

Faculdade de Engenharia da Universidade do Porto

FEUP BDAD

Bases de Dados
2019/2020



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

2MIEIC06_04:

Fábio Huang	up201806829@fe.up.pt
João Dossena	up201800174@fe.up.pt
Válter Castro	up201706546@fe.up.pt

Index

Introdução.....	1
Especificação do projeto.....	2
Diagrama UML.....	4
Esquema Relacional.....	5
Dependências Funcionais.....	6
Violações à Terceira Forma Normal.....	7
Violações à Formal Normal de Boyce-Codd.....	7
Restrições à Base de Dados.....	8
Interrogações (Queries).....	10
Gatilhos (Triggers).....	11
Intruções de Execução.....	12

Introdução

No nosso projeto da disciplina “Base de Dados” é desenvolvida uma base de dados relativa a alguns aspectos da FEUP. Nela, é conjugada informação envolta na superclasse PessoaFeup, nomeadamente estudantes, docentes, cursos, cadeiras, turmas práticas, épocas, ocorrências, componentes de avaliação e notas.

Especificação do projeto

O esquema é iniciado com uma superclasse “Pessoas da FEUP”, com atributos nome, id, morada, data de nascimento, sexo, e-mail, nif e telefone. Desta superclasse derivam duas subclasses:

“Estudante”, que contém o ano de inscrição e a classe “Docente”, que tem como atributos, categoria (por exemplo, Professor Associado, Professor Auxiliar, Assistente Convidado...) e departamento. Dado que um utilizador da FEUP pode não ser estudante ou docente, e dado que um estudante pode ser, a título de exemplo, um assistente convidado, a generalização é dada como {Incompleto, inclusivo}.

O estudante está associado a uma classe “Curso”, contendo esta, nome, grau académico, data de início, duração e média de entrada, sendo que o mesmo só pode frequentar um curso dentro da FEUP de cada vez.

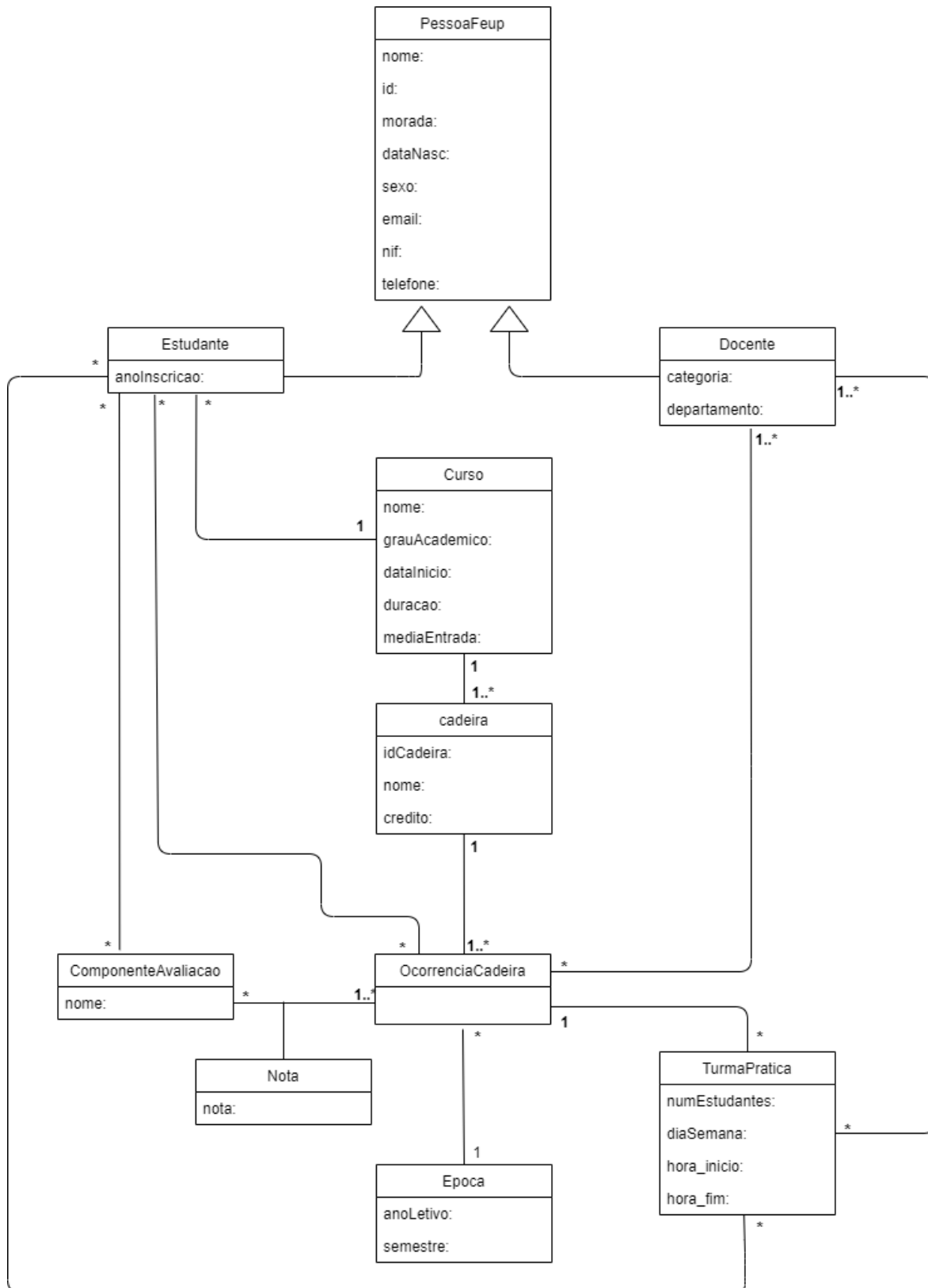
Um curso pode ter vários estudantes. Esta classe, por sua vez, está associada à classe “Cadeira”, que tem como atributos: código, nome e crédito. Ainda associado entre estes, sabemos que, um curso pode variar entre uma ou mais cadeiras, mas a mesma, só se relaciona a um curso.

A “Cadeira” tem como classes de associação, “OcorrênciaCadeira”, “Turma Prática” e “Docente”. Em relação a “OcorrênciaCadeira”, uma cadeira pode ter uma ou mais ocorrências, mas cada ocorrência só decorre uma vez. Em segundo, cada cadeira pode ter várias turmas práticas ou nenhuma, sendo que cada turma prática está associada a uma e uma só cadeira. Em terceiro cada cadeira tem um ou mais docentes, sendo que, cada docente pode lecionar várias cadeiras ou nenhuma.

A classe “OcorrênciaCadeira” está associada à classe “Época”, que preconiza como atributos, anoLetivo e semestre. Cada ocorrência pode ter ou não diversas épocas, verificando-se o mesmo no contrário, ou seja, cada época pode estar ou não relacionada a ocorrências. Também é de salientar a relação entre ocorrência e componente de avaliação, sendo que cada ocorrência pode ter ou não componentes de avaliação, mas cada componente de avaliação tem uma ou mais ocorrências.

A classe “ComponenteAvaliação” tem como atributo nome, e está associada à classe

“Nota” de forma que cada componente só tem uma nota e vice-versa. Uma componente pode conter ou não estudantes, e cada estudante pode estar relacionado a várias componentes ou nenhuma. A “Turma Prática” é uma classe cujos atributos se identificam como: numEstudantes, diaSemana, hora_inicio e hora_fim. Para além das relações já referidas, é também possível verificar que cada turma prática tem um ou mais docentes, sendo que estes podem ter ou não turmas práticas. Uma turma prática pode ter ou não estudantes e cada estudante pode pertencer a várias turmas práticas ou nenhuma.



Esquema Relacional

PessoaFeup(id, nome, nif, morada, dataNasc, sexo, email, telefone)

Estudante(id->PessoaFeup, anoInscricao, idCurso->Curso)

Docente(id->PessoaFeup, categoria, departamento)

Curso(idCurso, nome, grauAcademico, dataInicio, duracao, mediaEntrada)

Cadeira(idCadeira, nome, credito, idCurso->Curso)

OcorrenciaCadeira(idOcorrenciaCadeira, idCadeira->cadeira, idEpoca->Epoca)

TurmaPratica(idTurmaPratica, numEstudantes, diaSemana, horaInicio, horaFim, idOcorrenciaCadeira->OcorrenciaCadeira)

Epoca(idEpoca, anoLetivo, semestre)

ComponenteAvaliacao(idComponenteAvaliacao, nome)

EstudanteOcorrenciaCadeira(id->Estudante, idOcorrenciaCadeira->OcorrenciaCadeira)

EstudanteComponenteAvaliacao(id->Estudante, idComponenteAvaliacao->ComponenteAvaliacao)

EstudanteTurmaPratica(id->Estudante, idTurmaPratica->TurmaPratica)

DocenteTurmaPratica(id->Docente, idTurmaPratica->TurmaPratica)

DocenteOcorrenciaCadeira(id->Docente, idOcorrenciaCadeira->OcorrenciaCadeira)

Nota(idOcorrenciaCadeira->OcorrenciaCadeira, idComponenteAvaliacao->ComponenteAvaliacao, nota)

Dependências Funcionais

PessoaFEUP:

{id, nif, email, telefone} → {nome, morada, dataNasc, sexo}

Estudante:

{id} → {anoInscricao, idCurso}

Docente:

{id} → {categoria, departamento}

Curso:

{idCurso} → {nome, grauAcademico, dataInicio, duracao, mediaEntrada}

Cadeira:

{idCadeira} → {nome, credito, idCurso}

OcorrenciaCadeira:

{idOcorrenciaCadeira} → {idCadeira, idEpoca}

TurmaPratica:

{idTurmaPratica} → {numEstudantes, diaSemana, horaInicio, horaFim, idOcorrenciaCadeira}

Epoca:

{idEpoca} → {anoLetivo, semestre}

ComponenteAvaliacao:

{idComponenteAvaliacao} → {nome}

Nota:

{idComponenteAvaliacao, idOcorrenciaCadeira} → {nota}

Violações à Forma Normal de Boyce-Codd

Se uma relação não trivial $X \rightarrow Y$, X tem que ser um super chave, no nosso caso, todas as nossas dependências funcionais, têm na sua parte esquerda uma super chave então podemos garantir que esta na BCNF

Violações à 3ª Forma Normal

Se uma relação não trivial $x \rightarrow Y$, x for uma superchave ou Y for consistidos so por atributos primos, podemos afirmar esta forma normal, e como já tínhamos provado que as dependências funcionais estão na forma normal de boyce-codd então implica que esteja também na terceira forma normal, dando que BCNF é uma versão mais avançado da terceira forma normal.

Restricoes à base de dados

Estudante:

id deve ser única;

Docente:

id deve ser única;

Curso:

idCurso deve ser única;

mediaEntrada deve ser entre 0 e 20;

Docente:

id deve ser única;

Curso:

idCurso deve ser única;

mediaEntrada deve ser entre 0 e 20;

Cadeira:

idCadeira deve ser única;

OcorrenciaCadeira:

idOcorrenciaCadeira deve ser única;

Epoca:

idEpoca deve ser única;

semestre deve ser "Primeiro" ou "Segundo";

ComponenteAvaliacao:

idComponenteAvaliacao deve ser única;

Nota:

idNota deve ser única;

EstudanteOcorrenciaCadeira:

Não tem restrições

DocenteTurmaPratica:

Não tem restrições

CadeiraDocente:

Não tem restrições

OcorrenciaCadeiraEpoca:

Não tem restrições

TurmaPratica:

idTurmaPratica deve ser única;

diaSemana deve ser “Segunda-Feira”, “Terça-Feira”, “Quarta-Feira”, “Quinta-Feira”, “Sexta-Feira”;

PessoaFEUP:

id é único;

sexo deve ser ‘M’ ou ‘F’;

nif deve ser único;

telefone deve ser único;

emil deve ser único;

Interrogações (Queries)

1. Lista os nomes dos Estudantes da FEUP por ordem alfabética, como numa lista de chamada: verificação da associação entre as classes PessoaFeup e Estudante.
 2. Lista os tuplos nota, idCadeira, tal que a nota seja menor que 10. Ordena as notas por ordem crescente.
 3. Conta quantos Docentes têm um id relativo a anos anteriores a 2000 através da função COUNT.
 4. Lista os horários de início e fim das turmas práticas 01 do curso MIEIC, ordenados por duração da aula em ordem decrescente.
 5. Lista id, nome e número de cadeiras em que os Docentes lecionam.
 6. Mostra a média de número de estudantes nas Turmas Práticas com aula na Terça-Feira feita pelo cálculo da função matemática AVG.
 7. Mostra os Cursos com , respetivamente, maior e menor médias de entrada , ou seja, ordenados por ordem decrescente.
 8. Mostra os Estudantes e a soma do número de créditos das Cadeiras em que estão inscritos através da diretiva SUM que adiciona sucessivamente os créditos de cada cadeira.
 9. Lista os Estudantes que se inscreveram na FEUP em 2018 e que podem participar na Tuna Feminina da FEUP(Todas as mulheres).
 10. Lista quantos Docentes com categoria de Professor Auxiliar existem por departamento.
-

Gatilhos (Triggers)

Por fim, foram definidos 3 gatilhos que são úteis para a monitorização e manutenção da base de dados, dois dos quais estão no mesmo ficheiro.

1. Quando um Estudante é adicionado na classe EstudanteTurmaPratica, o atributo numEstudantes é incrementado por 1.
2. Verificar que o email está certo, caso contrário, envia uma mensagem “Formato de email errado!”.
3. Quando é eliminado um Estudante na classe EstudanteTurmaPratica, o número de estudantes é decrementado por 1.

Instruções de Execução

Estes passos deverão ser seguidos escrupulosamente, para evitar comportamento inesperado por parte da base de dados.

- 1.Executar o ficheiro create.sql
- 2.Executar o ficheiro povoar.sql
- 3.Correr as interrogações e gatilhos como pretendido, que se encontram nas pastas Querie e Triggers, respetivamente, no diretório principal da entrega.

Esta hierarquia de passos advém do facto das interrogações e gatilhos dependerem da correta população das tabelas declaradas no ficheiro create.sql.