



Universidade do Minho
Escola de Engenharia

Bases de Dados para a Bioinformática

Mestrado em Bioinformática

Resolução das Fichas de Trabalho

Docente: José Manuel Ferreira Machado

Aluno

Duarte Alves Velho A96815

2023/2024

Sumário

- **PL01** – Registo de Contactos de Profissionais;
- **PL02** – Empresa de Colheitas de Análises Clínicas;
- **PL03** – Serviços Académicos;
- **PL04** – Serviços Académicos;
- **PL05** – Registo da Realização de Consultas numa Clínica;
- **PL06** – Registo da Realização de Consultas numa Clínica;
- **PL07** – Registo da Realização de Consultas numa Clínica;
- **PL08** – Registo da Realização de Consultas numa Clínica.

➔ PL01 – “Registo de Contactos de Profissionais”

1 Registo de Contactos de Profissionais

Todas as empresas da área da saúde, de uma maneira ou de outra, têm de guardar a informação relativa aos contactos pessoais dos profissionais que nela trabalham, incluindo morada, telefone, telemóvel e *e-mail*, entre outros. A forma como é mantida essa informação depende muito das nossas pretensões de utilização ou das necessidades associadas.

Os “livros” de contactos podem apresentar estruturas e formatos muito diversos. Os seus utilizadores costumam personalizá-los frequentemente e neles guardam informação muito díspar e variada. Veja-se, a título de exemplo, a Figura 1.



Figura 1: Exemplo de um livro de contactos.

Com base no caso apresentado, pretende-se que:

1. Identifique e caracterize potenciais entidades envolvidas num livro de contactos idealizado por si, bem como os vários atributos que as constituem.

1. **Profissional de Saúde** (Nome do Profissional (atributo chave primária), Id profissional, Cargo na empresa, Morada(atributo composto: número, rua, código-postal, localidade), Email, Telemóvel (atributo multivalor), especialidade, id_empresa (chave estrangeira, referenciando a entidade Empresa)

2. **Empresa na área da saúde** (Nome da empresa (atributo chave primária), Tipo de serviço, Endereço, Email, Telefone da empresa, Website)

3. **Contacto** (ID de Colaborador (atributo chave primária), Telemóvel, Email)

4. **Livro de Contactos** (Entidade fraca, pois depende das outras para a sua existência)

2. Identifique e caracterize os diversos relacionamentos que possam existir entre as entidades estabelecidas na alínea anterior.

1. **Profissional de Saúde e Empresa da área da Saúde:** Vários profissionais de saúde trabalham numa empresa da área da saúde. Relacionamento de 1 para N (1:N).

2. **Profissional de Saúde e Livro de Contactos:** Um livro de contactos para vários Profissionais de Saúde. Relacionamento de 1 para N (1:N).

3. **Empresa da área da Saúde e Livro de Contactos:** Uma empresa da área da saúde tem vários livros de contactos. Relacionamento de 1 para N (1:N).

4. **Livro de Contactos e Contacto:** Um livro de contactos possui vários contactos. Relacionamento de 1 para N (1:N).

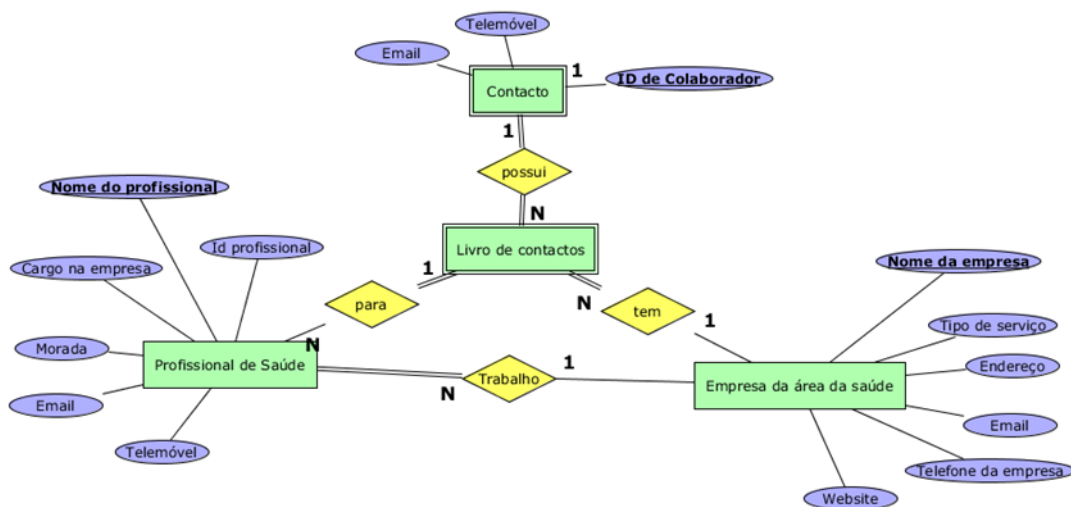


Figura 1. Diagrama E-R, utilizando a Notação de Chen, capaz de acolher os diversos objetos de dados identificados e caracterizados nas alíneas anteriores da questão nº1.

➔ PL02 – “Empresa de Colheitas de Análises Clínicas”

1 Empresa de Colheitas de Análises Clínicas

Uma empresa de colheitas de análises clínicas pretende desenvolver uma pequena base de dados para registar os equipamentos que os seus funcionários utilizam nas diversas tarefas de colheita e de armazenamento que realizam.

A empresa possui várias viaturas devidamente equipadas que se deslocam aos diferentes locais onde devem ser realizadas as colheitas. Os membros da direção da empresa acreditam que, dessa maneira, conseguirão fazer uma melhor gestão dos equipamentos da empresa, bem como reduzir os seus custos de manutenção.

Após uma reunião de trabalho com os membros da direção, com os chefes das equipas de colheita e, obviamente, com todos os funcionários integrados em equipas de colheita, o arquiteto da futura base de dados elaborou uma pequena lista de requisitos para a sua construção. Nessa lista figurava o seguinte: na definição de cada serviço de colheita é necessário caracterizar a colheita que será realizada (número, designação, data de início, data de fim, nome do cliente e local de realização), indicar quais os funcionários (número, nome e categoria) que nela vão participar, bem como o equipamento que vai ser utilizado no trabalho e o número de horas de utilização previsto para cada equipamento nesse trabalho. Além disso, será igualmente necessário registar o funcionário que é o responsável da equipa de colheita na realização de cada um dos diferentes serviços de colheita.

Com base no caso apresentado, pretende-se que:

1. Identifique e caracterize potenciais entidades envolvidas no estudo de caso apresentado, bem como os vários atributos que as constituem.

1. **Equipamento** (Nº de Horas Previsto (atributo chave primária), nome, número)
2. **Serviço de Colheita** (Número (atributo chave primária), Matrícula da viatura, Data de início, Designação, Data de fim, local da realização, Nome do cliente)
3. **Equipa** (ID da Equipa (atributo chave primária), funcionário_dirigente (chave estrangeira, referenciando a entidade “Funcionário”), número_viatura (chave estrangeira, referenciando a entidade “Viatura”)
4. **Responsável** (Nome (atributo chave primária), Número, Categoria)
5. **Funcionários** (ID Responsável (atributo chave primária), Nome Responsável)

2. Identifique e caracterize os diversos relacionamentos que possam existir entre as entidades estabelecidas na alínea anterior.

1. **Equipamento e Serviço de Colheita:** Muitos Equipamentos, requerem muitos Serviços de Colheita. Relacionamento de M para N (M:N).
2. **Equipas e Serviço de Colheita:** Várias Equipas participam num Serviço de Colheita. Relacionamento de 1 para N (1:N).
3. **Funcionários e Equipas:** Vários funcionários integram várias Equipas. Relacionamento de M para N (M:N).

4. **Equipa e Responsável:** Uma equipa é liderada por vários responsáveis. Relacionamento de 1 para N (1:N).

5. **Responsável e Funcionários:** Um Responsável chefia vários funcionários. Relacionamento de 1 para N (1:N).

6. **Responsável e Funcionário:** Um Responsável integra vários Funcionários. Relacionamento de 1 para N (1:N).

3. Desenhe um diagrama entidade relacionamento (ER) na notação Chen capaz de acolher os diversos objetos de dados identificados e caracterizados nas alíneas anteriores.

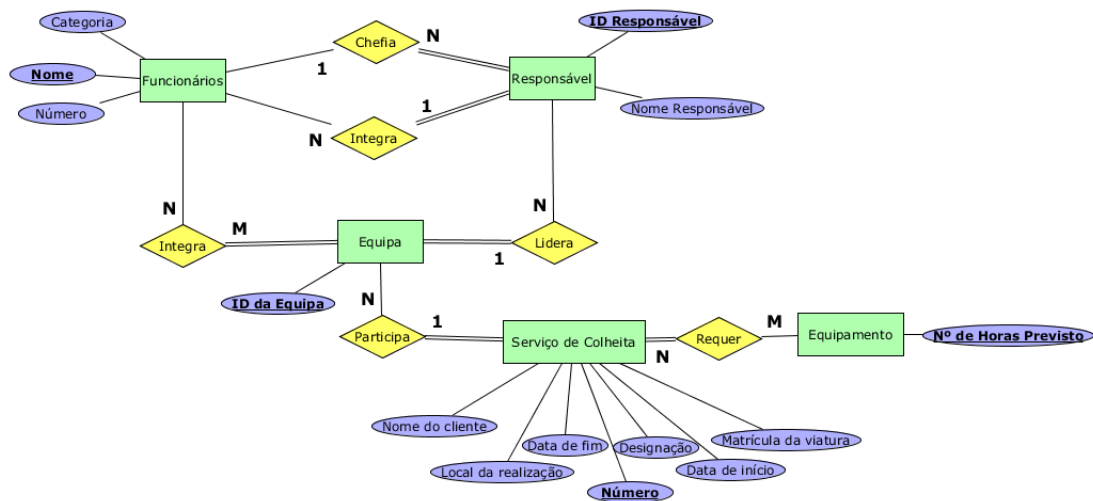


Figura 2. Diagrama E-R, utilizando a Notação de Chen, capaz de acolher os diversos objetos de dados identificados e caracterizados nas alíneas anteriores da questão nº2.

➔ PL03 – “Serviços Académicos”

1 Serviços Académicos

A inscrição e o registo das classificações nas unidades curriculares (UCs) que os alunos vão fazendo ao longo dos anos, há muito que representam um problema relevante para os serviços académicos. Os seus responsáveis, depois de várias reuniões de trabalho, decidiram finalmente avançar para a implementação de um novo sistema para a gestão dos processos referidos.

Nesse sentido, promoveram várias reuniões com os serviços de informação para discutirem o plano de trabalho para os futuros serviços de bases de dados, bem como a estrutura da futura base de dados em si. No final, foi apresentado um documento com os requisitos que acharam mais importantes e que gostariam de ver refletidos na futura base de dados. Os pontos essenciais desse documento são os seguintes:

- Na altura da sua inscrição num dado curso, um aluno tem que indicar o seu nome, o seu número de contribuinte, o seu número de cartão de cidadão, a sua morada, um ou mais números de contacto telefónico, os nomes dos seus pais e do seu encarregado de educação, a sua data de nascimento e o curso que vai frequentar. No processo de inscrição, um identificador único é atribuído ao aluno.
- Os cursos estarão catalogados no sistema de acordo com o seu identificador único, a sua designação, o seu ciclo de estudos (1.º, 2.º ou 3.º), o grau que confere e o número de alunos inscritos. Além disso, incluem também a identificação do seu diretor que é, obrigatoriamente, um dos docentes da instituição.
- No entanto, um aluno pode obviamente estar inscrito em vários cursos. Quando inscrito num curso, e para garantir a sua realização, um aluno tem que realizar todas as unidades curriculares que pertencem ao curso. O registo de dados do aluno associado ao curso irá conter o número de unidades curriculares realizadas, o correspondente valor em ECTS, bem como a sua média atual. As datas de início e de fim associadas ao(s) curso(s) que o aluno frequenta devem, igualmente, serem registadas.
- Um aluno não se pode inscrever de uma só vez em todas as unidades curriculares de um dado curso, isto é, ele tem de respeitar a regulamentação vigente sobre a frequência e a realização de unidades curriculares. Por outro lado, a sua data de inscrição em cada unidade curricular tem de estar registada no sistema.
- No final de cada semestre, os docentes das várias unidades curriculares comunicam as notas finais que os alunos obtiveram, sendo estas posteriormente lançadas no sistema, unidade curricular a unidade curricular, aluno a aluno. A cada classificação (nota final) está sempre associada a data de realização da unidade curricular pelo aluno.
- Como já foi referido anteriormente, as unidades curriculares estão integradas num dado curso. Uma unidade curricular apenas pode pertencer a um único curso, estando caracterizada pelo seu identificador único e a sua designação. Também é definida pela sua escolaridade, bem como o ano letivo e o semestre de leção.
- Na futura base de dados deverá, igualmente, figurar a informação relativa ao responsável das unidades curriculares (um docente), assim como o tipo e o número de horas semanais que cada docente associado a unidade curricular leciona. Os docentes das unidades curriculares pertencem a um único departamento dentro da instituição e são classificados através da sua categoria dentro da mesma, bem como são caracterizados pelo seu identificador único e o seu nome. Os departamentos são unicamente caracterizados pelo seu identificador único e a sua designação.

(...)

Assim, com base na resolução da Ficha PL03, pretende-se que:

1. Utilizando o *MySQL Workbench*, faça a geração do respetivo esquema físico para a base de dados em questão.

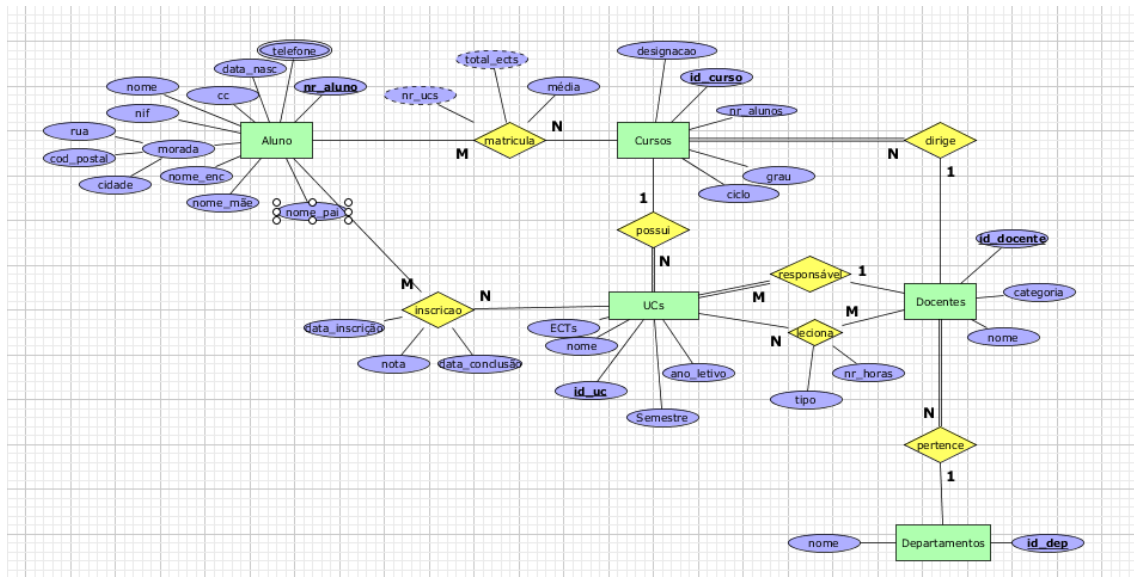


Figura 3. Diagrama E-R, utilizando a Notação de Chen, capaz de acolher os diversos objetos de dados identificados e caracterizados no enunciado da questão nº3.

2. Utilizando o *MySQL Workbench*, converta o esquema conceptual produzido na alínea anterior para o seu correspondente esquema lógico.

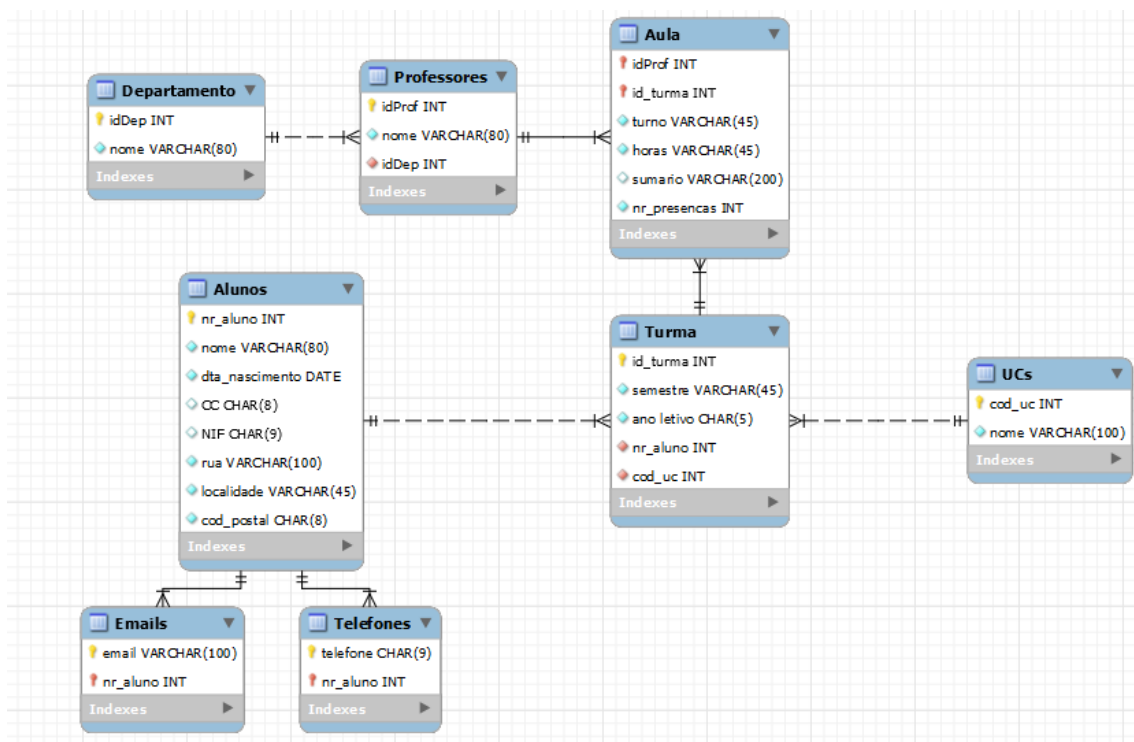


Figura 4. Esquema lógico correspondente ao esquema conceptual produzido na alínea anterior.

➔ PL04 – “Serviços Académicos”

1 Serviços Académicos

A inscrição e o registo das classificações nas unidades curriculares (UCs) que os alunos vão fazendo ao longo dos anos, há muito que representam um problema relevante para os serviços académicos. Os seus responsáveis, depois de várias reuniões de trabalho, decidiram finalmente avançar para a implementação de um novo sistema para a gestão dos processos referidos.

Nesse sentido, promoveram várias reuniões com os serviços de informação para discutirem o plano de trabalho para os futuros serviços de bases de dados, bem como a estrutura da futura base de dados em si. No final, foi apresentado um documento com os requisitos que acharam mais importantes e que gostariam de ver refletidos na futura base de dados. Os pontos essenciais desse documento são os seguintes:

- Na altura da sua inscrição num dado curso, um aluno tem que indicar o seu nome, o seu número de contribuinte, o seu número de cartão de cidadão, a sua morada, um ou mais números de contacto telefónico, os nomes dos seus pais e do seu encarregado de educação, a sua data de nascimento e o curso que vai frequentar. No processo de inscrição, um identificador único é atribuído ao aluno.
- Os cursos estarão catalogados no sistema de acordo com o seu identificador único, a sua designação, o seu ciclo de estudos (1.º, 2.º ou 3.º), o grau que confere e o número de alunos inscritos. Além disso, incluem também a identificação do seu diretor que é, obrigatoriamente, um dos docentes da instituição.
- No entanto, um aluno pode obviamente estar inscrito em vários cursos. Quando inscrito num curso, e para garantir a sua realização, um aluno tem que realizar todas as unidades curriculares que pertencem ao curso. O registo de dados do aluno associado ao curso irá conter o número de unidades curriculares realizadas, o correspondente valor em ECTS, bem como a sua média atual. As datas de início e de fim associadas ao(s) curso(s) que o aluno frequenta devem, igualmente, serem registadas.
- Um aluno não se pode inscrever de uma só vez em todas as unidades curriculares de um dado curso, isto é, ele tem de respeitar a regulamentação vigente sobre a frequência e a realização de unidades curriculares. Por outro lado, a sua data de inscrição em cada unidade curricular tem de estar registada no sistema.
- No final de cada semestre, os docentes das várias unidades curriculares comunicam as notas finais que os alunos obtiveram, sendo estas posteriormente lançadas no sistema, unidade curricular a unidade curricular, aluno a aluno. A cada classificação (nota final) está sempre associada a data de realização da unidade curricular pelo aluno.
- Como já foi referido anteriormente, as unidades curriculares estão integradas num dado curso. Uma unidade curricular apenas pode pertencer a um único curso, estando caracterizada pelo seu identificador único e a sua designação. Também é definida pela sua escolaridade, bem como o ano letivo e o semestre de leção.
- Na futura base de dados deverá, igualmente, figurar a informação relativa ao responsável das unidades curriculares (um docente), assim como o tipo e o número de horas semanais que cada docente associado a unidade curricular leciona. Os docentes das unidades curriculares pertencem a um único departamento dentro da instituição e são classificados através da sua categoria dentro da mesma, bem como são caracterizados pelo seu identificador único e o seu nome. Os departamentos são unicamente caracterizados pelo seu identificador único e a sua designação.

(...)

Assim, com base na resolução da Ficha PL03, pretende-se que:

1. Utilizando o *MySQL Workbench*, faça a geração do respetivo esquema físico para a base de dados em questão.

NOTA: Colei o seguinte código no documento word, porém também envio em anexo um script da pergunta. (Ficheiro – “Pergunta 1 - Ficha 4.sql”)

-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

```
SET @OLD_SQL_MODE=@@SQL_MODE,  
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR  
_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
-----
```

```
-- Schema aula
```

```
-----
```

```
-----
```

```
-- Schema aula
```

```
-----
```

```
CREATE SCHEMA IF NOT EXISTS `aula` DEFAULT CHARACTER SET utf8 ;  
USE `aula` ;
```

```
-----
```

```
-- Table `aula`.`Alunos`
```

```
-----
```

```
CREATE TABLE IF NOT EXISTS `aula`.`Alunos` (  
  `nr_aluno` INT NOT NULL,  
  `nome` VARCHAR(80) NOT NULL,  
  `dta_nascimento` DATE NOT NULL,  
  `CC` CHAR(8) NULL,  
  `NIF` CHAR(9) NULL,  
  `rua` VARCHAR(100) NOT NULL,  
  `localidade` VARCHAR(45) NOT NULL,  
  `cod_postal` CHAR(8) NOT NULL,  
  PRIMARY KEY (`nr_aluno`),  
  UNIQUE INDEX `CC_UNIQUE` (`CC` ASC) VISIBLE,  
  UNIQUE INDEX `NIF_UNIQUE` (`NIF` ASC) VISIBLE)  
ENGINE = InnoDB;
```

```
-----
```

```
-- Table `aula`.`Telefones`
```

```

-----

CREATE TABLE IF NOT EXISTS `aula`.`Telefones` (
  `telefone` CHAR(9) NOT NULL,
  `nr_aluno` INT NOT NULL,
  PRIMARY KEY (`telefone`, `nr_aluno`),
  INDEX `fk_Telefones_Alunos_idx` (`nr_aluno` ASC) VISIBLE,
  CONSTRAINT `fk_Telefones_Alunos`
    FOREIGN KEY (`nr_aluno`)
      REFERENCES `aula`.`Alunos` (`nr_aluno`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----

-- Table `aula`.`Emails`
-----

CREATE TABLE IF NOT EXISTS `aula`.`Emails` (
  `email` VARCHAR(100) NOT NULL,
  `nr_aluno` INT NOT NULL,
  PRIMARY KEY (`email`, `nr_aluno`),
  INDEX `fk_Emails_Alunos1_idx` (`nr_aluno` ASC) VISIBLE,
  CONSTRAINT `fk_Emails_Alunos1`
    FOREIGN KEY (`nr_aluno`)
      REFERENCES `aula`.`Alunos` (`nr_aluno`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----

-- Table `aula`.`UCs`
-----

```

```
CREATE TABLE IF NOT EXISTS `aula`.`UCs` (
  `cod_uc` INT NOT NULL,
  `nome` VARCHAR(100) NOT NULL,
  PRIMARY KEY (`cod_uc`))
ENGINE = InnoDB;
```

```
-----
-- Table `aula`.`Departamento`
-----
```

```
CREATE TABLE IF NOT EXISTS `aula`.`Departamento` (
  `idDep` INT NOT NULL,
  `nome` VARCHAR(80) NOT NULL,
  PRIMARY KEY (`idDep`))
ENGINE = InnoDB;
```

```
-----
-- Table `aula`.`Professores`
-----
```

```
CREATE TABLE IF NOT EXISTS `aula`.`Professores` (
  `idProf` INT NOT NULL,
  `nome` VARCHAR(80) NOT NULL,
  `idDep` INT NOT NULL,
  PRIMARY KEY (`idProf`),
  INDEX `fk_Professores_Departamento1_idx` (`idDep` ASC) VISIBLE,
  CONSTRAINT `fk_Professores_Departamento1`
    FOREIGN KEY (`idDep`)
      REFERENCES `aula`.`Departamento` (`idDep`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

-- Table `aula`.`Turma`

```
CREATE TABLE IF NOT EXISTS `aula`.`Turma` (  
  `id_turma` INT NOT NULL,  
  `semestre` VARCHAR(45) NOT NULL,  
  `ano letivo` CHAR(5) NOT NULL,  
  `nr_aluno` INT NOT NULL,  
  `cod_uc` INT NOT NULL,  
  PRIMARY KEY (`id_turma`),  
  INDEX `fk_Turma_Alunos1_idx` (`nr_aluno` ASC) VISIBLE,  
  INDEX `fk_Turma_UCs1_idx` (`cod_uc` ASC) VISIBLE,  
  CONSTRAINT `fk_Turma_Alunos1`  
    FOREIGN KEY (`nr_aluno`)  
      REFERENCES `aula`.`Alunos` (`nr_aluno`)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Turma_UCs1`  
    FOREIGN KEY (`cod_uc`)  
      REFERENCES `aula`.`UCs` (`cod_uc`)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `aula`.`Aula`

```
CREATE TABLE IF NOT EXISTS `aula`.`Aula` (  
  `idProf` INT NOT NULL,  
  `id_turma` INT NOT NULL,  
  `turno` VARCHAR(45) NOT NULL,
```

```

`horas` VARCHAR(45) NOT NULL,
`sumario` VARCHAR(200) NULL,
`nr_presencas` INT NOT NULL DEFAULT 0,
PRIMARY KEY (`idProf`, `id_turma`),
INDEX `fk_Professores_has_Turma_Turma1_idx` (`id_turma` ASC) VISIBLE,
INDEX `fk_Professores_has_Turma_Professores1_idx` (`idProf` ASC) VISIBLE,
CONSTRAINT `fk_Professores_has_Turma_Professores1`
    FOREIGN KEY (`idProf`)
    REFERENCES `aula`.`Professores` (`idProf`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_Professores_has_Turma_Turma1`
    FOREIGN KEY (`id_turma`)
    REFERENCES `aula`.`Turma` (`id_turma`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

2. Desenvolva as operações necessárias para fazer o povoamento das tabelas da base de dados.

-- Inserir Alunos

```

INSERT INTO Alunos (nr_aluno, nome, dta_nascimento, CC, NIF, rua, localidade, cod_postal)
VALUES
(1, 'João Silva', '1990-01-01', '12345678', '987654321', 'Rua A', 'Cidade X', '1234-567'),
(2, 'Maria Santos', '1992-05-15', '87654321', '123456789', 'Rua B', 'Cidade Y', '5678-901');

```

-- Inserir Telefones

```

INSERT INTO Telefones (telefone, nr_aluno)
VALUES
('912345678', 1),

```

```
('987654321', 2);
```

```
-- Inserir Emails
```

```
INSERT INTO Emails (email, nr_aluno)
```

```
VALUES
```

```
('joao@email.com', 1),
```

```
('maria@email.com', 2);
```

```
-- Inserir UCs
```

```
INSERT INTO UCs (cod_uc, nome)
```

```
VALUES
```

```
(1, 'Matemática'),
```

```
(2, 'Física');
```

```
-- Inserir Departamento
```

```
INSERT INTO Departamento (idDep, nome)
```

```
VALUES
```

```
(1, 'Departamento de Ciências Naturais'),
```

```
(2, 'Departamento de Matemática');
```

```
-- Inserir Professores
```

```
INSERT INTO Professores (idProf, nome, idDep)
```

```
VALUES
```

```
(1, 'Prof. Silva', 1),
```

```
(2, 'Prof. Santos', 2);
```

```
-- Inserir Turma
```

```
INSERT INTO Turma (id_turma, semestre, `ano letivo`, nr_aluno, cod_uc)
```

```
VALUES
```

```
(1, '1º Semestre', '2023', 1, 1),
```

```
(2, '2º Semestre', '2023', 2, 2);
```

```
-- Inserir Aula
```

```
INSERT INTO Aula (idProf, id_turma, turno, horas, sumario, nr_presencas)
```

```
VALUES
```

```
(1, 1, 'Manhã', '10:00-12:00', 'Introdução à Álgebra', 20),
```

```
(2, 2, 'Tarde', '14:00-16:00', 'Mecânica Quântica', 18);
```


➔ PL05 – “Registo da Realização de Consultas numa Clínica”

1 Registo da Realização de Consultas numa Clínica

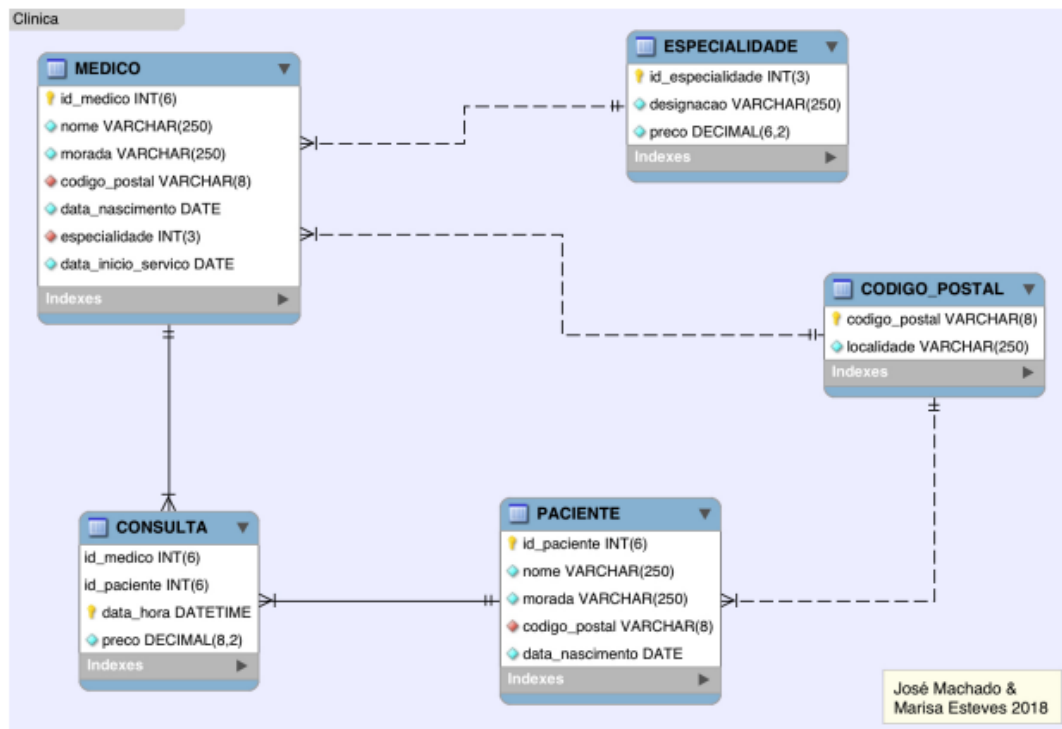


Figura 1: Esquema lógico da base de dados de uma determinada clínica.

Na Figura 1 é apresentado o esquema lógico, nomeadamente modelo relacional, da base de dados de uma determinada clínica.

A base de dados que o esquema representa está integrada no sistema operacional principal da clínica e suporta todos os processos relacionados com o registo e a faturação das consultas realizadas. É de referir que o valor pago pelo paciente em cada consulta pode ser superior ao valor de referência para a especialidade do médico (preço), uma vez que podem ser pagos procedimentos efetuados durante a realização da consulta.

Com base no esquema lógico apresentado, pretende-se que:

1. Elabore um pequeno caderno de requisitos que possa justificar o esquema lógico apresentado.

Com objetivo de fazer uma melhor gestão das consultas para os pacientes, uma determinada clínica quer desenvolver uma base de dados para conseguir registar as consultas realizadas pelos médicos das diferentes especialidades. Para isso, é necessário fazer um caderno de requisitos para a sua construção. Estes requisitos são: o Medico, Especialidade, Consulta, Paciente e Codigo_Postal.

Entidades:

- **Medico** (id_medico - chave primária, nome, morada, data_nascimento, codigo_postal - Chave Estrangeira, referenciando a entidade "CODIGO_POSTAL", data_nascimento, especialidade - Chave Estrangeira, referenciando a entidade "ESPECIALIDADE", data_inicio_serviço)
- **Especialidade** (id_especialidade - chave primária, designacao, preco)
- **Consulta** (entidade-relacionamento) (id_medico - chave primária, id_paciente -chave primária, data_hora - chave primária, preco)
- **Paciente** (id_paciente - chave primária, nome, morada, codigo_postal - chave estrangeira, referenciando a entidade "CODIGO_POSTAL", data_nascimento)
- **Codigo_Postal** (codigo_postal - chave primária, localidade)

Relações:

- **Médico e Especialidade:** Um médico pertence a uma especialidade e uma especialidade tem associados um ou mais médicos. muitos para um. Portanto, é um relacionamento de N para 1 (N:1). Neste tipo de relacionamentos, a entidade referente a (N) tem como chave estrangeira a chave primária da entidade referente a (1). Como tal, a chave "especialidade" é chave estrangeira da entidade "Médico".
- **Médico e Consulta:** um médico pode realizar várias consultas, mas uma consulta só pode estar associada a um único médico. Portanto, é um relacionamento de N para 1 (N:1). Neste tipo de relacionamentos, a entidade referente a (N) tem como chave estrangeira a chave primária da entidade referente a (1). Como tal, a chave "id_medico" é chave estrangeira da entidade "Consulta".
- **Médico e Cód.Postal:** Um médico pode ter associado um único código postal, mas um código postal pode estar associado a vários médicos. Portanto, é um relacionamento de N para 1 (N:1). Neste tipo de relacionamentos, a entidade referente a (N) tem como chave estrangeira a chave primária da entidade referente a (1). Como tal, a chave "id_medico" é chave estrangeira da entidade "Consulta".
- **Consulta e Paciente:** Um paciente pode estar associado a uma ou mais consultas, mas uma consulta só pode estar associada a um único paciente. Portanto, é um relacionamento de N para 1 (N:1). Neste tipo de relacionamentos, a entidade referente a (N) tem como chave estrangeira a chave primária da entidade referente a (1). Como tal, a chave "id_paciente" é chave estrangeira da entidade "Consulta".

• **Paciente e Cód.Postal:** Um paciente pode ter associado um único código postal, mas um código postal pode estar associado a vários pacientes. Portanto, é um relacionamento de N para 1 (N:1). Neste tipo de relacionamentos, a entidade referente a (N) tem como chave estrangeira a chave primária da entidade referente a (1). Como tal, a chave “código_postal” é chave estrangeira da entidade “Paciente”.

2. Desenhe um possível esquema conceptual equivalente ao esquema lógico apresentado na Figura 1.

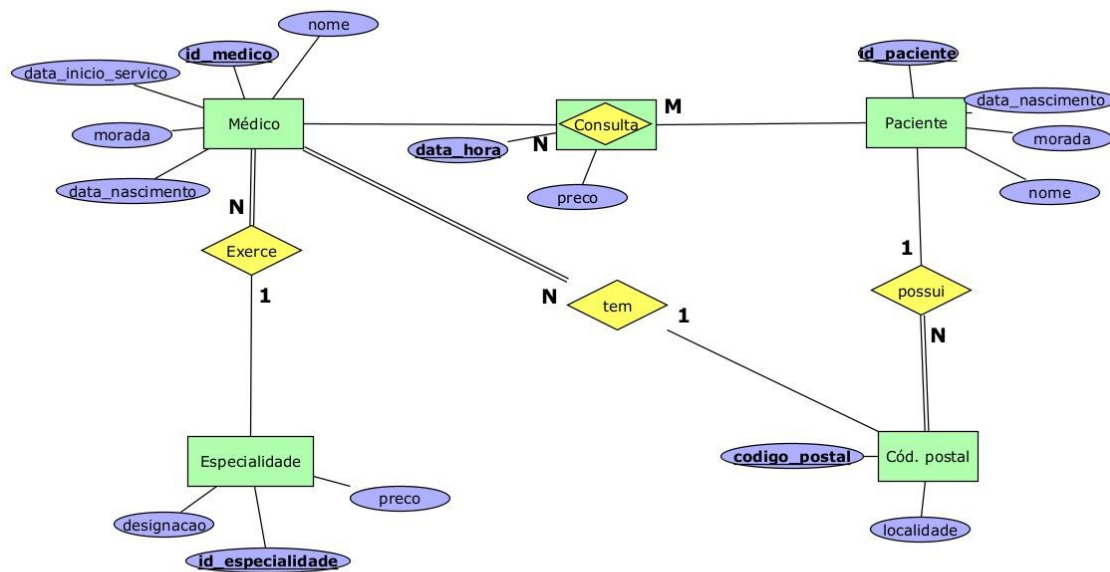


Figura 5. Diagrama E-R, utilizando a Notação de Chen, equivalente ao esquema lógico apresentado no enunciado.

3. Utilizando álgebra relacional, desenvolva as expressões necessárias, e respetivas árvores de resolução, que possam responder às seguintes questões:

- Qual é o nome dos médicos com mais de 10 anos de serviço?
- Qual é o nome de todos os médicos e a respetiva especialidade que cada um exerce?
- Qual é o nome e a morada dos pacientes residentes na localidade de Braga?
- Qual é o nome dos médicos da especialidade de Oftalmologia?
- Qual é o nome e a idade dos médicos com mais de 40 anos de idade da especialidade de Clínica Geral?
- Qual é o nome dos médicos da especialidade de Oftalmologia que consultaram pacientes da localidade de Braga?
- Qual é o nome e os anos de serviço dos médicos com mais de 50 anos de idade que deram consultas a partir das 12h a pacientes com menos de 20 anos de idade?
- Qual é o nome e a idade dos pacientes com mais de 10 anos de idade que nunca foram consultados na especialidade de Oftalmologia?

Questão 3 → Ficha 5

a) $\sigma_{\text{idade}}(ES) > 10$ Médicos

b) $\pi_{M,E} (\text{medico} \bowtie_E \text{especialidade})$

c) $\pi_{P,CP} (\text{paciente} \bowtie_{CP} GL = 'Brasyl' \text{Codigo - Postal})$

d) $\pi_{M,E} (\text{medico} \bowtie \text{especialidade } E \text{ } \& E.\text{nome} = 'Oftalmologia' \text{especialidade})$

e) $\sigma_{M.\text{idade} > 40} \pi_{M,E} (\text{medico } M \bowtie \text{especialidade } E \text{ } \& E.\text{nome} = 'Clínica Geral')$

f) $\pi_{M,E} \text{medico} \bowtie \text{especialidade } \& E.\text{nome} = 'Oftalmologia' (\text{especialidade } E)$

$\bowtie \text{Medico (Consulta} \bowtie \text{paciente} \bowtie \text{Codigo postal } \& \text{localidade} = 'Brasyl')$

g) $\sigma_{\text{idade} > 50 (\text{medico})} \bowtie_{id_medico} (\sigma_{\text{hora} > '12:00' (\text{Consulta})} \bowtie_{id_paciente})$

$\sigma_{\text{idade} < 20 (\text{paciente})}$

h) $\sigma_{\text{idade} (\text{data_medicatos}) > 10 (\text{paciente})} \setminus (\sigma_{\text{idade} (\text{data_medicatos})}$

$> 10 (\text{paciente}) \bowtie_p (\text{Consulta} \bowtie_p \text{medico} \bowtie_E$

$\& \text{designacao} = 'Oftalmologia' (\text{especialidade}))$

4. Utilizando o *MySQL Workbench*, e o ficheiro "Ficha5_ModeloLogico.mwb" enviado juntamente com esta ficha prática laboratorial, faça a geração do respetivo esquema físico para a base de dados em questão.

NOTA: Colei o seguinte código no documento word, porém também envio em anexo um script da pergunta. (Ficheiro – "Pergunta 1 - Ficha 4.sql")

-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- Schema Clinica

-- Schema Clinica

CREATE SCHEMA IF NOT EXISTS `Clinica` DEFAULT CHARACTER SET utf8 ;

USE `Clinica` ;

-- Table `Clinica`.`ESPECIALIDADE`

CREATE TABLE IF NOT EXISTS `Clinica`.`ESPECIALIDADE` (

 `id_especialidade` INT(3) NOT NULL,

 `designacao` VARCHAR(250) NOT NULL,

 `preco` DECIMAL(6,2) NOT NULL,

 PRIMARY KEY (`id_especialidade`))

ENGINE = InnoDB;

-- Table `Clinica`.`CODIGO_POSTAL`

```

CREATE TABLE IF NOT EXISTS `Clinica`.`CODIGO_POSTAL` (
  `codigo_postal` VARCHAR(8) NOT NULL,
  `localidade` VARCHAR(250) NOT NULL,
  PRIMARY KEY (`codigo_postal`))
ENGINE = InnoDB;

-----

-- Table `Clinica`.`MEDICO`
-----

CREATE TABLE IF NOT EXISTS `Clinica`.`MEDICO` (
  `id_medico` INT(6) NOT NULL,
  `nome` VARCHAR(250) NOT NULL,
  `morada` VARCHAR(250) NOT NULL,
  `codigo_postal` VARCHAR(8) NOT NULL,
  `data_nascimento` DATE NOT NULL,
  `especialidade` INT(3) NOT NULL,
  `data_inicio_servico` DATE NOT NULL,
  INDEX `fk_med_idx` (`especialidade` ASC) VISIBLE,
  INDEX `fk_cp_idx` (`codigo_postal` ASC) VISIBLE,
  PRIMARY KEY (`id_medico`),
  CONSTRAINT `fk_especialidade1`
    FOREIGN KEY (`especialidade`)
      REFERENCES `Clinica`.`ESPECIALIDADE` (`id_especialidade`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_codigo_postal1`
    FOREIGN KEY (`codigo_postal`)
      REFERENCES `Clinica`.`CODIGO_POSTAL` (`codigo_postal`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----

```

```

-- Table `Clinica`.`PACIENTE`
-----

CREATE TABLE IF NOT EXISTS `Clinica`.`PACIENTE` (
  `id_paciente` INT(6) NOT NULL,
  `nome` VARCHAR(250) NOT NULL,
  `morada` VARCHAR(250) NOT NULL,
  `codigo_postal` VARCHAR(8) NOT NULL,
  `data_nascimento` DATE NOT NULL,
  INDEX `fk_paci_idx` (`codigo_postal` ASC) VISIBLE,
  PRIMARY KEY (`id_paciente`),
  CONSTRAINT `fk_codigo_postal2`
    FOREIGN KEY (`codigo_postal`)
      REFERENCES `Clinica`.`CODIGO_POSTAL` (`codigo_postal`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- Table `Clinica`.`CONSULTA`
-----

CREATE TABLE IF NOT EXISTS `Clinica`.`CONSULTA` (
  `id_medico` INT(6) NOT NULL,
  `id_paciente` INT(6) NOT NULL,
  `data_hora` DATETIME NOT NULL,
  `preco` DECIMAL(8,2) NOT NULL,
  PRIMARY KEY (`id_medico`, `id_paciente`, `data_hora`),
  INDEX `FK_cons02_idx` (`id_paciente` ASC) VISIBLE,
  CONSTRAINT `fk_medico1`
    FOREIGN KEY (`id_medico`)
      REFERENCES `Clinica`.`MEDICO` (`id_medico`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_paciente1`
    FOREIGN KEY (`id_paciente`)

```

```
REFERENCES `Clinica`.`PACIENTE` (`id_paciente`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```


➔ PL06 – “Registo da Realização de Consultas numa Clínica”

1 Registo da Realização de Consultas numa Clínica

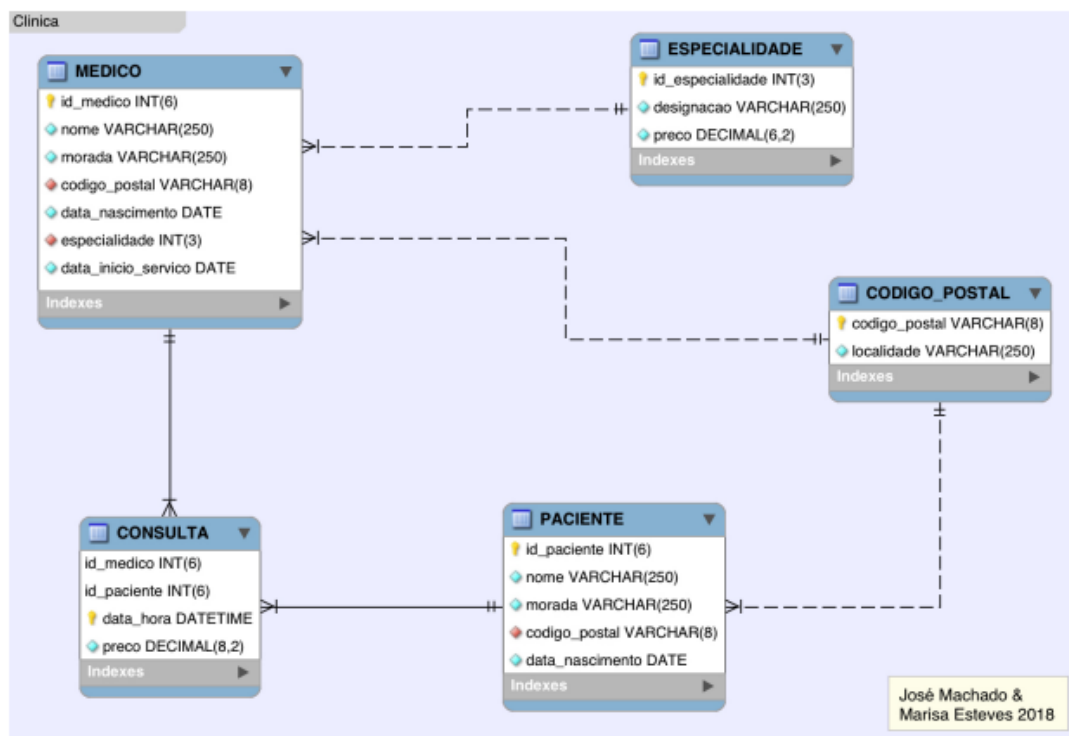


Figura 1: Esquema lógico da base de dados de uma determinada clínica.

Na Figura 1 é apresentado o esquema lógico, nomeadamente modelo relacional, da base de dados de uma determinada clínica.

A base de dados que o esquema representa está integrada no sistema operacional principal da clínica e suporta todos os processos relacionados com o registo e a faturação das consultas realizadas. É de referir que o valor pago pelo paciente em cada consulta pode ser superior ao valor de referência para a especialidade do médico (preço), uma vez que podem ser pagos procedimentos efetuados durante a realização da consulta.

Na folha de exercícios anterior utilizou o *MySQL Workbench* com o ficheiro “Ficha5_ModeloLogico.mwb”, disponibilizado juntamente com a 5.ª ficha prática laboratorial, de modo a gerar o respetivo esquema físico para a base de dados em questão.

Com base no esquema físico gerado, pretende-se que:

1. Povoie a base de dados criada, utilizando o ficheiro “Ficha6_PovoarTabelas.sql” disponibilizado juntamente com esta ficha prática laboratorial. O ficheiro disponibiliza igualmente a função “idade” que será muito útil para responder a alguns dos pontos da questão 2.

Inserção de dados para povoar as tabelas da base de dados:

-- Inserir dados na tabela CODIGO_POSTAL

```
INSERT INTO Clinica.CODIGO_POSTAL (codigo_postal, localidade) VALUES ('4700-006', 'BRAGA');
```

```
INSERT INTO Clinica.CODIGO_POSTAL (codigo_postal, localidade) VALUES ('4750-003', 'BARCELOS');
```

-- Inserir dados na tabela ESPECIALIDADE

```
INSERT INTO Clinica.ESPECIALIDADE (id_especialidade, designacao, preco)
```

```
VALUES(5, 'Dermatologia', 60);
```

```
INSERT INTO Clinica.ESPECIALIDADE (id_especialidade, designacao, preco)
```

```
VALUES(6, 'Pediatria', 40);
```

-- Inserir dados na tabela MEDICO

```
INSERT INTO Clinica.MEDICO (id_medico, nome, morada, codigo_postal,
```

```
data_nascimento, especialidade, data_inicio_servico) VALUES (623457, 'Ana
```

```
Rodrigues', 'Rua Nova n.º 456, 2.º ESQ', '4700-002', '1985-06-15', 4, '2010-03-10');
```

-- Inserir dados na tabela PACIENTE

```
INSERT INTO Clinica.PACIENTE (id_paciente, nome, morada, codigo_postal,
```

```
data_nascimento) VALUES (1023456, 'Rui Oliveira', 'Avenida Principal n.º 789, 3.º
```

```
DTO', '4700-001', '1995-12-07');
```

-- Inserir dados na tabela CONSULTA

```
INSERT INTO Clinica.CONSULTA (id_medico, id_paciente, data_hora, preco) VALUES
```

```
(623457, 1023456, '2016-02-14 11:45', 180);
```

2. Utilizando SQL, desenvolva os comandos necessários para responder às seguintes questões:

(a) Qual é o nome dos médicos com mais de 10 anos de serviço?

(b) Qual é o nome de todos os médicos e a respetiva especialidade que cada um exerce?

(c) Qual é o nome e a morada dos pacientes residentes na localidade de Braga?

(d) Qual é o nome dos médicos da especialidade de Oftalmologia?

(e) Qual é o nome e a idade dos médicos com mais de 40 anos de idade da especialidade de Clínica Geral?

(f) Qual é o nome dos médicos da especialidade de Oftalmologia que consultaram pacientes da localidade de Braga?

(g) Qual é o nome e os anos de serviço dos médicos com mais de 50 anos de idade que deram consultas a partir das 12h a pacientes com menos de 20 anos de idade?

(h) Qual é o nome e a idade dos pacientes com mais de 10 anos de idade que nunca foram consultados na especialidade de Oftalmologia?

(i) Qual é o nome das especialidades que tiveram consultas no mês de janeiro de 2016?

(j) Qual é o nome dos médicos com mais de 30 anos de idade ou menos de 5 anos de serviço?

(k) Qual é o nome e a idade dos médicos de Clínica Geral que não consultaram em janeiro de 2016?

(l) Qual é o nome e a idade dos pacientes que já foram consultados por todos os médicos?

(m) Qual é o nome das especialidades que tiveram consultas no mês de janeiro e março de 2016?

(n) Qual é o nome dos médicos que nunca consultaram pacientes residentes em Braga?

(o) Qual é o nome e a idade dos pacientes que só foram consultados a Clínica Geral?

- a)

```
SELECT nome
FROM Clinica.MEDICO
WHERE idade(data_inicio_servico) > 10;
```
- b)

```
SELECT M.nome, E.designacao AS especialidade
FROM Clinica.MEDICO M
JOIN Clinica.ESPECIALIDADE E ON M.especialidade = E.id_especialidade;
```
- c)

```
SELECT nome, morada
FROM Clinica.PACIENTE
WHERE codigo_postal IN (SELECT codigo_postal FROM Clinica.CODIGO_POSTAL WHERE
localidade = 'BRAGA');
```
- d)

```
SELECT nome
FROM Clinica.MEDICO
WHERE especialidade = (SELECT id_especialidade FROM Clinica.ESPECIALIDADE WHERE
designacao = 'Oftalmologia');
```
- e)

```
SELECT nome, idade(data_nascimento) AS idade
FROM Clinica.MEDICO
WHERE especialidade = (SELECT id_especialidade FROM Clinica.ESPECIALIDADE WHERE
designacao = 'Clínica Geral')
AND idade(data_nascimento) > 40;
```
- f)

```
SELECT DISTINCT M.nome
FROM Clinica.MEDICO M
JOIN Clinica.CONSULTA C ON M.id_medico = C.id_medico
JOIN Clinica.PACIENTE P ON C.id_paciente = P.id_paciente
JOIN Clinica.CODIGO_POSTAL CP ON P.codigo_postal = CP.codigo_postal
JOIN Clinica.ESPECIALIDADE E ON M.especialidade = E.id_especialidade
WHERE E.designacao = 'Oftalmologia' AND CP.localidade = 'BRAGA';
```
- g)

```
SELECT M.nome, idade(M.data_nascimento) AS idade, DATEDIFF(CURDATE(), M.data_inicio_servico) AS
anos_servico
FROM Clinica.MEDICO M
JOIN Clinica.CONSULTA C ON M.id_medico = C.id_medico
JOIN Clinica.PACIENTE P ON C.id_paciente = P.id_paciente
WHERE idade(M.data_nascimento) > 50
AND TIME(C.data_hora) >= '12:00:00'
AND idade(P.data_nascimento) < 20;
```
- h)

```
SELECT P.nome, idade(P.data_nascimento) AS idade
FROM Clinica.PACIENTE P
WHERE idade(P.data_nascimento) > 10
AND P.id_paciente NOT IN (
SELECT DISTINCT C.id_paciente
FROM Clinica.CONSULTA C
JOIN Clinica.MEDICO M ON C.id_medico = M.id_medico
JOIN Clinica.ESPECIALIDADE E ON M.especialidade = E.id_especialidade
WHERE E.designacao = 'Oftalmologia'
);
```

- i) `SELECT DISTINCT E.designacao
FROM Clinica.CONSULTA C
JOIN Clinica.MEDICO M ON C.id_medico = M.id_medico
JOIN Clinica.ESPECIALIDADE E ON M.especialidade = E.id_especialidade
WHERE YEAR(C.data_hora) = 2016 AND MONTH(C.data_hora) = 1;`
- j) `SELECT nome
FROM Clinica.MEDICO
WHERE idade(data_nascimento) > 30 OR DATEDIFF(CURDATE(), data_inicio_servico) < 5;`
- k) `SELECT M.nome, idade(M.data_nascimento) AS idade
FROM Clinica.MEDICO M
LEFT JOIN Clinica.CONSULTA C ON M.id_medico = C.id_medico
LEFT JOIN Clinica.ESPECIALIDADE E ON M.especialidade = E.id_especialidade
WHERE E.designacao = 'Clínica Geral' AND (C.data_hora IS NULL OR NOT (YEAR(C.data_hora) = 2016 AND
MONTH(C.data_hora) = 1));`
- l) `SELECT P.nome, idade(P.data_nascimento) AS idade
FROM Clinica.PACIENTE P
WHERE NOT EXISTS (
 SELECT M.id_medico
 FROM Clinica.MEDICO M
 WHERE NOT EXISTS (
 SELECT C.id_medico
 FROM Clinica.CONSULTA C
 WHERE C.id_medico = M.id_medico AND C.id_paciente = P.id_paciente
)
);`
- m) `SELECT DISTINCT E.designacao
FROM Clinica.CONSULTA C
JOIN Clinica.MEDICO M ON C.id_medico = M.id_medico
JOIN Clinica.ESPECIALIDADE E ON M.especialidade = E.id_especialidade
WHERE (YEAR(C.data_hora) = 2016 AND MONTH(C.data_hora) IN (1, 3));`
- n) `SELECT M.nome
FROM Clinica.MEDICO M
WHERE NOT EXISTS (
 SELECT P.id_paciente
 FROM Clinica.PACIENTE P
 JOIN Clinica.CONSULTA C ON P.id_paciente = C.id_paciente
 WHERE M.id_medico = C.id_medico AND P.codigo_postal = 'Braga'
);`
- o) `SELECT P.nome, idade(P.data_nascimento) AS idade
FROM Clinica.PACIENTE P
WHERE P.id_paciente IN (
 SELECT C.id_paciente
 FROM Clinica.CONSULTA C
 JOIN Clinica.MEDICO M ON C.id_medico = M.id_medico
 JOIN Clinica.ESPECIALIDADE E ON M.especialidade = E.id_especialidade
 WHERE E.designacao = 'Clínica Geral'
 GROUP BY C.id_paciente
 HAVING COUNT(DISTINCT E.id_especialidade) = 1
);`

➔ PL07 – “Registo da Realização de Consultas numa Clínica”

1 Registo da Realização de Consultas numa Clínica

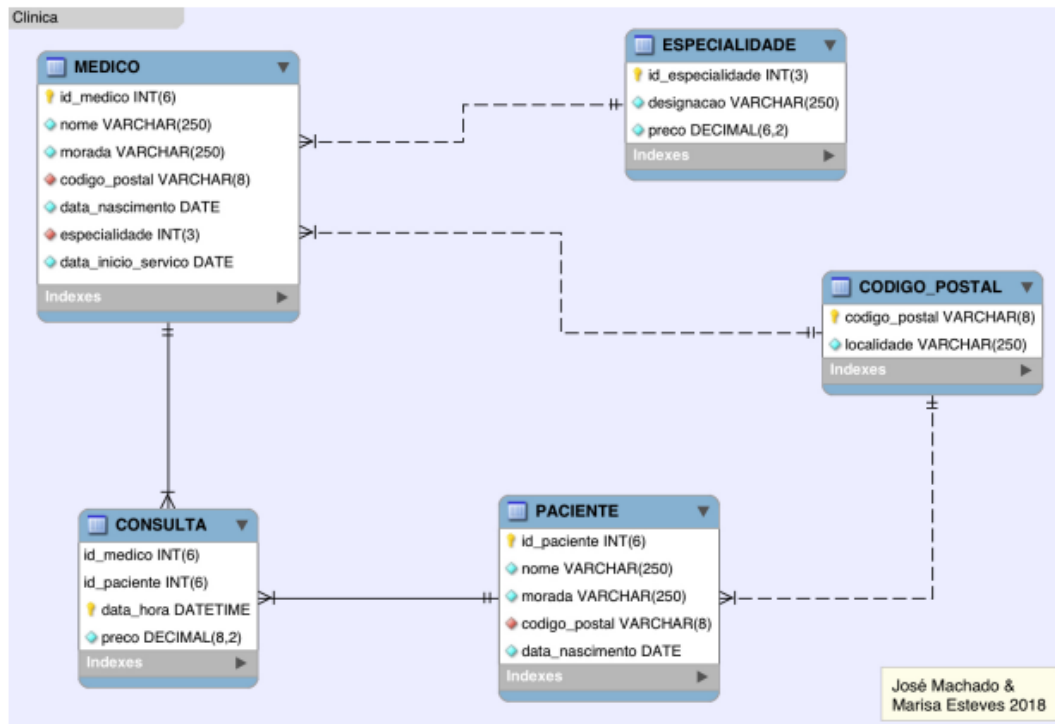


Figura 1: Esquema lógico da base de dados de uma determinada clínica.

Na Figura 1 é apresentado o esquema lógico, nomeadamente modelo relacional, da base de dados de uma determinada clínica.

A base de dados que o esquema representa está integrada no sistema operacional principal da clínica e suporta todos os processos relacionados com o registo e a faturação das consultas realizadas. É de referir que o valor pago pelo paciente em cada consulta pode ser superior ao valor de referência para a especialidade do médico (preço), uma vez que podem ser pagos procedimentos efetuados durante a realização da consulta.

Utilizando a base de dados povoada com o ficheiro “Ficha6_PovoarTabelas.sql”, disponibilizado juntamente com a 6.ª ficha prática laboratorial, pretende-se que:

1. Resolva utilizando funções de agregação as alíneas *k*, *l*, *n* e *o* da questão 2 da 6.ª ficha prática laboratorial.

```
k) SELECT M.nome, idade(M.data_nascimento) AS idade
FROM Clinica.MEDICO M
WHERE M.especialidade = (
    SELECT id_especialidade
    FROM Clinica.ESPECIALIDADE
    WHERE designacao = 'Clínica Geral'
)
AND NOT EXISTS (
    SELECT 1
    FROM Clinica.CONSULTA C
    WHERE C.id_medico = M.id_medico
    AND YEAR(C.data_hora) = 2016
    AND MONTH(C.data_hora) = 1
);
```

```

l) SELECT P.nome, idade(P.data_nascimento) AS idade
FROM Clinica.PACIENTE P
WHERE NOT EXISTS (
    SELECT M.id_medico
    FROM Clinica.MEDICO M
    WHERE NOT EXISTS (
        SELECT C.id_medico
        FROM Clinica.CONSULTA C
        WHERE C.id_medico = M.id_medico AND C.id_paciente = P.id_paciente
    )
);

n) SELECT M.nome
FROM Clinica.MEDICO M
WHERE NOT EXISTS (
    SELECT P.id_paciente
    FROM Clinica.PACIENTE P
    JOIN Clinica.CONSULTA C ON P.id_paciente = C.id_paciente
    WHERE M.id_medico = C.id_medico AND P.codigo_postal = 'Braga'
);

o) SELECT P.nome, idade(P.data_nascimento) AS idade
FROM Clinica.PACIENTE P
WHERE P.id_paciente IN (
    SELECT C.id_paciente
    FROM Clinica.CONSULTA C
    JOIN Clinica.MEDICO M ON C.id_medico = M.id_medico
    JOIN Clinica.ESPECIALIDADE E ON M.especialidade = E.id_especialidade
    WHERE E.designacao = 'Clínica Geral'
    GROUP BY C.id_paciente
    HAVING COUNT(DISTINCT E.id_especialidade) = 1
);

```

2. Utilizando SQL, desenvolva os comandos necessários para responder às seguintes questões:

- (a) Apresente a média das idades dos médicos com mais de 15 anos de serviço.
- (b) Apresente a média dos anos de serviço dos médicos para cada uma das especialidades. Devem ser apresentadas todas as especialidades, incluindo as que não tenham médicos associados.
- (c) Apresente o número de consultas que estão registradas por cada um dos médicos. Devem ser apresentados todos os médicos, incluindo os que nunca tenham dado consultas.
- (d) Apresente a média das idades dos pacientes por cada uma das localidades. Devem ser apresentadas todas as localidades, incluindo as que não tenham pacientes associados.
- (e) Apresente para cada médico o valor total faturado em 2016. Devem ser apresentados todos os médicos, incluindo os que nunca tenham dado consultas.
- (f) Apresente o número de médicos para cada uma das especialidades. Devem ser apresentadas todas as especialidades, incluindo as que não tenham médicos associados.
- (g) Para cada uma das especialidades com menos de dois médicos, apresente o valor máximo e o valor mínimo faturado para o conjunto das consultas, bem como o seu valor médio.
- (h) Apresente o nome do(s) médico(s) cujo valor médio faturado em 2016 por consulta seja superior à média por consulta desse mesmo ano, bem como o valor médio associado a cada um desses(s) médico(s).
- (i) Apresente o nome da(s) especialidade(s) que mais faturou (faturaram) em 2016, bem como o valor total associado a cada uma dessa(s) especialidade(s).
- (j) Apresente o nome dos três médicos que mais deram consultas em 2016, bem como o número de consultas associado a cada um desses médicos.

a) `SELECT AVG(idade(data_nascimento)) AS media_idades
FROM Clinica.MEDICO
WHERE DATEDIFF(CURDATE(), data_inicio_servico) > 15 * 365;`

b) `SELECT E.designacao AS especialidade, AVG(DATEDIFF(CURDATE(), M.data_inicio_servico) / 365) AS
media_anos_servico
FROM Clinica.ESPECIALIDADE E
LEFT JOIN Clinica.MEDICO M ON E.id_especialidade = M.especialidade
GROUP BY E.id_especialidade, E.designacao;`

c) `SELECT M.id_medico, M.nome, COUNT(C.id_medico) AS num_consultas
FROM Clinica.MEDICO M
LEFT JOIN Clinica.CONSULTA C ON M.id_medico = C.id_medico
GROUP BY M.id_medico, M.nome;`

d) `SELECT CP.localidade, AVG(idade(P.data_nascimento)) AS media_idades
FROM Clinica.CODIGO_POSTAL CP
LEFT JOIN Clinica.PACIENTE P ON CP.codigo_postal = P.codigo_postal
GROUP BY CP.codigo_postal, CP.localidade;`

e) `SELECT M.id_medico, M.nome, IFNULL(SUM(C.preco), 0) AS valor_total_faturado
FROM Clinica.MEDICO M
LEFT JOIN Clinica.CONSULTA C ON M.id_medico = C.id_medico
AND YEAR(C.data_hora) = 2016
GROUP BY M.id_medico, M.nome;`

f) `SELECT E.designacao AS especialidade, COUNT(M.id_medico) AS num_medicos
FROM Clinica.ESPECIALIDADE E
LEFT JOIN Clinica.MEDICO M ON E.id_especialidade = M.especialidade
GROUP BY E.id_especialidade, E.designacao;`

- g) `SELECT E.designacao AS especialidade,
MAX(C.preco) AS valor_maximo,
MIN(C.preco) AS valor_minimo,
AVG(C.preco) AS valor_medio
FROM Clinica.ESPECIALIDADE E
JOIN Clinica.MEDICO M ON E.id_especialidade = M.especialidade
LEFT JOIN Clinica.CONSULTA C ON M.id_medico = C.id_medico
GROUP BY E.id_especialidade, E.designacao
HAVING COUNT(DISTINCT M.id_medico) < 2;`
- h) `SELECT M.id_medico, M.nome, AVG(C.preco) AS valor_medio
FROM Clinica.MEDICO M
JOIN Clinica.CONSULTA C ON M.id_medico = C.id_medico
AND YEAR(C.data_hora) = 2016
GROUP BY M.id_medico, M.nome
HAVING AVG(C.preco) > (SELECT AVG(preco) FROM Clinica.CONSULTA WHERE YEAR(data_hora) = 2016);`
- i) `SELECT E.designacao AS especialidade,
SUM(C.preco) AS valor_total_faturado
FROM Clinica.ESPECIALIDADE E
JOIN Clinica.MEDICO M ON E.id_especialidade = M.especialidade
JOIN Clinica.CONSULTA C ON M.id_medico = C.id_medico
AND YEAR(C.data_hora) = 2016
GROUP BY E.id_especialidade, E.designacao
ORDER BY valor_total_faturado DESC
LIMIT 1;`
- j) `SELECT M.id_medico, M.nome, COUNT(C.id_medico) AS num_consultas
FROM Clinica.MEDICO M
LEFT JOIN Clinica.CONSULTA C ON M.id_medico = C.id_medico
AND YEAR(C.data_hora) = 2016
GROUP BY M.id_medico, M.nome
ORDER BY num_consultas DESC
LIMIT 3;`

➔ PL08 – “Registo da Realização de Consultas numa Clínica”

1 Registo da Realização de Consultas numa Clínica

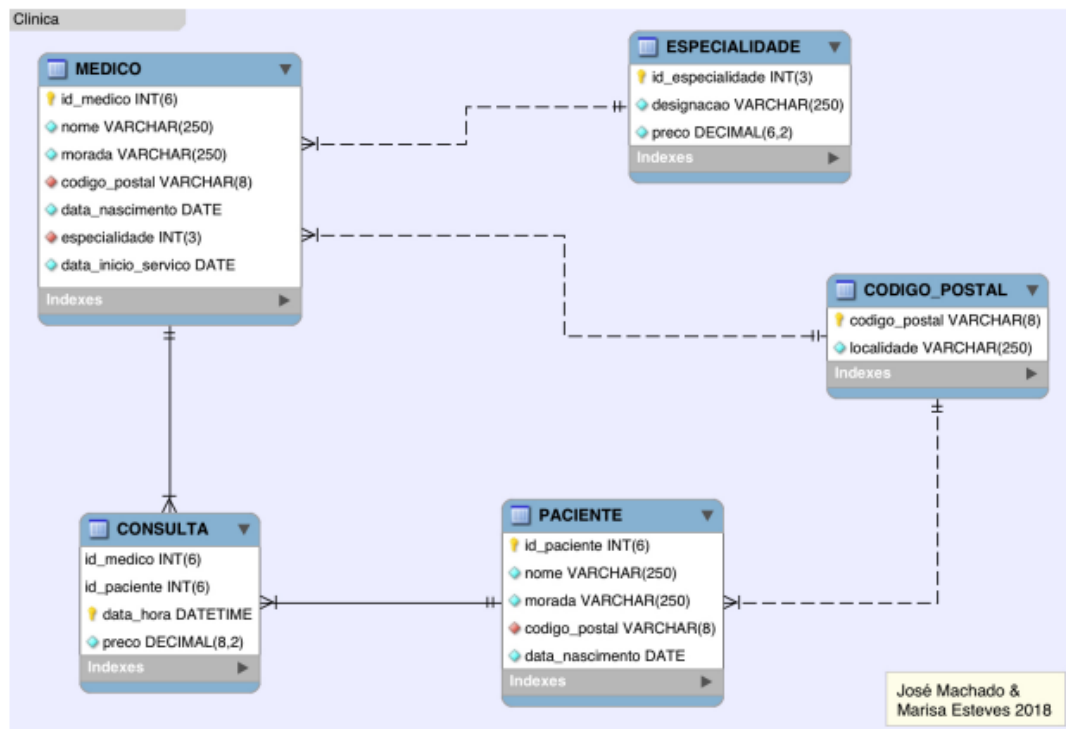


Figura 1: Esquema lógico da base de dados de uma determinada clínica.

Na Figura 1 é apresentado o esquema lógico, nomeadamente modelo relacional, da base de dados de uma determinada clínica.

A base de dados que o esquema representa está integrada no sistema operacional principal da clínica e suporta todos os processos relacionados com o registo e a faturação das consultas realizadas. É de referir que o valor pago pelo paciente em cada consulta pode ser superior ao valor de referência para a especialidade do médico (preço), uma vez que podem ser pagos procedimentos efetuados durante a realização da consulta.

1. Utilizando SQL, desenvolva os comandos necessários para responder às seguintes questões:

- Crie um procedimento denominado “Atualiza_Precio_Referencia_Especialidade” que atualize o preço de referência de cada especialidade com o valor médio cobrado nas consultas da especialidade num dado ano, acrescido de uma determinada percentagem. O ano e a percentagem são passados como parâmetros no procedimento. No entanto, caso uma determinada especialidade não tenha consultas associadas nesse mesmo ano, o valor médio cobrado nas consultas da especialidade no ano em questão é substituído pelo valor atual do preço de referência da especialidade na operação supramencionada.
- Remova da base de dados todos os médicos que nunca consultaram. Para implementar esta operação deverá criar uma função denominada “temConsultas(id_med)”.
- Remova da base de dados todos os códigos postais que não tenham ligação às tabelas *MEDICO* e *PACIENTE*. Para implementar esta operação deverá criar uma função denominada “utilizadoCodigoPostal(cp)”.
- Adicione um atributo denominado “total_faturado” na tabela *MEDICO* para acumular os valores faturados por cada um dos médicos nas suas consultas. Numa primeira etapa, pretende-se que este atributo seja carregado recorrendo aos dados já existentes na base de dados. Seguidamente, efetue igualmente as operações necessárias para que o referido atributo se mantenha sempre atualizado.
- Crie uma tabela denominada “PACIENTE_ACUMULADO_MENSAL”, com os seguintes atributos: “id_paciente”, “ano”, “mes” e “total_faturado”. Numa primeira etapa, pretende-se que esta tabela seja carregada recorrendo aos dados já existentes na base de dados. Seguidamente, implemente também as operações necessárias para que a tabela se mantenha sempre atualizada.

```

a) DELIMITER //
CREATE PROCEDURE Atualiza_Preco_Referencia_Especialidade (IN ano_consulta INT, IN
percentagem DECIMAL(5,2))
BEGIN
    DECLARE done INT DEFAULT FALSE;
    DECLARE id_esp INT;
    DECLARE preco_medio DECIMAL(8,2);
    DECLARE preco_referencia_atual DECIMAL(6,2);

    DECLARE cur CURSOR FOR
        SELECT id_especialidade, preco
        FROM Clinica.ESPECIALIDADE;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    OPEN cur;

    read_loop: LOOP
        FETCH cur INTO id_esp, preco_referencia_atual;
        IF done THEN
            LEAVE read_loop;
        END IF;

        -- Calcula o preço médio das consultas da especialidade no ano especificado
        SELECT AVG(preco) INTO preco_medio
        FROM Clinica.CONSULTA C
        JOIN Clinica.MEDICO M ON C.id_medico = M.id_medico
        WHERE M.especialidade = id_esp AND YEAR(C.data_hora) = ano_consulta;

        -- Atualiza o preço de referência
        IF preco_medio IS NOT NULL THEN
            UPDATE Clinica.ESPECIALIDADE
            SET preco = preco_medio + (preco_medio * (percentagem / 100))
            WHERE id_especialidade = id_esp;
        END IF;

    END LOOP;

    CLOSE cur;
END //
DELIMITER ;

b) DELIMITER //
CREATE FUNCTION temConsultas(id_med INT) RETURNS BOOLEAN
BEGIN
    DECLARE tem_consultas BOOLEAN;

    SELECT COUNT(*) INTO tem_consultas
    FROM Clinica.CONSULTA
    WHERE id_medico = id_med
    LIMIT 1;

    RETURN tem_consultas;

```

```

END //
DELIMITER ;

DELETE FROM Clinica.MEDICO
WHERE NOT temConsultas(id_medico);

c) DELIMITER //
CREATE FUNCTION utilizadoCodigoPostal(cp VARCHAR(8)) RETURNS BOOLEAN
BEGIN
    DECLARE utilizado BOOLEAN;

    SELECT EXISTS (
        SELECT 1
        FROM Clinica.MEDICO M
        WHERE M.codigo_postal = cp
        UNION
        SELECT 1
        FROM Clinica.PACIENTE P
        WHERE P.codigo_postal = cp
    ) INTO utilizado;

    RETURN utilizado;
END //
DELIMITER ;

DELETE FROM Clinica.CODIGO_POSTAL
WHERE NOT utilizadoCodigoPostal(codigo_postal);

d) -- Adiciona o atributo "total_faturado"
ALTER TABLE Clinica.MEDICO
ADD COLUMN total_faturado DECIMAL(10,2) DEFAULT 0;

-- Carrega os valores existentes
UPDATE Clinica.MEDICO M
SET M.total_faturado = (
    SELECT COALESCE(SUM(C.preco), 0)
    FROM Clinica.CONSULTA C
    WHERE C.id_medico = M.id_medico
);

DELIMITER //
CREATE TRIGGER atualiza_total_faturado
AFTER INSERT ON Clinica.CONSULTA
FOR EACH ROW
BEGIN
    UPDATE Clinica.MEDICO M
    SET M.total_faturado = M.total_faturado + NEW.preco
    WHERE M.id_medico = NEW.id_medico;
END;
//
DELIMITER ;

```

```

e) -- Criação da tabela
CREATE TABLE Clinica.PACIENTE_ACUMULADO_MENSAL (
  id_paciente INT(6) NOT NULL,
  ano INT,
  mes INT,
  total_faturado DECIMAL(10,2),
  PRIMARY KEY (id_paciente, ano, mes),
  FOREIGN KEY (id_paciente) REFERENCES Clinica.PACIENTE (id_paciente)
) ENGINE = InnoDB;

-- Carregamento inicial
INSERT INTO Clinica.PACIENTE_ACUMULADO_MENSAL (id_paciente, ano, mes, total_faturado)
SELECT
  P.id_paciente,
  YEAR(C.data_hora) AS ano,
  MONTH(C.data_hora) AS mes,
  COALESCE(SUM(C.preco), 0) AS total_faturado
FROM
  Clinica.CONSULTA C
JOIN
  Clinica.PACIENTE P ON C.id_paciente = P.id_paciente
GROUP BY
  P.id_paciente, YEAR(C.data_hora), MONTH(C.data_hora);

DELIMITER //
CREATE TRIGGER atualiza_paciente_acumulado_mensal
AFTER INSERT ON Clinica.CONSULTA
FOR EACH ROW
BEGIN
  DECLARE total_faturado_atual DECIMAL(10,2);

  -- Verifica se o registro já existe para o paciente no ano e mês da nova consulta
  SELECT total_faturado INTO total_faturado_atual
  FROM Clinica.PACIENTE_ACUMULADO_MENSAL
  WHERE id_paciente = NEW.id_paciente AND ano = YEAR(NEW.data_hora) AND mes =
  MONTH(NEW.data_hora);

  -- Se o registro já existe, atualiza o total_faturado
  IF total_faturado_atual IS NOT NULL THEN
    UPDATE Clinica.PACIENTE_ACUMULADO_MENSAL
    SET total_faturado = total_faturado_atual + NEW.preco
    WHERE id_paciente = NEW.id_paciente AND ano = YEAR(NEW.data_hora) AND mes =
    MONTH(NEW.data_hora);
  -- Se o registro não existe, insere um novo registro
  ELSE
    INSERT INTO Clinica.PACIENTE_ACUMULADO_MENSAL (id_paciente, ano, mes,
    total_faturado)
    VALUES (NEW.id_paciente, YEAR(NEW.data_hora), MONTH(NEW.data_hora), NEW.preco);
  END IF;
END;
//
DELIMITER ;

```