

Dependências Funcionais:

- Regra(id_regra) -> Regra(descrição)
- Infraestrutura(id_infraestrutura) -> Infraestrutura(nome)
- id_sala -> Sala(capacidade)
- UnidadeEscolar(código) -> UnidadeEscolar(cidade, estado, país, bloco, Localidade_Especific)
- Turma(id_turma) -> Turma(capacidade)
- Bolsa(id_bolsa) -> Bolsa(valor_bolsa, descrição, instituição)
- Desconto(id_desconto) -> Desconto(valor, motivo)
- Oferecimento(código_oferecimento) -> Oferecimento(código, id_turma, periodo)
- Oferecimento(código_oferecimento, código, id_turma) -> Oferecimento(periodo)
- Matrícula(data, código_oferecimento) -> Aluno(nomeAluno, sobrenomeAluno, telefoneAluno)
- Matrícula(nomeAluno, sobrenomeAluno, telefoneAluno, data) -> Matrícula(status, data_limite, valor_matricula)
- DepartamentoAcadêmico(código) -> DepartamentoAcadêmico(nome, nomeProfessor, sobrenomeProfessor, telefoneProfessor)
- Curso(código) -> Curso(carga_horária, total_vagas, nível, id_sala, codigoDepartamento, códigoUnidadeEscolar)
- Disciplina(código) -> Disciplina(num_aulas_semanais, material_recomendado, códigoUnidadeEscolar, nomeProfessor, sobrenomeProfessor, telefoneProfessor)
- AvisoGeral(id_aviso) -> AvisoGeral(texto, timestamp, nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)
- Mensagem(id_mensagem) -> Mensagem(timestamp, texto, nomeAluno, sobrenomeAluno, telefoneAluno, nomeProfessor, sobrenomeProfessor, telefoneProfessor, nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)
- Avaliação(id_avaliação) -> Avaliação(texto, didática, nomeAluno, sobrenomeAluno, telefoneAluno, nomeProfessor, sobrenomeProfessor, telefoneProfessor, código_oferecimento)
- AvisoGeral(id_aviso, texto, timestamp, nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

- Aluno(nomeAluno, sobrenomeAluno, telefoneAluno) -> Aluno(dataNasc, sexo, senha, códigoUnidadeEscolar, rua, cidade, estado, e-mail)
- Aluno(nomeAluno, sobrenomeAluno, telefoneAluno, e-mail) -> Aluno(dataNasc, sexo, senha, códigoUnidadeEscolar, rua, cidade, estado)
- Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor) -> Professor(dataNasc, sexo, senha, área_especialização, titulação, códigoUnidadeEscolar, rua, cidade, estado, e-mail)
- Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor, e-mail) -> Professor(dataNasc, sexo, senha, área_especialização, titulação, códigoUnidadeEscolar, rua, cidade, estado)
- FuncionárioAdministrativo(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario) -> FuncionárioAdministrativo(dataNasc, sexo, senha, rua, cidade, estado, e-mail)
- FuncionárioAdministrativo(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario, e-mail) -> FuncionárioAdministrativo(dataNasc, sexo, senha, rua, cidade, estado)

1. Primeira forma normal

Antes de começar o passo 1 do mapeamento ignoramos todos os atributos que não eram atômicos monovalorados. Após isso, no sétimo passo, transformamos todos os atributos não atômicos monovalorados em atômicos monovalorados. Sendo assim, é garantido que todas as relações estão na primeira forma normal.

2. Segunda e Terceira forma normal

Como queremos no final estar na terceira forma normal não iremos tentar fazer deduções de dependências transitivas, já que, essas, teriam que ser retiradas indo da segunda para a terceira.

Baseado nas dependências formais, todas as relações já estão na terceira forma normal, menos uma, que é a matrícula.

As dependências formais para ela são:

- Matrícula(data, código_oferecimento) -> Aluno(nomeAluno, sobrenomeAluno, telefoneAluno)
- Matrícula(nomeAluno, sobrenomeAluno, telefoneAluno, data) -> Matrícula(status, data_limite, valor_matricula)

E ela está:

Matricula(nomeAluno, sobrenomeAluno, telefoneAluno, data, código_oferecimento, status, data_limite, valor_matricula)

Sendo assim, aplicando o algoritmo para transformar para a segunda forma normal, que neste caso, também vai estar na terceira forma normal:

Matrícula(nomeAluno, sobrenomeAluno, telefoneAluno, data, status, data_limite, valor_matricula)

MatriculaDisciplina(data, código_oferecimento, nomeAluno, sobrenomeAluno, telefoneAluno)

3. BCNF

As únicas relações que contém chaves candidatas são: Aluno, Professor, FuncionárioAdministrativo e Oferecimento. Para todas elas, as dependências funcionais já mostram que elas estão na BCNF, já que, todos os atributos não chaves dependem da chave inteira e das chaves candidatas.

4. Esquema relacional final

Regra(id_regra, descrição)

Infraestrutura(id_infraestrutura, nome)

Sala(id_sala, capacidade)

UnidadeEscolar(código, cidade, estado, país, bloco, Localidade_Especificada)

Turma(id_turma, capacidade)

Bolsa(id_bolsa, valor_bolsa, descrição, instituição)

Desconto(id_desconto, valor, motivo)

Ministra(id_sala, id_turma, dia, hora_inicio, hora_fim)

☐ id_sala -> FK -> Sala(id_sala)

☐ id_turma -> FK -> Turma(id_turma)

Oferecimento(código_oferecimento, código, id_turma, periodo)

☐ código -> FK -> Disciplina(código)

☐ id_turma -> FK -> Turma(id_turma)

Matricula(nomeAluno, sobrenomeAluno, telefoneAluno, data, código_oferecimento, status, data_limite, valor_matricula)

☐ (nomeAluno, sobrenomeAluno, telefoneAluno) -> FK -> Aluno(nome, sobrenome, telefone)

☐ código_oferecimento -> FK -> Oferecimento(código_oferecimento)

Matrícula(nomeAluno, sobrenomeAluno, telefoneAluno, data, status, data_limite, valor_matricula)

- ☐ (nomeAluno, sobrenomeAluno, telefoneAluno) -> FK -> Aluno(nome, sobrenome, telefone)

MatriculaDisciplina(data, código_oferecimento, nomeAluno, sobrenomeAluno, telefoneAluno)

- ☐ código_oferecimento -> FK -> Oferecimento(código_oferecimento)
- ☐ data -> FK -> Matrícula(data)
- ☐ (nomeAluno, sobrenomeAluno, telefoneAluno) -> FK -> Aluno(nomeAluno, sobrenomeAluno, telefoneAluno)

DepartamentoAcadêmico(código, nome, nomeProfessor, sobrenomeProfessor, telefoneProfessor)

- ☐ (nomeProfessor, sobrenomeProfessor, telefoneProfessor) -> FK -> Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor)

Curso(código, nome, carga_horária, total_vagas, nível, id_sala, codigoDepartamento, códigoUnidadeEscolar)

- ☐ id_sala -> FK -> Sala(id_sala)
- ☐ codigoDepartamento -> FK -> DepartamentoAcadêmico(código)
- ☐ códigoUnidadeEscolar -> FK -> UnidadeEscolar(código)

Disciplina(código, num_aulas_semanais, material_recomendado, códigoUnidadeEscolar, nomeProfessor, sobrenomeProfessor, telefoneProfessor)

- ☐ códigoUnidadeEscolar -> FK -> UnidadeEscolar(código)
- ☐ nomeProfessor, sobrenomeProfessor, telefoneProfessor -> FK -> Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor)

AvisoGeral(id_aviso, texto, timestamp, nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

- ☐ (nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario) -> FK -> FuncionárioAdministrativo(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

Mensagem(id_mensagem, timestamp, texto, nomeAluno, sobrenomeAluno, telefoneAluno, nomeProfessor, sobrenomeProfessor, telefoneProfessor, nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

- ☐ (nomeAluno, sobrenomeAluno, telefoneAluno) -> FK -> Aluno(nomeAluno, sobrenomeAluno, telefoneAluno)
- ☐ (nomeProfessor, sobrenomeProfessor, telefoneProfessor) -> FK -> Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor)
- ☐ (nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario) -> FK -> FuncionarioAdministrativo(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

Avaliação(id_avaliação, texto, didática, nomeAluno, sobrenomeAluno, telefoneAluno, nomeProfessor, sobrenomeProfessor, telefoneProfessor, código_oferecimento)

- ☐ (nomeAluno, sobrenomeAluno, telefoneAluno) -> FK Aluno(nomeAluno, sobrenomeAluno, telefoneAluno)
- ☐ (nomeProfessor, sobrenomeProfessor, telefoneProfessor) -> FK Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor)
- ☐ código_oferecimento -> FK Oferecimento(código_oferecimento)

AvisoGeral(id_aviso, texto, timestamp, nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

- ☐ (nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario) -> FK FuncionarioAdministrativo(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

ContemRegra(id_regra, códigoCurso)

- ☐ id_regra -> FK Regra(id_regra)
- ☐ códigoCurso -> FK Curso(código)

PrecisaInfraestrutura(id_infraestrutura, códigoCurso)

- ☐ id_infraestrutura -> FK Infraestrutura(id_infraestrutura)
- ☐ códigoCurso -> FK Curso(código)

CursoTemDisciplina(códigoCurso, códigoDisciplina)

- ☐ códigoCurso -> FK Curso(código)
- ☐ códigoDisciplina -> FK Disciplina(código)

CursoPreRequisitoCurso(códigoCurso, códigoCursoPreRequisito)

- ☐ códigoCurso -> FK Curso(código)
- ☐ códigoCursoPreRequisito -> FK Curso(código)

DisciplinaPreRequisitoCurso(códigoCurso, códigoDisciplina)

- ☐ códigoCurso -> FK Curso(código)
- ☐ códigoDisciplina -> FK Disciplina(código)

TurmaRecebeAviso(id_turma, id_aviso)

- ☐ id_turma -> FK Turma(id_turma)
- ☐ id_aviso -> FK AvisoGeral(id_aviso)

TurmaRecebeMensagem(id_turma, id_mensagem)

- ☐ id_turma -> FK Turma(id_turma)
- ☐ id_mensagem -> FK AvisoGeral(id_mensagem)

AlunoRecebeMensagem(nomeAluno, sobrenomeAluno, telefoneAluno, id_mensagem)

- ☐ (nomeAluno, sobrenomeAluno, telefoneAluno) -> FK Aluno(nomeAluno, sobrenomeAluno, telefoneAluno)

☐ id_mensagem -> FK -> Mensagem(id_mensagem)

ProfessorRecebeMensagem(nomeProfessor, sobrenomeProfessor, telefoneProfessor, id_mensagem)

☐ (nomeProfessor, sobrenomeProfessor, telefoneProfessor) -> FK -> Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor)

☐ id_mensagem -> FK -> Mensagem(id_mensagem)

FuncionarioRecebeMensagem(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario, id_mensagem)

☐ (nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario) -> FK -> Funcionario(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario)

☐ id_mensagem -> FK -> Mensagem(id_mensagem)

MatriculaTemBolsa(código_oferecimento, data, nomeAluno, sobrenomeAluno, telefoneAluno, id_bolsa)

☐ data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento -> FK -> Matrícula(data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento)

☐ id_bolsa -> FK -> Bolsa(id_bolsa)

MatriculaTemBolsa(código_oferecimento, data, nomeAluno, sobrenomeAluno, telefoneAluno, id_desconto)

☐ data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento -> FK -> Matrícula(data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento)

☐ id_desconto -> FK -> Desconto(id_desconto)

Notas(nomeAluno, sobrenomeAluno, telefoneAluno, data, código_oferecimento, nota)

☐ data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento -> FK -> Matrícula(data, nomeAluno, sobreoneAluno, telefoneAluno, código_oferecimento)

Aluno(nomeAluno, sobrenomeAluno, telefoneAluno, dataNasc, sexo, senha, e-mail, códigoUnidadeEscolar, rua, cidade, estado)

☐ códigoUnidadeEscolar -> FK -> UnidadeEscolar(código)

Professor(nomeProfessor, sobrenomeProfessor, telefoneProfessor, dataNasc, sexo, senha, e-mail, área_especialização, titulação, códigoUnidadeEscolar, rua, cidade, estado)

☐ códigoUnidadeEscolar -> FK -> UnidadeEscolar(código)

FuncionárioAdministrativo(nomeFuncionario, sobrenomeFuncionario, telefoneFuncionario, dataNasc, sexo, senha, e-mail, rua, cidade, estado)