

Trabalho avaliativo de Banco de Dados da terceira unidade

Alunos(as):

Thereza Angélica Moura e Silva

Lucas Oliveira Machado

QUESTÃO 1)

A aplicação gerada tem com intuito manter o controle do trafego de dados para análise e conclusão da situação de alunos, professores e matérias dispostas durante os semestre em um faculdade. Os dados em questão que serão utilizados são:

PK - primaryKey

Fk - Foreign Key

Professor(idProfessor(PK), idBairro(FK), siape, nomeProfessor)

Aluno(idAluno(PK), idBairro(FK), Matricula, nomeAluno)

Disciplina(idDisciplina(PK), idProfessor(FK), codigo, nomeDisciplina, cargaHoraria)

Turma(idTurma(PK), ano, localTurma, numeroTurma, horario, turno, semestre)

AlunoTurmaDisciplina(idAlunoTurmaDisciplina(PK), idTurma(FK), idAluno(FK), idDisciplina(FK))

Historico(idHistorico(PK), idDisciplina(FK), idAluno(FK), frequencia, nota, situacao)

Bairro(idBairro(PK), nomeBairro, cidade)

Uma das restrições será que na tabela Historico o campo situacao só aceitará os valores 'APR', 'APRN', 'REP', 'CANC', 'MATR' e 'REPMF' e a outra restrição é para as cargaHorarias das disciplinas só sendo aceitas 30, 60, 90 e 120 horas de disciplina.

Consultas me algebra relacional:

1.1)selecionando a situação dos alunos com nome e matriula:

$(\pi \text{ nomeAluno, matricula, situacao } (\sigma \text{ Aluno.idAluno} = \text{Historico.idAluno} \text{ (Aluno } \bowtie \text{ Historico))})$

1.2)selecionando o nome dos professores das disciplinas presentes e suas carga horárias:

$(\pi \text{ nomeProfessor, nomeDisciplina, cargaHoraria } (\sigma \text{ Professor.idProfessor} = \text{Disciplina.idProfessor} \text{ (Professor } \bowtie \text{ Disciplina))})$

1.3) Listando todos os bairros na qual há algum membro da faculdade e é da cidade de Parnamirim

(π nomeBairro, cidade (σ cidade = 'parnamirim')(Bairro))

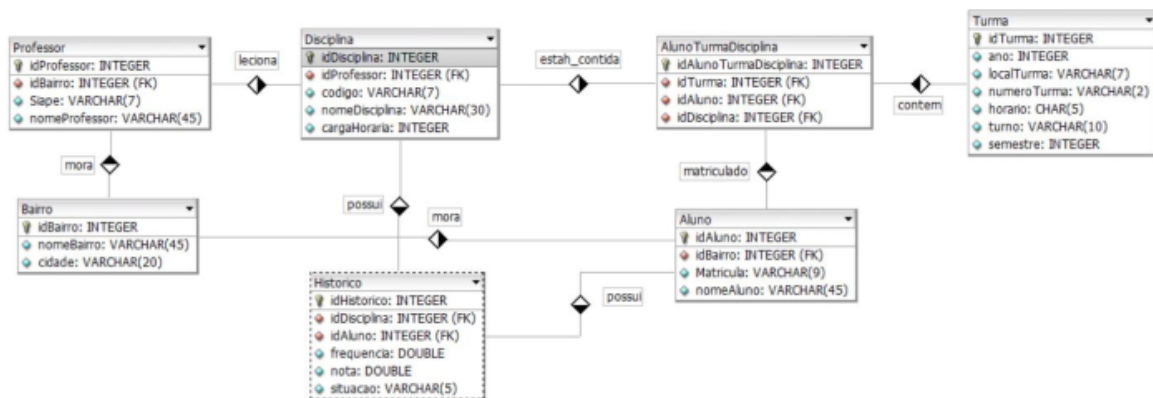
1.4) Selecionando todos os nomes de alunos e suas matriculas

(π nomeAluno, matricula (Aluno))

1.5) Listando todas as turmas presentes com local, numero da turma e horário

(π localTurma, numeroTurma, horario (Turma))

QUESTÃO 2)



QUESTÃO 3)

```

CREATE TABLE Bairro (
    idBairro INTEGER NOT NULL,
    nomeBairro VARCHAR(45),
    cidade VARCHAR(20),

```

```

    PRIMARY KEY(idBairro)

```

```

);

```

```

CREATE TABLE Professor(
    idProfessor INTEGER NOT NULL,
    idBairro INTEGER NOT NULL references Bairro,
    siape VARCHAR(7),
    nomeProfessor VARCHAR(45),

    PRIMARY KEY(idProfessor),
    FOREIGN KEY (idBairro) REFERENCES Bairro(idBairro)

```

```

);

```

```

CREATE TABLE Aluno(
    idAluno INTEGER NOT NULL,
    idBairro INTEGER NOT NULL references Bairro,

```

```

        Matricula VARCHAR(9),
        nomeAluno VARCHAR(45),

        PRIMARY KEY (idAluno),
        FOREIGN KEY (idBairro) REFERENCES Bairro(idBairro)
    );

CREATE TABLE Turma(
    idTurma INTEGER NOT NULL,
    ano INTEGER NOT NULL,
    localTurma VARCHAR(7),
    numeroTurma VARCHAR(2),
    horario CHAR(5),
    turno VARCHAR(10),
    semestre INTEGER NOT NULL,

    PRIMARY KEY (idTurma)
);

CREATE TABLE Disciplina(
    idDisciplina INTEGER NOT NULL,
    idProfessor INTEGER NOT NULL references Professor,
    codigo VARCHAR(7),
    nomeDisciplina VARCHAR(30),
    cargaHoraria INTEGER NOT NULL check(
        cargaHoraria = 30 or
        cargaHoraria = 60 or
        cargaHoraria = 90 or
        cargaHoraria = 120
    ),

    PRIMARY KEY (idDisciplina),
    FOREIGN KEY (idProfessor) REFERENCES Professor(idProfessor)
);

CREATE TABLE AlunoTurmaDisciplina(
    idAlunoTurmaDisciplina INTEGER NOT NULL,
    idTurma INTEGER NOT NULL references Turma,
    idAluno INTEGER NOT NULL references Aluno,
    idDisciplina INTEGER NOT NULL references Disciplina,

    PRIMARY KEY (idAlunoTurmaDisciplina),
    FOREIGN KEY (idTurma) REFERENCES Turma(idTurma),
    FOREIGN KEY (idAluno) REFERENCES Aluno(idAluno),
    FOREIGN KEY (idDisciplina) REFERENCES Disciplina(idDisciplina)
);

CREATE TABLE Historico(

```

```

idHistorico INTEGER NOT NULL ,
idDisciplina INTEGER NOT NULL references Disciplina,
idAluno INTEGER NOT NULL references Aluno,
frequencia DOUBLE PRECISION ,
nota DOUBLE PRECISION,
situacao VARCHAR(5) check(
    situacao = 'APR' or
    situacao = 'APRN' or
    situacao = 'REP' or
    situacao = 'CANC' or
    situacao = 'MATR' or
    situacao = 'REPMF'
),

PRIMARY KEY(idHistorico),
FOREIGN KEY (idAluno) REFERENCES Aluno(idAluno),
FOREIGN KEY (idDisciplina) REFERENCES Disciplina(idDisciplina)
);

```

|-----INSERTS-----|

```

INSERT INTO Bairro(idBairro, nomeBairro, cidade) values
(1, 'Nova Descoberta', 'Natal'),
(2, 'Areia Petra', 'Natal'),
(3, 'Centro', 'Parnamirim');

```

```

Insert into Professor (idProfessor, idBairro, siape, nomeProfessor) values
(1,1, 'LPsc1', 'Luiz Martins Silva'),
(2,2, 'Mtsc2', 'Adriana Lohane Pereira');

```

```

Insert into Aluno(idAluno, idBairro, Matricula, nomeAluno) values
(1,1, 'MA01A', 'Luana Marjory'),
(2,2, 'MA02B', 'Breno Lima'),
(3,1, 'MA03C', 'Thereza Angélica'),
(4,3, 'MA04D', 'Danilo Azambuja'),
(5,3, 'MA05E', 'Maria Milena');

```

```

INSERT INTO Turma(idTurma, ano, localTurma, numeroTurma, horario, turno, semestre)
values
(1, 2020, 'sala1', 01, '12M34', 'vespertino', 1),
(2, 2020, 'sala2', 03, '12N34', 'noturno', 1);

```

```

INSERT INTO Disciplina(idDisciplina, idProfessor, codigo, nomeDisciplina, cargaHoraria)
values
(1, 1, 'DESBA01', 'desenho basico 1', 90),
(2, 2, 'JGDIG01', 'jogos digitais', 60);

```

```
INSERT INTO AlunoTurmaDisciplina(idAlunoTurmaDisciplina, idTurma, idAluno,  
idDisciplina) values
```

```
(1, 2, 3, 2),
```

```
(2, 1, 4, 1),
```

```
(3, 1, 5, 1),
```

```
(4, 2, 1, 2),
```

```
(5, 2, 2, 2);
```

```
INSERT INTO Historico(idHistorico, idDisciplina, idAluno, frequencia, nota, situacao) values
```

```
(1, 1, 4, 80.0, 7.8, 'APR'),
```

```
(2, 1, 5, 88.7, 7.9, 'APR'),
```

```
(3, 2, 1, 75.0, 8.0, 'APR'),
```

```
(4, 2, 2, 94.6, 7.5, 'APR'),
```

```
(5, 2, 3, 90.0, 8.6, 'APR');
```

QUESTÃO 4) (vai está em um pacote com o programa com 3 inserts e um delete e as 5 buscas).