



UNIVERSIDADE FEDERAL DE OURO PRETO - UFOP
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS - ICEB
DEPARTAMENTO DE COMPUTAÇÃO - DECOM

18,8 / 25,0

TRABALHO PRÁTICO 2
MODELAGEM CONCEITUAL, RELACIONAL, LÓGICA E FÍSICA DE
BANCO DE DADOS

Brenda Sotero Ferreira - 15.1.5838

Gabriel Bruno Pereira Negri - 19.1.4976

Júlia Eduarda Miranda de Sousa - 18.1.4084

Joelington de Almeida Bernardo - 16.2.4348

Karina Brandão Pimenta - 16.1.1084

Thales Guedes Rodrigues - 17.1.1092

Ray da Silva Basílio - 17.2.5972

Renymara Hanna Macêdo Santos - 16.1.1503

Phillipi Murta Silva - 16.1.1395

OURO PRETO

2021

TRABALHO PRÁTICO 2
MODELAGEM CONCEITUAL, RELACIONAL, LÓGICA E FÍSICA DE
BANCO DE DADOS

Relatório do Trabalho Prático I da
disciplina de Banco de Dados I, que
consiste na elaboração dos requisitos de
dados de um sistema de banco de dados e
do consequente esquema conceitual
Entidade Relacionamento Estendido
(ERE)

OURO PRETO

2021

SUMÁRIO

INTRODUÇÃO	4
MINI-MUNDO	4
DICIONÁRIO DE DADOS	5
3.1 ENTIDADES	5
3.1.1 Cliente	5
3.1.2 Funcionário	6
3.1.3 Horista	7
3.1.4 Contratado	7
3.1.5 Recurso	8
3.1.6 Pagamento Locação	8
3.1.7 Pagamento em cartão	9
3.1.8 Boleto	10
3.1.9 Tipo do Esporte	11
3.1.10 Endereço	11
3.2 RELACIONAMENTOS	12
3.2.1 Reside	12
3.2.2 Agendamento	12
3.2.3 Possui	13
4.0 MODELO CONCEITUAL	14
5.0 MAPEAMENTO DO ERE PARA O RELACIONAL	15
6.0 SCRIPT	16

Item 1: 4,0 / 4,0

1. INTRODUÇÃO

Este trabalho consiste na elaboração dos requisitos de dados de um sistema de banco de dados e de seu esquema conceitual Entidade Relacionamento Estendido (ERE). A documentação contém uma descrição textual detalhada dos requisitos de dados do sistema, o esquema ERE completo dos dados na notação aprendida em sala de aula e um dicionário de dados com descrições, listagem de atributos e relacionamentos.

1,0 / 1,0

2. MINI-MUNDO

O mini-mundo deste trabalho consiste no gerenciamento do aluguel de quadras esportivas na cidade de Mariana – MG. A cidade de Mariana oferece dezenas de quadras espalhadas no município, sendo algumas delas particulares. O serviço de aluguel procura promover praticidade e melhor estrutura para quem deseja desfrutar de uma experiência mais completa e organizada.

Um exemplo desse serviço é aquele oferecido pelo clube Guaranix, que já faz parte da história da cidade. Este trabalho foi idealizado exatamente para auxiliar no gerenciamento dessa demanda para este clube, sendo sua motivação principal. A ideia é que qualquer pessoa que tenha interesse de fidelizar ao clube tornar-se-á seu cliente.

Como mini-mundo, tem-se que: um funcionário no sistema tem um nome, NSS (único), e-mail, senha, data de nascimento, CPF e nome. Além disso, um funcionário reside em um ou mais endereços. Para identificar cada endereço de um funcionário existe um número identificador (sequencial) e o CEP, também consta as demais informações como: rua, número, complemento, bairro. Um funcionário caracteriza como contratado, com matrícula, turno, cargo e salário, ou como horista, com escala, horas trabalhadas e o valor/hora.

Para realizar o agendamento de um horário numa quadra é necessário que haja pelo menos um funcionário responsável, um cliente e um recurso (que representa uma quadra) envolvidos. Também tem o pagamento da locação, que é gerado a cada agendamento feito. O agendamento ainda tem hora de início, hora final e a data que será marcado. Um cliente tem no sistema uma matrícula (único), seu nome e seu telefone de contato. Já um recurso tem um código (único), nome, valor por hora, se é uma quadra coberta ou não e um tipo. O tipo do recurso se caracteriza pelo seu código, modalidade e descrição.

Considerando o pagamento que deve ser realizado ao realizar um agendamento, ele tem um código (único), uma taxa que pode ser cobrada ou não, um desconto que também pode ser dado ou não, o valor bruto, o valor líquido e a hora que o pagamento foi realizado. Este pagamento pode ser feito através de boleto, que tem código data de vencimento e de emissão ou por cartão, com as parcelas, juros, bandeira, número do cartão, CVV e o nome do titular do cartão.

3. DICIONÁRIO DE DADOS

O Dicionário de dados contém uma descrição textual de cada entidade, relacionamento e atributo. Para cada entidade, estão descritas sua semântica e sua lista de atributos que a caracterizam. Para cada atributo, estão descritos sua semântica e categorias em que se enquadra. Para cada relacionamento, estão descritas as entidades envolvidas e as restrições estabelecidas de cardinalidade e participação.

3.1 ENTIDADES

3.1.1 Cliente

Cliente refere-se a um utilizador do sistema, um cliente pode realizar pagamentos. O tipo de entidade Cliente consiste em uma subclasse que descreve cada instância da entidade Cliente.

Atributo	Descrição	Características	Tipo do Dado
<u>Matrícula</u>	Matrícula é um número de inscrição do cliente	Simples, armazenado e monovalorado	INT(10) NOT NULL
Nome	Nome do cliente	Simples, armazenado e monovalorado	VARCHAR(100)
Telefone	Número de telefone do cliente	Simples, armazenado e monovalorado	INT(12)

3.1.2 Funcionário

Funcionário refere-se ao indivíduo contratado ou horista empregado pelo detentor dos recursos. O tipo de entidade Funcionário consiste em uma subclasse que descreve cada instância da entidade Funcionário.

Atributo	Descrição	Características	Tipo do Dado
<u>NSS</u>	NSS é o número de seguridade social do funcionário	Simples, armazenado e monovalorado	INT(8) NOT NULL
Nome	Nome do funcionário	Simples, armazenado e monovalorado	VARCHAR(100) NOT NULL
Email	Email correspondente ao funcionário	Simples, armazenado e monovalorado	VARCHAR(25) NOT NULL
DataNasc	Data de Nascimento do funcionário	Simples, armazenado e monovalorado	DATE NOT NULL
Senha	Senha do funcionário para acesso ao sistema	Simples, armazenado e monovalorado	VARCHAR(25) NOT NULL
CPF	Cadastro de pessoa fisica unico para cada funcionário	Simples, armazenado e monovalorado	VARCHAR(14) NOT NULL

3.1.3 Horista

Horista refere-se ao indivíduo que possui escala. Suas horas trabalhadas são contabilizadas. O tipo de entidade Horista consiste em uma subclasse derivada da superclasse Funcionário que descreve cada instância de Horista.

Atributo	Descrição	Características	Tipo do Dado
Horas_Trab	Determina as horas trabalhadas do horista	Simples, armazenado e monovalorado	INT(8)
Escala	Escala que o horista terá ao longo da semana	Simples, armazenado e monovalorado	VARCHAR(10)
Valor/Hora	Valor da hora de trabalho	Simples, armazenado e monovalorado	DOUBLE

3.1.4 Contratado

Contratado refere-se ao indivíduo que possui uma matrícula, turno e um cargo específico. O tipo de entidade Contratado consiste em uma subclasse derivada da superclasse Funcionário que descreve cada instância de Contratado.

Atributo	Descrição	Características	Tipo do Dado
Matrícula	Matrícula do funcionário contratado	Simples, armazenado e monovalorado	INT(10) NOT NULL
Turno	Turnos do dia: Matutino, vespertino e/ou noturno	Simples, armazenado e monovalorado	VARCHAR (20)
Cargo	Define o cargo do funcionário contratado	Simples, armazenado e monovalorado	VARCHAR (25)
Salário	Salário que corresponde ao funcionário	Simples, armazenado e monovalorado	DOUBLE

3.1.5 Recurso

Recurso refere-se às quadras existentes para agendamento. A entidade apresenta um código que é único para cada recurso, se a quadra é coberta ou não, o nome da quadra e o valor/hora. Essa entidade é necessária para que haja especificidade de como o aluguel da quadra será.

Atributo	Descrição	Características	Tipo do Dado
<u>Código</u>	Código de cada recurso que existe	Simples, armazenado e monovalorado	INT(10) NOT NULL
E_Coberta	Especifica se a quadra será coberta ou não	Simples, armazenado e monovalorado	BOOL
Nome	Declara o nome da quadra	Simples, armazenado e monovalorado	VARCHAR(60) NOT NULL
Valor/hora	Declara o valor da quadra (preço), dependendo do número de horas que se queira alugar.	Simples, armazenado e multivalorado	DOUBLE

3.1.6 Pagamento Locação

Pagamento Locação refere-se ao pagamento efetuado pelo cliente, quando um agendamento de quadra é feito. A entidade Pagamento, apresenta atributos necessários para que se obtenha controle dos pagamentos efetuados.

Atributo	Descrição	Características	Tipo do Dado
<u>Código</u>	Cada código está referente a um determinado	Simples, armazenado e monovalorado	INT(10) NOT NULL

	pagamento de locação de quadra		
Hora_Pagamento	Horário em que o pagamento foi efetuado	Simples, armazenado e monovalorado	TIME
Valor Bruto	Valor bruto do pagamento	Simples, armazenado e monovalorado	DOUBLE NOT NULL
Valor Líquido	Valor líquido do pagamento	Simples, derivado e monovalorado	DOUBLE
Taxa	Taxa referente ao valor, caso entre em atraso	Simples e armazenado e monovalorado	DOUBLE
Descontos	Descontos aos usuários	Simples e armazenado e monovalorado	DOUBLE

3.1.7 Pagamento em cartão

Pagamento Cartão refere-se ao pagamento da locação realizado através do cartão. A entidade Pagamento Cartão é uma especialização disjunta da superclasse Pagamento Locação.

Atributo	Descrição	Características	Tipo do Dado
Juros	Juros referente ao cartão, caso o pagamento fique em atraso	Simples, armazenado e monovalorado	DOUBLE
Número	Número do Cartão	Simples, armazenado e monovalorado	INT(25)

Parcelas	Número de parcelas referente ao valor que o Usuário venha a fazer	Simples, armazenado e monovalorado	INT(24)
Bandeira	Bandeira é a marca do cartão	Simples, armazenado e monovalorado	CHAR(30)
Nome	Nome escrito no cartão	Simples, armazenado e monovalorado	VARCHAR(30)
CVV	Valor verificador do cartão	Simples, armazenado e monovalorado	INT(3)

3.1.8 Boleto

Uma forma de ser efetuado o pagamento é pelo boleto, sendo possível para o cliente emitir o mesmo e conseguir pagar a locação. Pode ser rastreado pelo código e controlado pelas datas de emissão e vencimento.

Atributo	Descrição	Características	Tipo do Dado
Código	O código referente a cada boleto	Simples, armazenado, monovalorado	INT(15)
Data_Emissao	Data que o boleto foi emitido	Simples, armazenado, monovalorado	DATE NOT NULL
Data_Venc	Data de vencimento do boleto	Simples, armazenado, monovalorado	DATE NOT NULL

3.1.9 Tipo do Esporte

Como forma de especificar o tipo do esporte que o cliente optará por utilizar ao alugar a quadra, há o código do tipo de esporte, a modalidade e a descrição do esporte.

Atributo	Descrição	Características	Tipo do Dado
----------	-----------	-----------------	--------------

<u>Código</u>	O código referente a cada esporte	Simples, armazenado, monovalorado	INT(12)
Modalidade	Modalidade do esporte, o tipo de esporte	Simples, armazenado, monovalorado	VARCHAR(15)
Descrição	Descrição do esporte, detalhamento sobre a modalidade	Simples, armazenado, monovalorado	VARCHAR(30)

3.1.10 Endereço

Endereço dos funcionários, contendo informações como rua, número, bairro e as especificações necessárias de cada um. Esse tipo de entidade possui como chave primária Número ID + CEP.

Atributo	Descrição	Características	Tipo do Dado
<u>Número_ID</u>	O número ID de um determinado endereço	Simples, armazenado, monovalorado	INT(12) NOT NULL
Rua	Rua que reside o funcionário	Simples, armazenado, monovalorado	VARCHAR(100) NOT NULL
Bairro	Bairro que reside o funcionário	Simples, armazenado, monovalorado	VARCHAR(35) NOT NULL
Complemento	Complemento adicional para especificar o endereço do funcionário (bloco, apto, ala, vila...)	Simples, armazenado, monovalorado	VARCHAR(35)
Número	Número de onde mora o funcionário	Simples, armazenado, monovalorado	INT(5) NOT NULL

<u>CEP</u>	CEP de onde reside o funcionário	Simples, armazenado, monovalorado	INT(8) NOT NULL
------------	----------------------------------	-----------------------------------	--------------------

3.2 RELACIONAMENTOS

3.2.1 Reside

O relacionamento Reside descreve a relação entre o tipo de entidade Funcionário e o tipo de entidade Endereço, pelo qual, todo funcionário reside em pelo menos um endereço, sendo assim possível que um funcionário tenha mais que um endereço e um mesmo endereço pode pertencer a mais de um funcionário.

Entidade	Restrição de Cardinalidade	Restrição de Participação
Funcionário	N	Total
Endereço	M	Total

3.2.2 Agendamento

O relacionamento Agendamento ocorre entre Funcionário, Cliente, Recurso e Pagamento, formando assim, um relacionamento quaternário. Para esse tipo de relacionamento é necessário que todas entidades estejam relacionadas para que seja efetuado o agendamento com sucesso.

Entidade	Restrição de Cardinalidade	Restrição de Participação
Funcionário	N	Total
Cliente	M	Total
Recurso	P	Total
Pagamento_Locação	1	Total

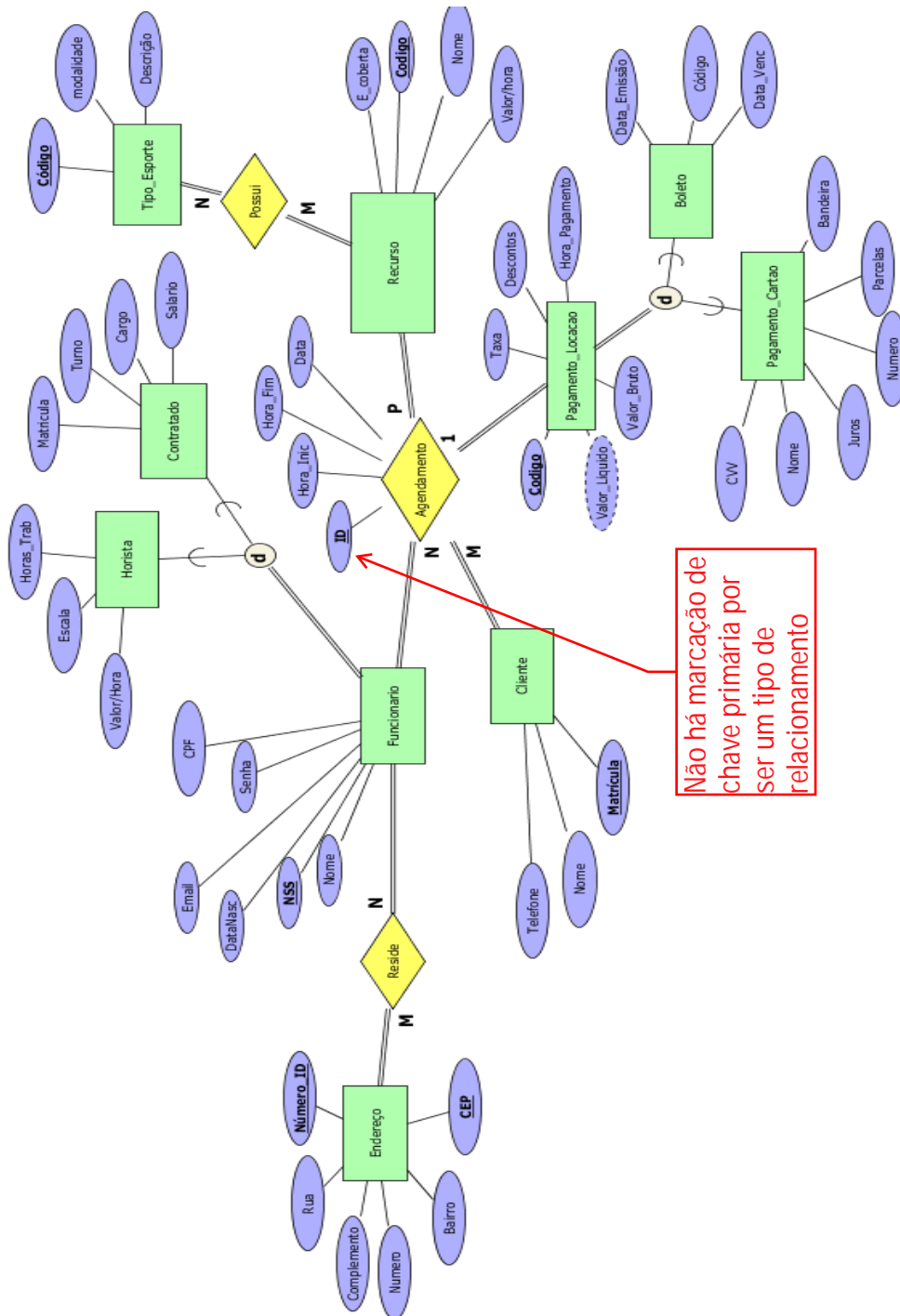
3.2.3 Possui

O relacionamento Possui ocorre entre as entidades de Tipo_Esporte e Recurso. Sendo descrito como, um tipo de esporte ou modalidade esportiva pode ser feita em várias quadras (Recurso), e uma quadra (Recurso) pode abrigar vários tipos de esporte.

Entidade	Restrição de Cardinalidade	Restrição de Participação
Tipo_Esporte	N	Total
Recurso	M	Total

4.0 MODELO CONCEITUAL

3,0 / 3,0



5.0 MAPEAMENTO DO ERE PARA O RELACIONAL

Recurso (**Código**, E_coberta, Nome, Valor/hora)

Tipo_Esporte (**Código**, modalidade, Descrição)

Cliente(**Matricula**, Nome, Telefone)

Endereço (**Número_ID**, **CEP**, Rua, Complemento, Numero, Bairro)

Funcionario (**NSS**, Nome, DataNasc, Email, Senha, CPF)

Contratado (**NSS_Funcionario**, Matrícula, Turno, Cargo, Salario)

Contratado [NSS_Funcionario] → {p} Funcionario [NSS]

Horista (**NSS_Funcionario**, Escala, Horas_Trab, Valor/Hora)

Horista [NSS_Funcionario] → {p} Funcionario [NSS]

Pagamento_Locação (**Código**, Valor_Bruto, Taxa, Descontos, Hora_Pagamento,)

Boleto (**Código_Pagamento**, Data_Emissão, Código, Data_Venc)

Boleto [Código_Pagamento] → {b} Pagamento_Locação [Codigo]

Pagamento_Cartao (**Código_Pagamento**, Juros, Numero, Parcelas, Bandeira, Nome, CVV)

Pagamento_Cartao [Código_Pagamento] → {b} Pagamento_Locação [Codigo]

Reside (**Número_IDReside**, **CEP_Reside**, **NSS_Funcionario**)

Reside [Número_IDReside] → {p} Endereço [Número_ID]

Reside [CEP_Reside] → {p} Endereço [CEP]

Reside [NSS_Funcionario] → {p} Funcionario [NSS]

Uma só restrição referencial de integridade já que as chaves estrangeira e primária vinculadas são compostas

Possui (**Código_TipoEsporte**, **Código_Recurso**)

Possui [Código_TipoEsporte] → {b} Tipo_Esporte [Código]

Possui [Código_Recurso] → {b} Recurso [Codigo]

Agendamento (**Agendamento_ID**, NSS_Funcionario, Matrícula_Cliente, Codigo_Recurso, Codigo_PagamentoLocacao, Data, Hora_Inic, Hora_Fim)

Agendamento [NSS_Funcionario] → {b} Funcionario [NSS]

Agendamento [Matrícula_Cliente] → {b} Cliente [Matrícula]
Agendamento [Codigo_Recurso] → {b} Recurso [Código]
Agendamento [Codigo_PagamentoLocacao] → {b} Pagamento_Locacao
[Codigo]

Item 3: 6,0 / 6,0
(considerando o relacional correto)

6.0 SCRIPT

CREATE TABLE Recurso

```
(    Codigo      INT(10)      NOT NULL,
    E_coberta    BOOL,
    Nome         VARCHAR(60)   NOT NULL,
    Valor/hora   DOUBLE,
    CONSTRAINT PK_Recurso PRIMARY KEY (Codigo)
);
```

CREATE TABLE Tipo_Esporte

```
(    Codigo      INT(10)      NOT NULL,
    Modalidade   VARCHAR(30),
    Descricao    VARCHAR(30),
    CONSTRAINT PK_Tipo_Esporte PRIMARY KEY (Codigo)
);
```

CREATE TABLE Cliente

```
(    Matricula   INT(10)      NOT NULL,
    Nome         VARCHAR(100),
    Telefone     INT (12),
    CONSTRAINT PK_Cliente PRIMARY KEY (Matricula)
);
```

CREATE TABLE Endereco

```
(    Numero_ID   INT(12)      NOT NULL,
```



```

        CEP            INT(8)            NOT NULL,
        Rua            VARCHAR(100)       NOT NULL,
        Numero         INT(5)            NOT NULL,
        Bairro         VARCHAR(35)       NOT NULL,
        Complemento    VARCHAR(35),
        CONSTRAINT PK_Endereco PRIMARY KEY (Numero_ID, CEP)
) AUTO_INCREMENT = 1;

CREATE TABLE Funcionario
(
    NSS            INT(8) NOT NULL,
    Nome          VARCHAR(100) NOT NULL,
    DataNasc      DATE NOT NULL,
    Email         VARCHAR(25) NOT NULL,
    Senha         VARCHAR(25) NOT NULL,
    CPF           VARCHAR(14) NOT NULL,
    CONSTRAINT PK_Funcionario PRIMARY KEY (NSS)
);

CREATE TABLE Horista
(
    NSS_Funcionario  INT(8) NOT NULL,
    Escala           VARCHAR(10),
    Horas_Trab       INT(8),
    Valor/Hora       DOUBLE,
    CONSTRAINT FK_Func FOREIGN KEY (NSS_Funcionario)
        REFERENCES Funcionario (NSS) ON DELETE CASCADE ON UPDATE
        CASCADE
);

CREATE TABLE Contratado
(
    NSS_Funcionario  INT(8) NOT NULL,
    Matricula        INT(10) NOT NULL,
    Turno            VARCHAR(25),

```

```

        Cargo          VARCHAR(25)      NOT NULL,
        Salario         DOUBLE,
        CONSTRAINT FK_Funcionario FOREIGN KEY (NSS_Funcionario)
        REFERENCES Funcionario (NSS) ON DELETE CASCADE ON UPDATE
        CASCADE
    );

```

```

CREATE TABLE Pagamento_Locacao
(
    Codigo          INT(10)      NOT NULL,
    Valor_Bruto     DOUBLE      NOT NULL,
    Valor_Liquido   DOUBLE      NOT NULL
    Taxa            DOUBLE,
    Descontos       DOUBLE,
    Hora_Pagamento TIME,
    CONSTRAINT PK_Pagamento_Locacao PRIMARY KEY (Codigo)
);

```

```

CREATE TABLE Boleto
(
    Codigo_Pagamento INT(10)      NOT NULL,
    Data_Emissao      DATE NOT NULL,
    Codigo             INT(15)      NOT NULL,
    Data_Venc          DATE NOT NULL,
    CONSTRAINT FK_Pag FOREIGN KEY (Codigo_Pagamento)
    REFERENCES Pagamento_Locacao (codigo) ON DELETE RESTRICT
);

```

```

CREATE TABLE Pagamento_Cartao
(
    Codigo_Pagamento INT(10)      NOT NULL,
    Juros             DOUBLE,
    Numero            INT(16)      NOT NULL,

```

```

Parcelas          INT,
Bandeira          CHAR(30)    NOT NULL,
Nome              VARCHAR(100) NOT NULL,
CVV               INT(3)      NOT NULL,
CONSTRAINT FK_Pagamento FOREIGN KEY (Codigo_Pagamento)
REFERENCES Pagamento_Locacao (codigo) ON DELETE RESTRICT
);

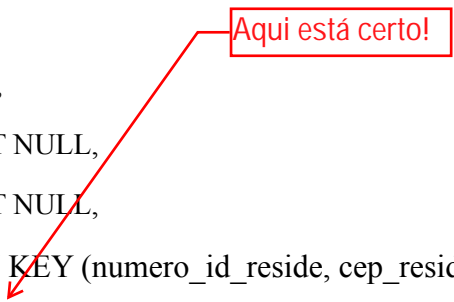
```

```

CREATE TABLE Reside
(
    Numero_id_Reside    INT    NOT NULL,
    CEP_Reside          INT (8)    NOT NULL,
    NSS_Funcionario     INT(8)    NOT NULL,
    CONSTRAINT FK_Endereco FOREIGN KEY (numero_id_reside, cep_reside)
    REFERENCES Endereco (numero_id, cep) ON DELETE CASCADE ON
UPDATE CASCADE,
    CONSTRAINT FK_Endereco_Func FOREIGN KEY (nss_funcionario)
    REFERENCES Funcionario(nss) ON DELETE CASCADE ON UPDATE
    CASCADE
);

```

Aqui está certo!



```

CREATE TABLE Possui
(
    Código_TipoEsporte  INT (10)    NOT NULL,
    Código_Recurso      INT(10)    NOT NULL,
    CONSTRAINT FK_Tipo_Esporte FOREIGN KEY (codigo_tipo_esporte)
    REFERENCES Tipo_Esporte(codigo) ON DELETE RESTRICT ON UPDATE
    CASCADE,
    CONSTRAINT FK_Recurso FOREIGN KEY (codigo_recurso)
    REFERENCES Recurso(codigo) ON DELETE RESTRICT ON UPDATE
    CASCADE,
);

```

CREATE TABLE Agendamento

```
(    NSS_Funcionario      INT(8)      NOT NULL,
    Matrícula_Cliente    INT(10)     NOT NULL,
    Codigo_Recurso       INT(10)     NOT NULL,
    Codigo_Pag           INT(10)     NOT NULL,
    Agendamento_ID       INT(10)     NOT NULL    AUTO_INCREMENT,
    Data                 DATE         NOT NULL,
    Hora_Inic            BIGINT(8),
    Hora_Fim            BIGINT(8),

    CONSTRAINT PK_Agendamento PRIMARY KEY (Agendamento_ID),
    CONSTRAINT FK_Funcionario FOREIGN KEY (NSS_Funcionario)
        REFERENCES Funcionario (NSS) ON DELETE RESTRICT ON UPDATE
        CASCADE,
    CONSTRAINT FK_Cliente FOREIGN KEY (Matrícula_Cliente)
        REFERENCES Cliente (Matricula) ON DELETE RESTRICT ON UPDATE
        CASCADE,
    CONSTRAINT FK_Recurso FOREIGN KEY (Codigo_Recurso)
        REFERENCES Recurso (Código) ON DELETE RESTRICT ON UPDATE
        CASCADE,
    CONSTRAINT FK_Pagamento_Locacao FOREIGN KEY
        (Codigo_Pagamento_Locacao)
        REFERENCES Pagamento_Locacao (Codigo) ON DELETE RESTRICT ON
        UPDATE CASCADE,
    )AUTO_INCREMENT = 1;
```

```
INSERT INTO Recurso
VALUES(1, TRUE, 'Quadra Society', 80);
INSERT INTO Recurso
VALUES(2, FALSE, 'Quadra de Areia', 70);
INSERT INTO Recurso
VALUES(3, FALSE, 'Quadra Vertical Areia', 70);
INSERT INTO Recurso
VALUES(4, FALSE, 'Quadra Society 2', 80);
INSERT INTO Recurso
VALUES(5, TRUE, 'Quadra Grama ', 90);
```

```
INSERT INTO Tipo_Esporte
VALUES(1, Futsal, Futebol de campo);
INSERT INTO Tipo_Esporte
VALUES(2, Volei, Volei de Areia);
INSERT INTO Tipo_Esporte
VALUES(3, Basquete, Basquete de quadra polida);
INSERT INTO Tipo_Esporte
VALUES(4, Dodgeball, Queimada de quadra);
INSERT INTO Tipo_Esporte
VALUES(5, Handball, Handball de quadra);
```

```
INSERT INTO Cliente
VALUES(122, 'Robesvaldo Silva', 31746574587);
INSERT INTO Cliente
VALUES(123, 'Marcio Pafuncio', 31723474532);
INSERT INTO Cliente
VALUES(124, 'Ana Toledo', 3135369537);
```

INSERT INTO Cliente

VALUES(125, 'Valério Silva', 316354739);

INSERT INTO Cliente

VALUES(126, 'Carla Furtado', 3172658237);

INSERT INTO Endereco

VALUES (-----, 30575180, 'Av. Professor Mario Werneck', 2241, 'Buritis', 'loja');

INSERT INTO Endereco

VALUES (-----, 30431214, 'Rua Xapuri', 785, 'Grajau', 'casa');

INSERT INTO Endereco

VALUES (-----, 30150370, 'Rua Professor Moraes', 624, 'Funcionarios',
'edificio');

INSERT INTO Endereco

VALUES (-----, 31310460, 'Rua Candido Naves', 215, 'Ouro Preto', 'casa');

INSERT INTO Endereco

VALUES (-----, 31320450, 'Rua Engenho Vermelho', 26, 'Engenho Nogueira',
'casa');

INSERT INTO Funcionario

VALUES (10345823, 'Thales Augusto', '1998-05-29', 'thalesguedes98@gmail.com',
'1234senha000', '107.523.546-47', 1500.50);

INSERT INTO Funcionario

VALUES (10345555, 'Brenda Ferreira', '1996-09-09', 'brenda@gmail.com', '1234senha',
'111.234.546-97', 2000.50);

INSERT INTO Funcionario

VALUES (49017482, 'Gabriel Augusto', '1999-10-10', 'gabriel@gmail.com',
'98765gabriel', '456.555.098-56', 2590.76);

INSERT INTO Funcionario

VALUES (49017619, 'Julia Siqueira', '1999-02-05', 'julia@gmail.com', '1234julia',
'411.135.908-11', 1967.34);

INSERT INTO Funcionario

VALUES (49019182, 'Karina Almeida', '1997-04-10', 'karina@gmail.com', '1357karina',
'687.132.038-72', 1600.45);

INSERT INTO Horista

VALUES (49017619, _, 35, 65);

INSERT INTO Horista

VALUES (10345555, 12/24, 15, 65);

INSERT INTO Contratado

VALUES (49017482, 10824, 'Matutino', 'limpeza', 2590.5);

INSERT INTO Contratado

VALUES (49019182, 19482, 'Vespertino', 'gerente', 1600.45);

INSERT INTO Contratado

VALUES (10345823, 74926, 'Vespertino', 'secretario', 1500.50);

INSERT INTO Pagamento_Locacao

VALUES (88910, 300.0, 150.0, 100.0, 50.0, 21:45:21);

INSERT INTO Pagamento_Locacao

VALUES (38910, 250.0, 150.0, 50.0, 50.0, 09:58:34);

INSERT INTO Pagamento_Locacao

VALUES (45829, 180.0, 100.0, 50.0, 30.0, 13:59:51);

INSERT INTO Pagamento_Locacao

VALUES (34571, 350.0, 200.0, 100.0, 50.0, 22:03:47);

INSERT INTO Pagamento_Locacao

VALUES (44465, 300.0, 150.0, 100.0, 50.0, 21:01:58);

```
INSERT INTO Boleto
VALUES (88910, '2021-08-02', 019364728, '2021-08-04');
INSERT INTO Boleto
VALUES (38910, '2021-08-10', 3591037582, '2021-08-13');
INSERT INTO Boleto
VALUES (45829, '2021-10-21', 3728550017, '2021-10-22');
```

```
INSERT INTO Pagamento_Cartao
VALUES (34571, 24.45, 132412339, 5, 'mastercard', 'Marcio Pafuncio', 988);
INSERT INTO Pagamento_Cartao
VALUES (44465, 32.11, 278934560, 7, 'visa', 'Valério Silva', 023);
```

```
INSERT INTO Reside
VALUES ( -----, 30575180, 10345823);
INSERT INTO Reside
VALUES ( -----, 30431214, 10345555);
INSERT INTO Reside
VALUES ( -----, 30150370, 49017482);
INSERT INTO Reside
VALUES ( -----, 31310460, 49017619);
INSERT INTO Reside
VALUES ( -----, 31320450, 49059182);
```

```
INSERT INTO Possui
VALUES(1, 1);
INSERT INTO Possui
VALUES(2, 2);
INSERT INTO Possui
```


VALUES(3, 3);

INSERT INTO Possui

VALUES(4, 4);

INSERT INTO Possui

VALUES(5, 5);

INSERT INTO Agendamento

VALUES(10345823, 122, 1, 88910, '2012-01-02', 15, 17);

INSERT INTO Agendamento

VALUES(49017482, 123, 2, 38910, '2011-04-05', 12, 14);

INSERT INTO Agendamento

VALUES(49017619, 124, 3, 45829, '2004-01-12', 07, 10);

INSERT INTO Agendamento

VALUES(49019182, 125, 4, 44465, '2002-11-03', 06, 10);

INSERT INTO Agendamento

VALUES(10345555, 126, 5, 34571, '2006-02-15', 12, 16);

Item 4: 0,0 / 6,0
(não realizado)