



Universidade do Minho

Mestrado Integrado em Engenharia Informática
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Letivo de 2016/2017

Projeto de um Sistema de Bases de Dados relacional

Bruno Miguel Salgado Machado (A74941)

Fábio Luís Baião da Silva (A75662)

João Rui de Sousa Miguel (A74237)

Luís Manuel Leite Costa (A74819)

Novembro 2016

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

Projeto de um Sistema de Bases de Dados relacional

Bruno Miguel Salgado Machado (A74941)
Fábio Luís Baião da Silva (A75662)
João Rui de Sousa Miguel (A74237)
Luís Manuel Leite Costa (A74819)

Novembro 2016

Resumo

Este projeto tem como objetivo a criação de uma Base de Dados relacional sustentável, para armazenar e gerir de forma adequada a informação, relacionada com uma empresa de viagens ferroviárias.

O método para a criação da Base de Dados irá ser composto por quatro fases: o levantamento de requisitos, o desenho do modelo conceptual, seguido do lógico e finalmente o modelo físico. O levantamento de requisitos serve para a recolha e análise de informação relevante para a implementação da Base de Dados na empresa. O modelo conceptual é um modelo de dados independente de considerações físicas que demonstra relações entre tipos de dados. De seguida é criado modelo logico à custa do modelo anterior onde se remove construções que não podem ser representadas em modelos relacionais. Por fim o modelo logico é traduzido num modelo físico onde é possível inserir e aceder a dados de forma segura.

Este relatório contém todas as etapas e passos seguidos na construção da Base de Dados relacional, tendo em conta fatores como a redundância, eficiência e normalização e consistência.

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

Palavras-Chave: Bases de Dados Relacionais, Modelo Conceptual, Modelo Lógico, Modelo Físico, Normalização de uma base de dados.

Índice

1. Introdução	1
1.1. Contextualização	1
1.2. Apresentação do Caso de Estudo	1
1.3. Motivação e Objetivos	1
1.4. Viabilidade, vantagens e desvantagens	2
1.5. Levantamento de Requisitos	3
1.6. Perfis de Utilização	3
2. Construção do Modelo Conceptual	5
2.1. Identificação dos tipos das Entidades	5
2.2. Identificação dos tipos dos relacionamentos	6
2.3. Identificação dos atributos de cada entidade	6
2.4. Atributos chave primária, candidatas e alternativas	8
2.5. Verificar redundância do modelo	10
2.6. Validar o modelo conceptual segundo transações do utilizador	10
3. Construção do Modelo Lógico	12
3.1. Relacionamentos do modelo lógico	12
3.2. Validar modelo através da normalização	17
3.3. Validar modelo para transações de utilizador	20
3.4. Validar limites de integridade	21
3.5. Restrições gerais	23
3.6. Verificar crescimento futuro	23
4. Modelo Físico	24
4.1. Esquematizar modelo base	24
4.2. Restrições gerais do modelo	27
4.3. Análise de transações	27
4.4. Estimativa de espaço em disco ocupado	30
5. Conclusões e Trabalho Futuro	31

Anexos

I. Inserir Comboios	35
II. Inserir Estações	36
III. Inserir Lugares dos Comboios	37
IV. Inserir Viagens	38

Índice de Figuras

Figura 1 - Esquema Modelo Conceptual	9
Figura 2 - Entidade Passageiro	12
Figura 3 - Entidade Estacao	12
Figura 4 - Entidade Viagem	13
Figura 5 - Entidade Comboio	13
Figura 6 - Entidade Bilhete	13
Figura 7 - Relacionamento Passageiro-Bilhete	14
Figura 8 - Relacionamento Bilhete-Viagem	14
Figura 9 - Relacionamento Viagem-Estacao	15
Figura 10 - Relacionamento Viagem-Comboio	15
Figura 11 - Multivalor NúmeroLugar	16
Figura 12 - Multivalor LugarComboio	16
Figura 13 - Esquema Modelo Lógico	17
Figura 14 - Transação Registrar Passageiro	20
Figura 15 - Transação Reservar Lugar	21
Figura 16 - Limites de integridade do Passageiro	21
Figura 17 - Limites de integridade do Bilhete	22
Figura 18 - Limites de integridade da Viagem	22
Figura 19 - Limites de integridade da Estacao	22
Figura 20 - Limites de integridade do Comboio	23
Figura 21 - Limites de integridade do LugarComboio	23
Figura 22 - Modelo físico Passageiro	24
Figura 23 - Modelo físico LugarComboio	24
Figura 24 - Modelo físico Bilhete	25
Figura 25 - Modelo físico Comboio	25
Figura 26 - Modelo físico Viagem	26
Figura 27 - Modelo físico Estacao	26
Figura 28 - Transação Registrar Passageiro	28
Figura 29 - Transação Consultar Viagens	28
Figura 30 - Transação Consultar Lugares	28

Figura 31 - Transação Reservar Lugar	29
Figura 32 - Transação Consultar Bilhetes	29

Índice de Tabelas

Tabela 1 - Passageiro	18
Tabela 2 - Bilhete	18
Tabela 3 - LugarComboio	18
Tabela 4 - Viagem	18
Tabela 5 - Comboio	18
Tabela 6 - Estação	19
Tabela 7 - Transações [I - Inserir L - Leitura]	27
Tabela 8 - Estimativa de espaço ocupado em 10 anos	30

1. Introdução

1.1. Contextualização

A Comboios Europeus (CE) é uma empresa de transporte ferroviário que opera nas principais cidades europeias. Efetua a ligação entre estas cidades com preços competitivos e de forma rápida, já que não efetua paragens intermédias.

1.2. Apresentação do Caso de Estudo

Nos primeiros anos como o número de passageiros era reduzido, não havia necessidade de gerir as viagens disponibilizadas, uma vez que a lotação dos comboios nunca era esgotada.

Assim, o sistema de reservas de viagens era, e é, feito da seguinte forma: o passageiro desloca-se à bilheteira e indica os seus dados e os dados da viagem que pretende fazer (partida, destino e data da viagem); o funcionário insere os dados no sistema informático e imprime um bilhete para o passageiro com os dados fornecidos, enquanto guarda no computador um ficheiro com as mesmas informações.

1.3. Motivação e Objetivos

Motivação

Nos últimos anos, o número de passageiros tem aumentado significativamente o que tem levado à contratação de mais funcionários e ao aumento do número de lugares dos comboios, para responder à procura.

No entanto, o processo de reserva de viagens, para além de demorado e repetitivo, não permite fazer uma gestão eficiente das reservas, uma vez que as reservas apenas ficam guardadas no computador onde são feitas.

Objetivos

A administração da CE concordou em alterar a forma como os dados são guardados, passando a utilizar um Sistema de Base de Dados.

Com esta implementação, a empresa pretende atingir os seguintes objetivos:

- Permitir o registo dos passageiros no sistema, para acelerar a reserva das viagens;
- Oferecer um serviço web que permita aos passageiros reservar as viagens a partir de casa, reduzindo assim o número de funcionários necessários;
- Possibilitar a escolha de lugares disponíveis nos comboios.

1.4. Viabilidade, vantagens e desvantagens

Viabilidade do Projeto

A implementação de um sistema de Base de Dados relacional na CE é viável, uma vez que os dados a guardar e gerir dependem uns dos outros, podendo-se assim tirar partido das funcionalidades de um sistema de gestão de base de dados relacional. A passagem de um sistema de ficheiros para um sistema de gestão de base de dados relacional não é simples, mas irá trazer elevados ganhos operacionais.

Vantagens e Ganhos operacionais

Com a implementação de um sistema de Base de Dados relacional a partilha de informação é rápida e simples pois a informação pode ser acedida remotamente por qualquer pessoa autorizada, eliminando-se, assim, o problema de a informação estar em apenas um local como no sistema de ficheiros. Outro fator que é melhorado, é o aumento na integridade da informação, porque é possível a criação de regras para a informação colocada na base de dados, como limites (Ex.: cada passageiro só pode reservar um lugar para cada viagem num dado dia) e tipos específicos.

Com o sistema de ficheiros é necessário fazer backup dos ficheiros regularmente, para o caso de perda ou destruição acidental da informação. No caso de perda de informação é necessário voltar a introduzir toda a informação desde o ultimo backup. Esta preocupação não é necessária num sistema de gestão de BD pois estes minimizam o processo no caso de uma falha.

1.5. Levantamento de Requisitos

- A Comboios Europeus oferece aos passageiros uma grande variedade de viagens. Cada viagem é feita diariamente e tem uma estação de partida e uma estação de chegada, ambas com a respetiva hora.
- As estações pertencem a uma cidade. Apesar de todas as viagens terem de ter uma estação de origem e uma estação de destino, as estações podem não ser origem e/ou destino de viagens.
- As viagens são efetuadas por comboios. Os comboios têm, também, todos os lugares (cada um com o seu número) e podem fazer várias viagens em cada dia.
- A reserva, feita por um passageiro, de uma viagem contém a data da viagem e o número do lugar escolhido no comboio que efetua a viagem.

1.6. Perfis de Utilização

Administrador da base de dados

- Os dados que não são controlados pelos passageiros (comboios, viagens e estações) serão geridos pelos administradores da base de dados. São estes que terão de adicionar, atualizar e, porventura, remover os dados necessários para que a empresa mantenha o seu normal funcionamento.

Passageiro

- Os passageiros para utilizarem os serviços oferecidos pela Comboios Europeus tem de estar registados no sistema. Para isso tem de fornecer o seu nome, email e password.
- Um dos serviços que os passageiros podem utilizar é a consulta de viagens. Para isso escolhem as cidades de origem e de destino.
- É possível, também, consultar os números dos lugares disponíveis para uma viagem. Para isso o passageiro tem de indicar a viagem, depois de a consultar, e o dia pretendido.
- Outro serviço que os passageiros podem utilizar é a reserva de viagens. Depois de consultar as viagens e os números dos lugares disponíveis para essa viagem o passageiro escolhe o número do lugar em que pretende viajar.

- Por fim pretende-se que seja possível, ao passageiro, consultar todas as viagens que reservou apresentando os detalhes de cada viagem (estações de origem e destino e a hora de cada uma) bem como a data da viagem e o lugar respetivo.

2. Construção do Modelo Conceptual

Depois de identificar os requisitos e os perfis de utilização, partimos para a construção do modelo conceptual. Para isso, vamos identificar as entidades, os relacionamentos e os atributos. De seguida iremos determinar os domínios dos atributos e as chaves primárias para cada entidade. Terminamos com a validação do modelo.

2.1. Identificação dos tipos das Entidades

Viagem

- Descrição: Todas as viagens disponibilizadas pela empresa.
- Sinónimos: Percurso.
- Ocorrência: As viagens são realizadas por um comboio e tem estação de origem e destino sendo feitas diariamente. As viagens podem ser consultadas pelos passageiros.

Estação

- Descrição: Todos as estações disponíveis para pontos de partida e/ou destino.
- Sinónimos: Paragem.
- Ocorrência: As estações são pontos de partida e/ou chegada de viagens. Os passageiros podem escolher as estações de origem e destino ao fazer a consulta de viagens.

Comboio

- Descrição: Todos os comboios que a empresa detém.
- Ocorrência: Os comboios realizam as viagens, podendo fazer várias viagens em cada dia. Contém os lugares que podem ser reservados.

Bilhete

- Descrição: Todas as reservas efetuadas.
- Sinónimos: Reserva.
- Ocorrência: As reservas são efetuadas por passageiros estando associadas a uma viagem, a uma data e a um lugar do comboio.

Passageiro

- Descrição: Todo o utilizador registado no sistema.
- Sinónimos: Cliente, Utilizador.
- Ocorrência: Os passageiros efetuam várias reservas de viagens. Podem também consultar as viagens bem como os lugares disponíveis em cada uma.

2.2. Identificação dos tipos dos relacionamentos

Entidade	Multiplicidade	Relacionamento	Multiplicidade	Entidade
Passageiro	1	Reserva	0..n	Bilhete
Bilhete	0..n	Referente	1	Viagem
Viagem	0..n	Inicia	1	Estação
Viagem	0..n	Termina	1	Estação
Viagem	0..n	É feita por	1	Comboio

2.3. Identificação dos atributos de cada entidade

Entidade	Atributo	Descrição	Domain	Nulo	Multivalor	Derivado
Passageiro	ID	Identificador do passageiro	Número inteiro positivo	Não	Não	Não
Passageiro	Nome	Nome do passageiro	String de tamanho variável com comprimento máximo de 45 caracteres	Não	Não	Não
Passageiro	Email	Email do passageiro	String de tamanho variável com comprimento máximo de	Não	Não	Não

			45 caracteres			
Passageiro	Password	Password do passageiro para se autenticar	<i>String</i> de tamanho variável com comprimento máximo de 45 caracteres	Não	Não	Não
Bilhete	DataViagem	Data em que vai realizar a viagem que reservou	Data	Não	Não	Não
Bilhete	ID	Identificador de reserva	Número inteiro positivo	Não	Não	Não
Bilhete	NúmeroLugar	Número do Lugar	Número inteiro positivo	Não	Não	Não
Viagem	ID	Identificador da viagem	Número inteiro positivo	Não	Não	Não
Viagem	HoraPartida	Hora a que o comboio parte	Tempo	Não	Não	Não
Viagem	HoraChegada	Hora a que o comboio chega ao destino	Tempo	Não	Não	Não
Estação	ID	Identificador da Estação	Número inteiro positivo	Não	Não	Não
Estação	Nome	Nome da estação	<i>String</i> de tamanho variável com comprimento máximo de 45 caracteres	Não	Não	Não
Estação	Cidade	Cidade a que	<i>String</i> de	Não	Não	Não

		pertence a estação	tamanho variável com comprimento máximo de 45 caracteres			
Comboio	ID	Identificador do Comboio	Número inteiro positivo	Não	Não	Não
Comboio	Observações	Observações sobre o comboio	Texto	Sim	Não	Não
Comboio	NúmeroLugar	Identificação dos lugares no comboio	Número inteiro positivo	Não	Sim	Não

2.4. Atributos chave primária, candidatas e alternativas

Passageiro

Chaves candidatas: ID e Email.

Ambos os atributos identificam um único passageiro, já que, quer o ID quer o Email são únicos. Escolhemos como chave primária o ID, pois este não irá ser alterado (numa implementação futura o Email poderá ser alterado) e é o atributo com o menor número de caracteres.

Chave primária: ID

Bilhete

Chaves candidatas: ID

Na entidade bilhete apenas encontramos uma chave candidata logo a escolha é simples.

Chave primária: ID

Viagem

Chaves candidatas: ID

Tal como na entidade viagem, também na entidade viagem apenas encontramos uma chave candidata logo a escolha também é simples.

Chave primária: ID

Comboio

Chaves candidatas: ID

Mais uma vez, neste caso na entidade comboio apenas encontramos uma chave candidata logo a escolha volta a ser simples.

Chave primária: ID

Estação

Chaves candidatas: ID e Nome+Cidade

Na entidade estação temos duas chaves candidatas. Decidimos escolher o atributo ID já que contém apenas um atributo (ao invés da outra chave candidata que continha dois).

Chave primária: ID

Depois de identificar as entidades, os relacionamentos e os atributos, construímos o seguinte esquema:

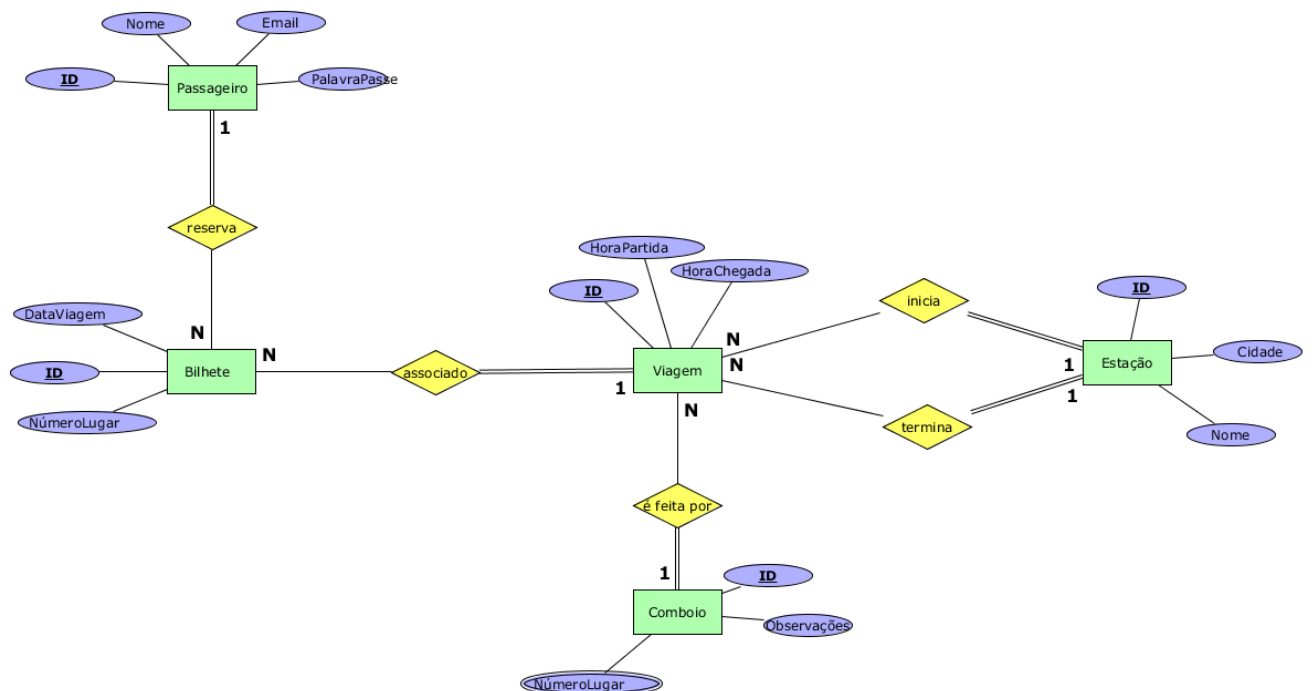


Figura 1 - Esquema Modelo Conceptual

2.5. Verificar redundância do modelo

Verificar relacionamentos um-para-um

Como não temos nenhum relacionamento um para um este passo não foi considerado.

Remoção de relacionamentos redundantes

Entre as entidades Viagem e Estação apresentamos dois relacionamentos: Viagem inicia Estação e Viagem termina Estação. Neste caso é necessária a existência de dois relacionamentos entre as viagens e as estações, visto que precisamos de duas estações diferentes como pontos de origem e destino do percurso que será realizado, prevenindo também a ocorrência de viagens com origem e destino iguais. Pode acontecer, contudo, que as estações não tenham nenhuma viagem associada, uma vez que é possível a existência de estações que não constem nos pontos de origem e destino de qualquer itinerário praticado pela empresa.

2.6. Validar o modelo conceptual segundo transações do utilizador

Transação Registar Passageiro

As informações necessárias para que um utilizador possa realizar o seu registo são os atributos Nome, Email e Password da entidade Passageiro.

Transação Consultar Viagens

Para que se proceda à consulta, recorre-se a todos os atributos da entidade Bilhete e duas vezes (para a partida e para a chegada) ao atributo Cidade da entidade Estação. Com os relacionamentos "Viagem inicia Estação" e "Viagem termina Estação" é criada a lista pretendida.

Transação Consultar Lugares

Para consultar os lugares disponíveis de uma dada viagem é necessário seleccionar o atributo ID da entidade Viagem (para o passageiro saber o ID tem de, anteriormente, ter feito uma consulta de viagens). Com o relacionamento "Viagem é feita por Comboio" e com o atributo multivalor Número do Lugar da entidade Comboio é produzida a lista de todos os lugares para essa viagem. Com o relacionamento "Bilhete reserva Viagem" e com os atributos Número do Lugar e Data da Viagem da entidade Bilhete são removidas da listagem anterior todos os lugares que estão ocupados sendo apresentados os números dos lugares disponíveis da viagem seleccionada para o dia escolhido.

Transação Reservar Lugar

Para que se proceda à reserva são feitos os mesmos passos da transação "Consulta Lugares". Ao obter-se a listagem final caso o lugar escolhido pelo passageiro pertença a essa lista a reserva é confirmada.

Transação Consultar Bilhetes

A fim de consultar todas as viagens reservadas por um dado passageiro é necessário selecionar o atributo ID/Email da entidade Passageiro. Com os atributos DataViagem e NúmeroLugar e o relacionamento "Passageiro reserva Bilhete" é conhecida a lista de Bilhetes. Com os atributos HoraPartida e HoraChegada e com o relacionamento "Bilhete associado Viagem" conhecemos alguns detalhes de cada Bilhete. Por fim com o atributo Cidade e os relacionamentos "Viagem inicia Estação" e "Viagem termina Estação" chega-se à lista pretendida.

3. Construção do Modelo Lógico

Neste passo construímos o modelo lógico a partir do modelo conceptual para, de seguida, o validarmos quer estruturalmente, quer para verificar que será possível realizar as transações pretendidas. Usamos a ferramenta *MySQL*.

3.1. Relacionamentos do modelo lógico

Tipos das Entidades

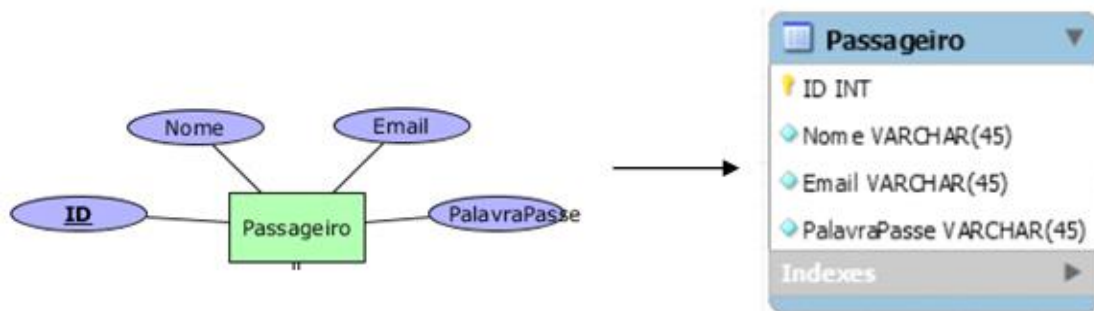


Figura 2 - Entidade Passageiro

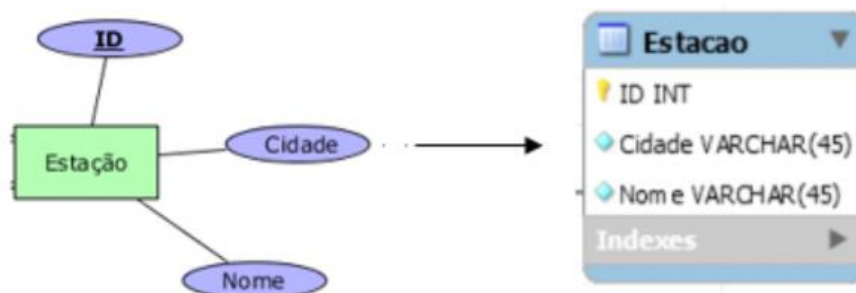


Figura 3 - Entidade Estacao

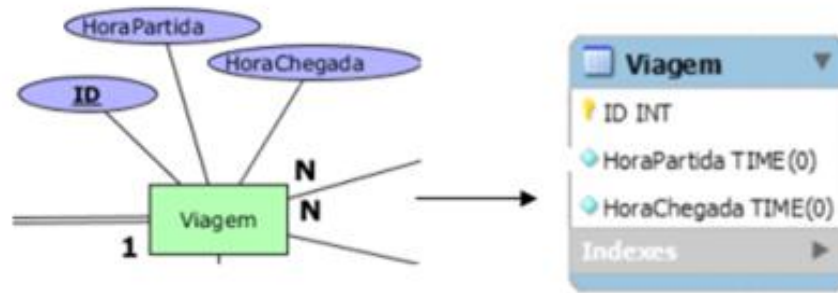


Figura 4 - Entidade Viagem

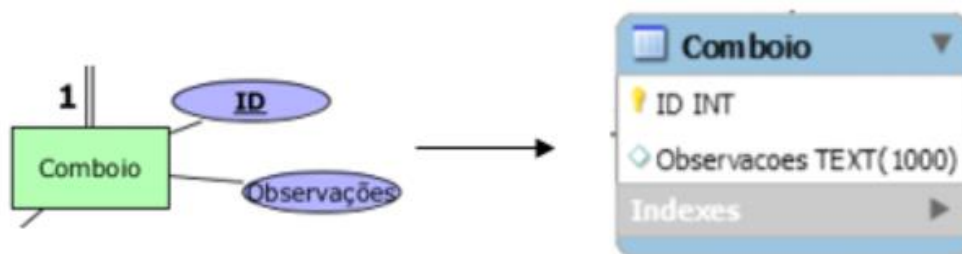


Figura 5 - Entidade Comboio

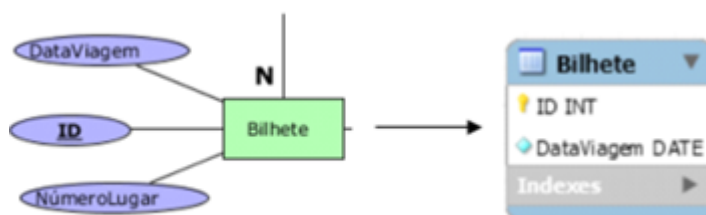


Figura 6 - Entidade Bilhete

Nota: na construção da tabela Bilhete não introduzimos o atributo Número de Lugar, como tínhamos feito no modelo conceptual. Esta "omissão" será explicada mais à frente.

Relacionamentos 1 para N

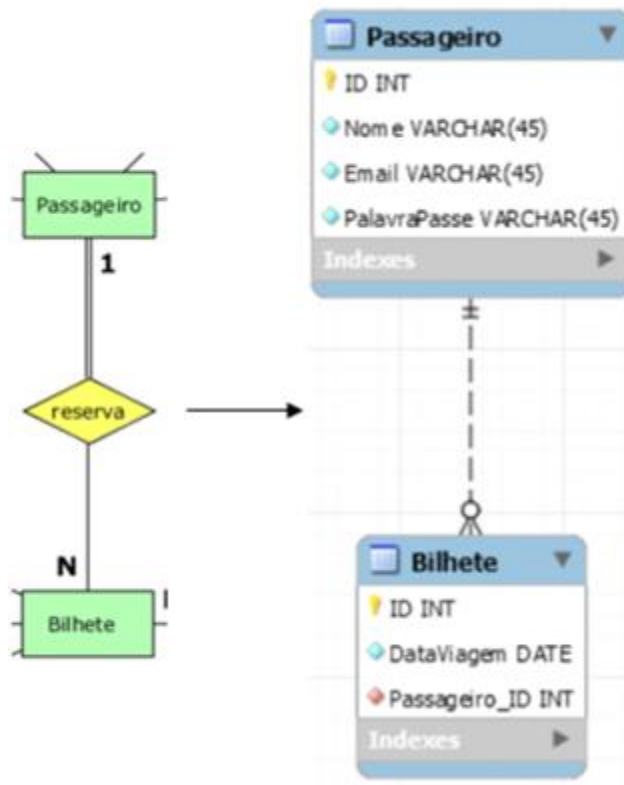


Figura 7 - Relacionamento Passagiro-Bilhete

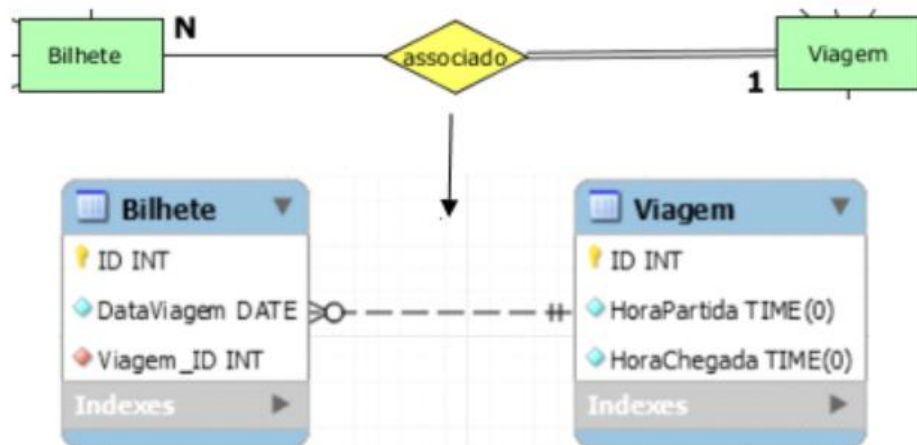


Figura 8 - Relacionamento Bilhete-Viagem

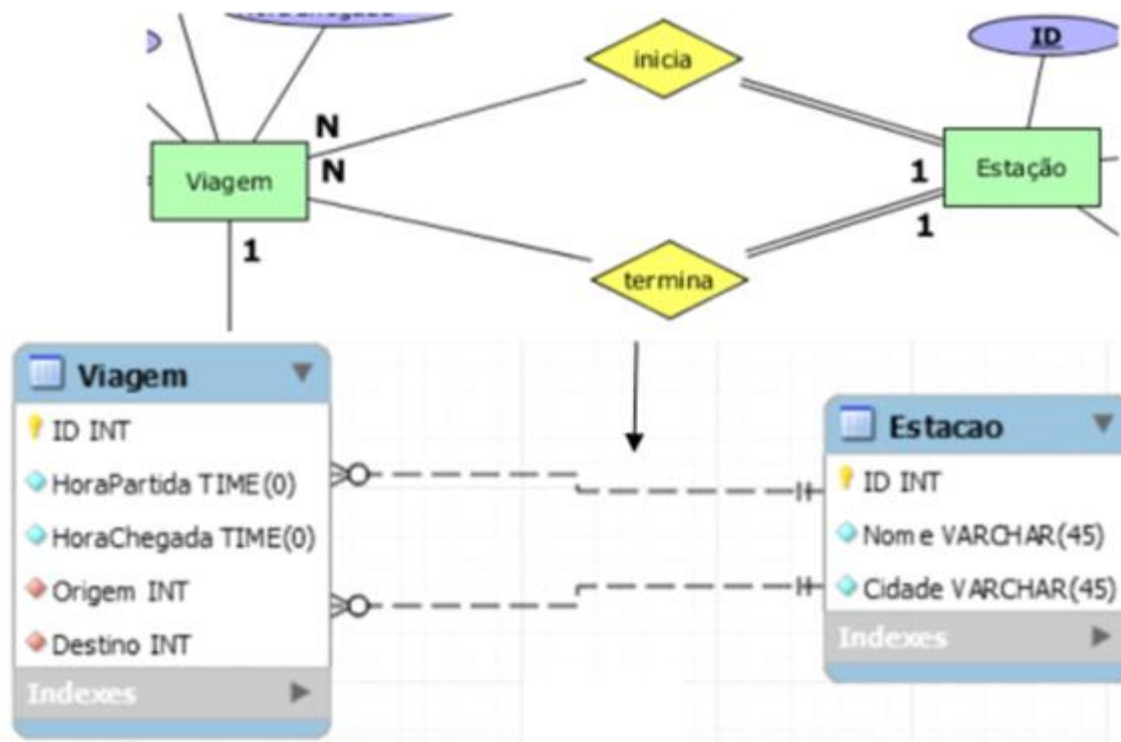


Figura 9 - Relacionamento Viagem-Estacao

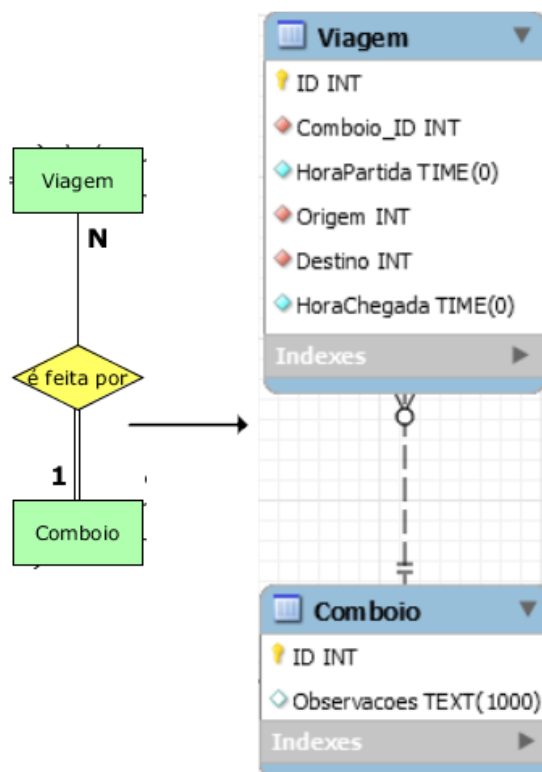


Figura 10 - Relacionamento Viagem-Comboio

Atributos multivalor

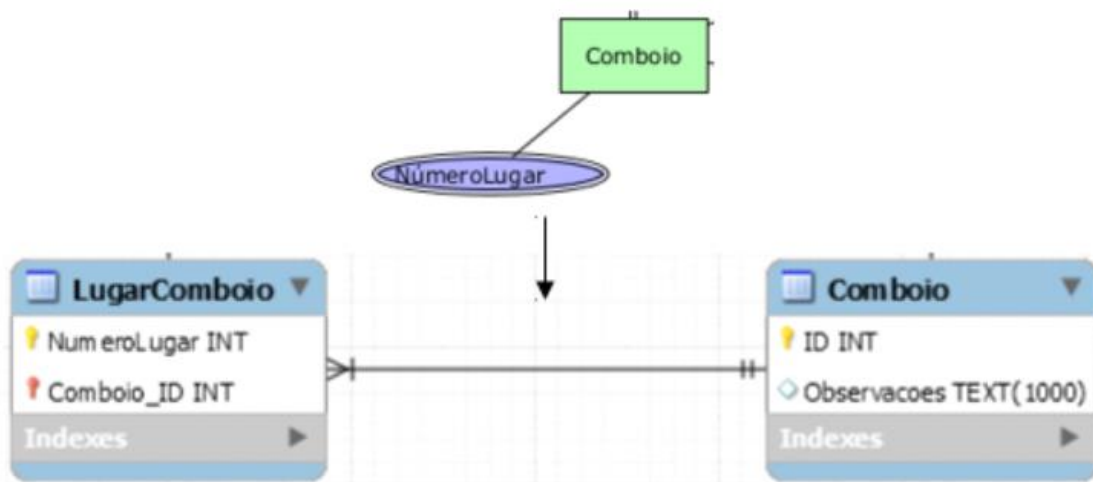


Figura 11 - Multivalor NúmeroLugar

NOTA: No modelo conceptual, por desconhecimento da notação a utilizar, não definimos uma relação entre o atributo Número do Lugar da entidade Bilhete e o atributo multi-valor da entidade Comboio. No entanto, no modelo lógico, uma vez que os atributos multi-valor são representados numa tabela, decidimos definir um relacionamento entre as entidades Bilhete e LugarComboio. Assim, o atributo Número de Lugar que pertencia à entidade Bilhete fica representado na tabela Bilhete como uma chave estrangeira.

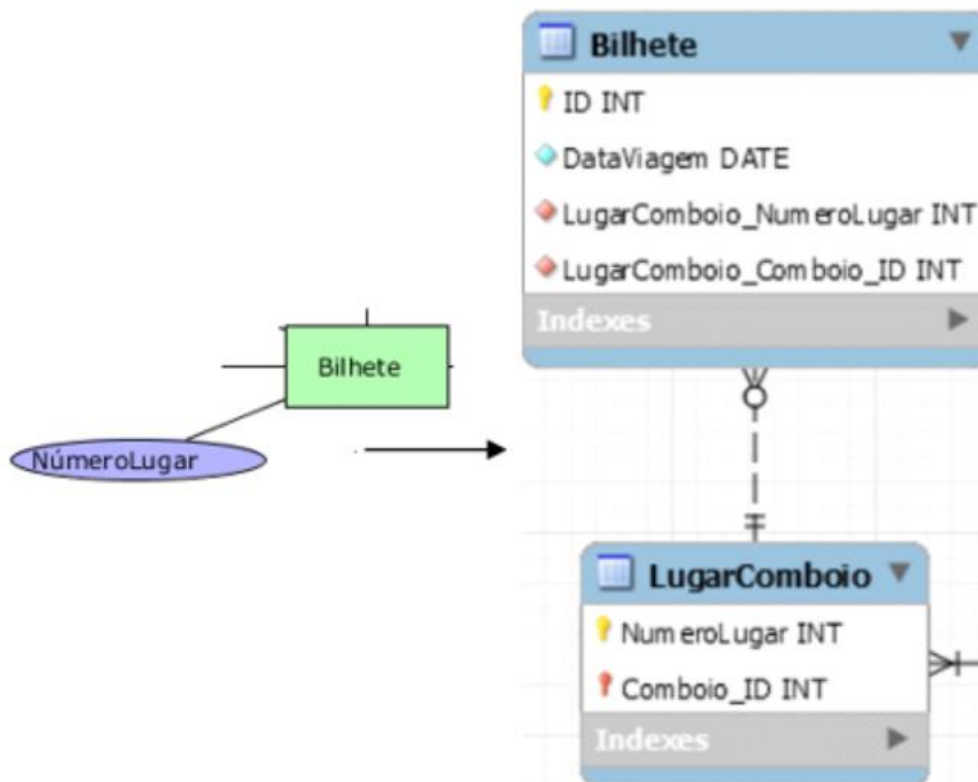


Figura 12 - Multivalor LugarComboio

Depois de derivar as entidades, os relacionamentos e os atributos, construímos o seguinte esquema lógico:

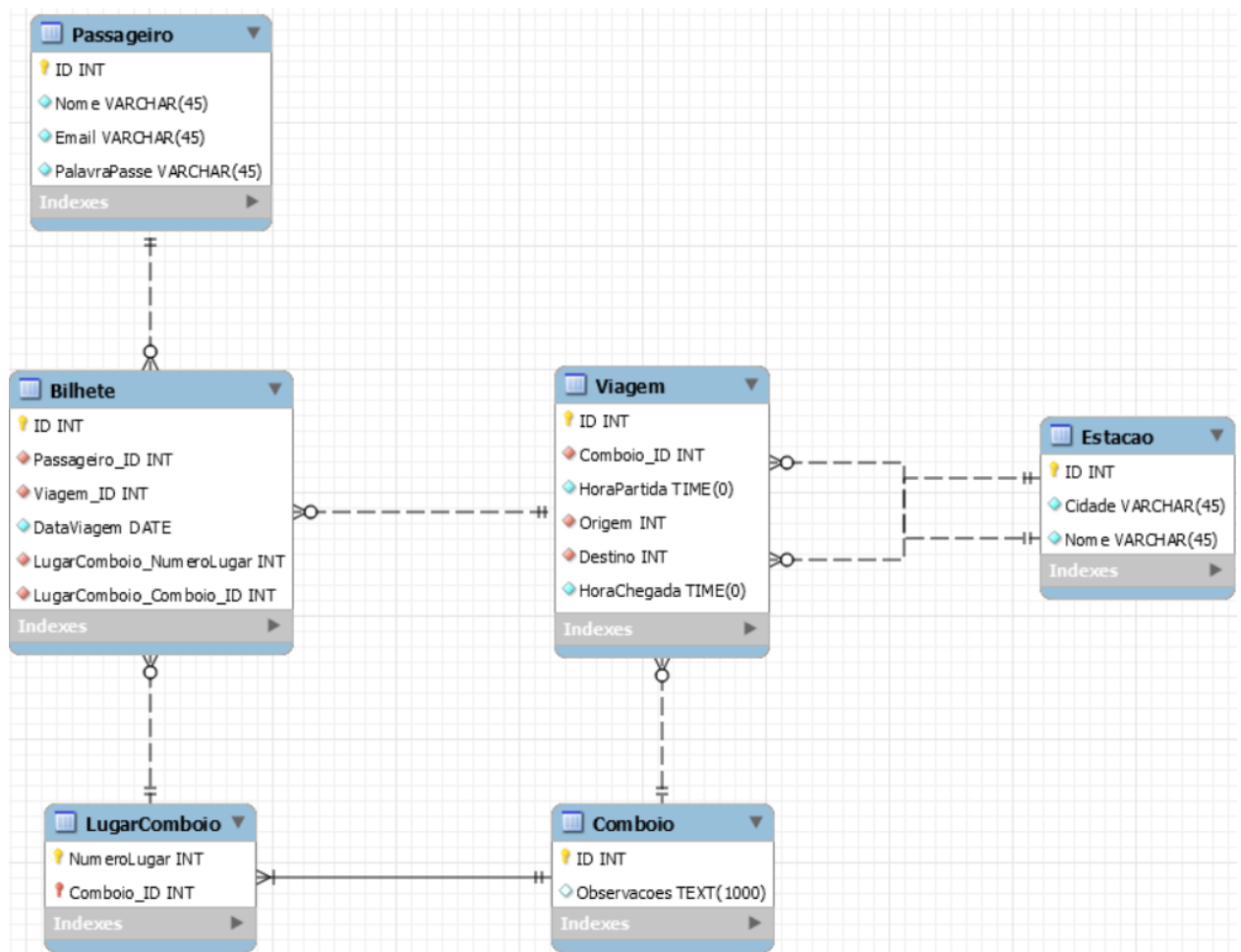


Figura 13 - Esquema Modelo Lógico

3.2. Validar modelo através da normalização

Primeira forma normal (1FN)

A primeira forma normal consiste em identificar, nas tabelas, atributos que estejam na forma não atômica, ou seja, verificar que nas interseções de cada linha e cada coluna contém apenas um valor.

Tabela 1 - Passageiro

ID	Nome	Email	Password
1	Bruno Machado	brunomachado@email.com	brunomachado
2	Fabio Baiao	fabiobaiao@email.com	fabiobaiao
3	Joao Rui	joaorui@email.com	joaorui
4	Luis Costa	luiscosta@email.com	luiscosta

Tabela 2 - Bilhete

ID	Passageiro_ID	Viagem_ID	DataViagem	LugarComboio_NúmeroLugar	LugarComboio_Comboio_ID
1	3	1	27/11/2016	3	1

Tabela 3 - LugarComboio

NúmeroLugar	Comboio_ID
1	1
2	1
3	1

Tabela 4 - Viagem

ID	Comboio_ID	HoraPartida	Origem	Destino	HoraChegada
1	1	07:00	1	2	11:45
2	3	12:30	2	3	17:00
3	2	18:00	3	5	22:30

Tabela 5 - Comboio

ID	Observações
1	Comboio com bar. Atinge velocidades na ordem dos 200 km/h
2	Comboio de luxo com restaurante, bar e casa de banho. Atinge velocidades de 300 km/h
3	Comboio simples, recomendado para viagens nacionais. Atinge velocidades de 150 km/h

Tabela 6 - Estação

ID	Cidade	Nome
1	Lisboa	Oriente
2	Madrid	Atocha
3	Barcelona	Sants
4	Bordeaux	Saint Jean
5	Lyon	Part Dieu
6	Paris	Gare de Lyon

Como se verifica todas as entradas das tabelas apresentam valores atômicos, logo podemos concluir que todas as tabelas se encontram na primeira forma normal.

Segunda forma normal (2FN)

Para uma tabela se encontrar na segunda forma normal, em primeiro lugar, tem de estar na primeira forma normal. Como vimos na secção anterior, todas as nossas tabelas já se encontram na primeira fase normal.

Na segunda forma normal, apenas temos de verificar as tabelas que tenham como chave primária uma composição de vários atributos, já que as tabelas com uma chave primária simples já se encontram na segunda forma normal. No nosso modelo lógico apenas a tabela LugarComboio apresenta esta característica.

A última condição para uma tabela estar na segunda forma normal é que todos os atributos que não pertençam à chave primária têm de ser totalmente dependentes da chave primária, isto é, não podem ser dependentes de apenas um dos atributos que compõem a chave primária por exemplo. Uma vez que a tabela LugarComboio não tem atributos que não pertençam à chave primária, já se encontra na segunda forma normal.

Terceira forma normal (3FN)

Um dos requisitos para uma tabela se encontrar na terceira forma normal é já estar na segunda forma normal (e, portanto, na primeira também). Já confirmamos anteriormente que todas as tabelas se encontram na segunda forma normal.

O outro requisito é que nenhum atributo que não pertença à chave primária seja transitivamente dependente da chave primária.

3.3. Validar modelo para transações de utilizador

Transação Registar Passageiro

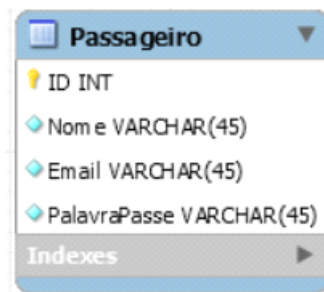


Figura 14 - Transação Registar Passageiro

Esta transação consiste em, simplesmente, inserir uma entrada nesta tabela com os dados (atributos) do passageiro.

Transação Consultar Viagens

$ViagensOrigem \leftarrow Viagem \bowtie_{(origem = ID)} (\sigma_{(cidade = partida)} Estação)$

$ViagensOrigemDestino \leftarrow ViagensOrigem \bowtie_{(destino = ID)} (\sigma_{(cidade = chegada)} Estação) \pi (ID, HoraPartida, HoraChegada)$

$ViagensOrigemDestino$

Transação Consultar Lugares

$LugaresViagem \leftarrow LugarComboio \bowtie_{(Comboio_ID = Comboio_ID)} (\pi_{(Comboio_ID)} (\sigma_{(ID = viagem)} Viagem))$

$LugaresReservados \leftarrow \pi (LugarComboio_NúmeroLugar, LugarComboio_Comboio_ID) (\sigma_{(Viagem_ID = viagem \text{ and } DataViagem = data) Bilhete}) \pi (NúmeroLugar) (LugaresViagem - LugaresReservados)$

Transação Reservar Lugar

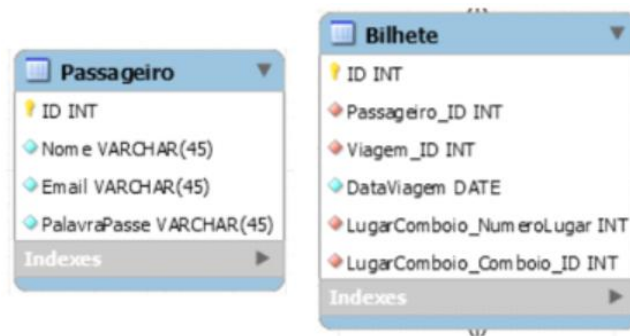


Figura 15 - Transação Reservar Lugar

Para efetuar uma reserva, em primeiro lugar, é necessário que o Email e a Password fornecidas pelo passageiro, existam e coincidam com os dados que existem na tabela Passageiro. Caso isto se verifique, é feita a inserção na tabela Bilhete dos dados (Viagem, Data e Número do Lugar) que o passageiro escolhe.

Transação Consultar Bilhetes

$\text{BilhetesPassageiro} \leftarrow \pi_{(\text{Viagem_ID}, \text{DataViagem}, \text{LugarComboio_NúmeroLugar})} (\pi_{(\text{ID})} \text{Passageiro}) \bowtie_{(\text{ID} = \text{Passageiro_ID})} \text{Bilhete}$

$\text{BilhetesPassageiroViagem} \leftarrow \pi_{(\text{DataViagem}, \text{LugarComboio_NúmeroLugar}, \text{HoraPartida}, \text{HoraChegada}, \text{Origem}, \text{Destino})} (\text{BilhetesPassageiro} \bowtie_{(\text{Viagem_ID} = \text{ID})} \text{Viagem})$

$\text{BilhetesPassageiroViagemEstações} \leftarrow (\text{BilhetesPassageiroViagem} \bowtie_{(\text{Origem} = \text{ID})} \text{Estação}) \bowtie_{(\text{Destino} = \text{ID})} \text{Estação} \pi_{(\text{DataViagem}, \text{LugarComboio_NúmeroLugar}, \text{HoraPartida}, \text{HoraChegada}, \text{Cidade}, \text{Cidade})} \text{BilhetesPassageiroViagemEstações}$

3.4. Validar limites de integridade

Passageiro

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ID	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(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"/>	
Email	VARCHAR(45)	<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"/>	
PalavraPasse	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 16 - Limites de integridade do Passageiro

Bilhete

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ID	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"/>	
Passageiro_ID	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"/>	
Viagem_ID	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"/>	
DataViagem	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"/>	
LugarComboio_NumeroLugar	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"/>	
LugarComboio_Comboio_ID	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 17 - Limites de integridade do Bilhete

Passageiro_ID: ON DELETE CASCADE, ON UPDATE CASCADE (ao eliminar um passageiro, os bilhetes desse passageiro são todos eliminados)

Viagem_ID: ON DELETE NO ACTION, ON UPDATE CASCADE (uma viagem só pode ser eliminada caso esta não tenha bilhetes associados)

LugarComboio_*: ON DELETE NO ACTION, ON UPDATE CASCADE (um lugar só pode ser eliminado se não estiver associado a nenhum bilhete)

Viagem

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ID	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_ID	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"/>	
HoraPartida	TIME(0)	<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"/>	
Origem	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"/>	
Destino	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"/>	
HoraChegada	TIME(0)	<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 18 - Limites de integridade da Viagem

Origem/Destino: ON DELETE NO ACTION, ON UPDATE CASCADE (uma estação só pode ser eliminada se não for origem nem destino de nenhuma viagem).

Comboio_ID: ON DELETE NO ACTION, ON UPDATE CASCADE (um comboio só pode ser eliminado se não estiver associado a nenhuma viagem)

Estação

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ID	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"/>	
Cidade	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"/>	
Nome	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 - Limites de integridade da Estacao

Comboio										
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ID	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"/>	
Observacoes	TEXT(1000)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 20 - Limites de integridade do Comboio

LugarComboio										
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
NumeroLugar	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_ID	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 21 - Limites de integridade do LugarComboio

Comboio_ID: ON DELETE CASCADE, ON UPDATE CASCADE (ao eliminar um comboio, todas os lugares desse comboio são eliminados)

Em todas as tabelas com chaves estrangeiras verificamos esta condição: ao fazer uma inserção de um tuplo numa tabela que contem chaves estrangeiras ou ao alterar uma chave estrangeira de um dos tuplos existentes é necessário verificar que a(s) chave(s) estrangeira(s) corresponde(m) a um tuplo existente.

Para todas as chaves estrangeiras definimos que ao ser feita uma alteração dessa chave na tabela "pai", essa alteração é propagada para a tabela "filho".

3.5. Restrições gerais

Cada passageiro só pode reservar um lugar por viagem (em cada dia). Logo, se o passageiro reservar um lugar na viagem A para o dia X, não deverá ser possível o mesmo passageiro voltar a reservar um lugar na viagem A para o dia X.

3.6. Verificar crescimento futuro

Uma tarefa a realizar no futuro seria criar um perfil de utilização para a administração. Este serviria para controlar todas a inserções da base de dados e, simultaneamente, verificar todas atualizações relacionadas com as entidades comboio, viagem e estação. Para isto seria necessário criar um novo modelo conceptual e, consequentemente, um novo modelo lógico, o qual seria integrado ao modelo lógico atual.

4. Modelo Físico

Para terminar a construção da base de dados construímos o modelo físico. A ferramenta MySQL contém uma opção designada *Forward Engineering* (FE) que gera o código SQL para a construção das tabelas a partir do modelo lógico. De seguida, populamos a base de dados com alguns dados (o código utilizado encontra-se em anexo) e criamos as transações a que nos propusemos anteriormente.

4.1. Esquematizar modelo base

Passageiro

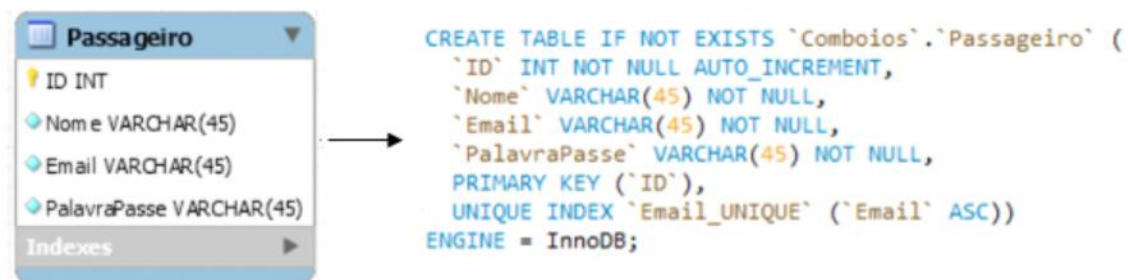


Figura 22 - Modelo físico Passageiro

LugarComboio

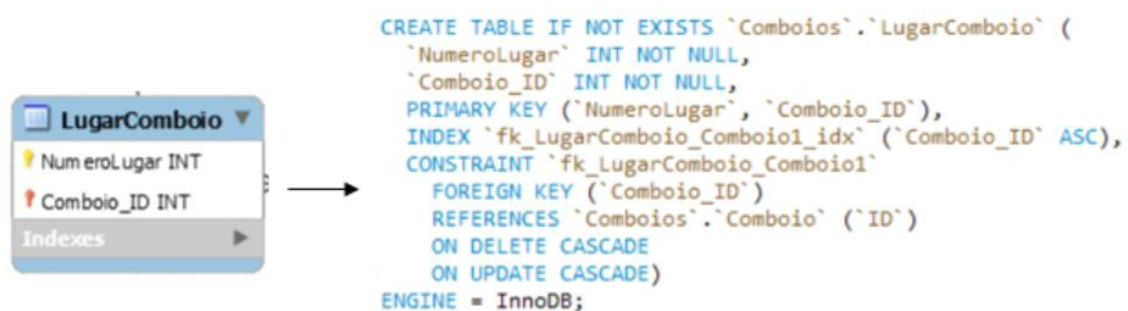


Figura 23 - Modelo físico LugarComboio

Bilhete



Figura 24 - Modelo físico Bilhete

Comboio

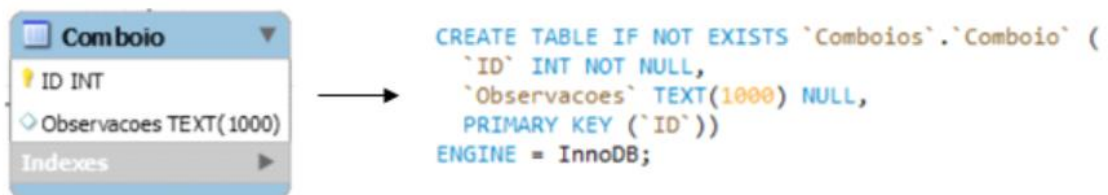


Figura 25 - Modelo físico Comboio

Viagem

Viagem
ID INT
Comboio_ID INT
HoraPartida TIME(0)
Origem INT
Destino INT
HoraChegada TIME(0)
Indexes

```
CREATE TABLE IF NOT EXISTS `Comboios`.`Viagem` (
  `ID` INT NOT NULL,
  `Comboio_ID` INT NOT NULL,
  `HoraPartida` TIME(0) NOT NULL,
  `Origem` INT NOT NULL,
  `Destino` INT NOT NULL,
  `HoraChegada` TIME(0) NOT NULL,
  PRIMARY KEY (`ID`),
  INDEX `fk_Viagem_Comboio1_idx` (`Comboio_ID` ASC),
  INDEX `fk_Viagem_Estacao1_idx` (`Origem` ASC),
  INDEX `fk_Viagem_Estacao2_idx` (`Destino` ASC),
  CONSTRAINT `fk_Viagem_Comboio1`
    FOREIGN KEY (`Comboio_ID`)
    REFERENCES `Comboios`.`Comboio` (`ID`)
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Viagem_Estacao1`
    FOREIGN KEY (`Origem`)
    REFERENCES `Comboios`.`Estacao` (`ID`)
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Viagem_Estacao2`
    FOREIGN KEY (`Destino`)
    REFERENCES `Comboios`.`Estacao` (`ID`)
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;
```

Figura 26 - Modelo físico Viagem

Estacao

Estacao
ID INT
Cidade VARCHAR(45)
Nome VARCHAR(45)
Indexes

```
CREATE TABLE IF NOT EXISTS `Comboios`.`Estacao` (
  `ID` INT NOT NULL,
  `Cidade` VARCHAR(45) NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`ID`))
ENGINE = InnoDB;
```

Figura 27 - Modelo físico Estacao

4.2. Restrições gerais do modelo

```

DELIMITER $$
CREATE TRIGGER UmaReservaPorViagem
BEFORE INSERT ON Bilhete
FOR EACH ROW
BEGIN
    IF (
        SELECT COUNT(*)
        FROM bilhete
        WHERE Passageiro_ID = NEW.Passageiro_ID AND
            DataViagem = NEW.DataViagem AND
            Viagem_ID = NEW.Viagem_ID
        ) >= 1
    THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Já comprou um bilhete para esta viagem para este dia';
    END IF;
END $$
DELIMITER ;

```

Figura 28 - Restrição do número de bilhetes reservados

4.3. Análise de transações

	Registrar Passageiro		Consultar Viagem		Consultar Lugares		Reservar Lugar		Consultar Bilhetes	
	I	L	I	L	I	L	I	L	I	L
Passageiro	X							X		X
Bilhete						X	X			X
Viagem				X		X		X		X
Estacao				X						X
LugarComboio						X				
Comboio										

Tabela 7 - Transações [I - Inserir | L - Leitura]

Chegamos, assim, a estas transações:

Registrar Passageiro

```
DELIMITER $$
CREATE PROCEDURE `RegistrarPassageiro`(IN Nome VARCHAR(45), IN Email VARCHAR(45), IN PalavraPasse VARCHAR(45))
BEGIN
    INSERT INTO passageiro (Nome, Email, PalavraPasse)
    VALUES (Nome, Email, PalavraPasse);
END$$
DELIMITER ;
```

Figura 29 - Transação Registrar Passageiro

Consultar Viagens

```
DELIMITER $$
CREATE PROCEDURE `ConsultarViagem`(IN Origem VARCHAR(45), IN Destino VARCHAR(45))
BEGIN
    SELECT viagem.ID, HoraPartida, HoraChegada, o.Nome AS 'Origem', d.Nome AS 'Destino'
    FROM viagem
    INNER JOIN (
        SELECT *
        FROM estacao
        WHERE Origem = Cidade) AS o
    ON viagem.origem = o.ID
    INNER JOIN (
        SELECT *
        FROM estacao
        WHERE Destino = Cidade) AS d
    ON viagem.destino = d.ID;
END$$
DELIMITER ;
```

Figura 30 - Transação Consultar Viagens

Consultar Lugares

```
DELIMITER $$
CREATE PROCEDURE `ConsultarLugares`(IN Viagem INT, IN DataV DATE)
BEGIN
    SELECT lc.NumeroLugar
    FROM viagem
    INNER JOIN lugarcomboio AS lc
    ON viagem.Comboio_ID = lc.Comboio_ID
    WHERE 1 = ID AND (NumeroLugar) NOT IN (
        SELECT LugarComboio_NumeroLugar
        FROM Bilhete
        WHERE Viagem = Viagem_ID AND DataViagem = DataV);
END$$
DELIMITER ;
```

Figura 31 - Transação Consultar Lugares

Reservar Lugar

```
DELIMITER $$
CREATE PROCEDURE `ReservarLugar` (IN Viagem INT, IN DataV DATE, IN NumLugar INT, IN Mail VARCHAR(45), IN Passe VARCHAR(45))
BEGIN
    DECLARE Comboio, IDPassageiro INT;

    SELECT Comboio_ID INTO Comboio
    FROM viagem
    WHERE ID = Viagem;

    SELECT ID INTO IDPassageiro
    FROM passageiro
    WHERE Email = Mail AND
        PalavraPasse = Passe;

    IF IDPassageiro IS NOT NULL
    THEN
        INSERT INTO bilhete (Passageiro_ID, Viagem_ID, DataViagem, LugarComboio_NumeroLugar, LugarComboio_Comboio_ID)
        VALUES (IDPassageiro, Viagem, DataV, NumLugar, Comboio);
    END IF;
END $$
DELIMITER ;
```

Figura 32 - Transação Reservar Lugar

Consultar Bilhetes

```
DELIMITER $$
CREATE PROCEDURE `ConsultarBilhetes` (IN e VARCHAR(45), IN pass VARCHAR(45))
BEGIN
    IF
    (
        SELECT Passageiro.PalavraPasse
        FROM Passageiro
        WHERE Passageiro.Email = e
    ) = pass
    THEN
        SELECT Bilhete.ID AS 'ID Bilhete', Bilhete.DataViagem AS 'Data de Viagem',
            Bilhete.LugarComboio_NumeroLugar AS 'Lugar Nr', Bilhete.LugarComboio_Comboio_ID AS 'Comboio Nr',
            Viagem.HoraPartida AS 'Hora de Partida', concat(ori.Cidade, ' - ', ori.Nome) AS 'Estação Origem',
            Viagem.HoraChegada AS 'Hora de Chegada', concat(dest.Cidade, ' - ', dest.Nome) AS 'Estação Destino'
        FROM Passageiro
        INNER JOIN Bilhete ON Bilhete.Passageiro_ID = Passageiro.ID
        INNER JOIN Viagem ON Bilhete.Viagem_ID = Viagem.ID
        INNER JOIN Estacao AS ori ON Viagem.Origem = ori.ID
        INNER JOIN Estacao AS dest ON Viagem.Destino = dest.ID
        WHERE Passageiro.Email = e;
    END IF;
END $$
DELIMITER ;
```

Figura 33 - Transação Consultar Bilhetes

4.4. Estimativa de espaço em disco ocupado

Nome da tabela	Espaço unitário	Nº atual	Ganho por ano	Nº total	Espaço total
Bilhete	8112	2	12	122	989664
Comboio	5461	3	1	13	70993
Estacao	2730	6	1	16	43680
LugarComboio	546	30	10	130	70980
Passageiro	1638	2	100	1002	1641276
Viagem	5461	2	2	22	120142
Total:	23948	45	126	1305	2936735

Tabela 8 - Estimativa de espaço ocupado em 10 anos

Pelas nossas estimativas a Base de dados irá ocupar 2936735 bits que corresponde a 358 KBytes de informação ao longo de um período de funcionamento de 10 anos.

5. Conclusões e Trabalho Futuro

Durante a realização do projeto surgiram algumas dúvidas, nomeadamente no modelo conceptual, onde inicialmente tínhamos definido “lugar” como entidade. Esta implementação foi desaconselhada pelo docente, pois “lugar” não faria sentido ser uma entidade, uma vez que as entidades podem existir por si próprias, e neste contexto a existência de lugar sem ter um comboio associado não faria sentido. Assim, decidimos implementar “lugar” como um atributo multi-valor, dado que este apenas existe quando associado a uma entidade.

Um ponto fraco do nosso trabalho é a inexistência de uma vista exclusiva para a administração da CE. Esta limitação faz com que seja difícil manter o controlo das inserções efetuadas sobre qualquer tabela (Viagens, Comboios, Estações). Sendo um dos papéis principais de um administrador de bases de dados a realização de uma boa gestão de todas as informações nela contidas, este papel complica-se quando não existem métodos que facilitam o acesso aos dados que não podem ser acedidos pelo utilizador comum.

Um dos pontos fortes do nosso trabalho foi a introdução de entidades que nos permitissem generalizar o problema em questão e simultaneamente garantissem a possibilidade de incluir, toda a informação referente à empresa ferroviária.

Outro aspeto que é importante mencionar do nosso trabalho assenta no sucesso do desenvolvimento de um modelo abrangente, que possibilita uma capacidade de adaptação elevada na eventualidade de necessidade de introdução de novos dados, ou alteração de dados já existentes. Podemos finalmente aferir que dada esta flexibilidade, será possível assegurar uma maior longevidade para todo o sistema de base de dados, diminuindo até os seus custos de manutenção, atualização e necessidade de implementação de uma base de dados de raiz.

Referências

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

Lista de Siglas e Acrónimos

Siglas e acrónimos utilizados durante a realização do trabalho.

BD	Base de Dados
CE	Comboios Europeus
1FN	Primeira Forma normal
2FN	Segunda Forma normal
3FN	Terceira Forma normal
FE	<i>Forward Engineering</i>

Anexos

I. Inserir Comboios

```
INSERT INTO comboio (ID, Observacoes)
VALUES (1, 'Comboio com bar. Atinge velocidades na ordem dos 200 km/h'),
       (2, 'Comboio de luxo com restaurante, bar e casa de banho. Atinge velocidades de 300 km/h'),
       (3, 'Comboio simples, recomendado para viagens nacionais. Atinge velocidades de 150 km/h');
```

II. Inserir Estações

```
INSERT INTO estacao (ID, Nome, Cidade)
VALUES (1, 'Oriente', 'Lisboa'),
       (2, 'Atocha', 'Madrid'),
       (3, 'Sants', 'Barcelona'),
       (4, 'Saint Jean', 'Bordeaux'),
       (5, 'Part Dieu', 'Lyon'),
       (6, 'Gare de Lyon', 'Paris');
```

III. Inserir Lugares dos Comboios

```
INSERT INTO lugarcomboio (NumeroLugar, Comboio_ID)
VALUES (1, 2),
       (2, 2),
       (3, 2),
       (4, 2),
       (5, 2),
       (6, 2),
       (7, 2),
       (8, 2),
       (9, 2),
       (10, 2),
       (11, 2),
       (12, 2),
       (13, 2),
       (14, 2),
       (15, 2);

INSERT INTO lugarcomboio (NumeroLugar, Comboio_ID)
VALUES (1, 1),
       (2, 1),
       (3, 1),
       (4, 1),
       (5, 1),
       (6, 1),
       (7, 1),
       (8, 1),
       (9, 1),
       (10, 1);

INSERT INTO lugarcomboio (NumeroLugar, Comboio_ID)
VALUES (1, 3),
       (2, 3),
       (3, 3),
       (4, 3),
       (5, 3);
```

IV. Inserir Viagens

```
INSERT INTO viagem (ID, Comboio_ID, HoraPartida, Origem, Destino, HoraChegada)
VALUES (1, 1, '07:00', 1, 2, '11:45'),
       (2, 3, '12:30', 2, 3, '17:00'),
       (3, 2, '18:00', 3, 5, '22:30');
```