



Quem se prepara, não para.



# Banco de Dados

2º período

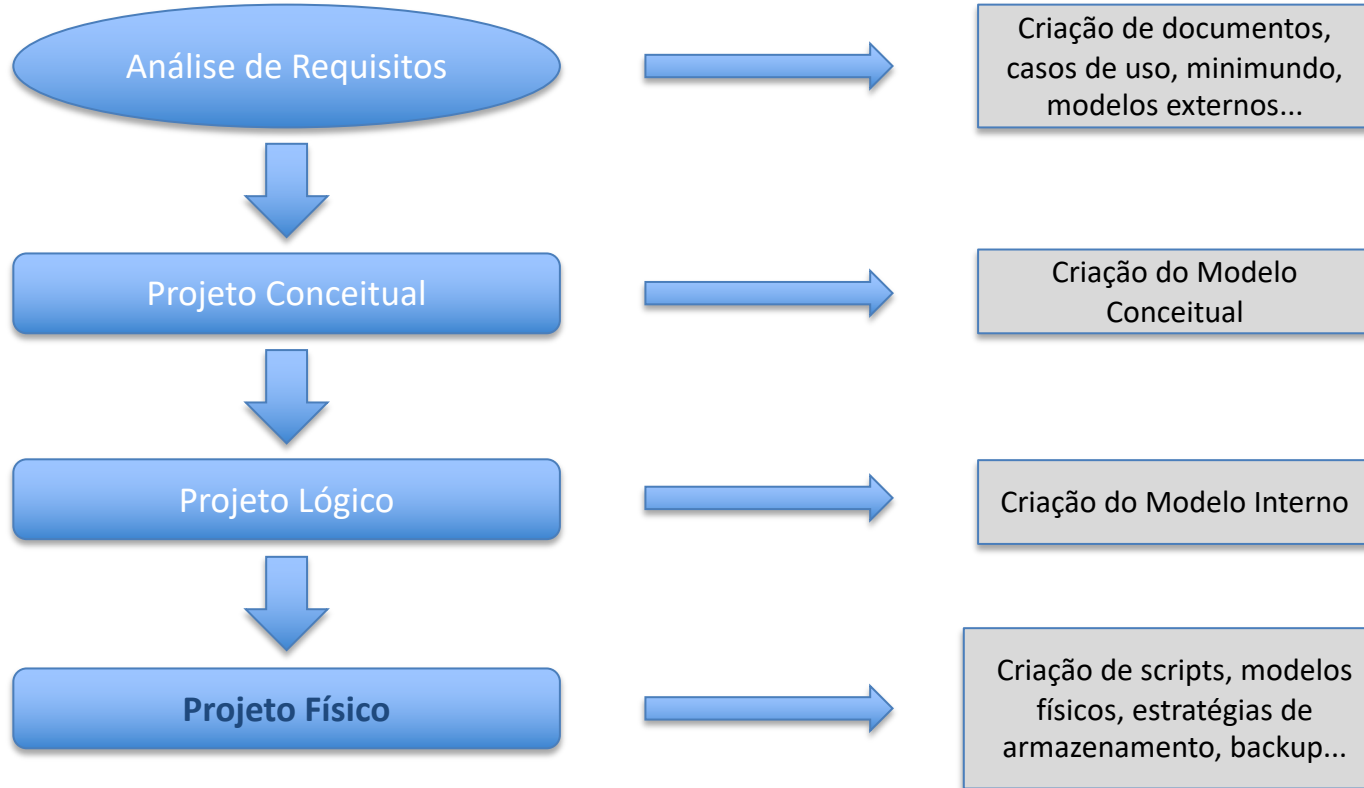
Prof. Dr. João Paulo Aramuni



# Projeto Físico de Bando de Dados

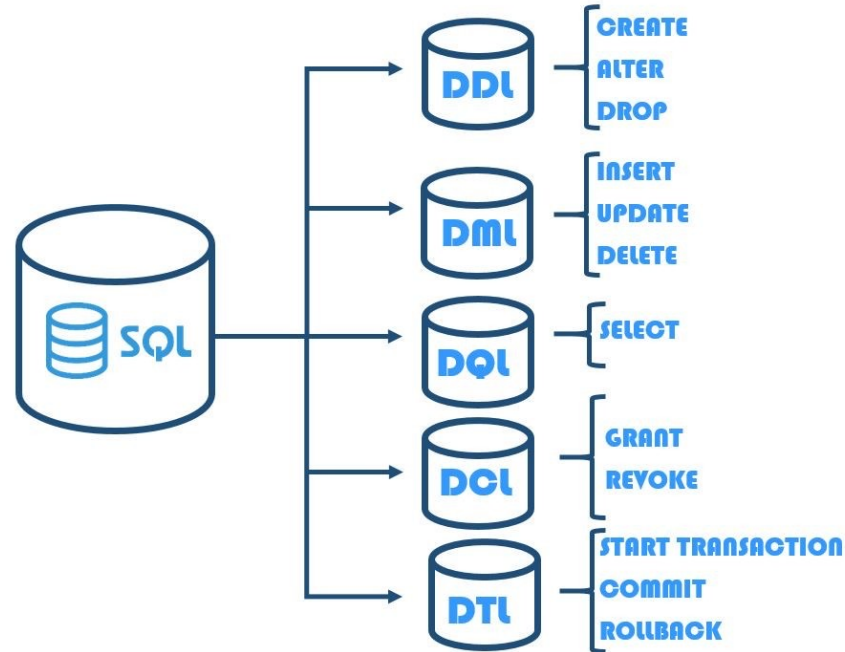
## Aula 10

# Projeto Físico de BD



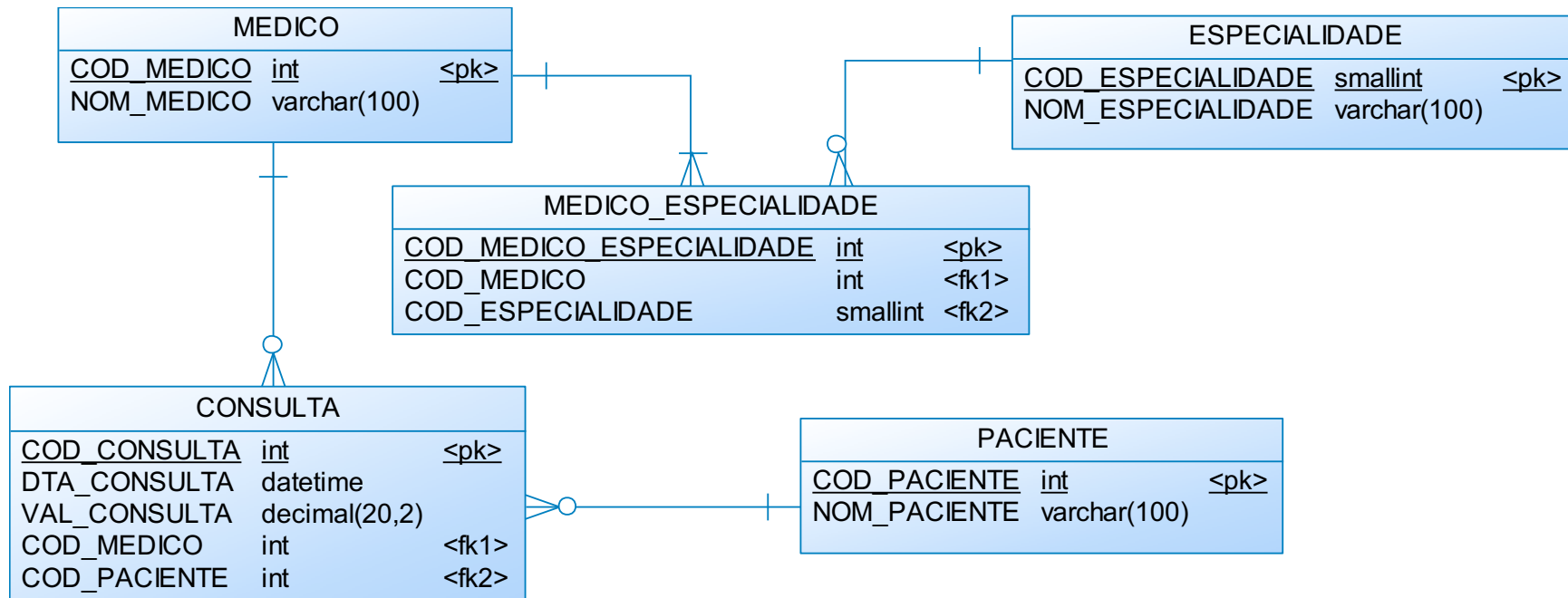
# Projeto Físico de BD

## Divisão da Linguagem SQL



# Projeto Físico de BD

Construa um arquivo .sql contendo o script DDL para o DER do BD abaixo:



# Projeto Físico de BD

- **DDL** (*Data Definition Language*): Linguagem de Definição de Dados.

-- Database **BD\_CLINICA\_MEDICA**

**CREATE DATABASE** BD\_CLINICA\_MEDICA;

**USE** BD\_CLINICA\_MEDICA;

-- Tabela **MEDICO**

**CREATE TABLE** MEDICO

(

COD\_MEDICO            **int not null auto\_increment primary key,**

NOM\_MEDICO           **varchar(100) not null**

);

# Projeto Físico de BD

- **DDL** (*Data Definition Language*): Linguagem de Definição de Dados.

-- Tabela ESPECIALIDADE

**CREATE TABLE ESPECIALIDADE**

```
(  
  COD_ESPECIALIDADE      smallint not null auto_increment primary key,  
  NOM_ESPECIALIDADE      varchar(100) not null  
);
```

-- Tabela PACIENTE

**CREATE TABLE PACIENTE**

```
(  
  COD_PACIENTE           int not null auto_increment primary key,  
  NOM_PACIENTE           varchar(100)  
);
```



# Projeto Físico de BD

- **DDL** (*Data Definition Language*): Linguagem de Definição de Dados.

-- Tabela MEDICO\_ESPECIALIDADE

**CREATE TABLE** MEDICO\_ESPECIALIDADE

(

COD_MEDICO_ESPECIALIDADE	int not null auto_increment primary key,
COD_MEDICO	int not null,
COD_ESPECIALIDADE	smallint not null

);

# Projeto Físico de BD

- **DDL** (*Data Definition Language*): Linguagem de Definição de Dados.

-- Tabela CONSULTA

**CREATE TABLE CONSULTA**

```
(  
    COD_CONSULTA          int not null auto_increment primary key,  
    DTA_CONSULTA          datetime not null,  
    VAL_CONSULTA          decimal(20,2),  
    COD_MEDICO            int not null,  
    COD_PACIENTE          int not null  
);
```

# Projeto Físico de BD

- **DDL** (*Data Definition Language*): Linguagem de Definição de Dados.

```
ALTER TABLE CONSULTA add constraint FK_MEDICO_CONSULTA  
foreign key (COD_MEDICO) references MEDICO (COD_MEDICO);
```

```
ALTER TABLE CONSULTA add constraint FK_PACIENTE_CONSULTA  
foreign key (COD_PACIENTE) references PACIENTE (COD_PACIENTE);
```

```
ALTER TABLE MEDICO_ESPECIALIDADE add constraint FK_ESPECIALIDADE_MEDICO_ESPECIALIDADE  
foreign key (COD_ESPECIALIDADE) references ESPECIALIDADE (COD_ESPECIALIDADE);
```

```
ALTER TABLE MEDICO_ESPECIALIDADE add constraint FK_MEDICO_MEDICO_ESPECIALIDADE  
foreign key (COD_MEDICO) references MEDICO (COD_MEDICO);
```

# Projeto Físico de BD

- **DDL** (*Data Definition Language*): Linguagem de Definição de Dados.

-- Foreign key alternativa

```
ALTER TABLE CONSULTA add foreign key (COD_MEDICO)  
references MEDICO (COD_MEDICO);
```

```
ALTER TABLE CONSULTA add foreign key (COD_PACIENTE)  
references PACIENTE (COD_PACIENTE);
```

```
ALTER TABLE MEDICO_ESPECIALIDADE add foreign key (COD_ESPECIALIDADE)  
references ESPECIALIDADE (COD_ESPECIALIDADE);
```

```
ALTER TABLE MEDICO_ESPECIALIDADE add foreign key (COD_MEDICO)  
references MEDICO (COD_MEDICO);
```

# Projeto Físico de BD

- **DDL** (*Data Definition Language*): Linguagem de Definição de Dados.

-- Primary key alternativa

**CREATE TABLE MEDICO**

```
(  
  COD_MEDICO      int not null auto_increment,  
  NOM_MEDICO      varchar(100) not null, primary key (COD_MEDICO)  
);
```

# Projeto Físico de BD

- **DDL** (*Data Definition Language*): Linguagem de Definição de Dados.

-- PK e FK dentro do CREATE TABLE

**CREATE TABLE CONSULTA**

```
(  
  COD_CONSULTA          int not null auto_increment,  
  DTA_CONSULTA          datetime not null,  
  VAL_CONSULTA          decimal(20,2),  
  COD_MEDICO            int not null,  
  COD_PACIENTE          int not null,  
  primary key (COD_CONSULTA),  
  foreign key (COD_MEDICO) references MEDICO (COD_MEDICO),  
  foreign key (COD_PACIENTE) references PACIENTE (COD_PACIENTE)  
);
```

- **DDL** (*Data Definition Language*): Linguagem de Definição de Dados.

-- Dicionário de dados no script

```
CREATE TABLE CONSULTA
```

```
(
```

```
COD_CONSULTA int not null auto_increment
```

```
comment 'PK da tabela CONSULTA, gerada automaticamente',
```

```
DTA_CONSULTA datetime not null
```

```
comment 'Data em que a consulta foi agendada pelo paciente',
```

```
VAL_CONSULTA decimal(20,2)
```

```
comment 'Valor a ser pago pela consulta',
```

```
COD_MEDICO int not null
```

```
comment 'Codigo do médico que vai realizar a consulta',
```

```
COD_PACIENTE int not null
```

```
comment 'Codigo do paciente agendado na consulta'
```

```
);
```

```
ALTER TABLE CONSULTA comment 'Tabela que contém os dados das consultas agendadas';
```

# Projeto Físico de BD

- **DDL** (*Data Definition Language*): Linguagem de Definição de Dados.

-- A cláusula **DEFAULT** e **CHECK**

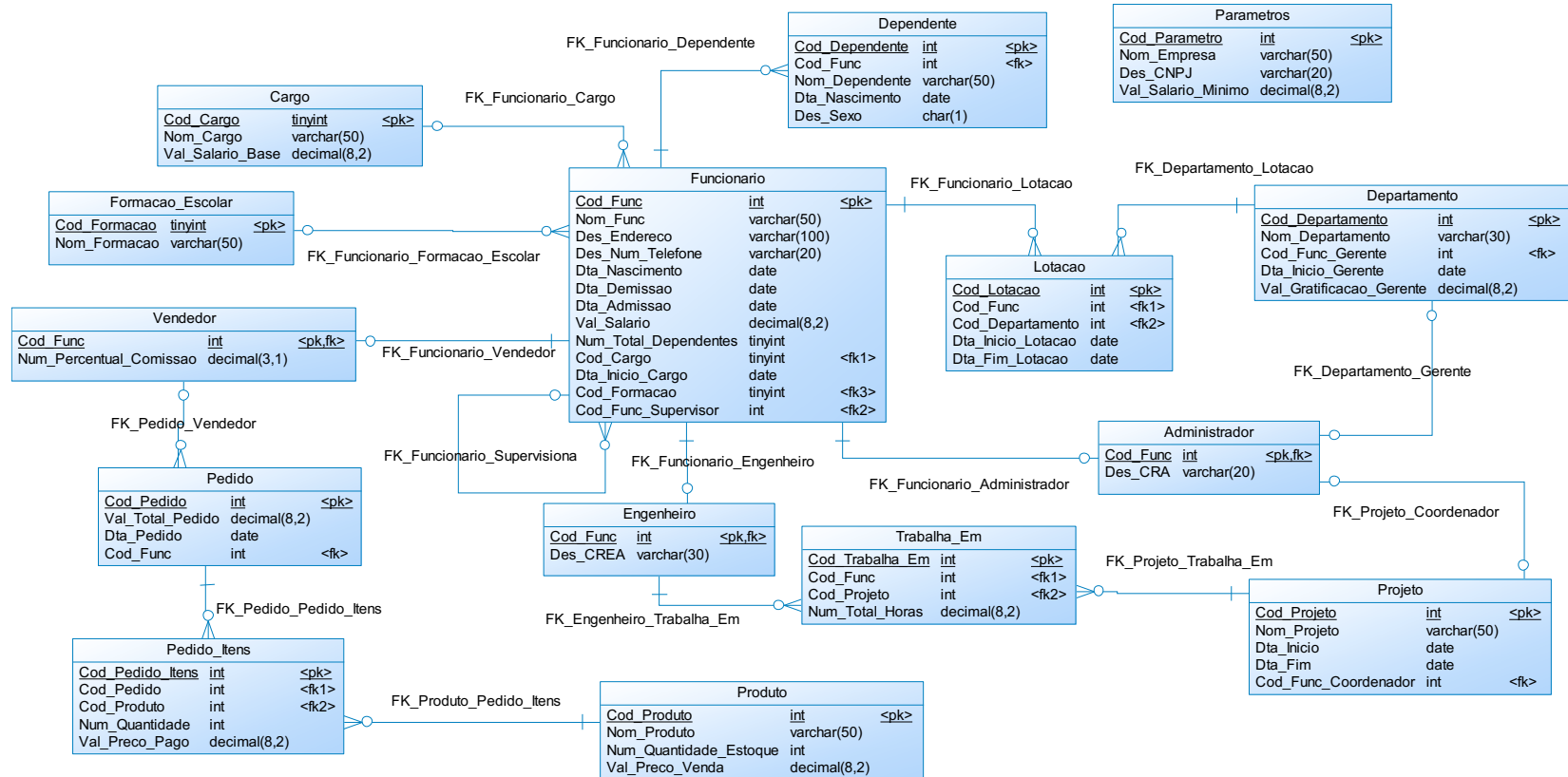
**CREATE TABLE CONSULTA**

```
(  
  COD_CONSULTA      int not null auto_increment primary key,  
  DTA_CONSULTA      datetime not null DEFAULT CURRENT_TIMESTAMP,  
  VAL_CONSULTA      decimal(20,2) CHECK (VAL_CONSULTA IS NULL OR VAL_CONSULTA >= 0),  
  COD_MEDICO        int not null,  
  COD_PACIENTE      int not null  
);
```

As cláusulas **DEFAULT** e **CHECK** são recursos do SQL que permitem definir valores padrão para colunas em tabelas (DEFAULT) e impor restrições em valores inseridos ou atualizados (CHECK).



Construa um arquivo .sql contendo o script DDL para o DER do BD abaixo:



# Projeto Físico de BD

## Pontos de atenção:

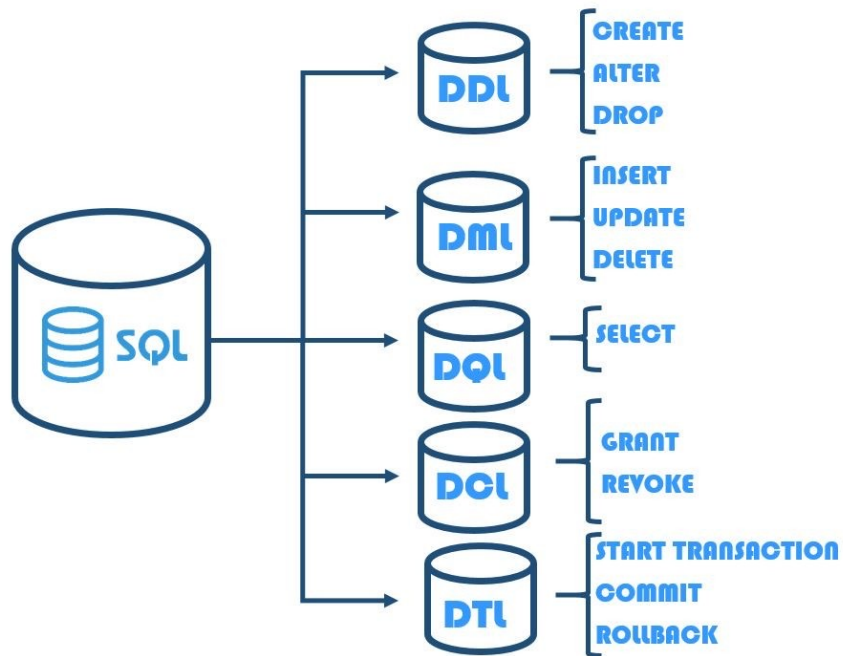
**ALTER TABLE** Funcionario **add constraint** FK\_Funcionario\_Supervisiona  
**foreign key** (Cod\_Func\_Supervisor) **references** Funcionario (Cod\_Func);

**ALTER TABLE** Administrador **add primary key** (Cod\_Func);

**ALTER TABLE** Administrador **add constraint** FK\_Funcionario\_Administrador  
**foreign key** (Cod\_Func) **references** Funcionario (Cod\_Func);

# Projeto Físico de BD

## Divisão da Linguagem SQL



# Projeto Físico de BD

- **DML** (*Data Manipulation Language*): Linguagem de Manipulação de Dados.

**INSERT INTO** Cargo (Cod\_Cargo, Nom\_Cargo, Val\_Salario\_Base) **VALUES** (1, 'Desenvolvedor Junior', 3000);

**UPDATE** Funcionario SET Val\_Salario = 3300 **WHERE** Cod\_Func = 1;

**DELETE FROM** Trabalha\_Em **WHERE** Cod\_Func = 2 **AND** Cod\_Projeto = 1;

**SELECT \* FROM** Funcionario;

Cuidado com Deletes sem Where!

# Projeto Físico de BD

- **DML** (*Data Manipulation Language*): Linguagem de Manipulação de Dados.

## Cláusula **WHERE**:

A cláusula **WHERE** é uma parte fundamental da linguagem SQL e é usada em consultas para filtrar os resultados de uma consulta com base em condições específicas. Ela permite que você especifique critérios para selecionar apenas as linhas que atendem a essas condições.

A condição na cláusula **WHERE** é uma expressão lógica que combina colunas e/ou valores usando operadores relacionais e lógicos. Por exemplo, você pode usar operadores como "=" (igual), ">" (maior que), "<" (menor que), ">=" (maior ou igual a), "<=" (menor ou igual a), "AND" (e), "OR" (ou), entre outros.

# Projeto Físico de BD

- **DML** (*Data Manipulation Language*): Linguagem de Manipulação de Dados.
- Mais alguns exemplos:

```
-- INSERT
-- Inserindo um registro na tabela Livro
INSERT INTO Livro (LivroId, Titulo, Autor, ISBN, Quantidade)
VALUES (1, 'Introdução aos Fundamentos da Computação', 'Newton José Vieira', 'ISBN 123456', 10);

-- Inserindo um registro na tabela Aluno
INSERT INTO Aluno (Matricula, Nome)
VALUES (1, 'Aramuni');

-- Inserindo um registro na tabela Emprestimo
INSERT INTO Emprestimo (EmprestimoId, LivroId, Matricula, DataHora_Emprestimo)
VALUES (1, 1, 1, '2023-07-08 19:00:00');

-- SELECT
SELECT * FROM Livro WHERE Titulo LIKE 'Introdução%';
SELECT * FROM Aluno WHERE Nome LIKE 'Ara%';
SELECT * FROM Emprestimo WHERE LivroId LIKE '%1'
```

**DDL**

```
-- UPDATE
UPDATE Livro SET Titulo = 'Introdução aos Compiladores' WHERE LivroId = 1;
UPDATE Aluno SET Nome = 'João Paulo' WHERE Matricula = 1;
UPDATE Emprestimo SET DataHora_Emprestimo = '2023-07-08 22:00:00' WHERE EmprestimoId = 1;

-- DELETE
DELETE FROM Emprestimo WHERE EmprestimoId = 1;
DELETE FROM Aluno WHERE Matricula = 1;
DELETE FROM Livro WHERE LivroId = 1;
```

# Projeto Físico de BD

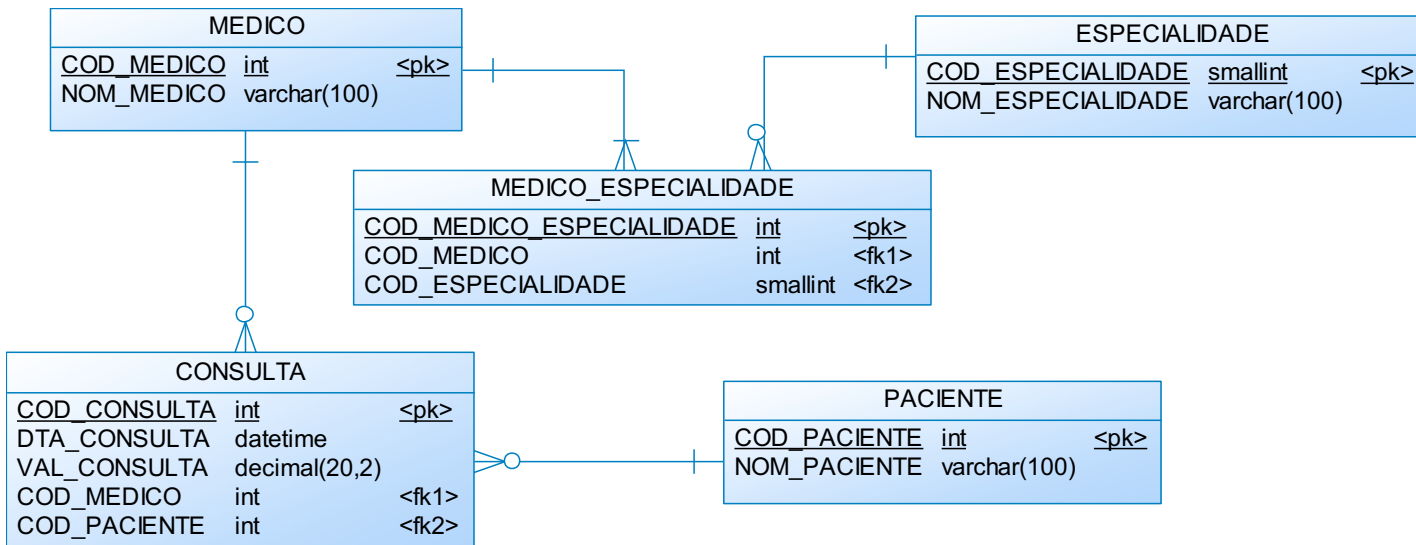
## Exercício 1

- Execute o script de criação e preenchimento do BD\_RH no MySQL.
- Insira mais um cargo na respectiva tabela.
- Insira mais um funcionário vinculado ao cargo acima e sem supervisor.
- Atualize o funcionário anterior incluindo um supervisor para ele.
- Remova o funcionário anterior.
- Atualize o salario dos funcionário no cargo 2 em 10%.

# Projeto Físico de BD

## Exercício 2

Crie o BD para o DER abaixo e insira 10 linhas.







Obrigado!

joaopauloaramuni@gmail.com