

Ficha de Trabalho nº 1	
Disciplina	Modelação de Dados em Engenharia
Ano Letivo	2024/2025
Objetivo	Modelar dados de um problema num contexto de Eng. ^a Eletrotécnica
Aulas	5 aulas x 3 horas + 14 horas extra
Estudantes/Grupo	O trabalho deve ser realizado em grupos de 3 estudantes . Os elementos do grupo devem pertencer TODOS ao MESMO turno prático.
Data de Entrega	2025/04/22
<p>Objetivos concretos:</p> <ol style="list-style-type: none"> 1. Modelação de dados <ol style="list-style-type: none"> a. Compreender e interiorizar os conceitos de modelo e o papel da modelação no contexto de engenharia Eletrotécnica. b. Representação de modelos (linguagens e ferramentas). c. Tipo de modelos com ênfase nos tipos conceptuais/lógicos/físicos para a modelação de dados. d. O Modelo de Entidades e Relacionamentos (DER). e. Modelar e implementar a solução para o problema apresentado em baixo. 2. DBMS (Database Management System) <ol style="list-style-type: none"> a. Passagem dum esquema lógico para um esquema físico b. SQL (para as componentes DDL¹ e DML²) c. MySQL 3. Utilização do modelo num contexto prático de aplicação (JAVA). 	

¹ Data Definition Language

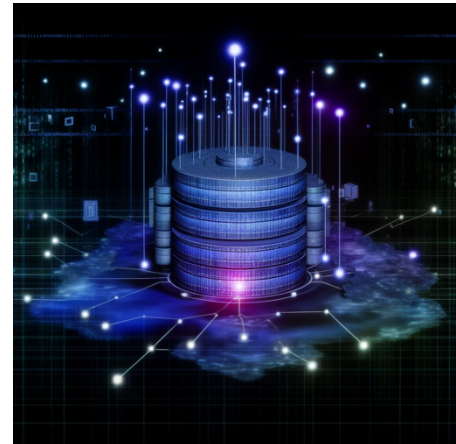
² Data Manipulation Language

Descrição do trabalho a efetuar

Este trabalho destina-se a proporcionar aos alunos um contacto com os conceitos básicos acerca do projeto/modelação de bases de dados, tendo em vista a sua posterior utilização em atividades enquadradas no seio da Engenharia Eletrotécnica e de Computadores.

Concretamente, o estudo e a implementação do problema proposto neste trabalho servirão para fazer uma abordagem aos seguintes conceitos, que constituem as fases de um projeto de criação de uma base de dados:

- Estudo do problema apresentado.
- Modelação dum diagrama de Entidades e Relacionamentos (*DER*) que satisfaça os requisitos do problema apresentado.
- Transformação do diagrama num esquema/modelo relacional e implementação numa base-de-dados, utilizando *SQL*.
- Implementação de questões (*SQL queries*), vistas (*SQL views*), triggers (*SQL triggers*) e funções/procedimentos (*SQL functions/procedures*) de forma a implementar as funcionalidades pretendidas no problema.
- Valorizam-se (até um máximo de 1 valor, **com limite máximo de 20 valores**) os trabalhos que implementem um conjunto de interfaces amigáveis para a visualização e manipulação dos dados armazenados na BD (*JDBC, outros*).



Descrição Geral do Problema

Considere que foi contratado para desenvolver um sistema de base de dados para uma empresa especializada na gestão de pequenos produtores agrícolas e otimização do uso de recursos. Este sistema deve gerir informações sobre produtores, cultivos, tratamentos, distribuidores e registos de produção e comercialização, considerando os seguintes pontos:

- **Produtores/Agricultores:** A empresa gere pequenos agricultores que cultivam diferentes tipos de produtos. Cada produtor é identificado por informações como nome, endereço, NIF (Número de Identificação Fiscal), telefone, área cultivada, entre outros.
- **Cultivos:** Cada produtor pode ter vários cultivos, identificados por um código único, tipo de cultura, período de plantação e colheita, rendimento esperado e práticas agrícolas adotadas.
- **Tratamentos:** Os produtores aplicam tratamentos agrícolas, como fertilizantes e pesticidas. Cada tratamento deve ser registado com seu nome, fornecedor, quantidade utilizada e data de aplicação.
- **Distribuidores:** Os produtos colhidos são enviados para distribuidores, que devem ser identificados por nome, localização, capacidade de armazenamento e histórico de compras.

- **Registos de Produção:** Cada lote de produção deve ser registado no sistema, incluindo data de colheita, quantidade produzida, qualidade do produto e destino da produção (supermercados por exemplo).
- **Registo de Comercialização:** O sistema deve registar os dados relacionados com a venda dos produtos agrícolas, registando informações sobre preços praticados, custos de logística e procura de mercado.
- **Sensores IoT:** O sistema deve permitir a integração com sensores IoT para monitorização em tempo real de variáveis agrícolas, como humidade do solo, temperatura, qualidade do ar e níveis de irrigação. Esses dados devem ser coletados, armazenados e analisados para otimizar a produção.
- **Criação de Alertas:** Criação de alertas para os produtores baseados nas condições atuais da produção, como por exemplo: humidade do solo a baixo de um determinado valor, ou temperatura da produção acima de um valor aceitável.
- **Simulação de Dados:** Devido à ausência inicial de um sistema operacional real, os dados referentes à produção e comercialização deverão ser simulados para testar e demonstrar as funcionalidades do sistema.
- **Dados Gerados:** Posteriormente, será disponibilizada a integração com fontes de dados externos, como previsões climáticas e tendências de mercado. A solução deverá ser capaz de integrar esses dados e sugerir boas práticas aos produtores.
- **Gestão de Recursos:** O sistema deve incluir funcionalidades para monitorizar e otimizar o uso de água, fertilizantes e energia na produção, minimizando desperdícios e promovendo a sustentabilidade.
- **Faturação:** Deve haver um registo de transações comerciais, detalhando o número da fatura, data, volume comercializado, destino da produção e estado da fatura (emitida/paga). As faturas devem ser geradas automaticamente aquando da venda dos produtos.

Este sistema de base de dados deve fornecer uma plataforma robusta para a gestão e operação eficiente de pequenas propriedades agrícolas, garantindo um melhor aproveitamento dos recursos e aumentando a rentabilidade dos produtores.



Criação do diagrama de entidades e relacionamentos (DER) e modelo relacional (a parte a azul é a mais importante)

O DER e modelo relacional devem ser criados conforme descrito nos slides das aulas teórico-práticas.

Para um trabalho com mais qualidade, recomendamos fazer-se uma especificação do diagrama o mais abrangente possível em termos de entidades e relacionamentos respetivos. Para tal, complemente e complete a descrição informal feita acima (descrição geral do problema). Depois, com base nos slides das aulas TP, relativos ao processo de modelação proposto por Peter Chan, proceder à identificação das entidades e dos relacionamentos. Alguns aspetos relevantes para a fase de modelação:

- O modelo deve estar correto, nomeadamente, reduzindo a possibilidade de ocorrência de informação redundante e a ocorrência de inconsistência nos dados.
- Ter especificamente atenção ao seguinte:
 - Numa fase inicial, ainda não se possui massa crítica para se conseguir avaliar o modelo obtido, nomeadamente, se está correto e completo.
 - Frequentemente, tem ocorrido que alguns alunos ficam durante as semanas em que o trabalho decorre inseguros sobre o modelo obtido, só conseguindo avançar para a fase de implementação na última semana do trabalho.
 - Para se evitar esta situação, recomenda-se um processo mais iterativo.
 - Ou seja, conseguir um modelo inicial que satisfaça o problema apresentado, ainda que não se tenha a certeza.
 - Avançar com o modelo para a geração da BD e tentar satisfazer os requisitos funcionais.
 - Se durante esta fase, for identificada alguma lacuna no modelo, então exportar os dados já introduzidos (para não se perderem), refinar o modelo, e prosseguir deste ponto.
 - Pode ocorrer que o ficheiro com os dados exportados necessite de alguns ajustamentos, se tiverem ocorrido alterações mais radicais no modelo.
 - Ao nível industrial (uma BD já em modo de produção), este processo iterativo de refinação está sempre a ocorrer. No entanto, as modificações às bases-de-dados são feitas de forma muito cuidada e cirúrgica, para garantir-se que não se perde ou compromete a informação armazenada.

No final, cada elemento do diagrama de entidades e relacionamentos especificado deverá ser justificável através do documento (em *.docx) semelhante ao anexo1.

Requisitos funcionais para o problema apresentado

O desenvolvimento do projeto para o cenário apresentado requer a construção dum modelo para a base-de-dados, que consiga satisfazer um conjunto considerável de requisitos. No entanto, dado o tempo disponível para este trabalho, é apenas necessário que o modelo consiga satisfazer os requisitos funcionais definidos na tabela 1.

Tabela 1 – Requisitos funcionais (para a BD)

Req. Func.	Descrição
RF 1	Implementar as operações CRUD da BD para as entidades identificadas.
RF 2	Registar valores provenientes das produções (quantidade, qualidade, comercialização, etc..).
RF 3	Visualizar o histórico de colheitas de um produtor num intervalo de tempo.
RF4	Ordenar e visualizar os produtores por ordem de produtos comercializados.
RF 5	Visualizar o estado atual dos tratamentos e stock de produtos.
RF6	Proceder ao registo de alertas para condições adversas
RF7	Extrair e processar dados de sensores IoT no campo agrícola. (Temperatura/humidade/luminosidade máxima/media/mínima)
RF 8	Gerar relatórios sobre eficiência agrícola e uso de recursos. Por exemplo, mostrar a produção total de um determinado produto e os recursos utilizados para o mesmo.
RF 9	<i>Proponha um requisito relevante e ainda por identificar que implique a criação e a especificação de uma ou mais entidades. Implemente.</i>
RF 10	<i>Proponha um requisito relevante ainda por identificar e que requeira uma query simples para o satisfazer (Query ou View usada numa Query). Implemente.</i>
RF 11	<i>Proponha um requisito relevante ainda por identificar e que requeira uma query com funções de agregação (sum, max, min, avg, etc) para o satisfazer. Implemente.</i>
RF 12	<i>Proponha um requisito relevante ainda por identificar e que requeira o desenvolvimento de functions / procedures para o satisfazer. Implemente.</i>
RF13	Implementação de uma aplicação em Java que seja cliente da base de dados desenvolvida (Realizar os requisitos anteriores numa aplicação)
RF14	Extração de dados reais da instalação de testes (Extrair e guardar os dados na base de dados respeitando os requisitos anteriores)

Para que se consiga avaliar adequadamente a qualidade do modelo de dados, é necessário que a BD possua dados em quantidade razoável. Por exemplo, calcular a média dum valor com base numa lista que contenha apenas um (ou poucos valores) não é satisfatório. **É necessário que cada tabela possua uma quantidade suficiente de dados que permita ilustrar os requisitos.**

Nota:

Ter em atenção que o projeto admite soluções alternativas, não existindo alguma que seja 100% correta à partida (e ainda não se tem experiência para isso). A melhor abordagem é ir começando com uma solução inicial plausível e experimentar o modelo com dados de testes. Conforme se vai testando cada requisito funcional, se algum não puder ser satisfeito, então reformula-se o modelo. Recomenda-se falar com os docentes sobre a estratégia para alterar o modelo, sem se perder os dados já existentes na BD.

De facto, num ambiente de produção (sistema em funcionamento em contexto real), o modelo (e respetivo esquema BD) é frequentemente alterado (ex: alter table ..., create view, ...) de forma a conseguir acompanhar a evolução do negócio/serviço/...

Modelação

No seu papel de analista do sistema, e face às necessidades da própria organização, já tomou a decisão que a melhor ferramenta para albergar os dados será o DBMS *MySQL*. Neste contexto, o seu trabalho consiste em três fases distintas, cada uma delas com um objectivo e um papel específico:

- Analista – Fazer a análise do sistema e construir um modelo abstrato (DER) dos dados
- Administrador de Base de Dados – Implementar o modelo abstrato em SQL, através de ferramentas adequadas, por exemplo o *mysql Workbench*. Garantir que as bases-de-dados por si geridas estão “sempre de boa saúde”, ou seja, servem o propósito para que foram projetadas e estão sempre corretas tendo em conta o seu esquema relacional. O administrador também tem o papel de gerir as permissões de acesso às tabelas e recursos existentes na base-de-dados.
- Programador – Enquadrar num contexto de aplicação a BD obtida. Neste caso, irá recorrer a linguagem Java, ou outra da sua preferência, e à utilização da conectividade JDBC/outra. O programador/utilizador que utiliza a base-de-dados fica assim sujeito ao modelo e regras de integridade definidas no esquema relacional (Valoriza-se o desenvolvimento da aplicação, no entanto não está contemplada na cotação do trabalho).

Na sua futura vida profissional poderá ser confrontado com as três tarefas em separado (em empresas de grande dimensão) ou as três em conjunto (nas empresas de pequena e média dimensão - mais típicas em Portugal). O trabalho de organização dos dados de uma empresa, bem como a melhoria da troca de informação, é muito importante para o aumento dos níveis de produtividade de qualquer organização.

Como analista do sistema deverá questionar as várias pessoas chave da organização, de modo que a estrutura de informação seja a mais adequada, começando, portanto, pela obtenção de um modelo abstrato que albergue as características importantes do problema considerado. Num cenário em que se pretende desenvolver uma base-de-dados ou um esquema relacional, procede-se à respetiva modelação através das seguintes fases de projeto:

1. Identificação das entidades presentes no problema (nível conceptual).
2. Identificação dos relacionamentos existentes entre essas entidades (nível conceptual).
3. Determinação dos atributos de cada entidade e as correspondentes propriedades desses atributos (nível logico).
4. Obtenção do esquema relacional (nível físico)

Implementação

Após obtido um *DER* para o problema proposto, passa-se para a fase de transformação desse diagrama num esquema relacional, que neste caso é implementado num *DBMS*.

Posteriormente, deverá construir as questões (*SQL queries*), vistas (*SQL views*) e funções (*SQL functions/procedures*) de forma a implementar as funcionalidades pretendidas no problema. Nesta fase é necessário que a base de dados obtida possua dados que possam ser utilizados para implementar e testar as *queries/procedures/...*; caso esses dados não existam, cria-se então um bom conjunto de dados, que corresponda o mais possível com a “realidade” do problema.

A fase final consiste de construir a interface com o utilizador, neste caso usando Java ou outra linguagem que considere mais adequada ou preferível, e que o aluno tenha já um bom domínio. Tal como o nome indica, e passando a redundância, a interface serve apenas como interface de visualização de resultados. Ou seja, os requisitos funcionais e não funcionais de utilização da BD deverão ser implementados na camada de base-de-dados. Mais uma vez o desenvolvimento desta componente está fora do objectivo deste trabalho.

Durante a execução deste trabalho utilizar-se-ão as seguintes ferramentas:

- *MySQL workBench / Models* – Permite definir *DERs* e posterior conversão em esquemas relacionais de forma automática.
- *MySQL workBench / Queries* – Interagir com a BD através de *Queries*.
- *MySQL* – Sistema de Gestão de bases de dados de código-aberto, pertencente à Oracle.
- *JDBC/ODBC/...* – Componente de interface (*driver*) que estabelece a conectividade entre um programa e um *DBMS*, que neste caso é necessária para a ilustração do problema desenvolvido num contexto de aplicação. Em princípio já vem instalada no sistema operativo.
- Adicionalmente, a máquina deverá ter o *JAVA SDK* instalado. Em algumas situações em que ocorreram problemas na instalação das ferramentas, estas deveram-se ao facto do sistema operativo corresponder a uma das versões “*Windows Home edition*”.

Elementos de Reflexão

Esta parte consiste em considerações mais reflexivas que poderiam servir como “inspiração” para a escrita dum relatório (no entanto, não é necessária entrega de relatório para este trabalho). No final do trabalho, as seguintes questões poderiam ser respondidas:

- O que se entende por modelação.
- Noção de modelo.
- Conceito de abstração (ilustrando com pormenores específicos do problema apresentado)
- A utilidade da modelação no contexto da Engenharia.
- O carácter ambíguo, redundante e incompleto da linguagem natural como formalismo de modelação.
- As características que um formalismo de modelação (ou linguagem) deve possuir para que possa representar modelos de forma adequada.

Planeamento das aulas

Este trabalho segue o seguinte plano:

0ª aula (em casa):

Instalação do *Software* necessário ao desenvolvimento do trabalho.

1ª aula:

Interação com a base-de-dados MySQL. “Queries” SQL, exemplo de aplicação.

Especificação de regras de integridade com SQL.

2ª aula:

Leitura e análise do trabalho prático.

Análise dos requisitos e modelação DER.

Implementação da BD (forward engineering).

3ª aula:

Implementação das funcionalidades requeridas no trabalho prático, CRUD e *Queries* para consulta de informação. Funções de agregação.

4ª aula:

SQL/PSM : procedimentos, funções e *Triggers*.

5ª aula:

Finalização do trabalho.

Entrega (no MOODLE)

O trabalho é entregue UNICAMENTE via Moodle³ e consiste em:

- ✓ [OBRIGATÓRIO] Preencher RELATÓRIO do trabalho (google form). Neste relatório serão pedidos (entre outros):
 - Descrição de como se identificaram as entidades e os relacionamentos a partir da descrição do problema (ver ANEXO 1).
 - Submissão do diagrama de entidades e relacionamentos (jpg, pdf, etc)
 - Submissão do modelo mySQL-Workbench respectivo (mwb)
 - Descrições como foram implementadas as queries relativas aos diferentes RFs
 - Submissão de pequeno vídeo (5-8 minutos) mostrando as funcionalidades desenvolvidas.
- ✓ [OBRIGATÓRIO] Submeter ficheiro ZIP/RAR file com tudo o que foi efectuado no desenvolvimento da base de dados:
 - DDL.sql (forward engineering – referente ao DER)
 - DML_funcional_requirements.sql (VIEWS e Queries para cada RF)
 - DML_procedures_functions_triggers.sql,
 - Data.sql (com os dados de teste)
 - Outros scripts sql criados durante o desenvolvimento do trabalho (opcional).
- ✓ Submeter projeto do programa Java.

É importante que cada grupo consiga assegurar-se que os docentes das práticas vão conseguir visualizar o modelo no “*mySQL workbench*” e executar os ficheiros “DDL.sql”, “DML.sql”, e “data.sql” dentro do “*mySQL workbench*”.

Para efeitos de verificação, aconselha-se que cada aluno/grupo construa um “zip/rar” do modelo e o extraia para outra directoria, verificando se se consegue abrir o modelo.

³ Entregas feitas através de email NÃO são consideradas.

Avaliação do Trabalho

É sujeito a apresentação oral, onde têm de estar presentes todos os elementos do grupo (alunos ausentes ficam com **0 valores** na avaliação da componente prática). Apesar do projeto ser feito em grupo, a nota desta componente será sempre individual.

Critérios de avaliação

Itens	Descrição	Cotação / penalização
Modelos	Diagrama de Entidades e Relacionamentos	1.5
	Modelo mySQL-Workbench	1.0
Requisitos Funcionais	RF 1 Implementar as operações CRUD da BD para as entidades identificadas.	1.0
	RF 2 Registrar valores provenientes das produções (quantidade, qualidade, comercialização, etc.).	0.5
	RF 3 Proceder ao registo de alertas para condições adversas.	1.5
	RF4 Visualizar o histórico de colheitas de um produtor num intervalo de tempo.	0.75
	RF 5 Visualizar os produtores que mais comercializam produtos.	1.0
	RF6 Visualizar o estado atual dos tratamentos e stock de produtos.	0.75
	RF7 Extrair e processar dados de sensores IoT no campo agrícola. (Temperatura/humidade/luminosidade máxima/media/mínima)	1.5
	RF 8 Gerar relatórios sobre eficiência agrícola e uso de recursos.	1.25
	RF 9 Proponha um requisito relevante e ainda por identificar que implique a especificação de uma ou mais entidades. Implemente.	0.75
	RF 10 Proponha um requisito relevante ainda por identificar e que requeira uma query simples para o satisfazer (Query ou View usada numa Query). Implemente.	1.0
	RF 11 Proponha um requisito relevante ainda por identificar e que requeira uma query com funções de agregação (sum, max, min, avg, etc) para o satisfazer. Implemente.	1.0
	RF 12 Proponha um requisito relevante ainda por identificar e que requeira o desenvolvimento de functions / procedures para o satisfazer. Implemente.	1.0
	RF13 Implementação de uma aplicação em Java que seja cliente da base de dados desenvolvida (Realizar os requisitos anteriores numa aplicação)	1.0
	RF14 Extração de dados reais da instalação de testes (Extrair e guardar os dados na base de dados respeitando os requisitos anteriores)	1.0
Discussão	Prestação do grupo (projeto, desempenho discussão oral, relatório)	1.5
	Prestação individual	2.0
Total		20.0
Vídeo	Entrega de vídeo demonstrando todas as funcionalidades é obrigatória (não entrega corresponde a uma não entrega de todo o projeto)	-20.0

Docentes

Teórico-prática:

João Rosas, jrosas@uninova.pt

Ana Inês Oliveira, aio@fct.unl.pt

Prática:

Filipa Ferrada, fam@fct.unl.pt

André Rocha, ad.rocha@fct.unl.pt

Ilustrando como se identificam entidades a partir da descrição do problema. Para modelos de grandes dimensões, uma boa abordagem consiste em representar parcelas pequenas do modelo, ou seja, um fragmento de cada.

