 POLITÉCNICO DE LEIRIA ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO	Instituto Politécnico de Leiria Escola Superior de Tecnologia e Gestão Departamento de Engenharia Informática
	Curso Técnico Superior Profissional de Programação de Sistemas de Informação Ano Letivo 2022/23 Conceção e Desenvolvimento de Bases de Dados 1.º ano 2.º Semestre

Projeto de Base de Dados

grupo 1 Base de dados de inscrições em exames

Versão 1.0 / 9 de Maio de 2023

Autores:

Estudante Cláudio Moreira, n.º 2220881

Estudante Hugo Gonçalves, n.º 2220891

Estudante Rafael Teodoso, n.º 2220886

1. Introdução

O presente documento apresenta uma Análise de Dados realizada sobre o cenário de uma base de dados de inscrição a exame. O documento foi realizado no âmbito da Unidade Curricular de Conceção e Desenvolvimento de Bases de Dados do curso Tecnológico Superior Profissional de Programação em Sistemas de Informação.

Pretende-se criar uma BD de Inscrições em exames, como tal é necessário guardar alunos, docentes, UC, escola, curso, inscrição, exame e vigias.

Escolhemos este tema pois como estudantes do Politécnico de Leiria frequentemente estamos contacto com a realização de exames e é um tema que estamos bastante familiarizados.

2. Descrição do Sistema

O Programa baseia-se na Inscrição de Exames. Onde o Programa armazena os dados dos alunos numa base de dados.

A Base de dados criada permite com que os dados dos alunos sejam divididos em diversas categorias para o aluno ser identificado corretamente assim facilitando a sua inscrição para os diversos Exames.

- A universidade tem vários cursos e cada curso é composto por várias UCs, estando uma UC associada a um único curso.
- Cada docente pode lecionar várias UCs e cada UC pode ser lecionada por vários docentes.
- Uma pessoa vai-se decompor nas seguintes categorias: aluno e/ou docente, pois pode ser aluno e docente em simultâneo.
- Um aluno pode realizar uma inscrição de curso e cada inscrição tem de ser realizada por um único aluno.
- Um aluno pode realizar um registo de uma UC e cada registo UC tem de ser realizada por um aluno.
- Um aluno pode realizar um registo de um exame e cada registo exame tem de ser realizada por um aluno.
- Cada inscrição de curso tem de estar associada um curso e cada curso pode ter uma ou várias inscrições.
- Cada aluno pode-se registar em várias UCs e uma uc pode ter vários registos.

- Cada registo de exame está associado obrigatoriamente a um exame e cada exame pode ter vários registos.
- Cada Registo UC tem uma UC.
- Cada Inscrição_curso tem um curso.
- Cada exame tem uma UC

Questões que serão importantes saber:

Q1: Quantos alunos estão inscritos em cada curso?

Q2: Quantos cursos tem cada UC?

Q3: Quantas UCs tem a universidade?

Q4: Quantas UCs leciona cada docente?

Q5: Quantos docentes leciona cada UC?

Q6: Quantos alunos estão inscritos em cada exame?

Q7: Quantos exames está cada aluno inscrito?

3. Modelo Conceptual

Este capítulo apresenta as várias componentes do Modelo Conceptual da base de dados a construir. Do Modelo Conceptual fazem parte o Diagrama de Entidade-Relacionamento (Figura 2) e ainda descrição pormenorizada das características de cada atributo de cada entidade (Tabela 1).

3.1. Descrição das entidades e dos relacionamentos

Esta secção descreve os atributos de cada entidade e as características de cada um. As entidades surgem por ordem alfabética do seu nome.

Entidade pessoa				
Atributo	Descrição	Domínio	Preenchimento Obrigatório	Chave Primária
Id	Identificador da pessoa	Número inteiro positivo	X	X
Nome	Nome da pessoa	Até 100 caracteres; Mínimo de 2 letras	X	
Morada	Observações a registar sobre a categoria	Até 500 caracteres		
Telemóvel	Número de telemóvel	Número inteiro até 9 caracteres	X	

Email	Email Eletrônico	Até 100 caracteres; Mínimo de 4 letra; Possui sempre um "@" e um "."	X	
Nif	Número de Identificação Fiscal	Número inteiro de 9 caracteres	X	
DtaNascimento	Data de Nascimento	DateTime	X	

Entidade aluno (herda de pessoa)				
Atributo	Descrição	Domínio	Preenchimento Obrigatório	Chave Primária
Estatuto	Estatuto do aluno	Até 100 caracteres; Mínimo de 2 letras	X	
Necessidades_esp	Necessidades especiais do aluno	Booleano		

Entidade docente (herda de pessoa)				
Atributo	Descrição	Domínio	Preenchimento Obrigatório	Chave Primária
Habilitações	Habilitações Profissionais do docente	Até 300 caracteres; Mínimo de 2 letras	X	
Cacifo	Cacifo do docente	Número inteiro até 3 caracteres		

Tabela 1 - Descrição das caraterísticas das entidades.

Entidade UC				
Atributo	Descrição	Domínio	Preenchimento Obrigatório	Chave Primária
Id	Identificador da unidade Curricular	Número inteiro positivo	X	X
DescricaoUC	Nome da UC	Até 300 caracteres; Mínimo de 2 letras	X	

Sigla	Sigla da UC	Até 100 caracteres; Mínimo de 2 letras	X	
ETCS	Sistema Europeu de Transferência e Acumulação de Créditos	Número inteiro até 2 caracteres	X	
Ano	Ano da UC	enum ("1", "2", "3")	X	
Semestre	Semestre da UC	enum ("1", "2")	X	

Entidade curso				
Atributo	Descrição	Domínio	Preenchimento Obrigatório	Chave Primária
Id	Identificador do curso	Número inteiro positivo	X	X
DescricaoCurso	Nome do Curso	Até 100 caracteres; Mínimo de 3 letras	X	
Sigla	Sigla do curso	Até 100 caracteres; Mínimo de 2 letras	X	

Entidade Registo_UC				
Atributo	Descrição	Domínio	Preenchimento Obrigatório	Chave Primária
id	Identificador do registo da UC	Número inteiro positivo	X	X
Status	Status da UC	Enum ("aprovado", "não aprovado")	X	
ano_letivo	Ano letivo da UC	Número inteiro até 4 caracteres	X	
data	Data do registo da UC	DateTime ("data atual")	X	

Entidade Inscricao_curso				
Atributo	Descrição	Domínio	Preenchimento Obrigatório	Chave Primária
id	Identificador da inscrição do curso	Número inteiro positivo	X	X

valor	Valor monetário da inscrição no curso	Número inteiro até 300	X	
data	Data da inscrição do curso	DateTime (“data atual”)	X	

(continuação) Tabela 1 - Descrição das características das entidades.

Entidade Registo_exame				
Atributo	Descrição	Domínio	Preenchimento Obrigatório	Chave Primária
id	Identificador do registo de exame	Número inteiro positivo	X	X
nota	Nota do Exame	Número inteiro de 0 até 20	X	
valor	Valor monetário da inscrição no Exame	Número inteiro até 100	X	
data	Data do Exame	Date time (“data atual”)	X	

Entidade exame				
Atributo	Descrição	Domínio	Preenchimento Obrigatório	Chave Primária
Id	Identificador de exame	Número inteiro positivo	X	X
Época	Época do exame	Até 100 caracteres; Mínimo de 3 letras	X	
Hora inicio de exame	Hora inicio de exame	Datetime	X	
Hora final de exame	Hora final de exame	Datetime	X	
Duração	Duração do exame	Float	X	
Nome da UC	Nome da UC	Até 100 caracteres; Mínimo de 3 letras	X	

Entidade DSD_Docente				
Atributo	Descrição	Domínio	Preenchimento Obrigatório	Chave Primária
Id	Identificador de DSD	Número inteiro positivo	X	X
data	data do ano letivo	Datetime	X	

(continuação) Tabela 1 - Descrição das características das entidades.

3.2 Diagrama de Entidade-Relacionamento

O diagrama de Entidade-Relacionamento apresentado nesta secção (Figura 2) representa os dados e relacionamentos entre dados do sistema. A simbologia usada no diagrama respeita os formalismos lecionados nas aulas de Conceção e Desenvolvimento de Bases de Dados do ano letivo 2016/2017. Uma legenda desta simbologia é apresentada na Figura 1.

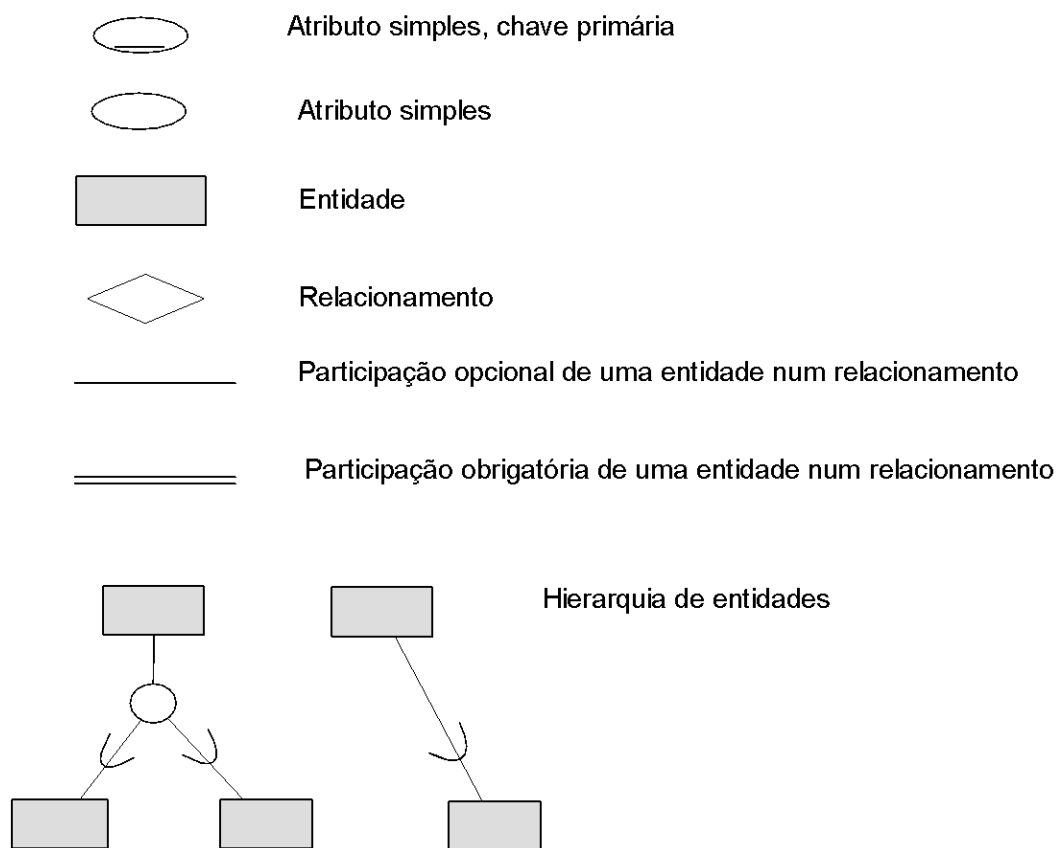


Figura 1 - Legenda da simbologia usada no Diagrama de Entidade-Relacionamento do presente relatório.

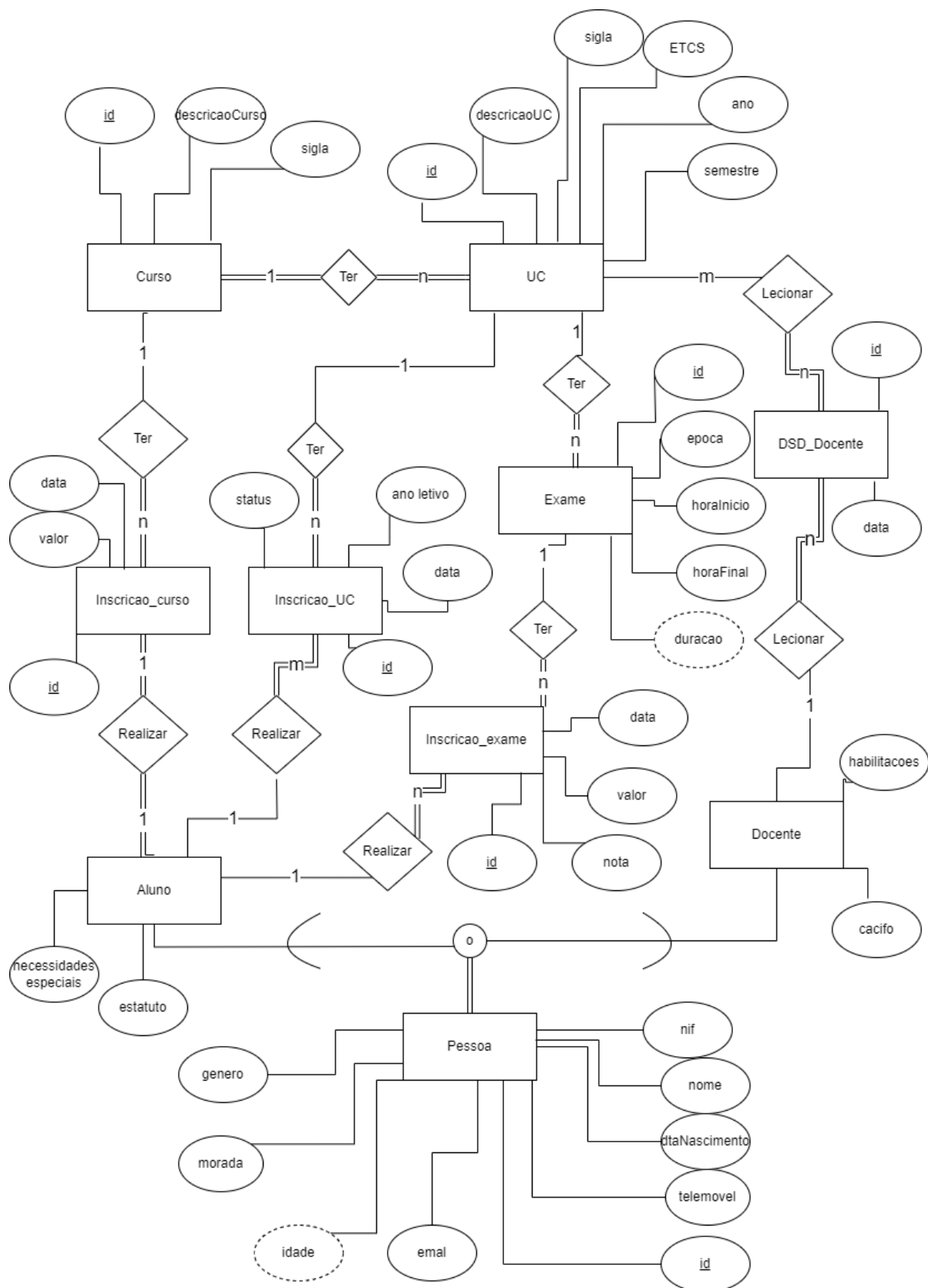


Figura 2 - Diagrama de Entidade-Relacionamento da base de dados.

4. Modelo Lógico

Este capítulo apresenta o Modelo Lógico derivado a partir do Modelo Conceptual da base de dados.

A derivação do Modelo Lógico a partir do Modelo conceptual foi realizada por aplicação das regras de transformação lecionadas na Unidade Curricular de Conceção e Desenvolvimento de Bases de Dados.

O Modelo Lógico segue a seguinte abordagem de representação:

- As chaves primárias das tabelas são representadas com estilo **negrito e sublinhado**.
- As chaves estrangeiras das tabelas são representadas com estilo *itálico*.
- A representação da chave primária correspondente a cada chave estrangeira é feita por intermédio de uma seta dirigida para o nome da tabela onde se encontra essa chave primária.

DSD_Docentes(**id**, *idDocente=>Docente(id)*, data)

UC_DSD_Docentes(**idUC**=>UC(id), **idDSD_Docente**=>DSD_Docentes(id))

Cursos(**id**, descricaoCuso, sigla)

Inscricoes_UCs(**id**, *idAluno=>Aluno(id)*, *idUC=>UC(id)*, data, ano_letivo, status)

UCs(**id**, *idCurso=>Curso(id)*, descricao, sigla, ETCS, ano, semestre)

Exames(**id**, *idUC=>Curso(id)*, epoca, horaInicio, horaFinal)obs: duracao será criada numa vista

Inscricoes_exames(**id**, *idExame=>Exame(id)*, *idAluno=>Aluno(id)*, nota, valor, data)

Alunos(**id**=>Pessoa(id), necessidades_especiais, estatuto, data_inscricao, valor_inscricao)

Docentes(**id**=>Pessoa(id), habilitacoes, cacifo)

Pessoas(**id**, nome, nif, dtaNascimento, telemovel, emai, genero, morada)obs: idade será criada numa vista

5. Permissões de acesso

Este capítulo apresenta as permissões de acesso aos dados associadas a cada tipo de utilizador final da base de dados. As tabelas seguintes resumem as permissões de consulta, de inserção, de eliminação e de atualização de dados que cada perfil de utilizador possui sobre a base de dados.

Operação	Regra	CLIENTE
Consultar	C1	Permissão: Consultar produtos Colunas: Produtos.nome, Produtos.preço, Produtos.marca
	C2	Permissão: Consultar categorias de produtos Colunas: Categorias.nome

Tabela 3 – Regras de acesso a dados para utilizadores do perfil "CLIENTE".

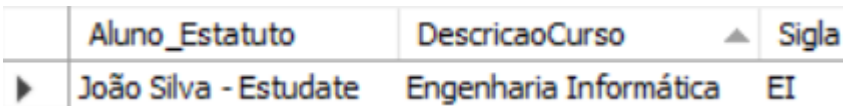
Operação	Regra	VENDEDOR
Consultar	C1	Permissão: Consultar produtos Colunas: Produtos (id, nome, preço, marca, cod barras, obs)
	C2	Permissão: Consultar categorias Colunas: Categorias (nome, obs)
Inserir	I1	Permissão: Inserir novas vendas Colunas: Vendas (data, obs)
	I2	Permissão: Inserir vendas de produtos Colunas: Produtos vendidos (id produto, preco unitario, numero unidades)
Atualizar	A1	Permissão: Atualizar quantidades vendidas de cada produto Colunas: Produtos vendidos (numero unidades)

Tabela 4 – Regras de acesso a dados para utilizadores do perfil "VENDEDOR".

6. Consultas à base de dados

Este capítulo apresenta as consultas programadas para responder a questões operacionais relevantes.

6.1 - Mostra os alunos no curso com sigla EI

Justificação	Código SQL
Mostra os alunos no curso com sigla EI	<pre>SELECT CONCAT(p.Nome, ' - ', a.Estatuto) AS Aluno_Estatuto, c.DescricaoCurso, c.Sigla FROM Pessoa p JOIN Aluno a ON p.Id = a.Id JOIN Curso c ON a.Id = c.IdAluno WHERE c.Sigla = 'EI' ORDER BY p.Nome;</pre>
Output	
	


6.2 - Conta o número de estudantes em cada Curso

Justificação	Código SQL																		
Conta o número de estudantes em cada Curso	<pre>SELECT c.DescricaoCurso AS Curso, COUNT(a.Id) AS NumStudents FROM Curso c JOIN Aluno a ON c.IdAluno = a.Id GROUP BY c.DescricaoCurso ORDER BY NumStudents DESC;</pre>																		
Output																			
	<table><tr><th>Curso</th><th>NumStudents</th></tr><tr><td>Engenharia Informática</td><td>1</td></tr><tr><td>Gestão de Empresas</td><td>1</td></tr><tr><td>Arquitetura</td><td>1</td></tr><tr><td>Engenharia Civil</td><td>1</td></tr><tr><td>Psicologia</td><td>1</td></tr><tr><td>Marketing</td><td>1</td></tr><tr><td>Medicina</td><td>1</td></tr><tr><td>Design Gráfico</td><td>1</td></tr></table>	Curso	NumStudents	Engenharia Informática	1	Gestão de Empresas	1	Arquitetura	1	Engenharia Civil	1	Psicologia	1	Marketing	1	Medicina	1	Design Gráfico	1
Curso	NumStudents																		
Engenharia Informática	1																		
Gestão de Empresas	1																		
Arquitetura	1																		
Engenharia Civil	1																		
Psicologia	1																		
Marketing	1																		
Medicina	1																		
Design Gráfico	1																		

6.3 - Apresentar informação acerca daquilo que foi vendido numa determinada venda.

Justificação	Código SQL																											
O número de UCs e ETS por Curso	<pre>SELECT c.DescricaoCurso AS Curso, SUM(uc.ETCS) AS TotalCredits, COUNT(uc.Id) AS NumUCs FROM Curso c JOIN Aluno a ON c.IdAluno = a.Id JOIN Inscricao_UC iu ON a.Id = iu.IdAluno JOIN UC uc ON iu.IdUC = uc.Id GROUP BY c.Id, c.DescricaoCurso ORDER BY TotalCredits DESC;</pre>																											
Output																												
<table><tr><th>Course</th><th>TotalCredits</th><th>NumUCs</th></tr><tr><td>Engenharia Informática</td><td>6</td><td>1</td></tr><tr><td>Gestão de Empresas</td><td>6</td><td>1</td></tr><tr><td>Engenharia Civil</td><td>6</td><td>1</td></tr><tr><td>Psicologia</td><td>6</td><td>1</td></tr><tr><td>Medicina</td><td>6</td><td>1</td></tr><tr><td>Arquitetura</td><td>4</td><td>1</td></tr><tr><td>Marketing</td><td>4</td><td>1</td></tr><tr><td>Design Gráfico</td><td>4</td><td>1</td></tr></table>		Course	TotalCredits	NumUCs	Engenharia Informática	6	1	Gestão de Empresas	6	1	Engenharia Civil	6	1	Psicologia	6	1	Medicina	6	1	Arquitetura	4	1	Marketing	4	1	Design Gráfico	4	1
Course	TotalCredits	NumUCs																										
Engenharia Informática	6	1																										
Gestão de Empresas	6	1																										
Engenharia Civil	6	1																										
Psicologia	6	1																										
Medicina	6	1																										
Arquitetura	4	1																										
Marketing	4	1																										
Design Gráfico	4	1																										


6.4 - Apresentar informação acerca daquilo que foi vendido numa determinada venda.

Justificação	Código SQL																		
Valor total por curso das inscrições	<pre> SELECT c.DescricaoCurso AS Curso, SUM(a.Valor_Inscricao) AS TotalValue FROM Curso c JOIN Aluno a ON c.IdAluno = a.Id GROUP BY c.Id, c.DescricaoCurso ORDER BY TotalValue ASC; </pre>																		
Output																			
 <table border="1"> <thead> <tr> <th>Curso</th><th>TotalValue</th></tr> </thead> <tbody> <tr> <td>Gestão de Empresas</td><td>150</td></tr> <tr> <td>Marketing</td><td>150</td></tr> <tr> <td>Engenharia Informática</td><td>200</td></tr> <tr> <td>Engenharia Civil</td><td>200</td></tr> <tr> <td>Medicina</td><td>200</td></tr> <tr> <td>Arquitetura</td><td>250</td></tr> <tr> <td>Design Gráfico</td><td>250</td></tr> <tr> <td>Psicologia</td><td>300</td></tr> </tbody> </table>		Curso	TotalValue	Gestão de Empresas	150	Marketing	150	Engenharia Informática	200	Engenharia Civil	200	Medicina	200	Arquitetura	250	Design Gráfico	250	Psicologia	300
Curso	TotalValue																		
Gestão de Empresas	150																		
Marketing	150																		
Engenharia Informática	200																		
Engenharia Civil	200																		
Medicina	200																		
Arquitetura	250																		
Design Gráfico	250																		
Psicologia	300																		

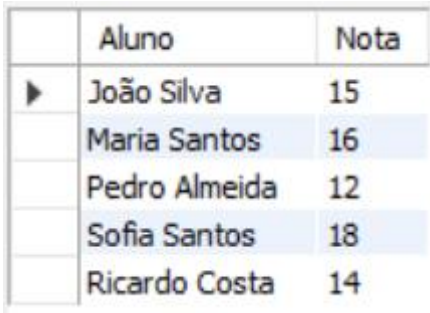
6.5 - Apresentar informação acerca daquilo que foi vendido numa determinada venda.

Justificação	Código SQL																																				
Mostra número de estudantes em cada UC e o somatório das respetivas ETs	<pre>SELECT uc.DescricaoUC AS UC, SUM(uc.ETCS) AS TotalCredits, COUNT(DISTINCT iu.IdAluno) AS NumStudents FROM UC uc JOIN Inscricao_UC iu ON uc.Id = iu.IdUC GROUP BY uc.Id, uc.DescricaoUC ORDER BY TotalCredits DESC;</pre>																																				
Output																																					
<table><tr><th></th><th>UC</th><th>TotalCredits</th><th>NumStudents</th></tr><tr><td>►</td><td>Programação</td><td>6</td><td>1</td></tr><tr><td></td><td>Contabilidade</td><td>6</td><td>1</td></tr><tr><td></td><td>Mecânica dos Materiais</td><td>6</td><td>1</td></tr><tr><td></td><td>Psicologia Experimental</td><td>6</td><td>1</td></tr><tr><td></td><td>Anatomia</td><td>6</td><td>1</td></tr><tr><td></td><td>Desenho Técnico</td><td>4</td><td>1</td></tr><tr><td></td><td>Marketing Digital</td><td>4</td><td>1</td></tr><tr><td></td><td>Design de Comunicação</td><td>4</td><td>1</td></tr></table>			UC	TotalCredits	NumStudents	►	Programação	6	1		Contabilidade	6	1		Mecânica dos Materiais	6	1		Psicologia Experimental	6	1		Anatomia	6	1		Desenho Técnico	4	1		Marketing Digital	4	1		Design de Comunicação	4	1
	UC	TotalCredits	NumStudents																																		
►	Programação	6	1																																		
	Contabilidade	6	1																																		
	Mecânica dos Materiais	6	1																																		
	Psicologia Experimental	6	1																																		
	Anatomia	6	1																																		
	Desenho Técnico	4	1																																		
	Marketing Digital	4	1																																		
	Design de Comunicação	4	1																																		

6.6 - Apresentar informação acerca daquilo que foi vendido numa determinada venda.

Justificação	Código SQL
Mostra nota mínima e máxima em cada UC	<pre>SELECT uc.DescricaoUC AS UC, MAX(ie.Nota) AS Max_Grade, MIN(ie.Nota) AS Min_Grade FROM UC uc JOIN Exame e ON uc.Id = e.IdUC JOIN Incricao_Exame ie ON e.Id = ie.IdExame GROUP BY uc.Id, uc.DescricaoUC;</pre>
Output	
	

6.7 - Apresentar informação acerca daquilo que foi vendido numa determinada venda.

Justificação	Código SQL												
Mostra alunos com notas acima de 10	<pre>SELECT p.Nome AS Aluno, ie.Nota AS Nota FROM Pessoa p JOIN Aluno a ON p.Id = a.Id JOIN Incricao_Exame ie ON a.Id = ie.IdAluno WHERE ie.Nota > 10;</pre>												
Output													
 <table border="1"> <thead> <tr> <th>Aluno</th><th>Nota</th></tr> </thead> <tbody> <tr><td>João Silva</td><td>15</td></tr> <tr><td>Maria Santos</td><td>16</td></tr> <tr><td>Pedro Almeida</td><td>12</td></tr> <tr><td>Sofia Santos</td><td>18</td></tr> <tr><td>Ricardo Costa</td><td>14</td></tr> </tbody> </table>		Aluno	Nota	João Silva	15	Maria Santos	16	Pedro Almeida	12	Sofia Santos	18	Ricardo Costa	14
Aluno	Nota												
João Silva	15												
Maria Santos	16												
Pedro Almeida	12												
Sofia Santos	18												
Ricardo Costa	14												

6.8 – Subconsultas Alunos aprovados nas UC_inscritas

Justificação	Código SQL																		
Mostra os alunos aprovados nas UC_inscritas	<pre>SELECT p.Nome, a.Estatuto FROM Pessoa p JOIN Aluno a ON p.Id = a.Id WHERE a.Id IN (SELECT IdAluno FROM Inscricao_UC WHERE Status = 'aprovado') ORDER BY p.Nome;</pre>																		
Output																			
<table><tr><th></th><th>Nome</th><th>Estatuto</th></tr><tr><td>►</td><td>João Silva</td><td>Estudate</td></tr><tr><td></td><td>Maria Santos</td><td>Estudante Trabalhador</td></tr><tr><td></td><td>Pedro Almeida</td><td>Estudate</td></tr><tr><td></td><td>Ricardo Costa</td><td>Estudante Trabalhador</td></tr><tr><td></td><td>Sofia Santos</td><td>Estudate</td></tr></table>			Nome	Estatuto	►	João Silva	Estudate		Maria Santos	Estudante Trabalhador		Pedro Almeida	Estudate		Ricardo Costa	Estudante Trabalhador		Sofia Santos	Estudate
	Nome	Estatuto																	
►	João Silva	Estudate																	
	Maria Santos	Estudante Trabalhador																	
	Pedro Almeida	Estudate																	
	Ricardo Costa	Estudante Trabalhador																	
	Sofia Santos	Estudate																	

6.9 – Subconsultas número de UCs que cada aluno está inscrito.

Justificação	Código SQL																											
Mostra o número de UCs que cada aluno está inscrito.	<pre>SELECT p.Nome, (SELECT COUNT(*) FROM Inscricao_UC WHERE IdAluno = a.Id) AS Num_UC_Inscritas FROM Pessoa p JOIN Aluno a ON p.Id = a.Id ORDER BY Num_UC_Inscritas DESC;</pre>																											
Output																												
<table><tr><th></th><th>Nome</th><th>Num_UC_Inscritas</th></tr><tr><td>▶</td><td>João Silva</td><td>1</td></tr><tr><td></td><td>Maria Santos</td><td>1</td></tr><tr><td></td><td>Pedro Almeida</td><td>1</td></tr><tr><td></td><td>Ana Pereira</td><td>1</td></tr><tr><td></td><td>Miguel Fernandes</td><td>1</td></tr><tr><td></td><td>Sofia Santos</td><td>1</td></tr><tr><td></td><td>Ricardo Costa</td><td>1</td></tr><tr><td></td><td>Carla Mendes</td><td>1</td></tr></table>			Nome	Num_UC_Inscritas	▶	João Silva	1		Maria Santos	1		Pedro Almeida	1		Ana Pereira	1		Miguel Fernandes	1		Sofia Santos	1		Ricardo Costa	1		Carla Mendes	1
	Nome	Num_UC_Inscritas																										
▶	João Silva	1																										
	Maria Santos	1																										
	Pedro Almeida	1																										
	Ana Pereira	1																										
	Miguel Fernandes	1																										
	Sofia Santos	1																										
	Ricardo Costa	1																										
	Carla Mendes	1																										

6.10 – Consultas usando grupos numero de alunos por Estatuto.

Justificação	Código SQL						
Mostra o número de alunos por Estatuto	<pre>SELECT a.Estatuto, COUNT(*) AS Num_Alunos FROM Aluno a GROUP BY a.Estatuto;</pre>						
Output							
<table><tr><td></td><td>Nome</td><td>Estatuto</td></tr><tr><td>▶</td><td>Miguel Fernandes</td><td>Estudante Trabalhador</td></tr></table>			Nome	Estatuto	▶	Miguel Fernandes	Estudante Trabalhador
	Nome	Estatuto					
▶	Miguel Fernandes	Estudante Trabalhador					

6.11 – Consultas usando grupos o valor de inscrição por curso.

Justificação	Código SQL									
Mostra o valor de inscrição por curso	<pre>SELECT c.DescricaoCurso, SUM(a.Valor_Inscricao) AS Total_Inscricao FROM Curso c JOIN Aluno a ON c.IdAluno = a.Id GROUP BY c.DescricaoCurso HAVING Total_Inscricao > 1;</pre>									
Output										
<table><tr><th></th><th>Estatuto</th><th>Num_Alunos</th></tr><tr><td>▶</td><td>Estudante</td><td>5</td></tr><tr><td></td><td>Estudante Trabalhador</td><td>3</td></tr></table>			Estatuto	Num_Alunos	▶	Estudante	5		Estudante Trabalhador	3
	Estatuto	Num_Alunos								
▶	Estudante	5								
	Estudante Trabalhador	3								

7. Triggers

Este capítulo apresenta os *triggers* criados.

Código SQL
<pre>DELIMITER \$\$ CREATE TRIGGER tri_bi_dtaAluno BEFORE INSERT ON Aluno FOR EACH ROW BEGIN SET NEW.Data_Inscricao = NOW(); END; END\$\$</pre>
Justificação
Triggers para update Data_Inscricao na tabela Aluno

Código SQL
<pre>DELIMITER \$\$ CREATE TRIGGER trg_bu_Data_inscricao_uc BEFORE UPDATE ON Inscricao_UC</pre>

```
FOR EACH ROW
BEGIN
  SET NEW.Data = NOW();
END$$
```

Justificação

Trigger para update Data na tabela Inscricao_UC

8. Referências

- Materiais da unidade curricular Conceção e Desenvolvimento de Bases de Dados (ano letivo 2019/2020, em ead.ipleiria.pt)
- *Fundamentals of Database Systems*, R. Elmasri & S. B. Navathe, Addison Wesley Pub. Co.
- *SQL - Structured Query Language* (6.^a edição), L. Damas, FCA, 2005
- ...

