

UNIVERSIDADE FEDERAL DE LAVRAS
DEPARTAMENTO DE CIÊNCIAS DA COMPUTAÇÃO
INTRODUÇÃO A SISTEMAS DE BANCO DE DADOS

AYRON SANFRA SILVA MARINHO
CAIO ALMEIDA SANTOS
GABRIEL DE PAULO COSTA
IGOR ABREU FORTUNATO
PEDRO HENRIQUE SANTANA QUINTILIANO

Trabalho Prático – Etapa 3

LAVRAS - MG
2023

AYRON SANFRA SILVA MARINHO
CAIO ALMEIDA SANTOS
GABRIEL DE PAULO COSTA
IGOR ABREU FORTUNATO
PEDRO HENRIQUE SANTANA QUINTILIANO

Trabalho Prático – Etapa 3

Relatório apresentado ao curso de
Introdução a Sistemas de Banco de Dados
como requisito parcial para aprovação na
disciplina.

Orientado pelo professor Denilson Alves
Pereira.

LAVRAS - MG

2023

Descrição:

O problema que está em pauta para ser resolvido é o gerenciamento, organização e disponibilização de informações de monitorias. Atualmente não temos de fácil acesso, informações de Monitorias das disciplinas da faculdade, visto que geralmente a informação chega até um estudante através de comunicados de professores ou informação passada de estudante para estudante, então a ideia é centralizar todo tipo de informação em um único lugar, e no futuro criar um aplicativo para disponibilização de funcionalidades a respeito dos problemas. Para isso, foram levantados os seguintes requisitos:

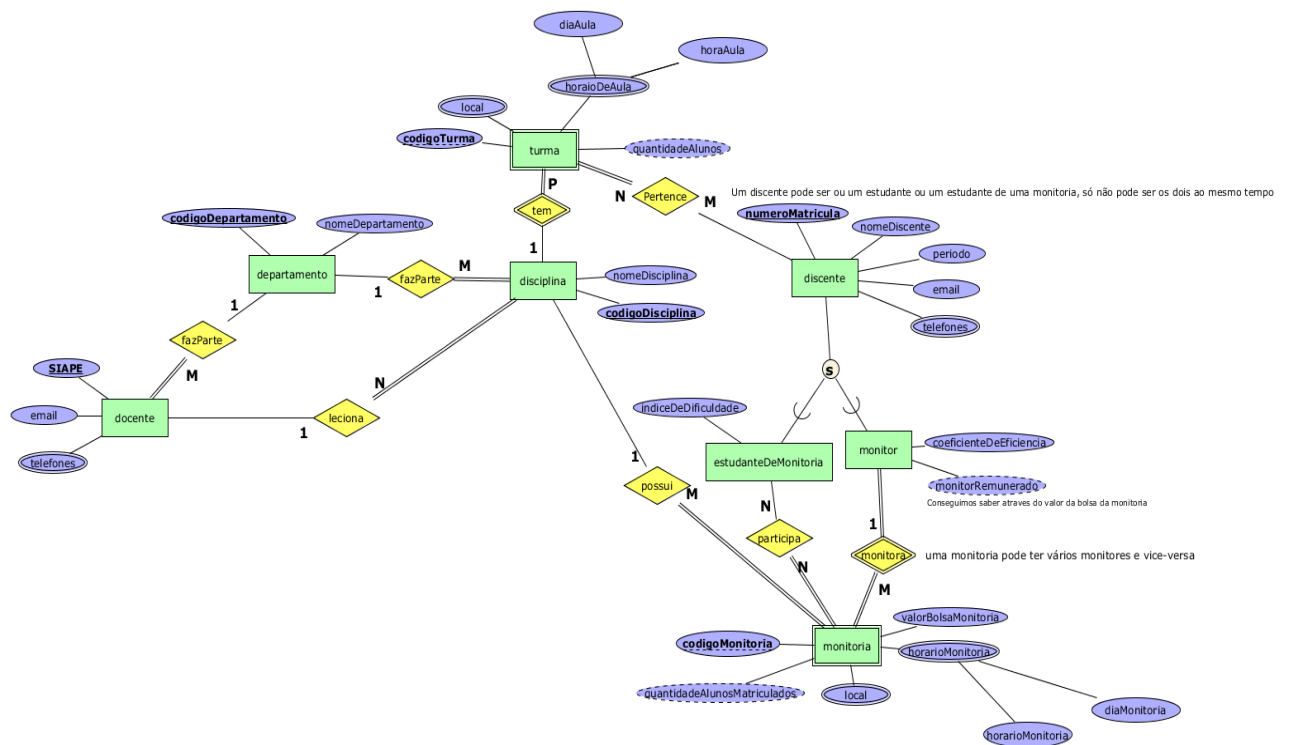
Um professor (Docente) possui as seguintes informações: email, nome, telefones, SIAPE (código identificador do servidor público) e faz parte de um departamento. É responsável por várias disciplinas, que são compostas por nome da disciplina, código da disciplina, e o departamento que ela faz parte.

Para toda disciplina, pode ou não existir uma turma, mas toda turma é de uma disciplina, uma turma pode ter uma ou mais disciplinas, o professor só conseguirá lecionar uma disciplina para um discente, caso a disciplina tenha alguma turma e se possui algum aluno matriculado na disciplina. A disciplina pode ou não ter uma monitoria.

A turma precisa ter um local, o dia e horário da aula, e a quantidade de alunos (que pode ser obtido através de uma consulta de quantos alunos fazem parte dessa disciplina).

A turma é composta por discentes, que possuem informações do período em que está, nome, email, telefones e matrícula. Um discente pode ser um estudante ou um monitor, o monitor também pode ser um estudante que participa em uma monitoria de outra disciplina, mas nunca os dois ao mesmo tempo para a mesma disciplina. A monitoria pode ou não ser remunerada, e precisa armazenar informações como: valor da bolsa de monitoria (Caso tenha, seria preenchida com um valor maior que zero), dia da semana, horário e quantidade de estudantes matriculados nessa monitoria. Para que sejam medidas as dificuldades do estudante, para saber se precisa de uma atenção especial na monitoria, é necessário armazenar um índice de dificuldade, definidos por: 1 - Pouca dificuldade; 2 - Média dificuldade; 3 - Muita dificuldade. E para medir se o monitor consegue dar uma boa monitoria, precisamos armazenar um coeficiente de eficiência de sua monitoria que seria um valor atualizado por uma média de uma pesquisa realizada pelos professores (De 0% a 100%).

Diagrama ER:



Dicionário de Dados ER:

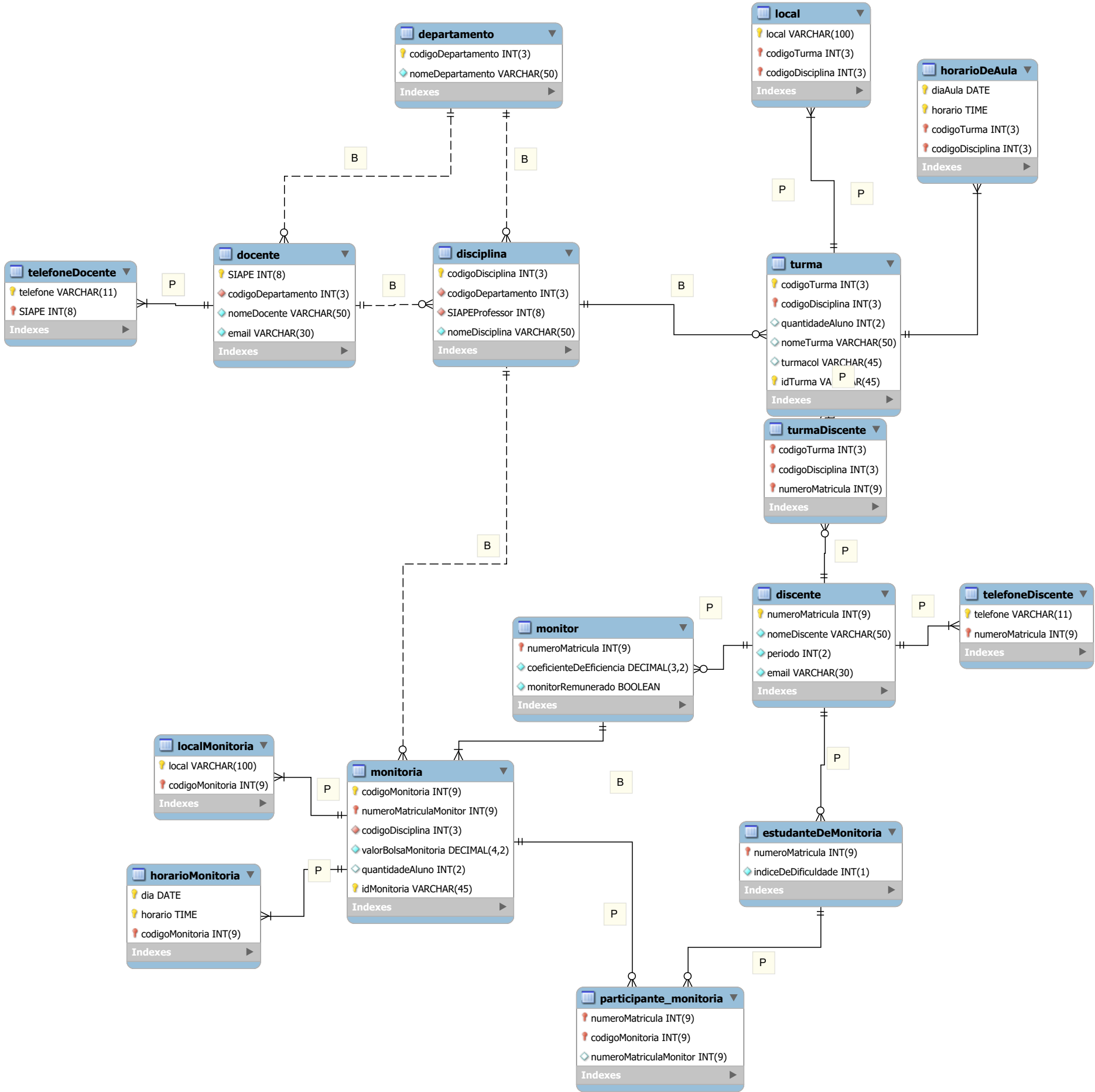
Tipo Entidade	Docente		
Descrição	Conjunto de professores que dão aula para várias disciplinas e faz parte de vários departamentos		
Atributos			
Nome	Descrição	Domínio	Permite nulo? (S/N)
SIAPÉ	Código de identificação do servidor público	Inteiro(8) positivo Formato: 00000000	N
EMAIL	E-mail institucional	Texto(30) Formato: NomeUtilizador@Dominio.com	N
TELEFONES	Telefones de contato	Texto(11) Multivalorado Formato: (00) 0 0000-0000 ou (00) 0000-0000	N
NOME	Nome do Docente	Texto(50)	N
Tipo Entidade	Departamento		
Descrição	Conjunto de departamentos da Instituição		
Atributos			
Nome	Descrição	Domínio	Permite nulo? (S/N)
CODIGODEPARTAMENTO	Código de identificação do Departamento	Inteiro(3) positivo	N
NOMEDEPARTAMENTO	Nome do Departamento	Texto(50)	N
Tipo Entidade	Disciplina		
Descrição	Conjunto de Disciplinas		
Atributos			
Nome	Descrição	Domínio	Permite nulo? (S/N)
CODIGODISCIPLINA	Código de identificação da Disciplina	Inteiro(3) positivo	N
NOMEDISCIPLINA	Nome da Disciplina	Texto(50)	N
Tipo Entidade	Turma		
Descrição	Conjunto de Turmas		
Atributos			
Nome	Descrição	Domínio	Permite nulo? (S/N)
CODIGOTURMA	Código de identificação da Turma	Inteiro(3) positivo	N
NOMETURMA	Nome da Turma	Texto(50)	N
LOCAL	Locais onde serão as aulas	Texto(100)	N
DIAAULA	Dia em que ocorrerá a aula	Data	N

HORARIOAULA	Horário em que ocorrerá a aula	Tempo	N
QUANTIDADEALUNOS	Quantidade de Alunos que pertencem a turma	Inteiro(2) positivo	N
Tipo Entidade	Discente		
Descrição	Conjunto de alunos que cursam na instituição		
Atributos			
Nome	Descrição	Domínio	Permite nulo? (S/N)
NUMEROMATRICULA	Código de identificação do Discente	Inteiro(9) positivo Formato: 000000000	N
NOMEDISCENTE	Nome do Aluno	Texto(50)	N
PERIODO	Periodo da graduação	Inteiro(2) positivo	N
EMAIL	Email do aluno	Texto(30) Formato: NomeUtilizador@Dominio.com	N
TELEFONES	Telefones de contato	Texto(11) Multivalorado Formato: (00) 0 0000-0000 ou (00) 0000-0000	N
Tipo Entidade	estudanteDeMonitoria		
Descrição	Conjunto de estudantes que participam de uma monitoria		
Atributos			
Nome	Descrição	Domínio	Permite nulo? (S/N)
INDICEDEDIFICULDADE	Indicador de dificuldade do estudante da monitoria	Inteiro(1) positivo 1 - Pouca dificuldade; 2 - Média dificuldade; 3 - Muita dificuldade.	N
Tipo Entidade	Monitor		
Descrição	Conjunto de docentes que lecionam uma monitoria		
Atributos			
Nome	Descrição	Domínio	Permite nulo? (S/N)
COEFICIENTEDEEFICIENCIA	Percentual de eficiencia do monitor em relação as suas monitorias	Decimal(3,2) positivo	N
MONITORREMUNERADO	Indicativo se o monitor possui remuneração	Booleano	N
Tipo Entidade	Monitoria		
Descrição	Conjunto de monitorias que são realizadas para as Disciplinas		
Atributos			
Nome	Descrição	Domínio	Permite nulo? (S/N)

CODIGOMONITORIA	Código de identificação da monitoria - Chave parcial	Inteiro(9) positivo Chave parcial	N
VALORBOLSAMONITORIA	Valor da bolsa da monitoria	Decimal(4,2) positivo	N
LOCAL	Locais onde serão as monitorias	Texto(100) Multavalorado	N
DIAMONITORIA	Dia em que ocorrerá a monitoria	Data	N
HORARIOMONITORIA	Horário em que ocorrerá a monitoria	Tempo	N
QUANTIDADEALUNOSMATRICULADOS	Quantidade de estudantes que fazem monitoria	Inteiro(2) positivo	N
Tipo Relacionamento	Tem		
Descrição	Relacionamento de pertencimento um para muitos entre turma e disciplina		
Atributos			
Nome	Descrição	Domínio	Permite nulo? (S/N)
Tipo Relacionamento	Pertence		
Descrição	Relacionamento de pertencimento entre turma e discente, muitos para muitos		
Atributos			
Nome	Descrição	Domínio	Permite nulo? (S/N)
Tipo Relacionamento	fazParte		
Descrição	Relacionamento um para muitos entre disciplina e departamento, onde uma disciplina faz parte de um departamento		
Atributos			
Nome	Descrição	Domínio	Permite nulo? (S/N)
Tipo Relacionamento	Pertence		
Descrição	Relacionamento de pertencimento entre turma e discente, muitos para muitos, onde muitas turmas possuem muitos discentes		
Atributos			
Nome	Descrição	Domínio	Permite nulo? (S/N)

Tipo Relacionamento	fazParte		
Descrição	Relacionamento um para muitos entre doscente e departamento, onde um doscente faz parte de muitos departamentos		
Atributos			
Nome	Descrição	Domínio	Permite nulo? (S/N)
Tipo Relacionamento	leciona		
Descrição	Relacionamento um para muitos entre doscente e disciplina, onde um doscente leciona muitas disciplinas		
Atributos			
Nome	Descrição	Domínio	Permite nulo? (S/N)
Tipo Relacionamento	participa		
Descrição	Relacionamento muitos para muitos entre estudante e monitoria, onde muitos estudantes participam de muitas monitorias		
Atributos			
Nome	Descrição	Domínio	Permite nulo? (S/N)
Tipo Relacionamento	monitora		
Descrição	Relacionamento um para muitos entre monitor e monitoria, onde um monitor monitora uma monitoria		
Atributos			
Nome	Descrição	Domínio	Permite nulo? (S/N)

Modelo Relacional



Dicionário de Dados Modelo Relacional

Tabela	departamento
Descrição	Conjunto de departamentos da Instituição
Atributos	
Nome	Descrição
codigoDepartamento	Código de identificação do departamento
nomeDepartamento	Nome do departamento
Tabela	docente
Descrição	Conjunto de professores que dão aula para várias disciplinas e fazem parte de vários departamentos
Atributos	
Nome	Descrição
SIAPE	Código de identificação do servidor público
codigoDepartamento	Código de identificação do departamento ao qual o servidor é relacionado
nomeDocente	Nome do docente
email	Email do docente
Tabela	telefoneDocente
Descrição	Telefones relacionados aos docentes da tabela docentes
Atributos	
Nome	Descrição
telefone	Número de telefone
SIAPE	Código de identificação do servidor público da tabela docentes
Tabela	disciplina
Descrição	Conjunto de Disciplinas
Atributos	
Nome	Descrição
codigoDisciplina	Código de identificação da Disciplina
codigoDepartamento	Referencia do departamento ao qual a disciplina pertence

Dicionário de Dados Modelo Relacional

SIAPEProfessor	Codigo de identificação do professor que leciona a disciplina
nomeDisciplina	Nome da disciplina
Tabela	turma
Descrição	Conjunto de Turmas
Atributos	
Nome	Descrição
codigoTurma	Código de identificação da turma
codigoDisciplina	Referencia a disciplina a qual a turma pertence
quantidadeAluno	Quantidade de alunos matriculados na turma
Tabela	local
Descrição	Atributo multivalorado da tabela turma, indica os locais de aula (sala) de cada turma
Atributos	
Nome	Descrição
local	Local da aula
codigoTurma	Referencia ao codigo de identificação da turma
codigoDisciplina	Referencia ao codigo de identificação da disciplina
Tabela	horarioDeAula
Descrição	Horarios de aula de cada turma
Atributos	
Nome	Descrição
diaAula	Dia da semana (segunda a sábado) em que ocorre a aula
horario	Horário em que ocorre a aula
codigoTurma	Referencia ao código de identificação da turma
codigoDisciplina	Referencia ao codigo de identificação da disciplina
Tabela	turmaDiscente
Descrição	Relacionamento m..m entre as tabelas discente e turma

Dicionário de Dados Modelo Relacional

Atributos	
Nome	Descrição
codigoTurma	Referencia ao código de identificação da turma
codigoDisciplina	Referencia ao código de identificação da disciplina
numeroMatricula	Referencia ao número de matrícula do discente (aluno)
Tabela	discente
Descrição	Conjunto de alunos que cursam na instituição
Atributos	
Nome	Descrição
numeroMatricula	Numero de matricula do aluno
nomeDiscente	Nome do aluno
periodo	Periodo letivo em que o aluno se encontra
email	Email institucional do aluno
Tabela	telefoneDiscente
Descrição	Numeros de telefone dos alunos
Atributos	
Nome	Descrição
telefone	Numero do telefone
numeroMatricula	Referencia ao numero de matricula da tabela discente
Tabela	monitor
Descrição	Conjunto de docentes que lecionam uma monitoria
Atributos	
Nome	Descrição
numeroMatricula	Referencia ao numero de matricula da tabela discente
coeficienteDeEficiencia	Indicador de eficiencia do monitor em suas atividades
monitorRemunerado	Indicador do tipo da monitoria (Remunerada ou Não Remunerada)

Dicionário de Dados Modelo Relacional

Tabela	localMonitoria
Descrição	Conjunto de locais em que serao prestadas as monitorias
Atributos	
Nome	Descrição
Local	Local em que sera prestado a monitoria para os alunos
codigoMonitoria	Referencia ao codigo de identificao da monitoria
Tabela	horarioMonitoria
Descrição	Conjunto de datas que serao prestadas as monitorias
Atributos	
Nome	Descrição
Dia	Dia da semana (segunda a sábado) em que ocorre a monitoria
Horario	Horário em que ocorre a monitoria
codigoMonitoria	Referencia ao codigo de identificao da monitoria
Tabela	Monitoria
Descrição	Conjunto de monitorias que são realizadas para as Disciplinas
Atributos	
Nome	Descrição
numeroMatriculaMonitor	Referencia ao codigo de identificao do monitor
codigoDisciplina	Referencia ao codigo de identificao da disciplina
valorBolsaMonitoria	Valor que o monitor ira receber por prestar a monitoria
quantidadeAluno	Quantidade de aluno cadastrado na monitoria
codigoMonitoria	Referencia ao codigo de identificao da monitoria
Tabela	estudanteDeMonitoria
Descrição	Conjunto de estudantes que participam de uma monitoria
Atributos	
Nome	Descrição
numeroMatricula	Referencia ao codigo de identificao do estudante

Dicionário de Dados Modelo Relacional

indiceDeDificuldade	Numero que representa a dificuldade do aluno
Tabela	participante_monitoria
Descrição	Relacionamento m..m entre as tabelas estudante e monitoria
Atributos	
Nome	Descrição
numeroMatricula	Referencia ao codigo de identificao do estudante
codigoMonitoria	Referencia ao codigo de identificao da monitoria

```
CREATE SCHEMA `gerenciador_monitoria` DEFAULT CHARACTER SET utf8 ;
USE `gerenciador_monitoria` ;
```

```
-- -----
-- Tabela `departamento`
-- -----
```

```
CREATE TABLE `departamento` (
  `codigoDepartamento` INT NOT NULL AUTO_INCREMENT,
  `nomeDepartamento` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`codigoDepartamento`))
ENGINE = InnoDB;
```

```
-- -----
-- Tabela `docente`
-- -----
```

```
CREATE TABLE `docente` (
  `SIAPE` INT NOT NULL,
  `codigoDepartamento` INT NOT NULL,
  `nomeDocente` VARCHAR(50) NOT NULL,
  `email` VARCHAR(30) NOT NULL,
  PRIMARY KEY (`SIAPE`),
  INDEX `fk_docente_departamento1_idx` (`codigoDepartamento` ASC),
  CONSTRAINT `fk_docente_departamento1`
    FOREIGN KEY (`codigoDepartamento`)
      REFERENCES `departamento` (`codigoDepartamento`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----
-- Tabela `telefoneDocente`
-- -----
```

```
CREATE TABLE `telefoneDocente` (
  `telefone` INT NOT NULL,
  `SIAPE` INT NOT NULL,
  PRIMARY KEY (`telefone`, `SIAPE`),
  INDEX `fk_telefoneDocente_docente_idx` (`SIAPE` ASC),
  CONSTRAINT `fk_telefoneDocente_docente`
    FOREIGN KEY (`SIAPE`)
      REFERENCES `docente` (`SIAPE`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
ALTER TABLE `telefonedocente`
  CHANGE COLUMN `telefone` `telefone` VARCHAR(11) NOT NULL;
```

-- Tabela `disciplina`

```
CREATE TABLE `disciplina` (  
  `codigoDisciplina` INT NOT NULL AUTO_INCREMENT,  
  `codigoDepartamento` INT NOT NULL,  
  `SIAPEProfessor` INT NOT NULL,  
  `nomeDisciplina` VARCHAR(50) NOT NULL,  
  PRIMARY KEY (`codigoDisciplina`),  
  INDEX `fk_disciplina_departamento1_idx` (`codigoDepartamento` ASC),  
  INDEX `fk_disciplina_docente1_idx` (`SIAPEProfessor` ASC),  
  CONSTRAINT `fk_disciplina_departamento1`  
    FOREIGN KEY (`codigoDepartamento`)  
      REFERENCES `departamento` (`codigoDepartamento`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_disciplina_docente1`  
    FOREIGN KEY (`SIAPEProfessor`)  
      REFERENCES `docente` (`SIAPE`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Tabela `turma`

```
CREATE TABLE `turma` (  
  `codigoTurma` INT NOT NULL AUTO_INCREMENT,  
  `codigoDisciplina` INT NOT NULL,  
  `quantidadeAluno` INT NULL,  
  PRIMARY KEY (`codigoTurma`, `codigoDisciplina`),  
  INDEX `fk_turma_disciplina1_idx` (`codigoDisciplina` ASC),  
  CONSTRAINT `fk_turma_disciplina1`  
    FOREIGN KEY (`codigoDisciplina`)  
      REFERENCES `disciplina` (`codigoDisciplina`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
ALTER TABLE `turma`  
  CHANGE COLUMN `quantidadeAluno` `quantidadeAluno` INT(11) NULL DEFAULT  
  '0';
```

-- Tabela `discente`

```
CREATE TABLE `discente` (  
  `numeroMatricula` INT NOT NULL,  
  `nomeDiscente` VARCHAR(50) NOT NULL,  
  `periodo` INT NOT NULL,  
  `email` VARCHAR(30) NOT NULL,  
  PRIMARY KEY (`numeroMatricula`))  
ENGINE = InnoDB;
```

```
ALTER TABLE `discente`  
  ADD UNIQUE INDEX `email` (`email`);
```

```
-----  
-- Tabela `local`  
-----
```

```
CREATE TABLE `local` (  
  `local` VARCHAR(100) NOT NULL,  
  `codigoTurma` INT NOT NULL,  
  `codigoDisciplina` INT NOT NULL,  
  PRIMARY KEY (`local`, `codigoTurma`, `codigoDisciplina`),  
  INDEX `fk_local_turma1_idx` (`codigoTurma` ASC, `codigoDisciplina` ASC),  
  CONSTRAINT `fk_local_turma1`  
    FOREIGN KEY (`codigoTurma`, `codigoDisciplina`)  
    REFERENCES `turma` (`codigoTurma`, `codigoDisciplina`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-----  
-- Tabela `horarioDeAula`  
-----
```

```
CREATE TABLE `horarioDeAula` (  
  `diaAula` INT NOT NULL,  
  `horario` TIME NOT NULL,  
  `codigoTurma` INT NOT NULL,  
  `codigoDisciplina` INT NOT NULL,  
  PRIMARY KEY (`diaAula`, `horario`, `codigoTurma`, `codigoDisciplina`),  
  INDEX `fk_horarioDeAula_turma1_idx` (`codigoTurma` ASC, `codigoDisciplina` ASC),  
  CONSTRAINT `fk_horarioDeAula_turma1`  
    FOREIGN KEY (`codigoTurma`, `codigoDisciplina`)  
    REFERENCES `turma` (`codigoTurma`, `codigoDisciplina`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-----  
-- Tabela `telefoneDiscente`  
-----
```

```

CREATE TABLE `telefoneDiscente` (
  `telefone` INT NOT NULL,
  `numeroMatricula` INT NOT NULL,
  PRIMARY KEY (`telefone`, `numeroMatricula`),
  INDEX `fk_telefoneDiscente_discente1_idx` (`numeroMatricula` ASC),
  CONSTRAINT `fk_telefoneDiscente_discente1`
    FOREIGN KEY (`numeroMatricula`)
      REFERENCES `discente` (`numeroMatricula`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

ALTER TABLE `telefoneDiscente`
  CHANGE COLUMN `telefone` `telefone` VARCHAR(11) NOT NULL;

```

```

-----
-- Tabela `turmaDiscente`
-----

CREATE TABLE `turmaDiscente` (
  `codigoTurma` INT NOT NULL,
  `codigoDisciplina` INT NOT NULL,
  `numeroMatricula` INT NOT NULL,
  PRIMARY KEY (`codigoTurma`, `codigoDisciplina`, `numeroMatricula`),
  INDEX `fk_turma_has_discente_discente1_idx` (`numeroMatricula` ASC),
  INDEX `fk_turma_has_discente_turma1_idx` (`codigoTurma` ASC, `codigoDisciplina`
ASC),
  CONSTRAINT `fk_turma_has_discente_turma1`
    FOREIGN KEY (`codigoTurma`, `codigoDisciplina`)
      REFERENCES `turma` (`codigoTurma`, `codigoDisciplina`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_turma_has_discente_discente1`
    FOREIGN KEY (`numeroMatricula`)
      REFERENCES `discente` (`numeroMatricula`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Tabela `monitor`
-----

CREATE TABLE `monitor` (
  `numeroMatricula` INT NOT NULL,
  `coeficienteDeEficiencia` DECIMAL(3,2) NOT NULL,
  `monitorRemunerado` TINYINT(1) NOT NULL,
  PRIMARY KEY (`numeroMatricula`),

```

```

CONSTRAINT `fk_monitor_discente1`
  FOREIGN KEY (`numeroMatricula`)
  REFERENCES `discente` (`numeroMatricula`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

ALTER TABLE `monitor`
  CHANGE COLUMN `coeficienteDeEficiencia` `coeficienteDeEficiencia`
  DECIMAL(6,2) NOT NULL;

```

```

-----
-- Tabela `estudanteDeMonitoria`
-----

```

```

CREATE TABLE `estudanteDeMonitoria` (
  `numeroMatricula` INT NOT NULL,
  `indiceDeDificuldade` INT NOT NULL,
  PRIMARY KEY (`numeroMatricula`),
  CONSTRAINT `fk_estudanteDeMonitoria_discente1`
    FOREIGN KEY (`numeroMatricula`)
    REFERENCES `discente` (`numeroMatricula`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Tabela `monitoria`
-----

```

```

CREATE TABLE `monitoria` (
  `codigoMonitoria` INT NOT NULL AUTO_INCREMENT,
  `numeroMatriculaMonitor` INT NOT NULL,
  `codigoDisciplina` INT NOT NULL,
  `valorBolsaMonitoria` DECIMAL(4,2) NOT NULL,
  `quantidadeAluno` INT NULL,
  PRIMARY KEY (`codigoMonitoria`, `numeroMatriculaMonitor`),
  INDEX `fk_monitoria_monitor1_idx` (`numeroMatriculaMonitor` ASC),
  INDEX `fk_monitoria_disciplina1_idx` (`codigoDisciplina` ASC),
  CONSTRAINT `fk_monitoria_monitor1`
    FOREIGN KEY (`numeroMatriculaMonitor`)
    REFERENCES `monitor` (`numeroMatricula`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_monitoria_disciplina1`
    FOREIGN KEY (`codigoDisciplina`)
    REFERENCES `disciplina` (`codigoDisciplina`)
    ON DELETE NO ACTION

```

```
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
ALTER TABLE `monitoria`
CHANGE COLUMN `valorBolsaMonitoria` `valorBolsaMonitoria` DECIMAL(7,2) NOT
NULL DEFAULT 0;
```

```
-- Tabela `participante_monitoria`
```

```
CREATE TABLE `participante_monitoria` (
  `numeroMatricula` INT NOT NULL,
  `codigoMonitoria` INT NOT NULL,
  PRIMARY KEY (`numeroMatricula`, `codigoMonitoria`),
  INDEX `fk_estudanteDeMonitoria_has_monitoria_monitoria1_idx` (`codigoMonitoria` ASC),
  INDEX `fk_estudanteDeMonitoria_has_monitoria_estudanteDeMonitoria1_idx`
(`numeroMatricula` ASC),
  CONSTRAINT `fk_estudanteDeMonitoria_has_monitoria_estudanteDeMonitoria1`
  FOREIGN KEY (`numeroMatricula`)
  REFERENCES `estudanteDeMonitoria` (`numeroMatricula`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT `fk_estudanteDeMonitoria_has_monitoria_monitoria1`
  FOREIGN KEY (`codigoMonitoria`)
  REFERENCES `monitoria` (`codigoMonitoria`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Tabela `localMonitoria`
```

```
CREATE TABLE `localMonitoria` (
  `local` VARCHAR(100) NOT NULL,
  `codigoMonitoria` INT NOT NULL,
  PRIMARY KEY (`local`, `codigoMonitoria`),
  INDEX `fk_localMonitoria_monitoria1_idx` (`codigoMonitoria` ASC),
  CONSTRAINT `fk_localMonitoria_monitoria1`
  FOREIGN KEY (`codigoMonitoria`)
  REFERENCES `monitoria` (`codigoMonitoria`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Tabela `horarioMonitoria`
```

```

-----
CREATE TABLE `horarioMonitoria` (
  `dia` INT NOT NULL,
  `horario` TIME NOT NULL,
  `codigoMonitoria` INT NOT NULL,
  PRIMARY KEY (`dia`, `horario`, `codigoMonitoria`),
  INDEX `fk_horarioMonitoria_monitoria1_idx` (`codigoMonitoria` ASC),
  CONSTRAINT `fk_horarioMonitoria_monitoria1`
    FOREIGN KEY (`codigoMonitoria`)
      REFERENCES `monitoria` (`codigoMonitoria`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
--                                INSERTS
-----

```

-- DEPARTAMENTO

```

INSERT INTO `departamento` (`nomeDepartamento`) VALUES ('Departamento 1');
INSERT INTO `departamento` (`nomeDepartamento`) VALUES ('Departamento 2');
INSERT INTO `departamento` (`nomeDepartamento`) VALUES ('Departamento 3');
INSERT INTO `departamento` (`nomeDepartamento`) VALUES ('Departamento 4');
INSERT INTO `departamento` (`nomeDepartamento`) VALUES ('Departamento 5');
INSERT INTO `departamento` (`nomeDepartamento`) VALUES ('Departamento 6');
INSERT INTO `departamento` (`nomeDepartamento`) VALUES ('Departamento 7');
INSERT INTO `departamento` (`nomeDepartamento`) VALUES ('Departamento 8');
INSERT INTO `departamento` (`nomeDepartamento`) VALUES ('Departamento 9');
INSERT INTO `departamento` (`nomeDepartamento`) VALUES ('Departamento 10');

```

-- DISCENTE

```

INSERT INTO `discente` (`numeroMatricula`, `nomeDiscente`, `periodo`, `email`) VALUES
(202320001, 'Discente 1', 1, 'discente1@estudante.ufla.br');
INSERT INTO `discente` (`numeroMatricula`, `nomeDiscente`, `periodo`, `email`) VALUES
(202320002, 'Discente 2', 2, 'discente2@estudante.ufla.br');
INSERT INTO `discente` (`numeroMatricula`, `nomeDiscente`, `periodo`, `email`) VALUES
(202320003, 'Discente 3', 3, 'discente3@estudante.ufla.br');
INSERT INTO `discente` (`numeroMatricula`, `nomeDiscente`, `periodo`, `email`) VALUES
(202320004, 'Discente 4', 4, 'discente4@estudante.ufla.br');
INSERT INTO `discente` (`numeroMatricula`, `nomeDiscente`, `periodo`, `email`) VALUES
(202320005, 'Discente 5', 5, 'discente5@estudante.ufla.br');
INSERT INTO `discente` (`numeroMatricula`, `nomeDiscente`, `periodo`, `email`) VALUES
(202320006, 'Discente 6', 6, 'discente6@estudante.ufla.br');
INSERT INTO `discente` (`numeroMatricula`, `nomeDiscente`, `periodo`, `email`) VALUES
(202320007, 'Discente 7', 1, 'discente7@estudante.ufla.br');
INSERT INTO `discente` (`numeroMatricula`, `nomeDiscente`, `periodo`, `email`) VALUES
(202320008, 'Discente 8', 8, 'discente8@estudante.ufla.br');

```

```
INSERT INTO `discente` (`numeroMatricula`, `nomeDiscente`, `periodo`, `email`) VALUES
(202320009, 'Discente 9', 4, 'discente9@estudante.ufla.br');
INSERT INTO `discente` (`numeroMatricula`, `nomeDiscente`, `periodo`, `email`) VALUES
(202320010, 'Discente 10', 6, 'discente10@estudante.ufla.br');
```

-- TELEFONE DISCENTE

```
INSERT INTO `telefonediscente` (`telefone`, `numeroMatricula`) VALUES ('9999999991',
202320001);
INSERT INTO `telefonediscente` (`telefone`, `numeroMatricula`) VALUES ('9999999992',
202320002);
INSERT INTO `telefonediscente` (`telefone`, `numeroMatricula`) VALUES ('9999999993',
202320003);
INSERT INTO `telefonediscente` (`telefone`, `numeroMatricula`) VALUES ('9999999994',
202320004);
INSERT INTO `telefonediscente` (`telefone`, `numeroMatricula`) VALUES ('9999999995',
202320005);
INSERT INTO `telefonediscente` (`telefone`, `numeroMatricula`) VALUES ('9999999996',
202320006);
INSERT INTO `telefonediscente` (`telefone`, `numeroMatricula`) VALUES ('9999999997',
202320007);
INSERT INTO `telefonediscente` (`telefone`, `numeroMatricula`) VALUES ('9999999998',
202320008);
INSERT INTO `telefonediscente` (`telefone`, `numeroMatricula`) VALUES ('9999999999',
202320009);
INSERT INTO `telefonediscente` (`telefone`, `numeroMatricula`) VALUES ('99999999910',
202320010);
```

-- DOCENTE

```
INSERT INTO `docente` (`SIAPE`, `codigoDepartamento`, `nomeDocente`, `email`)
VALUES (200010001, 1, 'Professor 1', 'professor1@ufla.br');
INSERT INTO `docente` (`SIAPE`, `codigoDepartamento`, `nomeDocente`, `email`)
VALUES (200010002, 1, 'Professor 2', 'professor2@ufla.br');
INSERT INTO `docente` (`SIAPE`, `codigoDepartamento`, `nomeDocente`, `email`)
VALUES (200010003, 1, 'Professor 3', 'professor3@ufla.br');
INSERT INTO `docente` (`SIAPE`, `codigoDepartamento`, `nomeDocente`, `email`)
VALUES (200010004, 1, 'Professor 4', 'professor4@ufla.br');
INSERT INTO `docente` (`SIAPE`, `codigoDepartamento`, `nomeDocente`, `email`)
VALUES (200010005, 1, 'Professor 5', 'professor5@ufla.br');
INSERT INTO `docente` (`SIAPE`, `codigoDepartamento`, `nomeDocente`, `email`)
VALUES (200010006, 1, 'Professor 6', 'professor6@ufla.br');
INSERT INTO `docente` (`SIAPE`, `codigoDepartamento`, `nomeDocente`, `email`)
VALUES (200010007, 1, 'Professor 7', 'professor7@ufla.br');
INSERT INTO `docente` (`SIAPE`, `codigoDepartamento`, `nomeDocente`, `email`)
VALUES (200010008, 1, 'Professor 8', 'professor8@ufla.br');
INSERT INTO `docente` (`SIAPE`, `codigoDepartamento`, `nomeDocente`, `email`)
VALUES (200010009, 1, 'Professor 9', 'professor9@ufla.br');
INSERT INTO `docente` (`SIAPE`, `codigoDepartamento`, `nomeDocente`, `email`)
VALUES (200010010, 1, 'Professor 10', 'professor10@ufla.br');
```

-- TELEFONE DOCENTE

```
INSERT INTO `telefonedocente` (`telefone`, `SIAPE`) VALUES ('9099999991', 200010001);
INSERT INTO `telefonedocente` (`telefone`, `SIAPE`) VALUES ('9099999992', 200010002);
INSERT INTO `telefonedocente` (`telefone`, `SIAPE`) VALUES ('9099999993', 200010003);
INSERT INTO `telefonedocente` (`telefone`, `SIAPE`) VALUES ('9099999994', 200010004);
INSERT INTO `telefonedocente` (`telefone`, `SIAPE`) VALUES ('9099999995', 200010005);
INSERT INTO `telefonedocente` (`telefone`, `SIAPE`) VALUES ('9099999996', 200010006);
INSERT INTO `telefonedocente` (`telefone`, `SIAPE`) VALUES ('9099999997', 200010007);
INSERT INTO `telefonedocente` (`telefone`, `SIAPE`) VALUES ('9099999998', 200010008);
INSERT INTO `telefonedocente` (`telefone`, `SIAPE`) VALUES ('9099999999', 200010009);
INSERT INTO `telefonedocente` (`telefone`, `SIAPE`) VALUES ('9099999910', 200010010);
```

-- DISCIPLINA

```
INSERT INTO `disciplina` (`codigoDepartamento`, `SIAPEProfessor`, `nomeDisciplina`)
VALUES (1, 200010001, 'DISCIPLINA 1');
INSERT INTO `disciplina` (`codigoDepartamento`, `SIAPEProfessor`, `nomeDisciplina`)
VALUES (2, 200010002, 'DISCIPLINA 2');
INSERT INTO `disciplina` (`codigoDepartamento`, `SIAPEProfessor`, `nomeDisciplina`)
VALUES (3, 200010003, 'DISCIPLINA 3');
INSERT INTO `disciplina` (`codigoDepartamento`, `SIAPEProfessor`, `nomeDisciplina`)
VALUES (4, 200010004, 'DISCIPLINA 4');
INSERT INTO `disciplina` (`codigoDepartamento`, `SIAPEProfessor`, `nomeDisciplina`)
VALUES (5, 200010005, 'DISCIPLINA 5');
INSERT INTO `disciplina` (`codigoDepartamento`, `SIAPEProfessor`, `nomeDisciplina`)
VALUES (6, 200010006, 'DISCIPLINA 6');
INSERT INTO `disciplina` (`codigoDepartamento`, `SIAPEProfessor`, `nomeDisciplina`)
VALUES (7, 200010007, 'DISCIPLINA 7');
INSERT INTO `disciplina` (`codigoDepartamento`, `SIAPEProfessor`, `nomeDisciplina`)
VALUES (8, 200010008, 'DISCIPLINA 8');
INSERT INTO `disciplina` (`codigoDepartamento`, `SIAPEProfessor`, `nomeDisciplina`)
VALUES (9, 200010009, 'DISCIPLINA 9');
INSERT INTO `disciplina` (`codigoDepartamento`, `SIAPEProfessor`, `nomeDisciplina`)
VALUES (10, 200010010, 'DISCIPLINA 10');
```

-- TURMA

```
INSERT INTO `turma` (`codigoTurma`, `codigoDisciplina`, `quantidadeAluno`) VALUES (1,
1, 1);
INSERT INTO `turma` (`codigoTurma`, `codigoDisciplina`, `quantidadeAluno`) VALUES (2,
2, 1);
INSERT INTO `turma` (`codigoTurma`, `codigoDisciplina`, `quantidadeAluno`) VALUES (3,
3, 1);
INSERT INTO `turma` (`codigoTurma`, `codigoDisciplina`, `quantidadeAluno`) VALUES (4,
4, 1);
INSERT INTO `turma` (`codigoTurma`, `codigoDisciplina`, `quantidadeAluno`) VALUES (5,
5, 1);
```

```

INSERT INTO `turma` (`codigoTurma`, `codigoDisciplina`, `quantidadeAluno`) VALUES (6, 1);
INSERT INTO `turma` (`codigoTurma`, `codigoDisciplina`, `quantidadeAluno`) VALUES (7, 1);
INSERT INTO `turma` (`codigoTurma`, `codigoDisciplina`, `quantidadeAluno`) VALUES (8, 1);
INSERT INTO `turma` (`codigoTurma`, `codigoDisciplina`, `quantidadeAluno`) VALUES (9, 1);
INSERT INTO `turma` (`codigoTurma`, `codigoDisciplina`, `quantidadeAluno`) VALUES (10, 1);

```

-- TURMA DISCENTE

```

INSERT INTO `turmadiscente` (`codigoTurma`, `codigoDisciplina`, `numeroMatricula`)
VALUES (1, 1, 202320001);
INSERT INTO `turmadiscente` (`codigoTurma`, `codigoDisciplina`, `numeroMatricula`)
VALUES (2, 2, 202320002);
INSERT INTO `turmadiscente` (`codigoTurma`, `codigoDisciplina`, `numeroMatricula`)
VALUES (3, 3, 202320003);
INSERT INTO `turmadiscente` (`codigoTurma`, `codigoDisciplina`, `numeroMatricula`)
VALUES (4, 4, 202320004);
INSERT INTO `turmadiscente` (`codigoTurma`, `codigoDisciplina`, `numeroMatricula`)
VALUES (5, 5, 202320005);
INSERT INTO `turmadiscente` (`codigoTurma`, `codigoDisciplina`, `numeroMatricula`)
VALUES (6, 6, 202320006);
INSERT INTO `turmadiscente` (`codigoTurma`, `codigoDisciplina`, `numeroMatricula`)
VALUES (7, 7, 202320007);
INSERT INTO `turmadiscente` (`codigoTurma`, `codigoDisciplina`, `numeroMatricula`)
VALUES (8, 8, 202320008);
INSERT INTO `turmadiscente` (`codigoTurma`, `codigoDisciplina`, `numeroMatricula`)
VALUES (9, 9, 202320009);
INSERT INTO `turmadiscente` (`codigoTurma`, `codigoDisciplina`, `numeroMatricula`)
VALUES (10, 10, 202320010);

```

-- HORARIO AULA

```

INSERT INTO `horariodeaula` (`diaAula`, `horario`, `codigoTurma`, `codigoDisciplina`)
VALUES (1, '19:00:00', 1, 1);
INSERT INTO `horariodeaula` (`diaAula`, `horario`, `codigoTurma`, `codigoDisciplina`)
VALUES (2, '19:00:00', 2, 2);
INSERT INTO `horariodeaula` (`diaAula`, `horario`, `codigoTurma`, `codigoDisciplina`)
VALUES (3, '19:00:00', 3, 3);
INSERT INTO `horariodeaula` (`diaAula`, `horario`, `codigoTurma`, `codigoDisciplina`)
VALUES (4, '19:00:00', 4, 4);
INSERT INTO `horariodeaula` (`diaAula`, `horario`, `codigoTurma`, `codigoDisciplina`)
VALUES (5, '19:00:00', 5, 5);
INSERT INTO `horariodeaula` (`diaAula`, `horario`, `codigoTurma`, `codigoDisciplina`)
VALUES (1, '21:00:00', 6, 6);

```



```

INSERT INTO `horariodeaula` (`diaAula`, `horario`, `codigoTurma`, `codigoDisciplina`)
VALUES (2, '21:00:00', 7, 7);
INSERT INTO `horariodeaula` (`diaAula`, `horario`, `codigoTurma`, `codigoDisciplina`)
VALUES (3, '21:00:00', 8, 8);
INSERT INTO `horariodeaula` (`diaAula`, `horario`, `codigoTurma`, `codigoDisciplina`)
VALUES (4, '21:00:00', 9, 9);
INSERT INTO `horariodeaula` (`diaAula`, `horario`, `codigoTurma`, `codigoDisciplina`)
VALUES (5, '21:00:00', 10, 10);

```

-- LOCAL

```

INSERT INTO `local` (`local`, `codigoTurma`, `codigoDisciplina`) VALUES ('DCC01', 1, 1);
INSERT INTO `local` (`local`, `codigoTurma`, `codigoDisciplina`) VALUES ('DCC02', 2, 2);
INSERT INTO `local` (`local`, `codigoTurma`, `codigoDisciplina`) VALUES ('DCC03', 3, 3);
INSERT INTO `local` (`local`, `codigoTurma`, `codigoDisciplina`) VALUES ('DCC04', 4, 4);
INSERT INTO `local` (`local`, `codigoTurma`, `codigoDisciplina`) VALUES ('DCC05', 5, 5);
INSERT INTO `local` (`local`, `codigoTurma`, `codigoDisciplina`) VALUES ('DCC01', 6, 6);
INSERT INTO `local` (`local`, `codigoTurma`, `codigoDisciplina`) VALUES ('DCC02', 7, 7);
INSERT INTO `local` (`local`, `codigoTurma`, `codigoDisciplina`) VALUES ('DCC03', 8, 8);
INSERT INTO `local` (`local`, `codigoTurma`, `codigoDisciplina`) VALUES ('DCC04', 9, 9);
INSERT INTO `local` (`local`, `codigoTurma`, `codigoDisciplina`) VALUES ('DCC05', 10, 10);

```

-- MONITOR

```

INSERT INTO `monitor` (`numeroMatricula`, `coeficienteDeEficiencia`,
`monitorRemunerado`) VALUES (202320001, 100, 1);
INSERT INTO `monitor` (`numeroMatricula`, `coeficienteDeEficiencia`,
`monitorRemunerado`) VALUES (202320002, 70, 0);
INSERT INTO `monitor` (`numeroMatricula`, `coeficienteDeEficiencia`,
`monitorRemunerado`) VALUES (202320003, 100, 1);
INSERT INTO `monitor` (`numeroMatricula`, `coeficienteDeEficiencia`,
`monitorRemunerado`) VALUES (202320004, 50, 0);

```

-- MONITORIA

```

INSERT INTO `monitoria` (`numeroMatriculaMonitor`, `codigoDisciplina`,
`valorBolsaMonitoria`, `quantidadeAluno`) VALUES (202320001, 1, 500, 0);
INSERT INTO `monitoria` (`numeroMatriculaMonitor`, `codigoDisciplina`,
`valorBolsaMonitoria`, `quantidadeAluno`) VALUES (202320002, 2, 0, 0);
INSERT INTO `monitoria` (`numeroMatriculaMonitor`, `codigoDisciplina`,
`valorBolsaMonitoria`, `quantidadeAluno`) VALUES (202320003, 3, 500, 0);
INSERT INTO `monitoria` (`numeroMatriculaMonitor`, `codigoDisciplina`,
`valorBolsaMonitoria`, `quantidadeAluno`) VALUES (202320004, 4, 0, 0);
INSERT INTO `monitoria` (`numeroMatriculaMonitor`, `codigoDisciplina`,
`valorBolsaMonitoria`, `quantidadeAluno`) VALUES (202320001, 5, 500, 0);
INSERT INTO `monitoria` (`numeroMatriculaMonitor`, `codigoDisciplina`,
`valorBolsaMonitoria`, `quantidadeAluno`) VALUES (202320003, 6, 500, 0);

```

-- LOCAL MONITORIA

```

INSERT INTO `localmonitoria` (`local`, `codigoMonitoria`) VALUES ('DCC1', 1);
INSERT INTO `localmonitoria` (`local`, `codigoMonitoria`) VALUES ('DCC2', 2);

```

```
INSERT INTO `localmonitoria` (`local`, `codigoMonitoria`) VALUES ('DCC3', 3);
INSERT INTO `localmonitoria` (`local`, `codigoMonitoria`) VALUES ('DCC4', 4);
INSERT INTO `localmonitoria` (`local`, `codigoMonitoria`) VALUES ('DCC5', 5);
INSERT INTO `localmonitoria` (`local`, `codigoMonitoria`) VALUES ('DCC1', 6);
```

-- HORARIO MONITORIA

```
INSERT INTO `horariomonitoria` (`dia`, `horario`, `codigoMonitoria`) VALUES (1, '17:00:00', 1);
INSERT INTO `horariomonitoria` (`dia`, `horario`, `codigoMonitoria`) VALUES (2, '17:00:00', 2);
INSERT INTO `horariomonitoria` (`dia`, `horario`, `codigoMonitoria`) VALUES (3, '17:00:00', 3);
INSERT INTO `horariomonitoria` (`dia`, `horario`, `codigoMonitoria`) VALUES (4, '17:00:00', 4);
INSERT INTO `horariomonitoria` (`dia`, `horario`, `codigoMonitoria`) VALUES (5, '17:00:00', 5);
INSERT INTO `horariomonitoria` (`dia`, `horario`, `codigoMonitoria`) VALUES (1, '16:00:00', 6);
```

-- ESTUDANTE DE MONITORIA

```
INSERT INTO `estudantedemonitoria` (`numeroMatricula`, `indiceDeDificuldade`) VALUES (202320001, 3);
INSERT INTO `estudantedemonitoria` (`numeroMatricula`, `indiceDeDificuldade`) VALUES (202320002, 3);
INSERT INTO `estudantedemonitoria` (`numeroMatricula`, `indiceDeDificuldade`) VALUES (202320003, 2);
INSERT INTO `estudantedemonitoria` (`numeroMatricula`, `indiceDeDificuldade`) VALUES (202320004, 2);
INSERT INTO `estudantedemonitoria` (`numeroMatricula`, `indiceDeDificuldade`) VALUES (202320005, 2);
INSERT INTO `estudantedemonitoria` (`numeroMatricula`, `indiceDeDificuldade`) VALUES (202320006, 1);
```

-- PARTICIPANTE MONITORIA

```
INSERT INTO `participante_monitoria` (`numeroMatricula`, `codigoMonitoria`) VALUES (202320001, 1);
INSERT INTO `participante_monitoria` (`numeroMatricula`, `codigoMonitoria`) VALUES (202320001, 2);
INSERT INTO `participante_monitoria` (`numeroMatricula`, `codigoMonitoria`) VALUES (202320002, 1);
INSERT INTO `participante_monitoria` (`numeroMatricula`, `codigoMonitoria`) VALUES (202320002, 3);
INSERT INTO `participante_monitoria` (`numeroMatricula`, `codigoMonitoria`) VALUES (202320004, 1);
INSERT INTO `participante_monitoria` (`numeroMatricula`, `codigoMonitoria`) VALUES (202320005, 6);
INSERT INTO `participante_monitoria` (`numeroMatricula`, `codigoMonitoria`) VALUES (202320006, 5);
```

-- EXEMPLO DE DROP

```
CREATE TABLE `tabela_teste` (  
  `ID` INT NOT NULL AUTO_INCREMENT,  
  PRIMARY KEY (`ID`)  
)  
ENGINE = InnoDB;
```

```
DROP TABLE tabela_teste;
```

-- EXEMPLO DE UPDATE

```
UPDATE docente  
SET nomeDocente = 'Docente da disciplina 1'  
WHERE SIAPE = (  
  SELECT SIAPEProfessor  
  FROM disciplina  
  WHERE nomeDisciplina = 'DISCIPLINA 1'  
);
```

```
UPDATE disciplina  
SET nomeDisciplina = 'Introdução ao Sistemas de Banco de Dados'  
WHERE nomeDisciplina = 'DISCIPLINA 2';
```

```
UPDATE horarioDeAula  
SET horario = '14:00:00'  
WHERE codigoTurma = 1 AND codigoDisciplina = 1;
```

```
UPDATE monitor  
SET coeficienteDeEficiencia = 0.85  
WHERE numeroMatricula = '202320002';
```

```
UPDATE discente  
SET email = 'newemail@example.com'  
WHERE numeroMatricula = '202320003';
```

-- SELECTs

```
-- Retorna o nome e o email de todos os docentes.  
SELECT nomeDocente, email  
FROM docente;
```

```
-- Retorna o nome da disciplina e a quantidade de alunos matriculados em cada turma.  
SELECT d.nomeDisciplina, t.quantidadeAluno  
FROM disciplina d
```

```
JOIN turma t ON d.codigoDisciplina = t.codigoDisciplina;
```

```
-- Retorna o nome da disciplina e a quantidade de alunos matriculados em cada turma.
```

```
SELECT d.nomeDisciplina, t.quantidadeAluno
```

```
FROM disciplina d
```

```
JOIN turma t ON d.codigoDisciplina = t.codigoDisciplina;
```

```
-- Calcula a média do coeficiente de eficiência dos monitores por disciplina e filtra as disciplinas com média superior a 0.8.
```

```
SELECT MT.codigoDisciplina, AVG(M.coeficienteDeEficiencia) AS media
```

```
FROM monitor M
```

```
LEFT JOIN monitoria MT ON MT.numeroMatriculaMonitor = M.numeroMatricula
```

```
GROUP BY MT.codigoDisciplina
```

```
HAVING media > 0.8;
```

```
-- Recupera o nome do discente e o código da turma, incluindo discentes sem turma.
```

```
SELECT d.nomeDiscente, t.codigoTurma
```

```
FROM discente d
```

```
LEFT JOIN turmaDiscente t ON d.numeroMatricula = t.numeroMatricula;
```

```
-- Recupera o nome e o período dos discentes nos períodos 1, 2 ou 3.
```

```
SELECT nomeDiscente, periodo
```

```
FROM discente
```

```
WHERE periodo IN (1, 2, 3);
```

```
-- Recupera o nome dos discentes cujos e-mails são da UFLA
```

```
SELECT nomeDiscente
```

```
FROM discente
```

```
WHERE email LIKE '%@estudante.ufla.br';
```

```
-- Recupera o nome dos discentes que não estão no primeiro período
```

```
SELECT nomeDiscente
```

```
FROM discente
```

```
WHERE NOT periodo = 1;
```

```
-- Retorna o nome das disciplinas cujos departamentos correspondem ao 'DepartamentoX'.
```

```
SELECT nomeDisciplina
```

```
FROM disciplina
```

```
WHERE codigoDepartamento IN (
```

```
    SELECT codigoDepartamento
```

```
    FROM departamento
```

```
    WHERE nomeDepartamento = 'DEX'
```

```
);
```

```
-- Recupera o nome dos discentes cujo período é igual a qualquer período superior a 5.
```

```
SELECT nomeDiscente
```

```
FROM discente
```

```
WHERE periodo = ANY (SELECT DISTINCT periodo FROM discente WHERE periodo > 5);
```

-- Retorna o nome dos discentes que estão no período 1 ou 2, removendo duplicatas.

```
SELECT nomeDiscente
      FROM discente
      WHERE periodo = 1
UNION
SELECT nomeDiscente
      FROM discente
      WHERE periodo = 2;
```

-- Recupera o nome dos docentes e a média de eficiência dos monitores associados a cada docente.

```
SELECT nomeDocente, (SELECT AVG(coeficienteDeEficiencia) FROM monitor WHERE
numeroMatricula = d.SIAPE) AS MediaEficiencia
FROM docente d;
```

-- Exemplo de ORDER BY – Seleciona todos os professores do departamento 1 retornando em ordem decrescente os nomes

```
SELECT SIAPE, nomeDocente
FROM docente
WHERE codigoDepartamento = 1
ORDER BY nomeDocente DESC;
```

-- Exemplo de ALL – Seleciona todas as monitorias que tem horário marcado antes das 19 h em dia que tenha disciplina neste horario

```
SELECT codigoMonitoria
FROM horarioMonitoria
WHERE '19:00:00' > ALL (
      SELECT horario
      FROM horarioDeAula
      WHERE diaAula = horarioMonitoria.dia
            AND horarioDeAula.codigoTurma = horarioMonitoria.codigoTurma
            AND horarioDeAula.codigoDisciplina = horarioMonitoria.codigoDisciplina
);
```

-- Exemplo de EXISTS – Verifica se há disciplinas associadas ao docente ' Professor 2' e, se existirem, retornar o nome dessas disciplinas e o nome do departamento

```
SELECT d.nomeDisciplina, dep.nomeDepartamento
FROM disciplina d
JOIN departamento dep ON d.codigoDepartamento = dep.codigoDepartamento
WHERE EXISTS (
      SELECT 1
      FROM docente
      WHERE nomeDocente = 'Professor 2' AND SIAPE = d.SIAPEProfessor
```

);

```
SELECT d.nomeDisciplina
FROM disciplina d
JOIN monitoria m ON m.codigoDisciplina = d.codigoDisciplina
JOIN discente di ON di.numeroMatricula = m.numeroMatriculaMonitor
JOIN estudantedemonitoria e ON e.numeroMatricula = di.numeroMatricula
WHERE e.indiceDeDificuldade BETWEEN 1 AND 2;
```

-- Visões

```
CREATE VIEW view_docentes AS
SELECT nomeDocente, email
FROM docente;
```

```
CREATE VIEW view_turma_disciplina AS
SELECT d.nomeDisciplina, t.quantidadeAluno
FROM disciplina d
JOIN turma t ON d.codigoDisciplina = t.codigoDisciplina;
```

```
CREATE VIEW view_media_monitor AS
SELECT MT.codigoDisciplina, AVG(M.coeficienteDeEficiencia) AS media
FROM monitor M
LEFT JOIN monitoria MT ON MT.numeroMatriculaMonitor = M.numeroMatricula
GROUP BY MT.codigoDisciplina;
```

```
SELECT * FROM view_docentes;
SELECT * FROM view_turma_disciplina;
SELECT * FROM view_media_monitor;
```

-- EXEMPLO DE GRANT E REVOKE

```
CREATE USER 'usuario1'@'localhost' IDENTIFIED BY '12345';
CREATE USER 'usuario2'@'localhost' IDENTIFIED BY '678910';
```

```
GRANT SELECT ON gerenciador_monitoria.* TO 'usuario1'@'localhost';
GRANT INSERT, UPDATE ON gerenciador_monitoria.* TO 'usuario2'@'localhost';
```

```
REVOKE SELECT ON gerenciador_monitoria.* FROM 'usuario1'@'localhost';
REVOKE INSERT, UPDATE ON gerenciador_monitoria.* FROM 'usuario2'@'localhost';
```

-- PROCEDURES/FUNCTIONS/TRIGGERS

```
-- Trigger para a tabela `local`
DELIMITER //
```

```
CREATE TRIGGER before_insert_local
BEFORE INSERT ON `local`
```

```

FOR EACH ROW
BEGIN
    DECLARE qtdLinhas INT;

    -- Verifica se já existe um registro com o mesmo local
    SELECT COUNT(*)
    INTO qtdLinhas
    FROM `local`
    WHERE `local` = NEW.`local`;

    -- Se existir, cancela a inserção
    IF qtdLinhas > 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Não é permitido inserir um registro com o mesmo local.';
    END IF;
END //

DELIMITER ;

-- Trigger para a tabela `horarioDeAula`
DELIMITER //

CREATE TRIGGER before_insert_horarioDeAula
BEFORE INSERT ON `horarioDeAula`
FOR EACH ROW
BEGIN
    DECLARE qtdLinhas INT;

    -- Verifica se já existe um registro com o mesmo dia e horário
    SELECT COUNT(*)
    INTO qtdLinhas
    FROM `horarioDeAula`
    WHERE diaAula = NEW.diaAula
        AND horario = NEW.horario;

    -- Se existir, cancela a inserção
    IF qtdLinhas > 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Não é permitido inserir um registro com o mesmo dia e
horário.';
    END IF;
END //

DELIMITER ;

-- Trigger para a tabela `localMonitoria`
DELIMITER //

```

```

CREATE TRIGGER before_insert_localMonitoria
BEFORE INSERT ON `localMonitoria`
FOR EACH ROW
BEGIN
    DECLARE qtdLinhas INT;

    -- Verifica se já existe um registro com o mesmo local
    SELECT COUNT(*)
    INTO qtdLinhas
    FROM `localMonitoria`
    WHERE `local` = NEW.`local`;

    -- Se existir, cancela a inserção
    IF qtdLinhas > 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Não é permitido inserir um registro com o mesmo local.';
    END IF;
END //

DELIMITER ;

-- Trigger para a tabela `horarioMonitoria`
DELIMITER //

CREATE TRIGGER before_insert_horarioMonitoria
BEFORE INSERT ON `horarioMonitoria`
FOR EACH ROW
BEGIN
    DECLARE qtdLinhas INT;

    -- Verifica se já existe um registro com o mesmo dia e horário
    SELECT COUNT(*)
    INTO qtdLinhas
    FROM `horarioMonitoria`
    WHERE dia = NEW.dia
        AND horario = NEW.horario;

    -- Se existir, cancela a inserção
    IF qtdLinhas > 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Não é permitido inserir um registro com o mesmo dia e
horário.';
    END IF;
END //

DELIMITER ;

DELIMITER //

```



```

-- Insere o novo departamento
CREATE PROCEDURE InserirDepartamento(
    IN nomeDepartamento VARCHAR(50)
)
BEGIN
    DECLARE codigoDepartamento INT;

    INSERT INTO `departamento` (`nomeDepartamento`)
    VALUES (nomeDepartamento);

END //

DELIMITER ;

CALL InserirDepartamento('Nome do Departamento');

DELIMITER //

-- Atualiza o nome do departamento
CREATE PROCEDURE EditarDepartamento(
    IN codigoDepartamento INT,
    IN novoNomeDepartamento VARCHAR(50)
)
BEGIN
    -- Verifica se o departamento com o código fornecido existe
    IF NOT EXISTS (SELECT 1 FROM `departamento` WHERE `codigoDepartamento` =
codigoDepartamento) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'O departamento com o código fornecido não existe.';
    ELSE
        UPDATE `departamento`
        SET `nomeDepartamento` = novoNomeDepartamento
        WHERE `codigoDepartamento` = codigoDepartamento;
    END IF;
END //

DELIMITER ;

CALL EditarDepartamento(1, 'Novo Nome do Departamento');

-- TRIGGER DE EXCLUSÃO
DELIMITER //

CREATE TRIGGER before_delete_departamento
BEFORE DELETE ON `departamento`
FOR EACH ROW
BEGIN

```

```

DECLARE totalDocentes INT;

-- Verifica se existem funcionários associados ao departamento
SELECT COUNT(*) INTO totalDocentes
FROM `docente`
WHERE `codigoDepartamento` = OLD.`codigoDepartamento`;

-- Se existirem funcionários, cancela a exclusão
IF totalDocentes > 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Não é permitido excluir o departamento. Existem docentes
associados a ele.';
END IF;
END //

DELIMITER ;

-- PROCEDIMENTO SEM PARAMETRO

DELIMITER //
CREATE PROCEDURE contar_turmas()
BEGIN
    DECLARE total_turmas INT;

    SELECT COUNT(*) INTO total_turmas
    FROM turma;

    IF total_turmas > 0 THEN
        SELECT CONCAT('Total de turmas: ', total_turmas) AS mensagem;
    ELSE
        SELECT 'Não há turmas cadastradas.';
    END IF;
END //
DELIMITER ;

-- Exemplo de execução
CALL contar_turmas();

-- EXEMPLO DE FUNCTION com PARAMETRO IN
DELIMITER //
CREATE FUNCTION avaliar_turma(codigoTurma INT) RETURNS VARCHAR(100)
BEGIN
    DECLARE quantidade_alunos INT;

    SELECT quantidadeAluno INTO quantidade_alunos
    FROM turma
    WHERE codigoTurma = codigoTurma;

```

```

RETURN CASE
    WHEN quantidade_alunos > 30 THEN 'Turma lotada'
    WHEN quantidade_alunos > 20 THEN 'Número ideal de alunos'
    ELSE 'Turma com baixa ocupação'
END;
END //
DELIMITER ;

call avaliar_turma(1);

-- EXEMPLO DE FUNCTION COM PARAMETRO OUT
DELIMITER //

CREATE PROCEDURE total_turmas_departamento(IN codigoDepartamento INT, OUT
totalTurmas INT)
BEGIN
    SELECT COUNT(*) INTO totalTurmas
    FROM turma t
    JOIN disciplina d ON t.codigoDisciplina = d.codigoDisciplina
    WHERE d.codigoDepartamento = codigoDepartamento;
END //

DELIMITER ;

-- Variável para armazenar o resultado
SET @totalTurmas := 0;

-- Chamada da função
CALL fn_total_turmas_departamento(1, @totalTurmas);

-- Exibição do resultado
SELECT @totalTurmas AS total_turmas;

-- Exemplo de execução
SELECT avaliar_turma(1); -- Supondo que 1 seja o código de uma turma

-- EXEMPLOS DE DELETE

-- Deletar local de monitoria DCC1
DELETE FROM localmonitoria
WHERE local = "DCC1";

-- Deletar Disciplinas sem Turmas
DELETE FROM `disciplina`
WHERE `codigoDisciplina` NOT IN (SELECT DISTINCT `codigoDisciplina` FROM `turma`);

-- Exclui participantes de monitoria associados à localização 'DCC5'
DELETE FROM participante_monitoria

```

```
WHERE codigoMonitoria IN (SELECT codigoMonitoria  
    FROM localMonitoria  
    WHERE local = 'DCC5');
```

```
-- exclui localizações 'DCC5' da tabela localMonitoria  
DELETE FROM localMonitoria  
WHERE local = 'DCC5';
```

```
-- exclui as monitorias associadas à localização 'DCC5'  
DELETE FROM monitoria  
WHERE codigoMonitoria IN (SELECT codigoMonitoria  
    FROM localMonitoria  
    WHERE local = 'DCC5');
```