



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Sistemas de Banco de Dados

Lista de exercícios 1

Felipe Menino Carlos

Este documento tem por objetivo apresentar a resolução das questões apresentadas na primeira lista de exercícios de Sistemas de Banco de Dados, da disciplina de Computação Aplicada 1.

Exercício 1

O objetivo principal deste exercício foi criar um diagrama entidade-relacionamento, seguindo os requisitos do banco de dados modelado para uma universidade. Neste contexto, as seguintes definições foram consideradas.

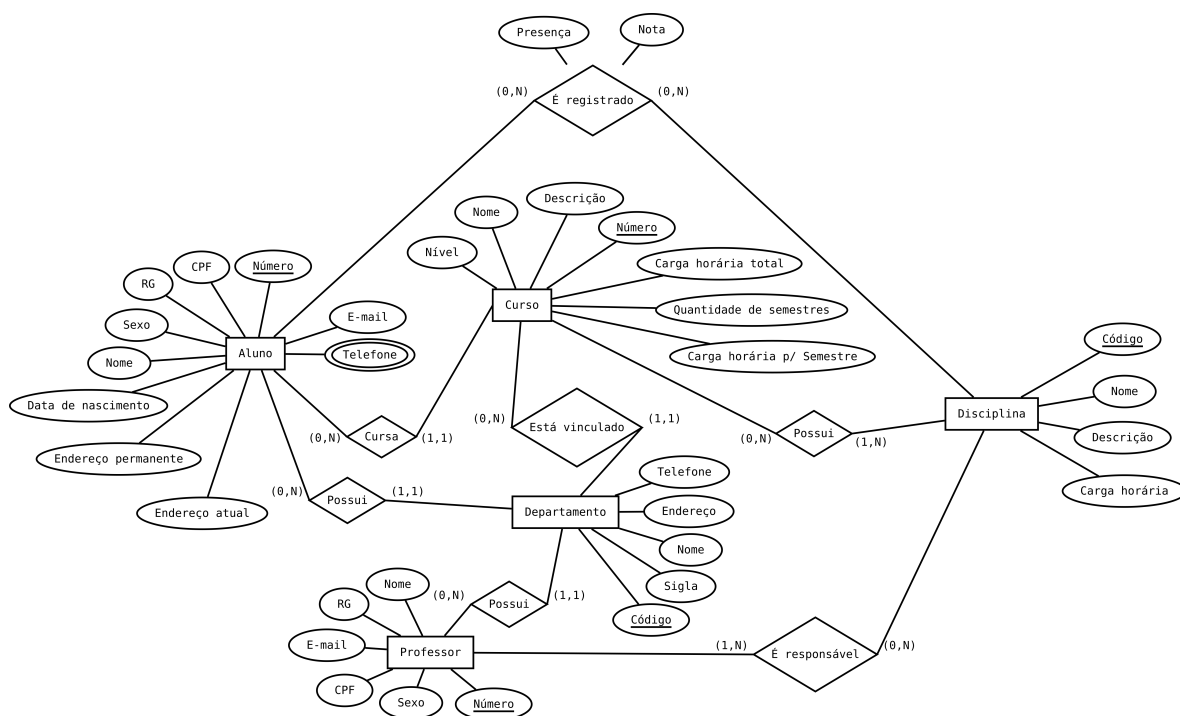
- A universidade mantém as informações de cada aluno: nome, número, CPF, RG, endereço atual, telefones, endereço permanente, e-mail, data de nascimento e sexo. CPF, RG e número devem ter valores únicos para cada estudante. Além disso, ele deve conter informações sobre qual curso o aluno está matriculado e qual departamento da universidade ele está vinculado;
- Cada departamento da universidade é descrito por seu nome, sigla, código, endereço e número de telefone. Nome, sigla e código devem ser únicos para cada departamento. Cada departamento tem um conjunto de professores associados a ele e cada professor da universidade pertence a apenas um departamento. Deve-se guardar as informações dos professores, como nome, CPF, endereço, etc;
- Cada curso oferecido pela universidade tem um nome, uma descrição, um número, a carga horária por semestre, a carga horária total, o número de semestres, o nível (se é graduação, mestrado ou doutorado) e o departamento que o oferece. O número do curso é único para cada curso;
- Cada curso tem uma grade curricular, ou seja, um conjunto de disciplinas que devem ser cursadas durante o curso. Cada disciplina tem um código, um nome,

uma descrição, uma carga horária e os professores responsáveis. O nome e código devem ser únicos para cada disciplina;

- Cada aluno se matricula em um conjunto de disciplinas por semestre. Para cada matrícula em cada disciplina, a universidade deve manter a nota final do aluno e o número de presenças e faltas.

Levando em consideração tais definições, o diagrama entidade-relacionado, apresentado na Figura 1 foi criado.

Figura 1: Modelo entidade-relacionamento

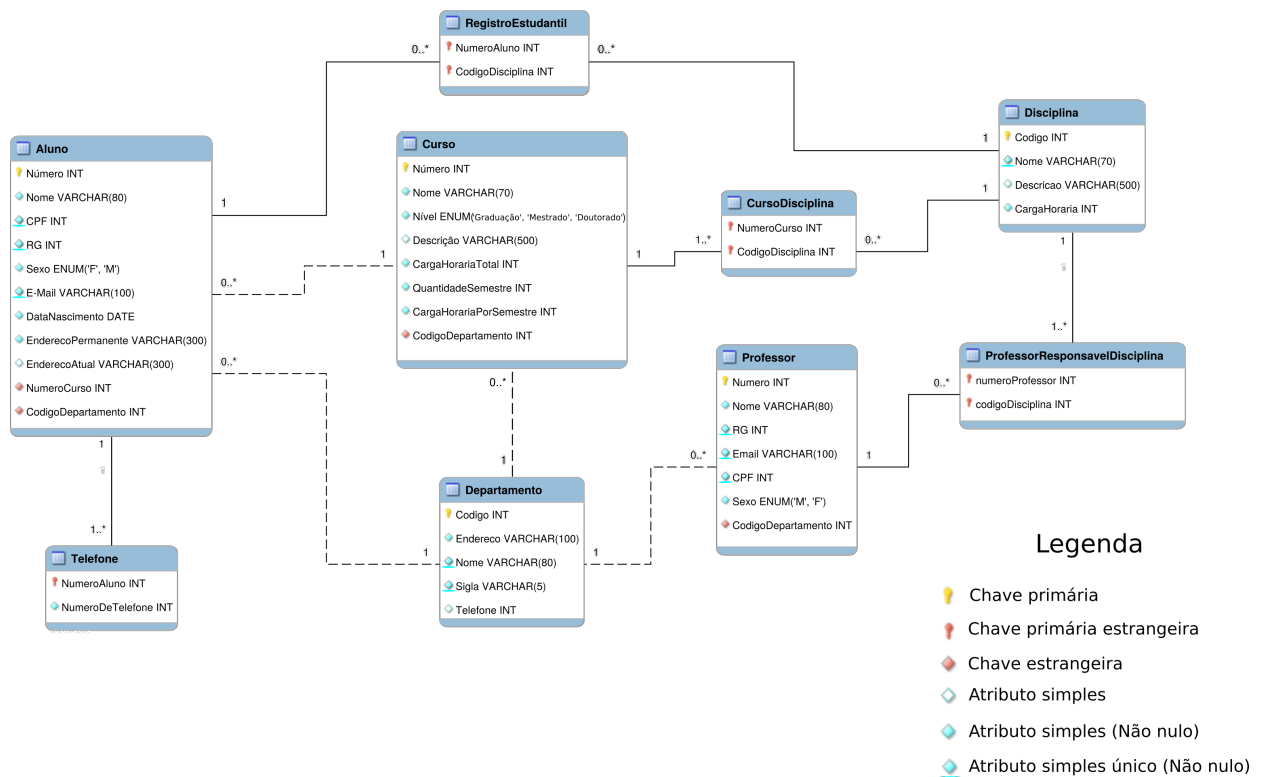


Para o diagrama apresentado na Figura 1, vale citar que, as definições de chave primária foram feitas considerando as características de identificação única dos atributos, apresentados nos requisitos. Além disto, para a relação entre curso e disciplina, é considerado que a disciplina é independente de qualquer curso, para que casos de disciplinas oferecidas de forma extra-curso, por um departamento ou professor, possam ser creditados dentro do histórico escolar de cada aluno.

Exercício 2

Com base no modelo entidade-relacionamento criado no exercício anterior, nesta segunda atividade, é pedido a criação do modelo relacional. Para a realização desta atividade, os 7 passos apresentados por 1 para o mapeamento entre o modelo entidade-relacionamento e o relacional foram aplicados. O modelo relacional gerado é apresentado na Figura 2, onde, para cada atributo é apresentado seu tipo e suas restrições.

Figura 2: Modelo relacional

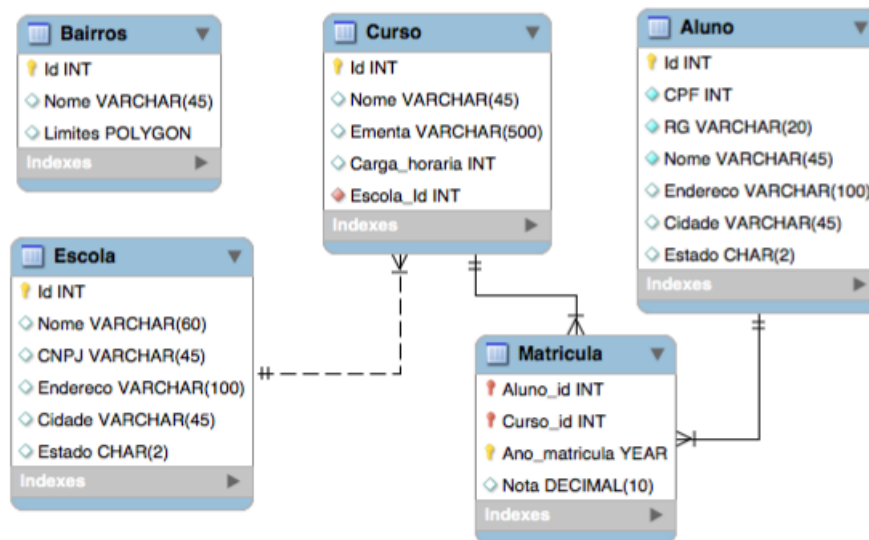


Exercício 3

O diagrama relacional da Figura 3 é formado por 5 tabelas:

1. Bairros: contém os nomes e os limites dos bairros de um município (PK: Id);
2. Escola: contém algumas escolas desse município (PK: Id);
3. Curso: contém os cursos oferecidos pelas escolas (PK: Id);
4. Aluno: contém os registros dos alunos (PK: Id);
5. Matricula: associa alunos a cursos, ou seja, quais cursos cada aluno cursou, em qual ano e qual foi sua nota (PK: Aluno_id, Curso_id e Ano_matricula).

Figura 3: Modelo relacional



Com base nestas informações, as questões abaixo foram respondidas

Questão 1 - Em um modelo relacional, o que é restrição de integridade referencial ?

Como apresentado por Elmasri e Navathe (2008) [1], a integridade referencial está relacionada as relações criadas dentro do modelo relacional, onde, a referência da relação em uma tupla, deve se referir a uma tupla existente do outro lado da relação. Tal restrição ajuda na garantia de que a relação será consistente, não havendo referências inválidas (Elmasri e Navathe 2008 [1]).

Questão 2 - Indique quais são as chaves estrangeiras (*Foreign Keys*) do diagrama da Figura 1 e quais colunas e tabelas elas associam

A relação é apresentada na Tabela 1.

Tabela 1: Relação de chave estrangeira e tabelas associadas

Tabela de origem	Chave estrangeira	Tabela associada
Matricula	Aluno_id	Aluno(id)
Matrícula	Curso_id	Curso(id)
Curso	Escola_id	Escola(id)

Questão 3 - A tabela *Matricula* pode conter matrículas de um mesmo aluno em um mesmo curso mais de uma vez? Por que?

Na solução deste exercício, duas respostas foram identificadas. A primeira considera que os valores da chave ‘Ano_matricula’ sempre serão diferentes (Uma matricula por ano, por exemplo), então, neste caso a resposta é sim, uma vez que como a chave primária da tabela é composta, quando um dos valores muda a chave é diferente das demais. Porém, ao considerar que o valor de ‘Ano_matricula’ pode se repetir, então, a resposta passa a ser não, e a lógica para isto é a mesma apresentada anteriormente.

Questão 4 - Se todas as chaves estrangeiras do diagrama forem criadas com a ação *ON DELETE CASCADE* e *ON UPDATE CASCADE*, o que acontece se

- (a) Eu remover o curso XXX da tabela Curso ?
 - R: Os elementos de todas as tabelas que fazem referência para o curso XXX serão removidos
- (b) Eu alterar o nome do curso XXX para YYY na tabela Curso ?
 - R: Considerando que as demais tabelas fazem referência somente ao id da tabela curso, não ocorre mais mudanças que não a do valor XXX para YYY, mantendo todas as demais relações iguais.
- (c) Eu remover o aluno ZZZZ da tabela Aluno ?
 - R: Ao remover o aluno ZZZZ, os registro presentes da tabela matrícula que fazem referência a este aluno serão excluídos.
- (d) Eu remover a escola EEEE da tabela Escola ?
 - R: Ao remover a escola EEEE, os registros que fazem referência a este da tabela Curso serão removidos, o que causa também a remoção dos elementos da tabela matrícula relacionados aos excluídos da tabela curso.
- (e) Eu alterar o ano de uma matricula (Ano_matricula) da tabela Matricula ?

-
- R: Como o atributo `Ano_matricula` é uma chave primária que nenhuma outra tabela faz referência, a modificação apenas altera o valor do registro. A troca só poderá ser feita por um valor que ainda não está presente na tabela;

(f) Eu alterar o id do curso de um aluno (`Curso_id`) da tabela `Matricula` ?

- R: A troca será feita, desde que, haja na tabela `curso` um registro com o id indicado.

Questão 5 - Cada turma, que é composta pelos alunos que se matricularam em um mesmo curso em um mesmo ano, elege um aluno representante para participar de reuniões com a diretoria das escolas. Como você incluiria essa informação no diagrama da Figura 1 ?

Uma tabela `Representante` poderia ser criada, este sendo considerada uma entidade fraca, passa a existir somente a partir do momento que o vínculo com uma matrícula é feito. Com isso, para cada turma, em cada ano, poderá existir um representante diferente.

Questão 6 - Quais recursos de um SGBD você usaria para implementar as restrições abaixo (descreva qual o recuso, como ele funciona e como seria implementado – sobre quais tabelas e colunas):

(a) As notas dos alunos nos cursos devem ser entre zero e dez .

A solução deste exercício é feita considerando que as mudanças poderia ser feitas na definição da tabela, porém, o mesmo processo poderia ser aplicado em tabelas já existentes, através dos comandos `ALTER TABLE`, visto em aula.

Assim, para solucionar este exercício, é utilizado a *constraint* ‘`CHECK`’, que verifica condições arbitrárias, definidas no momento de sua definição, indicando que um registro pode ou não ser salvo no banco de dados. Neste caso, tal *constraint* pode ser adicionada diretamente na coluna `nota` da tabela `Matricula`, verificando a necessidade da nota estar entre o intervalo $[0, 10]$. A mudança feita na tabela é apresentada abaixo.

Figura 4: Adição da *constraint* `CHECK`

```
1 CREATE TABLE Matricula
2 (
3     -- Comandos omitidos
4     nota DECIMAL(10) CHECK(nota >= 0 AND nota <= 10)
5     -- Comandos omitidos
6 )
```

(b) Não podem existir 2 ou mais alunos com o mesmo RG

Da mesma forma que o exercício anterior, aqui é considerado uma alteração na criação da tabela, porém tal modificação poderia ser aplicada em tabelas já existentes. A *constraint* aplicada para a solução deste problema é a `UNIQUE`, que garante a

unicidade dos valores das colunas marcadas. Para este caso, seu uso seria feito na coluna RG da tabela Aluno, como apresentado abaixo.

Figura 5: *Adição da constraint UNIQUE*

```
1 CREATE TABLE Aluno
2 (
3     -- Comandos omitidos
4     RG VARCHAR(20) UNIQUE
5     -- Comandos omitidos
6 )
```

- (c) Um aluno só pode estar matriculado em no máximo 3 cursos distintos em um mesmo ano

Para a solução deste exercício, a funcionalidade de **trigger** pode ser aplicada. Para isso, a **trigger** é criada e vinculada a tabela de matrícula, sendo configurada para funcionar antes de qualquer inserção. No corpo da função a ser executada pela **trigger**, é feita a verificação da quantidade de cursos que o id identificado na nova inserção está associado, emitindo um erro caso este valor seja acima de três.

Referências

- [1] ELMASRI, R., AND NAVATHE, S. B. *Sistemas de banco de dados*. Pearson Addison Wesley, São Paulo, 2008.