

Sistemas de Informação e Bases de Dados
Resolução dos Exercícios

Sandra Cristina Lopes Pedro Rangel Henriques

Novembro, 2001

Conteúdo

1	Método de Análise e Desenvolvimento do SI	2
1.1	Análise e Modelo ER	2
1.2	Implementação num SGBD (Access)	2
2	Resolução dos Problemas	4
2.1	Empresa Fabril	6
2.2	Direcção de Curso	7
2.3	Governo Civil	8
2.4	Clube de Vídeo	9
2.5	Comércio	10
2.6	Pastelaria	12
2.7	Centro de Saúde	13
2.8	Projectos de Investigação	16
2.9	Colóquio	18
2.10	Viagem	19

Capítulo 1

Método de Análise e Desenvolvimento do SI

Este caderno tem por objectivo apresentar a resolução de problemas, propostos em separado, relativos à criação em computador de sistemas de informação (SI) suportados por uma base de dados relacional (BDR).

Embora não se discuta aqui cada passo do processo de desenvolvimento dos SI pedidos, sugere-se sucintamente neste primeiro capítulo as fases do método que se propõe para passar do enunciado dum problema à respectiva BDR.

1.1 Análise e Modelo ER

A primeira fase consiste na análise do problema de modo a identificar detalhadamente tudo quanto é conhecido à partida (dados) e tudo quanto se pretende saber (resultados). Esta actividade é realizada através da leitura do enunciado, de conversas várias com os responsáveis por esse enunciado e, se necessário, da consulta a bibliografia relacionada com o universo do problema.

Desta etape de análise deve resultar um modelo (matemático) preciso e não ambíguo para descrever o SI em causa. Esse modelo é formado por Entidades e por Relações binárias entre essas Entidades. Cada Entidade—correspondente a uma unidade de informação com autonomia, isto é, com existência própria, independente das outras unidades—é identificada por um nome próprio e é caracterizada por um conjunto de atributos (simples, compostos, ou listas).

Cada Relação—ligação entre 2 Entidades—é, também, identificada por um nome, uma cardinalidade e, se necessário, um conjunto de atributos. A cardinalidade indica se a um elemento de uma das Entidades corresponde um e um só elemento da outra, ou se podem corresponder mais do que um elementos.

1.2 Implementação num SGBD (Access)

Uma vez analisado o problema e construído o DER que descreve com rigor o sistema de informação a criar deve proceder-se à sua implementação num Sistema de Gestão de Bases de Dados (SGBD).

Os passos a seguir para sistematizar esse processo são:

1. Criação das Tabelas

- Para implementar cada Entidade deve ser criada 1 Tabela, de acordo com as seguintes recomendações:
 - Dar à Tabela o mesmo nome da entidade
 - Incluir na Tabela um Campo por cada atributo simples da entidade
 - Incluir na Tabela um Campo por cada componente de um atributo composto da entidade
 - Não Incluir na Tabela nenhum Campo correspondente a atributos do tipo lista da entidade; neste caso deve ser criada uma outra Tabela, com o nome desse atributo e com 2 Campos: a chave desta e cada valor da lista
 - Identificar (marcar) o Campo (ou Campos) que serão a Chave
- Para implementar cada Relação $(1) \rightarrow (1)$ deve:
 - Incluir numa qualquer das Tabelas envolvidas na Relação um Campo que contenha como valor a Chave da outra Tabela
- Para implementar cada Relação $(1) \rightarrow (N)$ deve:
 - Incluir na Tabela que tem a cardinalidade N um Campo que contenha como valor a Chave da outra Tabela
- Para implementar cada Relação $(N) \rightarrow (N)$ deve:
 - Criar uma outra Tabela, com o nome dessa Relação e com, no mínimo, 2 Campos um para cada chave das 2 Tabelas envolvidas, juntando ainda os campos adicionais necessários para os atributos dessa Relação

2. Criação das Ligações

Uma vez criadas todas as Tabelas segundo os princípios enunciados, deve proceder-se à sua ligação, usando os Campos criados para o efeito.

3. Criação dos Formulários

Posteriormente, serão gerados automaticamente os formulários para inserção e consulta de dados em cada uma das Tabelas da BD.

4. Criação das Consultas

usando as facilidades de Projeção, Selecção e Junção, serão criadas novas Tabelas com os resultados da filtragem da BD

5. Criação dos Relatórios

Nesta fase e para obter documentos impressos com os resultados das Consultas (directas, ou filtradas) criam-se os Relatórios

Capítulo 2

Resolução dos Problemas

Neste capítulo, que é o fulcro do presente caderno, apresenta-se, em diferentes secções, uma possível solução para cada um dos problemas enunciados.

Porém, antes disso e na próxima secção, explica-se a estratégia adoptada para a organização do caderno.

Explicação da abordagem seguida e da notação usada

Em relação a cada um dos problemas propostos, apresenta-se apenas a solução final sugerida sem se discutir o caminho até lá se chegar. Exceptua-se o caso do problema 2.7 para o qual se inclui todo o raciocínio que levou do enunciado à construção do Modelo Entidades-Relações (DER), ao desenho da BDR e à formulação das questões.

Essa solução será sempre formada por 3 partes:

- O Modelo Entidades-Relações, ou Modelo Entidades-Associações, que especifica o Sistema de Informação (SI) que se quer construir
- O Modelo Físico, ou seja, a estrutura da Base de Dados, que estará conforme o modelo e implementará o dito SI
- As *Expressões necessárias para interrogar* a BD e obter a resposta a cada uma das questões que se quer colocar ao SI

A **notação** que será seguida para apresentar cada uma destas 3 componentes de cada solução é a indicada abaixo:

Modelo E-R

O Modelo E-R é apresentado identificando as Entidades presentes, caracterizando-as à custa de um conjunto de atributos, e também as Relações que ligam essas Entidades entre si (duas a duas).

Entidades:

nomeEntidade = { *AtribChave*: tipo; *AtribNormal*: tipo; ... }

Obs: o **nomeEntidade** é um substantivo no plural

Relações:

nomeRelação: Entidade₁ (*n*) → (*n*) Entidade₂ ; atribs = { *Atrib*: tipo; ... }

Obs: o **nomeRelação** é um verbo que relaciona (liga) a **Entidade₁** à **Entidade₂**, indicando-se a cardinalidade da relação (1:1; 1:N; N:1; N:N) à esquerda e à direita da seta

Base de Dados

A estrutura da Base de dados é apresentada através da identificação (indicação do *nome*) e caracterização (indicação das colunas, ou campos) de todas as Tabelas que a vão integrar.

Essas Tabelas obtém-se directamente das Entidades do DER e, eventualmente, das Relações; o Tipo das colunas (campos) é omitido pois foi definido para os correspondentes atributos das respectivas Entidades;

Para implementar as Relações, as Tabelas serão ligadas entre si através de atributos que se comportam como Referências Cruzadas, isto é, atributos de uma Tabela cujo valor é a Chave de uma outra Tabelas (Chave Estrangeira). Nas Relações 1:1 e N:1 não se recorre a nenhuma tabela extra, acrescentando-se apenas a Chave Estrangeira à Tabela da esquerda (para a qual cada registo só corresponde a 1 registo da outra), resultando então uma Tabela da forma geral:

nomeTabela = { *AtribChave*; *AtribNormal*; ... ; ChaveEstrang }

quanto à Tabela da direita (onde cada registo pode estar ligado a N registos da outra tabela), será em tudo semelhante a esta, só que não terá nenhum atributo correspondente à Chave Estrangeira.

Note-se que o **nome** destas Tabelas será exactamente o nome das Entidades correspondentes.

Nas Relações 1:N e N:N usa-se uma Tabela extra que conterà as Chaves Estrangeiras de ambas as Tabelas ligadas:

nomeTabela = { ChaveEstrang₁; ChaveEStrang₂; ... }

Neste caso e sempre que possível, o **nome** da Tabela será formado pelos nomes das 2 tabelas ligadas separados pelo caracter '/

Questões

Para cada questão é apresentada uma frase que descreve a **interrogação** (cujo termo técnico inglês é *query*) a enviar à Base de Dados para extrair dela os registos (total, ou parcialmente) que verificam

a Condição explicitada.

Cada frase é composta por três cláusulas, ou partes:

1. na primeira, JUNTANDO, indicam-se as Tabelas intervenientes na interrogação (cujos campos se querem ver ou comparar);
2. na segunda indicam-se os campos das tabelas a visualizar (VER), ou os campos de que se quer calcular o somatório (SOMAR), ou simplesmente a indicação que se pretende contar os registos que verificam a condição (CONTAR);
3. na terceira, ONDE, explicita-se a condição (ou condições) que os valores dos atributos de cada registo devem verificar (i.é, para os quais a condição tem de ser verdadeira) para serem seleccionados (extraídos da BD), de modo a serem mostrados, adicionados ou contados de acordo com o pedido expresso na segunda parte.

a) JUNTANDO Tabela₁, Tabela₂, ...

VER Tabela.Atrib ou VER TUDO ou CONTAR ou SOMAR Tabela.Atrib

ONDE (Condição)

sendo:

Condição = Tabela₁.Atrib *OpRel* Tabela₂.Atrib

Condição = Tabela.Atrib *OpRel* "valor"

Condição = (Condição) *OpBool* (Condição)

As Tabelas que se descrevem no início (cláusula JUNTANDO) terão de estar ligadas aos pares de modo a que as respectivas linhas se possam unir sempre que a Chave Estrangeira de uma tenha o valor da Chave de outra.

2.1 Empresa Fabril

Modelo E-R

Entidades:

Fornecedores = { *CodF*: str; Nome: str; Morada: str; Telef: str }

Produtos = { *CodP*: str; Nome: str }

Relações:

fornecem: Fornecedores (*N*) → (*N*) Produtos ; atribs = {Preço: moeda}

O Diagrama da figura 2.1 esquematiza o Modelo E-R acabado de descrever.

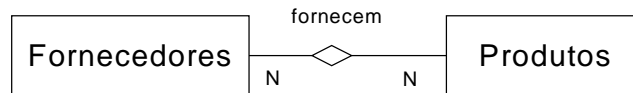


Figura 2.1: DER que modela o SI para gestão dos Produtos de uma Empresa Fabril

Base de Dados

Fornecedores = { *CodF*; Nome; Morada; Telef }

Produtos = { *CodP*; Nome }

Fornecedores/Produtos = { *CodF*; *CodP*; Preço }

2.2 Direcção de Curso

Modelo E-R

Entidades:

Disciplinas = { *CodD*: str; Nome: str; Objectivos: texto; HorasT: int; HorasP: int }

Cursos = { *CodC*: str; Nome: str; Duração: int }

Alunos = { *Número*: int; Nome: str; Morada: str; Telefone: str }

Docentes = { *CodDoc*: str; Nome: str; Depart: str; ExtTelef: str }

Relações:

são-leccionadas: Disciplinas (*N*) → (1) Cursos ; atribs = { }

leccionam: Docentes (*N*) → (*N*) Disciplinas ; atribs = { }

frequentam: Alunos (*N*) → (*N*) Disciplinas ; atribs = { }

O Diagrama da figura 2.2 esquematiza o Modelo E-R acabado de descrever.

Base de Dados

Disciplinas = { *CodD*; Nome; Objectivos; HorasT; HorasP; *CodC* }

Cursos = { *CodC*; Nome; Duração }

Alunos = { *Número*; Nome; Morada; Telefone }

Docentes = { *CodDoc*; Nome; Depart; Morada; Telefone }

Docentes/Disciplinas = { *CodDoc*; *CodD* }

Alunos/Disciplinas = { *Número*; *CodD* }

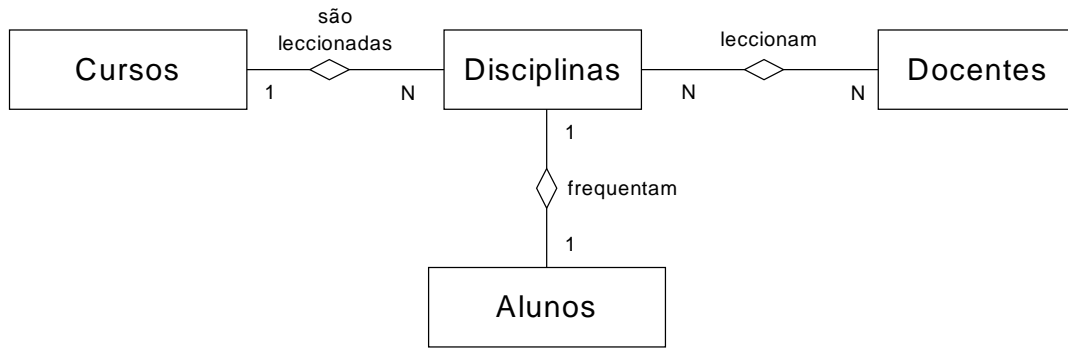


Figura 2.2: DER que modela o SI para apoio à Direcção de Curso

2.3 Governo Civil

Modelo E-R

Entidades:

Freguesias = { *CodF*: str; Nome: str; PresJ: str; Concelho: str; CoorG: str; Área: str; Orçamento: moeda }

Bens = { *CodB*: str; Nome: str; Tipo: { pedreira, parque, petróleo, mina } }

Indivíduos = { *BI*: int; Nome: str; Sexo: { F, M }; DataNasc: data; NEleitor: str }

Agregados = { *CodA*: str; Rua: str; NPrédio: str }

Relações:

têm: Freguesias (1) → (N) Bens ; atribs = { }

nasceram: Indivíduos (N) → (1) Freguesias ; atribs = { }

pertencem: Indivíduos (N) → (1) Agregados ; atribs = { }

vivem: Agregados (N) → (1) Freguesias ; atribs = { }

são-encabeçados: Agregados (N) → (1) Indivíduos ; atribs = { }

O Diagrama da figura 2.3 esquematiza o Modelo E-R acabado de descrever.

Base de Dados

Freguesias = { *CodF*; Nome; PresJ; Concelho; CoorG; Área; Orçamento }

Bens = { *CodB*; Nome; Tipo }

Freguesias/Bens = { *CodF*; *CodB* }

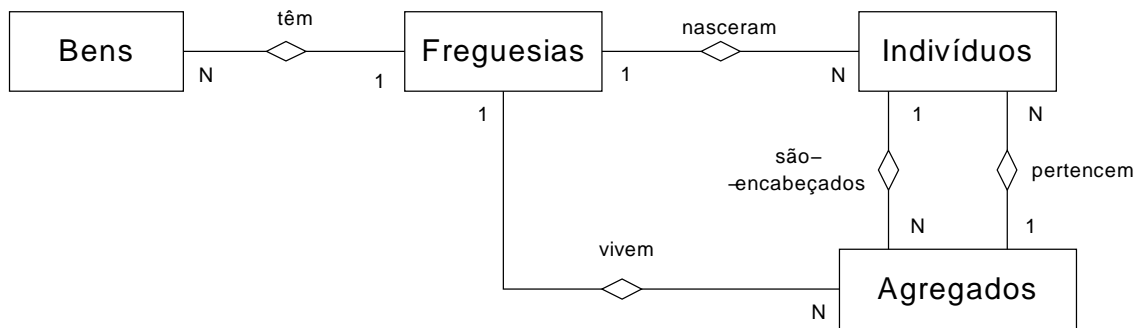


Figura 2.3: DER que modela o SI para apoio ao Governador Civil

Indivíduos = { *BI*; Nome; Sexo; DataNasc; NEleitor; CodF(FregNasc) }

Agregados = { *CodA*; Rua; NPrédio; BI(CabeçaCasal); CodF }

Indivíduos/Agregados = { BI; CodA }

2.4 Clube de Vídeo

Modelo E-R

Entidades:

Vídeos = { *CodV*: str; Título: str; Género: { drama, comedia, romance, aventura, policial, suspense }; EmpDistrib: str; Duração: int }

Clientes = { *NCli*: int; Nome: str; Morada: str; Telefone: str }

Alugueres = { *NumAluguer*: int; *CodV*: str; *NCli*: str; *DataS*: data; *DataE*: data }

Relações:

dizem-respeito: Alugueres (*N*) → (1) Vídeos ; atribs = { }

são-feitos: Alugueres (*N*) → (1) Clientes ; atribs = { }

O Diagrama da figura 2.4 esquematiza o Modelo E-R acabado de descrever.

Base de Dados

Vídeos = { *CodV*; Título; Género; EmpDistrib; Duração }

Clientes = { *NCli*; Nome; Morada; Telefone }

Alugueres = { *NumAluguer*; *DataS*; *DataE*; *CodV*; *NCli* }

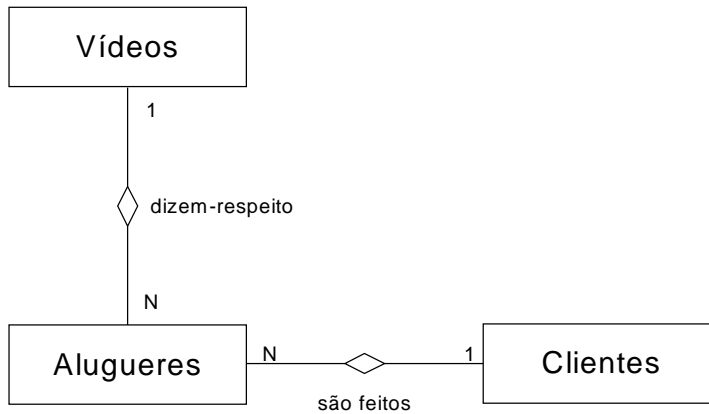


Figura 2.4: DER que modela o SI para gestão dum Clube de Vídeo

Questões

- JUNTANDO Videos
 VER TUDO
 ONDE (Videos.Género = "X")
- JUNTANDO Videos
 VER Videos.CodV, Videos.Titulo, Videos.Género
 ONDE (Videos.EmpDistrib = "Y")
- JUNTANDO Videos, Clientes, Alugueres
 VER Clientes.NCli, Clientes.Nome, Clientes.Morada
 ONDE (Videos.Titulo = "Z")
- JUNTANDO Videos, Alugueres
 VER Videos.CodV, Videos.Título
 ONDE ((Alugures.DataS <= "W") E (Alugures.DataE >= "W"))

2.5 Comércio

Modelo E-R

Entidades:

Encomendas = { *NumEnc*: int; *DataEnc*: data }

Clientes = { *CodC*: str; *Nome*: str; *Telefone*: str; *Endereço*: str }

Artigos = { *CodA*: str; *Nome*: str; *PreçoC*: moeda; *PreçoV*: moeda }

MatériasPrimas = { *CodMP*: str; *Nome*: str; *Tipo*: str }

Fornecedores = { *CodF*: str; *Nome*: str; *Telefone*: str; *Endereço*: str }

Relações:

são-feitas: Encomendas (N) \rightarrow (1) Clientes ; atribs = { }

pedem: Encomendas (N) \rightarrow (N) Artigos ; atribs = { QuantidEnc }

são-formados: Artigos (N) \rightarrow (N) MatériasPrimas ; atribs = { QuantidUsada: real }

são-fornecidas: MatériasPrimas (N) \rightarrow (N) Fornecedores ; atribs = { Preço: real }

O Diagrama da figura 2.5 esquematiza o Modelo E-R acabado de descrever.

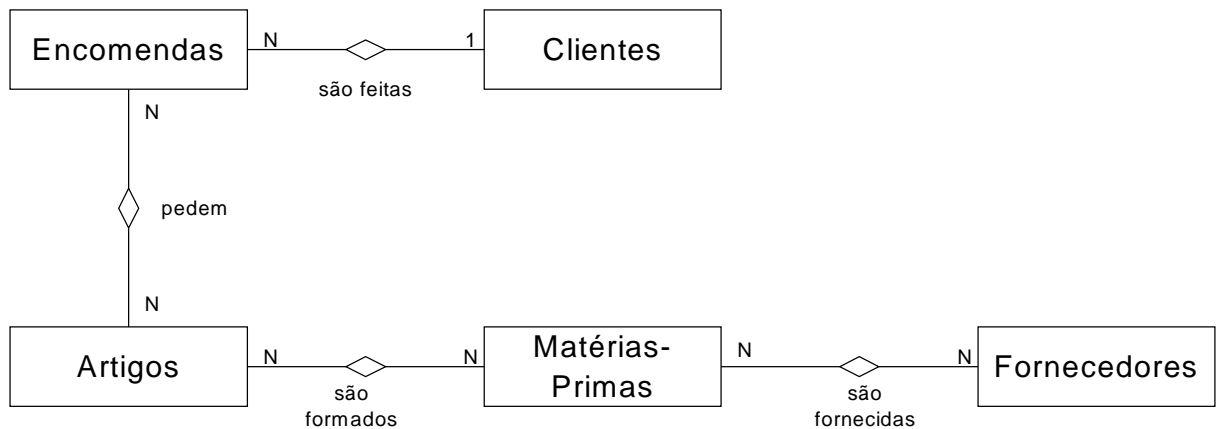


Figura 2.5: DER que modela o SI para gestão duma Casa Comercial

Base de Dados

Encomendas = { *NumEnc*; *DataEnc*; *CodC* }

Clientes = { *CodC*; *Nome*; *Telefone*; *Endereço* }

Encomendas/Artigos = { *CodE*; *CodA*; *QuantidEnc* }

Artigos = { *CodA*; *Nome*; *PreçoC*; *PreçoV* }

MatériasPrimas = { *CodMP*; *Nome*; *Tipo* }

Fornecedores = { *CodF*; *Nome*; *Telefone*; *Endereço* }

Artigos/MatPrimas = { *CodA*; *CodMP*; *QuantidUsada* }

MatPrimas/Fornecedores = { *CodMP*; *CodF*; *Preço* }

Questões

- a) JUNTANDO MatériasPrimas, Artigos, Artigos/MatPrimas
VER MatériasPrimas.CodMP, MatériasPrimas.Nome, MatériasPrimas.Tipo
ONDE (Artigos.Nome = "X")

- b) JUNTANDO Artigos, Encomendas, Clientes, Encomendas/Artigos
 VER Artigos.Nome, Artigos.PreçoV
 ONDE ((Clientes.Nome = "Y") E (Encomendas/Artigos.QuantidadeEnc > 25))
- c) JUNTANDO MatériasPrimas, Fornecedores, MatPrimas/Fornecedores
 VER Fornecedores.Nome, Fornecedores.Endereço, Fornecedores.Telefone
 ONDE (MatériasPrimas.Nome = "Z")

2.6 Pastelaria

Modelo E-R

Entidades:

Receitas = { *CodR*: string; Designação: string; TempoPrep: real; TempoRep: real; Peso: real; Calorias: real; Proteinas: real; Gorduras: real; Hcarbono: real; Custo: moeda }

Ingredientes = { *CodI*: string; Nome: str; Unidade: {g, dl }; Stock: int }

Encomendas = { *CodE*: string; dataE: data; Local: str; CP: str }

Clientes = { *CodC*: string; Nome: str; Morada: str; Telefone: str; Contribuinte: str }

Relações:

contêm: Receitas (*N*) → (*N*) Ingredientes ; atribs = { QuantidUsada: real }

pedem: Encomendas (*N*) → (*N*) Receitas ; atribs = { QuantidEnc: int }

são-feitas: Encomendas (*N*) → (1) Clientes ; atribs = { }

O Diagrama da figura 2.6 esquematiza o Modelo E-R acabado de descrever.

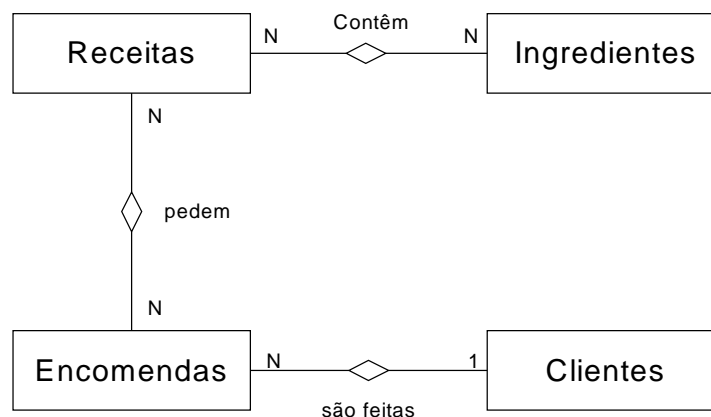


Figura 2.6: DER que modela o SI para gestão duma Pastelaria

Base de Dados

Receitas = { *CodR*; Designação; TempoP; TempoR; Peso; Cal; Pro; Gor; Hcar; Custo }

Ingredientes = { *CodI*; Nome; Unidade; Stock }

Clientes = { *CodC*; Nome; Morada; Telefone; Contribuinte }

Encomendas = { *CodE*; data; Local; CP; CodC }

Receitas/Ingredientes = { CodR; CodI; QuantidUsada }

Encomendas/Receitas = { CodE; CodR; QuantidEnc }

Questões

- JUNTANDO Receitas, Ingredientes
VER Ingredientes.Nome
ONDE (Receitas.Designação = "X")
- JUNTANDO Receitas, Clientes, Encomendas, Encomendas/Receitas
VER Receitas.Designação
ONDE ((Clientes.Nome = "Y") E (Encomendas/Receitas.Quantidade > 25))
- JUNTANDO Receitas, Clientes, Encomendas, Encomendas/Receitas
VER Receitas.Designação
ONDE ((Clientes.Nome = "Y") E (Receitas.Custo > 2500\$))

2.7 Centro de Saúde

Modelo E-R

Neste problema é o próprio enunciado que diz explicitamente que se pretende registar, manter e consultar certos dados (também explicitados) sobre os seguintes objectos: Doentes; Médicos; Consultas; Internamentos; e Medicamentos.

Cada um desses conjuntos de dados forma uma unidade de informação, que embora possa estar relacionada, é independente das demais; será por isso uma Entidade do SI.

Assim, chega-se fácil e directamente às Entidades que constituem a solução deste problema¹.

Entidades:

Doentes = { *NProcesso*: contador; Nome: str; DataNasc: data; Morada: str; Telefone: str; Observações: texto }

Médicos = { *CodM*: str; Nome: str; Espec: {oft, derm, nef, pneun, reum, card }; TM: str }

Consultas = { *NCons*: contador; Gabinete: str; Dia: data; Hora: hora; Preço: moeda; TCons: { prim, outras, seguim } }

Internamentos = { *NInt*: contador; DataE: data; DataS: data; NQuarto: str }

¹Para além dos atributos indicados no enunciado, acrescentaram-se mais alguns que entretanto pareceram necessários.

Medicamentos = { *CodFarm*: str; Designação: str; Indicações: texto }

Uma vez mais, seguindo as indicações dadas no enunciado (e recorrendo a algum conhecimento corrente e vulgar da realidade que nos cerca) é fácil reconhecer que uma Consulta envolve um Doente e um Médico, surgindo assim um par de relações binárias entre estas entidades. De igual modo se verifica que um Internamento diz respeito a um Doente, criando-se uma outra relação entre estas duas entidades, e que os Medicamentos são prescritos, ou receitados, aos Doentes, o que também origina uma outra relação entre estas duas entidades.

Em suma, podem neste problema identificar-se 4 relações a ligar as 5 entidades.

Relações:

são-destinadas: Consultas (*N*) → (1) Doentes ; atribs = { }

são-realizadas: Consultas (*N*) → (1) Médicos ; atribs = { }

são-ocupados: Internamentos (*N*) → (1) Doentes; atribs = { }

são-medicados: Doentes (*N*) → (*N*) Medicamentos ; atribs = { Posologia: str }

Obs: Posologia é uma fórmula do tipo

$$PA + A + L + J + C$$

que indica o número de comprimidos a tomar ao longo das 5 refeições do dia.

O Diagrama da figura 2.7 esquematiza o Modelo E-R acabado de descrever.

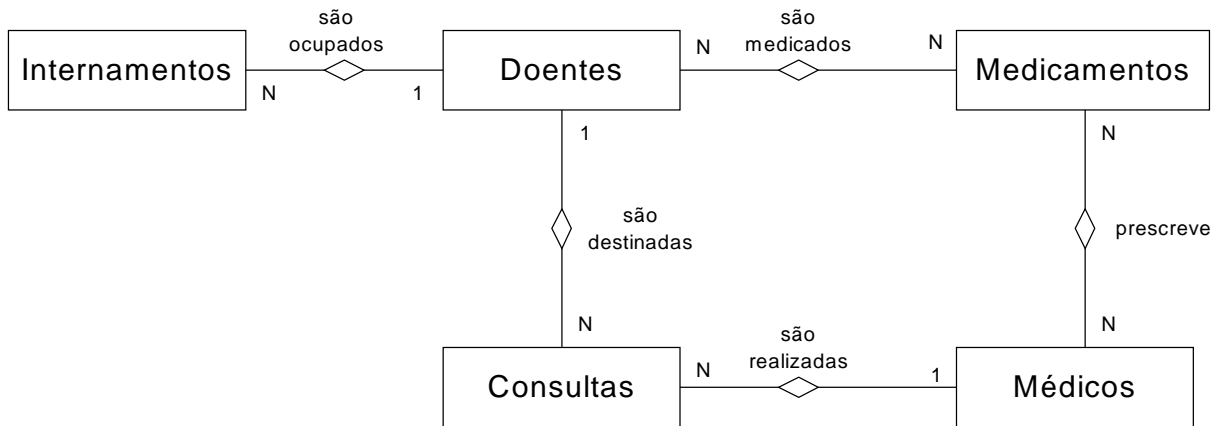


Figura 2.7: DER que modela o SI para gestão dum Centro de Saúde

Base de Dados

Construído o DER, é agora sistemática a forma de obter a Base de Dados que implementará o SI pedido.

Para implementar as 5 Entidades acima identificadas teremos as 5 tabelas seguintes:

Doentes = { *NProcesso*; Nome; DataNasc; Morada; Telefone; Observações }

Médicos = { *CodM*; Nome; Espec; TM }

Consultas = { *NCons*; Gabinete; Dia; Hora; Preço; TCons; *NProcesso*; *CodM* }

Internamentos = { *NInt*; DataE; DataS; NQuarto; *NProcesso* }

Medicamentos = { *CodFarm*; Designação; Indicações }

Os campos, ou colunas de cada uma deduzem-se directamente dos atributos escolhidos acima para caracterizar as respectivas Entidades.

Quanto a **Consultas**, esta tabela tem 2 campos extra —*Nprocesso* e *CodM*— que não são mais do que *Chaves Estrangeiras* que implementam as duas Relações N:1, **são-destinadas** e **são-realizadas**. Como a cada consulta só corresponde 1 Doente e só corresponde 1 Médico, basta usar estes campos extra, para estabelecer a ligação entre as tabelas Consultas e Doentes e Consultas e Médicos.

O mesmo sucede exactamente com a tabela **Internamentos** que tem um campo extra, *NProcesso*, para servir de *Chaves Estrangeira*, fazendo a ligação à entidade **Doentes** de modo a implementar a relação N:1 **são-ocupados** (a cada Internamento só corresponde 1 Doente).

Quanto à relação N:N **são-medicados** será mesmo necessário criar uma tabela extra com as *Chaves Estrangeiras* das 2 tabelas envolvidas, **Doentes** e **Medicamentos**, e com o atributo desta mesma relação:

Doentes/Medicamentos = { *NProcesso*; *CodFarm* , Posologia }

O par de Chaves Estrangeiras serve bem como Chave desta nova tabela.

Questões

Para formular as questões basta indicar os campos que se pretendem visualizar, precedidos do nome da tabela a que o campo pertence (o que é imediato perceber pela descrição feita no ponto anterior), exprimir as condições a que os registos a serem extraídos devem obdecer (comparando campos das tabelas com outros campos de outras tabelas ou com valores constantes) e, para terminar, indicar o nome de todas as tabelas que são necessárias, quer para ver os valores dos campos necessários, quer para escrever as condições.

Assim e atendendo aos pedidos feitos no enunciado do problema e ao desenho da BD elaborado acima, resultam as seguintes interrogações.

- a) JUNTANDO Doentes, Médicos, Consultas
VER Doentes.Nome, Consultas.Hora
ONDE ((Médicos.Nome = "X") E (Consultas.Dia = "Y"))
- b) JUNTANDO Doentes, Medicamentos, Doentes/Medicamentos
VER Medicamento.Designação, Medicamentos.Indicações
ONDE (Doentes.Nome = "W")
- c) JUNTANDO Doentes, Internamentos

VER Internamentos.NQuarto, Doentes.Nome
ONDE (TRUE)

Obs: note-se que nesta última questão a condição de seleção é **TRUE** para indicar que não se impõe nenhuma restrição, ou seja, que se quer seleccionar todos os registos que estejam ligados entre as duas tabelas.

2.8 Projectos de Investigação

Modelo E-R

Entidades:

Projectos = { *CodP*: contador; Título: str; Área: { mec, inf, aeronaut, hist, geol, electro }; Data: data; Duração: int; FPedido: moeda }

Instituições = { *CodI*: str; Nome: str; Morada: str; Telef: str; Fax: str; Email: str; Reitor: str; RelExt: str; NC: str; NIB: str }

Investigadores = { *CodInv*: str; Nome: str; Grau: { lic, mest, dout, agre }; Lugar: { ass, paux, passoc, pcat }; Idade: int }

Equipes = { *CodE*: contador; Nome: str }

Orçamentos = { *CodO*: str; RubricasValores: lista(CodRub: str, Valor: moeda) }

Relações:

pertencem: Projectos (N) \rightarrow (1) Instituições ; atribs = { }

pertencem: Orçamentos ($\underline{1}$) \rightarrow ($\underline{1}$) Projectos ; atribs = { }

pertencem: Investigadores (N) \rightarrow (1) Instituições ; atribs = { }

são-elaborados: Projectos ($\underline{1}$) \rightarrow ($\underline{1}$) Equipes ; atribs = { }

são-dirigidas: Equipes (1) \rightarrow ($\underline{1}$) Investigadores ; atribs = { }

são-constituídas: Equipes (N) \rightarrow (N) Investigadores ; atribs = { Cargo: { gestor, anal, impl, cons } }

O Diagrama da figura 2.8 esquematiza o Modelo E-R acabado de descrever.

Base de Dados

Neste problema a BD que vai ser proposta não resulta de uma implementação directa do Modelo ER acima, mas sim de algumas optimizações do primeiro desenho que se poderia ter obtido imediatamente.

As principais optimizações apareceram por se ter constatado que a relação **pertencem**, que liga

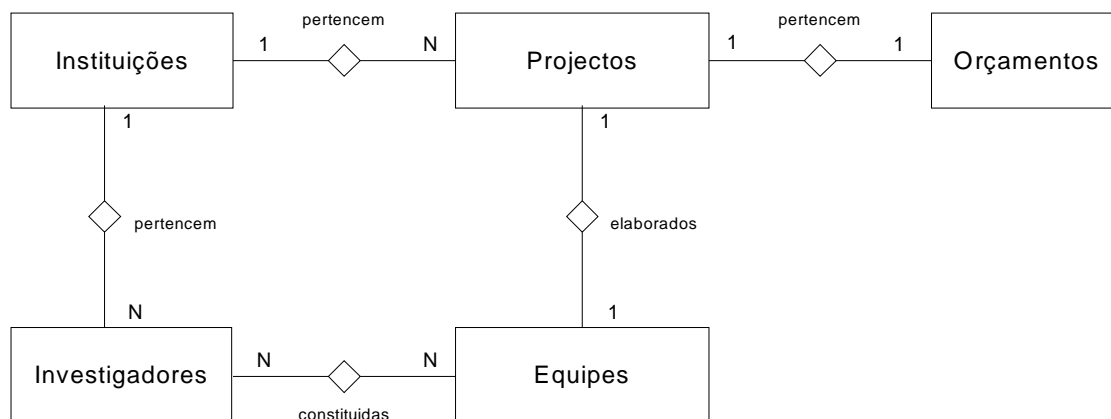


Figura 2.8: DER que modela o SI para gestão de Projectos de Investigação

Orçamentos e **Projectos**, e a relação **são-elaborados**, que liga **Projectos** e **Equipes**, serem relações (1:1) obrigatórias de ambos os lados. Como é sabido em tais situações as 2 Entidades podem fundir-se em 1 só, acrescentando a uma delas —a mais relevante para o problema (neste caso, **Projectos**)— os atributos da outra (é claro que então o código que representa o atributo chaves da Entidade que desaparece pode ser suprimido).

Note-se que também a relação **são-dirigidas**, que liga **Equipes** a **Investigadores** (para identificar o responsável), é do tipo (1:1) sendo sempre obrigatória a existência de um responsável. Por isso, o código desse responsável pode ser adicionado à informação da equipe, sem qualquer tabela extra.

Projectos = { *CodP*; Título; Área; Data; Duração; FPedido; CodI; NomeEquipe; CodInv(InvResp) }

Instituições = { *CodI*; Nome; Morada; Telef; Fax; Email; Reitor; RelExt; NC; NIB }

Investigadores = { *CodInv*; Nome; Grau; Lugar; Idade; CodI }

Na Tabela abaixo, que resultou da relação (N:N) **são-constituídas**, o código da Equipe é substituído pelo código do Projecto, devido à fusão de Equipe com Projecto, como explicado acima.

Equipes/Investigadores = { *CodP*; CodInv; Cargo }

A Tabela seguinte resulta apenas do facto do atributo **RubricasValores** da Entidade **Orçamentos** (que foi fundida com **Projectos**) ser um atributo do tipo *lista* (i.é, um Orçamento é composto por 1 ou mais (vários) pares formados pelo código da Rúbrica e pelo respectivo Valor orçamentado). Como é sabido em Tabelas de bases de dados relacionais não podem existir atributos deste tipo, pelo que é necessário criar uma Tabela adicional para guardar os vários pares associados a cada Orçamento (neste caso o seu código desapareceu devido à fusão referida e portanto é usado o código de Projecto).

Orçamentos = { *CodP*; CodRub; Valor }

Questões

- a) JUNTANDO Projectos, Instituições, Investigadores
VER Projectos.Título, Investigadores.Nome, Investigadores.Grau, Projectos.FPedido
ONDE (Instituição.Nome = "X")

2.9 Colóquio

Modelo E-R

Entidades:

Inscrições = { *CodInsc*: contador; *Nome*: str; *Morada*: str; *Telef*: str; *FormaP*: { num, che, vis } }

SitProfissional = { *CodSitP*: { est, prof, outra }; *Quantia*: moeda }

Historiadores = { *CodH*: contador; *Nome*: str }

Instituições = { *CodI*: str; *Nome*: str; *Morada*: str; *Telef*: str; *Fax*: str; *Email*: str }

Publicações = { *CodP*: str; *Título*: str; *Editora*: str; *Data*: data }

Pagamentos = { *CodP*: str }

Relações:

possuem: Inscritos (N) \rightarrow (1) SitProfissionais ; atribs = { }

pertencem: Historiadores (N) \rightarrow (1) Instituições ; atribs = { }

escrevem: Historiadores (N) \rightarrow (N) Publicações ; atribs = { }

O Diagrama da figura 2.9 esquematiza o Modelo E-R acabado de descrever.

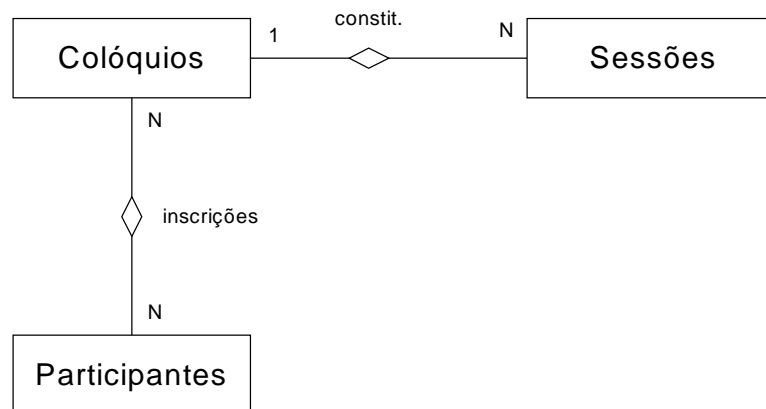


Figura 2.9: DER que modela o SI para gestão de um Colóquio

Base de Dados

Inscritos = { *CodInsc*; Nome; Morada; Telef; FormaP; CodSitProf }

SitProfissionais = { *CodSitProf*; Quantia }

Historiadores = { *CodH*; Nome; CodI }

Instituições = { *CodI*; Nome; Morada; Telef; Fax; Email }

Publicações = { *CodP*; Título; Editora; data }

Historiadores/Publicações = { *CodH*; *CodP*; }

2.10 Viagem

Modelo E-R

Entidades:

Peregrinos = { *CodPer*: contador; Nome: str; TipoP: { per, org, cavm, esc }; Morada: str; Telef: str; Idade: int, Prof: str; PontoP: str }

Doenças = { *CodD*: str; Nome: str; Sintomas: texto }

Alojamentos = { *CodA*: str; Nome: str; Local: str; Preço: moeda }

Pagamentos = { *CodP*: str; Quantia: moeda; Doação: moeda FormaP: { num, che, tb } }

Relações:

têm: Peregrinos (*N*) → (*N*) Doenças ; atribs = { Cuidados: texto }

ficam-alojados: Peregrinos (*N*) → (*N*) Alojamentos ; atribs = { Data: data }

fazem: Peregrinos (1) → (1) Pagamentos ; atribs = { }

O Diagrama da figura 2.10 esquematiza o Modelo E-R acabado de descrever.

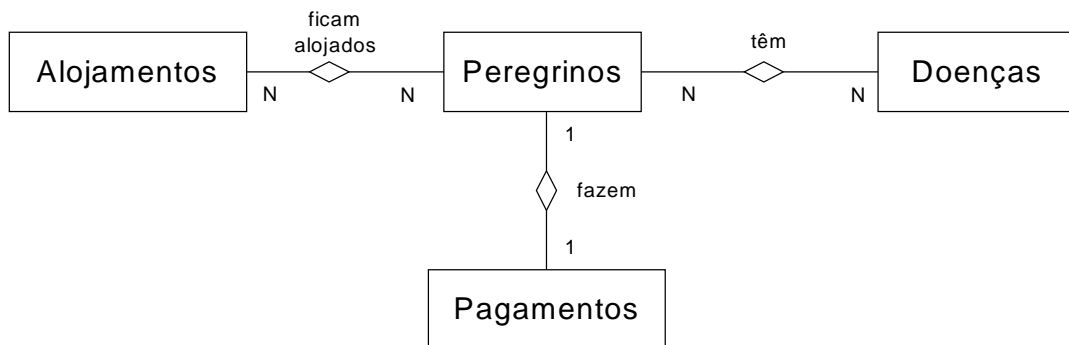


Figura 2.10: DER que modela o SI para gestão de uma Viagem de Peregrinação

Base de Dados

Peregrinos = { *CodPer*; Nome; TipoP; Morada; Telef; Idade , Prof; PontoP }

Pagamentos = { *CodPer*; Quantia; Doação; FormaP }

Doenças = { *CodD*; Nome; Sintomas }

Alojamentos = { *CodA*; Nome; Local }

Peregrinos/Doenças = { *CodPer*; *CodD*; Cuidados }

Peregrinos/Alojamentos = { *CodPer*; *CodA*; Data }

Neste caso, em alternativa às 2 primeiras tabelas, **Peregrinos** e **Pagamentos**, poderíamos fundir em 1 só, visto que a relação 1:1 é obrigatória em ambos os lados, isto é, cada peregrino faz 1 e 1 só pagamento e cada pagamento corresponde a 1 e 1 só peregrino.

Então, otimizando a solução, ficaria:

Peregrinos = { *CodPer*; Nome; TipoP; Morada; Telef; Idade , Prof; PontoP; Quantia; Doação; FormaP }