

Instituto Federal da Paraíba — IFPB

Campus João Pessoa

Banco de Dados de uma Universidade  
Dicionário de Dados

Catarina Lenice Lopes Pedrosa

Rafael Duarte de Sousa

Agosto de 2018

# 1 Descrição do Mini Mundo

O banco de dados desenvolvido tem como finalidade representar e armazenar as informações dos docentes, discentes, locais e atividades desenvolvidas em uma universidade.

Nela, alunos são matriculados em cursos e inscrevem-se em disciplinas, que podem ter outras disciplinas como pré-requisitos e são ministradas por um ou mais professores. Os professores podem ainda estar envolvidos em projetos de pesquisa que devem possuir pelo menos um discente, podendo este ser bolsista ou voluntário.

Um aluno pode pedir dispensa da disciplina, desde que com o aval do professor responsável pela mesma.

As aulas são realizadas em salas de aula, laboratórios ou auditórios. E ao serem realizadas, os professores que às ministram podem usar ou não usar algum recurso como um equipamento ou um software.

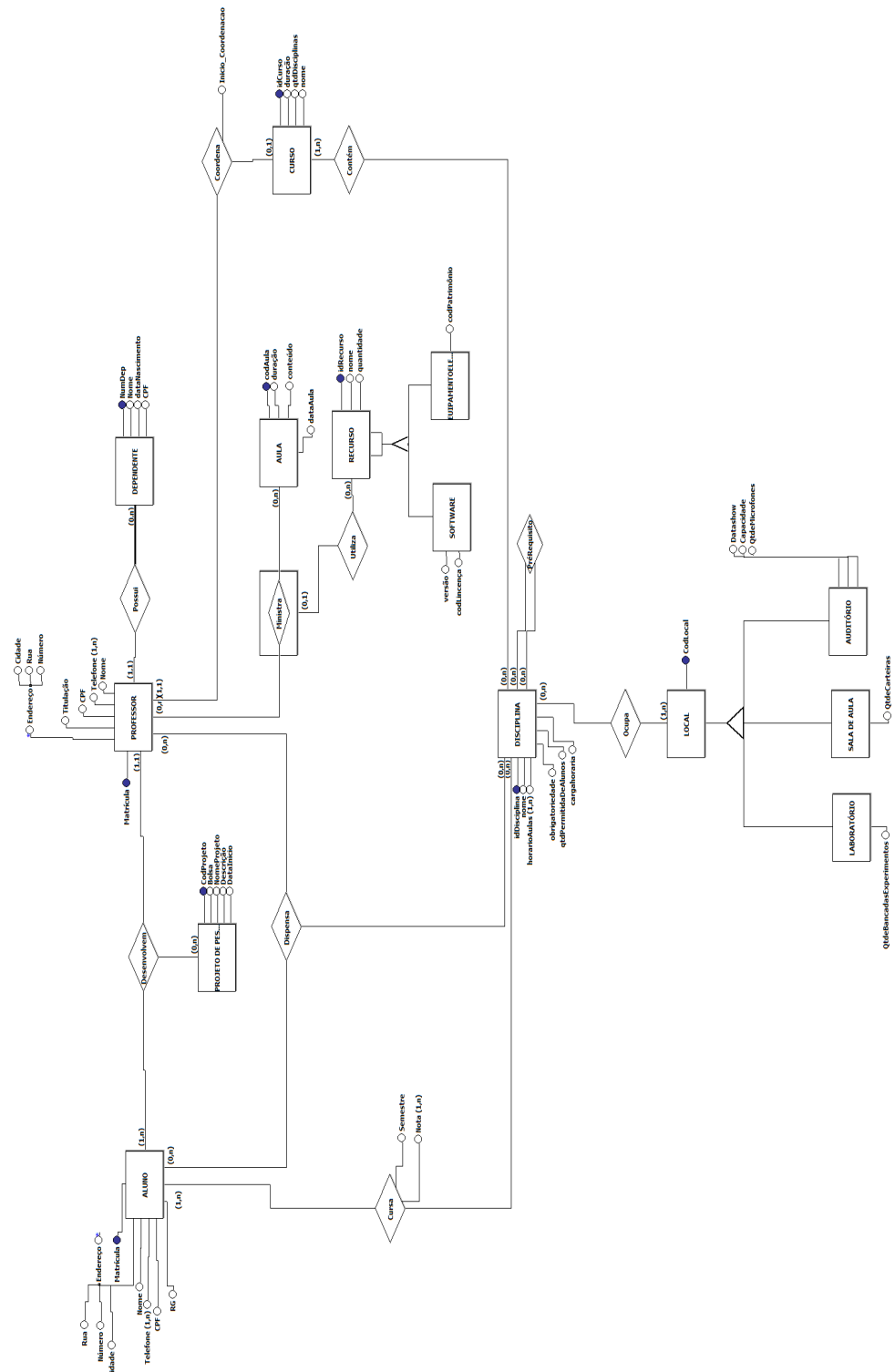
Cada curso tem um professor que o coordena e um professor pode coordenar apenas um curso.

São mantidas informações dos alunos como nome, CPF, endereço (rua, número, bairro), um ou mais números de telefone e matrícula. Dos professores armazena-se: Nome, CPF, matrícula, endereço (rua, número, bairro), um ou mais números de telefone, titulação. Das disciplinas: Horário, carga horária, limite de alunos, obrigatoriedade. Dos locais mantém-se um código de identificação. As salas de têm um número definido de carteiras, os laboratórios de bancadas de experimentos e os auditórios de capacidade e microfones, podendo estes últimos terem datashow.

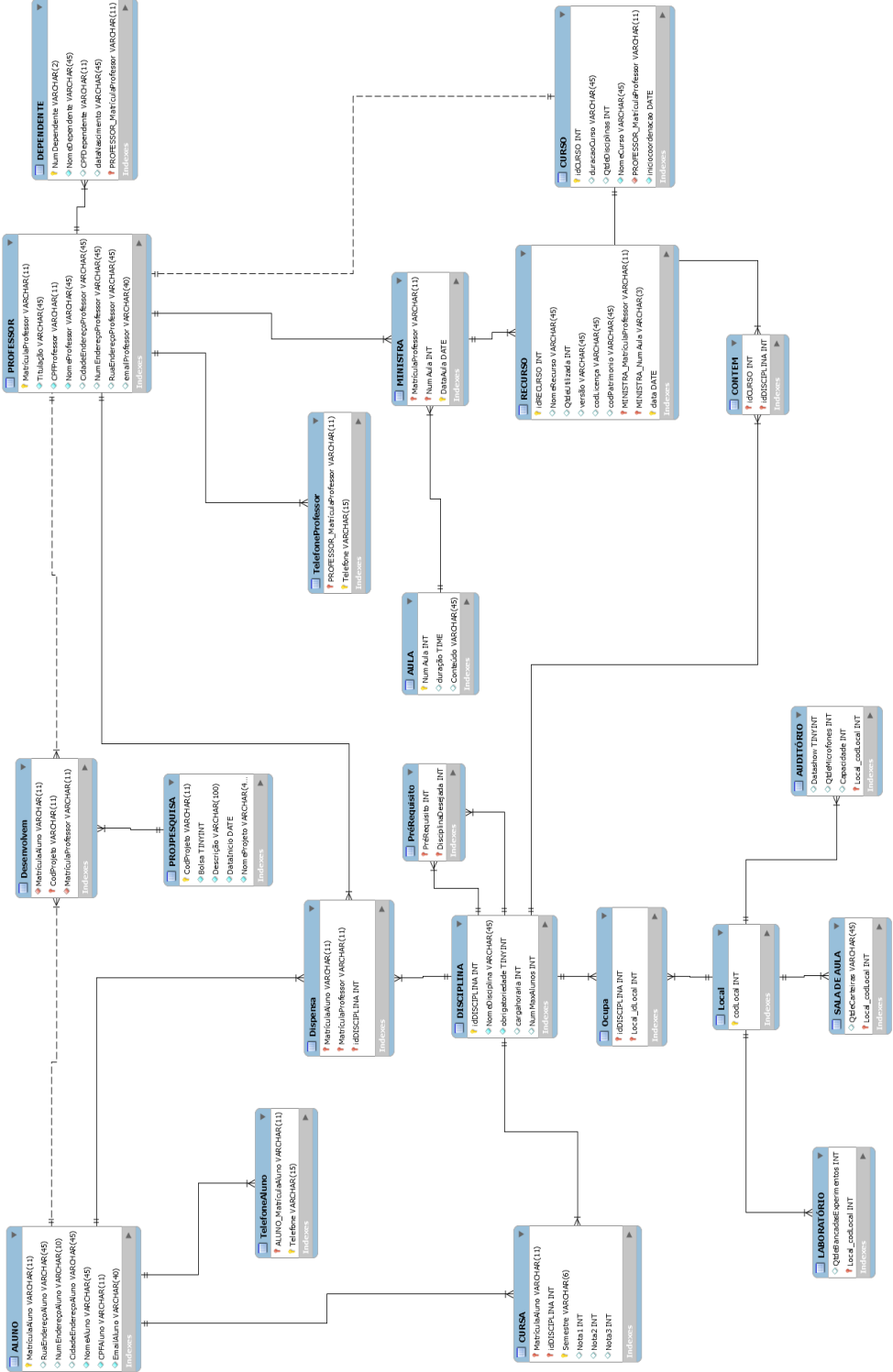
Além disso, mantém-se informações dos dependentes dos professores, como nome, data de nascimento e CPF para a associação dos docentes da universidade.

## 2 Modelos Conceitual e Lógico

## 2.1 Modelo Conceitual: Diagrama Entidade Relacionamento



## 2.2 Modelo Lógico



### 3 Tabela ALUNO

Armazena os dados dos alunos da universidade.

#### 3.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
matriculaaluno	varchar(11)	não	Matrícula do aluno	11 caracteres numéricos	X		
nomealuno	varchar(45)	não	Nome do aluno	Máximo de 45 caracteres			
cpfaluno	varchar(11)	não	Número do CPF do aluno	11 caracteres numéricos			X
ruaendereçaaluno	varchar(45)	sim	Rua em que o aluno mora	Máximo de 45 caracteres			
numendereçaaluno	int	sim	Número da residência do aluno	Número Inteiro			
cidadeendereçaaluno	varchar(45)	sim	Cidade em que o aluno mora	Máximo de 45 caracteres			
emailaluno	varchar(40)	não	E-mail do aluno	Formato a@b.c			

#### 3.2 Constraints

COLUNA	TIPO	NOME	EXPRESSÃO
matriculaaluno	Chave primária	PK_aluno	PRIMARY KEY (matriculaaluno)
matriculaaluno	Validação do domínio: 11 caracteres numéricos	CK_aluno_matriculaaluno	CHECK (LEN(matriculaaluno) = 11 AND matriculaaluno NOT LIKE '%[^0-9]%')
cpfaluno	Chave candidata	AK_aluno_cpfaluno	UNIQUE (cpfaluno)
cpfaluno	Validação do domínio: 11 caracteres numéricos	CK_aluno_cpfaluno	CHECK (LEN(cpfaluno) = 11 AND cpfaluno NOT LIKE '%[^0-9]%')
ruaendereçaaluno	Validação do domínio: Máximo 45 caracteres	CK_aluno_ruaendereçaaluno	CHECK (LEN(ruaendereçaaluno) <= 45)
cidadeendereçaaluno	Validação do domínio: Máximo 45 caracteres	CK_aluno_cidadeendereçaaluno	CHECK (LEN(cidadeendereçaaluno) <= 45)
nomealuno	Validação do domínio: Máximo 45 caracteres	CK_aluno_cpfaluno	CHECK (LEN(cpfaluno) <= 45)
emailaluno	Validação do domínio: Formato a@b.c	CK_aluno_emailaluno	CHECK (emailaluno LIKE '%@-%.-%' AND emailaluno NOT LIKE '%@%@%')

## 4 Tabela TelefoneAluno

Atributo multivalorado. Armazena os números de telefone dos alunos matriculados na universidade.

## 4.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
matriculaaluno	varchar(11)	não	Matrícula do aluno	Tabela ALUNO	X	X	
telefone	varchar(15)	sim	Número do telefone	' + % [0-9]'	X		

## 4.2 Constraints

COLUNA	TIPO	NOME	EXPRESSÃO
matriculaaluno	Chave estrangeira: Referencia matriculaaluno na tabela Aluno	FK_telefonealuno_matriculaaluno	FOREIGN KEY (matriculaaluno) REFERENCES Aluno
matriculaaluno, telefone	Chave primária	PK_telefonealuno	PRIMARY KEY(matriculaaluno, telefone)
telefone	Validação do domínio: 14 ou 15 caracteres numéricos. Formato+%	CK_telefonealuno_telefone	CHECK (LEN(telefone) > 13 AND (telefone LIKE '+[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9] -9[0-9][0-9]' OR telefone LIKE '+[0-9][0-9][0-9][0-9] -9[0-9][0-9][0-9][0-9][0-9][0-9][0-9]'))

## 5 Tabela PROFESSOR

Armazena os dados dos professores da universidade.

### 5.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
matriculaprofessor	varchar(11)	não	Matrícula do professor	11 caracteres numéricos	X		
nomeprofessor	varchar(45)	não	Nome do professor	Máximo de 45 caracteres			
cpfprofessor	varchar(11)	não	CPF do professor	11 caracteres numéricos			X
ruaendereçoprofessor	varchar(45)	sim	Rua em que o professor mora	Máximo de 45 caracteres			
numendereçoprofessor	int	sim	Número da residência do professor	Número Inteiro			
cidadeendereçoprofessor	varchar(45)	sim	Cidade em que o professor mora	Máximo de 45 caracteres			
titulação	varchar(20)	não	Titulação acadêmica do professor	Máximo de 20 caracteres			

### 5.2 Constraints

COLUNA	TIPO	NOME	EXPRESSÃO
matriculaprofessor	Chave primária	PK_professor	PRIMARY KEY (matriculaprofessor)
matriculaprofessor	Validação do domínio: 11 caracteres numéricos	CK_professor_matriculaprofessor	CHECK (LEN(matriculaprofessor) = 11 AND matriculaprofessor LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
cpfprofessor	Chave candidata	AK_professor_cpfprofessor	UNIQUE (cpfprofessor)
cpfprofessor	Validação do domínio: 11 caracteres numéricos	CK_professor_cpfprofessor	CHECK (LEN(cpfprofessor) = 11 AND cpfprofessor LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
ruaendereçoprofessor	Validação do domínio: Máximo de 45 caracteres	CK_professor_ruaendereçoprofessor	CHECK (LEN(ruaendereçoprofessor) <= 45)
nomeprofessor	Validação do domínio: Máximo de 45 caracteres	CK_professor_nomeprofessor	CHECK (LEN(nomeprofessor) <= 45)
cidadeendereçoprofessor	Validação do domínio: Máximo de 45 caracteres	CK_professor_cidadeendereçoprofessor	CHECK (LEN(cidadeendereçoprofessor) <= 45)
titulacao	Validação do domínio: Máximo de 20 caracteres	CK_professor_titulacao	CHECK (LEN(titulacao) <= 20)



## 6 Tabela TelefoneProfessor

Atributo multivalorado. Armazena os números de telefone dos professores que lecionam na universidade.

### 6.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
matriculaprofessor	varchar(11)	não	Matrícula do aluno	Tabela ALUNO	X	X	
telefone	varchar(15)	sim	Número do telefone	' + % [0-9] '	X		

### 6.2 Constraints

COLUNA	TIPO	NOME	EXPRESSÃO
matriculaprofessor	Chave estrangeira: Referencia matriculaprofessor na tabela Professor	FK_telefoneprofessor_matriculaprofessor	FOREIGN KEY (matriculaprofessor) REFERENCES Professor
matriculaprofessor, telefone	Chave primária	PK_telefoneprofessor	PRIMARY KEY(matriculaprofessor, telefone)
telefone	Validação do domínio: 14 ou 15 caracteres numéricos. Formato + %	CK_telefoneprofessor_telefone	CHECK (LEN(telefone) > 13 AND (telefone LIKE '+[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]' OR telefone LIKE '+[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'))

## 7 Tabela DEPENDENTE

Armazena os dados dos dependentes dos professores da universidade.

### 7.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
numdependente	varchar(2)	não	Identificador do dependente	2 caracteres numéricos		X	
nomedependente	varchar(45)	não	Nome do dependente	45 letras			
cpfdependente	varchar(11)	não	CPF do dependente	11 caracteres numéricos			X
datanascimento	date	não	Data de nascimento do dependente				
matriculaprofessor	varchar(11)	não	Matrícula do professor	11 caracteres numéricos	X	X	

### 7.2 Constraints

COLUNA	TIPO	NOME	EXPRESSÃO
matriculaprofessor	Chave estrangeira referenciando a coluna <b>matriculaprofessor</b> na tabela professor	FK_dependente_matriculaprofessor	FOREIGN KEY (matriculaprofessor) REFERENCES professor
matriculaprofessor, numdependente	Chave primária	PK_dependente	PRIMARY KEY (matriculaprofessor, numdependente)
numdependente	Validação do domínio: 2 caracteres numéricos	CK_dependente_numdependente	CHECK (LEN(numdependente) = 2 AND numdependente NOT LIKE '[^0-9]') UNIQUE (cpfdependente)
cpfdependente	Chave candidata	AK_dependente_cpfdependente	
cpfdependente	Validação do domínio: 11 caracteres numéricos	CK_dependente_cpfdependente	CHECK (LEN(cpfdependente) = 11 AND cpfdependente NOT LIKE '[^0-9]')
nomedependente	Validação do domínio: máximo 45 caracteres	CK_dependente_nomedependente	CHECK (LEN(nomedependente) <= 45)
matriculaprofessor	Validação do domínio: 11 caracteres numéricos	CK_dependente_matriculaprofessor	CHECK (LEN(matriculaprofessor) = 11)

## 8 Tabela CURSO

Armazena os dados dos cursos oferecidos pela universidade.

### 8.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
idcurso	int	não	Código do curso	Número inteiro	X		
nomecurso	varchar(45)	não	Nome do curso	45 caracteres			
duracaocurso	varchar(45)	não	Duração do curso	45 caracteres			
qtddisciplinas	int	não	Quantidade de disciplinas ofertadas pelo curso	Número inteiro			
iniociordenacao	date	não	Data em que o atual coordenador assumiu a coordenação	Data			
matriculaprofessor	varchar(11)	não	Matrícula do professor que coordena o curso	11 caracteres numéricos		X	

### 8.2 Constraints

COLUNA	TIPO	NOME	EXPRESSION
idcurso	Chave primária	PK_curso	PRIMARY KEY (idcurso)
matriculaprofessor	Chave estrangeira referenciando a coluna <b>matriculaprofessor</b> na tabela professor	FK_curso_matriculaprofessor	FOREIGN KEY (matriculaprofessor) REFERENCES Professor
nomecurso	Validação do domínio: máximo 45 caracteres	CK_curso_nomecurso	CHECK (LEN(nomecurso) <= 45)
matriculaprofessor	Chave candidata	AK_curso_matriculaprofessor	UNIQUE (matriculaprofessor)
matriculaprofessor	Validação do domínio: 11 caracteres numéricos	CK_curso_matriculaprofessor	CHECK (LEN(matriculaprofessor) = 11)
nomecurso	Validação do domínio: máximo 45 caracteres	CK_curso_nomecurso	CHECK (LEN(nomecurso) <= 45)
duracaocurso	Validação do domínio: máximo 45 caracteres	CK_curso_duracaocurso	CHECK (LEN(duracaocurso) <= 45)

## 9 Tabela DISCIPLINA

Armazena os dados relativos às disciplinas ministradas na universidade.

### 9.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
iddisciplina	int	não	ID da disciplina		X		
nomedisciplina	varchar(45)	não	Nome da disciplina				
obrigatoriedade	tinyint(1)	não	Obrigatoriedade da disciplina para conclusão do curso				
cargahorária	int	sim	Carga horária da disciplina	Não negativo			
nummaxalunos	int	sim	Número máximo de alunos a serem matriculados	Não negativo			

### 9.2 Constraints

COLUNA	TIPO	NOME	EXPRESSÃO
iddisciplina	Chave primária	PK_disciplina	PRIMARY KEY (iddisciplina)
cargahorária	Validação do domínio: Não negativo	CK_disciplina.cargahorária	CHECK (cargahorária >= 0)
nummaxalunos	Validação do domínio: Não negativo	CK_disciplina.numaxalunos	CHECK (nummaxalunos >= 0)
nomedisciplina	Validação do domínio: Máximo 45 caracteres	CK_disciplina.nomedisciplina	CHECK (nomedisciplina <= 45)

## 10 Tabela RECURSO

Entidade que armazena informações dos recursos que podem ser utilizados por um professor ao ministrar uma aula.

### 10.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
idrecurso	int	não	Identificador do recurso	Número Inteiro	X		
nomerecurso	varchar(45)	não	Nome do recurso	45 caracteres			
qtdutilizada	int	sim	Quantidade de recursos utilizados	Número Inteiro			X
versao	varchar(45)	sim	Versão do recurso	45 caracteres			
codlicenca	varchar(45)	sim	Código da licença do recurso	45 caracteres			
codpatrimonio	varchar(45)	sim	Código da patrimônio do recurso	45 caracteres			
matriculaprofessor	varchar(11)	não	Matrícula do professor	Tabela PROFESSOR	X	X	
numaula	int	não	Número da aula ministrada	Tabela AULA	X	X	

### 10.2 Constraints

COLUNA	TIPO	NOME		EXPRESSÃO
idrecurso, matriculaprofessor, numaula	Chave primária	PK_recurso		PRIMARY KEY (idrecurso, matriculaprofessor, numaula)
matriculaprofessor	Chave estrangeira: Referencia matrícula da tabela professor	FK_recurso_matriculaprofessor		FOREIGN KEY REFERENCES Professor
numaula	Chave estrangeira: Referencia número da tabela aula	FK_recurso_numaula		FOREIGN KEY REFERENCES Aula
nomerecurso	Validação do domínio: Máximo 45 caracteres	CK_recurso_nomerecurso		CHECK LEN(nomerecurso) <=45
versao	Validação do domínio: Máximo 45 caracteres	CK_recurso_ersao		CHECK LEN(versao) <=45
codlicenca	Validação do domínio: Máximo 45 caracteres	CK_recurso_codlicenca		CHECK LEN(codlicenca) <=45
codpatrimonio	Validação do domínio: Máximo 45 caracteres	CK_recurso_codpatrimonio		CHECK LEN(codpatrimonio) <=45
matriculaprofessor	Validação do domínio: 11 caracteres numéricos	CK_recurso_matriculaprofessor		CHECK LEN(matriculaprofessor) = 11

## 11 Tabela AULA

Armazena os dados relativos às aulas ministradas na universidade.

### 11.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
numaula	int	não	Número da aula ministrada	Não negativo	X		
duracao	time	sim	Duração da aula				
conteudo	varchar(45)	sim	Descrição do conteúdo ministrado				

### 11.2 Constraints

COLUNA	TIPO	NOME	EXPRESSÃO
numaula	Chave primária	PK_aula	PRIMARY KEY (numaula)
numaula	Validação do domínio: Não negativo	CK_aula_numaula	CHECK (numaula >= 0)
conteudo	Validação do domínio: Máximo de 45 caracteres	CK_aula_conteudo	CHECK (conteudo <= 45)

## 12 Tabela Local

Tabela com os espaços da Universidade, se especializa em 3 tipos: Sala de Aula, Auditório e Laboratório.

### 12.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
codlocal	int	não	Código do local	Número Inteiro	X	X	

### 12.2 Constraints

COLUNA	TIPO		NOME	EXPRESSÃO
codlocal	Chave primária		PK_local	PRIMARY KEY (codlocal)

## 13 Tabela SALA DE AULA

Armazena os dados relativos às salas de aula nas quais as aulas são ministradas.

### 13.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
codlocal	int	não	Código da sala de aula	Número Inteiro	X		
qtdecarteiras	int	sim	Quantidade de carteiras na sala	Não negativo			

### 13.2 Constraints

COLUNA	TIPO	NOME	EXPRESSÃO
codlocal	Chave estrangeira: Referencia código da tabela Local	FK_saladeaula_codlocal	FOREIGN KEY (codlocal) REFERENCES Local
codlocal	Chave primária	PK_saladeaula	PRIMARY KEY (codlocal)
qtdecarteiras	Validação do domínio: Não negativo	CK_saladeaula_qtdecarteiras	CHECK (qtdecarteiras >= 0)



## 14 Tabela LABORATÓRIO

Armazena os dados relativos aos laboratórios nos quais as aulas são ministradas.

### 14.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
codlocal	int	não	Código do laboratório		X		
qtdebandacadasexp	int	não	Quantidade de bancadas para experimentos	Não negativo			

### 14.2 Constraints

COLUNA	TIPO	NOME	EXPRESSÃO
codlocal	Chave estrangeira: Referencia código na tabela Local	FK_laboratorio_codlocal	FOREIGN KEY (codlocal) REFERENCES Local
codlocal	Chave primária	PK_laboratorio	PRIMARY KEY (codlocal)
qtdebandacadasexp	Validação do domínio: Não negativo	CK_laboratorio_qtdebandacadasexp	CHECK (qtdebandacadasexp >= 0)

## 15 Tabela AUDITÓRIO

Armazena os dados relativos aos auditórios nis quais as aulas são ministradas.

### 15.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
codlocal	int	não	Código do auditório		X		
datashow	tinyint	não	Presença ou não de datashow no auditório				
qtdmicrofones	int	não	Quantidade de microfones no auditório	Não negativo			
capacidade	int	não	Capacidade de pessoas do auditório	Não negativo			

### 15.2 Constraints

COLUNA	TIPO	NOME	EXPRESSÃO
codlocal	Chave estrangeira: Referencia código na tabela Local	FK_auditorio_codlocal	FOREING KEY (codlocal) REFERENCES Local
codlocal	Chave primária	PK_auditorio	PRIMARY KEY (codlocal)
qtdmicrofones	Validação do domínio: Não negativo	CK_auditorio_qtdmicrofones	CHECK (qtdmicrofones >= 0)
capacidade	Validação do domínio: Não negativo	CK_auditorio_capacidade	CHECK (capacidade >= 0)

## 16 Tabela PROJPEQUISA

Armazena informações sobre os projetos de pesquisa desenvolvidos na universidade.

### 16.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
codprojeto	varchar(11)	não	Código do projeto	Máximo 11 caracteres	X		
nomeprojeto	varchar(45)	não	Nome do projeto	Máximo 45 caracteres			
descrição	varchar(100)	não	Descrição do projeto	Máximo 100 caracteres			
datainicio	date	não	Data de início do projeto	DD/MM/AAAA			
bolsa	tinyint	não	Existência de bolsista no projeto	0 ou 1			

### 16.2 Constraints

COLUNA	TIPO	NOME	EXPRESSÃO
codprojeto	Chave primária	PK_projpesquisa	PRIMARY KEY (codprojeto)
codprojeto	Validação do domínio: Máximo 11 caracteres	CK_projpesquisa_codprojeto	CHECK LEN(codprojeto) <=11
nomeprojeto	Validação do domínio: Máximo 45 caracteres	CK_projpesquisa_nomeprojeto	CHECK LEN(nomeprojeto) <=45
descricao	Validação do domínio: Máximo 100 caracteres	CK_projpesquisa_descricao	CHECK LEN(descricao) <=100

## 17 Tabela CURSA

Relacionamento N:N entre as entidades ALUNO e DISCIPLINA. Armazena as informações, como notas, das disciplinas cursadas por um aluno em determinado período.

### 17.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
matriculaaluno	varchar(11)	não	Matrícula do aluno	Tabela ALUNO	X	X	
iddisciplina	int	não	ID da disciplina	Tabela DISCIPLINA	X	X	
semestre	varchar(6)	não	Semestre em que o aluno cursou a disciplina	Formato XXXX.X	X		
nota1	int	DEFAULT 0	Nota do aluno na primeira avaliação	Valor entre 0 e 100			
nota2	int	DEFAULT 0	Nota do aluno na segunda avaliação	Valor entre 0 e 100			
nota3	int	DEFAULT 0	Nota do aluno na terceira avaliação	Valor entre 0 e 100			

### 17.2 Constraints

COLUNA	TIPO	NOME	EXPRESSÃO
matriculaaluno	Chave estrangeira: Referencia matriculaaluno da tabela aluno	FK_cursa_matriculaaluno	FOREIGN KEY (matriculaaluno) REFERENCES Aluno
iddisciplina	Chave estrangeira: Referencia iddisciplina na tabela Disciplina	FK_cursa_iddisciplina	FOREIGN KEY (iddisciplina) REFERENCES Disciplina
matriculaaluno, iddisciplina, semestre	Chave primária	PK_cursa	PRIMARY KEY (matriculaaluno, iddisciplina, semestre)
semestre	Validação do domínio: 4 dígitos seguido de ',' mais 1 dígito	CK_cursa_semestre	CHECK ((LEN(semestre) = 6) AND (semestre LIKE '%.[1-2]'))
nota1	Validação do domínio: Valor entre 0 e 100	CK_cursa_nota1	CHECK (nota1 >= 0 AND nota1 <= 100)
nota2	Validação do domínio: Valor entre 0 e 100	CK_cursa_nota2	CHECK (nota2 >= 0 AND nota2 <= 100)
nota3	Validação do domínio: Valor entre 0 e 100	CK_cursa_nota3	CHECK (nota3 >= 0 AND nota3 <= 100)

## 18 Tabela OCUPA

Relacionamento N:N entre as entidades DISCIPLINA e LOCAL (Entidade genérica para LABORATÓRIO, AUDITÓRIO E SALA DE AULA).

### 18.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
codlocal	int	não	Código do Local	Tabela LOCAL	X	X	
iddisciplina	int	não	ID da disciplina	Tabela DISCIPLINA	X	X	

### 18.2 Constraints

COLUNA	TIPO	NOME	EXPRESSÃO
codlocal	Chave estrangeira: Referencia codlocal na tabela Local	FK_ocupa_codlocal	FOREIGN KEY (codlocal) REFERENCES Local
iddisciplina	Chave estrangeira: Referencia iddisciplina da tabela Disciplina	FK_ocupa_iddisciplina	FOREIGN KEY (iddisciplina) REFERENCES Disciplina
codlocal	Chave primária	PK_ocupa	PRIMARY KEY (codlocal, iddisciplina)

## 19 Tabela DISPENSA

Relacionamento ternário n:n:n entre as entidades ALUNO, PROFESSOR e DISCIPLINA.

### 19.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
matriculaaluno	varchar(11)	não	Matrícula do aluno	Tabela ALUNO	X	X	
matriculaprofessor	varchar(11)	não	Matrícula do professor	Tabela PROFESSOR	X	X	
iddisciplina	int	não	ID da disciplina	Tabela DISCIPLINA	X	X	

### 19.2 Constraints

COLUNA	TIPO	NOME	EXPRESSÃO
matriculaaluno	Chave estrangeira: Referencia matriculaaluno na tabela Aluno	FK_dispensa_matriculaaluno	FOREIGN KEY (matriculaaluno) REFERENCES Aluno
matriculaprofessor	Chave estrangeira: Referencia matriculaprofessor na tabela Professor	FK_dispensa_matriculaprofessor	FOREIGN KEY (matriculaprofessor) REFERENCES Professor
iddisciplina	Chave estrangeira: Referencia iddisciplina na tabela Disciplina	FK_dispensa_iddisciplina	FOREIGN KEY (iddisciplina) REFERENCES Disciplina
matriculaaluno, matriculaprofessor, iddisciplina	Chave primária	PK_dispensa	PRIMARY KEY (matriculaaluno, matriculaprofessor, iddisciplina)
matriculaprofessor	Validação do domínio: 11 caracteres numéricos	CK_dispensa_matriculaprofessor	CHECK LEN(matriculaprofessor) = 11
matriculaaluno	Validação do domínio: 11 caracteres numéricos	CK_dispensa_matriculaaluno	CHECK LEN(matriculaaluno) = 11

## 20 Tabela MINISTRA

Entidade associativa entre as entidades PROFESSOR, AULA e RECURSO. Armazena os dados de quem ministrou cada aula, e dos recursos que podem ter sido utilizados durante as aulas.

### 20.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
matriculaprofessor	varchar(11)	não	Matrícula do professor	Tabela PROFESSOR	X	X	
numaula	int	não	Código da aula	Tabela AULA	X	X	
dataaula	date	não	Data da aula	DD/MM/AAAA	X	X	

### 20.2 Constraints

COLUNA	TIPO	NOME	EXPRESSÃO
matriculaprofessor	Chave estrangeira: Referencia matrícula na tabela Professor	FK_ministra_matriculaprofessor	FOREIGN KEY(matriculaprofessor) REFERENCES Professor
numaula	Chave estrangeira: Referencia número na tabela Aula	FK_ministra_numaula	FOREIGN KEY(numaula) REFERENCES Aula
matriculaprofessor, numaula	PK_ministra	Chave primária	PRIMARY KEY (matriculaprofessor,numaula)
matriculaprofessor	Validação do domínio: 11 caracteres numéricos	CK_ministra_matriculaprofessor	CHECK LEN(matriculaprofessor) = 11

## 21 Tabela DESENVOLVEM

Relacionamento ternário entre as entidades ALUNO, PROFESSOR e PROJESQUISA. Armazena o professor e aluno em cada projeto.

### 21.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
codprojeto	varchar(11)	não	Código do projeto	Tabela PROJESQUISA	X	X	
matriculaaluno	varchar(11)	não	Matrícula do aluno	Tabela ALUNO	X	X	
matriculaprofessor	varchar(11)	não	Matrícula do professor	Tabela PROFESSOR	X	X	

### 21.2 Constraints



COLUNA	TIPO	NOME	EXPRESSÃO
codprojeto	Chave estrangeira: Referencia código na tabela ProjPesquisa	FK_desenvolvem_codprojeto	FOREIGN KEY (codprojeto) REFERENCES ProjPesquisa
matriculaaluno	Chave estrangeira: Referencia matriculaaluno na tabela Aluno	FK_desenvolvem_matriculaaluno	FOREIGN KEY (matriculaaluno) REFERENCES Aluno
matriculaprofessor	Chave estrangeira: Referencia matriculaprofessor na tabela Professor	FK_desenvolvem_matriculaprofessor	FOREIGN KEY (matriculaprofessor) REFERENCES Professor
codprojeto, matriculaaluno, matriculaprofessor	Chave primária	PK_desenvolvem	PRIMARY KEY(codprojeto, matriculaaluno, matriculaprofessor)
matriculaprofessor	Validação do domínio: 11 caracteres numéricos	CK_desenvolvem_matriculaprofessor	CHECK LEN(matriculaprofessor) = 11
matriculaaluno	Validação do domínio: 11 caracteres numéricos	CK_desenvolvem_matriculaaluno	CHECK LEN(matriculaaluno) = 11
codprojeto	Validação do domínio: 11 caracteres numéricos	CK_desenvolvem_codprojeto	CHECK LEN(codprojeto) <= 11

## 22 Tabela CONTEM

Relacionamento n:n entre as entidades CURSO e DISCIPLINA.

### 22.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
idcurso	int	não	ID do curso	Tabela CURSO	X	X	
iddisiplina	int	não	ID da disciplina	Tabela DISCIPLINA	X	X	

### 22.2 Constraints

COLUNA	TIPO	NOME	EXPRESSÃO
idcurso	Chave estrangeira: Referencia idcurso na tabela Curso	FK_contem_idcurso	FOREIGN KEY (idcurso) REFERENCES Curso
iddisiplina	Chave estrangeira: Referencia iddisciplina na tabela Disciplina	FK_contem_iddisciplina	FOREIGN KEY (iddisciplina) REFERENCES Disciplina
idcurso, iddisciplina	Chave primária	PK_curso	PRIMARY KEY (idcurso, iddisciplina)

## 23 Tabela PREREQUISITO

Relacionamento n:n da entidade DISCIPLINA consigo mesma. Representa as disciplinas que devem ser cursadas antes de outras.

### 23.1 Atributos

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					P	E	C
prerequisito	int	não	ID da disciplina que deve ser cursada antes	Tabela DISCIPLINA	X	X	
disciplinadesejada	int	não	ID da disciplina a ser cursada	Tabela DISCIPLINA	X	X	

### 23.2 Constraints

COLUNA	TIPO	NOME	EXPRESSÃO
prerequisito	Chave estrangeira: Referencia iddisciplina na tabela Disciplina	FK_prerequisito_prerequisito	FOREIGN KEY (prerequisito) REFERENCES Disciplina
disciplinadesejada	Chave estrangeira: Referencia iddisciplina na tabela Disciplina	FK_prerequisito_disciplinadesejada	FOREIGN KEY (disciplinadesejada) REFERENCES Disciplina
prerequisito, disciplinadesejada	Chave primária	PK_prerequisito	PRIMARY KEY (prerequisito, disciplinadesejada)

## 24 Script do Banco com Modificações Pedidas

```
/** Criação do banco de dados */
CREATE DATABASE bd_Universidade
GO
USE bd_Universidade
GO

/** Criação da tabela Aluno */
CREATE TABLE Aluno
(
    matriculaaluno varchar(11) NOT NULL,
    nomealuno varchar(45) NOT NULL,
    cpfaluno varchar(11) NOT NULL,
    ruaenderecoaluno varchar(45) NULL,
    numenderecoaluno int NULL,
    cidadeenderecoaluno varchar(45) NULL,

    emailaluno varchar(40) NOT NULL,
    CONSTRAINT PK_aluno PRIMARY KEY (matriculaaluno),
    CONSTRAINT CK_aluno_matriculaaluno CHECK (LEN(matriculaaluno) = 11 AND matriculaaluno LIKE '
        [0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
    CONSTRAINT AK_aluno_cpfaluno UNIQUE (cpfaluno),
    CONSTRAINT CK_aluno_cpfaluno CHECK (LEN(cpfaluno) = 11 AND cpfaluno LIKE '
        [0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
    CONSTRAINT CK_aluno_emailaluno CHECK (emailaluno LIKE '%@%.%' AND emailaluno NOT LIKE '%@%'),
    CONSTRAINT CK_aluno_ruaenderecoaluno CHECK (LEN(ruaenderecoaluno) <= 45),
    CONSTRAINT CK_aluno_nomealuno CHECK (LEN(nomealuno) <= 45),
    CONSTRAINT CK_aluno_cidadeenderecoaluno CHECK (LEN(cidadeenderecoaluno) <= 45)
)

/** Criação da Tabela TelefoneAluno */
CREATE TABLE TelefoneAluno
(
    matriculaaluno varchar(11) NOT NULL,
    telefone varchar(15) NOT NULL,
    CONSTRAINT FK_telefonealuno_matriculaaluno FOREIGN KEY(matriculaaluno) REFERENCES Aluno,
    CONSTRAINT PK_telefonealuno PRIMARY KEY (matriculaaluno, telefone),
    CONSTRAINT CK_telefonealuno_telefone CHECK (LEN(telefone) > 13 AND (telefone LIKE '
        +[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]' OR telefone LIKE '
        +[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'))
)

/** Criação da tabela Professor */
CREATE TABLE Professor
(
    matriculaprofessor varchar(11) NOT NULL,
    nomeprofessor varchar(45) NOT NULL,
    cpfprofessor varchar(11) NOT NULL,
    ruaenderecoprofessor varchar(45) NULL,
    numenderecoprofessor int NULL,
    cidadeenderecoprofessor varchar(45) NULL,

    titulacao varchar(20) NOT NULL,
    CONSTRAINT PK_professor PRIMARY KEY (matriculaprofessor),
    CONSTRAINT CK_professor_matriculaprofessor CHECK (LEN(matriculaprofessor) = 11 AND
        matriculaprofessor LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
```

```

CONSTRAINT AK_professor_cpfprofessor UNIQUE (cpfprofessor),
CONSTRAINT CK_professor_cpfprofessor CHECK (LEN(cpfprofessor) = 11 AND cpfprofessor LIKE '
    [0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
CONSTRAINT CK_professor_ruaendereçoprofessor CHECK (LEN(ruaendereçoprofessor) <= 45),
CONSTRAINT CK_professor_nomeprofessor CHECK (LEN(nomeprofessor) <= 45),
CONSTRAINT CK_professor_cidadeendereçoprofessor CHECK (LEN(cidadeendereçoprofessor) <= 45),
CONSTRAINT CK_professor_titulacao CHECK (LEN(titulacao) <= 20)
)

/**Criação da Tabela TelefoneProfessor**/
CREATE TABLE TelefoneProfessor
(
    matriculaprofessor varchar(11) NOT NULL,
    telefone varchar(15) NOT NULL,
    CONSTRAINT FK_telefoneprofessor_matriculaprofessor FOREIGN KEY(matriculaprofessor) REFERENCES
        Professor,
    CONSTRAINT PK_telefoneprofessor PRIMARY KEY (matriculaprofessor, telefone),
    CONSTRAINT CK_telefoneprofessor_telefone CHECK (LEN(telefone) > 13 AND (telefone LIKE '
        +[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]' OR telefone LIKE '
        +[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'))
)

/** Criação da tabela Dependente **/
CREATE TABLE Dependente
(
    numdependente varchar(2) NOT NULL,
    nomedependente varchar(45) NOT NULL,
    cpfdependente varchar(11) NOT NULL,
    datanascimento date NOT NULL,
    matriculaprofessor varchar(11) NOT NULL,
    CONSTRAINT FK_dependente_matriculaprofessor FOREIGN KEY (matriculaprofessor) REFERENCES
        Professor,
    CONSTRAINT PK_dependente PRIMARY KEY (numdependente, matriculaprofessor),
    CONSTRAINT CK_dependente_numdependente CHECK (LEN(numdependente) = 2 AND numdependente LIKE '
        [0-9][0-9]'),
    CONSTRAINT AK_dependente_cpfdependente UNIQUE (cpfdependente),
    CONSTRAINT CK_dependente_cpfdependente CHECK (LEN(cpfdependente) = 11 AND cpfdependente LIKE '
        [0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
    CONSTRAINT CK_dependente_nomedependente CHECK (LEN(nomedependente) <= 45),
    CONSTRAINT CK_dependente_matriculaprofessor CHECK (LEN(matriculaprofessor) = 11)
)

/** Criação da tabela Curso **/
CREATE TABLE Curso
(
    idcurso int NOT NULL,
    nomecurso varchar(45) NOT NULL,
    duracaocurso varchar(45) NOT NULL,
    qtddisciplinas int NOT NULL,
    matriculaprofessor varchar(11) NOT NULL,
    iniciocoordenacao date NOT NULL,
    CONSTRAINT PK_curso PRIMARY KEY (idcurso),
    CONSTRAINT FK_curso_matriculaprofessor FOREIGN KEY (matriculaprofessor) REFERENCES Professor,
    CONSTRAINT AK_curso_matriculaprofessor UNIQUE (matriculaprofessor),
    CONSTRAINT CK_curso_matriculaprofessor CHECK (LEN(matriculaprofessor) = 11),
    CONSTRAINT CK_curso_nomecurso CHECK (LEN(nomecurso) <= 45),
    CONSTRAINT CK_curso_duracaocurso CHECK (LEN(duracaocurso) <= 45)
)

/** Criação da tabela Disciplina **/
CREATE TABLE Disciplina

```

```

(
    iddisciplina int NOT NULL,
    nomedisciplina varchar(45) NOT NULL,
    obrigatoriedade tinyint NOT NULL,
    cargahoraria int NULL,
    nummaxalunos int NULL,
    CONSTRAINT PK_disciplina PRIMARY KEY (iddisciplina),
    CONSTRAINT CK_disciplina_cargahoraria CHECK (cargahoraria >= 0),
    CONSTRAINT CK_disciplina_nummaxalunos CHECK (nummaxalunos > 0),
    CONSTRAINT CK_disciplina_nomedisciplina CHECK (LEN(nomedisciplina) <= 45)
)

/** Criação da tabela Aula */
CREATE TABLE Aula
(
    numaula int NOT NULL,
    duracao time NULL,
    conteudo varchar(45) NULL,
    CONSTRAINT PK_aula PRIMARY KEY (numaula),
    CONSTRAINT CK_aula_numaula CHECK (numaula >= 0),
    CONSTRAINT CK_aula_conteudo CHECK (LEN(conteudo) <= 45)
)

/** Criação da tabela Recurso */
CREATE TABLE Recurso
(
    idrecurso int NOT NULL,
    nomerecurso varchar(45) NOT NULL,
    qtdutilizada int NULL,
    versao varchar(45) NULL,
    codlicenca varchar(45) NULL,
    codpatrimonio varchar(45) NULL,
    matriculaprofessor varchar(11) NOT NULL,
    numaula int NOT NULL,
    CONSTRAINT PK_recurso PRIMARY KEY (idrecurso, matriculaprofessor, numaula),
    CONSTRAINT FK_recurso_matriculaprofessor FOREIGN KEY (matriculaprofessor) REFERENCES Professor
,
    CONSTRAINT FK_recurso_numaula FOREIGN KEY (numaula) REFERENCES Aula,
    CONSTRAINT CK_recurso_nomecurso CHECK (LEN(nomerecurso) <= 45),
    CONSTRAINT CK_recurso_versao CHECK (LEN(versao) <= 45),
    CONSTRAINT CK_recurso_codlicenca CHECK (LEN(codlicenca) <= 45),
    CONSTRAINT CK_recurso_codpatrimonio CHECK (LEN(codpatrimonio) <= 45),
    CONSTRAINT CK_recurso_matriculaprofessor CHECK (LEN(matriculaprofessor) = 11)
)

/** Criação da tabela Local */
CREATE TABLE Local
(
    codlocal int NOT NULL,
    CONSTRAINT PK_oLocal PRIMARY KEY (codlocal),
)

/** Criação da tabela SalaDeAula */
CREATE TABLE SalaDeAula
(
    codlocal int NOT NULL,
    qtdecarteiras int NULL,
    CONSTRAINT PK_saladeaula PRIMARY KEY (codlocal),
    CONSTRAINT FK_saladeaula_codlocal FOREIGN KEY (codlocal) REFERENCES Local,
    CONSTRAINT CK_saladeaula_qtdecarteiras CHECK (qtdecarteiras >= 0)
)

```

```

/** Criação da tabela Laboratorio */
CREATE TABLE Laboratorio
(
    codlocal int NOT NULL,
    qtdebancadasexp int NULL,
    CONSTRAINT PK_laboratorio PRIMARY KEY (codlocal),
    CONSTRAINT FK_laboratorio_codlocal FOREIGN KEY (codlocal) REFERENCES Local,
    CONSTRAINT CK_laboratorio_qtdebancadasexp CHECK (qtdebancadasexp >= 0)
)

/** Criação da tabela Auditorio */
CREATE TABLE Auditorio
(
    codlocal int NOT NULL,
    datashow tinyint NOT NULL,
    qtdemicrofones int NOT NULL,
    capacidade int NOT NULL,
    CONSTRAINT PK_auditorio PRIMARY KEY (codlocal),
    CONSTRAINT FK_auditorio_codlocal FOREIGN KEY (codlocal) REFERENCES Local,
    CONSTRAINT CK_auditorio_qtdemicrofones CHECK (qtdemicrofones >= 0),
    CONSTRAINT CK_auditorio_capacidade CHECK (capacidade >= 0)
)

/** Criação da tabela ProjPesquisa */
CREATE TABLE ProjPesquisa
(
    codprojeto varchar(11) NOT NULL,
    nomeprojeto varchar(45) NOT NULL,
    descricao varchar(100) NOT NULL,
    datainicio date NOT NULL,
    bolsa tinyint NOT NULL,
    CONSTRAINT PK_projeto PRIMARY KEY (codprojeto),
    CONSTRAINT CK_projpesquisa_codprojeto CHECK (LEN(codprojeto) <= 11),
    CONSTRAINT CK_projpesquisa_nomeprojeto CHECK (LEN(nomeprojeto) <= 45),
    CONSTRAINT CK_projpesquisa_descricao CHECK (LEN(descricao) <= 100)
)

/** Criação da tabela Cursa */
CREATE TABLE Cursa
(
    matriculaaluno varchar(11) NOT NULL,
    iddisciplina int NOT NULL,
    semestre varchar(6) NOT NULL,
    nota1 int DEFAULT 0,
    nota2 int DEFAULT 0,
    nota3 int DEFAULT 0,
    CONSTRAINT FK_cursa_matriculaaluno FOREIGN KEY (matriculaaluno) REFERENCES Aluno,
    CONSTRAINT FK_cursa_iddisciplina FOREIGN KEY (iddisciplina) REFERENCES Disciplina,
    CONSTRAINT PK_cursa PRIMARY KEY (matriculaaluno, iddisciplina, semestre),
    CONSTRAINT CK_cursa_semestre CHECK ((LEN(semestre) = 6) AND (semestre LIKE '%.[1-2]')),
    CONSTRAINT CK_nota1 CHECK (nota1 >= 0 and nota1 <=100),
    CONSTRAINT CK_nota2 CHECK (nota2 >= 0 and nota2 <=100),
    CONSTRAINT CK_nota3 CHECK (nota3 >= 0 and nota3 <=100)
)

/** Criação da tabela Ocupa */
CREATE TABLE Ocupa
(
    codlocal int NOT NULL,
    iddisciplina int NOT NULL,

    CONSTRAINT FK_ocupa_codlocal FOREIGN KEY (codlocal) REFERENCES Local,

```

```

CONSTRAINT FK_ocupa_iddisciplina FOREIGN KEY (iddisciplina) REFERENCES Disciplina,
CONSTRAINT PK_ocupa PRIMARY KEY (codlocal, iddisciplina)
)

/**** Criação da tabela Dispensa ****/
CREATE TABLE Dispensa
(
    matriculaaluno varchar(11) NOT NULL,
    matriculaprofessor varchar(11) NOT NULL,
    iddisciplina int NOT NULL,
    CONSTRAINT FK_dispensa_matriculaaluno FOREIGN KEY (matriculaaluno) REFERENCES Aluno,
    CONSTRAINT FK_dispensa_matriculaprofessor FOREIGN KEY (matriculaprofessor) REFERENCES
        Professor,
    CONSTRAINT FK_dispensa_iddisciplina FOREIGN KEY (iddisciplina) REFERENCES Disciplina,
    CONSTRAINT PK_dispensa PRIMARY KEY (matriculaprofessor, iddisciplina, matriculaaluno),
    CONSTRAINT CK_dispensa_matriculaaluno CHECK (LEN(matriculaaluno) = 11),
    CONSTRAINT CK_dispensa_matriculaprofessor CHECK (LEN(matriculaprofessor) = 11)
)

/**** Criação da tabela Ministra ****/
CREATE TABLE Ministra
(
    matriculaprofessor varchar(11) NOT NULL,
    numaula int NOT NULL,
    dataaula date NOT NULL,
    CONSTRAINT FK_ministra_matriculaprofessor FOREIGN KEY (matriculaprofessor) REFERENCES
        Professor,
    CONSTRAINT FK_ministra_numaula FOREIGN KEY (numaula) REFERENCES Aula,
    CONSTRAINT PK_ministra PRIMARY KEY (matriculaprofessor, numaula),
    CONSTRAINT CK_ministra_matriculaprofessor CHECK (LEN(matriculaprofessor) = 11)
)

/**** Criação da tabela Desenvolvem ****/
CREATE TABLE Desenvolvem
(
    matriculaaluno varchar(11) NOT NULL,
    matriculaprofessor varchar(11) NOT NULL,
    codprojeto varchar(11) NOT NULL,
    CONSTRAINT FK_desenvolvem_matriculaaluno FOREIGN KEY (matriculaaluno) REFERENCES Aluno,
    CONSTRAINT FK_desenvolvem_matriculaprofessor FOREIGN KEY (matriculaprofessor) REFERENCES
        Professor,
    CONSTRAINT FK_desenvolvem_codprojeto FOREIGN KEY (codprojeto) REFERENCES ProjPesquisa,
    CONSTRAINT PK_desenvolvem PRIMARY KEY (matriculaprofessor, matriculaaluno, codprojeto),
    CONSTRAINT CK_desenvolvem_matriculaaluno CHECK (LEN(matriculaaluno) = 11),
    CONSTRAINT CK_desenvolvem_matriculaprofessor CHECK (LEN(matriculaprofessor) = 11),
    CONSTRAINT CK_desenvolvem_codprojeto CHECK (LEN(codprojeto) <= 11)
)

/**** Criação da tabela Contem ****/
CREATE TABLE Contem
(
    idcurso int NOT NULL,
    iddisciplina int NOT NULL,
    CONSTRAINT FK_contem_idcurso FOREIGN KEY (idcurso) REFERENCES Curso,
    CONSTRAINT FK_contem_iddisciplina FOREIGN KEY (iddisciplina) REFERENCES Disciplina,
    CONSTRAINT PK_contem PRIMARY KEY (idcurso, iddisciplina),
)

```



```

/**** Criação da tabela PreRequisito ****/
CREATE TABLE PreRequisito
(
    prerequisite int NOT NULL,
    disciplinadesejada int NOT NULL,
    CONSTRAINT FK_prerequisito_prerequisito FOREIGN KEY (prerequisito) REFERENCES Disciplina,
    CONSTRAINT FK_prerequisito_disciplinadesejada FOREIGN KEY (disciplinadesejada) REFERENCES
        Disciplina,
    CONSTRAINT PK_prerequisito PRIMARY KEY (prerequisito, disciplinadesejada),
)

SET DATEFORMAT 'DMY'

/**** Inserção de dados na tabela Aluno ****/
INSERT INTO Aluno VALUES ('20152610001','Laura_Avelar', '45893468710', 'Horacio_Trajano', 841, '
    João_Pessoa', 'lauravelar@gmail.com')
INSERT INTO Aluno VALUES ('20152610002','Cecilia_Fonseca', '12193477820', 'Avenida_Rui_Carneiro'
    , 310, 'João_Pessoa', 'ceciliafonseca@gmail.com')
INSERT INTO Aluno VALUES ('20152610003','Carlos_Almeida', '35976148560', 'Josefa_Taveira', 283,
    'João_Pessoa', 'carlosalmeida@gmail.com')
INSERT INTO Aluno VALUES ('20152610004','Alberto_Ferreira', '96486123348', 'Avenida_Primeiro_de_
    Maio', 897, 'João_Pessoa', 'albertoferreira@gmail.com')
INSERT INTO Aluno VALUES ('20152610005','Beatriz_Nobrega', '54086495213', 'Avenida_Guaraira',
    654, 'João_Pessoa', 'beatriznobrega@gmail.com')

/**** Inserção de dados na tabela TelefoneAluno ****/
INSERT INTO TelefoneAluno VALUES ('20152610001', '+55083996805941')
INSERT INTO TelefoneAluno VALUES ('20152610002', '+55083999885648')
INSERT INTO TelefoneAluno VALUES ('20152610003', '+55083999334688')
INSERT INTO TelefoneAluno VALUES ('20152610004', '+55083988869641')
INSERT INTO TelefoneAluno VALUES ('20152610005', '+55083988468797')
INSERT INTO TelefoneAluno VALUES ('20152610005', '+55083999554682')

/**** Inserção de dados na tabela Professor ****/
INSERT INTO Professor VALUES ('20101610100', 'Silvana_Cunha', '56879463522', 'João_Cabral_de_
    Lucena', 321, 'João_Pessoa', 'Doutorado')
INSERT INTO Professor VALUES ('20101610200', 'Pedro_Gonçalves', '76485297602', 'Av._Gov._
    Argemiro_de_Figueiredo', 51, 'João_Pessoa', 'Doutorado')
INSERT INTO Professor VALUES ('20101610300', 'Evandro_dos_Santos', '44897588620', 'Av._Camilo_de
    Holanda', 896, 'João_Pessoa', 'Doutorado')
INSERT INTO Professor VALUES ('20101610400', 'Alberto_Soares', '94862158797', 'Av._Hilton_Souto_
    Maior', 020, 'João_Pessoa', 'Mestrado')
INSERT INTO Professor VALUES ('20101610500', 'Giovana_Porto', '25168794350', 'Av._Manoel_Deodato
    ', 112, 'João_Pessoa', 'Doutorado')

/**** Inserção de dados na tabela TelefoneProfessor ****/
INSERT INTO TelefoneProfessor VALUES ('20101610100', '+55083999687458')
INSERT INTO TelefoneProfessor VALUES ('20101610200', '+55083988796428')
INSERT INTO TelefoneProfessor VALUES ('20101610300', '+55083996778451')
INSERT INTO TelefoneProfessor VALUES ('20101610400', '+55083988976553')
INSERT INTO TelefoneProfessor VALUES ('20101610500', '+55083999339317')
INSERT INTO TelefoneProfessor VALUES ('20101610500', '+55083988869641')

/**** Inserção de dados na tabela Dependente ****/
INSERT INTO Dependente VALUES ('01', 'Claudia_Soares', '78946532501', '03/10/1998', '20101610400
    ')
INSERT INTO Dependente VALUES ('02', 'Alvaro_Soares', '86479588420', '21/07/1995', '20101610400'
    )
INSERT INTO Dependente VALUES ('03', 'Milena_Cunha', '64895756210', '15/10/2000', '20101610100')
INSERT INTO Dependente VALUES ('04', 'Gabriela_Porto', '89746527886', '15/12/1993', '20101610500
    ')
INSERT INTO Dependente VALUES ('05', 'Valeria_dos_Santos', '56798436511', '05/11/1999', '
    20101610300')

```

```

/**** Inserção de dados na tabela Curso ****/
INSERT INTO Curso VALUES (1, 'Engenharia_Elétrica', '5_anos', 56, '20101610100', '15/04/2017')
INSERT INTO Curso VALUES (2, 'Sistemas_para_Internet', '3_anos', 36, '20101610200', '10/08/2012')
INSERT INTO Curso VALUES (3, 'Administração', '4_anos', 48, '20101610300', '21/12/2012')
INSERT INTO Curso VALUES (4, 'História', '4_anos', 48, '20101610400', '07/03/2011')
INSERT INTO Curso VALUES (5, 'Engenharia_Mecânica', '5_anos', 56, '20101610500', '05/03/2014')

/**** Inserção de dados na tabela Disciplina ****/
INSERT INTO Disciplina VALUES (60, 'Calculo_1', 1, 100, 50)
INSERT INTO Disciplina VALUES (61, 'Algebra_Vetorial', 1, 80, 40)
INSERT INTO Disciplina VALUES (62, 'Algebra_Linear', 1, 100, 40)
INSERT INTO Disciplina VALUES (63, 'Calculo_2', 0, 80, 40)
INSERT INTO Disciplina VALUES (64, 'Projeto_de_Filtros_e_Amplificadores', 0, 80, 20)
INSERT INTO Disciplina VALUES (65, 'Banco_de_Dados', 0, 80, 40)

/**** Inserção de dados na tabela Aula ****/
INSERT INTO Aula VALUES (1, '01:40:00', 'Apresentação_da_Disciplina_-_Cálculo_1')
INSERT INTO Aula VALUES (2, '02:30:00', 'Revisão_de_Funções')
INSERT INTO Aula VALUES (3, NULL, NULL)
INSERT INTO Aula VALUES (4, '02:30:00', 'Limites_em_Funções_de_uma_Variável')
INSERT INTO Aula VALUES (5, '01:40:00', 'Exercícios_-_Limites')
INSERT INTO Aula VALUES (6, '01:40:00', 'Apresentação_da_Disciplina_-_Álgebra_Linear')
INSERT INTO Aula VALUES (7, '01:40:00', 'Apresentação_da_Disciplina_-_Banco_de_Dados')
INSERT INTO Aula VALUES (8, '02:30:00', 'Apresentação_da_Disciplina_-_Algebra_Vetorial')

/**** Inserção de dados na tabela Recurso ****/
INSERT INTO Recurso VALUES (1, 'Caixa_de_Som_Portátil', 2, NULL, NULL, 'DEEZXC', '20101610100', 1)
INSERT INTO Recurso VALUES (2, 'Datashow', 1, NULL, NULL, 'DTI23490', '20101610200', 7)
INSERT INTO Recurso VALUES (3, 'MATLAB', NULL, 'R2014b', NULL, 'DEES879', '20101610300', 6)
INSERT INTO Recurso VALUES (4, 'GeogebraDeluxe', NULL, NULL, NULL, 'DMES230', '20101610400', 8)
INSERT INTO Recurso VALUES (5, 'GeogebraDeluxe', 1, NULL, NULL, 'DMES230', '20101610100', 2)

/**** Inserção de dados na tabela Local ****/
INSERT INTO Local VALUES (101)
INSERT INTO Local VALUES (102)
INSERT INTO Local VALUES (103)
INSERT INTO Local VALUES (104)
INSERT INTO Local VALUES (105)

INSERT INTO Local VALUES (201)
INSERT INTO Local VALUES (202)
INSERT INTO Local VALUES (203)
INSERT INTO Local VALUES (204)
INSERT INTO Local VALUES (205)

INSERT INTO Local VALUES (301)
INSERT INTO Local VALUES (302)
INSERT INTO Local VALUES (303)
INSERT INTO Local VALUES (304)
INSERT INTO Local VALUES (305)

/**** Inserção de dados na tabela SalaDeAula ****/
INSERT INTO SalaDeAula VALUES (101, 30)
INSERT INTO SalaDeAula VALUES (102, 25)
INSERT INTO SalaDeAula VALUES (103, 30)
INSERT INTO SalaDeAula VALUES (104, 40)
INSERT INTO SalaDeAula VALUES (105, 20)

/**** Inserção de dados na tabela Laboratorio ****/

```

```

INSERT INTO Laboratorio VALUES (201, 10)
INSERT INTO Laboratorio VALUES (202, 5)
INSERT INTO Laboratorio VALUES (203, 8)
INSERT INTO Laboratorio VALUES (204, 5)
INSERT INTO Laboratorio VALUES (205, 6)

/**** Inserção de dados na tabela Auditorio ****/
INSERT INTO Auditorio VALUES (301, 1, 2, 60)
INSERT INTO Auditorio VALUES (302, 1, 4, 100)
INSERT INTO Auditorio VALUES (303, 1, 2, 50)
INSERT INTO Auditorio VALUES (304, 1, 1, 65)
INSERT INTO Auditorio VALUES (305, 1, 4, 80)

/**** Inserção de dados na tabela ProjPesquisa ****/
INSERT INTO ProjPesquisa VALUES ('GPDS_111127', 'Espaço de Fases do Coração', 'Fazer a relação com o uso de machine learning do espaço de fases com cardiopatias', '15/05/2018', 1)
INSERT INTO ProjPesquisa VALUES ('GPDS_111128', 'Avaliação de Janelas em Filtros FIR', 'Avaliar a eficiência de diferentes janelas em filtros digitais de resposta finita', '07/03/2018', 0)
INSERT INTO ProjPesquisa VALUES ('GPDS_111129', 'Comunicações Moleculares', 'Estabelecer um modelo de comunicação por moléculas no corpo humano', '28/01/2018', 0)
INSERT INTO ProjPesquisa VALUES ('LINSCA_5789', 'Avaliação do Pré-Cálculo em Calouros', 'Comparar o desempenho de turmas de cálculo que tiveram curso pré-cálculo com as que não o tiveram', '17/04/2017', 1)
INSERT INTO ProjPesquisa VALUES ('GREL_578690', 'Laboratório de robótica no IFPB', 'Avaliar o equipamento e espaço necessário para a implantação de um laboratório de robótica', '14/08/2016', 1)

/**** Inserção de dados na tabela Cursa ****/
INSERT INTO Cursa VALUES ('20152610001', 60, '2018.1', DEFAULT, DEFAULT, DEFAULT)
INSERT INTO Cursa VALUES ('20152610001', 61, '2018.1', 80, 70, DEFAULT)
INSERT INTO Cursa VALUES ('20152610004', 65, '2018.1', DEFAULT, DEFAULT, DEFAULT)
INSERT INTO Cursa VALUES ('20152610004', 60, '2017.2', 70, 80, 70)
INSERT INTO Cursa VALUES ('20152610003', 61, '2016.2', 100, 50, 70)

/**** Inserção de dados na tabela Ocupa ****/
INSERT INTO Ocupa VALUES(301, 60)
INSERT INTO Ocupa VALUES(201, 64)
INSERT INTO Ocupa VALUES(101, 61)
INSERT INTO Ocupa VALUES(102, 62)

/**** Inserção de dados na tabela Dispensa ****/
INSERT INTO Dispensa VALUES('20152610002', '20101610100', 60)
INSERT INTO Dispensa VALUES('20152610002', '20101610100', 61)
INSERT INTO Dispensa VALUES('20152610002', '20101610200', 63)
INSERT INTO Dispensa VALUES('20152610005', '20101610400', 60)
INSERT INTO Dispensa VALUES('20152610005', '20101610400', 61)

/**** Inserção de dados na tabela Ministra ****/
INSERT INTO Ministra VALUES('20101610100', 1, '12/03/2018')
INSERT INTO Ministra VALUES('20101610100', 2, '14/03/2018')
INSERT INTO Ministra VALUES('20101610100', 3, '19/03/2018')
INSERT INTO Ministra VALUES('20101610100', 4, '21/03/2018')
INSERT INTO Ministra VALUES('20101610200', 6, '14/03/2018')

/**** Inserção de dados na tabela Desenvolvem ****/
INSERT INTO Desenvolvem VALUES('20152610005', '20101610100', 'GPDS_111127')
INSERT INTO Desenvolvem VALUES('20152610001', '20101610100', 'GPDS_111128')
INSERT INTO Desenvolvem VALUES('20152610002', '20101610300', 'GPDS_111127')
INSERT INTO Desenvolvem VALUES('20152610003', '20101610200', 'LINSCA_5789')
INSERT INTO Desenvolvem VALUES('20152610004', '20101610500', 'GREL_578690')

/**** Inserção de dados na tabela Contem ****/

```

```

INSERT INTO Contem VALUES (1,60)
INSERT INTO Contem VALUES (2, 63)
INSERT INTO Contem VALUES (1, 64)
INSERT INTO Contem VALUES (2, 62)
INSERT INTO Contem VALUES (5, 61)

```

```

/**/ Inserção de dados na tabela PreRequisito /**/
INSERT INTO PreRequisito VALUES (60, 62)
INSERT INTO PreRequisito VALUES (61, 62)
INSERT INTO PreRequisito VALUES (60, 63)
INSERT INTO PreRequisito VALUES (63, 64)
INSERT INTO PreRequisito VALUES (60, 65)

```

## 25 Consultas no Banco

Exibir todas as colunas da tabela Aluno ordenadas de forma ascendente pelo nome do Aluno para alunos que tem C como primeira letra do nome OU que não moram em avenidas.

```

SELECT * FROM Aluno
WHERE nomealuno LIKE 'C%' OR ruaendereçaaluno NOT LIKE 'Avenida%'
ORDER BY nomealuno

```

Selecionar o nome dos alunos e os nomes das disciplinas mesmo que o aluno não curse qualquer disciplina e que uma disciplina não tenha qualquer aluno. Ordenar os resultados pelo nome do aluno e pelo nome da disciplina.

```

SELECT nomealuno, nomedisciplina
FROM Aluno as al FULL JOIN Cursa as cu
ON al.matriculaaluno = cu.matriculaaluno
FULL JOIN Disciplina as di
ON cu.iddisciplina = di.iddisciplina

ORDER BY nomealuno, nomedisciplina

```

Exibir o nome dos professores e os nomes de seus dependentes.

```

SELECT nomeprofessor, nomedependente
FROM Professor as prof INNER JOIN Dependente as dep
ON prof.matriculaprofessor = dep.matriculaprofessor

```

Exibir o nome dos professores e sua quantidade de dependentes para professores com mais de um dependente.

```

SELECT nomeprofessor, COUNT(*) as Qtde_Dep
FROM Professor as prof INNER JOIN Dependente as dep
ON prof.matriculaprofessor = dep.matriculaprofessor
GROUP BY nomeprofessor
HAVING COUNT(*) > 1

```

Mostrar a menor, a maior e a média para a nota 1 na disciplina de Cálculo 1.

```
SELECT MIN(nota1) as 'Menor Nota 1', MAX(nota1) as 'Maior nota 1', AVG(nota1) as Media
FROM CURSA
WHERE iddisciplina = (SELECT(iddisciplina)
                        FROM Disciplina
                        WHERE nomedisciplina = 'Calculo 1')
```

Mostrar os nomes dos cursos cujos coordenadores assumiram em 2011 e 2012 e a data de início da coordenação deles.

```
SELECT nomecurso, iniciocoordenacao
FROM Curso
WHERE iniciocoordenacao BETWEEN '2011/01/01' AND '2012/12/31'
```

Mostrar a soma do número de carteiras de todas as salas de aula, sem considerar na soma as carteiras das salas de código entre 101 e 103.

```
SELECT SUM(qtdecarteiras) AS "Total_de_Carteiras"
FROM SalaDeAula
WHERE codLocal NOT BETWEEN 101 and 103
```

Mostrar num aula das aulas nas quais nenhuma informação acerca da duração e do conteúdo foi dada.

```
SELECT *
FROM AULA
WHERE duracao IS NULL AND conteudo IS NULL
```

Exibir o nome do aluno, nome do professor e o codprojeto. O nome de todos os alunos e de todos os professores devem ser exibidos, mesmo que não desenvolvam projeto.

```
SELECT nomealuno, nomeprofessor, codprojeto
FROM Aluno AS al LEFT JOIN desenvolvem as de
ON al.matriculaaluno = de.matriculaaluno
RIGHT JOIN professor as pr
ON de.matriculaprofessor = pr.matriculaprofessor
```

Exibir os dependentes de todos os professores que não tem endereço nas cidades Cabedelo e Conde.

```
SELECT * FROM Dependente
WHERE matriculaprofessor IN(SELECT matriculaprofessor
                             FROM Professor
                             WHERE cidadeendereçoprofessor NOT IN ('
                             Cabedelo', 'Conde'))
```

Exibir as informações de todos os recursos cuja a quantidade utilizada está disponível, ou seja, não é nula.

```
SELECT * FROM Recurso
WHERE qtdutilizada IS NOT NULL
```

## 26 Atualização de Dados com Condições

Alterar a nota3 da aluna Laura Avelar em Álgebra Vetorial para 60.

```
UPDATE Cursa
SET nota3 = 60
WHERE iddisciplina =
(SELECT iddisciplina
FROM Disciplina
WHERE nomedisciplina = 'Álgebra Vetorial') and matriculaaluno =
(SELECT matriculaaluno
FROM Aluno
WHERE nomealuno = 'Laura Avelar')
```

Atualizar a descrição para projeto encerrado e definir bolsa como 0 para projetos que iniciaram antes de 2018.

```
UPDATE ProjPesquisa
SET descricao = 'Projeto Encerrado'
WHERE YEAR(datainicio) < 2018

UPDATE ProjPesquisa
SET bolsa = 0
WHERE YEAR(datainicio) < '2018'
```

## 27 Exclusão de Dados com Condições

Remover todos os projetos de pesquisa que se iniciaram antes de 2018.

```
SELECT * FROM ProjPesquisa

DELETE FROM Desenvolvement
WHERE codprojeto IN (SELECT codprojeto
                     FROM ProjPesquisa
                     WHERE YEAR(datainicio) < '2018')

DELETE FROM ProjPesquisa
WHERE YEAR(datainicio) < '2018'

SELECT * FROM ProjPesquisa
```

Excluir todas as salas de aula com menos de 30 carteiras.

```
SELECT * FROM Local
SELECT * FROM Ocupa
SELECT * FROM SalaDeAula

DELETE FROM Ocupa
WHERE codlocal IN (SELECT codlocal
                  FROM SalaDeAula
                  WHERE qtdecarteiras < 30)
```

```
CREATE TABLE temp(  
codlocal_para_excluir INT NOT NULL)  
  
INSERT INTO temp(codlocal_para_excluir)  
SELECT codlocal  
FROM SalaDeAula  
WHERE qtdecarteiras < 30  
  
DELETE FROM SalaDeAula  
WHERE qtdecarteiras < 30  
  
DELETE FROM Local  
WHERE codlocal IN (SELECT *  
FROM temp)  
  
DROP TABLE temp  
  
SELECT * FROM Local  
SELECT * FROM Ocupa  
SELECT * FROM SalaDeAula
```