5.1

5.2

To represent the design of the base relations, we use an extended form of the

DBDL to define domains, default values, and null indicators.

Domínio ID Cliente: inteiro  
Domínio Nome: string de comprimento variável, comprimento 50  
Domínio Email: string de comprimento variável, comprimento 30  
Domínio Número Identificação Fiscal: inteiro  
Domínio Password: string de comprimento variável, comprimento 18

Cliente(  
 id\_cliente ID Cliente NOT NULL,  
 nome Nome NOT NULL,  
 email Email NOT NULL,  
 nif Número Identificação Fiscal NOT NULL,  
 password Password NOT NULL,  
 CHAVE PRIMÁRIA (id\_bilhete)  
 CHAVE ALTERNATIVA (nif));

Domínio ID Bilhete: inteiro  
Domínio Preço: valor monetário, entre 000.00 – 999.99  
Domínio Data Aquisição: temporal  
Domínio Classe: caracter (‘P’ ou ‘E’)  
Domínio Número: inteiro  
Domínio Cliente: inteiro  
Domínio Viagem: inteiro

Bilhete(  
 id\_bilhete ID Bilhete NOT NULL,  
 preco Preço NOT NULL,  
 data\_aquisicao Data Aquisição NOT NULL,  
 classe Classe NOT NULL,  
 numero Número NOT NULL,  
 cliente Cliente NOT NULL,  
 viagem Viagem NOT NULL,  
 CHAVE PRIMÁRIA (id\_bilhete),  
 CHAVE ESTRANGEIRA (cliente) REFERENCIA Cliente(id\_cliente)   
 ON UPDATE CASCADE  
 ON DELETE NO ACTION,  
 CHAVE ESTRANGEIRA (viagem) REFERENCIA Viagem(id\_viagem)   
 ON UPDATE CASCADE  
 ON DELETE NO ACTION);

Domínio ID Viagem: inteiro  
Domínio Data Partida: temporal  
Domínio Data Chegada: temporal  
Domínio Duração: temporal  
Domínio Preço Base: valor monetário, entre 000.00 – 999.99  
Domínio Comboio: inteiro  
Domínio Origem: inteiro  
Domínio Destino: inteiro

Viagem(  
 id\_viagem ID Viagem NOT NULL,  
 data\_partida Data Partida NOT NULL,  
 data\_chegada Data Chegada NOT NULL,  
 duração Duração NOT NULL,  
 preco\_base Preço Base NOT NULL,  
 comboio Comboio NOT NULL,  
 origem Origem NOT NULL,  
 destino Destino NOT NULL  
 CHAVE PRIMÁRIA (id\_viagem),  
 CHAVE ESTRANGEIRA (comboio) REFERENCIA Comboio(id\_comboio)   
 ON UPDATE CASCADE  
 ON DELETE NO ACTION,  
 CHAVE ESTRANGEIRA (origem) REFERENCIA Estacao(id\_estacao)   
 ON UPDATE CASCADE  
 ON DELETE NO ACTION,

CHAVE ESTRANGEIRA (destino) REFERENCIA Estacao(id\_estacao)   
 ON UPDATE CASCADE  
 ON DELETE NO ACTION);

Domínio ID Estação inteiro  
Domínio Nome string de comprimento variável, comprimento 20

Estacao(  
 id\_estacao ID Estação NOT NULL,  
 nome Nome NOT NULL,  
 CHAVE PRIMÁRIA (id\_estacao));

Domínio ID Comboio inteiro

Comboio(  
 id\_comboio ID Comboio NOT NULL,  
 CHAVE PRIMÁRIA (id\_comboio));

Domínio Classe: carater (‘P’ ou ‘E’)  
Domínio Numero: inteiro  
Domínio Comboio: inteiro

Lugar(  
 classe Classe NOT NULL,  
 numero Número NOT NULL,  
 comboio Comboio NOT NULL,  
 CHAVE PRIMÁRIA (classe, numero, comboio),  
 CHAVE ESTRANGEIRA (comboio) REFERENCIA Comboio(id\_comboio)   
 ON UPDATE CASCADE  
 ON DELETE NO ACTION);

**Representação da construção dos atributos derivados**

O cálculo dos atributos derivados presentes no modelo bla bla bla:

* **classe** e **numero** (Bilhete) – ambos os atributos são passados como argumento aquando da inserção de um bilhete na respetiva tabela. Estes argumentos são valores existentes na tabela Lugar.
* **preco** (Bilhete) – o cálculo do preço é efetuado sempre que é inserida uma entrada na tabela Bilhete através do seguinte *trigger*:  
  
* **duracao** (Viagem) – apenas é feita a diferença entre Viagem(data\_chegada) e Viagem(data\_partida) no momento que é introduzida uma nova viagem na tabela  
  

Todos eles são apenas calculados uma vez e guardados, pelo que não existe overhead relativo à sua atualização.

5.3

Escolher prints

5.4

Escolher prints

5.6

Para precaver qualquer cenário possível, a análise feita ao crescimento da base de dados tem por base a análise do seu pior caso, ou seja, assumindo que todas as viagens realizadas têm lotação esgotada.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Relação** | **Atributo** | **Data Type** | **Tamanho** | **Total** |
| Cliente | id\_cliente | INT | 4 bytes | 106 bytes |
|  | nome | VARCHAR(50) | 50 bytes |
|  | email | VARCHAR(30) | 30 bytes |
|  | nif | INT | 4 bytes |
|  | password | VARCHAR(18) | 18 bytes |
| Bilhete | id\_bilhete | INT | 4 bytes | 27 bytes |
| preco | FLOAT(5, 2) | 4 bytes |
| data\_aquisicao | DATETIME | 8 bytes |
| classe | CHAR(1) | 1 byte |
| numero | SMALLINT | 2 bytes |
| cliente | INT | 4 bytes |
| viagem | INT | 4 bytes |
| Viagem | id\_viagem | INT | 4 bytes | 39 bytes |
| data\_partida | DATETIME | 8 bytes |
| data\_chegada | DATETIME | 8 bytes |
| duração | TIME | 3 bytes |
| preco\_base | FLOAT(5, 2) | 4 bytes |
| comboio | INT | 4 bytes |
| origem | INT | 4 bytes |
| destino | INT | 4 bytes |
| Comboio | id\_comboio | INT | 4 bytes | 4 bytes |
| Estação | id\_estacao | INT | 4 bytes | 24 bytes |
| nome | VARCHAR(20) | 20 bytes |
| Lugar | classe | CHAR(1) | 1 byte | 7 bytes |
| numero | SMALLINT | 2 bytes |
| comboio | INT | 4 bytes |

Apesar da entidade Cliente ser a que maior quantidade de memória ocupa por entrada na respetiva tabela, não será a que ocupará mais no sistema. A quantidade de dados mais significativa será gerada pela entidade Bilhete, dado que será uma entrada para cada lugar em cada viagem.

Pelo Anexo 1, podemos verificar que existem 72 viagens diárias, ocupando estas 72\*39 = **2808 bytes/dia** no sistema, enquanto que os 250 lugares (50 premium + 200 económico) em cada um dos comboios ocupam um total (250\*7) \* 6 = **10500 bytes**. Posto isto, a cada dia serão adicionadas 72\*250 = 18000 entradas na tabela Bilhete, perfazendo um total de 18000\*27 = **486000 bytes/dia**. Somando tudo e prevendo um registo de 50000 clientes por ano teremos então 50000\*106 + 486000\*365 + 2808\*365 + 10500 = **175,2 MBytes/ano**.

Cliente:

O cliente é a entidade que, no pior caso, ocupa mais memória por cada entrada na tabela. Contudo não será a entidade a ocupar mais espaço no sistema.

Bilhete:

Viagem:

O total de viagens efetuadas num dia é 72 (ver anexo x). Assim sendo, por dia serão guardados na base de dados 72\*39 = 2808 bytes.

Comboio:

Como apenas é guardado o identificador do comboio, cada comboio ocupará 4 bytes de memória.

Lugar:

Cada comboio tem um total de 250 lugares (50 premium + 200 económico). Como cada entrada na base de dados ocupa 7 bytes, serão guardados 250\*7 = 1750 bytes por cada comboio.

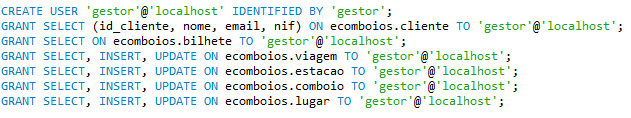
5.8

Para regular o acesso à base de dados de modo a evitar qualquer tipo de fraude ou ameaça para o sistema foram criados três perfis de utilização da base de dados: administrador, gestor e utilizador.

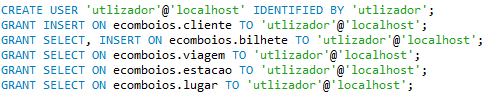
O administrador não tem qualquer tipo de restrição podendo, assim, realizar qualquer operação sobre a base de dados ou aceder a qualquer informação nela guardada.



A função do gestor é regular e assegurar a existência de viagens assim como de comboios. Ao adicionar um novo comboio tem também de assegurar que esse comboio tem os respetivos lugares. O gestor tem também a capacidade de adicionar novas estações.



O utilizador pode efetuar o registo, visualizar as viagens e lugares existentes, assim como as estações. Para além disso tem como principal função a compra de bilhetes.



ANEXO 1 -> HORARIO DIARIO

Braga <-> Porto

|  |  |  |  |
| --- | --- | --- | --- |
| **Braga** | **Porto** | **Porto** | **Braga** |
| **Hora Partida** | **Hora Chegada** | **Hora Partida** | **Hora Chegada** |
| 00:00:00 | 00:20:00 | 00:10:00 | 00:30:00 |
| 07:00:00 | 07:20:00 | 07:10:00 | 07:30:00 |
| 08:00:00 | 08:20:00 | 08:10:00 | 08:30:00 |
| 09:00:00 | 09:20:00 | 09:10:00 | 09:30:00 |
| 10:00:00 | 10:20:00 | 10:10:00 | 10:30:00 |
| 11:00:00 | 11:20:00 | 11:10:00 | 11:30:00 |
| 12:00:00 | 12:20:00 | 12:10:00 | 12:30:00 |
| 13:00:00 | 13:20:00 | 13:10:00 | 13:30:00 |
| 14:00:00 | 14:20:00 | 14:10:00 | 14:30:00 |
| 15:00:00 | 15:20:00 | 15:10:00 | 15:30:00 |
| 16:00:00 | 16:20:00 | 16:10:00 | 16:30:00 |
| 17:00:00 | 17:20:00 | 17:10:00 | 17:30:00 |
| 18:00:00 | 18:20:00 | 18:10:00 | 18:30:00 |
| 19:00:00 | 19:20:00 | 19:10:00 | 19:30:00 |
| 20:00:00 | 20:20:00 | 20:10:00 | 20:30:00 |
| 21:00:00 | 21:20:00 | 21:10:00 | 21:30:00 |
| 22:00:00 | 22:20:00 | 22:10:00 | 22:30:00 |
| 23:00:00 | 23:20:00 | 23:10:00 | 23:30:00 |

Porto <-> Lisboa

|  |  |  |  |
| --- | --- | --- | --- |
| **Porto** | **Lisboa** | **Lisboa** | **Porto** |
| **Hora Partida** | **Hora Chegada** | **Hora Partida** | **Hora Chegada** |
| 01:00:00 | 02:25:00 | 01:10:00 | 02:55:00 |
| 07:00:00 | 08:25:00 | 07:10:00 | 08:55:00 |
| 09:00:00 | 10:25:00 | 09:10:00 | 10:55:00 |
| 11:00:00 | 12:25:00 | 11:10:00 | 12:55:00 |
| 13:00:00 | 14:25:00 | 13:10:00 | 14:55:00 |
| 15:00:00 | 16:25:00 | 15:10:00 | 16:55:00 |
| 17:00:00 | 18:25:00 | 17:10:00 | 18:55:00 |
| 19:00:00 | 20:25:00 | 19:10:00 | 20:55:00 |
| 21:00:00 | 22:25:00 | 21:10:00 | 22:55:00 |
| 23:00:00 | 00:25:00 | 23:10:00 | 00:55:00 |

Braga <-> Lisboa

|  |  |  |  |
| --- | --- | --- | --- |
| **Braga** | **Lisboa** | **Lisboa** | **Braga** |
| **Hora Partida** | **Hora Chegada** | **Hora Partida** | **Hora Chegada** |
| 08:10:00 | 09:55:00 | 08:10:00 | 09:55:00 |
| 10:10:00 | 11:55:00 | 10:10:00 | 11:55:00 |
| 12:10:00 | 13:55:00 | 12:10:00 | 13:55:00 |
| 14:10:00 | 15:55:00 | 14:10:00 | 15:55:00 |
| 16:10:00 | 17:55:00 | 16:10:00 | 17:55:00 |
| 18:10:00 | 19:55:00 | 18:10:00 | 19:55:00 |
| 20:10:00 | 21:55:00 | 20:10:00 | 21:55:00 |
| 22:10:00 | 23:55:00 | 22:10:00 | 23:55:00 |