



Departamento de Ciência da Computação - UFJF

DCC019 - Linguagens de Programação

Relatório do 1º Trabalho Prático

Oromar Voit de Rezende - 201765122C

Dezembro / 2021

Sumário

1	Introdução	1
2	Modelagem da Base de Dados	1
3	Operações no Sistema	2
4	Pesquisas Sobre a Base	4
4.1	histórico escolar de um estudante	4
4.2	matriz curricular de um curso	4
4.3	relação de estudantes que já cursaram uma dada disciplina, podendo ou não incluir um critério de seleção por nota	4
4.4	relação de disciplinas que faltam ser cursadas para um dado estudante	5
4.5	relação de estudantes de um dado curso, podendo incluir critério de seleção por nota em disciplina ou IRA	5
4.6	relação de cursos que contém uma dada disciplina	5
5	Utilizando o Programa	5

1 Introdução

O projeto consiste no desenvolvimento de um simples Sistema de Gestão Acadêmico. Os atores envolvidos são estudantes, cursos e disciplinas. As relações entre eles devem ser tais que as consultas pedidas na especificação possam ser realizadas.

O trabalho foi desenvolvido na linguagem de programação *Prolog*, utilizando o ambiente *SWI-Prolog* e a biblioteca *persistence*, já presente neste ambiente e utilizada para fazer a persistência da base em disco.

2 Modelagem da Base de Dados

Como modelo da base de dados, foi desenvolvido um Modelo Entidade Relacionamento, como pode ser visto na figura .

Como pode ser visto no modelo, Disciplina e Curso são entidades fortes, ou seja, existem por si só no sistema. Estudante é uma entidade fraca, pois necessita estar associado a um Curso para existir. Se um Curso for deletado, todos os Estudantes relacionados a ele também deixam de existir.

Os atributos sublinados são encarados como chave primária. Isso significa que são únicos no sistema e, para facilitar a implementação e impedir inconsistências na base, são imutáveis.

Cada uma dessas entidades são representados como fatos em *Prolog*.

Os relacionamentos *Cursou* e *Grade* também são implementados como fatos na base de dados. São identificados pelas chaves primárias das entidades que relacionam. Apagar da base uma das entidades das quais estão ligadas, também as apaga da base.

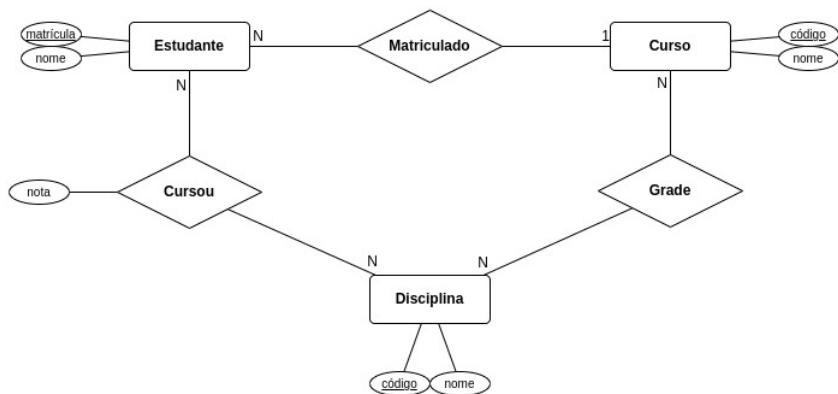


Figura 1. Modelo Entidade Relacionamento

3 Operações no Sistema

As tabelas abaixo mostram operações que podem ser feitas diretamente no gerenciamento da base de dados, adicionando, removendo, consultando e editando diretamente as entidades e relacionamentos.

Operação	Descrição	Exemplo
add_curso(Codigo, Nome)	Adiciona o Curso com código e Nome	add_curso('65361', 'Ciência da Computação')
current_curso(Codigo, Nome)	Pesquisa o Curso com o código informado	current_curso('65361', Nome)
set_curso(Codigo, Nome)	Edita o nome do curso que tem o código informado	set_curso('65361', 'Engenharia Computacional')
del_curso(Codigo)	Deleta o Curso que contém o código informado	del_curso('65361')

Tabela 1. Operações Sobre Curso

Operação	Descrição	Exemplo
add_disciplina(Codigo, Nome)	Adiciona a Disciplina com código e Nome	add_disciplina('MAT154', 'Cálculo I')
current_disciplina(Codigo, Nome)	Pesquisa a Disciplina com o código informado	current_disciplina('MAT154', Nome)
set_disciplina(Codigo, Nome)	Edita o nome da Disciplina que tem o código informado	set_disciplina('MAT154', 'Cálculo Um')
del_disciplina(Codigo)	Deleta a Disciplina que contém o código informado	del_disciplina('MAT154')

Tabela 2. Operações Sobre Disciplina

Operação	Descrição	Exemplo
add_estudante(Matricula, Nome, Cod_Curso)	Adiciona o Estudante com Matrícula, Nome e Código do Curso	add_estudante('001', 'Ian Mcdonnell', '65361')
current_estudante(Matricula, Nome, Cod_Curso)	Pesquisa o Estudante com a matrícula informada	current_estudante('001', Nome, Cod_Curso)
set_estudante(Matricula, Nome, Cod_Curso)	Edita nome e curso do Estudante com a matrícula informada	set_estudante('001', 'Yan', '65361')
del_estudante(Matricula)	Deleta o Estudante com a Matrícula informada	del_estudante('001')

Tabela 3. Operações Sobre Estudantes

Operação	Descrição	Exemplo
add_cursou(Matricula, Codigo, Nota)	Adiciona Disciplina informada pelo código como cursada pelo aluno informado pela matrícula	add_cursou('001', 'MAT155', 14)
current_cursou(Matricula, Codigo, Nota)	Pesquisa se aluno cursou a disciplina e retorna a nota	current_cursou('001', 'MAT155', Nota)
set_cursou(Matricula, Codigo, Nota)	Edita nota do aluno na disciplina	set_cursou('001', 'MAT155', 90)
del_cursou(Matricula, Codigo)	Deleta a informação de que o aluno cursou a disciplina	del_cursou('001', 'MAT155')

Tabela 4. Operações Sobre Cursou

Operação	Descrição	Exemplo
add_grade(Cod_Curso, Cod_Disciplina)	Adiciona Disciplina na Grade do Curso	add_grade('65361', 'DCC075').
current_grade(Cod_Curso, Cod_Disciplina)	Pesquisa se disciplina está na grade do Curso	current_grade('65361', 'DCC075')
del_grade(Cod_Curso, Cod_Disciplina)	Remove disciplina da grade do curso	del_grade('65361', 'DCC075')

Tabela 5. Operações Sobre Grade

4 Pesquisas Sobre a Base

Os comando abaixo, descrevem diversas consultas sobre a base de dados, conforme pedido na especificação do trabalho.

4.1 histórico escolar de um estudante

Comando: historico(Matricula, Disciplinas, Ira)

Exemplo: historico('001', Disciplinas, Ira)

4.2 matriz curricular de um curso

Comando: matriz(Curso, Matriz)

Exemplo: matriz('65361', Matriz)

4.3 relação de estudantes que já cursaram uma dada disciplina, podendo ou não incluir um critério de seleção por nota

Sem critério de Nota:

Comando: rel_estudantes(Disciplina, Estudantes)

Exemplo: rel_estudantes('MAT154', Estudantes)

Com critério de Nota:

Comando: rel_estudantes_nota(Disciplina, Estudantes, [Min, Max])

Exemplo: rel_estudantes_nota('MAT154', Estudantes, [90,100])

4.4 relação de disciplinas que faltam ser cursadas para um dado estudante

Comando: rel_disc_faltantes(Matricula, Disciplinas)

Exemplo: rel_disc_faltantes('002', Disciplinas)

4.5 relação de estudantes de um dado curso, podendo incluir critério de seleção por nota em disciplina ou IRA

Sem critério:

Comando: rel_estudantes_curso(Curso, Matriculas)

Exemplo: rel_estudantes_curso('65361', Matriculas)

Com critério de seleção por nota em disciplina:

Comando: rel_estudantes_curso_disciplina_nota(Curso, Matriculas, Disciplina, [Min, Max])

Exemplo: rel_estudantes_curso_disciplina_nota('65361', Matriculas, 'MAT154', [81, 100])

Com critério de seleção por IRA:

Comando: rel_estudantes_curso_ira(Curso, Matriculas, [Min, Max])

Exemplo: rel_estudantes_curso_ira('65361', Matriculas, [60, 100])

4.6 relação de cursos que contém uma dada disciplina

Comando: rel_cursos_disciplina(Disciplina, Cursos)

Exemplo: rel_cursos_disciplina('MAT154', Cursos)

5 Utilizando o Programa

O programa está dividido em módulos, cada um dos quais responsável por gerenciar uma parte da base de dados. Quando for utilizar o programa, basta carregar o arquivo *main.pl*.

Durante a utilização serão criados arquivos com a extensão *.journal*. Nesses arquivos estão as informações que persistem no sistema, e não devem ser manipulados diretamente.

Também existe um arquivo *init.txt*. Ele pode ser usado para popular a base de dados. Para tal, basta utilizar este arquivo como entrada para o programa, utilizando redirecionamento de entrada. No bash do Linux, uma opção seria: *swipl main.pl < init.txt*.