GUIA DE LABORATÓRIO 1.2 (Versão ß)
PROPRIEDADES E RELAÇÕES

OBJECTIVOS

- Criar colunas com várias propriedades: valor por omissão, restrições, chave- estrangeira, etc.
- Redefinir as tabelas da BD `ENSINO` e criar uma nova tabela: `EstudanteTrabalhador`

INSTRUÇÕES

1a Parte – Redefinição das tabelas

1. Inicie o HeidiSQL da forma habitual .

2. Redefina a tabela `Curso` de acordo com as instruções dos laboratórios anteriores e com a informação extra presente na tabela seguinte:

Campo	Propriedades
`ID`	Inalterado
`Nome`	Texto, 100 caracteres Obrigatório Comentário: `'Designação oficial do curso'`
`Duracao`	Inteiro pequeno Obrigatório Restrição: valor positivo Comentário: `'Duração em horas'`
`Tipo`	Texto, 50 caracteres Obrigatório Comentário: `'Tipo de curso'`

Agora, crie a tabela Accao. Esta tabela tem uma relação de 1 ↔ * com Curso.

Campo	Propriedades
`ID`	Chave-primária Inteiro, numeração automática
`Numero`	~~Deixa de ser chave-primária~~ (para acomodar diferentes acções com o mesmo número) Tipo de dados inalterado ~~Mantém-se~~ obrigatório

Campo	Propriedades
`DataInicial`	Data Obrigatório
`DataFinal`	Data
`Coordenador`	Texto, 250 caracteres
`IDCurso`	Chave-estrangeira para `Curso.ID`; alterações deverão ser propagadas Obrigatório
Restrição: Par `(Numero, IDCurso)` deve ser único	

Além do tipo de dados, é comum querermos associar determinadas propriedades a um campo. Isto permite que a BD verifique se os valores que estamos a inserir estão de acordo com essas propriedades ou não, emitindo um erro e impedindo a inserção de dados se tal não se verificar. A linguagem SQL permite especificar várias dessas propriedades através de alguns "parâmetros" que podemos acrescentar à definição de uma coluna.

Normalmente, as propriedades mais comuns são:
- Chave-primária: campo único e obrigatório que identifica uma entrada (ie, uma linha) na tabela; se for inteiro poderá possuir a propriedade ***`AUTO_INCREMENT`****; MySQL ☒* ***`PRIMARY KEY`***

- Chave-estrangeira: campo cujo valor deverá ser uma chave-primária numa outra tabela; Esta propriedade deve ser definida numa linha à parte.
MySQL ☒ ***`[CONSTRAINT`*** *nome* ***`] FOREIGN KEY`*** *[idx_nome]* ***`REFERENCES`*** *(<campo>...) <tabela c/ chave-primária> (<campo chave-primária>...)*

- Único: o valor a inserir numa determinada coluna não deverá ser repetido; MySQL ☒ ***`UNIQUE [KEY]`*** *[nome]*

- Obrigatório: temos que especificar um valor para essa coluna, ie, não pode ficar vazia: MySQL ☒ ***`NOT NULL`***

- Valor por omissão: se o campo não for obrigatório e um valor não for especificado, podemos atribuir um valor constante ou calcular um valor a partir de uma expressão: MySQL ☒ ***`DEFAULT`*** *0,* ***`DEFAULT`*** *1,* ***`DEFAULT`*** *'Bla, Bla' ,* ***`DEFAULT (CURRENT_DATE), DEFAULT (SALDO * JURO)`***

- Restrição: o valor deverá obedecer a uma determinada propriedade ad-hoc ou deverá pertencer a uma gama de valores. Quando fazendo parte da linha de definição de um campo a sintaxe é a seguinte: ***`CHECK(Expressao)`***
Exemplos: MySQL ☒ ***`CHECK (Coluna > 0) , CHECK(Date > '2021-05-10')`***
Esta restrição pode ser definida numa linha à parte. Neste caso podemos também associar um nome a esta restrição. Neste caso síntaxe é a seguinte:
`[CONSTRAINT` *nome]* ***`CHECK (Expressao) [[NOT] ENFORCED]`***
Até ao MySQL 8.0.16 esta propriedade era ignorada. A partir do MySQL 8.0.17 esta propriedade passou a ser validade pelo MySQL, a não ser que indiquemos a cláusula ***`NOT ENFORCED`****.*

- Comentário: Uma descrição da coluna que deverá preceder as outras propriedades;
MySQL ☒ ***`COMMENT 'Este campo é uma seca!'`***

Para mais informações consultar:
- https://dev.mysql.com/doc/refman/8.0/en/create-table.html
- https://dev.mysql.com/doc/refman/8.0/en/create-table-foreign-keys.html

3. Redefina a tabela **`Estudante`** de acordo com as instruções do laboratório anterior e com a informação extra presente na tabela seguinte.

Campo	Propriedades
`ID`	Inalterado
`Nome`	Substitui `NomeEstudante` Texto, 150 caracteres Obrigatório Comentário: `'Nomes pessoais'`
`Apelido`	Texto, 250 caracteres Obrigatório Comentário: `'Nomes de família'`
`Endereco`	Texto, 200 caracteres Obrigatório Comentário: `'Endereço excepto cidade e código postal'`
`Cidade`	Texto, 50 caracteres Obrigatório Valor por omissão: `'Lisboa'`
`CodigoPostal`	Numérico Obrigatório Restrição: valores entre 3000 e 4999 Em MySQL, crie esta restrição numa linha própria (ie, não a adicione à definição do campo) e dê-lhe um nome (`CodigoPostalCHK`). Mais à frente vamos perceber os motivos por trás deste requisito.
`DataNascimento`	Data Obrigatório
`NISS`	Inteiro Único Obrigatório Restrição: inteiro positivo Comentário: `'Número de Identificação da Segurança Social'`

4. Finalmente, a tabela `Inscricao`, que implementa a relação 0..* ↔ 1..* entre **`Estudante`** e **`Accao`**.

Campo	Propriedades
`ID`	Chave-primária Inteiro, numeração automática
`IDEstudante`	Chave-estrangeira para `Estudante.ID`; alterações deverão ser propagadas Obrigatório
`IDAccao`	Chave-estrangeira para `Accao.ID`; alterações deverão ser propagadas Obrigatório
`DataInscricao`	Data Valor por omissão: ~~`CURRENT_DATE`~~ **CURDATE()**

Campo	Propriedades
`Estado`	Texto 10 caracteres Obrigatório Um de: 'activa', 'suspensa', 'concluida' Valor por omissão: 'activa'
`ClassificacaoFinal`	Numérico com precisão Precisão: 4, Escala: 2 Restrição: números entre 0 e 20. Em MySQL, crie esta restrição numa linha própria (ie, não a adicione à definição do campo) e dê-lhe um nome (`ClassificacaoCHK`). Mais à frente vamos perceber os motivos por trás deste requisito.
Restrição: Par (`IDEstudante, IDAccao`) deve ser único	

5. Confirme que as tabelas criadas obedecem a estas especificações.

2a Parte – Re-introdução de dados

6. Introduza os seguintes cursos:

Nome	Duracao	Tipo
Agricultura Aplicada	1500	Ciências Agrárias
Biologia	1600	Ciências da Vida
Contabilidade	1700	Ciências Económicas e Fiscalidade

7. Introduza as seguintes acções:

Curso	Numero	DataInicial	DataFinal	Coordenador
Agricultura Ap.	1	02-07-2019	02-09-2021	Alberto Antunes
Agricultura Ap.	2	02-08-2021	02-04-2023	Alberto Antunes
Biologia	1	02-02-2021	02-12-2022	Armando Almeida
Contabilidade	1	02-09-2021	02-12-2022	Arnaldo Alves
Contabilidade	2	02-11-2021	02-02-2023	Arnaldo Alves

8. Volte a inserir todos os estudantes, atribuindo-lhes o `NISS` e ligando-os às acções apropriadas de acordo com a seguinte tabela:

Nome	Curso Accao	Data Inscricao	ClassificacaoFinal	NISS
António	Agricultura Ap. 2	15-05-2019	16	1111
Beatriz	Biologia 1	12-12-2020	15	1112

Nome	Curso Accao	Data Inscricao	ClassificacaoFinal	NISS
Catarina	Biologia 1	10-12-2020	8	1113
Diogo	Biologia 1	14-12-2020	7	1114
Eduardo	Agricultura Ap. 2	10-05-2019	18	1115
Filipa	Contabilidade 1	01-08-2021	17	1116

3a Parte – Criação da tabela `EstudanteTrabalhador`

9. Crie a tabela **`EstudanteTrabalhador`** cuja definição é apresentada na tabela seguinte. A tabela **`Estudante`** deverá ter uma relação de 1 ↔ 0..1 com esta tabela.

Nome	Propriedades
`IDEstudante`	Chave-primária. Chave-estrangeira para `Estudante.ID` c/ propagação de actualizações e remoção
`NIF`	Inteiro Único Obrigatório Restrição: valores com 6 dígitos Comentário: `'Número de identificação fiscal'`
`DataEstatuto`	Data Obrigatório Comentário: `'Data de início do estatuto de estudante trabalhador'`
`NumExames`	Inteiro pequeno Obrigatório Valor por omissão: 1 Comentário: `'Número de exames em época especial a que o estudante tem direito'`
`Profissao`	Texto, 20 caracteres Valor por omissão: `'N/D'`

10. Os seguintes estudantes são trabalhadores:

Nome	NIF	DataEstatuto	NumExames	Profissao
Catarina	2324540	2020-02-12	3	Restauração
Diogo	345761	2020-01-02	5	Entretenimento
Filipa	434456	2020-02-02	4	Programador

4a Parte - Testar integridade referencial e propagações automáticas

11. Verifique qual a chave-primária da Filipa e confirme que coincide (como seria de esperar) com sua chave-primária na tabela **`EstudanteTrabalhador`.**

12. Modifique o valor da chave primária da Filipa através do seguinte comando:

```
UPDATE Estudante SET ID = 10 WHERE ID = 6;
```
NOTA: Assumindo que é este o ID da Filipa

13. Verifique novamente a chave-primária da Filipa em ambas as tabelas.

14. Apague a Filipa das tabelas **Inscricao** e **Estudante** e de todas as tabelas relacionadas através dos seguintes comandos SQL:

```
DELETE FROM Inscricao WHERE IDEstudante = 10;
DELETE FROM Estudante WHERE ID = 10;
```

15. Confirme que a Filipa foi apagada de ambas as tabelas.

16. Insira novamente a Filipa na tabela **Estudante** e, depois, na tabela **EstudanteTrabalhador**. Reponha também as inscrições da Filipa.

5a Parte - Mais sobre o comando ALTER TABLE

17. Suponhamos que pretendemos renomear a tabela **Estudante** para **Aluno**. Podemos utilizar o comando ALTER TABLE para tal:

```
ALTER TABLE Estudante RENAME Aluno
```

18. Reponha o nome **Estudante** renomeando a tabela **Aluno**.

19. Vamos alterar o tipo de dados do atributo NumExames da tabela **EstudanteTrabalhador** de **SMALLINT** para **TINYINT**. Para além disso, doravante esse atributo deixará de ser obrigatório.

```
ALTER TABLE EstudanteTrabalhador MODIFY NumExames TINYINT NULL;
```

20. Vamos agora renomear para CodPostal o atributo CodigoPostal da tabela Estudante. Como o campo está "sob" uma restrição, temos que remover temporariamente a restrição, renomear o campo, e voltar a adicionar a restrição. Os comando ALTER TABLE são os seguintes:

```
ALTER TABLE Estudante DROP CONSTRAINT CodigoPostalCHK;
ALTER TABLE Estudante RENAME COLUMN CodigoPostal TO CodPostal;
ALTER TABLE Estudante ADD CONSTRAINT CodigoPostalCHK
```

```
CHECK(CodPostal BETWEEN 3000 AND 4999);
```

21. Volte a dar o nome `CodigoPostal` ao atributo `CodPostal` e modifique o tipo para `SMALLINT`.

EXERCÍCIOS

1. Desenhe e defina em SQL os modelos de dados para as situações descritas. Insira alguns dados para testar os modelos de dados.

1.1 Um produto, identificado pelo ID, e possuindo nome e quantidade, pertence a uma e uma só categoria. Uma categoria, que abrange vários produtos, é identificada pelo número de categoria e possui designação e descrição, pode abranger vários produtos.

Nome	Propriedades
`ID`	Chave-primária. Texto, 10 caracteres
`Nome`	Texto, 20 caracteres Único Obrigatório Comentário: `'Nome do produto'`
`Quantidade`	Inteiro pequeno Obrigatório Restrição: `valor >= 0` Valor p/ omissão: `0` Comentário: `'Quantidade em stock'`
`Numero`	Chave-primária Inteiro
`Designacao`	Texto, 20 caracteres Obrigatório Único Comentário: `'Nome único da categoria'`
`Descricao`	Texto, 100 caracteres

1.2 Um evento, identificado pelo ID, e possuindo título, descrição e data/hora, é enviado a vários convidados. Um convidado, identificado pelo ID, e possuindo primeiro e último nomes, endereço de email e data de nascimento, pode receber vários convites. Deve ficar registado (de alguma forma...) a data de envio do convite.

Nome	Propriedades
`ID` (todas as tabelas)	Chave-primária. Inteiro

Nome	Propriedades
`Titulo`	Texto, 20 caracteres Único Obrigatório Comentário: `'Designação do evento'`
`Descricao`	Texto, 1000 caracteres Obrigatório Comentário: `'Dados extra sobre o convite'`
`DataHora`	Data/hora Comentário: `'Data e hora do evento'`
`PrimNome`	Texto, 100 caracteres Obrigatório Comentário: `'Primeiro nome pessoal'`
`UltNome`	Texto, 100 caracteres Comentário: `'Último apelido'`
`EndEmail`	Texto, 150 caracteres Obrigatório Restrição: possui uma e uma só '@'
`DataNascimento`	Data Obrigatório

2. Altere o nome do campo `ClassificacaoFinal` da tabela **`Inscricao`** para `Classificacao`. Doravante, este campo irá indicar a classificação actual de cada estudante. Quando o estudante concluir a acção, então o valor neste campo passará a ser a sua classificação final.

Insira valores apropriados para os estudantes actuais.

NOTA: Deverá primeiro remover todas as restrições referentes a este campo. Depois introduz o comando para alterar o nome e, finalmente, repõe a restrição para este campo e ao nível da tabela.

3. Investigue como pode eliminar chaves-estrangeiras e chaves-únicas sem remover a respectiva tabela.

4. Considere uma aplicação para gestão de projectos. Um projecto é identificado por um código de projecto e possui descrição, data inicial e data final. Cada projecto possui pelo menos um membro e um membro participa em apenas um projecto.

Cada membro possui um código de membro, nome, datas de entrada e de saída do projecto e email. Existem 3 tipos de membros: gestor de projecto, coordenador técnico e engenheiro. Para os gestores é guardado o email do responsável pelo projecto por parte do cliente. Para os coordenadores técnicos guarda-se a área técnica que ele está a coordenar e um campo que indica se o coordenador está certificado nessa área ou não. Para os engenheiros é guardada a especialidade.

Um projecto possui várias tarefas, sendo que estas pertencem apenas a um só projecto. Cada tarefa pode estar atribuída a vários membros e cada membro pode trabalhar em várias tarefas. Uma tarefa é identificada por um código de tarefa e possui um campo com o estado da tarefa.

4.1 Elabore o modelo de dados da BD.

4.2 Implemente a BD tendo em atenção a seguinte tabela com as propriedades e tipos dos dados do atributos:

Atributo	Tipo	Propriedades
código de projecto	texto	10 caracteres
descrição do projecto	texto	50 caracteres
data de início do projecto	data/hora	obrigatório
data final do projecto	data/hora	obrigatório
código do membro	número automático	inteiro
nome de membro	texto	30 caracteres obrigatório único
data de entrada	data/hora	obrigatório
data de saída	data/hora	poderá ser nulo
email do membro	texto	30 caracteres obrigatório
email do responsável	texto	30 caracteres obrigatório
área técnica do coordenador	texto	20 caracteres obrigatório
certificado	sim/não	obrigatório valor pre-definido: "não"
especialidade	texto	20 caracteres obrigatório valor pre-definido: "software"
código de tarefa	número automático	inteiro
estado da tarefa	texto	obrigatório valor pré-definido: "por_iniciar" regra de validação: um de ("por_iniciar", "iniciada", "bloqueada", "terminada", "suspensa")

4.3 Popule a BD com: 2 projectos, 2 engenheiros para um projecto, 2 para outro e 1 gestor e 1 coordenador para cada. Acrescente 3 tarefas a cada projecto e faça com que, pelo menos, uma tarefa seja partilhada pelos 2 engenheiros de cada projecto e cada um deles tenha mais do que uma tarefa.