Imprimir

Relatório Final

Departamento de Engenharia Informática e de Sistemas

[nome do curso]

Bases de Dados

2022/2023

Tema

Gestão de aeroportos

Identificação dos Autores

Autores:

Nº Aluno	Prática	Nome	Email
2020139445	P5	Tânia Guedes	a2020139445@isec.pt
2018014484	P3	José Antunes	a21280272@isec.pt
2019130219	P7	Emanuel Saraiva	a2019130219@isec.pt
2019138441	P5	Duarte Gomez	a2019138441@isec.pt

A comunicação entre os elementos do grupo é maioritariamente efetuada pelo serviço “Discord”. Durante a sessão de trabalho um dos membros do grupo (podem ser mais), realiza a partilha do ecrã do seu computador e todos, em conversa, colaboram para a realização do trabalho. Após cada sessão, o membro do grupo que realizou a partilha envia o ficheiro do trabalho feito para os restantes membros, para que todos estejam sincronizados.

Distribuição do esforço:

Tarefa	Tânia Guedes	José Antunes	Emanuel Saraiva	Duarte Gomez
Análise da Situação	25%	25%	25%	25%
Restantes Tarefas	25%	25%	25%	25%

Cap 1 - Introdução

Em Portugal, a gestão de aeroportos é feita por uma empresa privada, a *Ana*.
Devido à falta de concorrência, e após avaliação do mercado, decidimos abrir uma empresa capaz de gerir aeroportos de forma a melhorar diversos aspectos negativos que ocorrem hoje em dia, com o objetivo de num futuro próximo sermos a "Nr. 1" em gestão dos mesmos.

Cap 2 - Enquadramento da Proposta (Objetivo)

Para gerir vários aeroportos, é necessário saber como é que estes são estruturados e quais as informações que têm de ser geridas. Desta forma, os aeroportos dividem-se em três partes, a primeira, a parte interna, a qual é constituída pelos funcionários que trabalham no aeroporto e pelos passageiros que vão viajar. Mas, para que estes possam fazê-lo, é necessário que adquiram bilhetes (os quais podem ser comprados no aeroporto em questão ou online) e que estes sejam válidos até ao momento do check-in, podendo a compra ser cancelada a qualquer momento. Cada Passageiro pode fazer várias compras (de bilhetes) para vários voos. É também preciso saber quais os dados de quem vai viajar e de quem trabalha no aeroporto (morada, nome, email, telefone, data de nascimento, cartão de cidadão e nif), para fins de proteção e segurança.

A segunda parte, a parte externa, é a que engloba toda a informação sobre a localização dos parques de estacionamento privados, isto é, se o parque está cheio ou não, quais os movimentos de entrada e saída do parque, a data em que esses movimentos foram feitos e qual o valor a ser pago com base nessa diferença. É necessário saber também qual a localização do aeroporto em questão, qual o seu nome e quantos funcionários lá trabalham. Cada aeroporto pode ter um conjunto de parques de estacionamento privados, mas cada parque só pode pertencer a um aeroporto.

A terceira e última parte, é a parte correspondente ao tráfego aéreo, onde as informações dos vários voos têm de ser controladas pela torre de controlo, a qual só pode existir uma em cada aeroporto, de modo a conseguir ser mostrado nos televisores do mesmo o número, o destino e a origem de cada voo disponível no momento, a porta de embarque e o terminal onde apanhar o voo. Se o voo tiver escala deve ser disponibilizada igualmente essa informação. Em caso de atrasos e/ou mudanças das portas de embarque, a Torre de controlo será encarregue de atualizar essa informação, a qual será disponibilizada de imediato nos televisores. Para os voos serem realizados, é necessário que tenham havido compras de bilhetes para os mesmos e também tem de haver um avião que esteja encarregue daquele voo, sem avião não há meio de transporte. Cada avião é composto pelo seu nome, tipo e por quantos lugares tem, de modo a que se saiba se o avião está cheio ou não. Este realiza vários voos e os diversos voos têm de ser também efetuados por vários aviões. Cada companhia aérea fornece os seus próprios aviões ao aeroporto de modo a que este tenha aviões suficientes para a realização dos diversos voos. Cada companhia tem uma sigla própria, um nome e a localização da sede principal da empresa.

Desta forma conseguimos gerir todas as informações necessárias para o bom funcionamento de vários aeroportos.

Sec 2.1 - Diagnóstico da Situação Atual

Atualmente, a gestão de aeroportos é feita por uma empresa privada, a ANA, a qual é responsável pela gestão de 10 aeroportos em Portugal Continental.

Esta empresa tem como principal objetivo explorar o serviço público aeroportuário de apoio à aviação civil em Portugal.

Para além disso, asseguram ainda a exploração de espaços comerciais e publicitários nos aeroportos, a oferta de imóveis (ligados à operação aeroportuária, edifícios comerciais e hotéis), parques de estacionamento e serviços de rent-a-car (designados de negócios não aviação).

Há umas semanas, a nossa empresa recebeu um pedido que visa mostrar outras maneiras de gerir de forma mais rápida e eficaz um ou mais aeroportos, e visto que na nossa zona de residência (Coimbra) não há nenhum aeroporto, decidimos realizar este pedido fazendo uma simulação, com um protótipo para um possível aeroporto a ser construído nesta zona.

Sec 2.2 - Problemas encontrados / Possíveis Melhorias

Após termos feito uma vistoria à situação atual, podemos constatar a existência de alguns problemas correlacionados com uma má gestão da *ANA*.

Neste tópico, serão apresentados todos os problemas e suas respetivas melhorias para o melhoramento da gestão do aeroporto em questão, e também para o próximo aeroporto que venha a ser construído na zona centro.

Problemas:

- Falta de informação sobre os aviões;
- Sistema disfuncional dos parques de estacionamento privados;
- Atraso nas informações dos voos;
- Problema nas emissões de bilhetes.

Melhorias:

- A nossa empresa pretende gerir toda a informação necessária sobre os aviões de forma automatizada, ou seja, há um bom balanceamento entre as várias informações, fazendo assim com que toda a gente que entra no aeroporto fique com todas as informações necessárias pretendidas;
- Em relação aos parques de estacionamento privados, teremos a capacidade de gerir funcionalmente as entradas e saídas, o valor a ser pago com base nessa diferença, o tipo de pagamento que foi efetuado, a data de acesso ao parque, quantos estacionamentos cada parque tem, os tipos de parques de estacionamento existentes, ou seja se é coberto ou ao ar livre, a sua localização, o seu horário, isto é, se é aberto 24h, quanto tempo se demora a chegar ao aeroporto a pé, e também o preço associado a cada parque;
- A informação sobre os voos é das mais importantes, senão a mais importante a ter no que diz respeito à gestão de um ou mais aeroportos. Assim, planeamos gerir esta informação de uma forma mais rápida e eficaz, conseguindo fazer com que as informações destes sejam atualizadas o mais rápido possível, devido à forma como é tratado este tipo de informação;
- Sobre os bilhetes a nossa empresa tenciona registar cada compra que foi realizada, tendo esta sido feita dentro do aeroporto ou online. Caso o autor da compra decida fazer o cancelamento da mesma, é possível ficar com essa informação guardada, entre outros dados, por exemplo o nome na fatura, o tipo de bilhete comprado, isto é, se é de primeira classe, se é classe económica, qual o lugar do passageiro no avião, qual o número do voo, a data em que o bilhete foi comprado, a sua validade, qual a data do voo, o tipo de pagamento realizado, se foi por multibanco ou por cartão de crédito, o código de barras do bilhete para poder ser feita a verificação no momento do check-in, e por último se o passageiro entrou realmente no avião.

Sec 2.3 - Descrição da Solução Proposta - Pormenorizada

A solução a ser implementada, usará um sistema de SGBD baseada em Oracle com vários clientes e pelo menos um servidor principal.

A Base de Dados a implementar, deverá ter permissões de leitura e escrita, contendo pelo menos 9 entidades, com 9 relações entre si, cerca de 80 atributos, dos quais simples, compostos, derivados e de multivalor.

Para um eficaz funcionamento da BD, deverão ser implementadas as 9 entidades e respetivas relações entre as mesmas, com as diversas funções entre si, da seguinte forma:

Entidades:

- **Aeroportos** - id do aeroporto [PK], nome [Atributo Simples], morada (código-postal, número, cidade e rua) [Atributo Composto], número de funcionários [Atributo Derivado] e número de estacionamentos [Atributo Derivado];

A nossa gestão está feita de forma a gerir um ou mais aeroportos. Desta forma esta entidade permite guardar todas as informações úteis para cada um.

- **Companhias Aéreas** - id da companhia [PK], nome [Atributo Simples], contrato [Atributo Simples], sigla [Atributo Simples], número de aviões [Atributo Derivado] e localização da sede da empresa [Atributo Multivalor];

Para o aeroporto funcionar não só é necessário ter trabalhadores, como é necessário que tenha os meios de transporte para a realização de voos. Os aviões pertencem às companhias, logo, na nossa opinião, é importante que tenhamos esta entidade ligada ao aeroporto, para sabermos: "este aeroporto em especifico, tem x companhias, o contrato é este e a companhia está neste sitio". Assim, caso haja algum problema, as informações estão todas guardadas.

- **Pessoas** - nome (apelido e nome próprio) [Atributo Composto], e-mail [Atributo Simples], cartão de cidadão [Atributo Simples], data de nascimento [Atributo Simples], telefone [Atributo Multivalor], nif [Atributo Simples], morada (código-postal, número, cidade e rua) [Atributo Composto] e passaporte [Atributo Simples];

Foi necessário criar esta entidade "mãe/base", devido ao facto de as entidades Passageiros e Funcionários, terem muitos atributos iguais. Assim, estas herdam estes atributos e apenas é necessário escrevê-los uma vez.

- **Funcionários** - id do funcionário [PK], função [Atributo Simples], contrato [Atributo Simples], gerente [Role] e trabalhador [Role];

Achámos por bem criar um relacionamento trabalha-para e ligá-lo a esta entidade, pois dentro do aeroporto, há vários setores, e dentro de cada setor há vários cargos, e como há muitos funcionários, deve existir pelo menos um gerente em cada sector, de modo a que haja uma boa organização dentro de cada estabelecimento.

Esta entidade é derivada da entidade Pessoas e serve para guardar todos os trabalhadores do aeroporto, com base na sua função e no contrato que têm com o mesmo. A sua existência é importante, pois cada aeroporto tem de saber os dados todos de quem lá trabalha.

- **Passageiros** - id do passageiro [PK] e patologias [Atributo Simples];

É importante saber os dados de quem vai no avião e de quem faz as compras dos bilhetes. De igual forma é muito importante saber também, se os passageiros levam ou não malas para o porão e se têm alguma partologia que pode implicar problemas quando lhes for servido comida no decorrer do voo.

Esta entidade é derivada da Entidade Pessoas.

- **Voos** - id do voo [PK], número de voo [Atributo Simples], origem [Atributo Simples], destino [Atributo Simples], hora de chegada [Atributo Simples], hora de partida [Atributo Simples], número do terminal [Atributo Simples], porta de embarque [Atributo Simples], preço-base do voo [Atributo Simples], data semanal do voo [Atributo Simples], escala [Atributo Simples], local da escala [Atributo Simples];

Esta entidade é muito importante, pois é aquela que permite às pessoas saber que voos é que existem no momento (o seu numero, origem, destino, etc). Esta informação é nos muito útil, pois é mostrada e atualizada nos televisores do aeroporto, bem como nos websites das compras de bilhetes, para que no realizar da compra, o autor saiba qual a data do voo, o terminal, as horas, a porta de embarque, qual o destino e a origem, o preço, se tem escala e se tiver, qual a escala do voo. Sem esta informação, o aeroporto passa a ser "inexistente".

- **Aviões** - id do avião [PK], nome [Atributo Simples], tipo do avião [Atributo Simples], número de lugares [Atributo Simples];

Com a esta entidade conseguimos saber quais são os aviões que estão a fazer "x" voos. Para além de que ao ter o atributo número de lugares, consoante o avião, conseguimos saber quantos lugares "aquele" avião em específico tem, de modo a saber se ainda é possível ou não comprar bilhetes para aquele voo. Caso o número de bilhetes comprados para aquele determinado voo, for igual ao número de lugares do avião que vai fazer esse mesmo voo, já não é possível comprar bilhetes. Mais uma vez estamos perante informações imprescindíveis sobre uma boa gestão de um ou mais aeroportos.

- **Torres de Controlo** - id da torre de controlo [PK], número de setores [Atributo Simples], número de funcionários [Atributo Simples], localização [Atributo Multivalor] e tipo de funcionamento [Atributo Simples];

A torre de controlo controla todo o tráfego aéreo até uma determinada altitude e distância. Esta entidade permite-nos registar diversar torres associadas a diversos aeroportos (sempre um para um, isto é, cada aeroporto, só pode ter uma torre de controlo e cada torre só pode pertencer a um aeroporto) com o objetivo de controlar os voos e os aviões. Caso seja necessário mudar o lugar onde o avião virá a estacionar, a torre será capaz de registar e controlar essa informação. O mesmo para as horas de entrada, saída e portas de embarque, caso haja algum imprevisto, esta conseguirá atualizar a informação de modo a ser mostrada às pessoas o mais depressa possível. Cada torre é composta por vários setores e cada setor precisa de ter um determinado número de funcionários a trabalhar lá. Porém, as torres não são todas iguais, há umas que têm mais setores do que outras. O mesmo para a sua localização, isto é, dependendo de onde a torre está localizada, o seu tipo de funcionamento vai variar entre ter um horário que é contínuo e um horário que é mais específico.

- **Parques de Estacionamento Privados** - id do parque de estacionamento [PK], número de lugares [Atributo Simples], tipo de parque de estacionamento [Atributo Simples], localização [Atributo Multivalor], preço-por-hora [Atributo Simples], tempo que as pessoas demoram a ir a pé até ao aeroporto [Atributo Simples] e horário de funcionamento [Atributo Simples].

Esta entidade é nos útil, pois cada aeroporto pode ter um ou mais parques de estacionamento privativos. Caso tenha, conseguimos saber quantos lugares temos ao todo para saber se este já está cheio ou não e consequentemente disponibilizar essa informação tanto aos funcionários como aos passageiros. Dependendo da sua localização, do tipo de parque e da distância até ao aeroporto, o preço vai ser diferente e por conseguinte registado, para posteriormente conseguirmos saber quanto é que cada pessoa que entrou no parque tem de pagar.

Relacionamentos:

- **Relacionamento compram** - tipo de bilhete [Atributo Simples], validade [Atributo Simples], tipo da compra (online ou no próprio estabelecimento) [Atributo Simples], nome na fatura [Atributo Simples], tipo de pagamento [Atributo Simples], lugar do passageiro [Atributo Simples], data da emissão do bilhete [Atributo Simples], cancelado (em caso de cancelamento) [Atributo Simples], verificação de entrada no avião [Atributo Derivado], número do voo [Atributo Derivado], código de barras do bilhete [Atributo Simples], data do voo para o qual o bilhete foi comprado [Atributo Simples] e mala no porão [Atributo Simples];

Este relacionamento engloba duas entidades: Passageiros e Voos com uma relação N para M respectivamente, com obrigatoriedade do lado dos passageiros, porque para os voos se realizarem têm de existir obrigatoriamente passageiros. No entanto os passageiros não são obrigados a comprar voos.

Neste relacionamento sempre que é feita uma compra de um bilhete fica registado na tabela desta relação todos os dados associados a essa compra. Exemplo: "comprei um bilhete hoje (dataEmissaoBilheteComprado), para o voo Nr. X (numVoo), o qual corresponde ao dia 2/2/2023".

- **Relacionamento controlam** - lugar onde o avião estaciona [Atributo Simples], porta de embarque alterada [Atributo Simples], hora de partida alterada [Atributo Simples] e hora de chegada alterada [Atributo Simples];

Este relacionamento engloba duas entidades: Torres de Controlo e Voos com uma relação 1 para N respectivamente, com obrigatoriedade do lado das torres e do lado dos voos, porque os voos têm de ser obrigatoriamente controlados pelas torres de controlo e as torres de controlo têm de controlar obrigatoriamente os voos, caso contrário, estes não podiam sequer existir e seria uma desordem muito grande.

É necessário que haja uma terceira tabela de relacionamento, pois os dados podem ser alterados ao longo do tempo, isto é, devido a imprevistos, tanto a hora de chegada como a de partida, como a porta de embarque e o lugar onde o avião vai estacionar, podem mudar. Ao mudar, estes dados são registados e os televisores do aeroporto vão buscar estas novas informações de modo a não perderem os voos.

- **Relacionamento estacionou** - hora de saída do parque [Atributo Simples], hora de entrada no parque [Atributo Simples], tipo de pagamento [Atributo Simples], data [Atributo Simples] e valor pago [Atributo Simples];

Este relacionamento é composto por duas entidades: Pessoas e Parques de estacionamento Privados com uma relação N para M respectivamente. Nesta relação não existem quaisquer obrigatoriedades, pois as pessoas não são obrigadas a estacionar naqueles parques. Do mesmo modo, os parques de estacionamento não precisam de ser estacionados obrigatoriamente pelas mesmas.

Nós criámos esta relação, pois seria prudente e necessário guardar os movimentos de entradas e saídas nos parques, de modo a termos vários conhecimentos, por exemplo o mês em que os parques estão mais cheios. Sendo também possível registar o valor pago com base na diferença do tempo lá estado e no preço por hora do estacionamento em questão. É ainda possível registar o tipo de pagamento que foi feito.

- **Aeroportos empregam Funcionários** - Este relacionamento engloba duas entidades: Aeroportos e Funcionários com uma relação 1 para N respectivamente, com obrigatoriedade do lado dos funcionários, porque sem funcionários o aeroporto não pode ser aberto e muito menos estar em desenvolvimento. Assim, este é obrigado a empregar funcionários para poder funcionar correctamente;

- **Aeroportos dispõem de Parques de estacionamento Privados** - Este relacionamento engloba duas entidades: Aeroportos e Parques de estacionamento Privados com uma relação 1 para N respectivamente, sem qualquer tipo de obrigatoriedade, pois não é obrigatório o aeroporto ter parques de estacionamento privados, nem é obrigatório que os parques de estacionamento privados sejam dispostos por aeroportos, podem ser dispostos por exemplo, por condominios de prédios;

- **Aeroportos possuem Torres de Controlo** - Este relacionamento engloba duas entidades: Aeroportos e Torres de Controlo com uma relação 1 para 1 respectivamente, com obrigatoriedade dos dois lados, porque para o aeroporto funcionar, este precisa de ter uma torre de controlo que lhe permite controlar o tráfego aéreo e todas as informações relativas aos voos e aviões, e as torres têm de estar situadas obrigatóriamente dentro do aeroporto, não faz sentido estarem noutro sítio que não no aeroporto;

- **Companhias Aéreas operam nos Aeroportos** - Este relacionamento engloba duas entidades: Aeroportos e Companhias Aéreas com uma relação N para M respectivamente, com obrigatoriedade do lado das companhias, porque sem aviões, os quais são fornecidos (pertencem às companhias) pelas companhias, não adianta o aeroporto estar aberto e estar a vender bilhetes de voos, quando não há meios de transporte para os realizar, logo é necessário que este seja operado obrigatoriamente pelas companhias. Por outro lado, não é da obrigatoriedade das companhias operar em aeroportos, estas podem operar noutros sítios;

- **Companhias Aéreas têm Aviões** - Este relacionamento engloba duas entidades: Companhias Aéreas e Aviões com uma relação 1 para N respectivamente, com obrigatoriedade do lado dos aviões, porque se a companhia não tem aviões, então não faz sentido ela estar associada a quaisquer aeroportos. Assim, é obrigatório que esta possua aviões, de modo a ser útil ao aeroporto e lucrativo para si. Por outro lado, os aviões só podem pertencer a uma companhia, mas não é obrigatório, ou seja, podem pertencer por exemplo ao estado;

- **Aviões efetuam Voos** - Este relacionamento engloba duas entidades: Voos e Aviões com uma relação N para M respectivamente, com obrigatoriedade dos dois lados, pois não há outra maneira para os voos serem realizados se não forem feitos pelos aviões, e os aviões se não fizerem aquilo para os quais foram construídos, que é fazer voos, não dá para estes serem realizados e muito menos para que o aeroporto seja activo. Um avião pode efetuar vários voos e um voo pode ser efetuado por vários aviões.

Cap 3 - Análise de Dados

Neste capítulo pretende-se descrever detalhadamente todas as entidades e relacionamentos envolvidos na solução proposta.

O modelo conceptual é enviado em formato PDF.

Sec 3.1 - Entidades

Nesta secção vão ser descritas todas as entidades relevantes para a gestão de um ou mais aeroportos. Após uma análise aprofundada do modelo de gestão de um possível aeroporto a ser construído na zona centro, constatou-se a necessidade das seguintes entidades:

- Aeroportos;
- Companhias Aéreas;
- Pessoas;
- Funcionários;
- Passageiros;
- Voos;
- Aviões;
- Torres de Controlo;
- Parques de Estacionamento Privado.

Sec 3.1.1 - Entidade Aeroportos

A entidade aeroportos representa a informação relativa a um ou mais aeroportos. É inserido um novo registo nesta entidade sempre que é aberto um novo aeroporto. A nossa gestão está feita de forma a gerir um ou mais aeroportos. Desta forma esta entidade permite guardar todas as informações úteis para cada um.

Atributos:

Nome do atributo	Tipo de Dados	Descrição

idAeroporto [PK]	Integer	Id (único) de cada aeroporto. Ex: Para 5 aeroportos, cada um terá o seu próprio id: 1; 2; 3; 4 e 5.
codigoPostal	Variable characters (12)	Código Postal do aeroporto. Tem no máximo 12 caracteres. Ex: 4580-345.
cidade	Variable characters (40)	Cidade onde está o aeroporto. Tem no máximo 40 caracteres. Ex: Porto.
rua	Variable characters (128)	Nome da rua onde o aeroporto se encontra. Tem no máximo 128 caracteres. Ex: Rua Andrade de Sousa.
numero	Integer	Numero da morada onde está o aeroporto. Ex: 42.
nome	Variable characters (40)	Nome do aeroporto. Tem no máximo 40 caracteres. Ex: Francisco Sá Carneiro.

Restrições dos atributos:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações

idAeroporto	N	S	Identificador (chave primária), não admite nulos. Não existem dois ou mais aeroportos com o mesmo código, logo tem valores únicos.
codigoPostal	N	N	Não admite valores nulos, uma vez que é necessário saber a localização exata do aeroporto. Não são valores únicos, pois duas artérias (vias) podem ter o mesmo código postal.
cidade	N	N	Não admite valores nulos, uma vez que é necessário saber a cidade onde se localiza o aeroporto. Não tem valores únicos, uma vez que os aeroportos podem estar na mesma cidade.
rua	N	N	Não admite valores nulos, uma vez que é necessário saber a rua onde se localiza o aeroporto. Não tem valores únicos, pois os aeroportos podem estar situados na mesma rua. Em Portugal não acontece, mas pode acontecer fora, e assim temos como gerir a situação.
numero	N	N	Não admite valores nulos, uma vez que é necessário saber o número respectivo da morada do aeroporto. Não tem valores únicos uma vez que existem aeroportos com o mesmo número.
nome	N	S	Não admite valores nulos, uma vez que é necessário saber o nome do aeroporto. Há valores únicos, pois cada aeroporto tem um nome único.

Relacionamentos:

Nome do relacionamento	Cardinalidade	Entidade relacionada	Participação obrigatória
Empregam	1:N	Funcionarios	Funcionarios
Dispoem	1:N	ParquesEstacionamentoPrivado	-

Operam	N:M	CompanhiasAereas	CompanhiasAereas
Possuem	1:1	TorresControlo	TorresControlo e Aeroporto

Sec 3.1.2 - Entidade Companhias Aéreas

A entidade Companhias Aéreas representa a informação relativa às companhias que operam num aeroporto. Para o aeroporto é necessário ter os meios de transporte para a realização de voos. os aviões, os quais pertencem às companhias, logo, na nossa opinião, é importante que tenhamos esta entidade ligada ao aeroporto, para sabermos: "este aeroporto em especifico, tem x companhias, o contrato é este e a companhia está neste sitio". Assim, caso haja algum problema, as informações estão todas guardadas.

Atributos:

Nome do atributo	Tipo de Dados	Descrição
idCompanhia [PK]	Integer	Id (único) de cada companhia. Ex: Para 3 companhias, cada uma terá o seu próprio id: 1; 2; e 3.
sigla	Variable characters (7)	Sigla da companhia, escrito obrigatoriamente em maiúsculas. Tem no máximo 7 caracteres. **Nota:** não é obrigatório que a companhia tenha um sigla. Ex: TAP
nome	Variable characters (40)	Nome da companhia. Tem no máximo 40 caracteres. Ex: Transporte Áereo Português.

contrato	Long variable characters (800)	Contrato da companhia aérea com o aeroporto. Tem no máximo 800 caracteres. Ex: Este contracto visa a integridade de aviões no...
localizaçãoDaSedePrincipal	Long characters (1024)	Localização da sede principal da companhia aérea. Pode vir em texto, com a morada, ou em coordenadas. Tem no máximo 1024 caracteres. Ex: Rua António Humberto Delgado, nº39, 7º Esquerdo, ou "38° 43' N 9° 09' O".

Restrições dos atributos:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
idCompanhia	N	S	Identificador (chave primária), não admite nulos. Não existem duas ou mais companhias com o mesmo id.
sigla	S	S	Admite nulos, pois há empresas/companhias que podem não ter sigla. Não existem duas companhias com a mesma sigla.
nome	N	S	Não admite valores nulos, uma vez que é necessário saber o nome da companhia aérea. Há valores únicos uma vez que cada companhia tem um nome único, logo terá uma sigla única.

contrato	N	N	Não admite valores nulos, uma vez que é necessário saber qual o contrato que diz respeito a cada companhia aérea. Não tem valores únicos, pois pode haver contractos iguais.
localizaçãoDaSedePrincipal	N	S	Não admite valores nulos porque é importante ter acesso à localização da sede principal da companhia aérea e admite valores únicos pois cada sede é de localização única.

Relacionamentos:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
Operam	N:M	Aeroportos	CompanhiasAereas
Tem	1:N	Avioes	Avioes

Sec 3.1.3 - Entidade ParquesEstacionamentoPrivado

Esta entidade é nos útil, pois cada aeroporto pode ter um ou mais parques de estacionamento privativos. Caso tenha, conseguimos saber quantos lugares temos ao todo para saber se este já está cheio ou não e consequentemente disponibilizar essa informação tanto aos funcionários como aos passageiros. Dependendo da sua localização, do tipo de parque e da distância até ao aeroporto, o preço vai ser diferente e por conseguinte registado, para posteriormente conseguirmos saber quanto é que cada pessoa que entrou no parque tem de pagar.

Atributos:

Nome do atributo	Tipo de Dados	Descrição
idParqueEstacionamento [PK]	Integer	Id (único) de cada parque de estacionamento. Ex: Para 2 parques, cada uma terá o seu próprio id: 1; e 2.

numLugares	Integer	Indica o número de lugares existentes no parque de estacionamento. Ex: 30.
localizacao	Long variable characters (1024)	Localização do parque. Pode vir em texto, com a morada, ou em coordenadas. Tem no máximo 1024 caracteres. Ex: Rua António Humberto Delgado, nº39, 7º Esquerdo, ou "38° 43' N 9° 09' O".
tipoParqueEstacionamento	Variable characters (20)	Tipo de parque de estacionamento. Tem no máximo 20 caracteres. Ex: Coberto, ao ar livre.
horario	Variable characters (40)	Horário de funcionamento do parque. Tem 40 caracteres. Ex: Aberto 24h.
precoPorHora	Integer	Preço a pagar por hora. Ex: 5 euros.
tempoIrAPeAteAoAeroporto	Integer	Indica a estimativa do tempo que se demora a ir a pé do parque de estacionamento ao aeroporto. Ex: 2 minutos.

Restrições dos atributos:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
idEstacionamento	N	S	Identificador (chave primária), não admite nulos. Não existem dois ou mais parques de estacionamento com o mesmo id.
numLugares	N	N	Não admite valores nulos, uma vez que é necessário saber o número de lugares de estacionamento do aeroporto. Não tem valores únicos, uma vez que pode acontecer que hajam parques que tenham o mesmo número de lugares.
localizacao	N	S	Não admite valores nulos, uma vez que é necessário saber onde se localizam os parques. Admite valores únicos uma vez que cada parque de estacionamento tem a sua localização própria.
tipoParqueEstacionamento	N	N	Não admite nulos, pois é necessário saber qual o tipo de parque. Não tem valores únicos uma vez que pode acontecer que hajam parques que sejam do mesmo tipo.
horario	S	N	Pode admitir valores nulos, caso tenha null é porque não há informação sobre o horário do parque, se ele está aberto ou não 24h e também não tem valores únicos, pois o horário pode ser o mesmo para diversos parques.
precoPorHora	N	N	Não admite nulos, isto é, tem de se saber qual o preço por hora de cada parque com base nas suas caracteristicas, e não tem valores únicos uma vez que cada pessoa poderá pagar a mesma quantia consoante o tempo de utilização do parque e do preço do mesmo.

tempoIrAPeAteAoAeroporto	N	N	Não admite nulos, pois é informação necessária. Não admite valores únicos pois o tempo a ir do parque ao aeroporto pode ser o mesmo entre os diversos parques e aeroportos das bases de dados.

Relacionamentos:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
Estacionou	N:M	Pessoas	-
Dispoem	N:1	Aeroportos	-

Sec 3.1.4 - Entidade TorresControlo

A torre de controlo controla todo o tráfego aéreo até uma determinada altitude e distância. Esta entidade permite-nos registar diversar torres associadas a diversos aeroportos (sempre um para um, isto é, cada aeroporto, só pode ter uma torre de controlo e cada torre só pode pertencer a um aeroporto), com o objetivo de controlar os voos e os aviões. Caso seja necessário mudar o lugar onde o avião virá a estacionar, a torre será capaz de registar e controlar essa informação. O mesmo para as horas de entrada, saída e portas de embarque, caso haja algum imprevisto, esta conseguirá atualizar a informação de modo a ser mostrada às pessoas o mais depressa possível. Cada torre é composta por vários setores e cada setor precisa de ter um determinado número de funcionários a trabalhar lá. Porém, as torres não são todas iguais, há umas que têm mais setores do que outras. O mesmo para a sua localização, isto é, dependendo de onde a torre está localizada, o seu tipo de funcionamento vai variar entre ter um horário que é contínuo e um horário que é mais específico.

Atributos:

Nome do atributo	Tipo de Dados	Descrição

idTorreControlo [PK]	Integer	Id (único) de cada torre de controlo. Ex: Para 3 torres associadas a diferentes aeroportos, pois cada torre só pode pertencer a um aeroporto, e esse aeroporto, só pode ter essa torre associadas, cada uma terá o seu próprio id: 1; 2; e 3.
numSetores	Integer	Campo que indica o número de setores existentes na torre de controlo. Ex: 12.
localizacao	Long variable characters (1024)	Localização da torre. Pode vir em texto, com a morada, ou em coordenadas. Tem no máximo 1024 caracteres. Ex: Rua António Humberto Delgado, nº39, 7º Esquerdo, ou "38° 43' N 9° 09' O".
numFuncionariosPosto	Integer	Indica o número de funcionário que está a trabalhar no posto da torre de controlo. Ex: 20.
tipoFuncionamento	Characters (30)	Indica a forma como funciona a torre de controlo, com base na sua localização. Tem no máximo 30 caracteres. Ex: Horário contínuo, ou seja está sempre em funcionamento 24/7, ou só trabalha de x em x horas.

Restrições dos atributos:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
idTorreControlo	N	S	Identificador (chave primária), não admite nulos. Não existem duas ou mais torres com o mesmo id.
numSetores	N	N	Não admite nulos uma vez que é importante saber quantos setores existem para as diversas torres. Também não admite valores únicos uma vez que podem existir torres que tenham o mesmo número de setores.
localizacao	N	S	Não admite nulos pois é importante saber a localização da torre de controlo. Tem um valor único porque a cada torre diz respeito uma e uma só localização.
numFuncionariosPosto	S	N	Admite valores nulos, pois não é informação que seja muito importante saber, e caso em algum momento não esteja lá ninguém a trabalhar, se o campo estiver a null então é isso que significa. Não é um valor único uma vez que cada torre poderá ter o mesmo número de funcionários em cada posto.
tipoFuncionamento	N	N	Não admite nulos, ou seja, é necessário saber qual o tipo de funcionamento da torre e não há valores únicos, pois o tipo de funcionamento pode ser o mesmo para outras torres.

Relacionamentos:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
Possuem	1:1	Aeroportos	Ambas as entidades
Controlam	1:N	Voos	Ambas as entidades

Sec 3.1.5 - Entidade Avioes

Com esta entidade conseguimos saber quais são os aviões que estão a fazer "x" voos. Para além de que ao ter o atributo número de lugares, consoante o avião, conseguimos saber quantos lugares "aquele" avião em específico tem, de modo a saber se ainda é possível ou não comprar bilhetes para aquele voo. Caso o número de bilhetes comprados para aquele determinado voo, for igual ao número de lugares do avião que vai fazer esse mesmo voo, já não é possível comprar bilhetes. Mais uma vez estamos perante informações imprescindíveis sobre uma boa gestão de um ou mais aeroportos.

Atributos:

Nome do atributo	Tipo de Dados	Descrição
idAviao [PK]	Integer	Id (único) de cada avião. Ex: Para 3 aviões, cada um terá o seu próprio id: 1; 2; e 3.
nome	Variable characters (40)	Nome do avião. Tem no máximo 40 caracteres. Ex: A320.
tipoAviao	Variable characters (50)	Indica o tipo do avião. Tem no máximo 50 caracteres. Ex: Comercial.
numLugares	Integer	Indica o número de lugares que cada avião disponibiliza. Ex: 120.

Restrições:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
idAviao	N	S	Identificador (chave primária), não admite nulos. Não existem dois ou mais aviões com o mesmo id.
nome	N	N	Não admite nulos, pois é importante saber os nomes dos diversos aviões e não admite nulos, pois há vários aviões que têm o mesmo nome.
tipoAviao	N	N	Não admite nulos, mais uma vez é informação necessária e não admite valores únicos, pois há aviões do mesmo tipo.
numLugares	N	N	Não admito nulos, pois temos mesmo de saber isso e não tem valores únicos, pois há aviões que têm o mesmo número de lugares

Relacionamentos:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
Efectuam	N:M	Voos	Ambas as entidades
Tem	N:1	CompanhiasAereas	Avioes

Sec 3.1.6 - Entidade Voos

Esta entidade é muito importante, pois é aquela que permite às pessoas saber que voos é que existem no momento (o seu numero, origem, destino, etc). Esta informação é nos muito útil, pois é mostrada e atualizada nos televisores do aeroporto, bem como nos websites das compras de bilhetes, para que no realizar da compra, o autor saiba qual a data do voo, o terminal, as horas, a porta de embarque, qual o destino e a origem, o preço, se tem escala, e se tiver, qual a escala do voo. Sem esta informação, o aeroporto passa a ser "inexistente".

Atributos:

Nome do atributo	Tipo de Dados	Descrição
idVoo	Integer	Id (único) de cada voo. Ex: Para 3 voos, cada um terá o seu próprio id: 1; 2; e 3.
horaPartida	Date & Time	Hora de partida de um determinado voo. Ex: 2023-12-10 12:30:00.
destino	Variable characters (50)	Destino de um determinado voo. Tem no máximo 50 caracteres. Ex: Berlim.
localEscala	Variable characters (50)	Nome do local onde um determinado voo pode efetuar escala. Tem no máximo 50 caracteres. Ex: Madrid.
origem	Variable characters (50)	Origem de um determinado voo. Tem no máximo 50 caracteres. Ex: Porto.
escala	Integer	Indica se um determinado voo vai efetuar escala ou não. Ex: 0 - Não há escala. 1 - Faz uma escala; 2 - Faz duas escalas.
precoVooBase	Integer	Preço de um voo. Ex: 173 euros.

dataSemana	Date	Data do voo. Ex: 18/18/2023.
numTerminal	Integer	Número do terminal um determinado voo. Ex: 1.
horaChegada	Date & Time	Hora de chegada de um determinado voo. Ex: 2023-12-18 17:30:00.
portaEmbarque	Variable characters (5)	Número da porta de embarque de um determinado voo. Tem no máximo 5 caracteres. Ex: 3AA.
numVoo	Variable characters (10)	Número do voo. Tem no máximo 10 caracteres. Ex: TAP9520.

Restrições dos atributos:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
idVoo	N	S	Identificador (chave primária), não admite nulos. Não existem dois ou mais voos com o mesmo id.
horaPartida	N	N	Não nulo, pois temos mesmo de saber. Considerando um hora de partida para um determinado voo, pode ocorrer uma outra partida de um outro voo, na mesma hora.
destino	N	N	Não admite nulos, pois é informação necessária. Vários voos podem ter o mesmo destino.

localEscala	S	N	Admite valores nulos. Caso o campo esteja a null assume-se que o voo não tem escala. Voos diferentes podem ter coincidentemente o mesmo lugar para efetuar uma escala.
origem	N	N	Não admite nulos, pois mais uma vez temos de saber as origens dos voos. Voos diferentes podem ter o mesmo lugar de origem, daá, não admitir valores únicos.
escala	N	N	Não admite nulo, ou seja, se não fizer escala, então é porque tem 0 escalas. Esta informação pode se repetir, logo não admite valores únicos.
precoVooBase	N	N	Não nulo, temos de ter esta informação. Não há valores únicos, pois o preço pode ser o mesmo para outros voos.
dataSemana	N	N	Não nulo, cada voo tem uma data. Para a mesma data da semana pode-se efetuar vários voos.
numVoo	N	S	Não nulo, é imprescindível registar o número de cada voo. Admite valores únicos, pois cada voo tem o seu próprio número. Não pode haver números repetidos.
numTerminal	N	N	Não nulo, é informação necessária. Não admite valores únicos, pois o terminal pode ser o mesmo para diversos voos.
horaChegada	N	N	Não nulo, temos de saber a hora de Chegada de cada voo. Considerando um hora de chegada para um determinado voo, pode ocorrer uma outra chegada de um outro voo, na mesma hora.
portaEmbarque	N	N	Não nulo, a porta de embarque tem de estar registada. Não admite nulos, pois a porta de embarque pode se repetir para outros voos.

Relacionamentos:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
Efetuam	N:M	Avioes	Ambas as entidades
Compram	N:M	Passageiros	Passageiros
Controlam	N:1	TorreControlo	Ambas as entidades

Sec 3.1. 7- Entidade Pessoas

Foi necessário criar esta entidade "mãe/base", devido ao facto de as entidades
Passageiros e Funcionários , terem muitos atributos iguais. Assim, estas herdam estes atributos e apenas é necessário escrevê-los uma vez.

Atributos:

Nome do atributo	Tipo de Dados	Descrição
idPessoa	Integer	Id (único) de cada pessoa. Ex: Para 2 pessoas, cada uma terá o seu próprio id: 1 e 2.
email	Variable characters (100)	Email de cada pessoa. Tem no máximo 100 caracteres. Ex: xpto@gmail.com.

cartaoCidadao	Variable Characters (12)	Número do cartão de cidadão da pessoa. Tem no máximo 12 caracteres. Ex: 4568507 x ZO6.
dataNascimento	Date	Data de nascimento da pessoa. Ex: 29/08/1987.
telefone	Integer	Número de telefone da pessoa Ex: 912345678.
nomeProprio	Variable characters (40)	Nome próprio da pessoa. Tem no máximo 40 caracteres. Ex: Maria.
nif	Integer	Número de contribuinte da pessoa. Ex: 123123123.
apelido	Variable characters (40)	Apelido da pessoa. Tem no máximo 40 caracteres. Ex: Santos.
codigoPostal	Variable characters (12)	Código-postal da zona onde a pessoa vive. Tem no máximo 12 caracteres. Ex: 4589-785.
cidade	Variable characters (40)	Cidade onde vive a pessoa. Tem no máximo 40 caracteres. Ex: Coimbra.

rua	Variable characters (128)	Rua onde a pessoa mora. Tem no máximo 128 caracteres. Ex: Rua da Sofia.
numero	Integer	Número da porta da casa, e caso a pessoa habite num apartamento é indicado o andar também Ex: 42.
passaporte	Variable Characters (10)	Número do passaporte da pessoa. Ex: 232 232 323

Restrições dos atributos:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
idPessoa	N	S	Identificador (chave primária), não admite nulos. Não existem duas ou mais pessoas com o mesmo id.
nomeProprio	N	N	Não admite nulos, cada pessoa tem de ter um nome próprio, podendo este ser o mesmo para outras pessoas - admite valores únicos.
apelido	S	N	Admite nulos, não é informação obrigatória. A pessoa pode apenas colocar o seu nome próprio. O apelido pode se repetir, logo são não tem valores únicos.
nif	N	S	Não admite valores nulos, é informação mesmo necessária e os valores são diferentes de pessoa para pessoa, ou seja, são únicos.
codigoPostal	N	N	Não admite nulos, o lugar onde cada pessoa mora é sempre identificado por um código postal. Pode acontecer várias pessoas terem o mesmo código postal, logo não tem valores únicos.

cidade	N	N	Não admite nulos, o lugar onde cada pessoa mora acarreta sempre uma cidade. Pode acontecer várias pessoas morarem em lugares diferentes mas que se situam dentro da mesma cidade, assim não admite valores únicos.
rua	S	N	Admite nulos, pois pode acontecer uma casa situar-se num lugar onde não esteja atribuído um nome à rua. Várias pessoas podem habitar na mesma rua, logo a rua não é única.
numero	S	N	Admite nulos, não é informação obrigatória, pois uma casa, que seja habitada por uma pessoa, pode não possuir um número. Caso esteja atribuído um número à mesma, esta pode não ter um valor único, pois em regiões diferentes podem existir casas com o mesmo número atribuído.
telefone	S	S	Admite nulos, uma pessoa não é obrigada a possuir um número de telefone. Caso possua, o número de telefone será único para cada pessoa.
dataNascimento	N	N	Não admite nulos. Todas as pessoas possuem uma data de nascimento. Várias pessoas podem nascer no mesmo dia.
cartaoCidadao	S	S	Admite nulos uma vez que cidadãos estrangeiros, que não pertencem à UE, não possuem cartão de cidadão. Cartões de cidadão têm valor único, não se repetem.
email	S	S	Admite nulos. Cada pessoa pode ter mais que um e-mail, se assim o entender. No entanto, uma pessoa não é obrigada a possuir um e-mail. Não há emails iguais.
passaporte	S	S	Admite nulos, uma vez que cidadãos nacionais podem não possuir passaporte para viajar na União Europeia. Não há passaportes iguais.

Relacionamentos:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
Estacionou	N:M	ParquesEstacionamentoPrivado	-

Sec 3.1. 8- Entidade Funcionarios

Achámos por bem criar um relacionamento trabalha-para e ligá-lo a esta entidade, pois dentro do aeroporto, há vários setores, e dentro de cada setor há vários cargos, e como há muitos funcionários, deve existir pelo menos um gerente em cada sector, de modo a que haja uma boa organização dentro de cada estabelecimento. Esta entidade é derivada da entidade Pessoas e serve para guardar todos os trabalhadores do aeroporto, com base na sua função e no contrato que têm com o mesmo. A sua existência é importante, pois cada aeroporto tem de saber os dados todos de quem lá trabalha, e também tem de ter funcionários, caso contrário, este não poderia estar em funcionamento.

Atributos:

Nome do atributo	Tipo de Dados	Descrição
idFuncionario	Integer	Id (único) de cada funcionário. Ex: Para 3 funcionários, cada um terá o seu próprio id: 1; 2; e 3.
contrato	Long Variable characters (800)	Campo que especifica o tipo de contrato que cada funcionário tem com o aeroporto. Tem no máximo 800 caracteres. Ex: Este contracto visa a integridade de funcionários no...
funcao	Variable characters (50)	Função do funcionário dentro do aeroporto. Tem no máximo 50 caracteres. Ex: Porteiro.

Restrições dos atributos:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações

idFuncionario	N	S	Identificador (chave primária), não admite nulos. Não existem dois funcionários com o mesmo id.
contrato	N	N	Não admite nulos, pois cada funcionário possui um contrato. Cada contrato varia de funcionário para funcionário, sendo que vários funcionários podem ter o mesmo contrato, logo não admite nulos.
funcao	S	N	Admite nulos, caso esteja a null, é porque aquele funcionário não tem nenhuma função neste momento, sendo esta informação não muito importante com as outras. Vários funcionários podem possuir as mesmas funções, logo não tem valores únicos.

Relacionamentos:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
Empregam	N:1	Aeroporto	Funcionarios

Sec 3.1.9 - Entidade Passageiros

É importante saber os dados de quem vai no avião e de quem faz as compras dos bilhetes. De igual forma é muito importante saber também, se os passageiros têm alguma patologia que pode implicar problemas quando lhes for servido comida no decorrer do voo.
Esta entidade é derivada da Entidade Pessoas.

Atributos:

Nome do atributo	Tipo de Dados	Descrição

idPassageiro	Integer	Id (único) de cada passageiro. Ex: Para 3 passageiros, cada um terá o seu próprio id: 1; 2; e 3.
patologias	Variable characters (50)	Indica se o passageiro tem alguma patologia. Ex: Diabetes tipo II.

Restrições dos atributos:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
idPassageiro	N	S	Identificador (chave primária), não admite nulos. Não existem dois ou mais passageiros com o mesmo código.
patologias	S	N	Admite nulos. Caso o passageiro não tenha patologias associadas deve indicar na mesma e se não tiver então aparece como null, indicando assim que não tem quaisquer patologias. Diversos passageiros podem ter as mesmas patologias, logo não tem valores únicos.

Relacionamentos:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
Compram	N:M	Voos	Passageiros

Sec 3.2 - Relacionamentos

Nesta secção são descritos todos os relacionamentos existentes entre as várias entidades. Após uma análise aprofundada, constatou-se a necessidade dos seguintes relacionamentos:

- Compram;
- Controlam;
- Estacionou;
- Empregam;
- Dispoem;
- Possuem;
- Operam;
- Tem;
- Efectuam.

Sec 3.2.1 - Relacionamento: Compram

Este relacionamento pretende expressar o relacionamento existente entre as entidades Passageiros e Voos, no que concerne à compra de voos. O objetivo é expressar quais os passageiros que compraram bilhetes para voos e vice-versa, onde é que fizeram essa compra, quando a fizeram, se esta foi ou não cancelada, como foi realizada, se o passageiro leva ou não malas no porão, se o passageiro entrou ou não no avião, qual o lugar deste no avião, qual o tipo de classe que o passageiro quis viajar, qual a validade do bilhete e qual o seu código de barras. Neste relacionamento sempre que é feita uma compra de um bilhete fica registado na tabela desta relação todos os dados associados a essa compra. Exemplo: "comprei um bilhete hoje (dataEmissaoBilheteComprado), para o voo Nr. X (numVoo), o qual corresponde ao dia 2/2/2023".

Após uma análise do sistema de funcionamento da compra de bilhetes, e as orientações definidas pelo cliente, definiu-se as seguintes condições:

- Um passageiro pode comprar vários voos;
- Um voo é necessariamente comprado por um ou mais passageiros, caso contrário o voo não se pode realizar.

Tomando estas condições em consideração, definiram-se as seguintes características:

Entidade	**Obrigatório**	**Cardinalidade**	**Obrigatório**	**Entidade**
Passageiros	SIM	N : M	NÃO	Voos
Observações				
• Um passageiro pode comprar vários voos; • Um voo é necessariamente comprado por um ou mais passageiros, caso contrário o voo não se pode realizar.				

Atributos:

Nome do atributo	Tipo de Dados	Descrição
tipoBilhete	Variable characters (30)	Tipo de bilhete comprado pelo passageiro. Tem no máximo 30 caracteres. Ex: 1ª Classe.
lugarPassageiro	Variable characters (5)	Lugar onde o passageiro se vai sentar dentro do avião. Tem no máximo 5 caracteres. Ex: Este contracto visa a integridade de funcionários no...
validadeBilhete	Date & Time	Data de validade do bilhete comprado. Ex: 2023-12-10 12:30:00.
codigoBarrasBilhete	Long characters (1024)	Código de barras do bilhete. Usado para destinguir os bilhetes uns dos outros e para quando for necessário passar o bilhete no check-in. Tem no máximo 1024 caracteres. Ex: 237428374682374673.
tipoCompra	Variable characters (30)	Tipo de compra efetuada. Tem no máximo 30 caracteres. Ex: Online ou no balcão do aeroporto.
tipoPagamento	Variable characters (100)	Tipo de pagamento da compra. Tem no máximo 100 caracteres. Ex: MBway, cartão de crédito...

nomeFatura	Variable characters (30)	Nome na fatura da compra do bilhete respectivo. Tem no máximo 30 caracteres. Ex: Maria.
malaPorao	Number	Número de malas que cada passageiro leva no porão. Ex: 1.
dataEmissaoBilheteComprado	Date & Time	Data em que o bilhete foi comprado. Ex: 2023-12-10 12:30:00.
dataVooDoBilheteComprado	Date & Time	Data do voo para o qual o bilhete foi comprado. Ex: 2023-12-23 12:30:00.
cancelado	Boolean	Indica se a compra do bilhete foi cancelada. Ex: 0 - Não está cancelada. 1 - Foi cancelada.
verificacaoEntradaAviao	Boolean	Indica se o passageiro entrou ou não no avião. Ex: Quando a compra do bilhete é feita, este campo fica a 0 - indicando que o passageiro ainda não entrou no avião. Mas quando ele passa o bilhete no momento do check-in, este campo passa 1 - indicando assim que o passageiro entrou no avião. Isto dá jeito para medidas de segurança.

Restrições dos atributos:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
cancelado	N	N	Não admite nulos nem valores únicos, pois estes podem ser os mesmo para diversas compras.
malaPorao	S	N	Admite nulos. Caso esteja a nulo é assumido que não existe malas para colocar no porão do avião. Os seus valores podem ser os mesmos para diversas compras.

validadeBilhete	N	N	Não admite nulos - cada bilhete tem de ter uma validade. Pode acontecer vários bilhetes possuírem a mesma data de validade.
codigoBarrasBilhete	N	S	Não admite nulos - cada bilhete tem de ter um código de barras. Cada código é único.
tipoCompra	N	N	Não admite nulos - o tipo de compra é necessário. Não admite valores únicos, pois o tipo de compra vai ser o mesmo diversas vezes.
lugarPassageiro	N	S	Não admite nulos. Cada passageiro apenas possui um único lugar.
tipoBilhete	N	N	Não admite nulos. Pode haver vários bilhetes do mesmo tipo (1ª clase, 2ª classe, ...).
tipoPagamento	N	N	Não admite nulos. As compras de bilhete podem ser pagas de diversas formas (cartão de multibanco, mbway, ...) várias vezes.
dataVooDoBilheteComprado	N	N	Não admite nulos. Todos os bilhetes possuem uma data que indica quando se vai efetuar o voo. Para a mesma data do voo pode existir vários bilhetes comprados.
dataEmissaoBilheteComprado	N	N	Não admite nulos. Todos os bilhetes possuem uma data de quando este foi comprado. Pode acontecer que vários bilhetes tenham sido comprados em simultâneo.
nomeFatura	N	N	Não admite nulos. Cada fatura possui um nome. Pode existir faturas com o mesmo nome.

Sec 3.2.2 - Relacionamento: empregam

Este relacionamento engloba duas entidades: Aeroportos e Funcionários com uma relação 1 para N respectivamente, com obrigatoriedade do lado dos funcionários, porque sem funcionários o aeroporto não pode ser aberto e muito menos estar em desenvolvimento. Assim, este é obrigado a empregar funcionários para poder funcionar correctamente.

Após uma análise do sistema de funcionamento de trabalho dos funcionários num aeroporto, e as orientações definidas pelo cliente, definiu-se as seguintes condições:

- Vários funcionários são empregados por um e somente um aeroporto;
- Por sua vez, o aeroporto emprega vários funcionários;
- Para o aeroporto funcionar, este tem de empregar obrigatoriamente funcionários, caso contrário nunca poderia estar em funcionamento.

Tomando estas condições em consideração, definiram-se as seguintes características:

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Funcionários	SIM	N : 1	NÃO	Aeroportos
Observações				
• Um aeroporto para funcionar necessita de ter um determinado número de funcionários; • Um funcionário apenas trabalha num aeroporto.				

Sec 3.2.3 - Relacionamento: dispoem

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Aeroporto e ParqueEstacionamentoPrivado, no que concerne ao controlo de parque de estacionamento privados, caso estes existam no aeroporto em questão.

Este relacionamento engloba uma relação 1 para N, sem qualquer tipo de obrigatoriedade, pois não é obrigatório o aeroporto ter parques de estacionamento privados, nem é obrigatório que os parques de estacionamento privados sejam dispostos por aeroportos, podem ser dispostos por exemplo, por condomínios de prédios.

Após uma análise do sistema de funcionamento de parques de estacionamento privados, e as orientações definidas pelo cliente, definiu-se as seguintes condições:

- Não é obrigatório o aeroporto ter parques de estacionamento privados;
- Um aeroporto pode ter zero, um ou mais parques de estacionamento de estacionamento privados;
- Cada parque de estacionamento privado apenas se encontra localizado num determinado aeroporto.

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Aeroporto	NÃO	1 : N	NÃO	ParqueEstacionamentoControlo
Observações				
• Não é obrigatório que os parques de estacionamento privados sejam dispostos por aeroportos e não é obrigatório que os aeroportos tenham parques de estacionamento privativos.				

Sec 3.2.4 - Relacionamento: possuem

Este relacionamento engloba duas entidades: Aeroportos e Torres de Controlo com uma relação 1 para 1 respectivamente, com obrigatoriedade dos dois lados, porque para o aeroporto funcionar, este precisa de ter uma torre de controlo que lhe permite controlar o tráfego aéreo e todas as informações relativas aos voos e aviões, e as torres têm de estar situadas obrigatóriamente dentro do aeroporto, não faz sentido estarem noutro sítio que não no aeroporto;

Após uma análise do sistema e as orientações definidas pelo cliente, definiu-se as seguintes condições:

- É obrigatório um aeroporto possuir uma e uma só torre de controlo;
- É obrigatório a torre de controlo estar num aeroporto.

Tomando estas condições em consideração, definiram-se as seguintes características:

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Aeroportos	SIM	1 : 1	SIM	TorresControlo
Observações				
• Para um aeroporto funcionar, este precisa de ter uma torre de controlo que lhe permite controlar o tráfego aéreo e todas as informações relativas aos voos e aviões. A torre, tem de pertencer obrigatóriamente ao aeroporto, caso contrário esta existe em vão.				

Sec 3.2.5 - Relacionamento: tem

Este relacionamento engloba duas entidades: Companhias Aéreas e Aviões com uma relação 1 para N respectivamente, com obrigatoriedade do lado dos aviões, porque se a companhia não tem aviões, então não faz sentido ela estar associada a quaisquer aeroportos. Assim, é obrigatório que esta possua aviões, de modo a ser útil ao aeroporto e lucrativo para si. Por outro lado, os aviões só podem pertencer a uma companhia, mas não é obrigatório, ou seja, podem pertencer por exemplo ao estado;

Após uma análise do sistema de aéreo e as orientações definidas pelo cliente, definiu-se as seguintes condições:

- Um ou mais aviões pertencem apenas a uma companhia aérea;
- Uma companhia aérea é constituída obrigatoriamente por vários aviões

Tomando estas condições em consideração, definiram-se as seguintes características:

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade

CompanhiasAereas	NÃO	1 : N	SIM	Avioes
Observações				
• Se a companhia não tem aviões, então não faz sentido ela estar associada a quaisquer aeroportos. • Os aviões só podem pertencer a uma companhia, mas não é obrigatório, ou seja, podem pertencer por exemplo ao estado.				

Sec 3.2.6 - Relacionamento: efectuam

Este relacionamento engloba duas entidades: Voos e Aviões com uma relação N para M respectivamente, com obrigatoriedade dos dois lados, pois não há outra maneira para os voos serem realizados se não forem feitos pelos aviões, e os aviões se não fizerem aquilo para os quais foram construídos, que é fazer voos, não dá para estes serem realizados e muito menos para que o aeroporto seja activo. Um avião pode efetuar vários voos e um voo pode ser efetuado por vários aviões.
Após uma análise do sistema e as orientações definidas pelo cliente, definiu-se as seguintes condições:

- Um avião pode efetuar vários voos;
- Um voo pode ser efetuado por vários aviões.

Tomando estas condições em consideração, definiram-se as seguintes características:

Entidade	**Obrigatório**	**Cardinalidade**	**Obrigatório**	**Entidade**
Voos	SIM	N : M	SIM	Avioes
Observações				
• Os voos são sempre efetuados por aviões, caso não sejam, não se podem realizar; • Os aviões têm de efectuar voos, caso contrário, não servem para desempenhar a função pela qual foram construídos.				

Sec 3.2.7 - Relacionamento: operam

Este relacionamento engloba duas entidades: Aeroportos e Companhias Aéreas com uma relação N para M respectivamente, com obrigatoriedade do lado das companhias, porque sem aviões, os quais são fornecidos (pertencem às companhias) pelas companhias, não adianta o aeroporto estar aberto e estar a vender bilhetes de voos, quando não há meios de transporte para os realizar, logo é necessário que este seja operado obrigatoriamente pelas

companhias. Por outro lado, não é da obrigatoriedade das companhias operar em aeroportos, estas podem operar noutros sítios;

Após uma análise do sistema e as orientações definidas pelo cliente, definiu-se as seguintes condições:

- É obrigatório que um aeroporto seja operado por companhias aéreas;
- Uma companhia aérea pode ou não operar num aeroporto, esta pode operar noutro sítio.

Tomando estas condições em consideração, definiram-se as seguintes características:

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Aeroportos	NÃO	N : M	SIM	**CompanhiasAereas**
Observações				
• Sem companhias associadas a aeroportos, o aeroporto não consegue funcionar correctamente, pois precisa de aviões, aviões esses, que são fornecidos pelas companhias.				

Sec 3.2.8 - Relacionamento: controlam

Este relacionamento engloba duas entidades: Torres de Controlo e Voos com uma relação 1 para N respectivamente, com obrigatoriedade do lado das torres e do lado dos voos, porque os voos têm de ser obrigatoriamente controlados pelas torres de controlo e as torres de controlo têm de controlar obrigatoriamente os voos, caso contrário, estes não podiam sequer existir e seria uma desordem muito grande.

É necessário que haja uma terceira tabela de relacionamento, pois os dados podem ser alterados ao longo do tempo, isto é, devido a imprevistos, tanto a hora de chegada como a de partida, como a porta de embarque e o lugar onde o avião vai estacionar, podem mudar. Ao mudar, estes dados são registados e os televisores do aeroporto vão buscar estas novas informações de modo a não perderem os voos.

Após uma análise do sistema e as orientações definidas pelo cliente, definiu-se as seguintes condições:

- Um voo é sempre controlado por uma torre de controlo;
- Uma torre de controlo controla um ou mais voos.

Tomando estas condições em consideração, definiram-se as seguintes características:

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
TorresControlo	SIM	1 : N	SIM	Voos
Observações				

- Todos os voos têm obrigatóriamente de ser controlados pela torre de controlo, e a torre tem obrigação de controlar todos os voos.

Atributos:

Nome do atributo	Tipo de Dados	Descrição
horaPartidaAlterada	Date & Time	Hora de partida do voo em caso de alteração da mesma. Ex: 2023-10-31 19:30:00.
horaChegadaAlterada	Date & Time	Hora de chegada do voo em caso de alteração da mesma. Ex: 2023-12-30 10:30:00.
portaEmbarqueAlterada	Variable characters (5)	Porta de embarque em caso de esta ser alterada. Tem no máximo 5 caracteres. Ex: A40.
lugarAviaoEstacionado	Variable characters (20)	Lugar onde o avião vai estacionar, em caso de alteração. Tem no máximo 20 caracteres. Ex: Terminal D.

Restrições dos atributos:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações

horaPartidaAlterada	S	N	Aceita valores nulos e não tem valores únicos, pois quando é inserido um registo nesta entidade-relação, a hora de partida pode ser a mesma, logo fica a null, quer dizer que não foi alterada. Caso seja alterada altera-se o registo e coloca-se a hora nova. As horas podem ser as mesmas para diversos registos.
horaChegadaAlterada	S	N	Aceita valores nulos e não tem valores únicos, pois quando é inserido um registo nesta entidade-relação, a hora de chegada pode ser a mesma, logo fica a null, quer dizer que não foi alterada. Caso seja alterada altera-se o registo e coloca-se a hora nova. As horas podem ser as mesmas para diversos registos.
portaEmbarqueAlterada	S	N	Aceita valores nulos e não tem valores únicos, pois quando é inserido um registo nesta entidade-relação, a porta de embarque pode ser a mesma, logo fica a null, quer dizer que não foi alterada. Caso seja alterada altera-se o registo e coloca-se a nova porta. As portas podem ser as mesmas para diversos registos.
lugarAviaoEstacionado	S	N	Aceita valores nulos e não tem valores únicos, pois quando é inserido um registo nesta entidade-relação, o lugar onde o avião vai estacionar pode ser o mesmo, logo fica a null, quer dizer que não foi alterado. Caso seja alterado altera-se o registo e coloca-se o novo lugar. Os lugares podem ser os mesmos para diversos registos.

Sec 3.2.9 - Relacionamento: estacionou

Este relacionamento é composto por duas entidades: Pessoas e Parques de estacionamento Privados com uma relação N para M respectivamente. Nesta relação não existem quaisquer obrigatoriedades, pois as pessoas não são obrigadas a estacionar naqueles parques. Do mesmo modo, os parques de estacionamento não precisam de ser estacionados obrigatoriamente pelas mesmas.

Nós criámos esta relação, pois seria prudente e necessário guardar os movimentos de entradas e saídas nos parques, de modo a termos vários conhecimentos, por exemplo o mês em que os parques estão mais cheios. Sendo também possível registar o valor pago com base na diferença do tempo lá estado e no preço por hora do estacionamento em questão. É ainda possível registar o tipo de pagamento que foi feito.

Após uma análise do sistema e as orientações definidas pelo cliente, definiu-se as seguintes condições:

- Várias pessoas podem estacionar em um ou mais parque de estacionamento privado;
- O parque pode ou não ser estacionado por uma ou mais pessoas.

Tomando estas condições em consideração, definiram-se as seguintes características:

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Pessoas	NÃO	N : M	NÃO	ParquesEstacionamentoPrivado
Observações				
• Uma ou mais pessoas, pode estacionar em vários parques, por exemplo se tiver dois carros. Pode ir estacionar no parque x e no parque y.				

Atributos:

Nome do atributo	Tipo de Dados	Descrição
horaEntradaParque	Date & Time	Hora que a pessoa entrou no parque de estacionamento. Ex: 2023-10-23 09:30:00.
horaSaidaParque	Date & Time	Hora que a pessoa saiu do parque de estacionamento. Ex: 2023-10-31 19:30:00.
valorPago	Float	Valor pago com base na diferença das horas de entrada e saída do parque, e também do preço por hora do parque respectivo. Ex: 29 euros.

tipoPagamento	Variable characters (100)	Tipo de pagamento efectuado por parte da pessoa que estacionou no parque. Tem no máximo 100 caracteres. Ex: Multibanco.

Restrições dos atributos:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
horaEntradaParque	N	N	Não aceita valores nulos, porque sempre que a pessoa entra no parque temos de ter a hora de entrada guardada. Os valores não são únicos, porque podem a hora de entrada pode se repetir com outras pessoas noutros parques. Quando a pessoa está a entrar no parque, esta hora é guardada e quando a pessoa está a sair, esta hora não é mexida.
horaSaidaParque	S	N	Aceita valores nulos, porque nesta relação, os dados vão sendo alterados consoante a pessoa entra e sai do parque. Assim, os valores não são únicos, porque podem se repetir com outras pessoas noutros parques. Quando a pessoa está a entrar no parque, esta hora está a null, porque a pessoa ainda não saiu, e quando a pessoa está a sair, o registo é alterado com a hora a que a pessoa saiu.
valorPago	S	N	Admite nulos, pois o valor pago no inicio ainda não pode ser calculado, pois necessita de saber a hora que a pessoa saiu. Só quando a pessoa sai é que é alterado o registo com a quantidade a pagar com base na diferença das horas e do preço por hora do parque. Pode haver valores iguais a pagar por várias pessoas.

tipoPagamento	S	N	Admite nulos, pois o tipo de pagamento no inicio ainda não pode ser registado, pois necessita de saber a hora que a pessoa saiu. Só quando a pessoa sai é que é alterado o registo com o tipo de pagamento que foi feito. A forma de pagamento pode ser a mesma para diversas pessoas.

Cap 4 - Modelo Físico

Neste capítulo documenta-se o *script* da criação da base de dados.

Sec 4.1 - Diagrama do Modelo Físico

O modelo físico (ou diagrama de tabelas) completo com todas as tabelas é enviado em formato PDF.

Sec 4.2 - *Script* de criação da Base de Dados

Nesta secção apresentam-se as instruções SQL necessárias para criar as tabelas descritas anteriormente na secção 7.1 no SGBDR Oracle. O código SQL apresentado permite criar as tabelas, as restrições de integridade suportadas pelo SGBD, assim como as validações de dados definidas e valores por omissão:

```
/*==============================================================*/
/* DBMS name:      ORACLE Version 11g                           */
/* Created on:     12/11/2022 2:08:37 PM                        */
/*==============================================================*/


alter table AVIOES
```

```
   drop constraint FK_AVIOES_TEM_COMPANHI;

alter table COMPRAM
   drop constraint FK_COMPRAM_RELATIONS_PASSAGEI;

alter table COMPRAM
   drop constraint FK_COMPRAM_RELATIONS_VOOS;

alter table CONTROLAM
   drop constraint FK_CONTROLA_REFERENCE_AEROPORT;

alter table CONTROLAM
   drop constraint FK_CONTROLA_RELATIONS_VOOS;

alter table EFECTUAM
   drop constraint FK_EFECTUAM_EFECTUAM_VOOS;

alter table EFECTUAM
   drop constraint FK_EFECTUAM_EFECTUAM2_AVIOES;

alter table ESTACIONOU
   drop constraint FK_ESTACION_RELATIONS_PESSOAS;
```

```
alter table ESTACIONOU
   drop constraint FK_ESTACION_RELATIONS_PARQUESE;

alter table FUNCIONARIOS
   drop constraint FK_FUNCIONA_EMPREGAM_AEROPORT;

alter table FUNCIONARIOS
   drop constraint FK_FUNCIONA_HERANCA2_PESSOAS;

alter table OPERAM
   drop constraint FK_OPERAM_OPERAM_COMPANHI;

alter table OPERAM
   drop constraint FK_OPERAM_OPERAM2_AEROPORT;

alter table PARQUESESTACIONAMENTOPRIVADO
   drop constraint FK_PARQUESE_DISPOEM_AEROPORT;

alter table PASSAGEIROS
   drop constraint FK_PASSAGEI_HERANCA_PESSOAS;
```

```
drop table AEROPORTOS cascade constraints;

drop index TEM_FK;

drop table AVIOES cascade constraints;

drop table COMPANHIASAEREAS cascade constraints;

drop index RELATIONSHIP_17_FK;

drop index RELATIONSHIP_16_FK;

drop table COMPRAM cascade constraints;

drop index RELATIONSHIP_13_FK;

drop table CONTROLAM cascade constraints;

drop index EFECTUAM2_FK;

drop index EFECTUAM_FK;
```

```
drop table EFECTUAM cascade constraints;

drop index RELATIONSHIP_15_FK;

drop index RELATIONSHIP_14_FK;

drop table ESTACIONOU cascade constraints;

drop index HERANCA2_FK;

drop table FUNCIONARIOS cascade constraints;

drop index OPERAM2_FK;

drop index OPERAM_FK;

drop table OPERAM cascade constraints;

drop index DISPOEM_FK;

drop table PARQUESESTACIONAMENTOPRIVADO cascade constraints;
```

```
drop index HERANCA_FK;

drop table PASSAGEIROS cascade constraints;

drop table PESSOAS cascade constraints;

drop table VOOS cascade constraints;

/*==============================================================*/
/* Table: AEROPORTOS                                            */
/*==============================================================*/
create table AEROPORTOS
(
   IDAEROPORTO          INTEGER              not null,
   CODIGOPOSTAL         VARCHAR2(12)         not null,
   CIDADE               VARCHAR2(40)         not null,
   RUA                  VARCHAR2(128)        not null,
   NUMERO               INTEGER              not null,
   NOME                 VARCHAR2(40)         not null,
   NUMSETORES           INTEGER              not null,
   NUMFUNCIONARIOSPOSTO INTEGER,
   TIPOFUNCIONAMENTO    VARCHAR2(30)         not null,
```

```
  LOCALIZACAO          CLOB                not null,
  IDTORRECONTROLO      INTEGER             not null,
  constraint PK_AEROPORTOS primary key (IDAEROPORTO, IDTORRECONTROLO)
);

/*==============================================================*/
/* Table: AVIOES                                                */
/*==============================================================*/
create table AVIOES
(
  IDAVIAO              INTEGER             not null,
  IDCOMPANHIA          INTEGER,
  NOME                 VARCHAR2(40)        not null,
  TIPOAVIAO            VARCHAR2(50)        not null,
  NUMLUGARES           INTEGER             not null,
  constraint PK_AVIOES primary key (IDAVIAO)
);

/*==============================================================*/
/* Index: TEM_FK                                                */
/*==============================================================*/
create index TEM_FK on AVIOES (
```

```
  IDCOMPANHIA ASC
);

/*==============================================================*/
/* Table: COMPANHIASAEREAS                                      */
/*==============================================================*/
create table COMPANHIASAEREAS
(
  IDCOMPANHIA          INTEGER              not null,
  NOME                 VARCHAR2(40)         not null,
  LOCALIZACAODASEDEPRINCIPAL CLOB           not null,
  CONTRATO             CLOB                 not null,
  SIGLA                VARCHAR2(7),
  constraint PK_COMPANHIASAEREAS primary key (IDCOMPANHIA)
);

/*==============================================================*/
/* Table: COMPRAM                                               */
/*==============================================================*/
create table COMPRAM
(
  IDPASSAGEIRO         INTEGER              not null,
```

```
    IDVOO               INTEGER             not null,
    TIPOBILHETE         VARCHAR2(30)        not null,
    LUGARPASSAGEIRO     VARCHAR2(5)         not null,
    VALIDADEBILHETE     DATE                not null,
    CODIGOBARRASBILHETE  CLOB               not null,
    TIPOCOMPRA          VARCHAR2(30)        not null,
    TIPOPAGAMENTO       VARCHAR2(100)       not null,
    NOMEFATURA          VARCHAR2(30)        not null,
    MALAPORAO           NUMBER,
    DATAEMISSAOBILHETECOMPRADO DATE             not null,
    DATAVOODOBILHETECOMPRADO DATE               not null,
    CANCELADO           SMALLINT            not null,
    VERIFICACAOENTRADAAVIAO SMALLINT            not null,
    constraint PK_COMPRAM primary key (IDPASSAGEIRO, IDVOO)
);

/*==============================================================*/
/* Index: RELATIONSHIP_16_FK                                    */
/*==============================================================*/
create index RELATIONSHIP_16_FK on COMPRAM (
    IDVOO ASC,
    IDPASSAGEIRO ASC
```

```
);

/*==============================================================*/
/* Index: RELATIONSHIP_17_FK                                    */
/*==============================================================*/
create index RELATIONSHIP_17_FK on COMPRAM (
   IDVOO ASC
);

/*==============================================================*/
/* Table: CONTROLAM                                             */
/*==============================================================*/
create table CONTROLAM
(
   IDTORRECONTROLO      INTEGER              not null,
   IDVOO                INTEGER              not null,
   HORAPARTIDAALTERADA  DATE,
   HORACHEGADAALTERADA  DATE,
   PORTAEMBARQUEALTERADA VARCHAR2(5),
   LUGARAVIAOESTACIONADO VARCHAR2(20),
   constraint PK_CONTROLAM primary key (IDTORRECONTROLO, IDVOO)
);
```

```
/*==============================================================*/
/* Index: RELATIONSHIP_13_FK                                    */
/*==============================================================*/
create index RELATIONSHIP_13_FK on CONTROLAM (
   IDVOO ASC
);

/*==============================================================*/
/* Table: EFECTUAM                                              */
/*==============================================================*/
create table EFECTUAM
(
   IDVOO               INTEGER             not null,
   IDAVIAO             INTEGER             not null,
   constraint PK_EFECTUAM primary key (IDVOO, IDAVIAO)
);

/*==============================================================*/
/* Index: EFECTUAM_FK                                           */
/*==============================================================*/
create index EFECTUAM_FK on EFECTUAM (
```

```
  IDVOO ASC
);

/*==============================================================*/
/* Index: EFECTUAM2_FK                                          */
/*==============================================================*/
create index EFECTUAM2_FK on EFECTUAM (
  IDAVIAO ASC
);

/*==============================================================*/
/* Table: ESTACIONOU                                            */
/*==============================================================*/
create table ESTACIONOU
(
  IDPESSOA            INTEGER             not null,
  IDPARQUEESTACIONAMENTO INTEGER              not null,
  HORAENTRADAPARQUE    DATE,
  HORASAIDAPARQUE      DATE,
  VALORPAGO            FLOAT,
  TIPOPAGAMENTO        VARCHAR2(100),
  constraint PK_ESTACIONOU primary key (IDPESSOA, IDPARQUEESTACIONAMENTO)
```

```
);

/*==============================================================*/
/* Index: RELATIONSHIP_14_FK                                    */
/*==============================================================*/
create index RELATIONSHIP_14_FK on ESTACIONOU (
   IDPESSOA ASC
);

/*==============================================================*/
/* Index: RELATIONSHIP_15_FK                                    */
/*==============================================================*/
create index RELATIONSHIP_15_FK on ESTACIONOU (
   IDPARQUEESTACIONAMENTO ASC
);

/*==============================================================*/
/* Table: FUNCIONARIOS                                          */
/*==============================================================*/
create table FUNCIONARIOS
(
   IDPESSOA            INTEGER             not null,
```

```
  IDFUNCIONARIO       INTEGER             not null,
  IDAEROPORTO         INTEGER,
  CONTRATO            CLOB                not null,
  FUNCAO              VARCHAR2(50),
  constraint PK_FUNCIONARIOS primary key (IDPESSOA, IDFUNCIONARIO)
);

/*==============================================================*/
/* Index: HERANCA2_FK                                           */
/*==============================================================*/
create index HERANCA2_FK on FUNCIONARIOS (
  IDPESSOA ASC
);

/*==============================================================*/
/* Table: OPERAM                                                */
/*==============================================================*/
create table OPERAM
(
  IDCOMPANHIA         INTEGER             not null,
  IDAEROPORTO         INTEGER             not null,
  constraint PK_OPERAM primary key (IDCOMPANHIA, IDAEROPORTO)
```

```
);

/*==============================================================*/
/* Index: OPERAM_FK                                             */
/*==============================================================*/
create index OPERAM_FK on OPERAM (
   IDCOMPANHIA ASC
);

/*==============================================================*/
/* Index: OPERAM2_FK                                            */
/*==============================================================*/
create index OPERAM2_FK on OPERAM (
   IDAEROPORTO ASC,
   IDCOMPANHIA ASC
);

/*==============================================================*/
/* Table: PARQUESESTACIONAMENTOPRIVADO                          */
/*==============================================================*/
create table PARQUESESTACIONAMENTOPRIVADO
(
```

```
  IDPARQUEESTACIONAMENTO INTEGER              not null,
  IDAEROPORTO          INTEGER,
  NUMLUGARES           INTEGER              not null,
  TIPOPARQUEESTACIONAMENTO VARCHAR2(20)        not null,
  HORARIO              VARCHAR2(40),
  PRECOPORHORA         INTEGER              not null,
  TEMPOIRAPEATEAOAEROPORTO INTEGER              not null,
  LOCALIZACAO          CLOB                 not null,
  constraint PK_PARQUESESTACIONAMENTOPRIVAD primary key (IDPARQUEESTACIONAMENTO)
);

/*==============================================================*/
/* Index: DISPOEM_FK                                            */
/*==============================================================*/
create index DISPOEM_FK on PARQUESESTACIONAMENTOPRIVADO (
  IDAEROPORTO ASC,
  IDPARQUEESTACIONAMENTO ASC
);

/*==============================================================*/
/* Table: PASSAGEIROS                                           */
/*==============================================================*/
```

```
create table PASSAGEIROS
(
   IDPESSOA            INTEGER             not null,
   IDPASSAGEIRO        INTEGER             not null,
   PATOLOGIAS          VARCHAR2(50),
   constraint PK_PASSAGEIROS primary key (IDPESSOA, IDPASSAGEIRO)
);

/*==============================================================*/
/* Index: HERANCA_FK                                            */
/*==============================================================*/
create index HERANCA_FK on PASSAGEIROS (
   IDPESSOA ASC
);

/*==============================================================*/
/* Table: PESSOAS                                               */
/*==============================================================*/
create table PESSOAS
(
   IDPESSOA            INTEGER             not null,
   NOMEPROPRIO         VARCHAR2(40)        not null,
```

```
  APELIDO            VARCHAR2(40),
  NIF                INTEGER            not null,
  CODIGOPOSTAL       VARCHAR2(12)       not null,
  CIDADE             VARCHAR2(40)       not null,
  RUA                VARCHAR2(128),
  NUMERO             INTEGER,
  TELEFONE           INTEGER,
  DATANASCIMENTO     DATE               not null,
  CARTAOCIDADAO      VARCHAR2(12),
  PASSAPORTE         VARCHAR2(10),
  EMAIL              VARCHAR2(100),
  constraint PK_PESSOAS primary key (IDPESSOA)
);

/*==============================================================*/
/* Table: VOOS                                                  */
/*==============================================================*/
create table VOOS
(
  IDVOO              INTEGER            not null,
  NUMVOO             VARCHAR2(35)       not null,
  PRECOVOOBASE       INTEGER            not null,
```

```
    DATASEMANA          DATE                not null,
    NUMTERMINAL         INTEGER             not null,
    PORTAEMBARQUE       VARCHAR2(5)         not null,
    HORACHEGADA         DATE                not null,
    HORAPARTIDA         DATE                not null,
    ORIGEM              VARCHAR2(50)        not null,
    DESTINO             VARCHAR2(50)        not null,
    ESCALA              INTEGER             not null,
    LOCALESCALA         VARCHAR2(50),
    constraint PK_VOOS primary key (IDVOO)
);

alter table AVIOES
   add constraint FK_AVIOES_TEM_COMPANHI foreign key (IDCOMPANHIA)
      references COMPANHIASAEREAS (IDCOMPANHIA);

alter table COMPRAM
   add constraint FK_COMPRAM_RELATIONS_PASSAGEI foreign key (IDVOO, IDPASSAGEIRO)
      references PASSAGEIROS (IDPESSOA, IDPASSAGEIRO);

alter table COMPRAM
   add constraint FK_COMPRAM_RELATIONS_VOOS foreign key (IDVOO)
```

```
    references VOOS (IDVOO);

alter table CONTROLAM
  add constraint FK_CONTROLA_REFERENCE_AEROPORT foreign key (IDVOO, IDTORRECONTROLO)
    references AEROPORTOS (IDAEROPORTO, IDTORRECONTROLO);

alter table CONTROLAM
  add constraint FK_CONTROLA_RELATIONS_VOOS foreign key (IDVOO)
    references VOOS (IDVOO);

alter table EFECTUAM
  add constraint FK_EFECTUAM_EFECTUAM_VOOS foreign key (IDVOO)
    references VOOS (IDVOO);

alter table EFECTUAM
  add constraint FK_EFECTUAM_EFECTUAM2_AVIOES foreign key (IDAVIAO)
    references AVIOES (IDAVIAO);

alter table ESTACIONOU
  add constraint FK_ESTACION_RELATIONS_PESSOAS foreign key (IDPESSOA)
    references PESSOAS (IDPESSOA);
```

```
alter table ESTACIONOU
   add constraint FK_ESTACION_RELATIONS_PARQUESE foreign key (IDPARQUEESTACIONAMENTO)
      references PARQUESESTACIONAMENTOPRIVADO (IDPARQUEESTACIONAMENTO);

alter table FUNCIONARIOS
   add constraint FK_FUNCIONA_EMPREGAM_AEROPORT foreign key (IDAEROPORTO, IDFUNCIONARIO)
      references AEROPORTOS (IDAEROPORTO, IDTORRECONTROLO);

alter table FUNCIONARIOS
   add constraint FK_FUNCIONA_HERANCA2_PESSOAS foreign key (IDPESSOA)
      references PESSOAS (IDPESSOA);

alter table OPERAM
   add constraint FK_OPERAM_OPERAM_COMPANHI foreign key (IDCOMPANHIA)
      references COMPANHIASAEREAS (IDCOMPANHIA);

alter table OPERAM
   add constraint FK_OPERAM_OPERAM2_AEROPORT foreign key (IDAEROPORTO, IDCOMPANHIA)
      references AEROPORTOS (IDAEROPORTO, IDTORRECONTROLO);

alter table PARQUESESTACIONAMENTOPRIVADO
   add constraint FK_PARQUESE_DISPOEM_AEROPORT foreign key (IDAEROPORTO, IDPARQUEESTACIONAMENTO)
```

```
    references AEROPORTOS (IDAEROPORTO, IDTORRECONTROLO);


alter table PASSAGEIROS
  add constraint FK_PASSAGEI_HERANCA_PESSOAS foreign key (IDPESSOA)
    references PESSOAS (IDPESSOA);
```

Cap 5 - Conclusões

Este relatório documenta o trabalho desenvolvido ao longo do primeiro semestre do ano letivo de 2022/2023 com as aprendizagens dadas durante as aulas Teóricas, Práticas e Teórico-Práticas.

O objetivo de estudo e criação de uma empresa fictícia de gestão de um ou mais aeroportos, neste caso sobre um possível aeroporto a ser construído na zona centro, foi proposto por todos nós, elementos do grupo, e realizado no âmbito da unidade curricular de Base de Dados do curso de Engenharia Informática. É um estudo e criação hipotética sobre o tema de gestão de base de dados de aeroportos que em grande parte trouxeram uma melhor aprendizagem e uma aprendizagem enriquecedora, não só sobre o conteúdo aprendido nas aulas, como também, foi uma melhor forma de aprendizagem sobre o funcionamento dos aeroportos e o seu sistema de gestão.

Aprendemos como criar entidades que façam sentido e não "criar por criar", aprendemos também a identificar em que circunstâncias é que devemos criar uma tabela de relacionamento, com campos e características próprias. Aprendemos a identificar atributos derivados, compostos e de multivalor com base no nosso problema.

Adquirimos conhecimentos sobre a cardinalidade das diversas relações, isto é, se há ou não obrigatoriedade e qual o tipo de relacionamento, bem como a relacionar de forma rápida e eficaz as diferentes tabelas. Aprendemos também em que medidas é que o atributo é ou não único e/ou nulo. Aprendemos a tranformar o modelo ER desenhado, para a ferramenta "Power Designer", e posteriormente no modelo físico usando essa mesma ferramenta. No final, depois de resolver pequenos "erros", conseguimos gerar a BD.

Assim, estamos prontos e preparados para gerir um ou mais aeroportos que venham a ser construídos na zona centro, ou em outras zonas pelo país.

Em suma, concluímos este trabalho com competências de trabalho em grupo aprimoradas, com competências práticas da matéria dada em aula e com a consciencialização do funcionamento de um aeroporto.

Referências Bibliográficas

Anexos