario1_idx` (`idFuncionario` ASC) VISIBLE,
  CONSTRAINT `fk_Encomenda_Cliente1`
    FOREIGN KEY (`Cliente_idCliente`)
      REFERENCES `BaseDeDados`.`Cliente` (`idCliente`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `fk_Encomenda_Funcionario1`
    FOREIGN KEY (`idFuncionario`)
      REFERENCES `BaseDeDados`.`Funcionario` (`idFuncionario`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

Figura 22- Conversão da Tabela



```
CREATE TABLE IF NOT EXISTS `BaseDeDados`.`Funcionario` (
  `idFuncionario` INT NOT NULL,
  `Nome` VARCHAR(45) NULL,
  `Telemove` VARCHAR(45) NULL,
  `Email` VARCHAR(45) NULL,
  PRIMARY KEY (`idFuncionario`))
ENGINE = InnoDB;
```

Figura 23- Conversão da Tabela



```
CREATE TABLE IF NOT EXISTS `BaseDeDados`.`Fornecedor` (
  `idFornecedor` INT NOT NULL,
  `Nome` VARCHAR(45) NULL,
  `Morada` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idFornecedor`))
ENGINE = InnoDB;
```

Figura 24- Conversão da Tabela

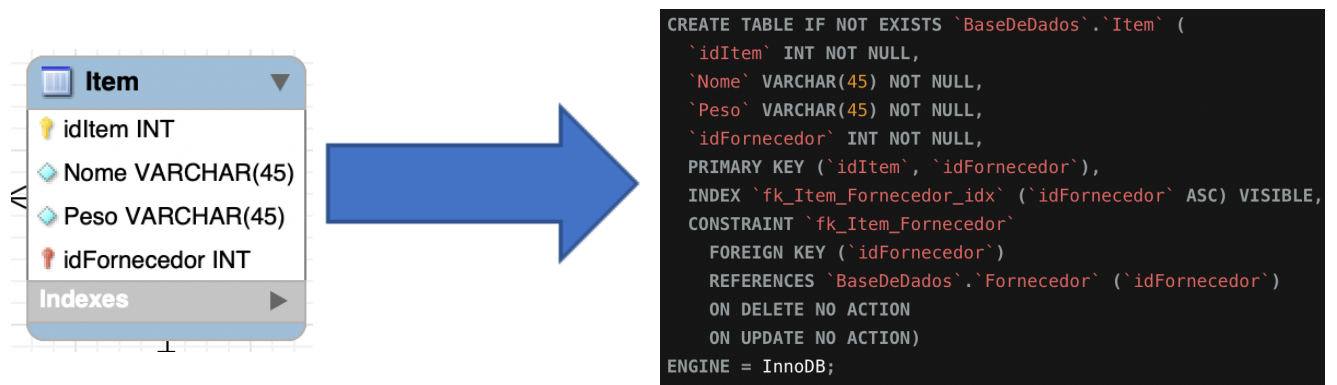


Figura 25- Conversão da Tabela

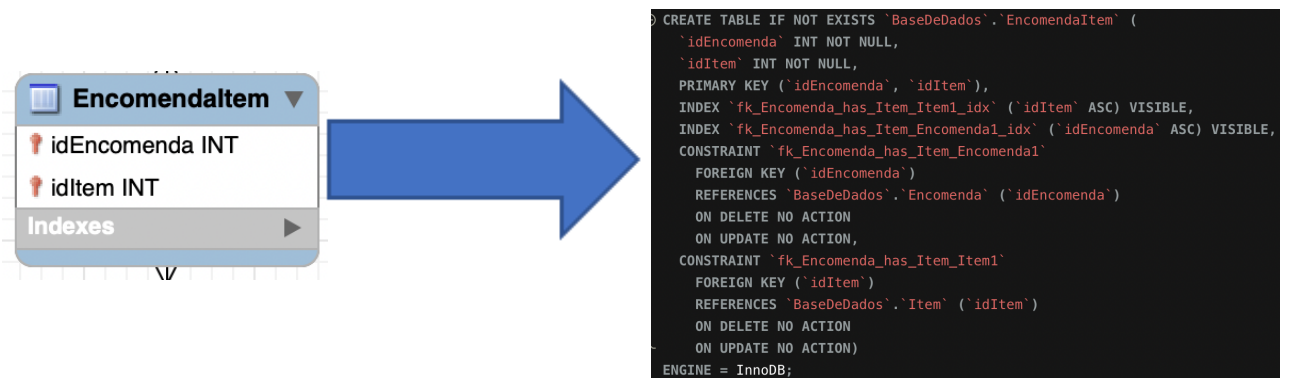


Figura 26- Conversão da Tabela

5.2 Tradução das interrogações do utilizador para SQL (alguns exemplos)

Com o modelo físico implementado e depois de fazermos o povoamento, passamos à fase de implementação das interrogações necessárias em SQL. De seguida, iremos mostrar apenas alguns exemplos de interrogações implementadas.

```
SELECT Item.Nome as Nome, Item.Peso as Peso, Fornecedor.Nome as Fornecedor
FROM Item
INNER JOIN Fornecedor
ON Fornecedor.idFornecedor = Item.idFornecedor;
```

Figura 27- Interrogação SQL

Com esta interrogação pretendemos mostrar todos os itens disponíveis para venda como também os seus respetivos fornecedores.

Deste modo, fizemos a junção utilizando um INNER JOIN de 2 tabela, a de Fornecedor, através do atributo em comum da tabela Item com Fornecedor.

```
SELECT Fornecedor.Nome as Nome, Item.Nome as Item_Disponivel
From Item
INNER JOIN Fornecedor
ON Fornecedor.idFornecedor = Item.idFornecedor
WHERE Fornecedor.Nome = 'Taberna Belga';
```

Figura 28- Interrogação SQL

Com esta interrogação pretendemos mostrar o que um certo fornecedor, neste caso Taberna Belga, tem disponível para venda.

Deste modo, fizemos a junção utilizando um INNER JOIN de 2 tabela, a de Fornecedor, através do atributo em comum da tabela Item com Fornecedor. No final foi feito um filtro para possibilitar ver o fornecedor que o administrador quiser ver através do WHERE comparando o atributo “Nome” da tabela Fornecedor com um nome do fornecedor, neste caso ‘Taberna Belga’.

```
SELECT Funcionario.Nome as Funcionario, Cliente.Nome as Cliente, Item.Nome as Item, Encomenda.Preço as Preço
From EncomendaItem
INNER JOIN Item
ON EncomendaItem.idItem = Item.idItem
INNER JOIN Encomenda
ON EncomendaItem.idEncomenda = Encomenda.idEncomenda
INNER JOIN Funcionario
ON Encomenda.idFuncionario = Funcionario.idFuncionario
INNER JOIN Cliente
ON Cliente.idCliente = Encomenda.Cliente_idCliente;
```

Figura 29- Interrogação SQL

Com esta interrogação pretendemos mostrar o funcionário responsável por cada encomenda realizada assim como o cliente que a realizou.

Começamos por fazer a junção utilizando um INNER JOIN de 5 tabelas. Primeiro, fizemos a junção das tabelas EncomendaItem com Item juntando o que é comum entre elas. De seguida, com o resultado da tabela anterior foi feita ainda a junção de Encomenda, Funcionário e Item onde a junção foi feita em prol do resultado da junção anterior.

```
SELECT Item.Nome as Item, COUNT(*) AS `Numero de vezes vendidos`  
  From EncomendaItem  
    INNER JOIN Encomenda  
      ON EncomendaItem.idEncomenda = Encomenda.idEncomenda  
    INNER JOIN Item  
      ON EncomendaItem.idItem = Item.idItem  
 WHERE DATE(Encomenda.Data) BETWEEN "2022-08-01" AND "2022-12-31"  
 GROUP BY Item  
 ORDER BY `Numero de vezes vendidos` DESC;
```

Figura 30- Interrogação SQL

Com esta interrogação pretendemos mostrar o item mais vendido num certo período de tempo.

Começamos, mais uma vez, por fazer a junção de 3 tabelas. Primeiro, a junção de EncomendaItem com Encomenda usando o que é comum entre as duas tabelas. De seguida, com o resultado anterior, foi feita a última junção com Item.

No final foi feito um filtro para possibilitar ver os itens comprados durante um certo período de tempo. Foi feita ainda uma agrupação no qual está organizado por ordem decrescente .

5.3 Cálculo do espaço da bases de dados (inicial e taxa de crescimento anual)

Tabela	Atributos	Tipo	Tamanho
Cliente	ID Nome Telemóvel Email Morada	INT VARCHAR(45) INT VARCHAR(45) VARCHAR(45)	4 Bytes 45Bytes 4 Bytes 45Bytes 45Bytes

Tabela 3 - Tabela estimativa de tamanho

Uma única tabela Cliente ocupa 143 bytes, logo como temos 8 clientes na base de dados tempos que no total esta tabela ocupa 1144 bytes.

Tabela	Atributos	Tipo	Tamanho
Funcionário	ID Nome Telemóvel Email	INT VARCHAR(45) INT VARCHAR(45)	4 Bytes 45 Bytes 4 Bytes 45 Bytes

Tabela 4 - Tabela estimativa de tamanho

Uma única tabela Funcionário ocupa 98 bytes, logo como temos 4 funcionários na base de dados tempos que no total esta tabela ocupa 392 bytes.

Tabela	Atributos	Tipo	Tamanho
Encomenda	ID Nome Data Preço ID	INT VARCHAR(45) VARCHAR(45) VARCHAR(45) INT	4 Bytes 45 Bytes 45 Bytes 45 Bytes 4 Bytes

	ID	INT	4 Bytes
--	----	-----	---------

Tabela 5 - Tabela estimativa de tamanho

Uma única tabela Funcionário ocupa 147 bytes, logo como temos 14 encomendas na base de dados tempos que no total esta tabela ocupa 2058 bytes.

Tabela	Atributos	Tipo	Tamanho
Item	ID Nome Peso ID	INT VARCHAR(45) INT INT	4 Bytes 45 Bytes 4 Bytes 4 Bytes

Tabela 6 - Tabela estimativa de tamanho

Uma única tabela Funcionário ocupa 57 bytes, logo como temos 16 itens na base de dados tempos que no total esta tabela ocupa 912 bytes.

Tabela	Atributos	Tipo	Tamanho
Fornecedor	ID Nome Morada	INT VARCHAR(45) VARCHAR(45)	4 Bytes 45 Bytes 45 Bytes

Tabela 7 - Tabela estimativa de tamanho

Uma única tabela Funcionário ocupa 94 bytes, logo como temos 5 fornecedores na base de dados temos que no total esta tabela ocupa 470 bytes.

5.4 Plano de segurança e recuperação de dados

A base de dados criada para a empresa também contém um plano de segurança em que é feito um backup de segurança semanalmente, para evitar eventuais “desastres” em que se perca a base de dados por inúmeros motivos. Outra medida de segurança que foi implementada nesta base de dados foi a utilização de um login com nome de utilizador e palavra-passe para os administradores da base de dados, para prevenir o acesso de desconhecidos à base de dados.

6. Conclusões e Trabalho Futuro

O nosso grupo reflete que este projeto foi desafiante, na medida em que nos fez pensar em várias possibilidades para problemas que poderíamos tratar, bem como as várias alterações que efetuamos nas primeiras etapas, tendo que relevar as informações importantes (Entidades e atributos), bem como as relações que teriam entre si e a cardinalidade das relações. Nesta última investimos mais tempo para garantir que implementaríamos as relações de forma correta, e foi a etapa onde recuamos um maior número de vezes por lembrar de algumas hipóteses que invalidariam o que achávamos estar correto. Eventualmente concordamos com todas as informações e construímos o Diagrama ER.

Construído o Diagrama, conseguimos avançar mais rapidamente para os modelos finais, uma vez que já conseguíamos definir o Modelo Conceptual que, por sua vez, nos permitiu conceber o Modelo Lógico. Restando apenas a tradução do esquema lógico para o SGBDR escolhido em SQL, inserimos a conversão de cada tabela do modelo lógico e iniciamos a fase de revisão, inserção de pormenores e detalhes, bem como a correção de erros do nosso trabalho.

De um modo sucinto, sentimos que apesar de não ser um trabalho complexo, este projeto conseguiu exigir algum esforço e atenção por parte do nosso grupo em vários detalhes importantes que nos levaram a crer que neste momento estamos mais aptos para entender e realizar as diferentes etapas/fases/passos que efetuamos ao longo do projeto, tornando o nosso conhecimento e aptidões sobre esta área mais expandidos.

Referências

Thomas Connolly, TC, Carolyn Begg, CB, 2005, Database System – A Practical
Approach to Design, Implementation, and Management, 4ª edição, Harlow: Pearson
Education Limited