

Universidade Lusófona de Humanidades e Tecnologias

LEI, LEIRT e LIG

Bases de dados

Trabalho prático de avaliação continua

João Caldeira

Luís Alexandre Gomes

Novembro 2023

Índice

Introdução	3
Descrição funcional	3
Modelo de dados	6
Trabalho solicitado	7
Etapa 1	7
Etapa 2	8
Etapa 3	9
Regras	13
Cotações	13
Formato de entrega e prazo	14
Anexo 1: Descrição de domínio e Input	15

Introdução

O Trabalho Final de Curso (TFC) é um elemento fundamental do currículo das Licenciaturas LIG e LEI da ULHT funcionando como um momento nuclear na formação dos alunos, nomeadamente em ordem à promoção da sua integração profissional ou iniciação ao trabalho científico

Os TFC decorrem no âmbito das Licenciaturas e sob a orientação de um dos Professores que leccionam no DEISI a unidade curricular de TFC. Para além do professor orientador, o TFC poderá ainda ter um coorientador, podendo este ser professor do DEISI, professor de outro departamento da ULHT ou oriundo de entidade externa onde decorre o projecto, caso esta exista.

Desde 2018 a gestão de TFC tem sido realizada por via de uma plataforma web dedicada, ela própria desenvolvida no âmbito de vários TFC, que tem vindo a evoluir em resposta às necessidades e à experiência que se vai acumulando com o seu uso. A plataforma tem uma forte componente de dados, de modo a suportar as componentes funcionais da plataforma – Atribuição de temas; Avaliações – mas também para permitir análise estruturada de dados e estatísticas.

Após alguns anos de uso desta plataforma web, é chegado o momento para rever o seu funcionamento e actualizar a plataforma. Esta revisão pretende consolidar as alterações que foram sendo realizadas ao longo do tempo de utilização, incorporar novas funcionalidades que se identificaram como necessárias e otimizar a plataforma

No presente trabalho, o objectivo é desenvolver uma possível estrutura de dados e implementar a base de dados correspondente em termos que possam vir a dar corpo a uma proposta de melhoria da plataforma de gestão de TFC

Descrição funcional

Nos termos implementados na plataforma, a gestão de TFC tem duas componentes essenciais: (i) Atribuição de temas para desenvolvimento, que consiste na disponibilização de temas, candidaturas dos alunos aos temas aprovados e a atribuição propriamente dita; (ii) Avaliações, a realizar em vários momentos ao longo do ano lectivo e com base nos quais se define a classificação a atribuir a cada aluno no final da Unidade Curricular de TFC

Por uma questão de complexidade e extensão, o presente trabalho irá centrar-se apenas na primeira componente:

A. Atribuição de temas para desenvolvimento

Esta componente inclui os seguintes subprocessos:

(i). Inscrição de alunos em TFC

1. Os trabalhos podem ser realizados individualmente ou em grupo, mas todos os alunos têm que se registar na plataforma antes de se poderem candidatar.

(ii). Submissão de temas para desenvolvimento

1. Os temas podem ser propostos por docentes, por entidades externas (parceiros) ou pelos próprios alunos, sendo que qualquer um pode submeter mais do que um tema

2. Todos proponentes são registados na base de dados, independentemente de poderem vir a ser orientadores de TFC
3. Alguns temas podem ser submetidos já com indicação de aluno/grupo que o irá desenvolver
4. Para cada tema, indicam-se tecnologias a utilizar, no mínimo de 1 e sem limite máximo. O mesmo para as disciplinas que os alunos terão que ter realizado para desenvolverem o tema
5. Todos os temas têm obrigatoriamente orientador, seleccionado da lista de docentes alocados a TFC, e pode ter um, e apenas um, co-orientador, seleccionado da mesma lista ou das listas de docentes que não são do departamento ou de representantes de entidades externas.

O proponente do tema pode não ser (co)orientador

6. Os temas propostos por docentes são automaticamente aceites. Os temas propostos por alunos ou entidades externas são avaliados quanto a pertinência e conteúdo, podendo ser aceites conforme f, aceites com alterações ou recusados

Eventuais alterações são registadas nos comentários de avaliação, não se altera a proposta original

(iii). Candidatura de alunos a temas

1. Cada aluno ou grupo candidata-se até ao máximo de 6 temas, sendo obrigatório o mínimo de 3, indicando ordem de preferência
2. Cada aluno pode inscrever-se individualmente ou em grupo, mantendo o limite de 6 temas. Do mesmo modo, um aluno que tenha proposto tema próprio pode candidatar-se a outros temas
3. Os temas propostos por alunos só podem ser realizados pelo aluno/grupo que o propôs, pelo que não podem aceitar candidaturas
4. Só alunos inscritos se podem candidatar

(iv). Atribuição de temas

1. Qualquer aluno que não tenha o mínimo de 90 ECTS será excluído da atribuição de temas, pelo que se torna obrigatória o registo de créditos de cada aluno inscrito
2. Os temas propostos com indicação de executante, individual ou em grupo – ponto (ii).3 – são automaticamente atribuídos. Do mesmo modo, os temas propostos por aluno são automaticamente atribuídos após validação
3. Os restantes temas são atribuídos de acordo com os seguintes critérios de seriação¹:
 - 1º. Ordem de escolha (ponto (iii).6)
 - 2º. Cronologia de submissão de candidatura
4. Cada TFC só pode ter uma atribuição.

A candidatura atribuída fica com estado “Atribuído”, enquanto as restantes ficam com o estado “Não Atribuído”

¹ Na realidade, existem outros critérios que requerem interactividade e que não serão considerados no presente trabalho

5. Cada aluno só pode ter uma atribuição, seja individualmente ou em grupo

A candidatura atribuída fica com estado “Atribuído”, enquanto as restantes ficam com o estado “Não Atribuído”

6. Após atribuição de todos os trabalhos, a lista de atribuições é publicada, trancando a fase inicial de TFC e disponibilizando os trabalhos para as fases seguintes, ou seja, avaliações

(v). Estados:

1. Temas de TFC:

- “Aguarda Aprovação”: Temas não validados quando submetidos por alunos, entidades externas ou docentes que não sejam do departamento
- “Recusado”: Temas com validação negativa
- “Aguarda Atribuição”: Temas submetidos por docentes do departamento ou validados positivamente quando submetidos por outros preponentes
- “Atribuído”: Temas já atribuídos a aluno ou grupo
- “Publicado”: Temas atribuídos e publicados
- “Indisponível”: Tema retirado por preponente
- “Anulado”: Tema retirado por indisponibilidade de recursos.

Pode ocorrer após atribuição, o que implica que o aluno/grupo volte ao estado de não ter tema atribuído

Todas as alterações realizadas em temas ficam registados num log

2. Inscrições:

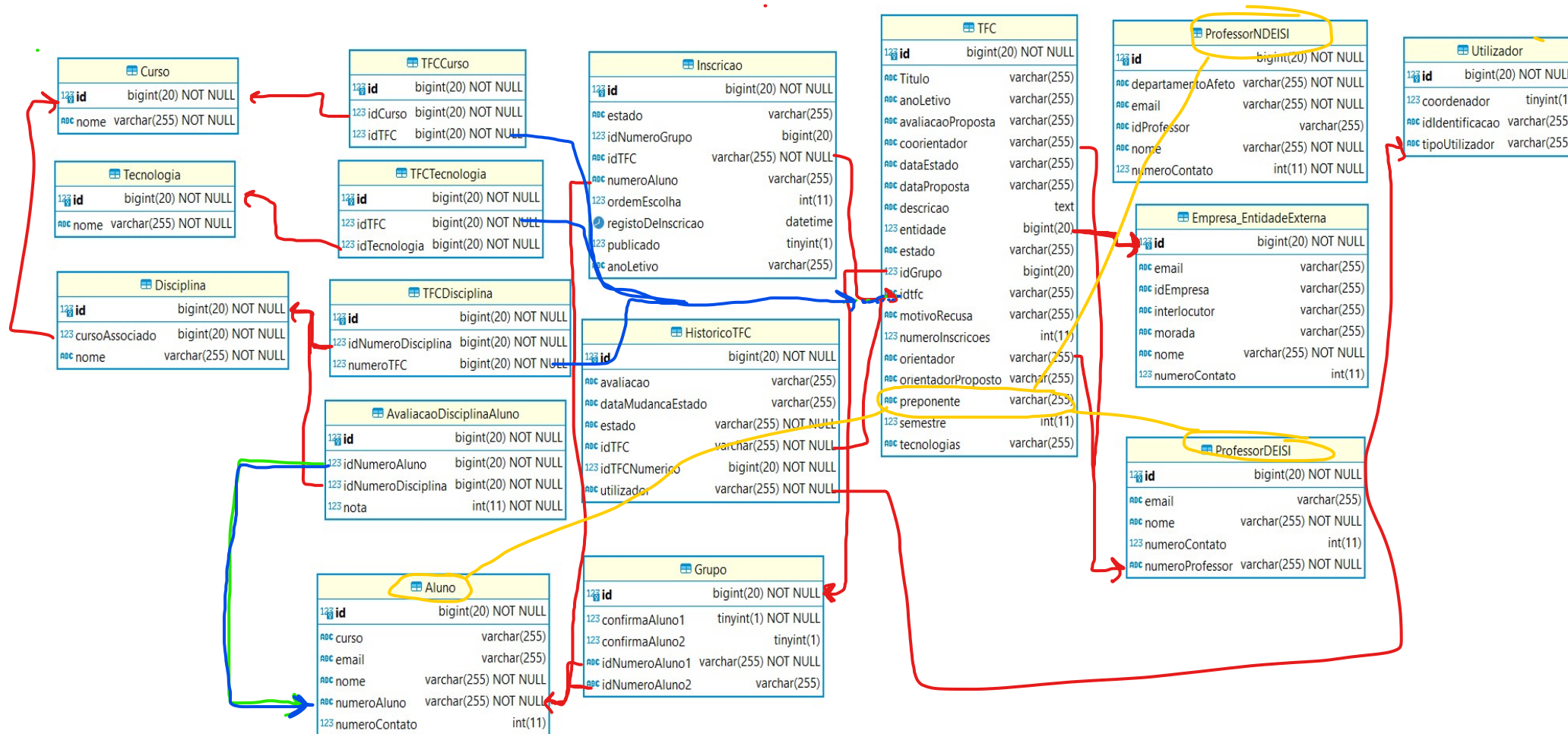
- “Aguarda Atribuição”: Candidatura submetida, sem apreciação
- “Atribuído”: Candidatura aprovada e com tema atribuído
- “Não Atribuído”: Candidatura com tem atribuído a outro candidato ou com candidato a quem foi atribuído outro tema
- “Anulado”: Inscrição retirada por anulação de tema ou por exclusão de aluno.

3. Aluno:

- “Em Validação”: Aluno inscrito, mas sem validação de ECTS
- “Válido”: Alunos com contagem de ECTS válida
- “Excluído”: Alunos sem ECTS mínimo

Modelo de dados

Para suportar este problema apresentamos os seguintes modelos de dados:



Modelo 1: Estrutura de dados para plataforma de gestão de TFC

Trabalho solicitado

Descrição geral do trabalho:

- Carregar os ficheiros de texto para as tabelas LOAD com nome [load_«nome_ficheiro»];
- Limpar dados carregados;
- Normalizar os dados carregados;
- Transformar dados, carregando-os para as tabelas definitivas, respeitando as relações definidas no modelo;
- Executar *queries* relevantes para a análise de dados, conforme indicados no enunciado;

Etapa 1

1. Criar as tabelas indicadas no modelo 1
2. Carregar os ficheiros fornecidos para dentro das tabelas base indicadas no modelo.

No MySQL/MariaDB pode-se usar o comando LOAD para carregar os ficheiros.

Todos os SGBD têm utilitários de importação, como o LOAD anteriormente referido para MySQL/MariaDB, o IMPDP do Oracle e o *Import* do SQL Server.

Por norma, os IDE também têm ferramenta própria. No caso do DBeaver, também existe um utilitário de importação, acessível a partir do menu de contexto de qualquer base de dados (fig. 1)

A importação pode ser realizada utilizando esta ferramenta do DBeaver, mas sob a condição de não ocorrer qualquer tratamento de dados na importação

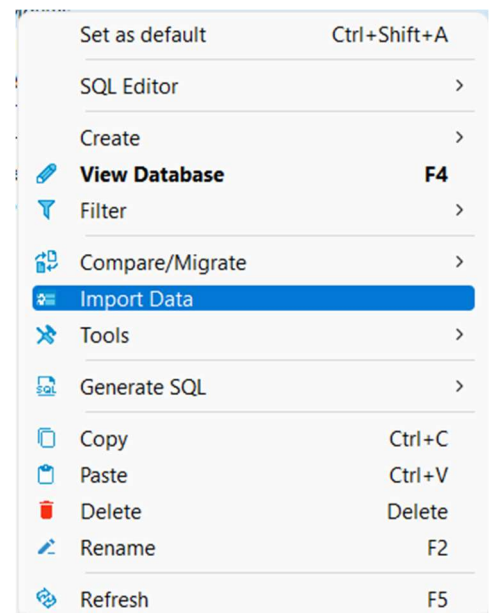


fig. 1 menu de contexto

3. Criar índices convenientes para acelerar as operações que vão ser feitas nos passos seguintes. No mínimo, devem ser criados índices em todas as colunas de identificação – [id] – de referência a TFC – [idTFC] - ou de referenciação entre tabelas

Sempre que conveniente, os índices devem aceitar valores repetidos;

4. Limpar os dados carregados:

4.1.Retirar espaços no início e no fim de cada coluna de texto;

4.2.Retirar caracteres especiais no início e no fim de cada coluna de texto;

Ex.: '', '[' ou ']';

4.3.Garantir que todos os valores de *estado* estão dentro da lista indicada acima

Os estados incorrectos devem ser substituídos por [Anulado]

5. Normalizar os dados carregados:

5.1. **Verificar** se existem dados repetidos:

Neste exercício, pedem-se *queries* que verifiquem eventuais falhas à normalização, a correcção será realizada no exercício seguinte

5.1.1. Na tabela de [TFC], não pode haver duplicação de identificadores [idTFC];

5.1.2. Não pode haver duplicações de [Aluno].[numeroAluno],
[ProfessorDEISI].[numeroProfessor] e [ProfessorNDEISI].[idProfessor] ;

5.1.3. Na tabela de [Grupo], o mesmo aluno não pode aparecer em dois grupos

5.2. Eliminar dados repetidos nas tabelas carregadas.

É preciso verificar se a eliminação provoca erros de integridade referencial (ex.: Eliminar um aluno duplicado deixa uma inscrição a identificar um aluno que deixou de existir)

A eliminação de duplicados, implica nomear uma das ocorrências como principal, substituindo todas as outras ocorrências (duplicadas) por esta antes de proceder à eliminação

Para este efeito pode ser adicionada uma coluna – [id_temp] – com um identificador temporário, que vai distinguir todas as linhas. A eliminação pode ser efectuada com recurso a *subquery* com *GROUP BY* para detecção de repetições, sendo depois eliminado o [id_temp] mais elevado para cada identificador repetido.

MySQL e o SQL Server possuem um DELETE JOIN, que com o identificador único permite eliminar as repetições.

A coluna [id_temp] também pode ser preenchida programaticamente utilizando variáveis de sessão;

5.3. Verificar a coerência nas referências entre tabelas, garantindo que se poderá estabelecer relações sem comprometer integridade referencial:

5.3.1. Todos os valores de [idTFC] em qualquer tabelas têm que existir em [TFC].[idTFC]

5.3.2. Todas as referências a orientadores têm que existir em
[ProfessorDEISI].[numeroProfessor]

5.3.3. Todas as referências a alunos têm que existir em [Aluno].[numeroAluno]; o mesmo com identificação de grupo e [Grupo].[id]

Sempre que os valores da tabela dependente não existirem, devem ser eliminados

Etapa 2

6. Criar chaves primárias para a tabelas do modelo

7. Inserir a coluna [Aluno].[ECTS], de tipo inteiro, preenchimento obrigatório e com DEFAULT 0

Preencher a coluna aleatoriamente para, pelo menos, 5 alunos de modo que se possa responder aos exercícios da 3ª etapa

8. Para os mesmo cinco alunos utilizados na alínea anterior e para o mesmo efeito, preencher avaliações – tabela [AvaliacaoDisciplinaAluno] – para as disciplinas referenciadas no TFC

Deve existir, pelo menos, um caso de aluno com aprovação a todas as disciplinas e outro caso onde o aluno não tenha aprovação a, pelo menos, uma das disciplinas

9. Acrescentar as restrições de integridade – UK, NN, CH – que entender necessárias tendo em conta o modelo apresentado e o domínio.

Ex.:

- Ano de datas de registo têm que ser coerente com o ano lectivo
- Ordem de escolha tem que estar entre 1 e 6
- Avaliações têm que ser número inteiro positivo, igual ou inferior a 20

10. Em [TFC] e [Inscricao] existem colunas autónomas para identificar aluno ou grupo, o que pode criar falha de coerência e integridade de dados

Defina uma regra que garanta que há sempre um, e apenas um, destes atributos preenchido.

Esta regra pode ser aplicada por CONSTRAINT ou com alterações ao modelo

Sugestão: Ler com atenção o restante enunciado para definir a melhor solução

11. Adicionar chaves estrangeiras (FK) indicadas no modelo dados

Ter em conta que os dados têm que estar previamente corrigidos para que as FK possam ser criadas, i.e., terá que se garantir a integridade referencial dos dados antes de adicionar as FK

12. O modelo de dados utiliza chaves *IDENTITY* com *auto incremento*, o que reduz a qualidade do modelo
Alterar o modelo de modo que, sempre que possível, sejam utilizadas **chaves naturais** para as entidades do modelo

Ex.: Em [TFC], passar a PK para [idTFC]

Em [Aluno], passar a PK para [numeroAluno]

Sugestão: De modo a poder fazer a transformação indicada deve-se ter índices nas colunas que passarão a ser PK

13. Como se pode verificar no modelo, a tabela [Aluno] não tem a coluna [estado]

Adicionar a coluna à tabela e preenchê-la de acordo com a seguinte regra:

Se o aluno tem tema atribuído, o estado é {válido}, caso contrário será {excluído}

14. Fortalecer a normalização do modelo, no sentido que este fique em FN3

Sugestões:

- Criar tabela(s) de estados e as respectivas dependências
- Retirar colunas com semântica duplicada e redundante (ex.: [Inscricao].[publicado] é uma redundância de [Inscricao].[estado])
- Sempre que aplicável, nomeadamente para efeitos de desempenho, alterar o tipo de dados
- Retirar valores calculados (ex.: [TFC].[numeroInscricoes])

Etapa 3

15. Como as tabelas de trabalho são extensas os queries podem ser demorados. Para mitigar estas limitação inserir índices necessários para melhorar os queries dos exercícios abaixo;

16. Produza as seguintes pesquisas

Para os *queries* mais complexos devem ser criadas VIEWS, mesmo que não indicado, a serem chamadas com as variáveis indicadas

Sempre que aplicável, desenvolver módulos programáticos para resolução do exercícios, mesmo que não indicado

Ex.	Resultado	Descrição	variáveis
1	Funções de calculo	Para cada um dos atributos com valor calculado removidos no exercício 10, criar uma função para o obter - FUNÇÃO escalar	
2	Candidaturas ordenadas	Devolve lista de inscrições a tema dado, ordenada pelos critérios indicados em (iv).3 O output devem ser linhas com o formato: {TFC, aluno, ordemEscolha, registoSinscricao, ECTS, Média, orientador} - VIEW Onde {Média} indica a avaliação média de cada aluno nas disciplinas associadas ao tema, incluindo disciplinas não avaliadas; No caso de inscrições em grupo, deve-se indicar a média ponderada de ambos os elementos Sugestão: desenvolver uma função para calculo da média	• TFC
3	Lista temas atribuídos	Listar temas já atribuídos num ano lectivo dado O output devem ser linhas com o formato {TFC, aluno, orientadores} - VIEW	• Ano lectivo
4	Lista de orientações	Listar temas atribuídos por orientador O output devem ser linhas com o formato: {docente, TFC, título, aluno} - VIEW Notas: A listagem ter que incluir co-orientações A filtragem só deve ocorrer na chamada à VIEW	• Número de docente
5	Alunos sem trabalho atribuído	Devolve alunos que efectuaram inscrição em, pelo menos, um tema, mas ainda não têm trabalho atribuído Para cada aluno, deve apresentar tantas linhas quanto inscrições Formato {Aluno; TFC; ordemEscolha; estadoTFC; atribuição} Onde [atribuição] deverá indicar aluno/grupo a quem o trabalho foi atribuído ou o literal "Disponível" se não houver atribuição - VIEW	

6	Temas disponíveis	Devolve inscrições não atribuídas onde o tema ainda não tenha sido atribuído e o candidato, aluno ou grupo, ainda não tenha tema atribuído, {TFC, aluno, ordemEscolha, registoScricao, orientador} - VIEW	
7	Lista de trabalhos atribuídos por alunos	Listar temas atribuídos por aluno O output devem ser linhas com o formato: {aluno, TFC, título, orientador} - VIEW	
8	Histórico de tema	Devolve todos os registos referentes a um tema dados, por ordem descendente de cronologia (primeiro registo refere-se à ocorrência mais recente) Formato {TFC; alteração; estado; data; utilizador}	• TFC
9	Contagem de orientações	Lista os docentes do departamento com a respectiva contagem de orientações, co-orientações incluídas Docentes sem orientações também devem ser incluídos	
10	Melhores candidatos	Identificar os 2 (dois) melhores candidatos para um tema dado em função de avaliações e ECTS A resposta deve ser ordenada por média, desempatando por ECTS e registo de inscrição Onde {Média} indica a avaliação média de cada aluno nas disciplinas associadas ao tema, incluindo disciplinas não avaliadas; No caso de inscrições em grupo, deve-se indicar a média ponderada de ambos os elementos Sugestão: desenvolver uma função para calculo da média - FUNÇÃO vectorial	• TFC
11	Alteração de estados com atribuição de temas	Conforme indicado em (iv).4, quando um tema é publicado, o estado passa para “Atribuído”, passando as restantes inscrições ao mesmo tema com o estado “Não Atribuído”. O mesmo com as inscrições do aluno/grupo As mudanças de estado para “Não Atribuído” devem ocorrer automaticamente com a atribuição do tema - TRIGGER	
12	Verificar elegibilidade de aluno para realização de TFC	Dada uma inscrição, verificar se o aluno cumpre os critérios regulamentares para a realização do tema a que se candidata A função deve devolver o literal “válida” se forem cumpridos os critérios ou indicar o critério em falta – ECTS ou disciplina sem aprovação – caso contrário	• Inscrição

		- FUNÇÃO escalar Nota: basta indicar o primeiro critério em falha Sugestão: Utilizar CURSOR	
13	Composição de grupo	Devolver elementos de um grupo dada a sua identificação A função deve devolver apenas um resultado indicando os nomes de ambos os elementos - FUNÇÃO escalar	• Identificação de Grupo
14	Verificar estado de TFC	Dado um TFC, verifica se já foi atribuído. Devolve o nome do(s) executante(s) caso já tenha sido atribuído e o estado, caso contrário - FUNÇÃO escalar	• Número de TFC
15	Alterar atribuição de tema	Dado um tema de TFC, altera a atribuição para um novo executante Necessita de: ○ Alterar a atribuição do tema sem mudar o estado do TFC ○ Alterar o estado a ambos os executantes, o actual (passa a “não atribuído”) e o novo (passa a atribuído) ○ Registrar a alteração no histórico - PROCEDIMENTO com transacção	• Tema • Novo executante (aluno ou grupo)
16	Tema atribuído não pode ser apagado	Verifica o estado de um tema quando ocorre um comando DELETE e impede a eliminação - TRIGGER	• Número de TFC
17	Anular Atribuição de tema	Reverter uma atribuição de tema. É necessário 3 acções: ○ Mudar estado da inscrição; ○ Mudar estado do TFC ○ Registrar em histórico - PROCEDIMENTO com transacção	• Número de TFC

NOTA:

1. As cores indicam grau de dificuldade de cada exercício, do mais **simples** ao mais **exigente**. O grau de dificuldade será reflectido na cotação de avaliação
2. Nos resultados dos *queries*, sempre que se indica {TFC} deve-se apresentar {idTFC;titulo}
3. Nos resultados dos *queries*, sempre que se indica {aluno} deve-se apresentar {numeroAluno;nomeAluno}
4. Quando se indica {aluno}, em inscrição ou atribuição, deve-se considerar a totalidade de executantes do TFC. Sempre que aplicável, devem indicar-se os alunos de um grupo e não alunos de forma individualizada

Sempre que aplicável, pode utilizar-se a função do exercício 11 para obter o nome dos alunos de um grupo

Excepção a esta regra são os exercícios 4 e 6, onde se devem listar alunos individualmente, mesmo que façam parte de um grupo

Regras

Os modelos de dados apresentados anteriormente são de uso obrigatório, devendo ser respeitados os nomes das tabelas e das colunas, assim como os respectivos tipos de dados. A parte 2 do trabalho permite alterações ao modelo, de forma a melhorar a sua qualidade e desempenho. Neste âmbito, todas as alterações são válidas, mas têm que ser elaboradas sem perda de dados nem novas inserções.

Em cada etapa deverá ser entregue um ficheiro único de texto contendo os comandos SQL que implementam o que é solicitado. Esse script deverá utilizar a sintaxe da versão de SQL utilizada nas aulas práticas. Podem ser utilizadas outras sintaxes, mas é necessária validação prévia com docente.

Os comandos do script devem seguir a sequência dos desafios, o que significa que o script da etapa 2 tem que aproveitar os objetos criados na etapa 1 sem reconstrução de elementos, perda de dados ou novas inserções para dados carregados. Por exemplo, se na etapa 1 é criada uma tabela e na etapa 2 é introduzida uma alteração à tabela, terá que resolver o problema recorrendo a um ALTER TABLE, nunca DROP seguido de CREATE.

Cada script deve identificar claramente os membros do grupo e ser profusamente comentado de modo a descrever a abordagem na resolução de cada exercício. Particularmente, os comandos que respondem a um desafio devem ser precedidos por um comentário que identifique o respectivo número de desafio. A ordem dos desafios também tem que ser respeitada, porque segue uma sequência lógica. Um desafio não respondido deve ter na mesma o respectivo comentário identificativo acrescido da indicação de não resolução.

Todos os comandos devem ser testados no respetivo SGBD, pelo que não podem ocorrer erros na sua execução. Quando os desafios estão encadeados, um erro na etapa precedente anula todas as que dela dependem. Por exemplo, se tiver um erro na criação das tabelas, não será possível avaliar a sua alteração, inserção de dados nem as suas consultas.

Erros na resolução de qualquer das fases pode ser corrigido nas entregas posteriores, possibilitando que se possam realizar os exercícios encadeados.

Cotações

Etapa	Cotação
1	25%
2	35%
3	40%

Podendo haver correcções entre fases, o resultado final da avaliação será apenas o da última entrega, mantendo-se as cotações indicadas na tabela.

Nestes termos, em cada etapa terão que ser entregues todos os scripts das fases anteriores, mesmo que não alterados. A avaliação pressupõe que todos os exercícios possam ser executados pela ordem indicada no enunciado

Para as etapas 1 e 2 será disponibilizada avaliação qualitativa do trabalho entregue, mas sem valorização

Formato de entrega e prazo

Os scripts de resolução deverão ser submetidos em Moodle nas datas indicadas no calendário da disciplina

Cada grupo deverá entregar um, e apenas um, ficheiro em cada etapa com todos os exercícios que tenha resolvido, incluindo os das etapas anteriores mesmo que não alterados

Não é solicitado relatório formal, embora se permita entrega deste documento a quem o decida fazer. Por outro lado, independentemente de entrega de relatório, os scripts entregues devem identificar os autores e ser profusamente comentados, nomeadamente na identificação dos exercícios e na indicação de abordagens à resolução.

Scripts não identificados serão excluídos de avaliação.

Anexo 1: Descrição de domínio e Input

O *input* do programa consiste em **16** ficheiros em formato CSV, com dados correspondentes a cada uma das tabelas do modelo de dados:

Ficheiro	Conteúdo
Tfc.csv	Lista de TFC ao longo dos anos em que a plataforma foi utilizada
Inscrição.csv	Candidaturas de alunos a temas de TFC
HistoricoTFC.csv	Registo histórico (log) de alterações a TFC
ProfessorDEISI.csv	Lista de professores dos cursos, onde se incluem orientadores
ProfessorNDEISI.csv	Lista de professores de outros departamentos da ULHT
Empresa_EntidadeExterna.csv	Lista de empresas e parceiros para a realização de TFC
Aluno.csv	Alunos candidatos a TFC
Grupo.csv	Grupos candidatos a TFC
AvaliacaoDisciplinaAluno.csv	Resultados de alunos nas disciplinas indicadas como obrigatórias para a realização de um determinado tema de TFC
Curso.csv	Lista de cursos com TFC onde a gestão é realizada na plataforma
Tecnologias.csv	Lista de tecnologias utilizadas em TFC
Disciplinas.csv	Lista de disciplinas consideradas como obrigatórias para temas de TFC
TFCCurso.csv	Associação de temas de TFC a curso
TFCTecnologia.csv	Associação de tecnologias a temas de TFC
TFCDisciplina.csv	Associação de disciplinas a temas de TFC

Não são conhecidas, à priori, as dimensões (número de linhas) destes ficheiros.

Podendo haver erro com caracter especial nos dados originais, em virtude de diferença de idioma ou de *codepage*, deve-se proceder à verificação dos documentos disponibilizados e à correcção ortográfica dos mesmos por forma a garantir a legibilidade dos dados importados