



Universidade do Minho
Mestrado integrado em Engenharia Informática
Licenciatura em Ciências da Computação

Unidade Curricular de
Bases de Dados
Ano Letivo de 2016/2017

Gestão de dados das reservas da agência de viagens *Antunes & Bastos, Lda.*

Bruno Pereira (a72628), Nadine Oliveira(a75614),
Rodrigo Ferreira (a74216), Ana Marques (a75248)

Novembro, 2016

BD

Data de Receção	
Responsável	
Avaliação	
Observações	

Gestão de dados das reservas da agência de viagens *Antunes & Bastos, Lda.*

Bruno Pereira(a72628), Nadine Oliveira(a75614),
Rodrigo Ferreira(a74216), Ana Marques(a75248)

Novembro, 2016

Resumo

O projeto realizado consiste na implementação de uma base de dados para a gestão de reservas de viagens de comboio nacionais e internacionais da agência de viagens Antunes & Bastos, Lda. O projeto está dividido em três etapas: modelo concetual, lógico e físico.

A primeira etapa inclui a criação de um modelo de dados concetual. O primeiro passo na realização deste trabalho foi o levantamento dos requisitos a ter em conta, de acordo com as necessidades e intenções dos clientes para com a empresa, para a consequente definição de entidades, e os relacionamentos entre estas, para a sua inclusão no modelo conceptual. Note-se que, houve um processo de validação do modelo de dados segundo os requisitos e transações.

Em seguida, a segunda etapa teve por base a conceção de um modelo lógico em sequência da derivação do modelo conceptual criado. Além da derivação, validou-se o modelo de dados segundo regras de normalização e transações, como também se impôs regras de integridade. Adicionalmente, estimou-se o tamanho inicial da base de dados e consequente crescimento futuro.

Finalmente, a fase final do projeto tem por base a criação de um modelo de dados físico, a partir do modelo lógico anteriormente descrito. O primeiro passo foi a tradução do modelo de dados lógico para um SGBD, e em seguida, a respetiva criação (modelo físico) e povoamento da base de dados em SQL (*Structured Query Language*). De forma análoga ao modelo lógico e concetual, analisaram-se transações, aqui traduzidas para a linguagem SQL. Em seguida, realizou uma nova estimativa do tamanho da base de dados, em termos de ocupação de espaço em disco, e por último definiram-se vistas de utilização e regras de acesso.

Área de Aplicação: Desenho e arquitetura de Sistemas de Bases de Dados.

Palavras-Chave: Modelação Conceptual, Modelação Lógica, Modelação Física, Notação *Chen*, Diagramas ER, Normalização Bases de Dados Relacionais, Gestão de Índices, *Structured Query Language*.

Índice

Resumo	i
Índice	ii
Índice de Figuras	v
Índice de Tabelas	vi
1. Introdução	7
1.1. Contextualização	7
1.2. Apresentação do Caso de Estudo	8
1.3. Motivação	9
1.4. Objetivos	10
1.5. Estrutura do Relatório	10
2. Modelo Conceitual	11
2.1. Levantamento de Requisitos	11
2.2. Identificação de Entidades	13
2.2.1. Comboio	13
2.2.2. Percurso e Itinerário	13
2.2.3. Cliente	14
2.2.4. Reserva	14
2.3. Identificação de Relacionamentos	14
2.3.1 Cliente – Reserva	15
2.3.2 Reserva – Itinerário	15
2.3.3 Comboio – Percurso	15
2.3.4 Percurso - Itinerário	15
2.4. Identificação de Atributos	16
2.4.1 Atributos de Cliente	17
2.4.2 Atributos de Reserva	18
2.4.3 Atributos de Comboio	19
2.4.4 Atributos de Percurso	19
2.4.5 Atributos de Itinerário	20
2.5. Identificação das chaves	20
2.5.1 Comboio	20
2.5.2 Percurso	20
2.5.3 Itinerário	21
2.5.4 Reserva	21
2.5.5 Cliente	21

2.6. Validação do Modelo Conceptual segundo as transações	22
2.6.1 Transações simples	22
2.6.2 Transações mais complexas	27
2.7. Modelo Conceptual Final	29
3. Modelo lógico	30
3.1. Derivação dos Relacionamentos	30
3.1.1 Derivação do Modelo para Obtenção de Tabelas	30
3.1.2 Escolha das Chaves Estrangeiras	32
3.2. Validação segundo Regras de Normalização	35
3.2.1 1ª Forma Normal (1FN)	35
3.2.2 2ª Forma Normal (2FN)	35
3.2.3 3ª Forma Normal (3FN)	38
3.3. Validação do modelo lógico segundo Transações	39
3.3.1 Obter dados para documento comprovativo de compra	40
3.3.2 Efetuar uma reserva de um ou vários lugares	43
3.3.3 Consultar os lugares livres de um dado itinerário	45
3.4. Verificação das Restrições de Integridade	47
3.4.1 Integridade de Domínio	47
3.4.2 Integridade de Entidade	47
3.4.3 Integridade Referencial	47
3.4.4 Restrições de Multiplicidade	48
3.4.5 Restrições Gerais	48
3.5. Tamanho inicial e Crescimento Futuro	49
3.5.1 Tamanho Inicial	49
3.5.2 Crescimento Futuro	51
3.6. Validação de modelo lógico com utilizador	52
4. Modelo físico	53
4.1. Tradução e implementação do modelo lógico para um SGBD	53
4.1.1 Escolha do SGDB alvo para o modelo físico	53
4.1.2 Relações Base	53
4.1.3 Representação de Atributos Derivados	57
4.1.4 Restrições Gerais	57
4.2. Escolha de Índices	58
4.3. Análise de Transações	59
4.3.1 Obter dados para documento comprovativo de compra	60
4.3.2 Efetuar uma reserva de um ou vários lugares	60
4.3.3 Consultar os lugares livres de um dado itinerário	64

4.4. Estimativa dos Requisitos de Espaço em Disco	67
4.4.1 Povoamento e tamanho inicial	67
4.4.2 Crescimento Futuro	68
4.5. Definição das Vistas de Utilização e Regras de Acesso	68
5. Conclusões e Trabalho Futuro	70
Referências	71
Lista de Siglas e Acrónimos	72
Anexos	73
A. Dicionário de dados - Entidades	74
B. Dicionário de Dados – Relacionamentos	75
C. Dicionário de Dados – Atributos	76
D. Dicionário de Dados-Tabelas	81
E. Dicionário De Dados-Atributos das Tabelas	82

Índice de Figuras

Figura 1 – Entidade Cliente	22
Figura 2 – Entidade comboio.	23
Figura 3 – Entidade Percurso	24
Figura 4 – Entidade Itinerário	24
Figura 5 – Relacionamento Reserva – Itinerário.	25
Figura 6 – Relacionamento Cliente – Reserva.	25
Figura 7 – Relacionamento Percurso – Itinerário	26
Figura 8 – Relacionamento Comboio – Percurso	26
Figura 9 – Entidade Reserva	28
Figura 10 – Modelo concetual final	29
Figura 11 - Esquema Lógico Final	34
Figura 12 – Mapa de Transação - Documento comprovativo.	41
Figura 13 - Mapa de Transação - Efetuar uma reserva	44
Figura 14 - Mapa de Transação – Consultar lugares livres	46
Figura 15 - Tabela <i>Cliente</i>	55
Figura 16 - Tabela <i>Reserva</i>	55
Figura 17 - Tabela <i>Percurso</i>	55
Figura 18 - Tabela <i>Itinerario</i>	55
Figura 19 - Tabela <i>Comboio</i>	56
Figura 20 - Tabela <i>ComboioLugar</i>	56
Figura 21 - Tabela <i>ReservaBilhete</i>	56
Figura 22 – Índice em DataReserva	59
Figura 23 – Índices secundários em Itinerario	59
Figura 24 – RIGHT JOIN	65
Figura 25 – N° de registo e tamanho inicial das tabelas	67
Figura 26 – Tamanho total em disco	68

Índice de Tabelas

Tabela 1 - Permissões de utilizador	69
Tabela 2 – Dicionário de dados – Entidades	74
Tabela 3 – Dicionário de dados Relacionamentos	75
Tabela 4 – Atributos de Cliente	76
Tabela 5 – Atributos de Reserva	77
Tabela 6 – Atributos de Comboio	78
Tabela 7 – Atributos de Percurso	79
Tabela 8 – Atributos de Itinerário	80
Tabela 9 – Tabela com Relacionamentos – modelo lógico	81
Tabela 10 – Tabela Cliente	82
Tabela 11 – Tabela Reserva	83
Tabela 12 – Tabela Comboio	84
Tabela 13 – Tabela Percurso	85
Tabela 14 – Tabela Itinerario	86
Tabela 15 – Tabela ComboioLugar	87
Tabela 16 – Tabela ReservaBilhete	88

1. Introdução

O projeto consiste na implementação de uma base de dados para a gestão de reservas de viagens de comboio nacionais e internacionais da agência de viagens *Antunes & Bastos, Lda*. Neste capítulo faz-se a contextualização do caso de estudo, onde se descreve o setor de mercado do turismo, em particular, das viagens nacionais e internacionais de comboio, onde a empresa está inserida. Seguidamente faz-se a apresentação do caso de estudo, onde são identificados os principais serviços prestados pela empresa. Por último, apresentam-se as motivações para o desenvolvimento desta base de dados em concreto e os objetivos que se pretende atingir com a mesma.

1.1. Contextualização

O mercado do setor do turismo a nível mundial, variável importante para o desenvolvimento económico e social, tem vindo a crescer devido à procura estimulada pelo aumento do rendimento disponível, pelo crescimento exponencial dos mercados emergentes (p.e. Angola, Brasil, China), mudanças demográficas, tecnológicas e sociais, etc. Na Europa o crescimento, embora menor que noutros destinos como os asiáticos, que têm tido a preferência, continua a ter um volume considerável, sendo os países com maior número de chegadas a França, Espanha, Itália, Reino Unido, Alemanha, Áustria e Federação Russa.

De igual modo, o turismo nacional tem vindo a crescer tentando acompanhar as tendências da procura a nível externo e um dos fatores de sucesso passa pelo investimento na qualidade e sustentabilidade de infraestruturas e equipamentos, tais como meios de transporte. Com a crescente preocupação com o meio ambiente, e consumo de combustíveis fósseis, causadores de poluição, os impostos sobre estes têm aumentado o custo de transporte, a nível nacional e internacional, e a mais recente crise económica retirou rendimentos a muitas famílias e empresas. Meios de transporte como comboios, ou voos *low-cost* são a preferência de muitos turistas, uma vez que são mais baratos.

Adicionalmente, o uso de tecnologias da informação para gestão de modelos de negócios é praticamente ubíquo, e qualquer atividade que espere crescimento futuro e, que queira ser competitiva, tem que implementar com eficácia um sistema de informação apoiado pelas tecnologias de informação, onde no seu cerne está a gestão dos dados referentes ao sistema de informação.

1.2. Apresentação do Caso de Estudo

A agência de viagens *Antunes & Bastos, Lda.* foi recentemente criada insere-se no contexto anteriormente citado, em particular, viagens nacionais e internacionais de comboio, e como tal, procura uma penetração eficaz nesses mercados, de tal forma que seja possível uma gestão inteligente e eficaz do seu modelo de negócios, tendo em vista a missão da empresa de duplicar o volume de negócios em quatro anos, utilizando um sistema informatizado para gestão de reservas e faturação. Numa primeira fase, a empresa adquiriu à concessão alguns destinos pela Comboios de Portugal (CP), com especial foco nos destinos europeus.

O sistema baseia-se em reservas de viagens naquele meio de transporte e a empresa pretende implementar um sistema de informação de forma a ser competitiva e moderna, que estará integrado com um sistema de software com uma base de dados com sincronização com as bases de dados das empresas ferroviárias com tem a concessão.

A agência de viagens em questão, é um pequeno negócio familiar, onde os próprios donos fazem as reservas de viagens e operações da empresa. As concessões concedidas são os seguintes destinos nacionais e internacionais, onde as viagens são de percurso direto entre duas localizações, sem escalas ou conexões:

- Braga - Porto
- Porto - Braga
- Porto - Lisboa
- Lisboa - Porto
- Lisboa -Faro
- Faro - Lisboa
- Lisboa - Madrid

- Lisboa - Paris
- Lisboa - Amesterdão
- Lisboa - Londres
- Lisboa - Berlim
- Lisboa - Budapeste
- Lisboa - Praga
- Lisboa – Viena

1.3. Motivação

Os sistemas de informação estão a crescer em dimensão e existe uma necessidade premente de uma resposta que para além de eficiente, seja eficaz. O caso de estudo não só permite perceber o funcionamento de um sistema de reservas de viagens de comboios, numa primeira abordagem, bem como entender o funcionamento da agência de viagens.

Além do mais, o caso de estudo pode ser reaproveitado para outros casos de estudo, intimamente ligados com o setor de atividade da empresa, como por exemplo viagens de avião, camioneta, como também outro tipo de sistemas, como sistemas de reservas de quartos em hotelaria ou mesas na restauração, dado que existe uma convergência entre a descrição de meios a ser reservados – lugares, quartos ou mesas – e, de igual modo necessário gerir reservas desses meios. O caso de estudo é bastante genérico, o que permite uma extensão, após nova análise dos requisitos ao modelo de dados de reservas.

De igual modo, as soluções com o modelo de dados adaptado a um novo caso de estudo podem ser de mais baixo custo, caso se deseje implementar algo do género num projeto particular para outra empresa, uma vez que podem integrar um sistema multicamada de software, com manutenção com um maior controlo, podendo ser integradas com outras tecnologias, como JAVA, .NET, etc.

Finalmente, o caso de estudo é uma oportunidade para solidificar de forma estruturada conhecimentos fundamentais na conceção, desenho e implementação de uma base de dados relacional.

1.4. Objetivos

O objetivo principal deste caso de estudo é criar um sistema de Base de Dados que respeite os requisitos das atividades da empresa *Antunes & Bastos, Lda.* de forma a satisfazer os seus utilizadores. Para alcançar este propósito foi seguido um modelo de desenvolvimento de base de dados por etapas: começou-se pela recolha das informações relevantes para a base de dados a desenvolver, seguindo-se uma análise das informações recolhidas, elaboração dos modelos conceptual, lógico e físico em conformidade com os requisitos, seguindo-se uma validação dos mesmos. A documentação de todo o processo encontra-se agregada no presente relatório.

1.5. Estrutura do Relatório

Este relatório está distribuído em quatro capítulos principais para além da introdução e conclusão: modelo concetual, modelo lógico e modelo físico. Além do mais, na conclusão encontra-se uma secção sobre trabalho futuro deste projeto.

O capítulo sobre o modelo conceptual está dividido nas seguintes secções: análise de requisitos do caso de estudo, descrição das entidades identificadas, identificação e descrição dos relacionamentos entre entidades, identificação dos atributos de cada entidade, identificação do domínio dos atributos, identificação das chaves primárias, a partir da identificação de chaves-candidatas, a validação do modelo concetual segundo os requisitos e transações, e a visão global do diagrama do modelo concetual.

Em seguida, encontra-se documentado o processo de transição e validação do modelo conceptual para o modelo lógico e a análise do crescimento do modelo de dadas.

Finalmente, a modelação física é descrita, nomeadamente a implementação do sistema no sistema de gestão de bases de dados escolhido, juntamente com a justificação da sua utilização e descrição do processo de implementação da base de dados.

2. Modelo Conceitual

Como já foi previamente mencionado, este capítulo aborda todo o processo de conceção e modelação dos dados num modelo conceitual, em notação *Chen*, nomeadamente o levantamento e descrição dos requisitos do modelo de dados, bem como o processo de identificação de todos os componentes pertinentes para a conceção do modelo de dados, i.e., entidades, relacionamentos, atributos, como também as chaves identificadas para cada entidade. A última parte deste capítulo refere-se à validação do modelo de dados com os requisitos descritos.

2.1. Levantamento de Requisitos

A primeira fase de qualquer projeto começa pela investigação e identificação dos requisitos nucleares do modelo de dados, que descrevem operações essenciais do funcionamento do sistema de informação em causa. Em particular neste projeto, esses dados fundamentais obtiveram-se junto da empresa, que descreveu o processo como similar ao processo de reserva de bilhete *online* do sítio da internet, com a exceção a forma de pagamento que era por cartão de crédito, entre outras simplificações, que puderam ser observadas na reserva de bilhete na CP.

Do mesmo modo, houve lugar para outras pesquisas, nomeadamente a outros modelos de dados, similares ao tema central do sistema requerido, i.e., modelos de dados de reservas de viagens de aviões, comboios e camionetas, que puderam ser encontradas *online*.

Adicionalmente, consultaram-se vários entendidos em regime de consultadoria para obter uma segunda opinião.

Com efeito, a empresa pretende ter um registo do comboio e os lugares que o compõem. Dado que a empresa tem destinos à concessão, os comboios não puderam ser escolhidos pela empresa, pelo que a empresa ferroviária (CP) concedeu apenas comboios com o mesmo tamanho e, que não têm diferenciação de classe entre as carruagens. O número máximo de lugares é de 200, sendo estes repartidos em 4 carruagens e os lugares são numerados no intervalo de 1 a 200, independentemente das carruagens, sendo estas é numerada de 1 a 4. Adicionalmente cada comboio tem um número único associado, e dado que um comboio dificilmente deixa de existir, esse

número persiste. Por uma questão de completude, as empresas ferroviárias enviaram a marca e o modelo do comboio.

De igual modo, a empresa tem apenas alguns percursos concedidos, sendo estes descritos por um local de partida, um local de chegada. Um comboio pode fazer vários percursos, uma vez que pode, no término de um percurso, fazer outro percurso. Por exemplo, um comboio que faça o percurso Braga-Lisboa, que parte às 8 horas da manhã de determinado dia, chega às 10 horas ao seu destino. No entanto, existe um outro itinerário às 10 horas de Lisboa para Madrid, onde o comboio que chegou de Braga pode fazer. Além disso, cada percurso pode ter mais que um itinerário, onde entra a componente temporal, i.e., hora de partida, chegada e data da realização do itinerário. Cada itinerário possui um código identificador, tal como o percurso e o preço desse percurso.

A agência efetua uma reserva, introduzindo os dados do cliente no registo, caso não exista, para efeito de faturação. Como tal, é necessário introduzir o nome, contacto telefónico, número do documento de identificação, o número de contribuinte e a sua morada (rua, localidade, código postal e país).

A reserva pode ter uma lista de bilhetes até um máximo de 4, onde, em cada elemento, estão discriminados o número do bilhete, o número do lugar, o tipo de lugar e o número de carruagem. Adicionalmente, a empresa quer que em cada reserva exista o total dessa reserva. No entanto, a agência pesquisa, em primeiro lugar, o número de lugares disponíveis para serem reservados no itinerário escolhido pelo cliente e, caso esteja disponível, efetua o registo da reserva. Esta ordem é necessária, uma vez que, a reserva esteja registada, não pode ser cancelada. O cliente paga no ato de reserva, *in loco* na agência de viagens. Adicionalmente, a reserva possui data de pagamento, para efeitos de faturação e contabilidade, sendo o documento fatura entregue ao cliente e essa entrega pode ser feita no balcão da agência ou pelo correio.

Em suma, as seguintes operações têm que ser possíveis:

- Registrar um cliente com nome, número de identificação, número de contribuinte e morada (rua, localidade, código postal e país).
- Saber a informação de um comboio, incluindo todos os lugares
- Saber o preço, o local de partida e o local de chegada de um percurso.
- Saber as horas de partida, horas de chegada de um itinerário num intervalo de data de partida e/ou datas de chegada.
- Saber os percursos de um dado comboio.

- Saber as reservas de um dado itinerário.
- Saber o itinerário de uma dada reserva.
- Saber os itinerários de um dado percurso.
- Saber os dados de reserva de um dado cliente.
- Consultar os dados de um cliente, pelo seu número de documento de identificação, que tem uma determinada reserva efetuada. A reserva deve ter data de pagamento e possui uma lista de bilhete^o de s (nº de bilhete, nº de lugar, nº de carruagem e tipo de lugar) num itinerário, com determinada data e hora de partida e de chegada. De igual modo, este itinerário terá o local de partida e local de chegada e, associados a estes o número de comboio. Com estes dados pretende-se criar um documento tipo fatura.
- Efetuar uma reserva de um ou vários lugares de determinado tipo (janela ou coxia), numa ou várias carruagens, num dado itinerário, com a data de pagamento respetiva.
- Consultar os lugares livres de um comboio dado um itinerário.

2.2. Identificação de Entidades

Esta secção contém uma breve descrição das entidades do modelo de dados, justificadas pelos requisitos já mencionados. Em anexo (Anexo I.A) encontra-se um dicionário de dados com a respetiva informação mais completa.

2.2.1. Comboio

Para efetuar uma reserva é necessário haver a descrição sucinta desses lugares, tal como está descrito nos requisitos, i.e., é necessário saber os lugares existentes para reservar os mesmos. Uma vez que, a agência de viagens reserva viagens de comboios, naturalmente que tem que existir uma entidade comboio para efetuar uma reserva de um lugar nesse comboio.

2.2.2. Percurso e Itinerário

O percurso e itinerário estão intimamente ligados ao modelo de negócios da empresa. Um comboio precisa de efetuar um percurso com local de partida e chegada, e, de igual modo, o conceito de viagem impõe que haja uma hora de partida e chegada, bem como a data da sua realização. A descrição de percurso e o itinerário é necessária, para indicar a disponibilidade de percurso e itinerário ao cliente para efetuar uma

reserva. De igual modo, um comboio pode efetuar vários percursos, e cada percurso, vários itinerários. Com efeito, pretende-se que não haja redundância no modelo de dados e, deste modo, há que categorizar de forma independente e hierárquica as ocorrências de percurso e itinerário. Assim sendo, percurso e itinerário são entidades no modelo de dados, onde um percurso referencia um comboio, um itinerário referencia um percurso, e por outro lado um comboio pode estar em vários percursos e um percurso em vários itinerários. Em consequência, não há redundância de dados.

2.2.3. Cliente

De acordo com os requisitos, a reserva é feita para um cliente, podendo este reservar vários bilhetes, até um número máximo de 4. Existe um conjunto de propriedades que é necessário existir para determinadas operações da empresa, nomeadamente para faturação e futuras reservas. Daí a identificação do cliente pelo seu nome, número de documento de identificação, com ênfase nesta última, uma vez que identifica o cliente de forma unívoca.

2.2.4. Reserva

A reserva é o ponto central do modelo de negócios da empresa, uma vez que é a operação base da empresa, e consequentemente o ponto unificador do modelo de dados. A agência efetua uma reserva de bilhetes numa viagem para um cliente, i.e., a agência regista a escolha da viagem numa reserva, com o seu itinerário e a lista de bilhetes associados a um lugar, bem como a data de pagamento, total da reserva e total de bilhetes. Apesar de reserva poder ser um relacionamento, as propriedades que a agência pretende numa reserva como data de pagamento, total de reserva, total de bilhetes e lista de bilhetes, justificam uma existência única e concetual, uma vez que haverá um determinado número de operações na reserva, bem como existe a necessidade por parte da empresa de entregar um documento do tipo fatura, ao cliente, que terá por base a reserva e todas as outras entidades que se relacionam com aquela.

2.3. Identificação de Relacionamentos

Após identificação das entidades, procuram-se relacionamentos entre as mesmas, conforme os requisitos. O dicionário de dados referente aos relacionamentos esta no Anexo **Erro! A origem da referência não foi encontrada..**

2.3.1 Cliente – Reserva

Relacionamento: uma reserva é feita por um cliente

Descrição: identifica o ato da reserva do cliente

Cardinalidade: Cliente (1); Reserva (N) - Um cliente pode ter várias reservas, mas uma reserva só pode ser de um cliente.

Participação: Cliente (T); Reserva (T) – Todos os clientes estão associados a uma reserva, e por sua vez, todas as reservas têm associado um cliente

Obrigatoriedade: um livro é obrigatoriamente escrito por um autor e um autor escreve obrigatoriamente um livro.

Atributos: Este relacionamento não tem atributos.

2.3.2 Reserva – Itinerário

Relacionamento: uma reserva tem um itinerário.

Descrição: identifica a referencia de um itinerário a uma reserva

Cardinalidade: Reserva (N); Itinerário (1) – Um itinerário pode estar em várias reservas, mas uma reserva só tem um itinerário.

Participação: Reserva (P); Itinerário (T) – Todas as reservas têm que ter um itinerário, mas pode haver itinerários sem reservas.

Atributos: Este relacionamento não tem atributos.

2.3.3 Comboio – Percurso

Relacionamento: um comboio percorre um percurso.

Descrição: identifica a associação de um comboio a um percurso.

Cardinalidade: Comboio (1); Percurso (N) – Um comboio pode fazer vários percursos, mas um percurso só pode ser feito por um comboio.

Participação: Comboio (T); Percurso (T) – Todos os comboios têm que ter um percurso, e todos os percursos têm que ter um comboio associado.

Atributos: Este relacionamento não tem atributos.

2.3.4 Percurso - Itinerário

Relacionamento: um comboio percorre um percurso.

Descrição: identifica a associação de um comboio a um percurso.

Cardinalidade: Comboio (1); Percurso (N) – Um comboio pode fazer vários percursos, mas um percurso só pode ser feito por um comboio.

Participação: Comboio (T); Percurso (T) – Todos os comboios têm que ter um percurso, e todos os percursos têm que ter um comboio associado.

Atributos: Este relacionamento não tem atributos.

2.4. Identificação de Atributos

A caracterização das diferentes entidades e de alguns relacionamentos foi conseguida com a identificação dos atributos que os caracterizam. No anexo C encontra-se a descrição dos atributos de cada entidade, nos respetivos dicionários de dados. Nesta secção, pretende-se complementar os dicionários de dados com algumas justificações face aos requisitos que levaram a considerar os atributos que temos no esquema conceptual.

2.4.1 Atributos de Cliente

Os atributos que descrevem funcionalmente esta entidade, para além do identificador são:

- Número de documento de identificação.
- Nome do cliente
- Morada
 - Rua
 - Localidade
 - Código Postal
 - País

Pretende -se que o cliente seja pesquisável de forma intuitiva, daí o número de documento de identificação e também é necessário para incluir numa fatura. Note-se que, não se considera uma fatura como entidade, uma vez que o documento depende mais do que uma entidade, pode ser gerado através de apenas uma consulta, sendo o documento entregue ao cliente. A entrega pode ser feita no balcão da agência ou pelo correio. Como tal, para criar esse tipo de documento são necessários a morada e o nome do cliente. Note-se que, a morada é um atributo composto, uma vez que se precisa apenas de uma referência da rua, localidade, código postal e país.

2.4.2 Atributos de Reserva

Para além do número identificador de reserva, a entidade tem como atributos:

- Número de bilhete
- Total de Bilhetes
- Total de Reserva
- Data de reserva
- Bilhete
 - Número de bilhete
 - Número de lugar
 - Tipo de lugar
 - Número de carruagem

O atributo bilhete é um atributo multivalor composto, onde cada elemento bilhete é constituído pelo número de bilhete, número de lugar, tipo de lugar (J - Janela - ou C - Cooxia) e número de carruagem. Cada número do bilhete é obtido junto da empresa ferroviária e inserido na base de dados, sendo este único e não transmissível, bem como não pode ser reservado cada lugar mais do que uma vez. O total de reserva – atributo derivado – irá conter o valor total da reserva, calculado com base no número total de bilhetes – atributo também derivado – e o preço de um percurso. A data de reserva é necessária para saber quando um cliente efetuou uma reserva, uma vez que é para efeito de faturação e contabilidade.

2.4.3 Atributos de Comboio

Os atributos de um comboio são os seguintes:

- Número de comboio.
- Marca
- Modelo
- Lugar
 - Número de lugar
 - Tipo de lugar
 - Número de carruagem

O número de comboio é o identificador único do comboio e corresponde ao número do comboio da empresa ferroviária. A marca e o modelo descrevem o comboio e lugar descreve todos os lugares do comboio. Note-se que, o lugar é um atributo multivalor composto, que inclui o número de lugar, o tipo de lugar (J -Janela, C – Cobia) e o número de carruagem.

2.4.4 Atributos de Percurso

Os atributos de percurso, para além do identificador de percurso são:

- Local de partida
- Local de chegada
- Preço

O local de partida descreve o local onde começa a viagem e, o local de chegada descreve o local onde termina a viagem. O preço é o custo dessa viagem, nesse percurso.

2.4.5 Atributos de Itinerário

Os atributos de itinerário, para além do identificador de itinerário são:

- Hora de partida
- Hora de chegada
- Data de partida
- Data de chegada

A data e a hora de partida descrevem o instante temporal onde começa uma viagem e, a data e a hora de chegada descrevem o instante temporal onde termina a viagem.

2.5. Identificação das chaves

2.5.1 Comboio

Chaves candidata: NroComboio

Chave primária escolhida: NroComboio

Justificação: O único atributo que pode identificar univocamente uma instância de entidade.

Chaves secundárias: nenhuma.

2.5.2 Percurso

Chaves candidata: idPercurso.

Chave primária escolhida: idPercurso.

Justificação: O único atributo que pode identificar univocamente uma instância de entidade.

Chaves secundárias: nenhuma

2.5.3 Itinerário

Chaves candidata: idItinerario.

Chave primária escolhida: idItinerario.

Justificação: O único atributo que pode identificar univocamente uma instância de entidade.

Chaves secundárias: nenhuma.

2.5.4 Reserva

Chaves candidata: idReserva.

Chave primária escolhida: idReserva.

Justificação: O único atributo que pode identificar univocamente uma instância de entidade.

Chaves secundárias: nenhuma.

2.5.5 Cliente

Chaves candidata: idCliente, NroDocIdentif

Chave primária escolhida: idCliente

Justificação: Atributo de menor tamanho

Chaves secundárias: NroDocIdentif

2.6. Validação do Modelo Conceptual segundo as transações

Para validar o modelo corretamente devem ser possíveis as transações descritas nos requisitos. Neste capítulo pretende-se analisar se transações mais simples e mais complexas. As transações mais simples usam uma entidade e seus atributos ou relacionamentos binários entre entidades, enquanto as mais complexas necessitam de ver mais que duas entidades e mais que um relacionamento.

2.6.1 Transações simples

As transações simples têm que ser possíveis:

- Registrar um cliente com nome, número de identificação, número de contribuinte e morada (rua, localidade, código postal e país).

A entidade Cliente guarda os dados referentes ao cliente, nomeadamente nome, nº de documento de identificação e morada (rua, localidade, cód. postal e país).

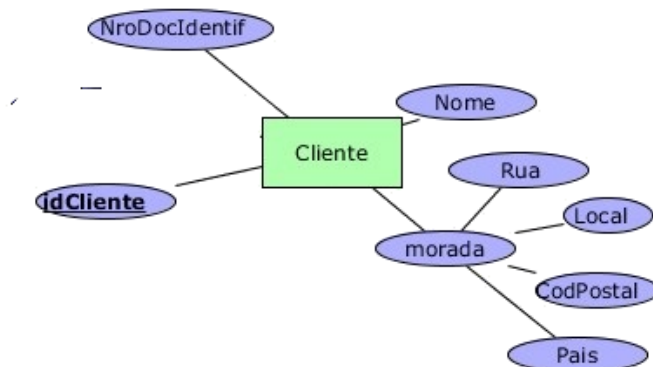


Figura 1 – Entidade Cliente

- Saber a informação de um comboio, incluindo todos os lugares.

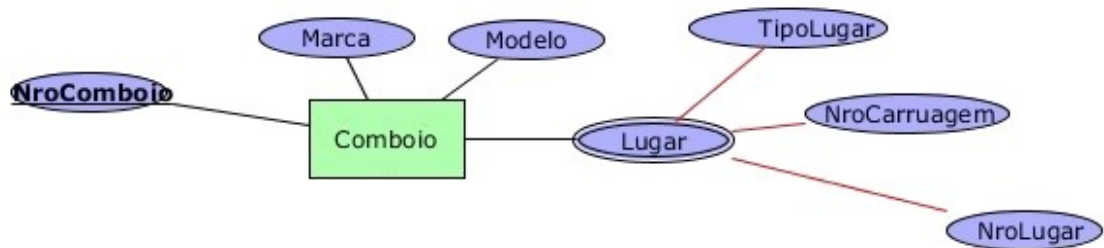


Figura 2 – Entidade comboio.

A entidade Comboio possui informação sobre todos os lugares de um comboio (tipo de lugar, nº de carruagem e nº de lugar), incluindo dados sobre o comboio, como marca e modelo.

- Saber o preço, o local de partida e o local de chegada de um percurso.

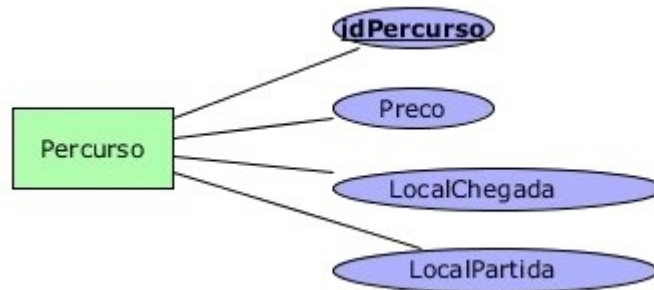


Figura 3 – Entidade Percurso

A entidade Percurso tem como atributo os dados necessários para saber toda a informação de um percurso.

- Saber as horas de partida, horas de chegada de um itinerário num intervalo de data de partida e/ou datas de chegada.

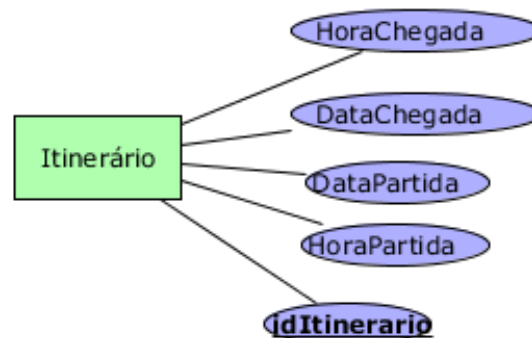


Figura 4 – Entidade Itinerário

A entidade Itinerário tem como atributo os dados necessários para saber toda a informação de um itinerário.

- Saber as reservas de um dado itinerário.
- Saber o itinerário de uma dada reserva.
- Reservar um local de determinado tipo (janela ou coxia), numa determinada carruagem, num dado itinerário.

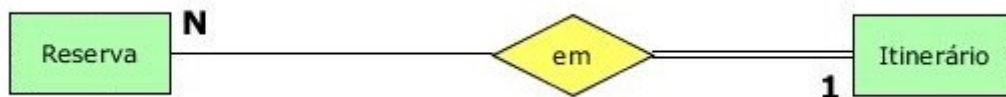


Figura 5 – Relacionamento Reserva – Itinerário.

As três operações necessitam de ter o relacionamento Reserva – Itinerário, uma vez que, a Reserva referencia o itinerário.

- Saber os dados de reserva de um dado cliente.



Figura 6 – Relacionamento Cliente – Reserva.

O relacionamento Cliente – Reserva permite associar os dados do cliente com a sua reserva.

- Saber os itinerários de um dado percurso.



Figura 7 – Relacionamento Percurso – Itinerário

O relacionamento Percurso – Itinerário permite associar os dados do percurso com o seu itinerário.

- Saber os percursos de um dado comboio.

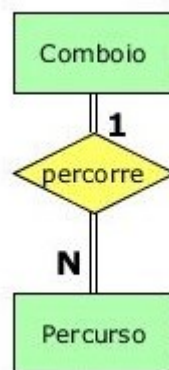


Figura 8 – Relacionamento Comboio – Percurso

O relacionamento Comboio – Percurso permite associar os dados do comboio com o seu percurso, para obter os todos os percursos de um comboio.

2.6.2 Transações mais complexas

Ao nível mais complexo, i.e., transações necessárias que atravessem o modelo de dados, usando vários relacionamentos, entidades e atributos, tem-se as seguintes operações:

- Consultar os dados de um cliente, pelo seu número de documento de identificação, que tem uma determinada reserva efetuada. A reserva deve ter data de pagamento e possui uma lista de bilheteº de s (nº de bilhete, nº de lugar, nº de carruagem e tipo de lugar) num itinerário, com determinada data e hora de partida e de chegada. De igual modo, este itinerário terá o local de partida e local de chegada e, associados a estes o número de comboio. Com estes dados pretende-se criar um documento tipo fatura.

A entidade Cliente tem todos os dados referentes a um cliente, como anteriormente referenciado, que pode ser identificado pelo seu número de documento de identificação. Pode-se consultar os dados de reserva deste cliente e lista de bilhetes, filtrando pela data da reserva, que por sua vez tem referenciada um itinerário. Com efeito, pode-se obter todos os dados da entidade Itinerário de uma forma mais intuitiva, com a data e hora de partida e a data e hora de chegada. A partir da entidade anterior, pode-se obter o local de partida e o local de chegada e o preço do percurso associado, e uma vez que este é referenciado pela entidade Comboio, obtemos o número do comboio. Note-se que esta transação atravessa o modelo de dados e menciona todas as entidades, pelo que não se coloca aqui um diagrama.

- Efetuar uma reserva de um ou vários lugares de determinado tipo (janela ou coxia), numa ou várias carruagens, num dado itinerário, com a data de pagamento respetiva.

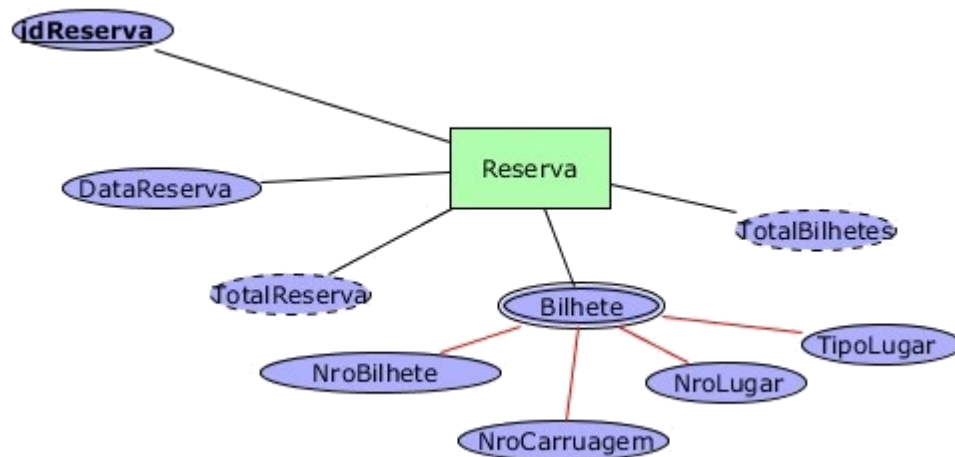


Figura 9 – Entidade Reserva

A entidade Reserva agrupa todos os dados, suportando a operação de reserva, onde o total de bilhetes e o total de reserva guardam o nº de bilhetes a ser reservados e o preço total da reserva, respetivamente. Para cada reserva existe uma lista de bilhetes com o número de bilhete, o número de lugar, o número de lugar e o tipo de lugar. Para registar a reserva, esta tem que ser feita de forma atómica, onde é inserido a data de reserva, o itinerário e associados a ela são inseridos os bilhetes. O total de reserva e o total de bilhetes é atualizado, consoante o preço, que é obtido do preço do itinerário e multiplicado pelo total de bilhetes pretendido (max. 4 bilhetes).

- Consultar os lugares livres de um comboio dado um itinerário.

Para consultar os lugares ocupados de um itinerário é necessário obter todos os lugares nas Reservas, associados àquele itinerário. Por sua vez, pode-se obter os lugares disponíveis do Comboio, por transitividade (como nas outras transações desta secção), uma vez que este é referenciado pelo Percurso, e este pelo Itinerário. Os lugares livres será o resultado da diferença entre lugares disponíveis com lugares ocupados.

2.7. Modelo Conceptual Final

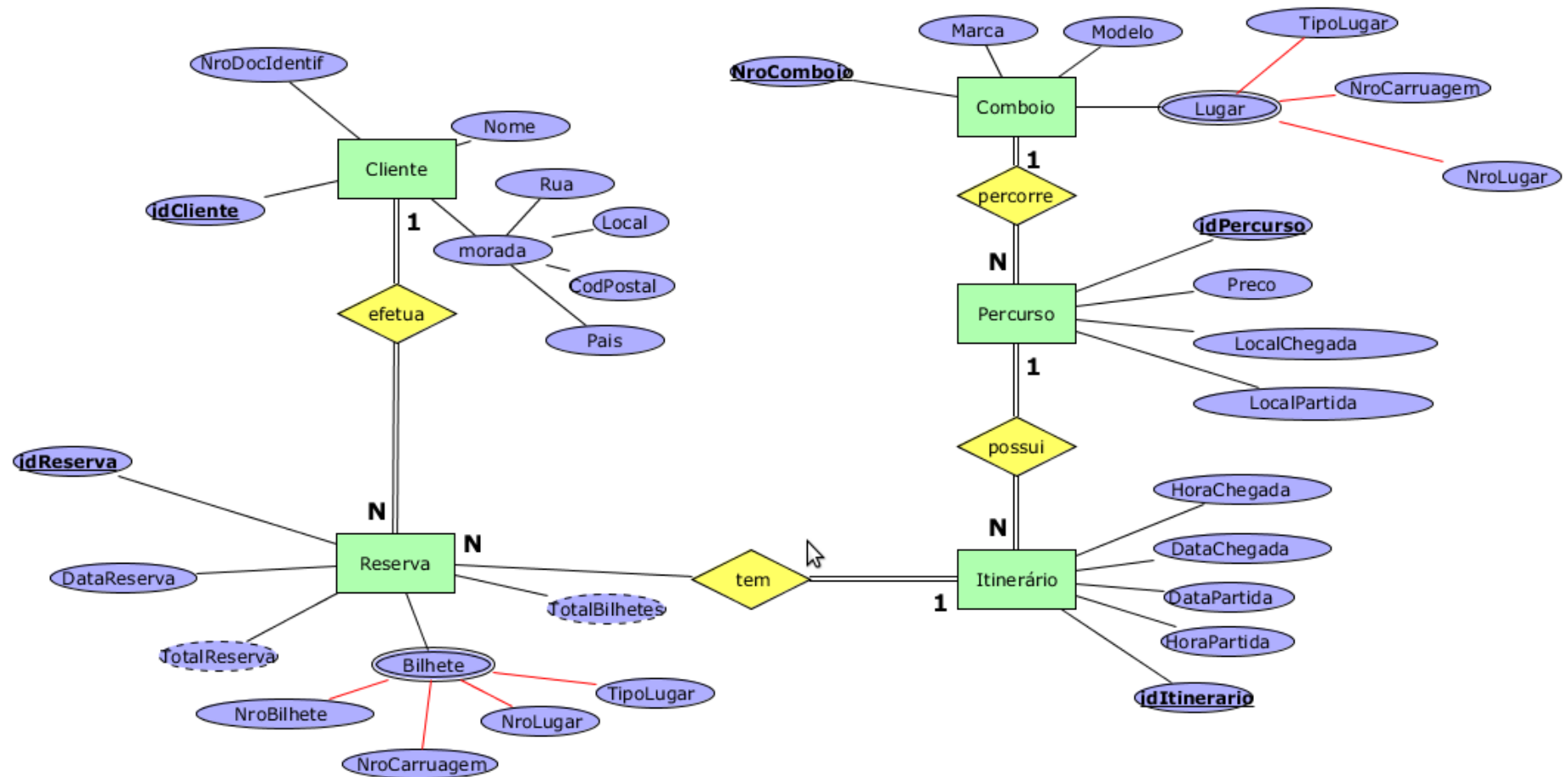


Figura 10 – Modelo concetual final

3. Modelo lógico

Este capítulo explicará o processo de derivação do modelo concetual desenhado para o modelo lógico, nomeadamente os passos para obter a derivação de relacionamentos, alguns tipos de atributos, bem como a validação do modelo de dados segundo a normalização de dados, transações de utilizador, restrições de integridade e otimizações ao modelo de dados. No fim desta etapa, pretende-se a validação do modelo de dados com o utilizador.

3.1. Derivação dos Relacionamentos

3.1.1 Derivação do Modelo para Obtenção de Tabelas

Nesta secção explicita-se a forma de obtenção das tabelas do modelo lógico, apresentando-se os atributos, chaves primárias e estrangeiras correspondentes, como também a motivação da sua ocorrência no modelo de dados lógico. Como complemento, pode-se verificar o Anexo I.E onde figuram as relações, atributos, tipos de atributos, chaves primárias e estrangeiras, como também em que atributos se aplicou um índice e afins.

Cliente = {idCliente, Nome, NrDocIdentif, Rua, Local, CodPostal, Pais}

Chave Primária: idCliente

Chave Estrangeira: não tem

Observações: Relação de base. *Cliente* é uma entidade no modelo conceptual. *Rua*, *Local*, *CodPostal* e *Pais* derivam de um atributo composto *Morada*.

Reserva = {idReserva, Data de reserva, TotalBilhetes, TotalReserva, Cliente, Itinerario}

Chave Primária: idReserva

Chave Estrangeira: Cliente, Itinerario

Observações: Relação de base. *Reserva* é uma entidade no modelo conceptual. Os atributos *TotalBilhetes* e *TotalReserva* apesar de terem uma tipologia especial (atributo derivado) são elevados a atributo simples no modelo lógico.

Itinerario = {idItinerario, DataHoraPartida, DataHoraChegada, Percurso}

Chave Primária: idItinerario

Chave Estrangeira: Percurso

Observações: Relação de base. *Itinerario* é uma entidade no modelo conceptual.

Note-se que *DataHoraPartida* e *DataHoraChegada* foram criados como otimização do modelo concetual, uma vez que, a data de chegada, a hora de chegada, a data de partida e hora de partida eram todos atributos simples. A otimização deve-se ao facto do tipo de dados utilizado (*DATETIME*) ser de mais fácil uso, caso seja necessário ordenar, efetuar cálculos de duração, bem como está limitado a gama de valores correto (as horas são de 00:00:00 até 23:59:29).

Percurso = {idPercurso, Preço, Local de Partida, Local de Chegada, Comboio}

Chave Primária: idPercurso

Chave Estrangeira: Comboio

Observações: Relação de base. *Percurso* é uma entidade no modelo conceptual.

Comboio = {NroComboio, Marca, Modelo}

Chave Primária: NroComboio

Chave Estrangeira: não tem

Observações: Relação de base. *Comboio* é uma entidade no modelo conceptual.

LugarComboio = {NroLugar, NroBillhete, TipoLugar, NroCarruagem, Comboio}

Chave Primária: NumeroLugar, Comboio

Chave Estrangeira: Reserva

Observações: Derivação do atributo multivalor *Lugar* que pertence á entidade *Reserva*.

ReservaBilhete = {NroLugar, TipoLugar, NroCarruagem, Comboio, Reserva, NroBilhete}

Chave Primária: NumeroLugar, Reserva

Chave Estrangeira: Comboio, NroLugar, Reserva

Observações: Derivação do atributo multivalor *Bilhete* que pertence á entidade *Comboio*. Note-se que como otimização ao modelo de dados para certificar a existência de um lugar num comboio, se adicionou mais um relacionamento entre

LugarComboio e *ReservaLugar* de um para muitos, respetivamente. Daí ter *Comboio* como chave estrangeira (explicado sucintamente na secção

Obtêm-se assim sete tabelas: cinco tabelas de base, uma tabela resultante da derivação do atributo multivalor composto *Lugar* e outra tabela resultante da derivação do atributo multivalor composto *Bilhete*.

3.1.2 Escolha das Chaves Estrangeiras

A secção anterior apresentou a tanto chaves primárias, como estrangeiras, sendo que o objetivo desta secção a acrescentar informação complementar àquilo que foi mencionado sobre chaves estrangeiras. Ou seja, pretende-se enunciar a motivação por trás de cada decisão de escolha de chaves estrangeiras para cada tabela, tendo por base as regras de derivação de um modelo concetual para um modelo lógico e otimizações ao modelo lógico.

Tabela Reserva

- **Cliente** – Este atributo representa uma cópia da chave primária da tabela *Cliente*.

Existe uma relação de 1 (*Cliente*) para N (*Reserva*), deste modo, a tabela *Reserva* funciona como tabela filha e recebe uma cópia da chave primária da tabela mãe *Cliente*.

Tabela Itinerario

- **Percurso** – Este atributo representa uma cópia da chave primária da tabela *Percurso*. Existe uma relação de 1(*Itinerario*) para N(*Reserva*), onde a tabela *Reserva* funciona como tabela filha e recebe uma cópia da chave primária da tabela mãe *Itinerario*.

Tabela Percurso

- **Comboio** – Este atributo representa uma cópia da chave primária da tabela *Comboio*. Existe uma relação de 1(*Comboio*) para N(*Percurso*), onde a tabela *Percurso* funciona como tabela filha e recebe uma cópia da chave primária da tabela mãe *Comboio*.

Tabela LugarComboio

Esta tabela resulta da derivação do atributo multivalor *Lugar* da entidade *Comboio* contendo por isso uma cópia do da chave primária da entidade *Comboio*.

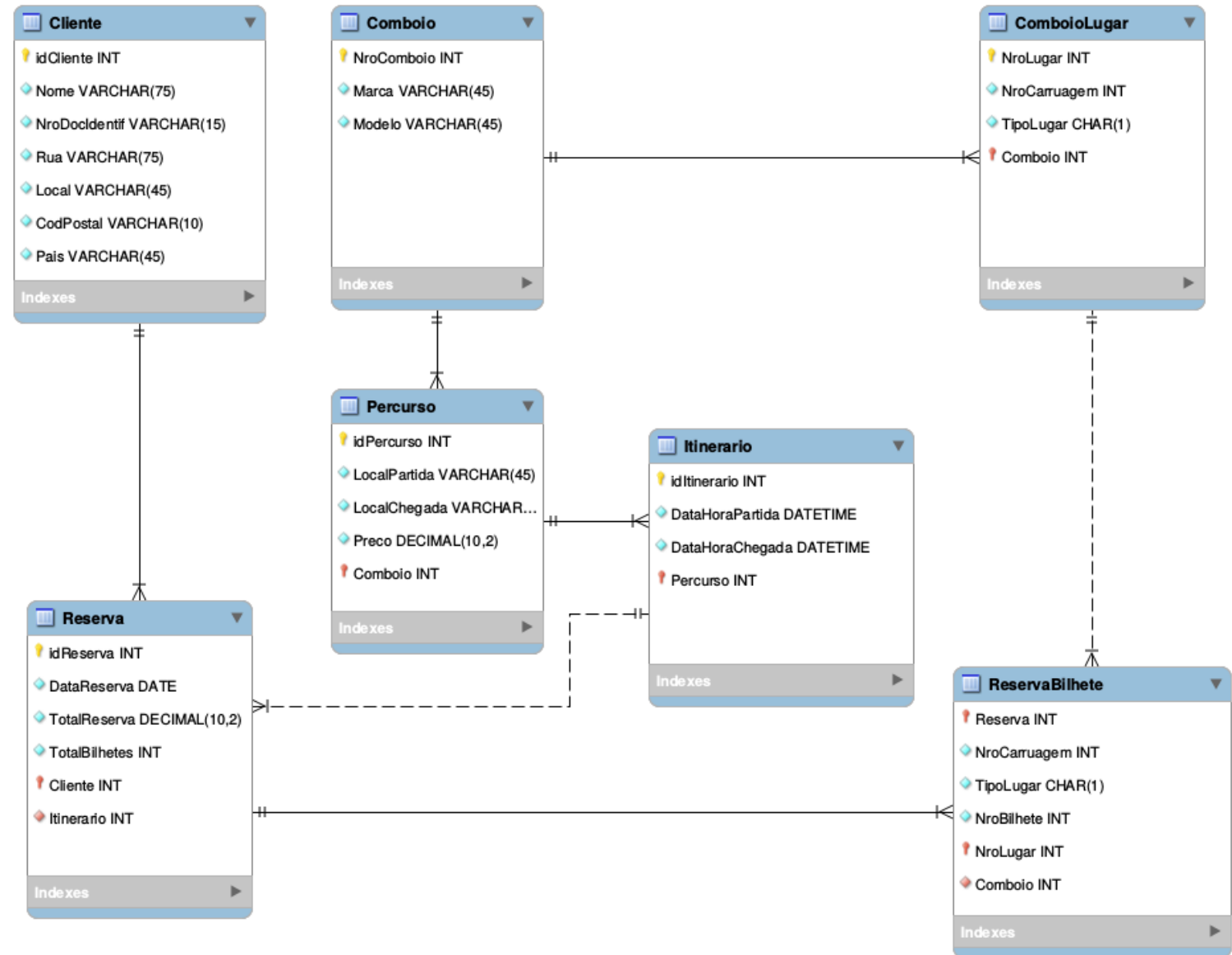
- **Comboio** – Este atributo representa uma cópia da chave primária da tabela *Comboio*.

Tabela ReservaBilhete

Esta tabela resulta da derivação do atributo multivalor Bilhete da entidade Reserva contendo por isso uma cópia do da chave primária da entidade Reserva. Não obstante, além desta chave estrangeira, por motivos de otimização, que não se podia fazer no modelo concetual sem comprometer este, foi adicionada mais uma chave estrangeiras para refletir um relacionamento 1 (*LugarComboio*) para N (*ReservaLugar*).

- **Reserva** – Este atributo representa uma cópia da chave primária da tabela *Reserva*.
- **Comboio, NroLugar** – Este par representa a cópia da chave primária composta de *LugarComboio*. Existe uma relação de 1(*LugarComboio*) para N(*ReservaLugar*), onde a tabela *ReservaLugar* funciona como tabela filha e recebe uma cópia da chave primária da tabela mãe *LugarComboio*.

Figura 11 - Esquema Lógico Final



3.2. Validação segundo Regras de Normalização

Esta secção apresenta os passos efetuados para validar o modelo de dados lógico, segundo as regras de normalização até a terceira forma normal. Deste modo, pretende-se garantir que, em primeiro lugar, o modelo de dados não tem redundâncias, em segundo lugar que o modelo de dados é consistente. De igual modo, pretende-se manter a satisfação dos requisitos dos requisitos.

Com recurso à análise das dependências funcionais entre os atributos de cada tabela, nesta secção apresenta-se o processo de validação segundo as regras de normalização até à 3ª forma normal.

3.2.1 1ª Forma Normal (1FN)

Em cada tabela existe uma chave primária como identificador único, o que garante que não existem conjuntos de atributos repetidos que descrevam as mesmas características ou propriedades. Não existem tabelas com o mesmo nome nem atributos com o nome repetido dentro da mesma tabela como também, os valores assumidos para cada atributo têm uma valoração atómica. Em consequência, o modelo lógico obtido encontra-se na 1ª Forma Normal.

3.2.2 2ª Forma Normal (2FN)

De modo a verificar que as relações existentes se encontram na 2ª Forma normal é necessário que todos os atributos tenham uma dependência funcional total em relação á chave primária. Por conseguinte, eliminam-se as dependências parciais, i.e., dependências que ocorrem quando um atributo depende funcionalmente de atributos que fazem parte da chave primária da relação, mas não de toda a chave ou, as dependências que ocorrem quando um atributo depende apenas de parte de uma chave primária composta. A análise das dependências funcionais em cada relação encontra-se a seguir.

Relação Cliente

Atributos da Relação: idCliente, Nome, NroDocIdentif, Rua, Local, CodPostal, Pais

Dependências Funcionais:

IdCliente - Nome, NroDocIdentif, Rua, Local, CodPostal, Pais (Chave Primária)

NroDocIdentif - IdCliente, Nome, Rua, Local, CodPostal, Pais (Chave Candidata)

Verifica-se que todos os atributos da relação têm uma dependência funcional total em relação à chave primária e candidata.

Relação Reserva

Atributos da Relação: idReserva, DataReserva, TotalReserva, TotalBilhetes, Cliente, Itinerario

Dependências Funcionais:

IdReserva - TipoLugar, NroBilhete, NroCarruagem, DataReserva, NroLugar, TotalReserva, TotalBilhetes, Cliente, Itinerario (Chave Primária)

Verifica-se que todos os atributos da relação têm uma dependência funcional total em relação à chave primária.

Relação Itinerario

Atributos da Relação: idItinerario, DataHoraPartida, DataHoraChegada, Percurso

Dependências Funcionais:

idItinerario - DataHoraPartida, DataHoraChegada, Percurso (Chave Primária).

Verifica-se que todos os atributos da relação têm uma dependência funcional total em relação à chave primária.

Relação Percurso

Atributos da Relação: idPercurso, Preço, LocalPartida, LocalChegada, Comboio

Dependências Funcionais:

IdPercurso - Preço, LocalPartida, LocalChegada, Comboio (Chave Primária).

Verifica-se que todos os atributos da relação têm uma dependência funcional total em relação à chave primária.

Relação Comboio

Atributos da Relação: NroComboio, Marca, Modelo

Dependências Funcionais:

NroComboio -> Marca, Modelo (Chave Primária).

Verifica-se que todos os atributos da relação têm uma dependência funcional total em relação à chave primária.

Relação ComboioLugar

Atributos da Relação: NroLugar, Comboio, TipoLugar, NroCarruagem

Dependências Funcionais:

NroLugar, Comboio - TipoLugar, NroCarruagem, (Chave Primária).

Verifica-se que todos os atributos da relação têm uma dependência funcional total em relação à chave primária composta (um número de carruagem e um tipo de lugar são descritos totalmente, por um lugar num comboio).

Relação ReservaBilhete

Atributos da Relação: Reserva, TipoLugar, NroCarruagem, NroBilhete, NroLugar, Comboio.

Dependências Funcionais:

Reserva, NroLugar - NroCarruagem, TipoLugar, NroBilhete, Comboio (Chave Primária).

NroBilhete - Reserva, NroCarruagem, TipoLugar, NroLugar, Comboio (Chave Candidata).

Verifica-se que todos os atributos da relação têm uma dependência funcional total em relação à chave primária e candidata.

Em suma, todos os atributos de todas relações dependem funcionalmente de forma total das suas chaves primárias e candidatas. Em consequência, pode-se admitir que o modelo de dados está na 2ª forma normal.

3.2.3 3ª Forma Normal (3FN)

Nesta secção pretende-se mostrar que o modelo está na 3ª forma normal, ou seja, o modelo de dados não possui dependências transitivas. Com efeito, dependências transitivas ocorrem quando um atributo que não faz parte da chave primária da relação depende funcionalmente de um atributo que também não faz parte da chave primária da relação. Assim, de acordo com as dependências funcionais apresentadas na Secção 3.2.2, conclui-se que os atributos não chave depende de outros atributos também não chave. Por conseguinte, o modelo de dados encontra-se na 3ª forma normal.

3.3. Validação do modelo lógico segundo Transações

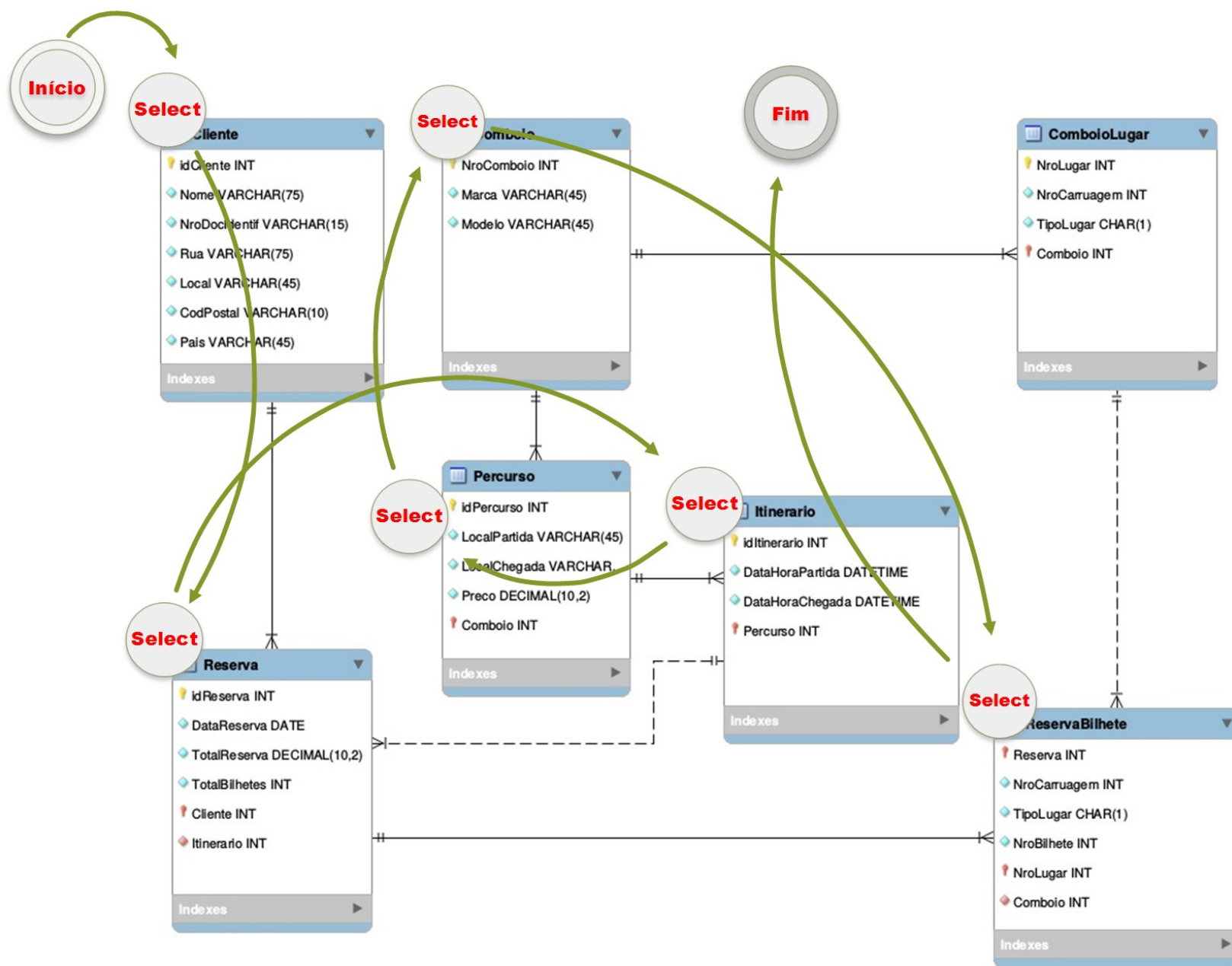
Na secção 2.6, demonstrou-se a validade do modelo após análise das transações. Nessa secção dividiu-se as transações em duas categorias: transações simples e transações complexas. Nesta secção pretende-se apenas analisar a validade do modelo de dados lógico, de acordo com as transações complexas, descrevendo-as e apresentando um mapa de transações para as mesmas.

3.3.1 Obter dados para documento comprovativo de compra

A agência pretende consultar os dados de um cliente e identificar todos os valores de uma reserva numa dada data e num dado itinerário para obter um comprovativo de compra. A transação já foi analisada na Secção 2.6.2, não obstante será analisada do ponto de vista do modelo lógico aqui.

Através das chaves primárias e estrangeiras de cada relação pretende-se fazer uma junção por esses valores das seguintes relações: *Cliente*, *Reserva*, *ReservaBilhete*, *Itinerario*, *Percurso* e *Comboio*. Note-se que, poderia se pensar em obter o valor do nº do comboio através da relação *ReservaBilhete*, no entanto é necessário saber qual o percurso associado a um comboio. Em primeiro lugar obtêm-se todos os dados de um cliente dado o seu identificados, fazendo uma junção com todos os dados da relação *Reserva*. Por sua vez, obtêm-se todos os dados de *Reserva*, fazendo de seguida uma nova junção com a relação *Itinerario*, obtendo também todos os dados desta última. Do *Itinerario* ao *Percurso* com uma nova junção, obtendo, de igual modo os dados de *Percurso*. De seguida faz-se uma junção de *Percurso* com *Comboio*, obtendo deste último apenas o nº de comboio. Por último, faz-se uma junção entre *Reserva* e *ReservaBilhete*, obtendo assim os lugares reservados. Note-se, que se pretende filtrar os dados que não sejam identificadores para uma melhor leitura. Os dados posteriormente serão filtrados pela camada de negócio, para remoção de repetidos.

Figura 12 –
Mapa de
Transação -



Documento comprovativo.

3.3.2 Efetuar uma reserva de um ou vários lugares

Antes de efetuar uma reserva são necessários o preço de percurso e o número de comboio. Assim, é necessário fazer uma consulta na relação *Comboio* com uma junção pela chave estrangeira de *Percurso* com o nome *Comboio*, para fazer uma nova junção com o a relação resultado anterior com pela chave primária de *Percurso* com uma junção pela chave estrangeira de *Itinerario* com mesmo nome *Percurso*, filtrando pelo valor do itinerário pretendido. No fim, projetam-se o valor do preço e valor do número do comboio, guardando estes valores em variáveis.

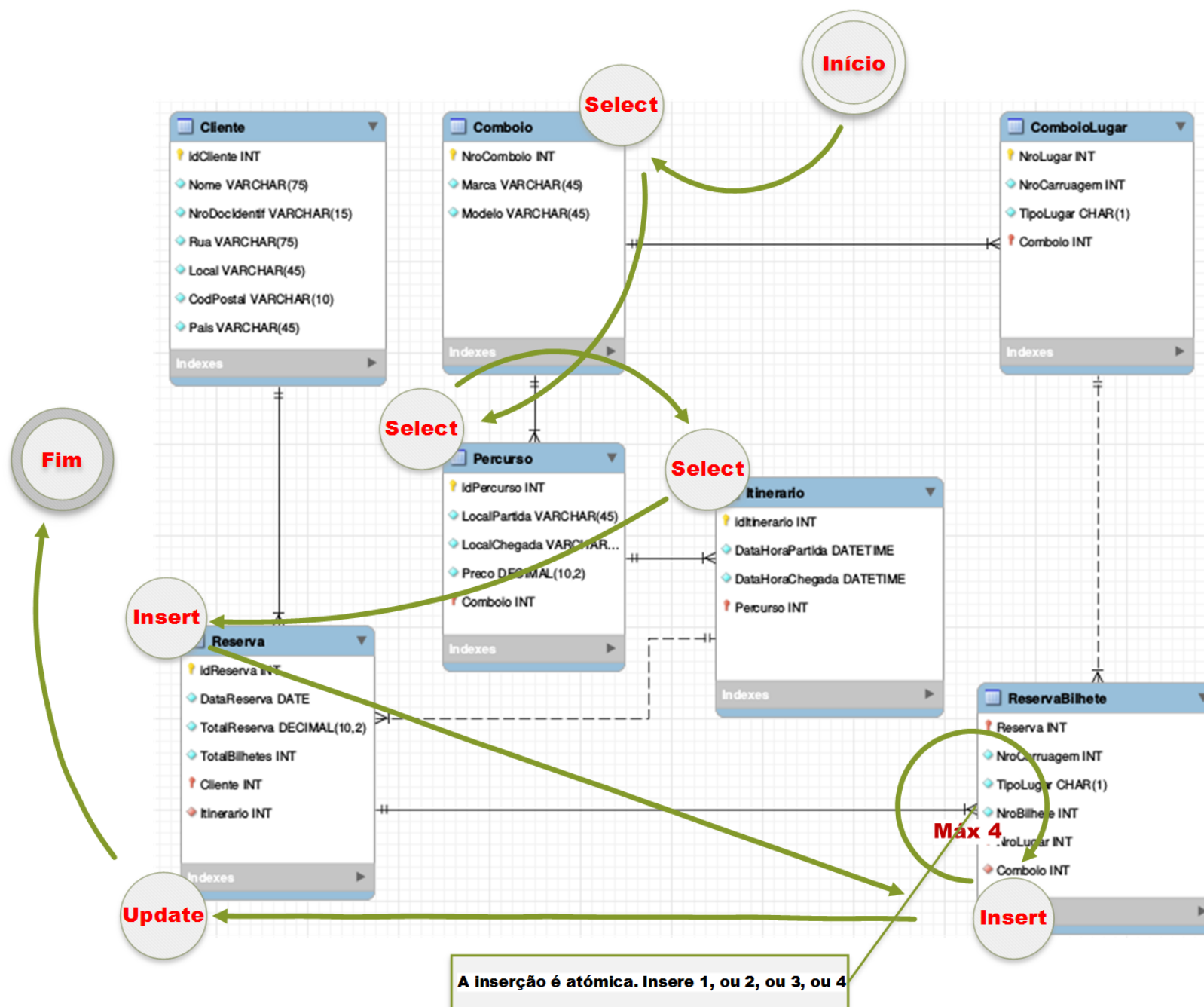
Para efetuar uma reserva é necessário introduzir num procedimento o identificador do cliente, a data de reserva e o identificador do itinerário pretendido, bem como o nº de lugar, nº de carruagem, nº de comboio, tipo de lugar e número de bilhete. Note-se que, pode-se reservar de 1 até 4 bilhetes. No entanto, será feita a explicação para reserva de um bilhete. Assim, prossegue-se o processo de reserva fazendo uma inserção na relação *Reserva*, com os valores da data de reserva, nº de cliente e nº de itinerário.

Em seguida, insere-se na relação *ReservaBilhete* o identificador da reserva, que é o último identificador inserido em *Reserva*, obtido depois com uma consulta do valor máximo na mesma. Além do mais, é necessário introduzir em *ReservaBilhete* o nº de lugar, o nº de comboio, o nº de carruagem, o tipo de lugar e o nº de bilhete. Atente-se que é necessário testar se o tipo de lugar está dentro da gama definida (que será discutida na secção 3.4.5). O valor total da reserva é calculado através do nº de bilhetes inseridos (nº de últimas atualizações na base de dados) ou predefinido no procedimento, cujo valor *TotalBilhetes* ou é atualizado ou inserido diretamente¹, sendo este multiplicado pelo valor do preço de percurso, sendo necessário proceder a atualização do valor de *TotalReserva*.

Apenas se chama a atenção de que os valores de *TotalBilhetes* e *TotalReserva* são predefinidos, como poder-se-á ver no anexo I.E.

¹ Pretende-se inserir diretamente por motivos de eficiência. No entanto, concetualmente, este é um atributo derivado.

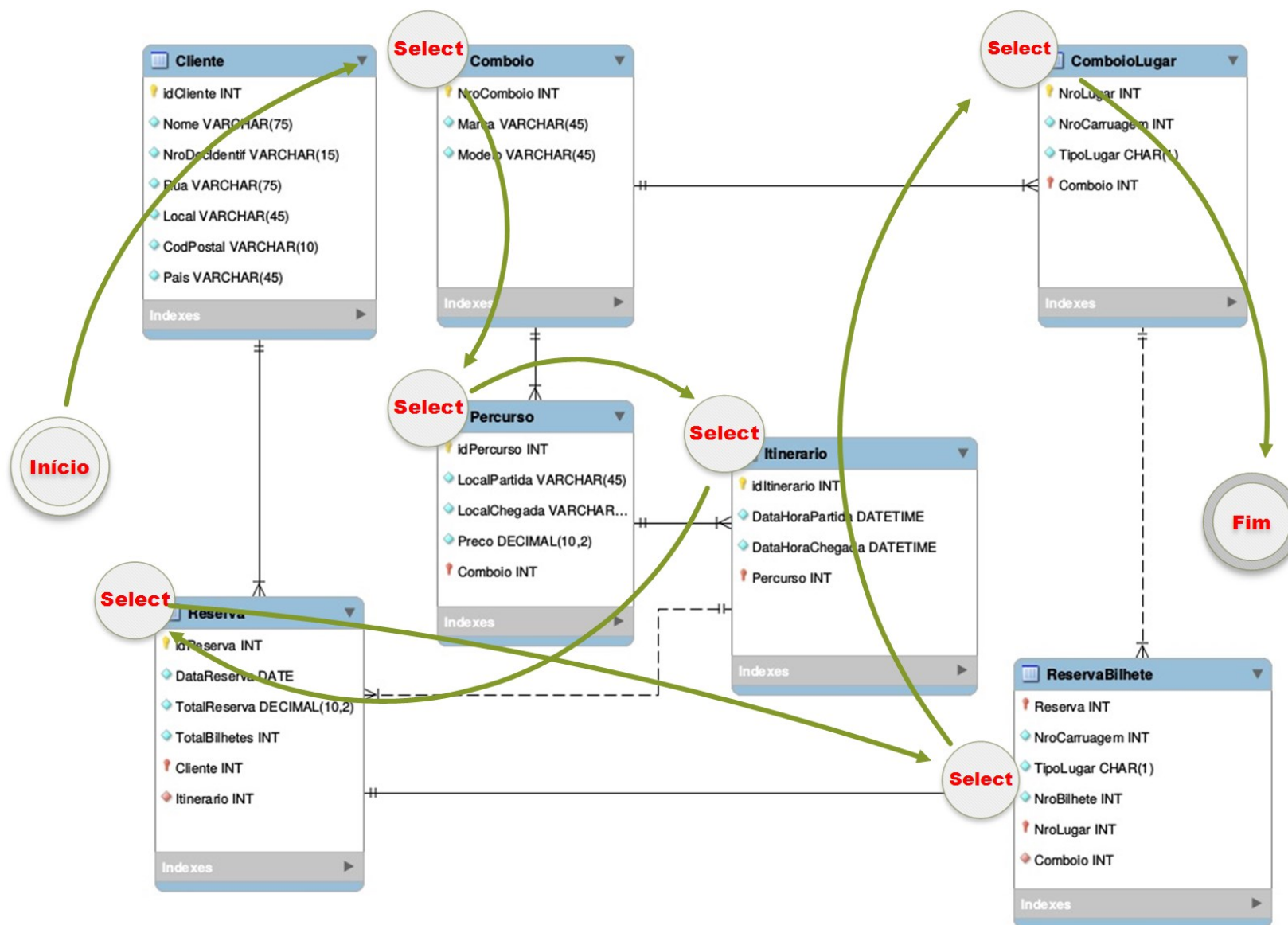
Figura 13 -
Mapa de
Transação -
Efetuar uma
reserva



3.3.3 Consultar os lugares livres de um dado itinerário

Para consultar o número de lugares livres de um dado itinerário, apenas é necessário fazer uma junção da relação *Reserva* com a relação *ReservaBilhete*, pela chave primária *idReserva* a chave estrangeira *Reserva* em *ReservaBilhete*, filtrando pelo valor da chave estrangeira *Itinerario* em *Reserva*. No entanto, é necessário obter o número de comboio com uma consulta, de forma análoga à secção anterior. Após estes passos é necessário efetuar uma junção à direita com a relação *ComboioLugar*. No entanto, como a junção à esquerda apenas funciona com relações com o mesmo grau, i.e., o mesmo número de atributos, é necessário fazer uma projeção com os atributos equivalentes à relação *ComboioLugar* do resultado da consulta anterior, para ficar com o mesmo grau. Em seguida, efetua-se a junção à direita pela chave estrangeira composta por *NroLugar* e *Comboio* da tabela resultante, com a chave primária composta com os atributos com o mesmo nome em *ComboioLugar*. Finalmente, é necessário filtrar os valores da chave primária composta que estarão com o valor a nulo numa junção deste tipo, i.e., os valores que não ocorrem na junção do lado esquerdo da consulta.

Figura 14 -
Mapa de
Transação –
Consultar
lugares livres



3.4. Verificação das Restrições de Integridade

Para garantir a consistência, a validade e atomicidade dos dados há que impor restrições de integridade sobre a mesma, com base nos requisitos analisados para o caso de estudo.

Para garantir que a base de dados contém apenas dados válidos, ou seja, exatos e consistentes (de acordo com os requisitos), são impostas restrições de integridade sobre a mesma. Assim, nesta secção são analisadas restrições de domínio, regras de integridade como integridade de entidade e referencial, restrições de multiplicidade e restrições gerais aos dados.

3.4.1 Integridade de Domínio

A integridade de domínio refere-se aos valores inseridos numa base de dados, i.e., que tipo de dados tem o atributo e que valores pode assumir dentro desse tipo de dados. Os valores permitidos foram avaliados aquando a análise do modelo concetual e podem ser consultados no anexo I.C de uma forma mais genérica, estando de forma explícita no anexo I.E, relativo ao modelo de dados lógico.

De igual modo, pode-se dizer que o facto de um atributo requerer ou não dados faz parte da integridade do domínio, i.e., se pode ou não ter valor nulos. Remete-se esta consulta para os anexos supracitados.

3.4.2 Integridade de Entidade

A integridade de entidade dita que nenhum valor de chave primária pode ser nulo, e o valor de uma chave primária é único. O SGBD (dependendo do motor) garante que uma relação não tenha dois registos com o mesmo valor e que a chave primária tenha um valor nulo. Esta verificação foi feita aquando a análise e validação do modelo concetual na secção 2.2.

3.4.3 Integridade Referencial

A integridade referencial permite garantir que todas as chaves estrangeiras são válidas. Uma chave estrangeira liga cada registo de uma relação filha à relação com a chave pai

candidata correspondente. Assim, integridade referencial significa se uma chave estrangeira contém um valor, então esse valor deve-se que referir a um registo existente numa relação pai. No entanto, existem duas questões que se deve colocar:

- Se uma chave estrangeira permite valores nulo, no caso de ocorrência opcionais ou obrigatórias;
- Como é assegurada a integridade referencial, tendo em conta especificações de existência em que uma chave candidata ou estrangeira são inseridas, atualizadas ou apagadas (definir CASCADE, NO ACTION, SET NULL, SET DEFAULT ou NO CHECK).

No caso do modelo de dados em estudo e, com base nos requisitos a ação definida NO ACTION, ou seja, não se poder apagar, nem se alterar em cascata.

3.4.4 Restrições de Multiplicidade

As restrições de multiplicidade foram decididas na secção 2.3, podendo ver o dicionário de dados correspondente no anexo I.B.

3.4.5 Restrições Gerais

De acordo com os requisitos, têm que se verificar as seguintes restrições organizacionais:

- Tipo de lugar tem que ser, ou da coxia ou junto à janela.
- O número máximo de bilhetes numa reserva é 4.

Não existem outras restrições organizacionais no modelo de dados.

3.5. Tamanho inicial e Crescimento Futuro

Nesta secção apresentará uma estimativa inicial do tamanho da base de dados e uma previsão do seu crescimento. A estimativa tem por base dados obtidos junto da empresa e os cálculos necessários para essa estimativa.

3.5.1 Tamanho Inicial

Acerca do tamanho inicial da base de dados, propõem-se o seguinte estado inicial:

500 clientes: Este valor é baseado num número de clientes do valor esperado de clientes pela a agência de viagens nos primeiros dois meses.

14 percursos: os percursos existentes são os definidos nos requisitos, sendo seis nacionais e oito internacionais.

10 comboios: os comboios existentes são os que a CP deu à concessão para os percursos nos requisitos. Cada comboio tem 200 lugares, em 4 carruagens (50 lugares por carruagem).

400 itinerários: cada percurso nacional é efetuado uma vez por dia, durante os sete dias da semana e cada percurso internacional é efetuado uma vez por semana, sendo que as viagens para Viena e Praga são feitas no mesmo dia. Ou seja, o total de itinerários por semana são 50, 200 por mês. Inicialmente, empresa espera ter itinerários estipulados para dois meses, i.e., 400 itinerários.

Os itinerários são os que se seguem:

- Braga-Porto: será efetuado diariamente nos seguintes horários:
 - Hora de partida 8:00:00; Hora de chegada 12:30:00;
- Porto-Braga: será efetuado diariamente no seguinte horário:
 - Hora de partida 17:00:00; Hora de chegada 20:00:00;
- Porto-Lisboa: será efetuado diariamente no seguinte horário:
 - Hora de partida 10:00:00; Hora de chegada 14:30:00;
- Lisboa-Porto: será efetuado diariamente no seguinte horário:
 - Hora de partida 15:00:00; Hora de chegada 20:30:00;
- Lisboa-Faro: será efetuado diariamente no seguinte horário:
 - Hora de partida 10:00:00; Hora de chegada 14:30:00;
- Faro-Lisboa: será efetuado diariamente no seguinte horário:

- Hora de partida 15:00:00; Hora de chegada 20:30:00;
- Lisboa-Madrid: será efetuado semanalmente no seguinte horário:
 - Hora de partida 6:00:00; Hora de chegada 22:00:00;
- Lisboa-Paris: será efetuado semanalmente no seguinte horário:
 - Hora de partida 05:00:00; Hora de chegada 02:00:00
(dois dias após);
- Lisboa-Londres: será efetuado semanalmente no seguinte horário:
 - Hora de partida 09:00:00; Hora de chegada 02:00:00 (do
dia seguinte);
- Lisboa-Berlim: será efetuado semanalmente no seguinte horário:
 - Hora de partida 09:00:00; Hora de chegada 01:00:00
(dois dias após);
- Lisboa-Budapeste: será efetuado semanalmente no seguinte horário:
 - Hora de partida 09:00:00; Hora de chegada 05:00:00
(dois dias após);
- Lisboa-Praga: será efetuado semanalmente no seguinte horário:
 - Hora de partida 00:09:00; Hora de chegada 01:00:00 (do
dia seguinte);
- Lisboa-Viena: será efetuado semanalmente no seguinte horário:
 - Hora de partida 00:09:00; Hora de chegada 01:00:00 (do
dia seguinte);

1200 reservas: a empresa espera que haja 1000 reservas (nº de lugares variável) nos dois primeiros meses.

3.5.2 Crescimento Futuro

A agência espera que o volume de negócios duplique em 4 anos. Pelo previsto pelo departamento financeiro da empresa, o aumento do volume de negócios é o seguinte:

Aumento de 50 clientes por mês: sendo esta uma agencia de pequena dimensão não se prevê um grande crescimento em relação ao numero de clientes.

Aumento de 100 reservas por mês: sendo que um cliente faz em média 0,1 reserva por mês (devido ao facto de existirem clientes que não efetuam reservas em certos meses), é previsto que se obtenha um aumento de 100 reservas por mês.

A empresa em conjunto com o seu departamento financeiro decidiu não obter mais concessões de percursos e respetivos comboios, uma vez planearam a estratégia da empresa com vista o turismo internacional e na qualidade de serviço (custos mais elevados), assim não esperando obter mais concessões da CP nos próximos 4 anos.

3.6. Validação de modelo lógico com utilizador

Terminada a fase de modelação lógica, o modelo foi revisto com os donos da agência de viagens, uma vez que, são eles os únicos utilizadores da base de dados. De acordo com a direção da agência, confirmou-se que garantia os requisitos.

4. Modelo físico

Neste capítulo está demonstrado todo o processo de implementação do modelo de dados físico, partido do modelo lógico antecedente.

4.1. Tradução e implementação do modelo lógico para um SGBD

4.1.1 Escolha do SGBD alvo para o modelo físico

Antes de fazer a implementação do modelo lógico num modelo físico, é necessário proceder à escolha de SGBD, tendo em conta os requisitos, tamanho da base de dados, suporte e sistema operativos suportados. Adicionalmente, é necessário verificar se o SGBD, suporta definição de chaves, definição de obrigatoriedade de dados (dados não nulos), definição de domínios, restrições de integridade relacional e definição restrições gerais.

Dado que a agencia quer uma solução de baixo custo, a melhor opção vai para o software livre. O sistema operativo onde o SGBD irá correr será um Windows 10, em conjunto com uma aplicação. Pretende-se também que o SGBD tenha suporte técnico e seja bem documentado, e que seja de fácil manutenção, além de suportar as funcionalidades acima descritas.

Por conseguinte, escolheu-se o *MySQL* uma vez que suporta todas as características acima mencionadas (corre em Windows, tem suporte técnico e é bem documentado, a Oracle é proprietária, o que dá continuidade de suporte, uma vez que é uma empresa sólida e é software livre).

No entanto, o *MySQL* também tem algumas limitações, mas adequa-se à dimensão da base de dados.

4.1.2 Relações Base

O primeiro passo para a tradução do modelo lógico para o modelo físico é o desenho das relações de base. Uma vez que, se implementou o modelo lógico no

MySQL Workbench, o processo de criação do esquema da base de dados foi automático, usando a ferramenta de *Forward Engineering*. O modelo lógico, nessa ferramenta já continha a informação sobre relações, domínio de atributos, chaves primárias e estrangeiras, e as relações do modelo lógico servem de suporte ao esquema físico. Esta informação pode ser vista com detalhe no dicionário de dados no anexo I.D e no anexo I.E. ² A definição de cada uma das relações do físico, consta em seguida:

A informação corresponde àquela no dicionário de dados, e a definição de cada uma das tabelas do físico consta em seguida.

Legenda

PK – Primary Key

NN – Not Null

UQ – Unique

B – Binary

UN – Unsigned

ZF – Zero-Filed

AI – Auto Incremental

G – Generated Column

Default/Expression – Valor por defeito

² Note-se que poder-se-ia usar uma linguagem de desenho de base de dados (ou DBDL – *DataBase Design Language*), para complementar os dados, no entanto decidiu-se que a informação em anexo como complemento é suficiente.

- *Cliente*

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idCliente	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Nome	VARCHAR(75)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
NroDocIdentif	VARCHAR(15)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Rua	VARCHAR(75)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Local	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
CodPostal	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Pais	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 15 - Tabela *Cliente*

- *Reserva*

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idReserva	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
DataReserva	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
TotalReserva	DECIMAL(10,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0.00
TotalBilhetes	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
Cliente	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Itinerario	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 16 - Tabela *Reserva*

- *Percurso*

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idPercurso	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
LocalPartida	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
LocalChegada	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Preco	DECIMAL(10,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Comboio	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 17 - Tabela *Percurso*

- *Itinerario*

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idItinerario	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
DataHoraPartida	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
DataHoraChegada	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Percurso	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 18 - Tabela *Itinerario*

- Comboio

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
NroComboio	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Marca	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 19 - Tabela *Comboio*

- *ComboioLugar*

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
NroLugar	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
NroCarruagem	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
TipoLugar	CHAR(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Comboio	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 20 - Tabela *ComboioLugar*

- *ReservaBilhete*

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Reserva	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
NroCarruagem	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
TipoLugar	CHAR(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
NroBilhete	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
NroLugar	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Comboio	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 21 - Tabela *ReservaBilhete*

4.1.3 Representação de Atributos Derivados

Os atributos derivados foram identificados no modelo concetual e no modelo lógico e podem ser consultados nos anexos I.C e I.E e correspondem ao total da reserva e ao total de bilhetes da reserva. O valor total de reserva será sempre calculado, aquando uma inserção com base no preço do percurso. O número total de bilhetes será inserido diretamente na reserva, por questões de eficiência, dado ser necessário um procedimento para cada número de bilhetes. Assim, controla-se o número de bilhetes que é inserido.

Relativamente atributos derivados identificados na tradução para o modelo físico, identificou-se o número de comboio, a relação *ReservaBilhete*. De forma a garantir que, o número de comboio que corresponde ao do itinerário, inclui-se a consulta desse valor na introdução de cada bilhete. Deste modo, garante-se que o número de comboio é consistente com o número de comboio associado ao itinerário.

4.1.4 Restrições Gerais

As restrições gerais do modelo de dados foram implementadas em verificações de condições da nos procedimentos da base de dados, apesar de haver a declaração CKECK, padrão do SQL. Infelizmente, esta instrução não é utilizada em alguns SGBD, em particular no *MySQL*, por razões de eficiência.

- **Número máximo de 4 bilhetes numa reserva.**

Por questões de eficiência, preferiu-se criar um procedimento para cada reserva, variando o número de parâmetros para efetuar uma reserva, e inserindo o valor total de bilhetes aquando a execução do procedimento, como mencionado na secção anterior.

- **Tipo de lugar 'J' ou 'C'**

Para verificar esta restrição, deve-se incluir uma condição na transação. Do seguinte modo:

```
-- Verifica restrição dos lugares  
IF tipo_lugar NOT IN ('J', 'C') THEN ROLLBACK; END IF;
```

Fragmento 1 - Restrição geral do tipo dos lugares

4.2. Escolha de Índices

Por defeito, no *MySQL Workbench*, os índices de chave primária e de chave estrangeira são automáticos. Uma das razões para este facto, é a utilização frequente de junções entre tabelas, pelas chaves primárias e chaves estrangeiras, justificando assim a sua aplicação automática. Por outro lado, a chave primária necessita de um índice único, uma vez que é o identificador de uma relação, e a procura pelo identificador é justificação mais que suficiente.

Para além dos índices de chave primária e chave estrangeira, foram adicionados índices únicos para os valores do documento de identificação do cliente e, para o número de bilhete. A existência do primeiro índice sobre o nº de documento de identificação, justifica-se pelo requisito em que o utilizador faz pesquisa na relação *Cliente*, pelo documento de identificação, e índice único porque nº de documento de identificação é uma chave candidata. O segundo índice aplicado ao número de bilhete tem uma explicação análoga ao anterior.

Adicionalmente, aplicaram-se mais três índices secundários: um em *DataHoraPartida*, outro em *DataHoraChegada* e outro em *DataReserva*. Os dois atributos estão na tabela *Itinerario* e o terceiro está na relação *Reserva*. Justifica-se a sua existência, uma vez que são muito frequentes as consultas por intervalos de datas e, estão em relações com elevada cardinalidade. Salienta-se que estes três índices não são únicos.

Note-se que, segundo as boas práticas, atributos onde sejam necessários filtros nas consultas com a cláusula *WHERE* devem ter um índice, i.e., todos os índices mencionados. No entanto deve-se analisar o *overhead* e se se justifica esse *overhead*. Daí que nem todos os atributos têm índice. Por exemplo atributos que são muito

utilizados, *LocalPartida* e *LocalChegada* na relação *Percurso* não justificam o uso de um índice, uma vez que são poucos os percursos.

Por último, a estrutura dos índices escolhida é que está por defeito no motor *InnoDB*, i.e., árvores binárias. Note-se que, poder-se-ia ter escolhido o motor *Memory* em que tínhamos mais uma estrutura para escolha (tabelas de *hash*), mas desconhece-se se este motor garante restrições de integridade.

Os índices foram implementados diretamente no *script* de criação.

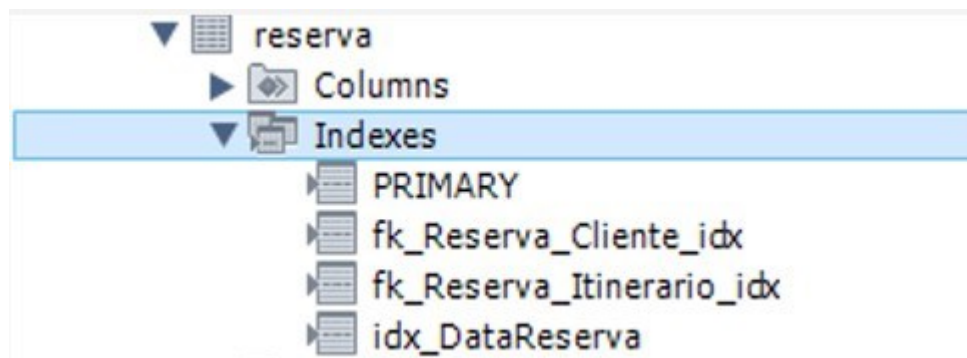


Figura 22 – Índice em DataReserva

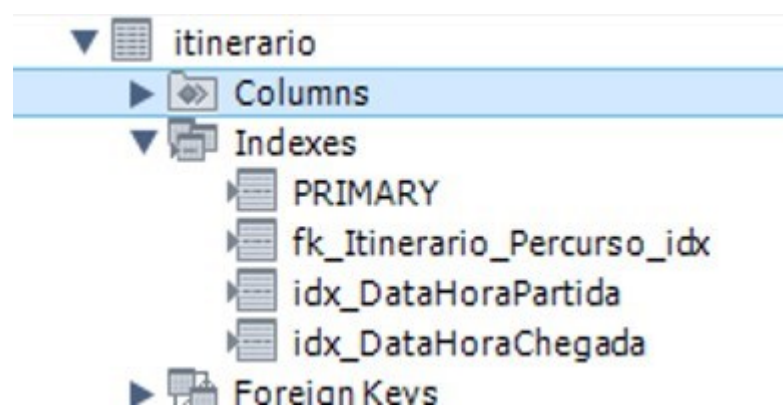


Figura 23 – Índices secundários em Itinerario

4.3. Análise de Transações

Nesta secção analisa-se de que forma as transações analisadas no na validação do modelo conceptual e lógico foram implementadas usando SQL.

4.3.1 Obter dados para documento comprovativo de compra

Para obter os dados para um comprovativo de compra ou fatura, criou-se o procedimento que se encontra a seguir:

```
CREATE PROCEDURE `sp_obter_comprovativo` (IN utilizador_id INT, IN
reserva_nro INT)
BEGIN
SELECT
    cl.*, rs.*,
    it.DataHoraPartida, it.DataHoraChegada,
    pc.LocalPartida, pc.LocalChegada, rb.*
FROM
    Cliente AS cl
    JOIN
    Reserva AS rs ON cl.idCliente = rs.Cliente
    JOIN
    Itinerario AS it ON rs.Itinerario = it.idItinerario
    JOIN
    Percurso AS pc ON it.Percurso = pc.idPercurso
    JOIN
    Comboio AS cb ON pc.Comboio = cb.NroComboio
    JOIN
    ReservaBilhete AS rb ON rs.idReserva = rb.Reserva
WHERE
    (cl.idCliente = utilizador_id)
    AND (rs.idReserva = reserva_nro);
END
```

Fragmento 2 – Procedimento para obter comprovativo

4.3.2 Efetuar uma reserva de um ou vários lugares

Para efetuar uma reserva, implementou-se o procedimento que se encontra a seguir. Note-se que, este é um exemplo dos 4 procedimentos, sendo que variam em número de parâmetros, têm mais condições de verificação de tipos de lugar e tem mais tuplos na clausula VALUES. De resto a explicação é análoga.

O fragmento que se encontra a seguir corresponde à assinatura do procedimento para reserva de um bilhete.

```

CREATE PROCEDURE sp_efetuar_reserva_1_bilhete
    (IN nro_cliente INT,
    IN data_reserva DATE,
    IN nro_itinerario INT,
    IN nro_lugar INT,
    IN nro_carr INT,
    IN tipo_lugar CHAR(1),
    IN nro_bilhete INT)

```

Fragmento 3 – Assinatura do procedimento sp_efetuar_reserva_1_bilhete

Dado que o código é bastante extenso far-se-á uma análise por partes. Em primeiro lugar, no fragmento de código que se encontra a seguir, está a declaração de variáveis para uso na transação, bem como a declaração de um *handler* para controlar exceções na transação.

```

DECLARE PrecoVenda DECIMAL(10, 2);
DECLARE id_reserva INT;
DECLARE nro_comboio INT;
DECLARE ErroTransacao BOOL DEFAULT 0;
DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET ErroTransacao = 1;

```

Fragmento 4 – Declaração de variáveis para procedimento da reserva

Depois é necessário obter o preço de percurso e o comboio associado ao itinerário.

```
SELECT
    Preço, NroComboio INTO PreçoVenda, nro_comboio
FROM
    Comboio
    INNER JOIN
    Percurso ON Comboio.NroComboio = Percurso.Comboio
    INNER JOIN
    Itinerario ON Itinerario.Percurso = Percurso.idPercurso
WHERE
    Itinerario.idItinerario = nro_itinerario;
```

Fragmento 5 - Consulta do preço e do nº de comboio

Em seguida testa-se se os valores dos lugares estão dentro da restrição.

Fragmento 6 – Teste da restrição geral

```
-- Verifica restrição dos lugares
IF tipo_lugar NOT IN ('J', 'C') THEN ROLLBACK; END IF;
```

Depois são inseridos os valores para a reserva. Note-se que, o total de bilhetes é 1, e o valor de total de reserva não é inserido.

```
-- Insere reserva
INSERT INTO Reserva
    (DataReserva, TotalBilhetes, Cliente, Itinerario)
VALUES (data_reserva, 1, nro_cliente, nro_itinerario);
```

Fragmento 7 – Inserção de dados em reserva

Seleciona-se o último identificador, i.e., o maior identificador uma vez que o atributo é auto-incrementável.

```
-- Seleciona o último id de reserva
SELECT max(Reserva.idReserva) INTO id_reserva FROM Reserva;
```

Fragmento 8 – Seleção do último identificador de reserva

Em seguida, inserem-se os valores do bilhete, incluindo o número de reserva obtido anteriormente.

```
-- Insere bilhete
INSERT INTO ReservaBilhete
(ReservaBilhete.Reserva, ReservaBilhete.NroCarruagem,
ReservaBilhete.TipoLugar, ReservaBilhete.NroBilhete,
ReservaBilhete.NroLugar, ReservaBilhete.Comboio)
VALUES(id_reserva, nro_carr, tipo_lugar, nro_bilhete, nro_lugar,
nro_comboio);
```

Fragmento 9 – Inserção de dados na reserva

Após a inserção dos dados, o valor do total de venda é atualizado. Por uma questão de completude, o valor de preço de venda é multiplicado por um para demonstrar de forma análogo o que sucede com os outros procedimentos de reserva.

```
-- Atualiza reserva
UPDATE Reserva
SET Reserva.TotalReserva = PrecoVenda*1
WHERE idReserva = id_reserva;
```

Fragmento 10 – Atualização do preço de venda da reserva

Por fim, o valor da variável do *handler* é testada. Caso tenha ocorrido alguma exceção, as operações são revertidas. Caso contrário, são realizadas.

```
-- Verificação da ocorrência de um erro.
IF ErroTransacao THEN
    -- Desfazer as operações realizadas.
    ROLLBACK;
ELSE
    -- Confirmar as operações realizadas.
    COMMIT;
END IF;
```

Fragmento 11 – Teste à variável do *handler*

4.3.3 Consultar os lugares livres de um dado itinerário

Para efetuar esta consulta criou-se o seguinte procedimento:

```
CREATE PROCEDURE `sp_consultar_lugares_livres` (IN  
itinerario_nro int)
```

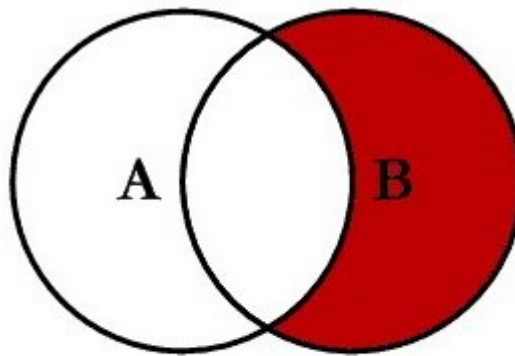
Fragmento 12 Assinatura do procedimento de consulta de lugares livres

Antes de proceder com o resto da consulta, é necessário obter o valor do comboio associado ao itinerário, para posteriormente filtrar os resultados por este número de comboio.

```
-- Obtem número do comboio correspondente ao itinerario  
SELECT  
  NroComboio  
INTO nro_comboio FROM  
  Comboio  
    INNER JOIN  
  Percurso ON Comboio.NroComboio = Percurso.Comboio  
    INNER JOIN  
  Itinerario ON Itinerario.Percurso = Percurso.idPercurso  
WHERE  
  Itinerario.idItinerario = itinerario_nro;
```

Fragmento 13 – Primeira consulta pelo número do comboio

Pretende-se obter a diferença entre o conjunto da direita (lugares disponíveis) e os da esquerda (lugares ocupados). No entanto, para realizar o resto da consulta teve-se em conta o diagrama que está a seguir, e forma como os OUTER JOIN's funcionam, i.e., o grau em cada um dos conjuntos deve ser igual, e no conjunto com menor numero de valores da junção aparecem nulos.



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```

Figura 24 – RIGHT JOIN

Deste modo, pretende-se obter os valores dos bilhetes reservados, projetando os atributos de *ReservaBilhete* equivalentes em *ComboioLugar*

```
SELECT
    ReservaBilhete.NroLugar,
    ReservaBilhete.NroCarruagem,
    ReservaBilhete.TipoLugar,
    ReservaBilhete.Comboio
FROM
    Reserva
INNER JOIN ReservaBilhete ON Reserva.idReserva =
    ReservaBilhete.Reserva
WHERE
    Reserva.Itinerario = itinerario_nro
```

Fragmento 14 – Segunda consulta para obter os lugares ocupados do itinerário

Em seguida inclui-se esta consulta numa outra maior, onde seria feita a junção à direita da tabela resultante (Tabela T), com a tabela *ComboioLugar*, filtrando os nulos da tabela T, pelo valor da sua chave composta e pelo valor do número de comboio respetivo. No fim, é efetuada a projeção para obter os lugares livres daquele itinerário.

```
-- Obtem os lugares livres do comboio correspondente ao itinerario
SELECT
    ComboioLugar.NroLugar,
    ComboioLugar.NroCarruagem,
    ComboioLugar.TipoLugar,
    ComboioLugar.Comboio
FROM
    (
        SELECT
            ReservaBilhete.NroLugar,
            ReservaBilhete.NroCarruagem,
            ReservaBilhete.TipoLugar,
            ReservaBilhete.Comboio
        FROM
            Reserva
        INNER JOIN ReservaBilhete ON Reserva.idReserva =
ReservaBilhete.Reserva
        WHERE
            Reserva.Itinerario = itinerario_nro
        ) AS T
    RIGHT JOIN
        ComboioLugar ON T.NroLugar = ComboioLugar.NroLugar
        AND T.Comboio = ComboioLugar.Comboio
WHERE
    ComboioLugar.Comboio = nro_comboio
    AND T.NroLugar IS NULL
    AND T.Comboio IS NULL;
```

Fragmento 15 – Consulta completa com RIGHT JOIN

4.4. Estimativa dos Requisitos de Espaço em Disco

4.4.1 Povoamento e tamanho inicial

Para perceber melhor o tamanho do espaço inicial em disco, inseriram-se dados que refletissem o indicado na secção 3.5.1. Assim, a visão final do espaço em disco é o mais realista possível. Após o povoamento, para conhecer o tamanho das tabelas, executou-se a consulta em SQL que se encontra a seguir:

```
SELECT TABLE_NAME, table_rows, data_length / 1024 'Dados (KB)',  
index_length / 1024 'Indices (KB)',  
round(((data_length + index_length) / 1024 / 1024),5) 'Size in MB'  
FROM information_schema.TABLES WHERE table_schema = 'Agencia';
```

Obtendo como resultado a seguinte tabela:

TABLE_NAME	table_rows	Dados (KB)	Indices (KB)	Size in MB
cliente	500	80.0000	16.0000	0.09375
comboio	10	16.0000	0.0000	0.01563
comboiolugar	2000	112.0000	64.0000	0.17188
itinerario	434	48.0000	48.0000	0.09375
percurso	14	16.0000	16.0000	0.03125
reserva	1100	80.0000	112.0000	0.18750
reservabilhete	1508	96.0000	112.0000	0.20313

Figura 25 – Nº de registo e tamanho inicial das tabelas

De seguida foi efetuada a seguinte consulta para saber o espaço total ocupado pela base de dados:

```
SELECT table_schema 'Agencia', SUM(data_length + index_length) / 1024 /  
1024 'DB Size in MB'  
FROM information_schema.tables  
WHERE table_schema = 'Agencia';
```

Tendo sido obtido o seguinte resultado:

Agencia	DB Size in MB
agencia	0.79687500

Figura 26 – Tamanho total em disco

Podendo-se concluir que a base de dados necessita de cerca de 1 MB inicialmente.

4.4.2 Crescimento Futuro

Assim como foi referido na secção 3.5, as requisições e reservas serão o principal fator de aumento do espaço em disco da base de dados. Por mês é previsto um aumento de cerca de 25 reservas. Ao fim de um ano há $25 \times 365 = 9125$ reservas. Há ainda um registo de em média $1518/1100 = 1.38$ novos bilhetes por cliente, resultando, portanto, num aumento de $50 \times 1.38 = 69$ bilhetes por mês $69 \times 12 = 828$ bilhetes por ano. Por ano há ainda cerca de $50 \times 12 = 600$ novos clientes. Usando os dados da figura 25 pode-se então prever o aumento aproximado da base de dados ao fim de um ano:

$$9125 \times 0.18750 + 600 \times 0.09375 + 828 \times 0.20313 = 1935,3791 \text{KB} = 1.9 \text{ MB}$$

4.5. Definição das Vistas de Utilização e Regras de Acesso

Aquando a elaboração do modelo concetual em conjunto com a análise dos requisitos, não se identificaram vistas de utilização de utilizador para além dos proprietários da agência de viagens, i.e., o modelo de dados era monovista. No entanto, existem operações que devem ser geridas para evitar acesso indevido, e corrupção dos dados e, possivelmente, destruir a base de dados. Assim existe o superutilizador, ou administrador da base de dados, e os utilizadores da base de dados.

Os utilizadores da base de dados têm um acesso limitado à base de dados. Uma vez que uma reserva não pode ser apagada, nem alterada no sistema, as operações de atualização e remoção estão restringidas para estes utilizadores. De igual modo, pretende-se um histórico das reservas, o que implica que todos os dados não possam ser removidos e apenas os dados do cliente podem ser atualizados. A restrição de atualização dos dados de reserva tem a ver com o que já foi mencionado. Como a reserva tem uma lista de bilhetes, os dados em *ReservaBilhete* também não podem ser removidos e atualizados. Adicionalmente, os comboios, os percursos e itinerário

existentes não podem ser atualizados pelos utilizadores, porque os dados daqueles são geridos por uma aplicação que sincroniza com o sistema da CP. Na tabela abaixo estão especificadas as permissões de cada utilizador:

Tabela \ Perfil de Utilizador	Utilizador			
	I	U	D	S
Cliente	X	X		X
Reserva	X			X
Percurso				X
Comboio				X
Itinerario				X
ReservaBilhete	X			X
ComboioLugar				X

Tabela 1 - Permissões de utilizador

Em seguida demonstra-se como se procedeu para a criação de permissões de utilizador na relação *Cliente*.

```
CREATE USER 'agencia.cliente'@'localhost'
IDENTIFIED BY 'Agencia';
/*Concedem-se os privilégios de inserção, atualização
e seleção:*/
GRANT INSERT,UPDATE,SELECT
ON Agencia.Cliente
TO 'agencia.cliente'@'localhost';
/*E retira-se o privilégio de remoção na tabela
Cliente :*/
REVOKE DELETE
ON Agencia.Cliente
FROM 'agencia.cliente'@'localhost';
```

Fragmento 16 – Criação de utilizador e permissões na tabela *Cliente*

5. Conclusões e Trabalho Futuro

Em conclusão, o balanço dos objetivos é positivo, onde todos os passos de uma metodologia canónica do desenho de uma base de dados foram seguidos, desde da recolha de requisitos, passando pelo modelo concetual, modelo lógico e implementação de um modelo físico num SGBD.

A etapa inicial, baseada no levantamento dos requisitos junto da empresa foi bastante importante, visto que é a base para a construção dos consequentes modelos. físico. No entanto, houve várias reavaliações dos requisitos e vários modelos de dados concetuais, onde houve certa dificuldade na representação de lugares e outros atributos. No modelo lógico, a maioria das etapas decorreram como previsto, todavia os mapas e respetivas transações tiveram que ser reavaliados, aquando a documentação do modelo físico, por causa da decisão de tornar o atributo *Comboio* na tabela *ReservaLugar* como derivado. Note-se que, esta decisão foi tomada no modelo físico, na etapa da secção 4.1.3. De igual modo, foi necessário otimizar o modelo lógico com mais alguns relacionamentos, que possivelmente poderiam estar melhor representados no modelo concetual. No modelo físico, houve algumas dificuldades relativamente à implementação de procedimentos mais complexos, no entanto, estão em conformidade. Relativamente, às permissões de utilizador, poderiam ter sido melhor desenhadas. As permissões deviam ser refinadas ao nível dos procedimentos e não das tabelas. Adicionalmente, tiveram que ser efetuadas algumas otimizações na transação de reserva, nomeadamente a introdução direta de um atributo que, concetualmente, era um atributo derivado, nomeadamente, o total de bilhetes numa reserva.

Referências

Connolly, T. M. & Begg, C. E., 2015. *Database Systems - A Pratical Approach to Design, Implementation, and Design*. Sixth Edition / Global Edition ed. Harlow: Pearson Education Limited.

DuBois, P., 2007. *MySQL Cookbook*. Second Edition ed. s.l.:O'Reilly Media, Inc.

Garcia-Molina, H., Ullman, J. D. & Widom, J., 2009. *Database Systems - The Complete Book*. Second Edition ed. s.l.:Pearson Education, Inc.

Kofler, M., 2005. *The Definitive Guide to MySQL 5*. Third Edition ed. s.l.:Apress.

Teorey, T., Lightstone, S. & Nadeau, T., 2006. *Database Modeling & Design*. Fourth Edition ed. s.l.:Elsevier Inc..

Lista de Siglas e Acrónimos

BD	Base de Dados
CP	Comboios de Portugal
SGBD	Sistema de Gestão de Base de Dados
SQL	Structured Query Language
DBDL	DataBase Design Language

Anexos

A. Dicionário de dados - Entidades

ENTIDADE	DESCRIÇÃO	ALIAS	OCORRÊNCIA
Cliente	Obra disponível na BGUM para requisição e/ou consulta.	Comprador	Cada cliente efetua uma reserva.
Reserva	Identifica uma reserva de um conjunto de bilhetes.	Compra; Aquisição	Ocorre sempre que um cliente faz uma reserva de um conjunto de bilhetes. Uma reserva pode ser feita por um cliente, mas pode haver muitas reservas de um cliente.
Comboio	Veículo que faz a viagem.	Trem	Um comboio descreve o meio de transporte, e tem um ou mais percursos associados.
Percurso	Par chegada/destino que identifica, localmente, o início e fim de uma viagem e o seu custo	Trajetos; Caminho; Rota.	Um percurso é feito por um comboio, tem um local de partida e um local de chegada e o seu custo. O percurso é a componente local de uma deslocação. O percurso de A para B e o percurso de B para A são percursos diferentes.
Itinerário	Par chegada/destino que identifica, temporalmente, o início e fim de uma viagem, como também o dia da sua realização.	Viagem; Horário	Um itinerário tem um percurso associado e descreve a componente temporal de uma deslocação, ou seja, a data e hora que começa e, a data e a hora que acaba.

Tabela 2 – Dicionário de dados – Entidades

B. Dicionário de Dados – Relacionamentos

ENTIDADE	MULTIPLICIDADE	RELACIONAMENTO	MULTIPLICIDADE	ENTIDADE (RELACIONADA)
Cliente	1 (obrigatório)	Efetua; Faz.	N (obrigatório)	Reserva
Reserva	N (opcional)	Tem; Possui; Referencia.	1 (obrigatório)	Itinerário
Itinerário	N (obrigatório)	Tem; Possui; Faz.	1 (obrigatório)	Percurso
Comboio	1 (obrigatório)	Está; Faz; Percorre.	N (obrigatório)	Percurso

Tabela 3 – Dicionário de dados Relacionamentos

C. Dicionário de Dados – Atributos

NOME DA ENTIDADE	ATRIBUTO	DESCRIÇÃO	TIPO DE DADOS & COMPRIMENTO/DOMÍNIO	NULO	COMPOSTO	MULTIVALOR	DERIVADO	SIMPLES	VALOR POR DEFEITO
Cliente	idCliente	Identificador único de um cliente	Número natural	Não	Não	Não	Não	Sim	Auto incrementado
	Nome	Nome do cliente	String de 75 caracteres tamanho variável	Não	Não	Não	Não	Sim	-
	nroDocIdentif	Número de documento de identificação do cliente	String de 15 caracteres tamanho variável	Não	Não	Não	Não	Sim	-
	Morada			-	Sim	Não	-	Não	-
		Rua	Rua do Cliente	Não	Não	Não		Sim	
		Local	Localidade do cliente	Não	Não	Não	Não	Sim	
		CodPostal	Código postal do cliente	Não	Não	Não	Não	Sim	-
		País	País onde reside o cliente	Não	Não	Não	Não	Sim	-

Tabela 4 – Atributos de Cliente

NOME DA ENTIDADE	ATRIBUTO	DESCRIÇÃO	TIPO DE DADOS & COMPRIMENTO/DOMÍNIO	NULO	COMPOSTO	MULTIVALOR	DERIVADO	SIMPLES	VALOR POR DEFEITO
Reserva	idReserva	Identifica unicamente uma reserva	Número natural	Não	Não	Não	Sim	Sim	Auto incrementado
	TotalBilhetes	Número total de bilhetes da reserva	Número natural	Não	Não	Não	Sim	Não	0
	TotalReserva	Número do total do valor dos bilhetes da reserva	Número não-negativo de valor decimal (duas casas decimais)	Não	Não	Não	Sim	Não	0.00
	DataReserva	Data da reserva	Data (aaaa/mm/dd)	Não	Não	Não		Sim	-
	Lugar			-	Sim	Não	-	Não	-
	NroLugar	Indica o nº do lugar reservado	Número natural	Não	Não	Não	Não	Sim	-
	NroCarruagem	Indica o nº da carruagem reservada	Número natural	Não	Não	Não	Não	Sim	-
	TipoLugar	Indica o nº do lugar reservado	1 caractere/ J ou C	Não	Não	Não	Não	Sim	-
	NroBilhete	Indica o nº do bilhete reservado	Número natural	Não	Não	Não	Não	Sim	

Tabela 5 – Atributos de Reserva

NOME DA ENTIDADE	ATRIBUTO	DESCRIÇÃO	TIPO DE DADOS & COMPRIMENTO/DOMÍNIO	NULO	COMPOSTO	MULTIVALOR	DERIVADO	SIMPLES	VALOR POR DEFEITO
Comboio	NroComboio	Identificador único do comboio	Número natural	Não	Não	Não	Não	Sim	-
	Marca	Descreve a marca do comboio.	<i>String</i> de 45 caracteres tamanho variável	Não	Não	Não	Não	Sim	-
	Modelo	Descreve o modelo do comboio.	<i>String</i> de 45 caracteres tamanho variável	Não	Não	Não	Não	Sim	-
	Lugar			-	Sim	Sim	-	Não	-
	NroLugar	Indica o nº do lugar reservado	Número natural	Não	Não	Não	Não	Sim	
	NroCarruagem	Indica o nº da carruagem reservada	Número natural	Não	Não	Não	Não	Sim	-
	TipoLugar	Indica o nº do lugar reservado	1 caractere/ J ou C	Não	Não	Não	Não	Sim	-

Tabela 6 – Atributos de Comboio

NOME DA ENTIDADE	ATRIBUTO	DESCRIÇÃO	TIPO DE DADOS & COMPRIMENTO/DOMÍNIO	NULO	COMPOSTO	MULTIVALOR	DERIVADO	SIMPLES	VALOR POR DEFEITO
Percurso	idPercurso	Identificador único percurso	Número natural	Não	Não	Não	Não	Sim	Auto incrementado
	LocalPartida	Identifica local de partida de um percurso	<i>String</i> de 45 caracteres tamanho variável	Não	Não	Não	Não	Sim	-
	LocalChegada	Identifica local de chegada de um percurso	<i>String</i> de 45 caracteres tamanho variável	Não	Não	Não	Não	Sim	-
	Preco	Indica o custo de um percurso.	Número não negativo de valor de decimal (duas casas decimais)	Não	Não	Não	Não	Sim	-

Tabela 7 – Atributos de Percurso

NOME DA ENTIDADE	ATRIBUTO	DESCRIÇÃO	TIPO DE DADOS & COMPRIMENTO/DOMÍNIO	NULO	COMPOSTO	MULTIVALOR	DERIVADO	SIMPLES	VALOR POR DEFEITO
Itinerário	idItinerario	Identificador único de um itinerário	Número natural	Não	Não	Não	Não	Sim	Auto incrementado
	HoraPartida	Hora de partida de um itinerário	Tempo (hh:mm)	Não	Não	Não	Não	Sim	-
	HoraChegada	Hora de chegada de um itinerário	Tempo (hh:mm)	Não	Não	Não	Não	Sim	-
	DataPartida	Data da realização da partida do itinerário	Data (aaaa/mm/dd)	Não	Não	Não	Não	Sim	-
	DataChegada	Data da realização da chegada do itinerário	Data (aaaa/mm/dd)	Não	Não	Não	Não	Sim	-

Tabela 8 – Atributos de Itinerário

D. Dicionário de Dados-Tabelas

TABELA	MULTIPLICIDADE	RELACIONAMENTO	MULTIPLICIDADE	TABELA (RELACIONADA)
Cliente	1 (obrigatório)	Efetua; Faz.	N (obrigatório)	Reserva
Reserva	N (opcional)	Tem; Possui; Referencia.	1 (obrigatório)	Itinerário
Itinerario	N (obrigatório)	Tem; Possui; Faz.	1 (obrigatório)	Percurso
Comboio	1(obrigatório)	Está; Faz; Percorre.	N (obrigatório)	Percurso
ComboioLugar	1(obrigatório)	Pertence	N (obrigatório)	Comboio
ReservaBilhete	1(obrigatório)	Pertence	N (obrigatório)	Reserva
ComboioLugar	1(obrigatório)	Está em; Existe.	N (obrigatório)	ReservaBilhete

Tabela 9 – Tabela com Relacionamentos – modelo lógico

E. Dicionário De Dados-Atributos das Tabelas

NOME DA TABELA	ATRIBUTO	DESCRIÇÃO	TIPO DE DADOS & COMPRIMENTO/DOMÍNIO	NULO	CHAVE (OU ÚNICO)	INDEX	VALOR POR DEFEITO
Cliente	idCliente	Identificador único de um cliente	INT	Não	PK	Sim	Auto incrementado
	Nome	Nome do cliente	VARCHAR (75)	Não	-	Não	-
	NroDocIdentif	Número de documento de identificação do cliente	VARCHAR (15)	Não	UQ	Sim	-
	Rua	Rua do Cliente	VARCHAR (75)	Não	-	Não	-
	Local	Localidade do cliente	VARCHAR (45)	Não	-	Não	-
	CodPostal	Código postal do cliente	VARCHAR (10)	Não	-	Não	-
	Pais	País onde reside o cliente	VARCHAR (45)	Não	-	Não	-

Tabela 10 – Tabela Cliente

NOME DA TABELA	ATRIBUTO	DESCRIÇÃO	TIPO DE DADOS & COMPRIMENTO/DOMÍNIO	NULO	CHAVE (OU ÚNICO)	INDEX	VALOR POR DEFEITO
Reserva	idReserva	Identifica unicamente uma reserva	INT	Não	PK	Sim	Auto incrementado
	DataReserva	Data da reserva	DATE	Não	-	Sim	-
	TotalBilhetes	Indica o nº do lugar reservado	INT	Não	-	Não	0
	TotalReserva	Indica o nº da carruagem	INT	Não	-	Não	0.00
	Cliente	Indica o número do cliente que fez a reserva	INT	Não	FK	Sim	-

Tabela 11 – Tabela Reserva

NOME DA TABELA	ATRIBUTO	DESCRIÇÃO	TIPO DE DADOS & COMPRIMENTO/DOMÍNIO	NULO	CHAVE (OU ÚNICO)	INDEX	VALOR POR DEFEITO
Comboio	NroComboio	Identificador único do comboio	INT	Não	PK	Sim	-
	Marca	Descreve a marca do comboio.	VARCHAR (45)	Não	-	Não	-
	Modelo	Descreve o modelo do comboio.	VARCHAR (45)	Não	-	Não	-

Tabela 12 – Tabela Comboio

NOME DA TABELA	ATRIBUTO	DESCRIÇÃO	TIPO DE DADOS & COMPRIMENTO/DOMÍNIO	NULO	CHAVE (OU ÚNICO)	INDEX	VALOR POR DEFEITO
Percurso	idPercurso	Identificador único percurso	INT	Não	PK	Sim	Auto incrementado
	LocalPartida	Identifica local de partida de um percurso	VARCHAR (45)	Não	-	Não	-
	LocalChegada	Identifica local de chegada de um percurso	VARCHAR (45)	Não	-	Não	-
	Preco	Indica o custo de um percurso.	DECIMAL (10, 2)	Não	-	Não	-
	Comboio	Indica o número do comboio	INT	Não	FK	Sim	

Tabela 13 – Tabela Percurso

NOME DA TABELA	ATRIBUTO	DESCRIÇÃO	TIPO DE DADOS & COMPRIMENTO/DOMÍNIO	NULO	CHAVE (OU ÚNICO)	INDEX	VALOR POR DEFEITO
Itinerario	idItinerario	Identificador único de um itinerário	INT	Não	PK	Sim	Auto incrementado
	DataHoraPartida	Data e hora de partida de um itinerário	DATETIME	Não	-	Sim	-
	DataHoraChegada	Data e hora de chegada de um itinerário	DATETIME	Não	-	Sim	-
	Percurso	Indica o nº do percurso	INT	Não	FK	Sim	-

Tabela 14 – Tabela Itinerario

NOME DA TABELA	ATRIBUTO	DESCRIÇÃO	TIPO DE DADOS & COMPRIMENTO/DOMÍNIO	NULO	CHAVE (OU ÚNICO)	INDEX	VALOR POR DEFEITO
ComboioLugar	NroComboio	Indica o nº do comboio	INT	NÃO	PK/FK	Sim	
	NroLugar	Indica o nº do lugar disponível no comboio	INT	Não	PK	Não	
	NroCarruagem	Indica o nº da carruagem do comboio	INT	Não	-	Não	-
	TipoLugar	Indica o tipo do lugar do comboio	CHAR (1) / 'J' ou 'C'	Não	-	Não	-

Tabela 15 – Tabela ComboioLugar

NOME DA TABELA	ATRIBUTO	DESCRIÇÃO	TIPO DE DADOS & COMPRIMENTO/DOMÍNIO	NULO	CHAVE (OU ÚNICO)	INDEX ³	VALOR POR DEFEITO
ReservaBilhete	NroComboio ⁴	Indica o nº do comboio	INT	Não	FK	Sim	-
	NroLugar	Indica o nº do lugar disponível no comboio	INT	Não	PK/FK	Sim	-
	NroCarruagem	Indica o nº da carruagem do comboio	INT	Não	-	Não	-
	TipoLugar	Indica o tipo do lugar do comboio	CHAR (1) / 'J' ou 'C'	Não	-	Não	-
	Reserva	Identifica uma reserva	INT	Não	PK/FK	Sim	-
	NroBilhete	Identifica um bilhete	INT	Não	UQ	Sim	-

Tabela 16 – Tabela ReservaBilhete

³ Alguns índices são sobre dois atributos, nomeadamente NroComboio e NroLugar (chave estrangeira)

⁴ Atributo derivado no modelo físico