

## Trabalho Prático – Parte 3 – Implementação

**Docentes:** José Barateiro e Fernando Lobo

### Entrega do trabalho (Parte 3)

- O trabalho deve ser entregue na tutoria eletrónica até ao dia **12 de dezembro de 2025**;
- O ficheiro deve ser um zip, com o nome BD202526\_Grupo\_##\_P3.zip (em que ## deve ser substituído pelo número atribuído ao grupo);
- O zip deve conter o relatório e todos os scripts usados (ddl.sql, dml\_insert.sql, queries.sql, proc\_func.sql, views.sql);
- O relatório deve poder ser impresso, mantendo-se todo o conteúdo legível. Se necessário, podem decompor o diagrama ER em mais do que uma imagem;
- **A entrega da Parte 3 é obrigatória.** A falta de entrega impede a conclusão da realização do trabalho prático e, conseqüentemente, a realização da Unidade Curricular de Bases de Dados;
- Cada dia de **atraso** tem uma **penalização de 1 valor na nota final** (após a discussão final do trabalho prático na última semana de aulas).

### Objetivo da Parte 3 do trabalho prático

O objetivo da Parte 3 do trabalho prático consiste em implementar e explorar a base de dados, incluindo toda a definição do esquema e povoamento de dados.

Todos os trabalhos devem cumprir com o conjunto de requisitos definido na secção requisitos deste enunciado. Para esse efeito, poderá ser necessário alterar o entregável 2.

### Requisitos

- Pelo menos 1 restrição de integridade que tem de ser implementada com um check
- Pelo menos 1 restrição de integridade que tem de ser implementada com um trigger
- Pelo menos 1 stored procedure ou uma função para dar resposta a um requisito funcional e/ou query complexa
- Necessidade de criação de duas vistas para limitar o acesso a dados

### Queries

Para dar resposta às queries, devem formalizar uma query em texto livre (exemplo: obter os nomes de todos os alunos que têm uma média superior a 15 valores no curso de Eng<sup>a</sup> Informática) e responder a essa query apresentando o respetivo código SQL.

As queries são numeradas de Q.1 a Q.12, e devem cumprir com os requisitos indicados para cada uma das queries, conforme a listagem seguinte:

- Q.1: Deve seguir o padrão “select from where”, com duas tabelas no from e pelo menos um filtro no where
- Q.2: Deve seguir o padrão “select from where group by”
- Q.3: Deve seguir o padrão “select from where group by having”,
- Q.4: Deve incluir um outer join (de qualquer tipo)
- Q.5: Deve usar uma nested query
- Q.6: Deve usar uma tabela temporária, através da utilização de “with”
- Q.7: Deve usar exists ou not exists
- Q.8: Deve usar um self-join, i.e., ter duas vezes a mesma tabela na cláusula from
- Q.9: Deve implementar a expressão sql equivalente a uma divisão em álgebra relacional
- Q.10: Deve filtrar campos de texto através da comparação com “like”

- Q.11: Deve usar uma nested query com comparação usando a cláusula ALL
- Q.12: Query livre, isto é, podem elaborar qualquer tipo de query, mas a complexidade da mesma é relevante para avaliação

### Estrutura do relatório

O relatório é composto pelas seguintes secções:

- *Descrição do problema*: corresponde a uma revisão/melhoria do texto entregue na Parte 2 do trabalho prático.
- *Modelo Entidade-Associação (ER)* – revisão/melhoria do modelo ER entregue na Parte 2 do trabalho prático. Devem anotar todas as restrições de integridade e justificar as decisões / opções tomadas na elaboração do modelo.
- *Modelo relacional* – revisão/melhoria do modelo relacional elaborado na Parte 2 do trabalho prático. Devem apresentar o modelo relacional que resulta da conversão do modelo ER para relacional. Devem apresentar todas as restrições de integridade.
- *Criação do esquema (ddl)* – devem apresentar o código ddl para criação do esquema da base de dados. As restrições de integridade, incluindo a que resulta da necessidade de criação de um trigger, devem ser implementadas. Além deste código constar no relatório, devem entregar um script, funcional, com o nome ddl.sql, que será parte do zip entregue.
- *Povoamento de dados* – devem povoar as tabelas com um conjunto de dados suficiente para demonstrar a coerência das queries (o volume de dados não será valorizado na avaliação). O povoamento de dados deve ser feito através de um conjunto de “insert”, que devem ser incluídos num script funcional (notem que devido às dependências, a ordem dos inserts é relevante). Além deste código constar no relatório, devem entregar um script, funcional, com o nome dml\_insert.sql, que será parte do zip entregue.
- *Queries* – devem constar todas as queries descritas em texto livre, seguidas da sua implementação, na sequência: Q1 texto livre / Q1 SQL; Q2 texto livre / Q2 SQL, etc. Além de constar no relatório, devem entregar um script denominado queries.sql em que apresentam todas as queries elaboradas (a definição em texto livre deve estar comentada, usando “--” ou “/\* \*/”
- *Stored procedure ou função* – devem indicar o requisito ou query complexa que leva à necessidade de criação do stored procedure ou da função e apresentar o código de criação, bem como da sua utilização. Além de constar no relatório, devem entregar um script, denominado proc\_func.sql com o código de criação do procedimento ou da função e um exemplo da sua utilização.
- *Vistas* – devem apresentar os requisitos que levam à necessidade de criação de duas vistas, apresentar o código de criação das mesmas e mostrar um exemplo da sua utilização. Além de constar no relatório, devem entregar um script, denominado views.sql com a criação das vistas e o exemplo da sua utilização.