|  |
| --- |
| iselPreto.gif |
|  |
| Departamento de Engenharia de Electrónica e Telecomunicações e de Computadores  Lic. Engenharia Informática e Computadores |
| Sistemas de Informação I |
| Relatório Trabalho Prático  Parte 2 – Fase 1 |
| Semestre Inverno |
| 2009/2010 |
|  |
|  |
| Elaborado por:  Grupo 1 |
| 30896 – Ricardo Canto |
| 31401 – Nuno Cancelo |
| 33595 – Nuno Sousa |
|  |
|  |
| A/C: Eng. Nuno Datia |
| ISEL, 27 de Janeiro de 2010 |

Índice

[Introdução 3](#_Toc252230595)

[Etapa 1 – Revisões e Correcções ao Relatório da 1ª Fase 4](#_Toc252230596)

[Revisão do Modelo Conceptual 4](#_Toc252230597)

[Construção de Modelo Lógico 6](#_Toc252230598)

[Etapa 2 – Modelo Físico: Especificação SQL/DDL 7](#_Toc252230599)

[Script de criação 7](#_Toc252230600)

[Script de destruição 10](#_Toc252230601)

[Etapa 3 – Carregamento inicial da Base de Dados: Especificação SQL/DML 12](#_Toc252230602)

[Etapa 4 – Interrogações: Especificação SQL/DML 17](#_Toc252230603)

[Conclusão 21](#_Toc252230604)

# Introdução

A 2ª Parte do trabalho tem como objectivo a revisão do modelo apresentado na 1ªParte e a criação do modelo físico. A fase I é dividida em quatro etapas, sendo que a 1ª passa pela revisão e correcção do relatório da 1ª Parte, fazendo a revisão do modelo conceptual e construção do respectivo modelo lógico. A etapa 2 é a especificação em SQL/DDL do modelo relacional, criando scripts que permitam a criação e destruição do modelo físico. A 3ª etapa tem como objectivo o carregamento de dados para a realização de testes. Na 4ª e última etapa, passa pela criação de interrogações que permitam obter da base de dados a informação pretendida.

# Etapa 1 – Revisões e Correcções ao Relatório da 1ª Fase

O objectivo desta etapa é apresentar o modelo lógico a adoptar para esta segunda parte do trabalho. Inicialmente será revisto o modelo conceptual obtido na 1ª Parte do trabalho, e posteriormente, com base nas alterações efectuadas, será construído o modelo relacional, na terceira Forma Normal.

## Revisão do Modelo Conceptual

Analisemos as diferenças entre o modelo apresentado na 1ª parte e o modelo sugerido.

Entidade Paciente e entidade Terapeuta  
No modelo sugerido a entidade paciente surge de uma generalização, a entidade Pessoa, que terá também como sub-Entidade a especialização Terapeuta. Este tipo de modelação permite que por exemplo possamos ter uma pessoa que seja paciente e terapeuta, sendo que com a modelação proposta da 1ª parte teríamos de ter entradas em separado em cada tabela. Na entidade Terapeuta foi adicionado um atributo “emFuncoes” para que seja possível determinar se o terapeuta se encontra ainda a trabalhar na clínica. Na entidade Pessoa foi adicionado o atributo “dataNasc”.

Entidade Tratamento  
Em ambos os modelos opta-se por generalizações, com sub-Entidades Presencial e Não-Presencial, com cobertura total e generalização exclusiva. No entanto optamos por associar os tratamentos a uma especialização de forma a garantir que apenas o terapeuta habilitado (associado à mesma especialização) possa realizar o tratamento.

Entidade Sintoma  
Consideramos que a entidade Sintoma deverá ficar associada a Paciente já que ao inscrever-se na clínica, é recolhida informação sobre os sintomas actuais que o cliente demonstra.



## Construção de Modelo Lógico

* Sintoma (NumSintoma, Nome, Descrição)
  + **Chaves Candidatas** ={NumSintoma}
* PadraoClinico (NumPadrão, Designação, Descrição)
  + **Chaves Candidatas** ={NumSintoma}
* Sintomas\_Paciente ( NumSintoma, BIPaciente, DataSintoma, Notas )
  + **Chaves Candidatas** = { NumSintoma, BIPaciente}
  + **Chaves Estrangeiras** = {NumSintoma}, {BIPaciente}
* Sintomas\_Padrao( NumSintoma, NumPadrao)
  + **Chaves Candidatas** = { NumSintoma, NumPadrao }
  + **Chaves Estrangeiras** = { NumSintoma }, { NumPadrao }
  + **Obrigatoriedades**:
    - Um sintoma poderá ter um ou vários padrões clínicos associados
    - Um padrão clínico deverá ter um ou vários sintomas associados
* Pessoa (BI, NIF, nomePessoa, morada, telefone, idade, dataNasc)
  + **Chaves Candidatas** = { BI, NIF }
* Paciente (BIPaciente, Profissão, EstadoCivil, declRespons, Email, DataRegisto)
  + **Chaves Candidatas** = {BIPaciente}
  + **Chaves Estrangeiras =** {BIPaciente}
* Pacientes\_Sintomas\_Terapeuta (BIPaciente, numPadrao, BITerapeuta)
  + **Chaves Candidatas** = {numPadrão, BITerapeuta}
  + **Chaves Estrangeiras** = {numPadrão}, {BITerapeuta}, {BIPaciente}
  + **Nota**: Um ou vários terapeutas poderão associar um ou vários padrões a um paciente.
* Terapeuta (BITerapeuta, DataConclusao, emFuncoes)
  + **Chaves Candidatas** = {BITerapeuta}
  + **Chaves Estrangeiras =** {BITerapeuta}
  + **Nota**: O atributo “emFuncoes” é do tipo Booleano.
* Especializacao (Nome Especializacao)
  + **Chaves Candidatas** = {Nome Especializacao }
* Terapeuta\_Especializacao (BITerapeuta, NomeEspecializacao)
  + **Chaves Candidatas** = {BITerapeuta, NomeEspecializacao}
  + **Chaves Estrangeiras** = {BITerapeuta}, {NomeEspecializacao}
  + **Obrigatoriedade**: Um terapeuta terá de ter associado uma especialização. No entanto, uma especialização poderá não ter um terapeuta associado.
* Tratamento (NumTratamento, DescTratamento, NomeEspecializacao, GeneroTratamento)
  + **Chaves Candidatas** = {NumTratamento}
  + **Chaves Estrangeiras** = {NomeEspecialização}
  + **Obrigatoriedades**:
    - A um tratamento terá de ficar associada uma Especialização.
    - GeneroTratamento tem dois valores possíveis: “Presencial” ou “Não Presencial”
  + **Nota**: Um terapeuta está associado a uma especialização, que determina os tratamentos que o terapeuta está habilitado a executar.
* Presencial (NumTratamento, TipoTratamento)
  + **Chaves Candidatas** = {NumTratamento}
  + **Chaves Estrangeiras** = {NumTratamento}
* NaoPresencial (numTratamento, numFormula, nomeChines, composição)
  + **Chaves Candidatas** = {NumTratamento}, {NumFormula}
  + **Chaves Estrangeiras** = {NumTratamento}
* Tratamento\_Padrao (NumTratamento, NumPadrao)
  + **Chaves Candidatas** = {NumTratamento, NumPadrao)
  + **Chaves Estrangeiras** = {NumTratamento},{NumFormula}
  + **Obrigatoriedade**: A um tratamento tem de estar associado um ou mais padrões clínicos.
* Consulta (BIPaciente, numConsulta, dataConsulta, relatório, BITerapeuta)
  + **Chaves Candidatas** = {BIPaciente, numConsulta}
  + **Chaves Estrangeiras** = {BIPaciente}, {BITerapeuta}
  + **Obrigatoriedades**: A uma consulta tem de estar associado um terapeuta.
* Consulta\_NaoPresencial (BIPaciente, numConsulta, NumTratamento, Quantidade, VezesDia, Ocasioes)
  + **Chaves Candidatas** = { BIPaciente, numConsulta, NumTratamento }
  + **Chaves Estrangeiras** = {BIPaciente}
* Sessao (numTratamento, numSessão, numConsulta , Duração, Descrição, BITerapeuta)
  + **Chaves Candidatas** = { numTratamento, numSessão, numConsulta}
  + **Chaves Estrangeiras** = {numTratamento}, {numSessão}, {BITerapeuta}, {numConsulta}
  + **Obrigatoriedades**: É necessário que a uma sessão esteja sempre associado um terapeuta, não sendo necessário que seja o responsável pela consulta.
  + **Nota**: Consideramos que sessão está sempre associada a uma consulta, podendo haver diversas sessões de tratamento numa consulta.

# Etapa 2 – Modelo Físico: Especificação SQL/DDL

## Script de criação

/\*

\* Sistemas de Informação I

\* 2009/2010 Semestre Inverno

\* Trabalho Prático (Parte II)

\*

\* Elaborado por:

\* Grupo 1

\* 30896 – Ricardo Canto

\* 31401 – Nuno Cancelo

\* 33595 – Nuno Sousa

\*/

use Zhongyixue

BEGIN TRANSACTION createTables

CREATE TABLE SINTOMA(

numSintoma int CONSTRAINT pk\_SINTOMA PRIMARY KEY,

nomeSintoma varchar(20) not null,

descSintoma varchar(500)

) ;

CREATE TABLE PADRAO(

numPadrao int CONSTRAINT pk\_PADRAO PRIMARY KEY,

nomePadrao varchar(20) not null,

descPadrao varchar(500)

) ;

CREATE TABLE SINTOMA\_PADRAO(

numPadrao int,

numSintoma int,

CONSTRAINT pk\_SINTOMA\_PADRAO PRIMARY KEY ( numPadrao, numSintoma ),

CONSTRAINT fk1\_SINTOMA\_PADRAO FOREIGN KEY ( numPadrao )

REFERENCES PADRAO ( numPadrao ),

CONSTRAINT fk2\_SINTOMA\_PADRAO FOREIGN KEY ( numSintoma )

REFERENCES SINTOMA ( numSintoma )

/\*OBRIGATORIEDADES:

Um sintoma poderá ter um ou vários padões clinicos associados;

Um padrão clinico poderá ter um ou vários sintomas associados.

\*/

);

CREATE TABLE PESSOA(

BI varchar(10) CONSTRAINT pk\_PESSOA PRIMARY KEY,

NIF varchar(10) CONSTRAINT ak\_PESSOA UNIQUE,

nomePessoa varchar(50) not null,

morada varchar(30) not null,

telefone varchar(15) not null, /\*O número poderá ser internacional\*/

idade tinyint not null,

dataNasc date not null

) ;

CREATE TABLE PACIENTE(

BI varchar(10) not null,

profissao varchar(20) not null,

estadoCivil varchar(15) not null,

/\*solteiro, divorciado, casado, união facto, viuvo, separado\*/

declRespon varbinary(max),

email varchar(20),

dataRegisto date not null,

CONSTRAINT pk\_PACIENTE PRIMARY KEY ( BI ),

CONSTRAINT fk\_PACIENTE FOREIGN KEY ( BI )

REFERENCES PESSOA ( BI )

) ;

CREATE TABLE SINTOMA\_PACIENTE(

numSintoma int not null,

BIPaciente varchar(10) not null,

dataSintoma date not null,

notas varchar(500),

CONSTRAINT pk\_SINTOMA\_PACIENTE PRIMARY KEY (numSintoma, BIPaciente),

CONSTRAINT fk1\_SINTOMA\_PACIENTE FOREIGN KEY (numSintoma)

REFERENCES SINTOMA (numSintoma),

CONSTRAINT fk2\_SINTOMA\_PACIENTE FOREIGN KEY (BIPaciente)

REFERENCES PACIENTE (BI)

) ;

CREATE TABLE TERAPEUTA(

BI varchar(10) not null,

dataConclusao date not null,

/\*dataConclusao varchar(20) not null,\*/

emFuncoes bit not null,

/\*1: Em funções, 0: Já não trabalha na clínica\*/

CONSTRAINT pk\_TERAPEUTA PRIMARY KEY ( BI ),

CONSTRAINT fk\_TERAPEUTA FOREIGN KEY ( BI )

REFERENCES PESSOA ( BI ),

) ;

CREATE TABLE PACIENTE\_SINTOMA\_TERAPEUTA(

BIPaciente varchar(10) not null,

numPadrao int not null,

BITerapeuta varchar(10) not null,

CONSTRAINT pk\_PACIENTE\_SINTOMA\_TERAPEUTA PRIMARY KEY (BITerapeuta, numPadrao),

CONSTRAINT fk1\_PACIENTE\_SINTOMA\_TERAPEUTA FOREIGN KEY (BITerapeuta)

REFERENCES TERAPEUTA (BI),

CONSTRAINT fk2\_PACIENTE\_SINTOMA\_TERAPEUTA FOREIGN KEY (numPadrao)

REFERENCES PADRAO (numPadrao)

);

CREATE TABLE ESPECIALIZACAO(

nomeEspecializacao varchar(20) CONSTRAINT pk\_ESPECIALIZACAO PRIMARY KEY

);

CREATE TABLE TERAPEUTA\_ESPECIALIZACAO(

BITerapeuta varchar(10),

nomeEspecializacao varchar(20),

CONSTRAINT fk1\_TERAPEUTA\_ESPECIALIZACAO FOREIGN KEY ( BITerapeuta )

REFERENCES TERAPEUTA ( BI ),

CONSTRAINT fk2\_TERAPEUTA\_ESPECIALIZACAO FOREIGN KEY ( nomeEspecializacao )

REFERENCES ESPECIALIZACAO ( nomeEspecializacao )

);

CREATE TABLE TRATAMENTO(

numTratamento int,

descTratamento varchar(500) not null,

tipoTramento varchar(15) not null,

/\*"Presencial" ou "Não Presencial"\*/

nomeEspecializacao varchar(20),

CONSTRAINT pk\_TRATAMENTO PRIMARY KEY ( numTratamento ),

CONSTRAINT fk\_TRATAMENTO FOREIGN KEY ( nomeEspecializacao )

REFERENCES ESPECIALIZACAO ( nomeEspecializacao )

);

CREATE TABLE PRESENCIAL(

numTratamento int,

tipoTratamento varchar(11),

/\*"Massagem", "Acupunctura" ou "Moxabustão"\*/

CONSTRAINT pk\_PRESENCIAL PRIMARY KEY ( numTratamento ),

CONSTRAINT fk\_PRESENCIAL FOREIGN KEY ( numTratamento )

REFERENCES TRATAMENTO ( numTratamento )

);

CREATE TABLE NAOPRESENCIAL(

numTratamento int,

numFormula int,

nomeFormula varchar(20) not null,

composicao varchar(100) not null,

CONSTRAINT pk\_NAOPRESENCIAL PRIMARY KEY ( numTratamento ),

CONSTRAINT fk\_NAOPRESENCIAL FOREIGN KEY ( numTratamento )

REFERENCES TRATAMENTO ( numTratamento ),

CONSTRAINT ak\_NAOPRESENCIAL UNIQUE ( numFormula ),

);

CREATE TABLE TRATAMENTO\_PADRAO(

numTratamento int,

numPadrao int,

CONSTRAINT fk1\_TRATAMENTO\_PADRAO FOREIGN KEY ( numTratamento )

REFERENCES TRATAMENTO ( numTratamento ),

CONSTRAINT fk2\_TRATAMENTO\_PADRAO FOREIGN KEY ( numPadrao )

REFERENCES PADRAO ( numPadrao )

);

CREATE TABLE CONSULTA(

BIPaciente varchar(10),

BITerapeuta varchar(10),

numConsulta int not null,

dataConsulta date not null,

relatorio varchar(500),

CONSTRAINT pk\_CONSULTA PRIMARY KEY ( BIPaciente, numConsulta ),

CONSTRAINT fk1\_CONSULTA FOREIGN KEY ( BIPaciente )

REFERENCES PACIENTE ( BI ),

CONSTRAINT fk2\_CONSULTA FOREIGN KEY ( BITerapeuta )

REFERENCES TERAPEUTA ( BI ),

);

CREATE TABLE CONSULTA\_NAOPRESENCIAL(

BIPaciente varchar(10),

numConsulta int,

numTratamento int,

quantidade int not null,

vezesDia tinyint not null,

ocasioes varchar(20) not null,

CONSTRAINT pk\_CONSULTA\_NAOPRESENCIAL PRIMARY KEY ( BIPaciente, numConsulta, numTratamento ),

CONSTRAINT fk1\_CONSULTA\_NAOPRESENCIAL FOREIGN KEY ( BIPaciente, numConsulta )

REFERENCES CONSULTA ( BIPaciente, numConsulta ),

CONSTRAINT fk2\_CONSULTA\_NAOPRESENCIAL FOREIGN KEY ( numTratamento )

REFERENCES NAOPRESENCIAL ( numTratamento )

);

CREATE TABLE SESSAO(

numSessao int,

numTratamento int,

numConsulta int,

BIPaciente varchar(10),

BITerapeuta varchar(10),

duracao time,

descSessao varchar(500) not null,

CONSTRAINT pk\_SESSAO PRIMARY KEY ( numSessao, numTratamento, numConsulta ),

CONSTRAINT fk1\_SESSAO FOREIGN KEY ( numTratamento )

REFERENCES PRESENCIAL ( numTratamento ),

CONSTRAINT fk2\_SESSAO FOREIGN KEY ( BIPaciente, numConsulta )

REFERENCES CONSULTA ( BIPaciente, numConsulta ),

CONSTRAINT fk3\_SESSAO FOREIGN KEY ( BITerapeuta )

REFERENCES TERAPEUTA ( BI )

);

COMMIT TRANSACTION createTables ;

## Script de destruição

/\*

\* Sistemas de Informação I

\* 2009/2010 Semestre Inverno

\* Trabalho Prático (Parte II)

\*

\* Elaborado por:

\* Grupo 1

\* 30896 – Ricardo Canto

\* 31401 – Nuno Cancelo

\* 33595 – Nuno Sousa

\*/

USE Zhongyixue ;

GO

IF OBJECT\_ID('SINTOMA\_PACIENTE') IS NOT NULL BEGIN

DROP TABLE SINTOMA\_PACIENTE ;

END ;

IF OBJECT\_ID('SINTOMA\_PADRAO') IS NOT NULL BEGIN

DROP TABLE SINTOMA\_PADRAO ;

END ;

IF OBJECT\_ID('PACIENTE\_SINTOMA\_TERAPEUTA') IS NOT NULL BEGIN

DROP TABLE PACIENTE\_SINTOMA\_TERAPEUTA ;

END ;

IF OBJECT\_ID('TERAPEUTA\_ESPECIALIZACAO') IS NOT NULL BEGIN

DROP TABLE TERAPEUTA\_ESPECIALIZACAO ;

END ;

IF OBJECT\_ID('TRATAMENTO\_PADRAO') IS NOT NULL BEGIN

DROP TABLE TRATAMENTO\_PADRAO ;

END ;

IF OBJECT\_ID('SINTOMA') IS NOT NULL BEGIN

DROP TABLE SINTOMA ;

END ;

IF OBJECT\_ID('CONSULTA\_NAOPRESENCIAL') IS NOT NULL BEGIN

DROP TABLE CONSULTA\_NAOPRESENCIAL ;

END ;

IF OBJECT\_ID('SESSAO') IS NOT NULL BEGIN

DROP TABLE SESSAO ;

END ;

IF OBJECT\_ID('PADRAO') IS NOT NULL BEGIN

DROP TABLE PADRAO ;

END ;

IF OBJECT\_ID('NAOPRESENCIAL') IS NOT NULL BEGIN

DROP TABLE NAOPRESENCIAL ;

END ;

IF OBJECT\_ID('PRESENCIAL') IS NOT NULL BEGIN

DROP TABLE PRESENCIAL ;

END ;

IF OBJECT\_ID('TRATAMENTO') IS NOT NULL BEGIN

DROP TABLE TRATAMENTO ;

END ;

IF OBJECT\_ID('ESPECIALIZACAO') IS NOT NULL BEGIN

DROP TABLE ESPECIALIZACAO ;

END ;

IF OBJECT\_ID('CONSULTA') IS NOT NULL BEGIN

DROP TABLE CONSULTA ;

END ;

IF OBJECT\_ID('PACIENTE') IS NOT NULL BEGIN

DROP TABLE PACIENTE ;

END ;

IF OBJECT\_ID('TERAPEUTA') IS NOT NULL BEGIN

DROP TABLE TERAPEUTA ;

END ;

IF OBJECT\_ID('PESSOA') IS NOT NULL BEGIN

DROP TABLE PESSOA ;

END ;

# Etapa 3 – Carregamento inicial da Base de Dados: Especificação SQL/DML

/\*

\* Sistemas de Informação I

\* 2009/2010 Semestre Inverno

\* Trabalho Prático (Parte II)

\*

\* Elaborado por:

\* Grupo 1

\* 30896 – Ricardo Canto

\* 31401 – Nuno Cancelo

\* 33595 – Nuno Sousa

\*/

use Zhongyixue ;

set nocount on;

set xact\_abort on;

set dateformat dmy;

go

begin transaction populateTables

raiserror('Starting...',0,1) --info

/\*

\* dbo.SINTOMA

\*/

insert into dbo.SINTOMA( numSintoma, nomeSintoma, descSintoma ) values(1,'Febre','Descrição do sintoma Febre');

insert into dbo.SINTOMA( numSintoma, nomeSintoma, descSintoma ) values(2,'Dor Abdominal','Descrição do sintoma Dor Abdominal');

insert into dbo.SINTOMA( numSintoma, nomeSintoma, descSintoma ) values(3,'Dor de cabeça','Descrição do sintoma Dor de Cabeça' );

insert into dbo.SINTOMA( numSintoma, nomeSintoma, descSintoma ) values(4,'Desmaio','Descrição do sintoma Desmaio' );

insert into dbo.SINTOMA( numSintoma, nomeSintoma, descSintoma ) values(5,'Perda de Apetite','Descrição do sintoma Perda de Apetite' );

insert into dbo.SINTOMA( numSintoma, nomeSintoma, descSintoma ) values(6,'Astenia','Descrição do sintoma Astenia' );

/\*

\* dbo.PADRAO

\*/

insert into dbo.PADRAO( numPadrao, nomePadrao, descPadrao ) values (1,'Estase de Sangue','Padrao Clinico Estase de Sangue')

insert into dbo.PADRAO( numPadrao, nomePadrao, descPadrao ) values (2,'Calor no Sangue ','Padrao Clinico Calor no Sangue ')

insert into dbo.PADRAO( numPadrao, nomePadrao, descPadrao ) values (3,'Qi rebelde','Padrao Clinico Qi rebelde')

insert into dbo.PADRAO( numPadrao, nomePadrao, descPadrao ) values (4,'Vazio de Sangue','Padrao Clinico Vazio de Sangue')

insert into dbo.PADRAO( numPadrao, nomePadrao, descPadrao ) values (5,'Afundamento do Qi','Padrao Clinico Afundamento do Qi')

/\*

\* dbo.SINTOMA\_PADRAO

\*/

insert into dbo.SINTOMA\_PADRAO( numPadrao, numSintoma ) values (1,1);

insert into dbo.SINTOMA\_PADRAO( numPadrao, numSintoma ) values (1,3);

insert into dbo.SINTOMA\_PADRAO( numPadrao, numSintoma ) values (2,1);

insert into dbo.SINTOMA\_PADRAO( numPadrao, numSintoma ) values (3,2);

insert into dbo.SINTOMA\_PADRAO( numPadrao, numSintoma ) values (4,4);

insert into dbo.SINTOMA\_PADRAO( numPadrao, numSintoma ) values (4,6);

insert into dbo.SINTOMA\_PADRAO( numPadrao, numSintoma ) values (5,2);

/\*

\* dbo.PESSOA

\*/

insert into dbo.PESSOA(BI,NIF,nomePessoa,morada,telefone,idade,dataNasc) values ('12345678-9','12345678','José Gomercindo','Rua B, Lote 23','+351 219983879',134,'05-10-1876') ;

insert into dbo.PESSOA(BI,NIF,nomePessoa,morada,telefone,idade,dataNasc) values ('13345679-9','13345778','Aníbel Fonseca','Rua F, Lote 4','+44 20083879',45,'06-11-1965') ;

insert into dbo.PESSOA(BI,NIF,nomePessoa,morada,telefone,idade,dataNasc) values ('14345680-9','14345878','Cátia Vanessa','Rua J, Lote 67','+351 201238579',27,'11-05-1982') ;

insert into dbo.PESSOA(BI,NIF,nomePessoa,morada,telefone,idade,dataNasc) values ('15345681-9','15345978','Urubina João','Rua B, Lote 23','+351 219543879',56,'10-10-1954') ;

insert into dbo.PESSOA(BI,NIF,nomePessoa,morada,telefone,idade,dataNasc) values ('16345682-9','16346078','Boaventura Francisco','Rua F, Lote 54','+351 219984879',34,'06-07-1975') ;

insert into dbo.PESSOA(BI,NIF,nomePessoa,morada,telefone,idade,dataNasc) values ('16345700-9','16346700','Joaquina Felisberta','Rua A, Lote 3','+351 210084879',34,'06-06-1975') ;

insert into dbo.PESSOA(BI,NIF,nomePessoa,morada,telefone,idade,dataNasc) values ('34343434-3','34343434','Miguel Micael','Rua K, Nº98','+351 098987786',87,'05-05-1922') ;

/\*

\* dbo.PACIENTE

\*/

insert into dbo.PACIENTE(BI,profissao,estadoCivil,declRespon,email,dataRegisto) values ('12345678-9','Sapateiro','Casado',null,'gomercindo@si1.pt','01-01-2001');

insert into dbo.PACIENTE(BI,profissao,estadoCivil,declRespon,email,dataRegisto) values ('13345679-9','Estudante','Separado',null,null,'01-01-2009');

insert into dbo.PACIENTE(BI,profissao,estadoCivil,declRespon,email,dataRegisto) values ('14345680-9','Astronauta','Casado',null,null,'01-01-2010');

insert into dbo.PACIENTE(BI,profissao,estadoCivil,declRespon,email,dataRegisto) values ('34343434-3','Saltibancos','Viúvo',null,'1@2.pt','05-07-2009');

/\*

\* dbo.SINTOMA\_PACIENTE

\*/

insert into dbo.SINTOMA\_PACIENTE(numSintoma,BIPaciente,dataSintoma,notas) values (4,'12345678-9','01-01-2010','xpto');

insert into dbo.SINTOMA\_PACIENTE(numSintoma,BIPaciente,dataSintoma,notas) values (6,'14345680-9','12-12-2009','abcd');

insert into dbo.SINTOMA\_PACIENTE(numSintoma,BIPaciente,dataSintoma,notas) values (3,'13345679-9','08-08-2009','qwer');

/\*

\* dbo.TERAPEUTA

\*/

insert into dbo.TERAPEUTA(BI,dataConclusao,emFuncoes) values ('15345681-9','01-12-1972',1);

insert into dbo.TERAPEUTA(BI,dataConclusao,emFuncoes) values ('16345682-9','01-10-1985',1);

insert into dbo.TERAPEUTA(BI,dataConclusao,emFuncoes) values ('16345700-9','01-10-1945',1);

insert into dbo.TERAPEUTA(BI,dataConclusao,emFuncoes) values ('34343434-3','11-05-1982',0);

/\*\*

\* dbo.PACIENTE\_SINTOMA\_TERAPEUTA

\*/

insert into dbo.PACIENTE\_SINTOMA\_TERAPEUTA(BIPaciente,numPadrao,BITerapeuta) values ('12345678-9',1,'15345681-9');

insert into dbo.PACIENTE\_SINTOMA\_TERAPEUTA(BIPaciente,numPadrao,BITerapeuta) values ('14345680-9',3,'15345681-9');

/\*\*

\* dbo.ESPECIALIZACAO

\*/

insert into dbo.ESPECIALIZACAO(nomeEspecializacao) values ('A');

insert into dbo.ESPECIALIZACAO(nomeEspecializacao) values ('B');

insert into dbo.ESPECIALIZACAO(nomeEspecializacao) values ('C');

/\*\*

\* dbo.TERAPEUTA\_ESPECIALIZACAO

\*/

insert into dbo.TERAPEUTA\_ESPECIALIZACAO(BITerapeuta, nomeEspecializacao) values ('15345681-9','A');

insert into dbo.TERAPEUTA\_ESPECIALIZACAO(BITerapeuta, nomeEspecializacao) values ('16345682-9','B');

insert into dbo.TERAPEUTA\_ESPECIALIZACAO(BITerapeuta, nomeEspecializacao) values ('16345700-9','B');

insert into dbo.TERAPEUTA\_ESPECIALIZACAO(BITerapeuta, nomeEspecializacao) values ('16345700-9','A');

insert into dbo.TERAPEUTA\_ESPECIALIZACAO(BITerapeuta, nomeEspecializacao) values ('16345700-9','C');

insert into dbo.TERAPEUTA\_ESPECIALIZACAO(BITerapeuta, nomeEspecializacao) values ('34343434-3','C');

insert into dbo.TERAPEUTA\_ESPECIALIZACAO(BITerapeuta, nomeEspecializacao) values ('34343434-3','B');

insert into dbo.TERAPEUTA\_ESPECIALIZACAO(BITerapeuta, nomeEspecializacao) values ('34343434-3','A');

/\*\*

\* dbo.TRATAMENTO

\*/

insert into dbo.TRATAMENTO(numTratamento,descTratamento,tipoTramento,nomeEspecializacao) values (1,'Tratamento 1','Presencial','B');

insert into dbo.TRATAMENTO(numTratamento,descTratamento,tipoTramento,nomeEspecializacao) values (2,'Tratamento 2','Presencial','B');

insert into dbo.TRATAMENTO(numTratamento,descTratamento,tipoTramento,nomeEspecializacao) values (3,'Tratamento 3','Não Presencial','A');

insert into dbo.TRATAMENTO(numTratamento,descTratamento,tipoTramento,nomeEspecializacao) values (4,'Tratamento 4','Presencial','A');

insert into dbo.TRATAMENTO(numTratamento,descTratamento,tipoTramento,nomeEspecializacao) values (5,'Tratamento 5','Não Presencial','B');

insert into dbo.TRATAMENTO(numTratamento,descTratamento,tipoTramento,nomeEspecializacao) values (6,'Tratamento 6','Não Presencial','C');

/\*\*

\* dbo.PRESENCIAL

\*/

insert into dbo.PRESENCIAL(numTratamento,tipoTratamento) values (1,'Moxabustão');

insert into dbo.PRESENCIAL(numTratamento,tipoTratamento) values (2,'Acupunctura');

insert into dbo.PRESENCIAL(numTratamento,tipoTratamento) values (4,'Massagem');

/\*\*

\* dbo.NAOPRESENCIAL

\*/

insert into dbo.NAOPRESENCIAL(numTratamento,numFormula,nomeFormula,composicao) values (3,1,'Tai Hoy Xue','dasd');

insert into dbo.NAOPRESENCIAL(numTratamento,numFormula,nomeFormula,composicao) values (5,2,'Ju Goi To','adasdasdasd');

insert into dbo.NAOPRESENCIAL(numTratamento,numFormula,nomeFormula,composicao) values (6,33,'Kio Ji Ti','adsadasdas');

/\*\*

\* dbo.TRATAMENTO\_PADRAO

\*/

insert into dbo.TRATAMENTO\_PADRAO(numTratamento,numPadrao) values (1,4);

insert into dbo.TRATAMENTO\_PADRAO(numTratamento,numPadrao) values (2,3);

insert into dbo.TRATAMENTO\_PADRAO(numTratamento,numPadrao) values (4,5);

/\*\*

\* dbo.CONSULTA

\*/

insert into dbo.CONSULTA(BIPaciente, BITerapeuta, numConsulta, dataConsulta, relatorio) values ('12345678-9','15345681-9',1,'01-11-2009','');

insert into dbo.CONSULTA(BIPaciente, BITerapeuta, numConsulta, dataConsulta, relatorio) values ('12345678-9','16345682-9',2,'01-05-2009','');

insert into dbo.CONSULTA(BIPaciente, BITerapeuta, numConsulta, dataConsulta, relatorio) values ('14345680-9','34343434-3',3,'23-02-2009','');

insert into dbo.CONSULTA(BIPaciente, BITerapeuta, numConsulta, dataConsulta, relatorio) values ('14345680-9','34343434-3',4,'30-06-2009','');

insert into dbo.CONSULTA(BIPaciente, BITerapeuta, numConsulta, dataConsulta, relatorio) values ('14345680-9','34343434-3',5,'20-10-2009','');

insert into dbo.CONSULTA(BIPaciente, BITerapeuta, numConsulta, dataConsulta, relatorio) values ('14345680-9','34343434-3',6,'18-01-2010','');

insert into dbo.CONSULTA(BIPaciente, BITerapeuta, numConsulta, dataConsulta, relatorio) values ('14345680-9','34343434-3',7,'04-01-2009','');

insert into dbo.CONSULTA(BIPaciente, BITerapeuta, numConsulta, dataConsulta, relatorio) values ('34343434-3','16345700-9',8,'04-11-2009','');

/\*\*

\* dbo.CONSULTA\_NAOPRESENCIAL

\*/

insert into dbo.CONSULTA\_NAOPRESENCIAL(BIPaciente,numConsulta,numTratamento,quantidade,vezesDia,ocasioes) values ('12345678-9',2,3,400,2,'almoço e jantar') ;

insert into dbo.CONSULTA\_NAOPRESENCIAL(BIPaciente,numConsulta,numTratamento,quantidade,vezesDia,ocasioes) values ('12345678-9',2,6,500,2,'almoço e jantar') ;

insert into dbo.CONSULTA\_NAOPRESENCIAL(BIPaciente,numConsulta,numTratamento,quantidade,vezesDia,ocasioes) values ('14345680-9',4,6,400,2,'almoço e jantar') ;

insert into dbo.CONSULTA\_NAOPRESENCIAL(BIPaciente,numConsulta,numTratamento,quantidade,vezesDia,ocasioes) values ('14345680-9',5,6,200,2,'almoço e jantar') ;

insert into dbo.CONSULTA\_NAOPRESENCIAL(BIPaciente,numConsulta,numTratamento,quantidade,vezesDia,ocasioes) values ('14345680-9',6,6,100,2,'almoço e jantar') ;

insert into dbo.CONSULTA\_NAOPRESENCIAL(BIPaciente,numConsulta,numTratamento,quantidade,vezesDia,ocasioes) values ('14345680-9',7,6,100,2,'almoço e jantar') ;

/\*\*

\* dbo.SESSAO

\*/

insert into dbo.SESSAO(numSessao,numTratamento,numConsulta,BIPaciente,BITerapeuta,duracao,descSessao) values (1,1,2,'12345678-9','16345682-9','01:30','asçldjalksfjal') ;

insert into dbo.SESSAO(numSessao,numTratamento,numConsulta,BIPaciente,BITerapeuta,duracao,descSessao) values (2,2,3,'14345680-9','34343434-3','02:30','Adsasdl') ;

insert into dbo.SESSAO(numSessao,numTratamento,numConsulta,BIPaciente,BITerapeuta,duracao,descSessao) values (3,2,8,'34343434-3','16345700-9','00:30','Adsasdl') ;

raiserror('... OK',0,1) --info

commit

# Etapa 4 – Interrogações: Especificação SQL/DML

1. Quais os pacientes que ainda não efectuaram nenhuma consulta. *Nota: apresentar informação sobre BI, nome e data de registo no centro.*

SELECT PESSOA.BI,PESSOA.nomePessoa,PACIENTE.dataRegisto

FROM

PESSOA INNER JOIN PACIENTE ON (PESSOA.BI = PACIENTE.BI)

WHERE

PACIENTE.BI NOT IN (

SELECT DISTINCT CONSULTA.BIPaciente

FROM CONSULTA

);

1. Quais os tratamentos e respectivos terapeutas habilitados (BI e nome), para cada um dos padrões clínicos já diagnosticados a algum paciente, desde 1 de Dezembro de 2009. *Nota: apresentar também a informação sobre o respectivo padrão clínico.*

SELECT TERAPEUTA.BI, PESSOA.nomePessoa as Terapeuta,TRATAMENTO.descTratamento as Tratamento,PADRAO.nomePadrao as Padrão,PADRAO.descPadrao

FROM

PESSOA INNER JOIN TERAPEUTA ON (PESSOA.BI = TERAPEUTA.BI)

INNER JOIN TERAPEUTA\_ESPECIALIZACAO ON (TERAPEUTA.BI =

TERAPEUTA\_ESPECIALIZACAO.BITerapeuta)

INNER JOIN TRATAMENTO ON (TERAPEUTA\_ESPECIALIZACAO.nomeEspecializacao

=TRATAMENTO.nomeEspecializacao)

INNER JOIN TRATAMENTO\_PADRAO ON (TRATAMENTO.numTratamento =

TRATAMENTO\_PADRAO.numTratamento)

INNER JOIN SINTOMA\_PADRAO ON (SINTOMA\_PADRAO.numPadrao =

TRATAMENTO\_PADRAO.numPadrao)

INNER JOIN SINTOMA\_PACIENTE ON (SINTOMA\_PADRAO.numPadrao =

SINTOMA\_PACIENTE.numSintoma)

INNER JOIN PADRAO ON (SINTOMA\_PADRAO.numPadrao = PADRAO.numPadrao);

1. Quais os terapeutas, em funções, que ainda não realizaram nenhuma consulta mas estão habilitados a realizar algum dos tratamentos recomendados para o padrão clínico “Vazio de Sangue”. *Nota: mostrar a informação sobre o BI, nome e telefone dos terapeutas.*

SELECT TOP 1 COUNT(PADRAO.numPadrao)as Frequência,

PADRAO.numPadrao as Referência, PADRAO.nomePadrao as Designação, PADRAO.descPadrao

FROM

PADRAO INNER JOIN SINTOMA\_PADRAO ON (PADRAO.numPadrao =

SINTOMA\_PADRAO.numPadrao)

INNER JOIN SINTOMA\_PACIENTE ON (SINTOMA\_PADRAO.numSintoma =

SINTOMA\_PACIENTE.numSintoma)

INNER JOIN PACIENTE ON (SINTOMA\_PACIENTE.BIPaciente =

PACIENTE.BI)

WHERE PACIENTE.estadoCivil = 'Casado'

GROUP BY PADRAO.numPadrao, PADRAO.nomePadrao, PADRAO.descPadrao

ORDER BY 1 DESC;

1. Qual o padrão clínico mais frequentemente diagnosticado aos pacientes cujo estado civil é “casado”. *Nota: mostrar a informação sobre a referência, a descrição e a designação.*

SELECT TOP 1 COUNT(PADRAO.numPadrao)as Frequência,

PADRAO.numPadrao as Referência, PADRAO.nomePadrao as Designação, PADRAO.descPadrao

FROM

PADRAO INNER JOIN SINTOMA\_PADRAO ON (PADRAO.numPadrao =

SINTOMA\_PADRAO.numPadrao)

INNER JOIN SINTOMA\_PACIENTE ON (SINTOMA\_PADRAO.numSintoma =

SINTOMA\_PACIENTE.numSintoma)

INNER JOIN PACIENTE ON (SINTOMA\_PACIENTE.BIPaciente =

PACIENTE.BI)

WHERE PACIENTE.estadoCivil = 'Casado'

GROUP BY PADRAO.numPadrao, PADRAO.nomePadrao, PADRAO.descPadrao

ORDER BY 1 DESC;

1. Quais os terapeutas, que também são pacientes, e que já realizaram algum tratamento para o qual estão também habilitados. *Nota: mostrar a informação sobre o BI, nome, data de nascimento, email.*

SELECT TERAPEUTA.BI as Identificação,

PESSOA.nomePessoa as Nome,

PESSOA.dataNasc as 'Data de Nascimento',

PACIENTE.email as 'Email'

FROM

TERAPEUTA INNER JOIN PESSOA ON (TERAPEUTA.BI = PESSOA.BI)

INNER JOIN PACIENTE ON (TERAPEUTA.BI = PACIENTE.BI)

INNER JOIN SESSAO ON (PACIENTE.BI = SESSAO.BIPaciente)

INNER JOIN TRATAMENTO ON (SESSAO.numTratamento =

TRATAMENTO.numTratamento)

INNER JOIN ESPECIALIZACAO ON

(ESPECIALIZACAO.nomeEspecializacao =

TRATAMENTO.nomeEspecializacao)

WHERE (ESPECIALIZACAO.nomeEspecializacao =

TRATAMENTO.nomeEspecializacao)

UNION

SELECT TERAPEUTA.BI as Identificação,

PESSOA.nomePessoa as Nome,

PESSOA.dataNasc as 'Data de Nascimento',

PACIENTE.email as 'Email'

FROM

TERAPEUTA INNER JOIN PESSOA ON (TERAPEUTA.BI = PESSOA.BI)

INNER JOIN PACIENTE ON (TERAPEUTA.BI = PACIENTE.BI)

INNER JOIN CONSULTA\_NAOPRESENCIAL ON (PACIENTE.BI =

CONSULTA\_NAOPRESENCIAL .BIPaciente)

INNER JOIN TRATAMENTO ON

(CONSULTA\_NAOPRESENCIAL.numTratamento = TRATAMENTO.numTratamento)

INNER JOIN ESPECIALIZACAO ON

(ESPECIALIZACAO.nomeEspecializacao = TRATAMENTO.nomeEspecializacao)

WHERE (ESPECIALIZACAO.nomeEspecializacao = TRATAMENTO.nomeEspecializacao);

1. Qual (is) o (s) paciente (s) que não frequenta(m) a clínica há mais tempo. *Nota: contemplar toda a informação disponível sobre o paciente.*

SELECT CONSULTA.dataConsulta as 'Última Consulta',

PESSOA.nomePessoa as Nome,

PESSOA.BI, PESSOA.NIF, PESSOA.idade,

PESSOA.dataNasc as 'Data Nascimento',PESSOA.morada,

PESSOA.telefone, PACIENTE.dataRegisto as 'Data Registo',

PACIENTE.declRespon as Declaração, PACIENTE.email, ~ PACIENTE.estadoCivil as 'Estado Civil',

PACIENTE.profissao as Profissão

FROM PESSOA INNER JOIN PACIENTE ON (PESSOA.BI = PACIENTE.BI)

INNER JOIN CONSULTA ON (PACIENTE.BI = CONSULTA.BIPaciente)

WHERE CONSULTA.dataConsulta IN

(SELECT TOP 1 MAX(CONSULTA.dataConsulta)

FROM

CONSULTA INNER JOIN PACIENTE ON (CONSULTA.BIPaciente

= PACIENTE.BI)

GROUP BY PACIENTE.BI

)

1. Para cada um dos tratamentos presenciais, qual o somatório total de horas de terapia já realizadas no centro. *Nota: ordenar por ordem decrescente do total de horas.*

SELECT SUM(DATEDIFF(MINUTE, '0:00:00', SESSAO.duracao))/60.0 as 'Total Horas', SESSAO.numTratamento as Tratamento

FROM

SESSAO

GROUP BY SESSAO.numTratamento

ORDER BY 1 DESC;

1. Quais os terapeutas que estão habilitados a realizar todos os tratamentos disponíveis no centro. *Nota: ordenar os terapeutas por ordem decrescente de idade.*

SELECT PESSOA.nomePessoa as Terapeuta, PESSOA.idade as Idade

FROM PESSOA INNER JOIN TERAPEUTA ON (PESSOA.BI = TERAPEUTA.BI)

WHERE TERAPEUTA.emFuncoes = 1 AND NOT EXISTS (

SELECT ESPECIALIZACAO.nomeEspecializacao

FROM ESPECIALIZACAO

WHERE nomeEspecializacao NOT IN (

SELECT TERAPEUTA\_ESPECIALIZACAO.nomeEspecializacao FROM TERAPEUTA\_ESPECIALIZACAO

WHERE (TERAPEUTA.BI = TERAPEUTA\_ESPECIALIZACAO.BITerapeuta)

))

ORDER BY PESSOA.idade DESC

1. Quais os pacientes a quem já foi receitada mais do que 3 vezes a fórmula de fitoterapia nº 33. *Nota: mostrar toda a informação relativa ao paciente.*

SELECT PACIENTE.\*, PESSOA.\*

FROM

PESSOA INNER JOIN PACIENTE ON (PESSOA.BI = PACIENTE.BI)

INNER JOIN CONSULTA\_NAOPRESENCIAL ON (PACIENTE.BI =

CONSULTA\_NAOPRESENCIAL.BIPaciente)

INNER JOIN NAOPRESENCIAL ON

(CONSULTA\_NAOPRESENCIAL.numTratamento = NAOPRESENCIAL.numTratamento)

WHERE (NAOPRESENCIAL.numFormula = 33)

GROUP BY PACIENTE.BI,PESSOA.nomePessoa, PESSOA.NIF,

PACIENTE.profissao,PACIENTE.estadoCivil,PESSOA.dataNasc,

PACIENTE.declRespon,PACIENTE.email,PACIENTE.dataRegisto,

PESSOA.idade,PESSOA.telefone,PESSOA.BI,PESSOA.morada

HAVING (COUNT(PACIENTE.BI) > 3);

1. Quais os pacientes que estejam inscritos na clinica e que tenham desempenhado funções como terapeutas.

SELECT PESSOA.nomePessoa,TERAPEUTA.emFuncoes

FROM

TERAPEUTA INNER JOIN PESSOA ON (TERAPEUTA.BI = PESSOA.BI)

WHERE TERAPEUTA.emFuncoes = 0 AND EXISTS (

SELECT PACIENTE.BI FROM PACIENTE

WHERE PACIENTE.BI = TERAPEUTA.BI

# Conclusão

Apresentam-se diferenças entre o modelo sugerido e o modelo adoptado para a 2ª Parte do trabalho, no entanto entendemos que é um modelo que corresponde às necessidades do cliente abrangendo os requisitos do sistema enunciados na 1ª parte. Aproveitámos, no entanto para efectuar algumas alterações de forma a responder às várias interrogações colocadas na etapa 4.

Não sendo possível carregar a base de dados com uma quantidade desejável de dados de forma a testar exaustivamente o modelo, os dados que possuímos permitiram a resposta às *queries* mencionadas.