

Faculdade de Engenharia da Universidade do Porto



Projeto de BD 21/22 – 2ª entrega

Definição do Esquema Relacional
Análise de Dependências Funcionais e Formas Normais
Criação da Base de Dados em SQLite
Adição de Restrições à Base de Dados
Carregamento de Dados

L.EIC | Base de Dados 2021/2022

Carla Teixeira Lopes & Michel Ferreira

Turma 2LEIC06 (grupo 601):

António Ferreira – up202004735@edu.fe.up.pt

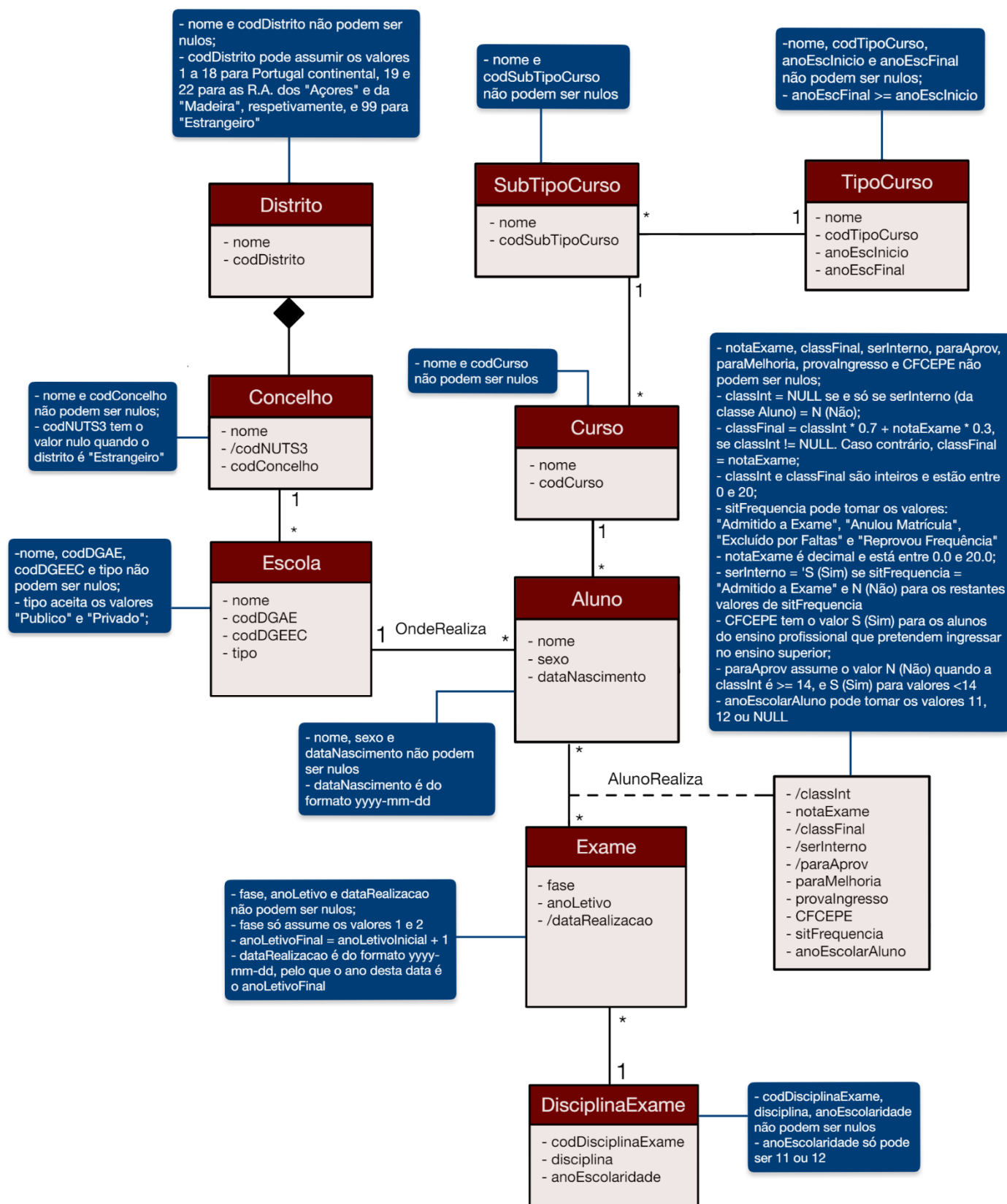
João Maldonado – up202004244@edu.fe.up.pt

Tomás Gomes – up202004393@edu.fe.up.pt

Índice

I.	Contexto.....	4
II.	Diagrama UML.....	
III.	Diagrama UML (Revisto).....	5
IV.	Esquema Relacional.....	6
V.	Análise Dependências Funcionais e Formas Normais.....	
VI.		

Diagrama UML (Revisto)



Esquema Relacional

- **Distrito**(codDistrito, nome);
- **Concelho**(idConcelho, codConcelho, nome, codNUTS3, codDistrito -> Distrito);
- **Escola**(idEscola, nome, codDGAE, codDGEEC, tipo, idConcelho -> Concelho);
- **Aluno**(idAluno, nome, sexo, dataNascimento, serInterno, anoEscolaridade, codCurso -> Curso, idSitFrq -> sitFreq);
- **OndeRealiza**(idEscola -> Escola, idAluno -> Aluno);
- **Exame**(idExame, fase, anoLetivo, dataRealizacao, codDisciplinaExame -> DisciplinaExame);
- **DisciplinaExame**(codDisciplinaExame, disciplina, anoEscolaridade)
- **AlunoRealiza**(idAluno -> Aluno, idExame -> Exame, classInt, notaExame, classFinal, serInterno, paraAprov, paraMelhoria, provaIngresso, CFCEPE, sitFrequencia, anoEscolarAluno);
- **Curso**(codCurso, nome, codSubTipoCurso -> SubTipoCurso);
- **SubTipoCurso**(codSubTipoCurso, nome, codTipoCurso-> TipoCurso);
- **TipoCurso**(codTipoCurso, nome, anoEscInicio, anoEscFinal).

Análise de Dependências Funcionais e de Formas Normais

- **Distrito**(codDistrito, nome):
 - **FDs:**
codDistrito -> nome
nome -> codDistrito
Keys: {codDistrito}, {nome}

Formas: BCNF? Sim
3NF? Sim
- **Concelho**(idConcelho, codConcelho, nome, codNUTS3, codDistrito -> Distrito):
 - **FDs:**
idConcelho -> codConcelho, nome, codDistrito, codNUTS3
nome -> idConcelho, codConcelho, codDistrito, codNUTS3
Keys: {idConcelho}, {nome}

Formas: BCNF? Sim
3NF? Sim
- **Escola**(idEscola, nome, codDGAE, codDGEEC, tipo, idConcelho -> Concelho):
 - **FDs:**
idEscola -> nome, codDGAE, codDGEEC, tipo, idConcelho
codDGAE -> idEscola, nome, codDGEEC, tipo, idConcelho
codDGEEC -> idEscola, nome, codDGAE, tipo, idConcelho
Keys: {idEscola}, {codDGAE}, {codDGEEC}

Formas: BCNF? Sim
3NF? Sim
- **Aluno**(idAluno, nome, sexo, dataNascimento, codCurso -> Curso):
 - **FDs:**
idAluno -> nome, sexo, dataNascimento, serInterno, anoEscolaridade, codCurso
Key: {idAluno}

Formas: BCNF? Sim
3NF? Sim
- **OndeRealiza**(idEscola -> Escola, idAluno -> Aluno):
 - **FDs:** -

Formas: BCNF? Sim
3NF? Sim

- **Exame**(idExame, fase, anoLetivo, dataRealizacao, codDisciplinaExame -> DisciplinaExame):

- **FDs:**
idExame -> fase, anoLetivo, dataRealizacao
Key: {idExame}

Formas: BCNF? Sim
3NF? Sim

- **DisciplinaExame**(codExame, disciplina, anoEscolaridade):

- **FDs:**
codExame -> disciplina, anoEscolaridade
Key: {codExame}

Formas: BCNF? Sim
3NF? Sim

- **AlunoRealiza**(idAluno -> Aluno, idExame -> Exame, classInt, notaExame, classFinal, serInterno, paraAprov, paraMelhoria, provaIngresso, CFCEPE, sitFrequencia, anoEscolarAluno):

- **FDs:**
idAluno, idExame -> classInt, notaExame, classFinal, serInterno, paraAprov, paraMelhoria, provaIngresso, CFCEPE, sitFrequencia, anoEscolarAluno
classInt, notaExame -> classFinal

Key: {idAluno, idExame}

Formas: BCNF? Não
3NF? Não

A relação viola a Forma Normal de Boyce-Codd, visto que classInt e notaExame não são uma '(super)key'. Para além disso, esta também viola a 3ª Forma Normal, uma vez que o atributo classFinal não é um atributo primo (não é um membro de nenhuma 'key').

- **Decomposição para Forma Normal de Boyce-Codd:**

FD que viola BCFN: classInt, notaExame -> classFinal.

$\{classInt, notaExame\}^+ = \{classInt, notaExame, classFinal\}$

S1(classInt, notaExame, classFinal)

S2(classInt, notaExame, serInterno, paraAprov, paraMelhoria, provaIngresso, CFCEPE, sitFrequencia, anoEscolarAluno, idAluno, idExame)

FDs para S1:

classInt, notaExame -> classFinal

Key: {classInt, notaExame}

FDs para S2:

idAluno, idExame -> classInt, notaExame, serInterno, paraAprov, paraMelhoria, provaIngresso, CFCEPE, sitFrequencia, anoEscolarAluno
sitFrequencia -> serInterno

Key: {idAluno, idExame}

Formas: BCNF? Não

3NF? Não

FD que viola BCFN: sitFrequencia -> serInterno

$\{sitFrequencia\}^+ = \{sitFrequencia, serInterno\}$

S3(sitFrequencia, serInterno)

S4(sitFrequencia, classInt, notaExame, paraAprov, paraMelhoria, provaIngresso, CFCEPE, anoEscolarAluno, idAluno, idExame)

FDs para S3:

sitFrequencia -> serInterno

Key: {sitFrequencia}

FDs para S4:

idAluno, idExame -> classInt, notaExame, paraAprov, paraMelhoria, provaIngresso, CFCEPE, sitFrequencia, anoEscolarAluno

Key: { idAluno, idExame }

S1, S3 e S4 encontram-se então em BCNF, pois em cada dependência funcional destas duas relações, o lado esquerdo é uma 'key'.

- **Curso**(codCurso, nome, codSubTipoCurso -> SubTipoCurso):

- **FDs:**

codCurso -> nome, codSubTipoCurso

nome -> codCurso, codSubTipoCurso

Keys: {codCurso}, {nome}

Formas: BCNF? Sim
3NF? Sim.

- **SubTipoCurso**(codSubTipoCurso, nome, codTipoCurso -> TipoCurso):

- **FDs:**
codSubTipoCurso -> nome, codTipoCurso
nome -> codSubTipoCurso, codTipoCurso
Keys: {codSubTipoCurso, nome}

Formas: BCNF? Sim
3NF? Sim

- **TipoCurso**(codTipoCurso, nome, anoEscInicio, anoEscFinal):

- **FDs:**
codTipoCurso -> nome, anoEscInicio, anoEscFinal
nome -> codTipoCurso, anoEscInicio, anoEscFinal
Keys: {codTipoCurso}, {nome}

Formas: BCNF? Sim
3NF? Sim