

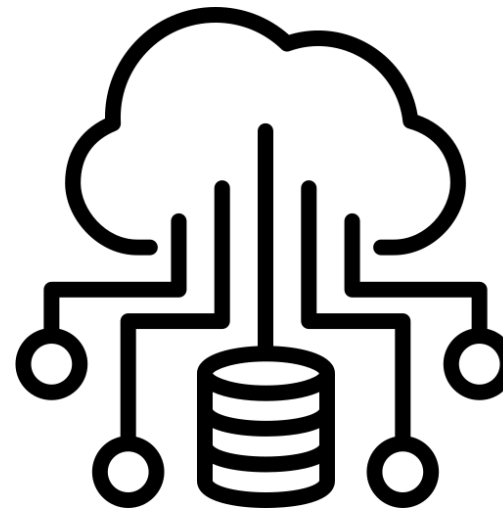


Bases de Dados

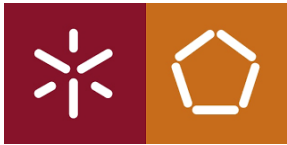
Aula Prática 5 – Modelação Lógica de Bases de Dados

Docente: Regina Sousa

Horário de Atendimento:
Quarta Feira 10h às 11h
Sala: 1.15



Versão 1.2



Aula Prática 5

- Revisão do esquema conceptual desenvolvido.
- Instalação do MySQL e MySQLWorkbench
- Selecionar um modelo concetual de uma vista de utilização e rever os diversos elementos
- Elemento a elemento, aplicar as regras e heurísticas de conversão do esquema concetual para o seu correspondente esquema lógico
- Utilizar o MySQL Workbench para criar o esquema lógico
- Validar o esquema lógico



Bibliografia

- **Capítulo 4 (Relational Model), 12 (ER Modeling), 16 (Conceptual Design) e 17 (Logical Design)**

Connolly, T., Begg, C., Database Systems, A Practical Approach to Design, Implementation, and Management , Addison-Wesley, 4a Edição, 2004.

- Teorey, T., Database Modeling and Design: The Fundamental Principles, II Edição, Morgan Kaufmann, 1994.

Webgrafia

Terra ER: <http://www.terraer.com.br>

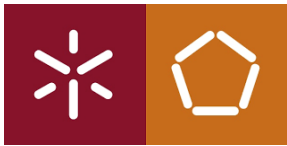
MySQL Workbench: <https://dev.mysql.com/downloads/workbench/>

DBSchema: https://dbschema.com/?AFFILIATE=96594&_c=1

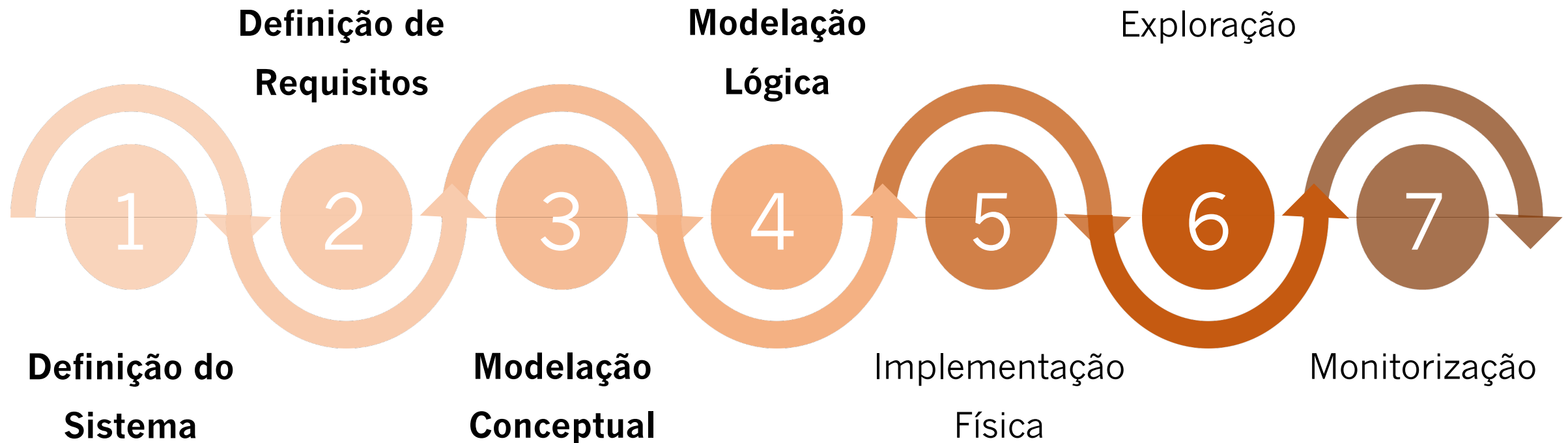
https://www.tutorialspoint.com/dbms/relational_data_model.htm

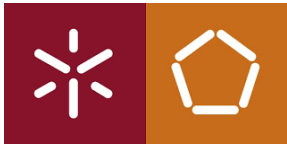
https://www.tutorialspoint.com/dbms/dbms_data_schemas.htm

<https://www.guru99.com/data-modelling-conceptual-logical.html>



Ciclo de Vida de Desenvolvimento





Instalação MySQL

Windows

- <https://dev.mysql.com/downloads/installer/>
- <https://dev.mysql.com/doc/mysql-installation-excerpt/5.7/en/windows-installation.html>

Linux

- <https://dev.mysql.com/doc/refman/8.0/en/linux-installation.html>

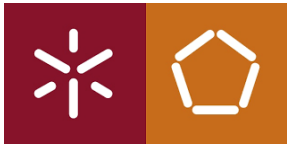
MacOS

- <https://dev.mysql.com/downloads/mysql/>
- <https://dev.mysql.com/doc/refman/8.0/en/macos-installation.html>

Instalação MySQL Workbench

<https://dev.mysql.com/downloads/workbench/>

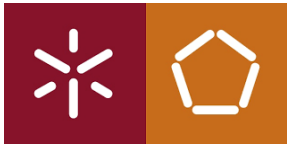
<https://dev.mysql.com/doc/mysql-installation-excerpt/5.7/en/>



Construção e Validação do Modelo de Dados Lógicos

Traduzir o modelo de dados concetual num modelo de dados lógico e, em seguida, validar esse modelo para verificar se ele é estruturalmente correto e capaz de suportar as transações necessárias.

1. Derivar relações para modelo de dados lógicos;
2. Validar as relações utilizando a normalização;
3. Validar as relações contra transações de utilizadores;
4. Verificar restrições de integridade;
5. Rever modelo de dados lógicos com o utilizador;
6. Fundir modelos de dados lógicos em modelos globais (**opcional**);
7. Verificar crescimento futuro.



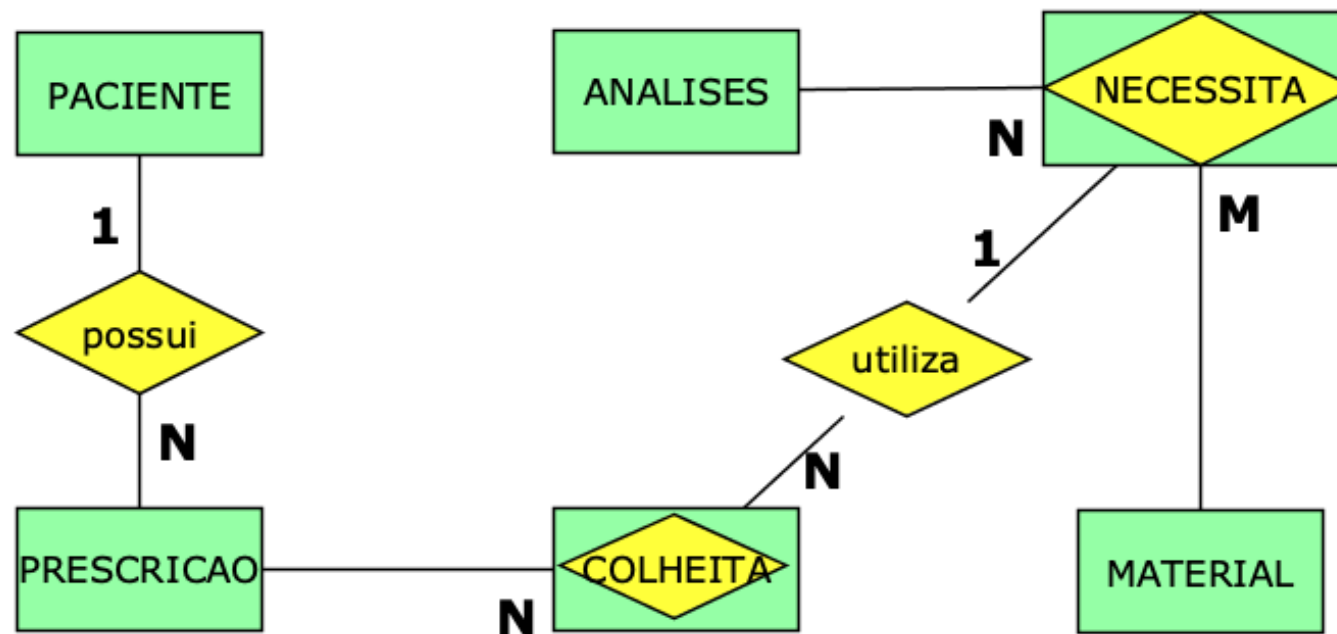
Vistas a Considerar

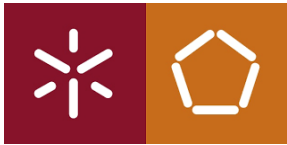
De acordo com o definido na ultima aula, durante o levantamento de requisitos, identificam-se 2 vistas de utilização.

- Pacientes e Colheitas;
- Gestão e Recursos;



Vista Paciente – Colheita





Modelação Lógica

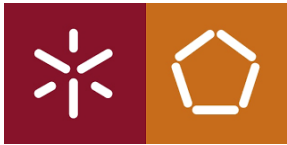
1. Derivar relações para modelo de dados lógicos

Questão 1: Crie relações para o modelo de dados lógico para representar as entidades, relacionamentos e atributos que foram identificados.

Nota: O relacionamento que uma entidade tem com outra entidade é representado pelo mecanismo de **chave primária/chave estrangeira**.

Ao decidir onde colocar o(s) atributo(s) de chave estrangeira, devemos primeiro identificar as entidades 'pai' e 'filho' envolvidas no relacionamento.

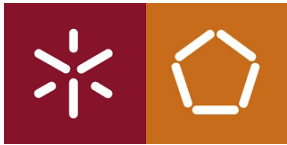
A entidade **pai** refere-se à entidade que **envia uma cópia da sua chave primária** na relação que representa a entidade **filho**, para atuar como a **chave estrangeira**.



Modelação Lógica

O processo de derivação passa por descrever como as relações são derivadas para as seguintes estruturas que podem ocorrer em um modelo de dados concetual:

1. Entidades Fortes
2. Entidades Fracas
3. Relacionamento Binário de um para muitos (1:N)
4. Relacionamento Binário de um para um (1:1)
5. Relacionamentos recursivos um-para-um (1:1)
6. Relacionamento superclasse/subclasse
7. Relacionamento binário muitos-para-muitos (N:M)
8. Relacionamentos complexos;
9. Atributos Multivalor



Modelação Lógica

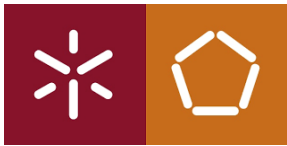
1.1. Entidades Fortes

Para cada entidade forte do modelo de dados, crie uma relação que inclua todos os atributos simples dessa entidade. Para atributos compostos, inclua apenas os atributos simples constituintes.

Paciente (id_pac, Nome, Data_Nascimento, Telemovel, Email, N_Contribuinte, Rua, Andar_Porta, Cod_Postal)

Chave primária id_pac

Chave candidata N_Contribuinte

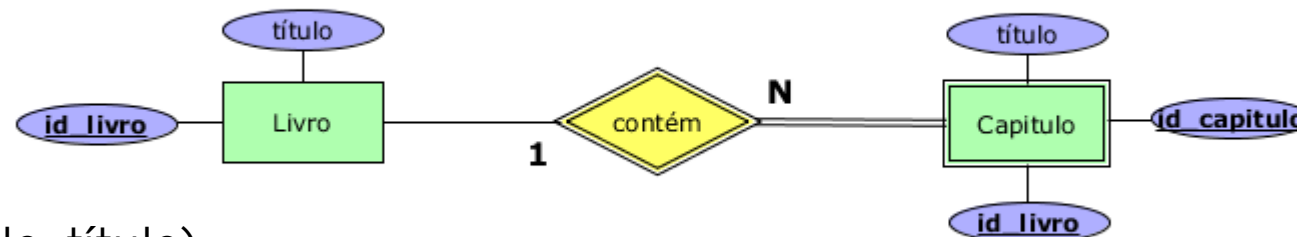


Modelação Lógica

1.2. Entidades Fracas

Para cada entidade fraca no modelo de dados, crie uma relação que inclua todos os atributos simples dessa entidade.

A chave primária de uma entidade fraca é parcial ou totalmente derivada de cada entidade proprietária e, portanto, a identificação da chave primária de uma entidade fraca não pode ser feita até que todos os relacionamentos com as entidades proprietárias tenham sido mapeados.

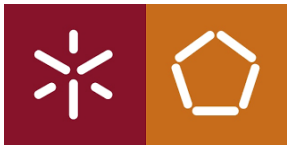


Chave Parcial

Capítulo (id_capítulo, título)

Chave primária (id_livro, id_capítulo)

Chave Estrangeira



Modelação Lógica

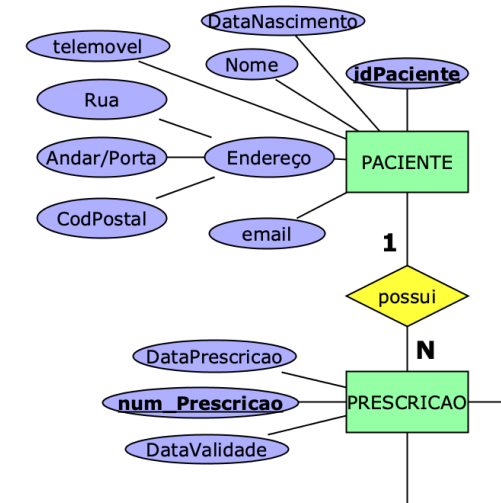
1.3. Relacionamento Binário de um para muitos (1:N)

Para cada relacionamento binário 1:*, a entidade no '**um lado**' do relacionamento é designada como a **entidade pai** e a entidade no '**muitos lado**' é designada como a **entidade filha**.

Para representar esse relacionamento, cria-se uma **cópia do(s) atributo(s) de chave primária** da entidade pai na relação que representa a entidade filho, para atuar como **chave estrangeira**.

Paciente (id_pac, Nome, Data_Nascimento, Telemovel, Email, N_Contribuinte, Rua, Andar_Porta, Cod_Postal)
Chave primária id_pac
Chave candidata N_Contribuinte

Prescrição (Num_Prescrição, Data_Prescrição, Data_validade, id_pac)
Chave primária Num_Prescrição
Chave estrangeira id_pac **referencia** Paciente(id_pac)





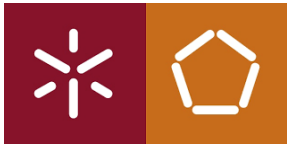
Modelação Lógica

1.4. Relacionamento Binário de um para um (1:1)

Nestes casos, a criação de relações é mais complexa, porque a cardinalidade não pode ser usada para identificar as entidades pai e filho num relacionamento.

Em vez disso, as restrições de participação são usadas para decidir se é preferível combinar as entidades numa relação ou se é mais adequado criar duas relações e colocar uma cópia da chave primária de uma relação para a outra:

- (a) participação obrigatória em ambos os lados do relacionamento 1:1;
- (b) participação obrigatória num lado do relacionamento 1:1;
- (c) participação opcional em ambos os lados do relacionamento 1:1.



Modelação Lógica

1.4. Relacionamento Binário de um para um (1:1)

a) Participação obrigatória em ambos os lados do relacionamento 1:1

Combinar as entidades envolvidas numa relação e escolher uma das chaves primárias das entidades originais para ser a chave primária da nova relação, enquanto outra (se existir) é usada como chave candidata.

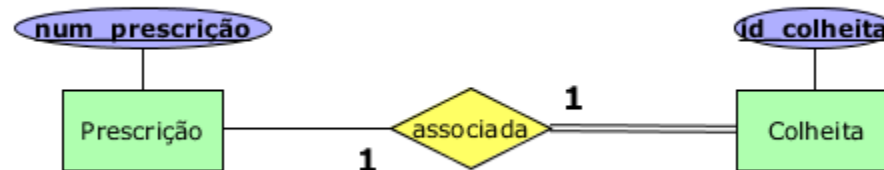


Modelação Lógica

1.4. Relacionamento Binário de um para um (1:1)

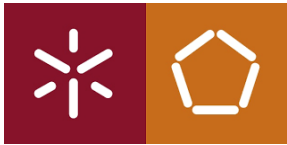
b) Participação obrigatória num lado do relacionamento 1:1

Entidade com participação opcional é designada como entidade-pai e a outra entidade como entidade-filho.
Cópia da chave primária do pai colocada na relação que representa a entidade filho.



Prescrição (Num_Prescrição, Data_Prescrição, Data_validade, id_pac)
Chave primária Num_Prescrição
Chave estrangeira id_pac **referencia** Paciente(id_pac)

Colheita (id_colheita, Resultado, Data_Colheita, num_Prescricao)
Chave primária id_colheita
Chave estrangeira num_prescrição **referencia** Colheita(num_prescricao)



Modelação Lógica

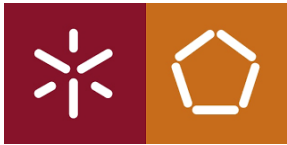
1.4. Relacionamento Binário de um para um (1:1)

c) Participação opcional em ambos os lados do relacionamento 1:1

A designação das entidades pai e filho é arbitrária, a menos que se possa descobrir mais sobre o relacionamento.

Considere a relação 1:1 Funcionário Usa Carro com participação opcional de ambos os lados. Suponha que a maioria dos carros, mas não todos, sejam usados pelos funcionários e que apenas uma minoria dos funcionários use carros.

A entidade Carro, embora opcional, está mais próxima de ser obrigatória do que a entidade Funcionário. Portanto, neste caso deveríamos designar o Funcionário como entidade-pai e o Carro como entidade-filho.

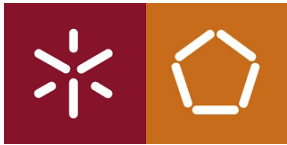


Modelação Lógica

1.5 Relacionamentos recursivos um-para-um (1:1)

Os relacionamentos recursivos de 1:1 seguem as regras de participação de um relacionamento binário de 1:1.

- **participação obrigatória de ambos os lados:** relação única com duas cópias da chave primária.
- **participação obrigatória em apenas um lado:** opção de criar uma relação única com duas cópias da chave primária, ou criar uma nova relação para representar o relacionamento. A nova relação teria apenas dois atributos, ambas cópias da chave primária.
- **participação opcional de ambos os lados:** novamente crie uma nova relação conforme descrito acima.



Modelação Lógica

1.6 Relacionamento superclasse/subclasse

- Identifique a superclasse como entidade pai e a entidade subclasse como entidade filha.
- Existem várias opções para representar um relacionamento deste tipo. A opção mais adequada depende do número de fatores, tais como:
 - restrições de disjunção e participação no relacionamento superclasse/subclasse;
 - se as subclasses estão envolvidas em relacionamentos distintos;
 - número de participantes no relacionamento superclasse/subclasse.



Modelação Lógica

1.6 Relacionamento superclasse/subclasse

Restrições de Participação	Restrições de Disjunção	Relações Requeridas
Obrigatória	Não disjunto {And}	Relação única
Opcional	Não disjunto {And}	Duas relações: uma relação para a superclasse e uma relação para todas as subclasses
Obrigatória	Disjunto {Or}	Muitas relações (uma relação para cada combinação superclasse/subclasse)
Opcional	Disjunto {Or}	Muitas relações (uma relação para a superclasse e uma para cada subclasse)



Modelação Lógica

1.6 Relacionamento superclasse/subclasse

Option 1 – Mandatory, nondisjoint

AllOwner (ownerNo, address, telNo, fName, lName, bName, bType, contactName, pOwnerFlag, bOwnerFlag)

Primary Key ownerNo

Option 2 – Optional, nondisjoint

Owner (ownerNo, address, telNo)

Primary Key ownerNo

OwnerDetails (ownerNo, fName, lName, bName, bType, contactName, pOwnerFlag, bOwnerFlag)

Primary Key ownerNo

Foreign Key ownerNo references Owner(ownerNo)

Option 3 – Mandatory, disjoint

PrivateOwner (ownerNo, fName, lName, address, telNo)

Primary Key ownerNo

BusinessOwner (ownerNo, bName, bType, contactName, address, telNo)

Primary Key ownerNo

Option 4 – Optional, disjoint

Owner (ownerNo, address, telNo)

Primary Key ownerNo

PrivateOwner (ownerNo, fName, lName)

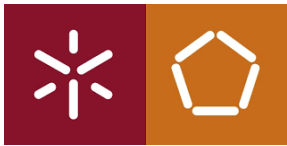
Primary Key ownerNo

Foreign Key ownerNo references Owner(ownerNo)

BusinessOwner (ownerNo, bName, bType, contactName)

Primary Key ownerNo

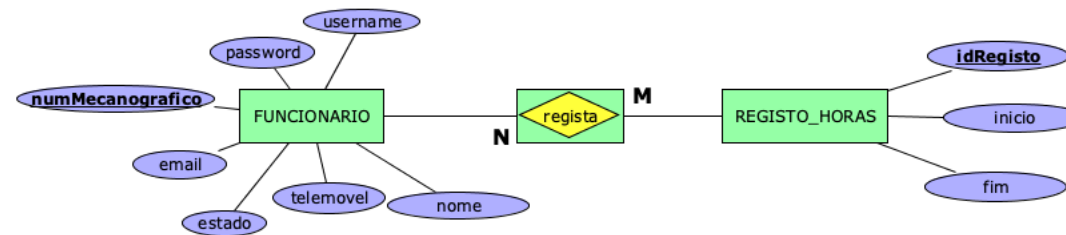
Foreign Key ownerNo references Owner(ownerNo)

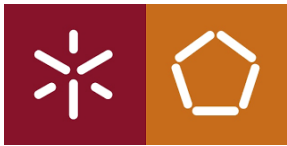


Modelação Lógica

1.7 Relacionamento binário muitos-para-muitos (N:M)

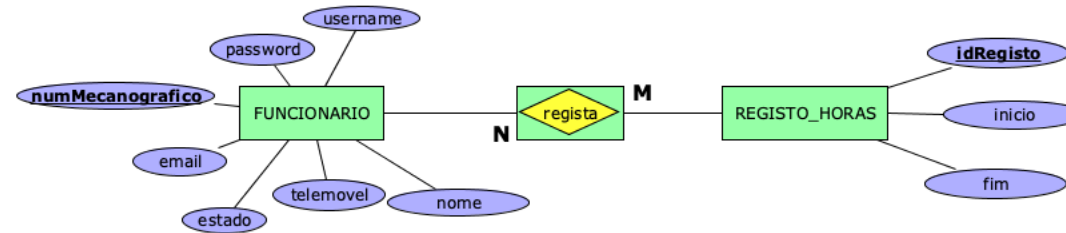
- Crie uma relação para representar o relacionamento e inclua quaisquer atributos que façam parte do relacionamento.
- Crie uma **cópia** do(s) atributo(s) de **chave primária** das entidades que participam no relacionamento numa nova relação, para atuar como **chaves estrangeiras**. Essas **chaves estrangeiras** também formarão a **chave primária** da nova relação, possivelmente em combinação com outros atributos do relacionamento.





Modelação Lógica

1.7 Relacionamento binário muitos-para-muitos (N:M)



Funcionario(Num_Mecanografico, Nome,
Email, Telemovel, Estado, Password, Username)
Chave primária Num_Mecanografico

Registo_Horas(id_registo, inicio, fim)
Chave primária id_registo

registra (Num_Mecanografico, id_registo)
Chave primária Num_Mecanografico , id_registo
Chave estrangeira Num_Mecanografico **referencia** Funcionario(Num_Mecanografico)
Chave estrangeira id_registo **referencia** Registo_Horas(id_registo)

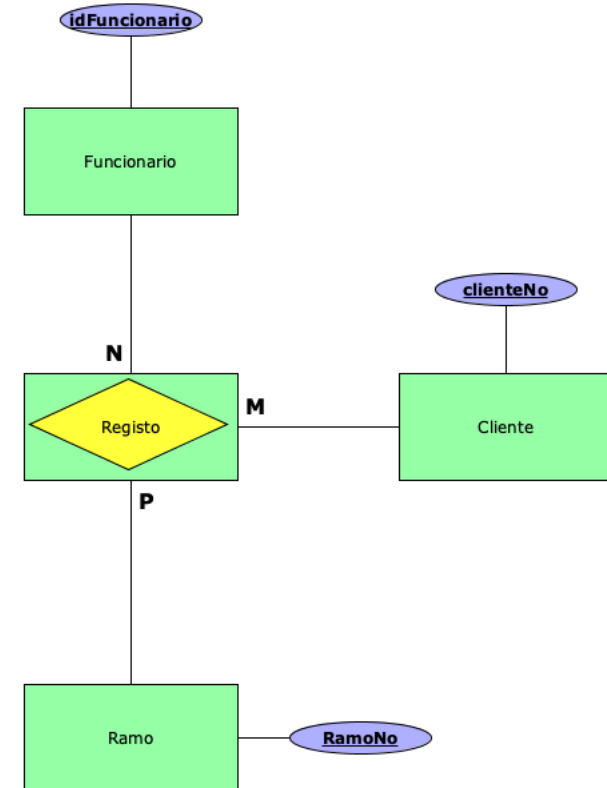


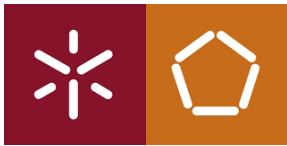
Modelação Lógica

1.8 Relacionamentos complexos

Para cada relacionamento complexo (triplo), criar uma relação para representar o relacionamento e incluir quaisquer atributos que façam parte do relacionamento.

Colocamos uma cópia do(s) atributo(s) principal(is) das entidades que participam no relacionamento complexo na nova relação, para atuar como chaves estrangeiras. Quaisquer chaves estrangeiras que representem um relacionamento "muitos" (1:N) geralmente também formarão a chave primária desta nova relação, possivelmente em combinação com alguns dos atributos do relacionamento.





Modelação Lógica

1.8 Relacionamentos complexos

Funcionario(id_Funcionario, Nome, ...)

Chave primária id_Funcionario

Cliente(clienteNo, Nome, ...)

Chave primária clienteNo

Ramo(ramoNo, Descricao, ...)

Chave primária ramoNo

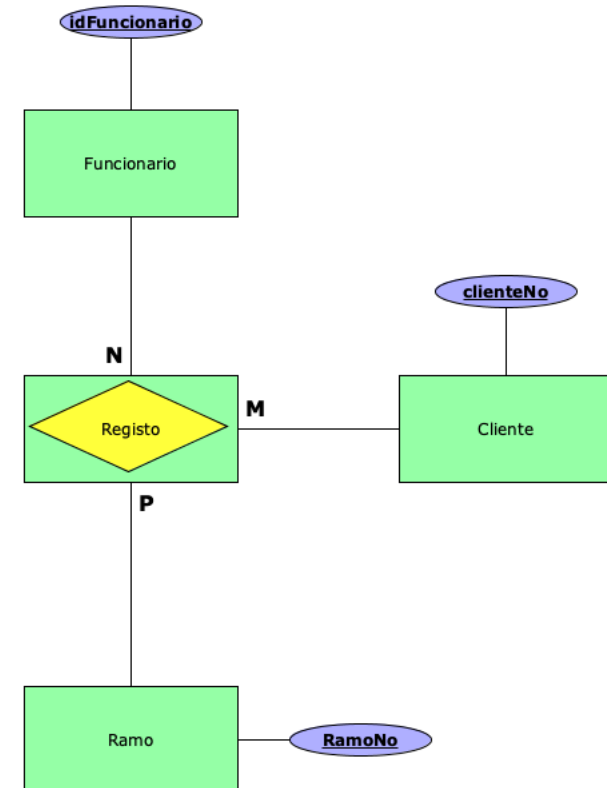
Registo(, clienteNo, ramoNo, ...)

Chave primária id_Funcionario, clienteNo, ramoNo

Chave Estrangeira id_Funcionario **referencia** Funcionario(id_Funcionario)

Chave Estrangeira clienteNo **referencia** Cliente(clienteNo)

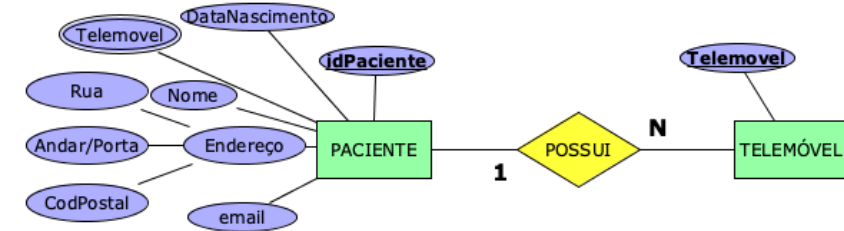
Chave Estrangeira ramoNo **referencia** Ramo(ramoNo)





Modelação Lógica

1.9 Atributos Multivalor



Para cada atributo multivalor numa entidade, criar uma nova relação para representar o atributo multivalor e incluir a chave primária da entidade na nova relação, para atuar como uma chave estrangeira.

Exemplo: Na vista do Paciente para representar a situação em que um único paciente tem até três números de telemóvel, o atributo telemóvel da entidade Paciente foi definido como sendo um atributo multi-valor. Para representar isto, criamos uma relação para a entidade Paciente e criamos uma nova relação chamada Telemóvel para representar o atributo multi-valor Telemóvel

Paciente (id_pac, Nome, Data_Nascimento, Telemovel, Email, N_Contribuinte, Rua, Andar_Porta, Cod_Postal)
Chave primária id_pac

Telemovel (telemóvel, id_pac)
Chave primária telemóvel
Chave Estrangeira id_pac referencia Paciente(id_pac)

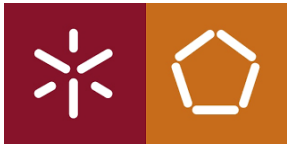


Modelação Lógica

2. Validar as relações utilizando a normalização

Questão 2: Valide as relações no modelo de dados lógicos utilizando a normalização.

O objetivo da normalização é assegurar que o conjunto de relações tenha um número mínimo e ainda assim suficiente de atributos necessários para suportar os requisitos de dados da empresa. Alguma redundância é essencial para permitir a junção de relações relacionadas.



Normalização

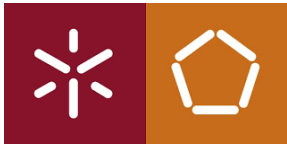
A normalização é uma técnica para produzir um conjunto de relações com propriedades desejáveis, dados requisitos de uma empresa. A normalização é um método formal que pode ser utilizado para identificar relações baseadas nas suas chaves e nas dependências funcionais entre os seus atributos.

Forma não normal (UNF): Uma tabela que contém um ou mais grupos de repetição.

Primeira forma normal (1NF): Uma relação na qual a intersecção de cada linha e coluna contém um e apenas um valor.

Segunda Forma Normal (2NF) : Uma relação que está na Primeira Forma Normal e cada atributo-chave não-primária depende inteiramente funcionalmente da chave primária. A dependência funcional total indica que se A e B são atributos de uma relação, B é totalmente dependente funcionalmente de A se B é funcionalmente dependente de A mas não de qualquer subconjunto adequado de A.

Terceira Forma Normal (3NF): Uma relação que está na Primeira e Segunda Forma Normal e na qual nenhum atributo chave não primário depende transitoriamente da forma primária chave. A dependência transitória é uma condição em que A, B, e C são atributos de uma relação tal que se $A \rightarrow B$ e $B \rightarrow C$, então C é transitoriamente dependente de A via B (desde que A não seja funcionalmente dependente de B ou C).



Modelação Lógica

3. Validar as relações contra transações de utilizadores

Questão 3: Assegure que as relações no modelo lógico de dados suportam todas as transações necessárias.

Nota:

Se conseguirmos resolver todas as transações desta forma, validámos o modelo de dados lógicos contra as transações.

No entanto, se não formos capazes de realizar uma transação manualmente, deve haver um problema com o modelo de dados, que tem de ser resolvido.

Neste caso, é provável que tenha sido introduzido um erro durante a criação das relações, e devemos voltar atrás e verificar as áreas do modelo de dados a que a transação está a aceder para identificar e resolver o problema



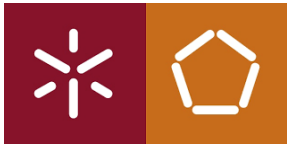
Modelação Lógica

4. Verificar restrições de integridade

Questão 4: Especifique quais as restrições de integridade necessárias, independentemente da forma como isso possa ser conseguido.

Nota: Um modelo lógico de dados que inclua todas as restrições de integridade importantes é uma representação "verdadeira" dos requisitos de dados para a empresa. Consideramos os seguintes tipos de restrições de integridade:

- **Dados necessários/obrigatórios:** Alguns atributos devem conter sempre um valor válido;
- **Restrições de domínio de atributos:** Cada atributo tem um domínio, ou seja, um conjunto de valores que são legais. Por exemplo, o sexo de uma pessoa ou é 'M' ou 'F' ou 'O';
- **Multiplicidade:** A multiplicidade representa os constrangimentos que são colocados nas relações entre os dados da base de dados (Análises - Colheita);
- **Integridade da entidade:** A chave primária de uma entidade não pode conter nulos;
- **Integridade referencial:** Uma chave estrangeira liga cada tuplo na relação Clinica ao tuplo na relação Funcionário contendo o valor da chave do candidato correspondente.;
- **Restrições gerais:** Atualizações às entidades podem ser controladas por restrições que regem as transações "do mundo real" que são representadas pelas atualizações. Por exemplo: Um funcionário não pode trabalhar em mais do que 2 clinicas



Modelação Lógica

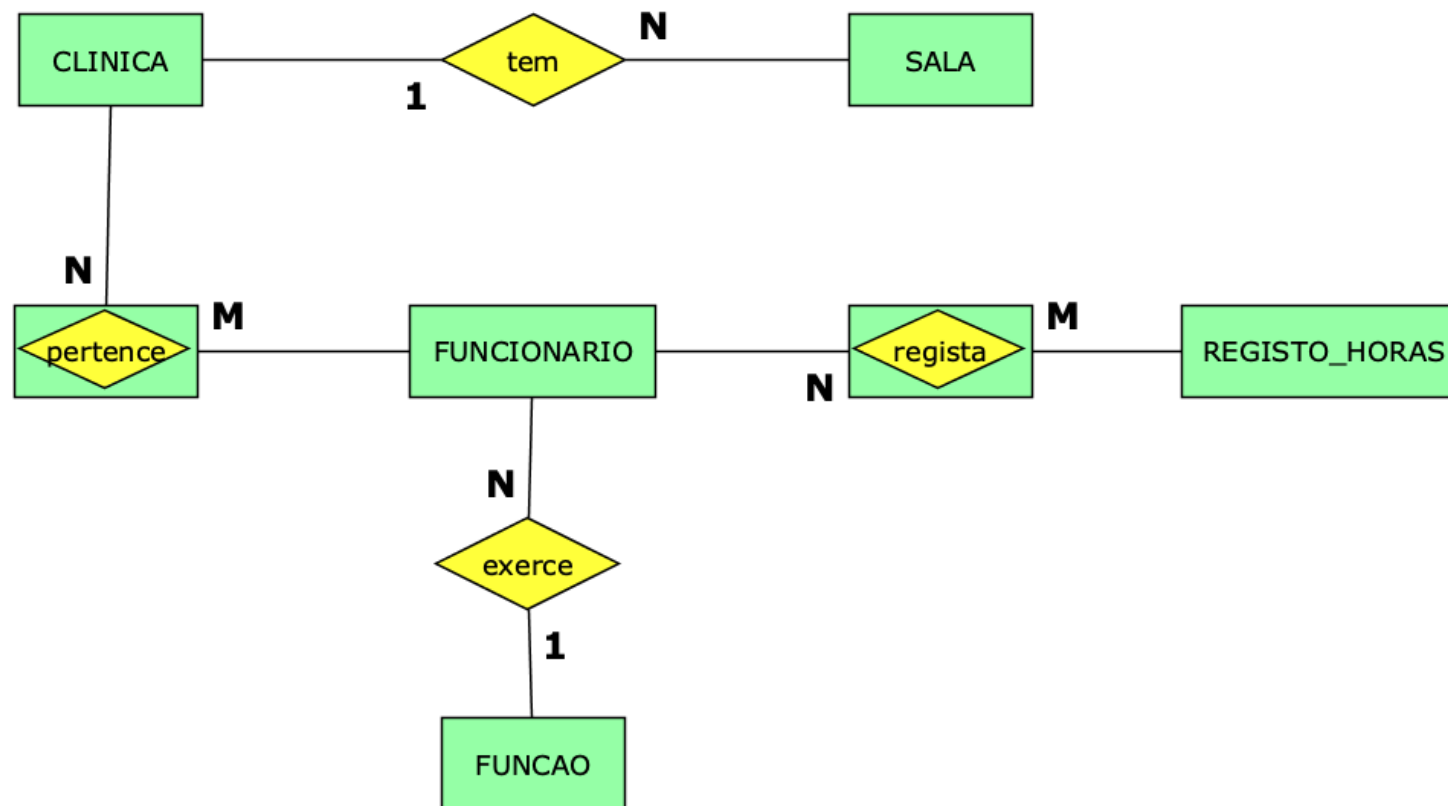
5. Rever modelo de dados lógicos com o utilizador

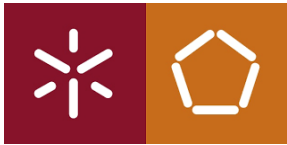
Questão 5: Para confirmar a representatividade do seu modelo, reúna com a empresa de forma a assegurar que consideram o modelo como uma verdadeira representação dos requisitos dos dados.

Se os utilizadores não estiverem satisfeitos com o modelo, poderá ser necessária alguma repetição de passos anteriores na metodologia.



Vista Gestão e Recursos





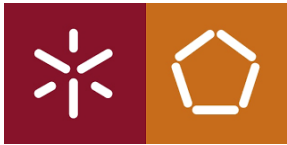
Modelação Lógica

6. Fundir modelos de dados lógicos em modelos globais (opcional)

Questão 6: Se considerou a modelação por vistas, considere fundir os diversos modelos lógicos. Considere as seguintes etapas:

- 6.1. Fundir modelos de dados lógicos locais em modelo global
- 6.2. Validar modelo global de dados lógicos
- 6.3. Rever modelo de dados lógicos globais com utilizadores

Como resultado deve ter o modelo global de dados lógicos que represente os requisitos de uma dada empresa.

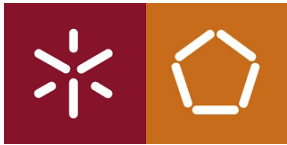


Modelação Lógica

6. Fundir modelos de dados lógicos em modelos globais (opcional)

Tarefas típicas do ponto 6.1:

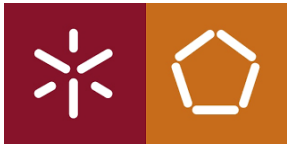
1. Rever os nomes e conteúdos das entidades/relações e as suas chaves candidatas.
2. Rever os nomes e conteúdos das relações/ chaves estrangeiras
3. Fundir entidades/relações a partir dos modelos de dados locais
4. Incluir (sem fundir) entidades/relações únicas para cada modelo de dados local.
5. Fundir relações/chaves estrangeiras a partir dos modelos de dados locais.
6. Incluir (sem fusão) relações/chaves estrangeiras exclusivas de cada modelo de dados local.
7. Verificar se faltam entidades/relações e relações/chaves estrangeiras.
8. Verificar chaves estrangeiras.
9. Verificar restrições de integridade.
10. Desenhar o diagrama global ER/relações.
11. Atualizar a documentação.



Modelação Lógica

7. Verificar crescimento futuro

Questão 7: Determine se existem quaisquer mudanças significativas prováveis num futuro previsível e avaliar se o modelo de dados lógico pode acomodar essas mudanças.



Modelação Lógica

NOTA: Este modelo é a fonte de informação para a conceção física da base de dados descrita como etapas 3 a 8 nos Capítulos 17 e 18.



Ficha de Trabalho 5: Sumário

Questão 1: Crie relações para o modelo de dados lógico para representar as entidades, relacionamentos e atributos que foram identificados.

Questão 2: Valide as relações no modelo de dados lógicos utilizando a normalização.

Questão 3: Assegure que as relações no modelo lógico de dados suportam todas as transações necessárias.

Questão 4: Especifique quais as restrições de integridade necessárias, independentemente da forma como isso possa ser conseguido.

Questão 5: Para confirmar a representatividade do seu modelo, reúna com a empresa de forma a assegurar que consideram o modelo como uma verdadeira representação dos requisitos dos dados.

Questão 6: Se considerou a modelação por vistas, considere fundir os diversos modelos lógicos.

Questão 7: Determine se existem quaisquer mudanças significativas prováveis num futuro previsível e avaliar se o modelo de dados lógico pode acomodar essas mudanças.