

Exercício 6.1 A

ÑN

PACIENTE(nro_paciente , nome_paciente, nro_quarto, descrição_quarto, nro_cômodos_quarto, (cod_médico, nome_médico, fone_médico))

1FN: (Eliminar tabelas aninhadas)

PACIENTE(nro_paciente , nome_paciente, nro_quarto, descrição_quarto, nro_cômodos_quarto)
PACIENTE_MEDICO(nro_paciente , cod_médico , nome_médico, fone_médico)

2FN: (Dependência funcional parcial)

PACIENTE(nro_paciente , nome_paciente, nro_quarto, descrição_quarto, nro_cômodos_quarto)
PACIENTE_MEDICO(nro_paciente , cod_médico)
MEDICO(cod_médico , nome_médico, fone_médico)

3FN: (Dependência Funcional Transitiva)

PACIENTE(nro_paciente , nome_paciente, nro_quarto)
PACIENTE_MEDICO(nro_paciente , cod_médico)
MEDICO(cod_médico , nome_médico, fone_médico)
QUARTO(nro_quarto , descrição_quarto, nro_cômodos_quarto)

Observem que a coluna nro_quarto será um atributo (chave estrangeira) da tabela QUARTO.

Exercício 6.1 B

Modelo lógico - Esquema textual resumido

Medico (medicold , nome, telefone)

Quarto (quartold , numero, descricao, qtdcomodos)

Paciente (pacienteld, nome, quartold)

quartold referencia Quarto

Consulta (pacienteld, medicold)

pacienteld referencia Paciente

medicold referencia Medico

Exercício 6.1 C

Modelo físico

```
CREATE TABLE Medico(  
    medicold INTEGER NOT NULL,  
    nome VARCHAR(60),  
    telefone VARCHAR(20),  
    CONSTRAINT MEDICO_PK PRIMARY KEY (medicold));
```

```
CREATE TABLE Quarto(  
    quartold INTEGER NOT NULL,  
    numero INTEGER,  
    descricao VARCHAR(50),  
    qtdcomodos INTEGER,  
    CONSTRAINT QUARTO_PK PRIMARY KEY (quartold));
```

```
CREATE TABLE Paciente(  
    pacienteld INTEGER NOT NULL,  
    nome VARCHAR(60),  
    quartold INTEGER,  
    CONSTRAINT PACIENTE_PK PRIMARY KEY (pacienteld),  
    CONSTRAINT QUARTO_FK FOREIGN KEY (quartold) references QUARTO (quartold));
```

```
CREATE TABLE Consulta (  
    pacienteld INTEGER NOT NULL,  
    medicold INTEGER NOT NULL,  
    CONSTRAINT CONSULTA_PK PRIMARY KEY (pacienteld, medicold),  
    CONSTRAINT PACIENTE_FK FOREIGN KEY (pacienteld) references PACIENTE (pacienteld),  
    CONSTRAINT MEDICO_FK FOREIGN KEY (medicold) references MEDICO (medicold));
```

Exercício 6.2 A

ÑÑ

ALUNO (nro_aluno, nome_aluno, nro_créditos, ano_ingresso, (nro_curso, descrição_curso, nro_depto, nome_depto))

1FN: (Eliminar tabelas aninhadas)

ALUNO (nro_aluno, nome_aluno, nro_créditos, ano_ingresso)

ALUNO_CURSO (nro_curso, nro_aluno, descrição_curso, nro_depto, nome_depto)

2FN: (Dependência Funcional Parcial)

ALUNO (nro_aluno, nome_aluno, nro_créditos, ano_ingresso)

ALUNO_CURSO (nro_curso, nro_aluno)

CURSO (nro_curso, descrição_curso, nro_depto, nome_depto)

3FN: (Dependência Funcional Transitiva)

ALUNO (nro_aluno, nome_aluno, nro_créditos, ano_ingresso)

ALUNO_CURSO (nro_curso, nro_aluno)

CURSO (nro_curso, descrição_curso, nro_depto)

DEPARTAMENTO (nro_depto, nome_depto)

Exercício 6.2 B

Modelo lógico - Esquema textual resumido

Departamento (departamentold, nome)

Curso (cursold, descricao, departamentold)

departamentold referencia Departamento (departamentold)

Aluno (alunold, nome, nrocreditos, anoingresso)

Matricula (cursold, alunold)

cursold referencia Curso (cursold)

alunold referencia Aluno (alunold)

Exercício 6.2 C

Modelo físico

```
CREATE TABLE Departamento(  
    departamentoid INTEGER NOT NULL,  
    nome VARCHAR(50),  
    CONSTRAINT DEPARTAMENTO_PK PRIMARY KEY (departamentoid));
```

```
CREATE TABLE Curso(  
    cursoid INTEGER NOT NULL,  
    descricao VARCHAR(50),  
    departamentoid INTEGER,  
    CONSTRAINT CURSO_PK PRIMARY KEY (cursoid),  
    CONSTRAINT DEPARTAMENTO_FK FOREIGN KEY (departamentoid) references Departamento (departamentoid));
```

```
CREATE TABLE Aluno(  
    alunoid INTEGER NOT NULL,  
    nome VARCHAR(50),  
    nrocreditos INTEGER,  
    anoingresso INTEGER,  
    CONSTRAINT ALUNO_PK PRIMARY KEY (alunoid));
```

```
CREATE TABLE Matricula(  
    cursoid INTEGER NOT NULL,  
    alunoid INTEGER NOT NULL,  
    CONSTRAINT MATRICULA_PK PRIMARY KEY (cursoid, alunoid),  
    CONSTRAINT CURSO_FK FOREIGN KEY (cursoid) references Curso (cursoid),  
    CONSTRAINT ALUNO_FK FOREIGN KEY (alunoid) references Aluno (alunoid));
```