Bases de Dados

Engenharia Informática [Regime Diurno]

2.º Ano 1.º Semestre 2025/2026

Título	**PROGBD 1 – *Refactoring* de uma base de dados**
Contexto	**Aulas práticas**
Capítulo	**Capítulo 4 – Programação em bases de dados relacionais**
Conteúdos abordados	▪ Reconstruir modelos ER guiados por novos requisitos ▪ Adicionar tabelas com regras de integridade variadas ▪ Reconstruir tabelas e dados

A utilização destes exercícios pressupõe a preparação da arquitetura e cenário definidos na ficha *SQL 1 - Introdução à linguagem SQL*.

O Modelo Lógico das fichas de SQL está representado na Figura 1.
Este modelo utiliza a notação conhecida por *Chicken Feet* **[1]** ou *Crow's Foot* **[2]**.

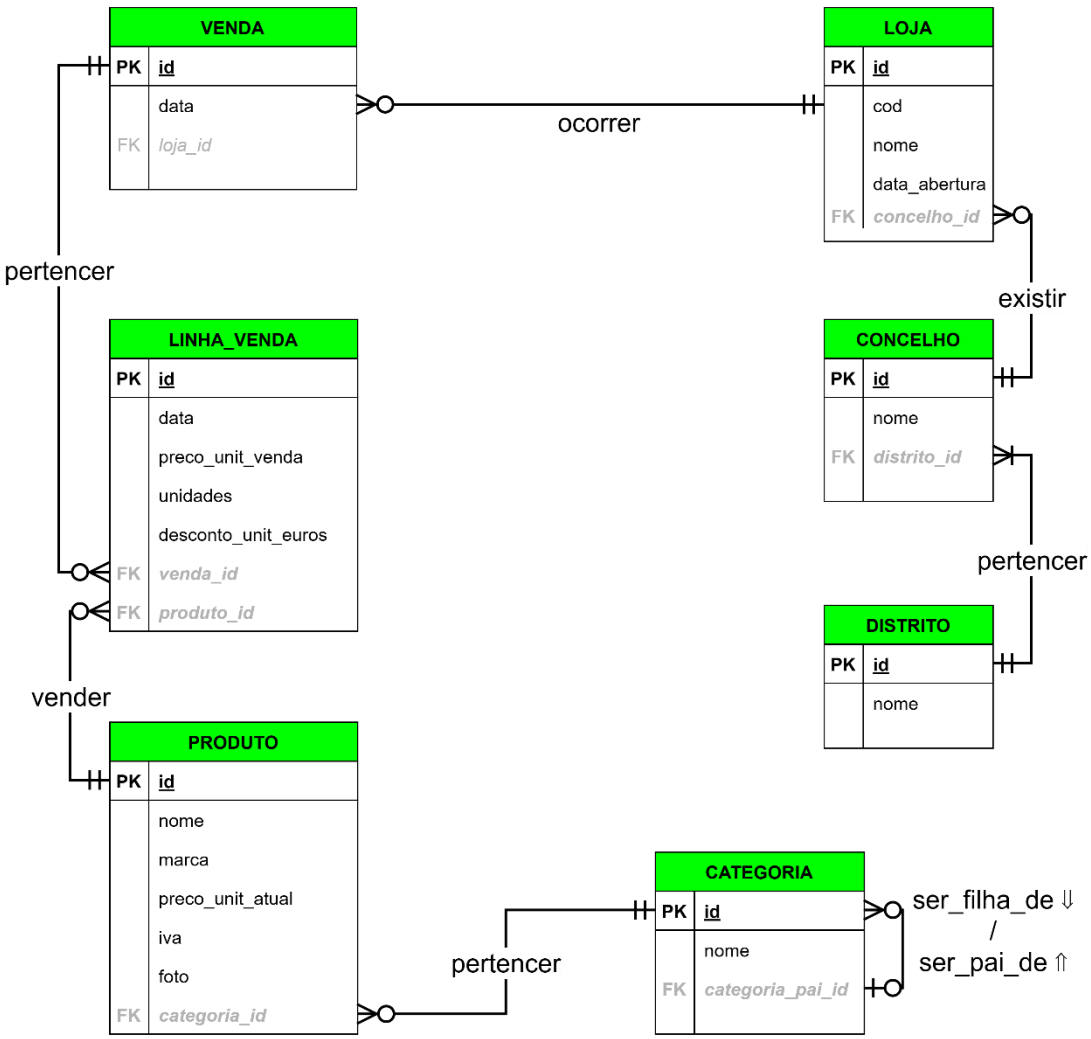

Figura 1 – Modelo Lógico do cenário das fichas das aulas práticas.

[1] Elmasri, R. & Navathe, S. (2016). Fundamentals of Database Systems (7ª edição). Pearson, **APPENDIX A**
[2] Coronel, C., Morris, S. & and Rob, P. (2011). Database Systems – Design, Implementation and Management (9ª edição). Course Technology – Cengage Learning, **CAPÍTULO 4**

A notação do estilo *Crow's Foot* apresenta os relacionamentos entre as tabelas num estilo fortemente gráfico. Por exemplo, eis o que pode saber-se sobre o relacionamento entre as tabelas VENDA e LOJA.

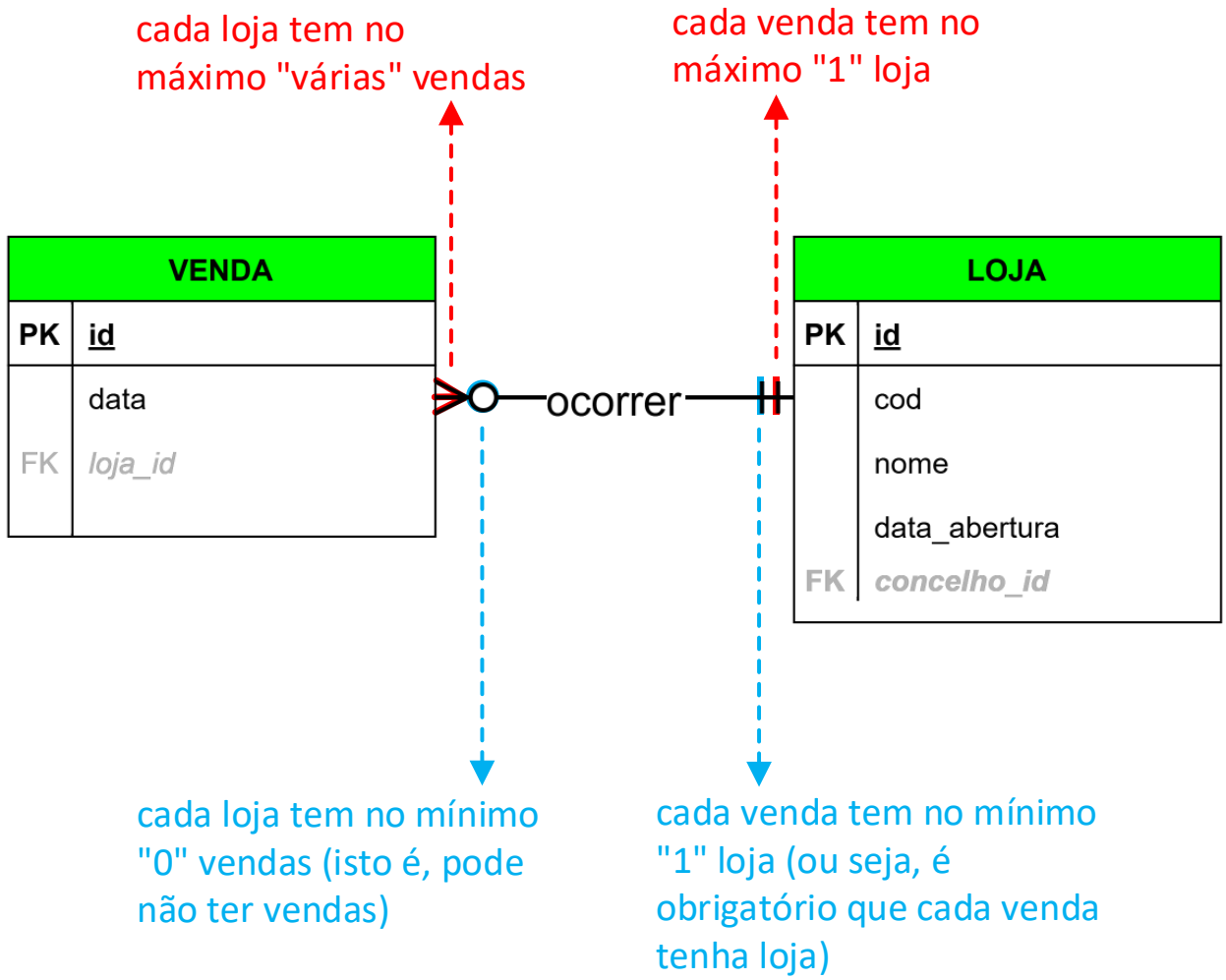

Figura 2 – Explicação da notação *Crow's Foot*.

REFACTORING (passo 1): atualizar os modelos de dados

1. Abrir o Modelo Lógico fornecido. Para isso:

a) Aceder à ferramenta *online* draw.io

b) Importar o ficheiro ***ficha_progbd1.drawio*** clicando em ***Open Existing Diagram***

2. Comparar este modelo lógico sem dados com o modelo lógico com dados do cenário (ver ficha SQL 1).

3. Considerar que a empresa definiu novos requisitos:

- É necessário conhecer os clientes e as vendas nas quais participam.
- Para cada cliente é necessário conhecer o seu concelho de residência.
- Podem existir concelhos na base de dados sem clientes associados.
- Cada cliente possui as seguintes características:

característica	domínio	valor por omissão	preenchimento obrigatório	único
identificador	números inteiros **[gerados sequencialmente pela bd]** **[min=1; max=999999999]**	-	**sim**	**sim**
número de identificação fiscal	9 dígitos **[inicia em 2 ou 5]**	-	*não*	**sim**
nome	caracteres **[primeira letra é maiúscula]** **[min=1; max=50]**	-	**sim**	*não*
data de nascimento	data **[max=data atual]**	-	*não*	*não*
data de adesão	data **[max=data atual]**	data atual	**sim**	*não*
género	uma letra do conjunto **{F, M, O}**	F	*não*	*não*

Considerando esses novos requisitos, editar o Modelo Lógico fornecido.

4. Adicionar a tabela de **CLIENTE**, com todas as **regras de integridade** necessárias, nomeadamente:

a) Regras para ***tipos de dados***

b) Regras de nulidade ***not null***

c) Regras de ***chave primária***

d) Regras de ***chave estrangeira***

e) Regras de ***unicidade***

f) Regras para ***valores por omissão***

g) Regras para ***imposição de domínio***

5. Usando o *software SQL Developer* e com comandos SQL, inserir dois clientes:

nif	nome	nascido a	aderiu a	concelho	género
22010100	Maria Gargalhada	1980-Out-12		Coimbra	F
	Anónimo		1900-jan-01	Porto	

Nota: o cliente *Anónimo* é utilizado frequentemente em ambientes reais para permitir uma venda quando o cliente envolvido não tiver registo pessoal no sistema.

6. No final da operação, confirmar as alterações de dados.

REFACTORING (passo 3): novas chaves estrangeiras

7. Modificar a tabela VENDA:

a) Acrescentar a nova coluna para chave estrangeira.

Nota: no final da operação, analisar como ficaram os dados da tabela.

b) Como não se sabe a quem pertencem as vendas já existentes, associá-las todas ao cliente Anónimo.

c) Implementar a regra de ***chave estrangeira*** .

Testes ao significado de chave estrangeira

8. Na tabela dos clientes, tentar remover o cliente *Anónimo*.

Não é possível, mas porquê?

Que incoerências seriam geradas se tal fosse possível?

9. Tentar, usando SQL, que as vendas não tenham cliente.

Não é possível, mas porquê?

Que incoerências seriam geradas se tal fosse possível?