

## EXERCÍCIO – MAPEAMENTO MER-RELACIONAL

Considere o sistema descrito abaixo e o DER correspondente à modelagem ER-X apresentado após a descrição:

- 1) Crie o Esquema Lógico do sistema realizando o mapeamento MER-Relacional. Indique todas as restrições pertinentes ao modelo relacional.
- 2) **Justifique as soluções** de mapeamento adotadas em casos em que há mais de uma possibilidade de mapeamento.
- 3) **Indique restrições** apresentadas no Projeto Conceitual (modelo ER) e na descrição do problema **que não podem ser garantidas** no Esquema Relacional e comente como serão tratadas.
- 4) **Indique aspectos de consistência** inerentes à lógica da aplicação que não são atendidos no banco de dados e deverão ser tratados na implementação.

## SISTEMA AEROPORTOS

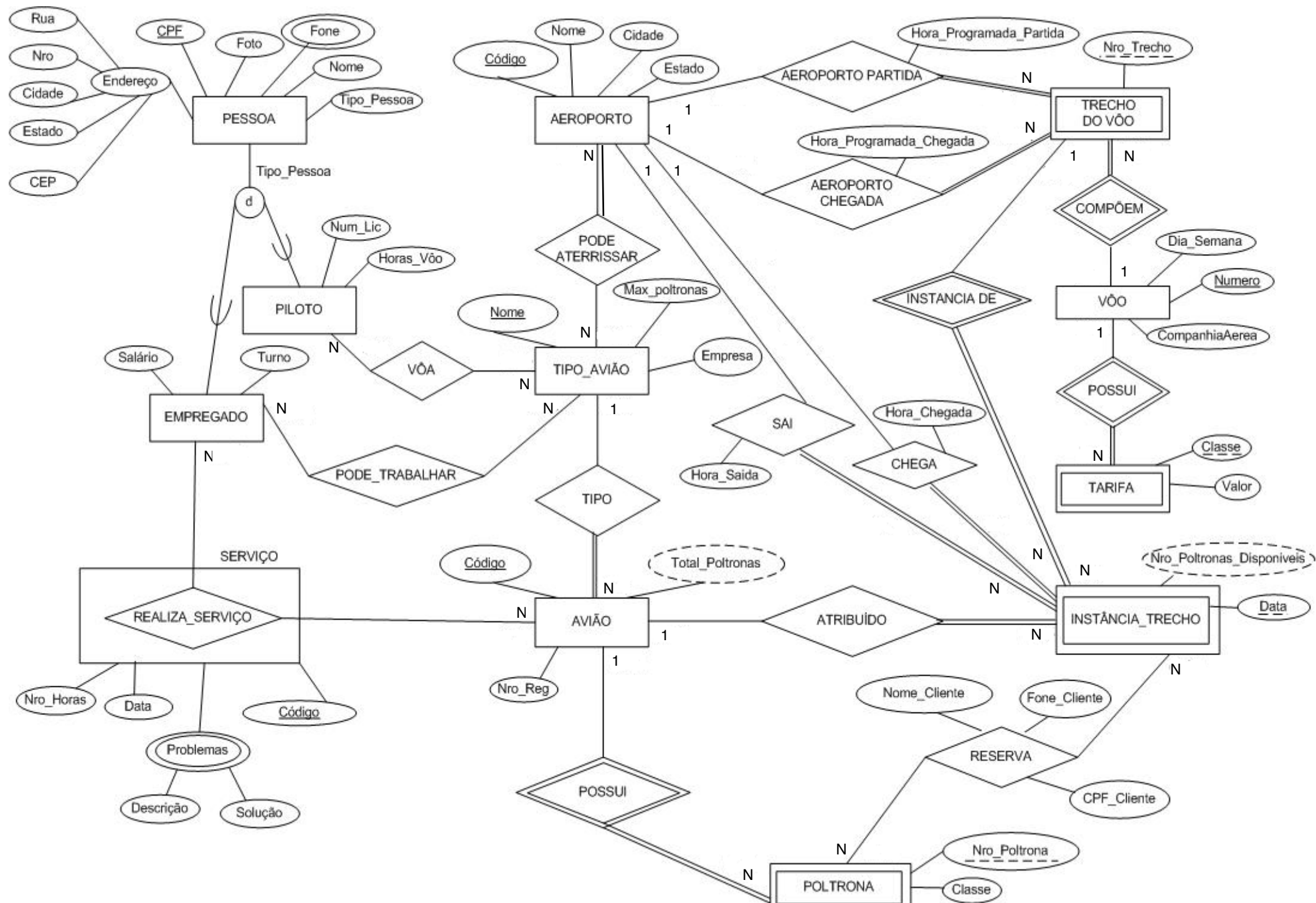
O sistema **Aeroportos** armazena informações de vôos, funcionários e aviões que atuam em aeroportos do Brasil.

- **Vôos** - São armazenados os dados dos vôos oferecidos pelas companhias aéreas. Os vôos são semanais, com horários programados. Por exemplo: vôo XX 666, da Companhia XXX, parte todas as 2as-feiras às 10:00h do aeroporto de Guarulhos (São Paulo) e chega às 10:50h no aeroporto do Galeão (Rio de Janeiro). Um vôo pode ter um ou mais **trechos**, que representam as possíveis escalas (**mesmo vôo, mesmo avião**). Por exemplo: o vôo XX 666 tem um único trecho (São Paulo/Rio de Janeiro), mas um vôo de Porto Alegre para Fortaleza pode ter um trecho Porto Alegre/São Paulo e um trecho São Paulo/Fortaleza. Para a efetiva ocorrência de um trecho de vôo em uma data específica (**Instância Trecho**), são armazenados os horários reais, o avião alocado e as reservas efetuadas. As tarifas de cada vôo dependem da classe, que pode ser Econômica, Executiva e Primeira Classe.

- **Aeroportos** – São armazenados os dados de identificação e localização, e os tipos de avião que podem aterrissar em cada aeroporto.

- **Aviões e Tipos de Aviões** – Sobre os tipos de avião, são armazenados um nome único (ex: Boeing 747, AirBus A340, Embraer 190), número máximo de poltronas que um avião desse tipo pode ter, os pilotos e empregados que estão habilitados a voar e a trabalhar em cada tipo. Sobre os aviões, são armazenados o tipo, um identificador único (número sequencial) e um número de registro, que também é único. Além disso, são registrados todos os serviços realizados no avião. Os serviços possuem um código único de identificação (número sequencial) e, para cada serviço, são registrados o número de horas gasto, a data, os problemas encontrados no avião e as respectivas soluções. Um mesmo empregado pode trabalhar no mesmo avião várias vezes, mas para cada uma delas é gerado um novo código de serviço.

- **Pilotos e empregados** – São armazenadas informações pessoais sobre pilotos e empregados. Entre as informações específicas do Piloto, estão o número de horas de vôo que possui e o número de sua licença, que é único.



\* NOT NULL

Aeroporto = { cód, nome, cidade, estado }

Tipo-avião = { nome, max poltronas, empresa }

Avião = { cód, nro registro, qtd poltronas, TIPO AVIÃO }

Poltrona = { AVIÃO, nro, classe }

Voo = { nro, dia semana, companhia }

Tarifa = { Voo, classe, valor }

Trecho\_Voo = { Voo, nro trecho, AEROPORTO SAÍDA, hora saída, AEROPORTO CHEGADA, hora chegada }

Instância-trecho = { Voo, NRO TRECHO, data, AEROPORTO SAÍDA, AEROPORTO CHEGADA, hora saída, hora chegada, AVIÃO }

Pessoa = { CPF, Tipo pessoa, nome, foto, endereço, telefone 1, telefone 2 }

\* Importante ter a pessoa registrada, pois tem muitas informações e é possível salvar ex-funcionários

Piloto = { CPF, nro licença, horas voo }

\* É preciso garantir consistência com [Tipo pessoa]

Empregado = { CPF, salário, turno }

Voo = { Piloto, TIPO AVIÃO }

Trabalha = { EMPREGADO, TIPO AVIÃO }

Aterrisa = { AEROPORTO, TIPO AVIÃO }

\* Impossível garantir totalidade

Reserva = { AVIÃO, NRO POLTRONA, Voo, NRO TRECHO, DATA, Nome cliente, telefone cliente, cpf cliente }

Serviço = { Código, AVIÃO, EMPREGADO, data, qtd horas }

\* Permite que um avião seja arrendado por mais de um empregado, e um empregado arrende mais de um avião enquanto avião é possível identificar cada contrato

Problema = { Serviço, nro, descrição, solução }

\* Como um avião pode ter muitos problemas, uma tabela que para cada serviço armazene os problemas nos da possibilidade de listar todos os problemas. O custo disso é o join