

Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores

Reservas Online - Relatório da 1ª Fase

47192 : Alexandre Marques da Silva (a47192@alunos.isel.pt)

47220 : João Francisco Fernandes da Silva Nunes (a47220@alunos.isel.pt)

Relatório para a Unidade Curricular de Sistemas de Informação 1
da Licenciatura em Engenharia Informática e de Computadores

Professora : Doutora Matilde Pós-de-Mina Pato

Resumo

O relatório da primeira fase do primeiro trabalho da unidade curricular de Sistemas de Informação 1 é constituído por uma breve introdução aos conceitos de reservas *online* nomeadamente de transportes públicos e de “esquemas relacionais”, o qual também tem um pedaço de história anexado. O desenvolvimento do mesmo contém uma solução de um esquema relacional para um site de reservas de viagens. É desenvolvida por análise em termos de chaves, restrições, dependências funcionais e utilização de normalizações até à terceira forma normal (no qual reduz a redundância de dados e aumenta a integridade dos mesmos). Por último, é tecida uma conclusão que estende os pontos feitos em capítulos anteriores e é feita uma especulação sobre possíveis usos práticos de um esquema relacional no futuro.

Abstract

The report of the first work phase for the Information Systems I course unit consists in a brief introduction to the concepts of online reservations namely public transports and *relational schemes*, which also implies a piece of history. The development of this report contains a solution based in a relational scheme for a website of online reservations. It is by analysis in terms of keys, restrictions, functional dependencies and use of normalizations up to the third normal form (which reduces data redundancy and increases data integrity). Finally, a conclusion is drawn that extends the points made in previous chapters and a speculation is also made about the possible practical uses of a relational scheme in the future.

Índice

Lista de Figuras	ix
Lista de Tabelas	xi
1 Introdução	1
1.1 Contexto	1
1.2 Objetivos	1
2 Desenvolvimento	3
2.1 Características	3
2.2 Análise	3
2.3 Modelo Lógico	4
2.4 Dependências Funcionais	7
2.5 Normalização	8
3 Conclusão	11
3.1 Observações	11
3.2 Conhecimento	11
Referências	13
A Esquema Final Exemplificativo	i

Lista de Figuras

A.1	Esquema Travel4Me	i
-----	-----------------------------	---

Lista de Tabelas

2.1	Relação PASSAGEIRO	4
2.2	Relação RESERVA	5
2.3	Relação VIAGEM	5
2.4	Relação ESTACAO	5
2.5	Relação LOCALIDADE	5
2.6	Relação AUTOCARRO	6
2.7	Relação COMBOIO	6
2.8	Relação LOCOMOTIVA	6
2.9	Relação CARRUAGEM	6
2.10	Relação LUGAR	6
2.11	Relação TIPO_LUGAR	7
2.12	Relação VIAGEM_RESERVA	9
2.13	Relação PAGAMENTO	9
2.14	Relação TIPO_AUTOCARRO	9
2.15	Relação LUGAR_RESERVADO	9



Introdução

1.1 Contexto

Reservas Online Estas surgem para facilitar o cliente e a empresa a realizar reservas de forma muito conveniente, permitindo ao utilizador a partir de casa, trabalho, onde quer que seja, verificar a disponibilidade para reservar e garantir a sua vaga. Deste modo, permite planear férias ou viagens de forma antecipada, rápida e sem a imposição de falar com outra pessoa, ignorando assim muitos dos inconvenientes usuais e economizar tempo.

Modelo Relacional Este conceito foi criado por Edgar Frank Codd em 1970, sendo descrito no artigo “Relational Model of Data for Large Shared Data Banks”. Defini-se teoricamente e logicamente através de predicados e da teoria de conjuntos e baseia-se nos conceitos de entidade e relação. O mesmo veio a substituir antigos modelos de bases de dados, como o modelo hierárquico e o modelo em rede[1].

1.2 Objetivos

Realização de um modelo relacional com base num sistema de reservas online respeitando, portanto, as suas regras e características, e apresentando, os seus respetivos esquemas de relação, chaves, restrições e normalização até à sua terceira forma normal(3NF)[2] sem perdas de dependências funcionais.



Desenvolvimento

2.1 Características

Este presente trabalho relata um sistema de gestão de reservas de viagens realizadas de comboio e/ou de autocarro. Para realizar uma viagem é necessário reservar pelo menos um lugar disponível, o qual pertence a uma classe. Para concluir a reserva terá que efetuar o pagamento, confirmando assim a viagem realizada entre estações.

2.2 Análise

Numa primeira leitura ao enunciado, extrai-se informação necessária para construir as relações bases. Estas referem-se a esquemas de relação muito simples e sem aplicando todas as regras de um modelo relacional, contendo apenas informação básica como: o **nome da relação**, a **chave primária** e alguns **atributos**. Este método estabelece uma *forte raiz nesta árvore de relações* e, por sua vez, facilita o futuro desenvolvimento da solução ao problema. Aplicando este processo, encontram-se as seguintes relações:

PASSAGEIRO (número de identificação, nome, e-mail, data de nascimento, género)

RESERVA (identificador, número de passageiros, origem, destino, preço total, meio de transporte, data de reserva, hora da reserva, bilhete)

VIAGEM (identificador, data, hora de partida, hora de chegada, distância, estação de origem, estação de chegada, meio de transporte)

ESTAÇÃO (nome, tipo, número de plataformas, localidade)

LOCALIDADE (código-postal, nome)

AUTOCARRO (matrícula, velocidade máxima, data de entrada, número de quilómetros, data da próxima revisão, tipo, marca, modelo, número de lugares, lugares)

COMBOIO (locomotiva, carruagens, tipo, número de carruagens, lugares)

LOCOMOTIVA (número de série, marca, velocidade máxima, data de entrada ao serviço, número de quilómetros)

CARRUAGEM (número, classe, número total de lugares sentados)

LUGAR (número, tipo)

TIPO_LUGAR (identificador, classe, preço)

2.3 Modelo Lógico

Após obter todo o tipo de informação e inserir em relações simples, tratamos de identificar mais pormenores, nos quais podem ser: PK¹, AK², FK³, o domínio dos atributos e as restrições de integridade (negócio, coluna, etc). Os nomes dos atributos são abreviados e escolhidos de forma lógica e, se necessário, no formato *lower CamelCase*. As restrições e o domínio é informação previamente fornecida, caso contrário, atribui-se o valor mais correto.

Tabela 2.1: Relação **PASSAGEIRO**

<i>Atributo</i>	<i>Domínio</i>	<i>Restrição</i>	
[PK] nid	int	Número do cartão de cidadão ou do passaporte	
nome	varchar	Obrigatório	
email	varchar	Obrigatório	
dataNascimento	datetime	Formato "DD-MM-AAAA"	Obrigatório
genero	varchar	Valores: 'M', 'F' ou 'N/R'	

¹PK - Primary Key(Chave primária)

²AK - Alternate Key(Chave candidata)

³FK - Foreign Key(Chave estrangeira)

Tabela 2.2: Relação **RESERVA**

Atributo	Domínio	Restrição	
[PK] id	int	Inteiro positivo	
nrPassageiros	int		
[FK] origem	varchar	origem→ESTACAO.nome	Local de origem diferente de destino
[FK] destino	varchar	destino→ESTACAO.nome	
precoTotal	decimal	Em euros	
tipoTransporte	varchar	Valores: "autocarro"ou "comboio"	
[FK] viagens	int	Referencia VIAGEM.id	
data	datetime	Formato: "DD-MM-AAAA"	
hora	timestamp	Formato: "hh:mm"	
ticketid	int		
pagamento	varchar	Valores: "MB", "MB Way", "Pay Pal" ou "Cartão de crédito"	
nrTelemovel	varchar	Se o pagamento for MB Way, este atributo é obrigatório.	

Tabela 2.3: Relação **VIAGEM**

<i>Atributo</i>	<i>Domínio</i>	<i>Restrição</i>	
[PK] id	int		
data	datetime	Formato: "DD-MM-AAAA"	
horaPartida	timestamp	Formato: "hh:mm"	Hora de chegada é superior à de partida
horaChegada	datetime		
distancia	int	Valor em km (quilómetros)	
[FK] estacaoPartida	varchar	Refere ESTACAO.nome	Estação de chegada é diferente da saída
[FK] estacaoChegada	varchar	Refere ESTACAO.nome	
[FK] transporte	varchar	Refere que transporte é utilizado (comboio ou autocarro)	

Tabela 2.4: Relação **ESTACAO**

<i>Atributos</i>	<i>Domínio</i>	<i>Restrição</i>
[PK] nome	varchar	
tipo	varchar	Valores: "terminal" ou "paragem"
nrPlataformas	int	
[FK] localidade	varchar	localidade→LOCALIDADE.codigopostal

Tabela 2.5: Relação **LOCALIDADE**

<i>Atributos</i>	<i>Domínio</i>	<i>Restrição</i>
[PK] codigopostal	varchar	
nome	varchar	

Tabela 2.6: Relação **AUTOCARRO**

<i>Atributo</i>	<i>Domínio</i>	<i>Restrição</i>
[PK] matricula	varchar	
veloMaxima	int	Valor em km/h (quilómetros por hora)
dataEntrada	datetime	
nrKm	int	Valor em km (quilómetros)
dataProxRevisao	datetime	
tipo	varchar	
marca	varchar	
modelo	varchar	
nrLugares	int	
[FK] lugares	int	Referencia LUGAR.numero

Tabela 2.7: Relação **COMBOIO**

<i>Atributo</i>	<i>Domínio</i>	<i>Restrição</i>
[PK][FK] locomotiva	int	locomotiva→LOCOMOTIVA.nrSerie
[PK][FK] carruagens	int	Referencia CARRUAGEM.numero
nrCarruagens	int	
tipo	varchar	Valores: "alfa-pendular - AP", "inter-cidades - IC", "inter-regionais - IR"ou "regionais - R"
[FK] lugares	int	Referencia LUGAR.numero

Tabela 2.8: Relação **LOCOMOTIVA**

<i>Atributo</i>	<i>Domínio</i>	<i>Restrição</i>
[PK] nrSerie	int	
marca	varchar	
veloMaxima	int	Valor em km/h (quilómetros por hora)
dataEntrada	datetime	
nrKm	int	

Tabela 2.9: Relação **CARRUAGEM**

<i>Atributo</i>	<i>Domínio</i>	<i>Restrição</i>
[PK] numero	varchar	
classe	varchar	Valores: "conforto"ou "turística"
nrLugares	int	

Tabela 2.10: Relação **LUGAR**

<i>Atributo</i>	<i>Domínio</i>	<i>Restrição</i>
[PK] numero	int	
[FK] tipo	int	tipo→TIPO_LUGAR.id

Tabela 2.11: Relação TIPO_LUGAR

<i>Atributo</i>	<i>Domínio</i>	<i>Restrição</i>
[PK] id	int	
classe	varchar	
preco	decimal	

2.4 Dependências Funcionais

As dependências funcionais são obtidas a partir de análise do texto, sentido crítico e conhecimento comum. Tanto podem ser avaliadas a partir da sua importância em relação aos atributos participantes como podem ser mera observação do mundo real. A interpretação desta última avaliação pode ser alvo de discussão de como atribuir a melhor relação entre atributos.

Conjunto DF's de:

- **PASSAGEIRO** Chave determina funcionalmente todos os atributos.
- **RESERVA**
 - F1 = id → {nid, nrPassageiros, origem, destino, precoTotal, tipoTransporte, viagens, data, hora, ticketid}
 - F2 = ticketid → {pagamento, nrTelemovel}
 - F3 = viagens → precoTotal
- **VIAGEM**
 - F1 = id → estacaoPartida, estacaoChegada, distancia, data, horaPartida, horaChegada, transporte
 - F2 = estacaoPartida, estacaoChegada → distancia
- **ESTACAO** Chave determina funcionalmente todos os atributos.
- **LOCALIDADE** Chave determina funcionalmente todos os atributos.
- **AUTOCARRO**
 - F1 = matricula → dataEntrada, nrKm, dataProxRevisao, marca, modelo, veloMaxima, lugares, tipo
 - F2 = tipo → marca, modelo, veloMaxima, nrLugares
- **COMBOIO**
 - F1 = locomotiva, carruagens → nrCarruagens, tipo
 - F2 = carruagens → lugares

- **LOCOMOTIVA** Chave determina funcionalmente todos os atributos.
- **CARRUAGEM** Chave determina funcionalmente todos os atributos.
- **LUGAR** Chave determina funcionalmente todos os atributos.
- **TIPO_LUGAR**
F1 = id → classe, preco
F2 = classe → preco

2.5 Normalização

A partir das DF's acima, é possível normalizar as relações de modo a obter um bom esquema relacional, minimizando a redundância de dados criando um forte relacionamento entre os mesmos. A explicação seguinte não tem como exemplos todas as relações, mas sim as de caráter mais importante, introduzindo o processo utilizado.

1FN Todos os atributos têm de ser atômicos, isto é, decompõe-se os atributos compostos/multivalor.

Exemplo: A relação **RESERVA** não se encontra nesta forma porque existe um atributo multi-valor (*viagens*). Para resolver, cria-se uma nova relação **RESERVA_VIAGEM**, com os atributos idReserva (refere RESERVA.id) e idViagem (refere VIAGEM.id) como PK's. Por fim, verifica-se atomicidade na relação.

2FN Para estar nesta forma normal, é necessário estar na 1FN. Para além disso, todos os atributos não-chave tem que depender da totalidade da chave.

Exemplo: Na relação **COMBOIO** temos uma chave composta (locomotiva e carruagens). A partir das suas DF's, os atributos não dependem da totalidade desta chave. A solução encontrada foi a seguinte: cada carruagem pertence a um comboio, logo a relação **CARRUAGEM** passa a ter o atributo que identifica o comboio (COMBOIO.id). As restantes já pertencem ao caso especial da 2NF: a PK consistir num único atributo.

3NF Tem que estar na 2NF e não pode ter atributos não primos⁴ com dependências transitivas. Isto significa que, se existir um atributo que é determinado transitivamente (observar DF's) na relação, esta não está na 3FN.

Exemplo 1: A relação **RESERVA** contém os atributos pagamento e nrTelemovel. Estes

⁴atributo primo - atributo que pertence a alguma chave candidata

atributos são determinados funcionalmente por *ticketid*. E *ticketid* é determinado funcionalmente pela PK da relação (RESERVA.id). Esta dependência é transitiva. A solução implica criar uma nova relação, **PAGAMENTO**, que tem como PK o *ticketid* e os atributos não-chave *metodo* e *nrTelemovel*.

Exemplo 2: Aplica-se o mesmo processo para a relação **AUTOCARRO**. Uma vez que a *marca*, *modelo*, *veloMaxima* e *nrLugares* são determinados transitivamente. A chave *matricula* não os determina funcionalmente mas sim o atributo *tipo*. Cria-se uma relação **TIPO_AUTOCARRO** com *tipo* como PK e este como FK para a relação inicial.

Conclusão Aplicando o raciocínio anterior para todas as relações conclui-se:

Tabela 2.12: Relação **VIAGEM_RESERVA**

<i>Atributo</i>	<i>Domínio</i>	<i>Restrição</i>
[PK][FK] idReserva	int	idReserva → RESERVA.id
[PK][FK] idViagem	int	idViagem → VIAGEM.id
preco	decimal	Valor em euros

Esta relação contém também o atributo *preco* que representa o preço de uma só viagem. O *precoTotal* em **RESERVA** é a soma de todos estes preços.

Tabela 2.13: Relação **PAGAMENTO**

<i>Atributo</i>	<i>Domínio</i>	<i>Restrição</i>
[PK] ticketid	int	
metodo	varchar	Valores: "MB", "MB Way", "Pay Pal" ou "Cartão de crédito"
nrTelemovel	varchar	Se o pagamento for MB Way, este atributo é obrigatório

Tabela 2.14: Relação **TIPO_AUTOCARRO**

<i>Atributo</i>	<i>Domínio</i>	<i>Restrição</i>
[PK] id	int	
marca	varchar	
modelo	varchar	
veloMaxima	int	Valor em km/h (quilómetros por hora)
nrLugares	int	

Tabela 2.15: Relação **LUGAR_RESERVADO**

<i>Atributo</i>	<i>Domínio</i>	<i>Restrição</i>
[PK][FK] numero	int	numero → LUGAR.numero
[PK][FK] viagem	int	viagem → VIAGEM.id
[FK] transporte	varchar	Refere COMBOIO.id ou AUTOCARRO.matricula



Conclusão

Este sistema de reservas *online* fica concluído com um esquema relacional desenvolvido até à 3ª forma normal. Garante-se também a não perda de dependências funcionais entre os atributos. Isto facilita o desenvolvimento de uma SGBD¹, mais tarde, em Microsoft SQL Server[3].

3.1 Observações

Um dos aspetos a ter em conta é o facto de existirem várias soluções para o problema enunciado. A solução que o relatório reporta tem em conta somente a interação e conversa (*brainstorming*) entre os colegas do grupo. Esta é justificada de acordo com as regras do Modelo Relacional com o objetivo final de reduzir a redundância de dados, aumentar a integridade de dados e o desempenho.

3.2 Conhecimento

O estudo desta geração de SGBD serviu para uma melhor abordagem ao tema de Sistemas de Informação. O uso de uma nova plataforma (L^AT_EX)[4] foi algo desafiante para o grupo mas tem em conta uma aprendizagem contínua o que poderá contribuir para um trabalho mais fluído no futuro.

¹Sistema de Gestão de Base de Dados

Referências

- [1] Edgar Frank Codd, *A Relational Model Of Data for Large Shared Data Banks*. ACM, 1970.
- [2] Edgar Frank Codd & Thomas J. Watson, *Further Normalization of the Data Base Relational Model*. IBM Research Center, 1971, Presented at Courant Computer Science Symposia Series 6.
- [3] Microsoft Corporation. (2019). "Sql server 2019", URL: <https://www.microsoft.com/sql-server/sql-server-2019/>.
- [4] Leslie Lamport, *LaTeX: A Document Preparation System*. Addison-Wesley, 1986.



Esquema Final Exemplificativo

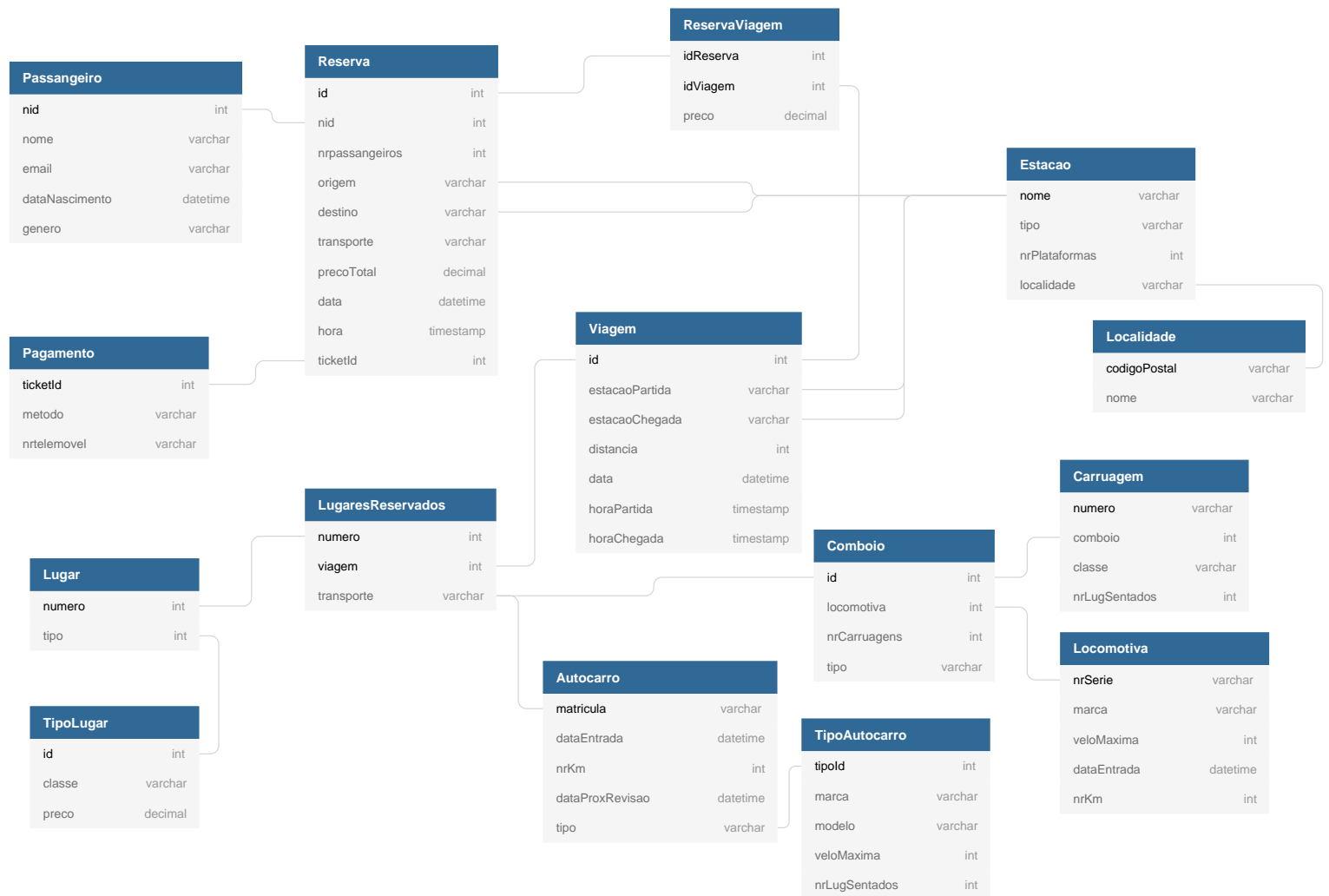


Figura A.1: Esquema Travel4Me

