



UNIVERSIDADE D
COIMBRA

Faculdade de Ciências e Tecnologias

Base de Dados - Licenciatura em Engenharia Informática

BD

Trabalho realizado por:

Teodoro Marques e Matheus D'Oliveira

Dezembro, 2024

PL: 4

Índice

Introdução.....	3
Descrição do Projetos.....	3
Detalhamento.....	4
Detalhes do Modelo Conceitual.....	4
Detalhes do Modelo Físico.....	6
Diagrama de Relações e Modelo Físico.....	12
Relações no Modelo Físico.....	12
Tabela de Relações do Modelo Físico.....	12
Explicações das Relações no Modelo Físico.....	15
Por que essas relações são necessárias.....	16
Transações.....	17
Exemplo Prático: Matrícula em uma Edição de Curso.....	17
Concorrência.....	17
Refinamento Técnico: Chaves Primárias Compostas.....	18
Problemas Potenciais:.....	18
Alternativa Mais Eficiente.....	18
Quando Usar PK Composta?.....	18
Conclusão.....	19

Realização:

Teodoro Marques - 2023211717 - teomarques@gmail.com

Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

Introdução

Aqui, vamos explicar a nossa linha de raciocínio que nos levou à versão final do ER Diagram pedido no projeto. Um aspecto importante é garantir data consistency e respeitar as relações entre as entidades. Vamos também explicar como serão feitas as transações e como é gerida a concorrência. Iremos detalhar melhor mais à frente.

Descrição do Projetos

O objetivo do projeto é criar um sistema para administrar o banco de dados de uma universidade, – similar ao Inforestudante, permitindo que a equipe realize transações e acesse informações sobre:

- Cursos
- Cadeiras
- Edições das courses
- Alunos
- Instrutores
- Atividades extracurriculares
- Aulas

As transações incluem inserir e remover todas as entidades, com o desafio de garantir que a remoção de uma entidade, como um aluno, também remova dados relacionados, como matrículas em edições de courses.

Realização:

Teodoro Marques - 2023211717 - teomarques@gmail.com

Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

Detalhamento

Detalhes do Modelo Conceitual

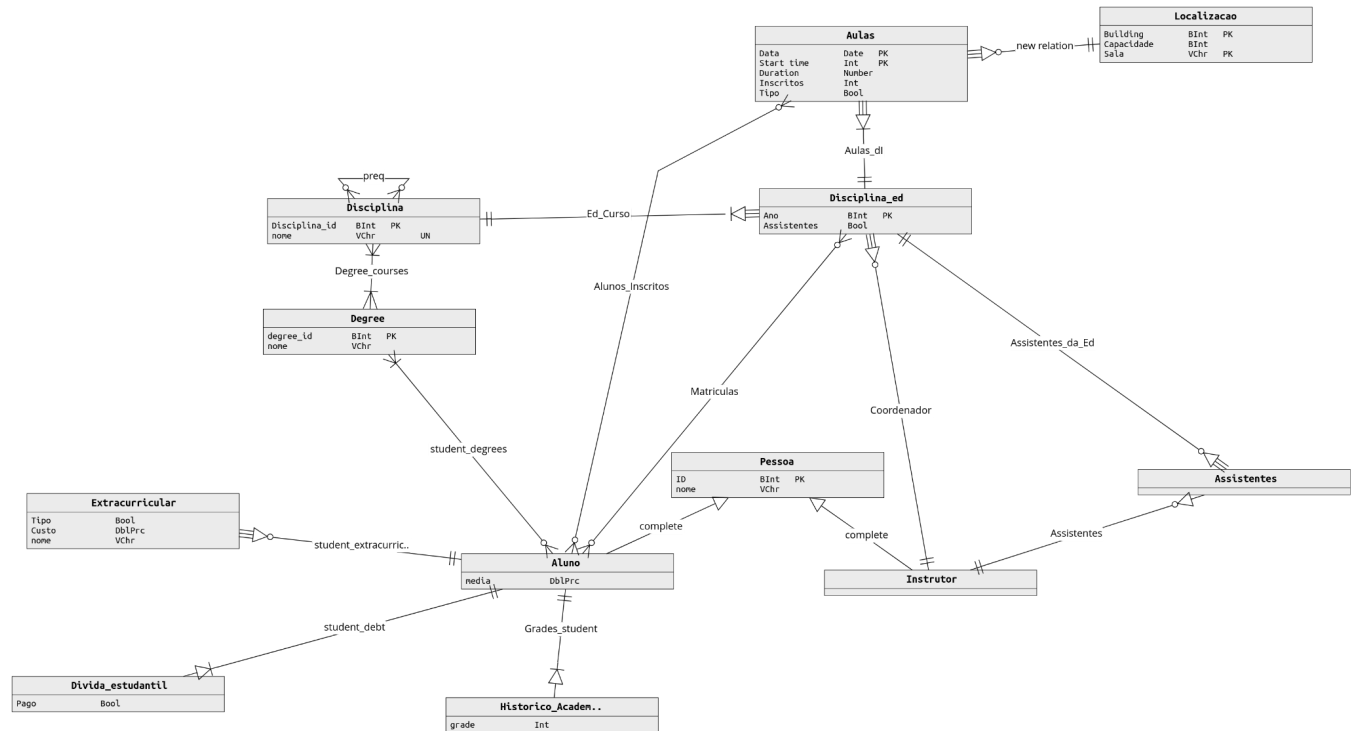


figura 4.1: Conceptual model

O modelo conceitual, (figura 4.1), define as entidades e suas relações de alto nível. A imagem não apresenta grande qualidade, entretanto também se encontra anexa na entrega para melhor visualização.

Abaixo, detalhamos cada entidade e suas relações:

- **AULAS (Classes):**
 - Atributos: Data, Hora de Início, Duração, Inscritos, Tipo (é boolean pois pode tanto ser teórica -> true como ser prático-laboratorial -> false)
 - Relações: Cada aula está associada a uma course (edição) e a uma LOCALIZACAO.
- **LOCALIZACAO (Localização):**
 - Atributos: building, capacidade, sala
 - Relações: Uma localização pode ter várias AULAS.
- **course (cadeira):**
 - Atributos: course_id, Nome
 - Relações: Cada course pertence a um ou mais Degrees e está relacionada a uma ou mais edições de course (course_ed)

Realização:

Teodoro Marques - 2023211717 - teomarques@gmail.com

Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

- **Degree (Curso):**
 - Atributos: Nome, Degree_ID
 - Relações: Um curso de grau pode ter uma ou vários COURSES e está associado a vários Alunos_inscritos.
- **Aluno (Alunos Matriculados):**
 - Atributos: media (média académica)
 - Relações: Subgrupo de Pessoa. um aluno pode ter várias aulas, várias matrículas (0 ou mais edições de course), um ou mais degrees (cursos), vários extracurriculares, pode ter uma dívida estudantil e um histórico académico.
- **course_ed (Edição da cadeira):**
 - Atributos: Ano, Assistentes
 - Relações: Cada edição está associada a um Coordenador e a vários Assistentes. Também tem uma course relacionada, pode ter vários alunos e tem várias aulas.
- **Instrutor (Coordenador):**
 - Relações: Cada coordenador está ligado a uma course_ed, que é gereciada por ele. O coordenador é um instrutor, e definimos o mesmo na tabela, que herda as características de uma pessoa. O coordenador é a relação entre o instrutor e uma edição de course. Um instrutor coordenador pode ter zero ou mais assistentes.
- **Assistentes (Assistentes de Edição):**
 - Relações: Vários assistentes podem estar associados a uma course_ed. Um assistente é assistente de um instrutor coordenador.
- **PESSOA (Pessoa):**
 - Atributos: ID, Nome
 - Subtipos: Assistentes, Aluno, Instrutor
- **DIVIDA_ESTUDANTIL (Dívida do Aluno):**
 - Atributos: Pago (é boolean pois pode tanto ter sido -> true como não ter sido -> false paga).
 - Relações: Cada dívida está ligada a um Aluno, e a relação entre a dívida e o aluno é aluno_divida.
- **Extracurricular (aluno_extracurric.):**
 - Atributos: Tipo (boolean para atribuir clubs ou sports), Custo, Nome
 - Relações: Cada aluno pode participar de várias atividades extracurriculares aluno_extracurric é a relação entre o estudante e suas extracurriculares.
- **Historico_Academico (Grades_aluno):**
 - Atributos: grade (nota)
 - Relações: Cada histórico está ligado a um e só um aluno. Sua relação fornece Grades_aluno.

Realização:

Teodoro Marques - 2023211717 - teomarques@gmail.com

Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

[illegible]

O modelo físico, mostrado na figura 6.1, traduz o modelo conceitual em tabelas de banco de dados reais. É um diagrama ER que mostra as entidades e suas relações de alto nível. Abaixo, detalhamos cada entidade, seus atributos e relações:

- **Atributos:**
 - *data*: DATE (data da aula)
 - *start_time*: TIME (hora de início)
 - *duration*: INT (duração da aula)
 - *inscritos*: INT (número de inscritos)
 - *tipo*: BOOL (indica se a aula é teórica -> true ou prático-laboratorial -> false)
 - *localizacao_building*: VCHR, FK (referencia LOCALIZACAO)
 - *localizacao_sala*: VCHR, FK (referencia LOCALIZACAO)
 - *course_ed_ano*: BINT, FK (referencia COURSE_ED)
 - *course_ed_course_id*: BINT, FK (referencia COURSE_ED)
 - *course_ed_instr_pessoa_id*: BINT, FK (referencia COURSE_ED)
- **Chave Primária (PK):** Combinada (*data*, *start_time*, *localizacao_building*, *localizacao_sala*, *course_ed_ano*, *course_ed_course_id*, *course_ed_instr_pessoa_id*)
- **Relações:** Cada aula está associada a uma edição de curso (*COURSE_ED*) através das chaves estrangeiras *course_ed_ano*, *course_ed_course_id* e *course_ed_instr_pessoa_id*, e a uma localização (*LOCALIZACAO*) através de *localizacao_building* e *localizacao_sala*.

Teodoro Marques - 2023211717 - teomarques@gmail.com
Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

LOCALIZACAO (Localização):

- **Atributos:**
 - *building*: BINT (identificador do prédio)
 - *capacidade*: INT (capacidade da sala)
 - *sala*: VCHR (identificador da sala)
- **Chave Primária (PK):** Combinada (*building*, *sala*)
- **Relações:** Uma localização pode estar associada a várias aulas (*AULAS*), referenciada pelas chaves estrangeiras *localizacao_building* e *localizacao_sala*.

COURSE (Cadeira):

- **Atributos:**
 - *course_id*: BINT (identificador da cadeira)
 - *nome*: VCHR (nome da cadeira)
- **Chave Primária (PK):** *course_id*
- **Relações:** Cada cadeira (*COURSE*) está relacionada a uma ou mais edições (*COURSE_ED*) através do atributo *course_id* (FK) e pertence a um ou mais cursos de grau (*DEGREE*) através da tabela *DEGREE_COURSES*.

DEGREE (Curso):

- **Atributos:**
 - *degree_id*: BINT (identificador do curso)
 - *nome*: VCHR (nome do curso)
- **Chave Primária (PK):** *degree_id*
- **Relações:** Um curso de grau (*DEGREE*) pode ter várias cadeiras (*COURSE*) através da tabela *DEGREE_COURSES*, referenciada por *degree_id* (FK), e está associado a vários alunos (*ALUNO*) através da tabela *ALUNO_DEGREES*.

ALUNO (Alunos Matriculados):

- **Atributos:**
 - *pessoa_id*: BINT (identificador da pessoa, herdado de *PESSOA*)
 - *media*: INT (média acadêmica)
- **Chave Primária (PK):** *pessoa_id*
- **Relações:** Subtipo de *PESSOA*. Um aluno pode estar associado a várias aulas (*AULAS*) através de uma relação implícita, várias matrículas em edições de cadeiras (*COURSE_ED*) via *COURSE_ED_ALUNO*, um ou mais cursos (*DEGREE*) via *ALUNO_DEGREES*, várias atividades extracurriculares (*EXTRACURRICULAR*) via *ALUNO_EXTRACURRICULAR*, uma dívida estudantil (*DIVIDA_ESTUDANTIL*) e um histórico acadêmico (*HISTORICO_ACADEMICO*).

COURSE_ED (Edição da Cadeira):

- **Atributos:**
 - *ano*: BINT (ano da edição)
 - *course_id*: BINT, FK (referencia *COURSE*)

Realização:

Teodoro Marques - 2023211717 - teomarques@gmail.com

Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

- *instr_pessoa_id*: BINT, FK (referencia *INSTRUTOR*)
- **Chave Primária (PK)**: Combinada (*ano*, *course_id*, *instr_pessoa_id*)
- **Relações**: Cada edição (*COURSE_ED*) está associada a uma cadeira (*COURSE*) via *course_id* (FK), a um coordenador (*INSTRUTOR*) via *instr_pessoa_id* (FK), a vários assistentes (*ASSISTENTES*) via *ASSISTENTES_ED*, a vários alunos (*ALUNO*) via *COURSE_ED_ALUNO* e a várias aulas (*AULAS*).

INSTRUTOR (Coordenador):

- **Atributos**:
 - *pessoa_id*: BINT (identificador da pessoa, herdado de *PESSOA*)
- **Chave Primária (PK)**: *pessoa_id*
- **Relações**: Subtipo de *PESSOA*. Um instrutor pode ser o coordenador de uma edição de cadeira (*COURSE_ED*), referenciado por *instr_pessoa_id* (FK), e pode ter zero ou mais assistentes (*ASSISTENTES*) via *ASSISTENTES_ED*.

ASSISTENTES (Assistentes de Edição):

- **Atributos**:
 - *pessoa_id*: BINT (identificador da pessoa, herdado de *PESSOA*)
- **Chave Primária (PK)**: *pessoa_id*
- **Relações**: Subtipo de *PESSOA*. Vários assistentes podem estar associados a uma edição de cadeira (*COURSE_ED*) através da tabela *ASSISTENTES_ED*, referenciada por *assistentes_pessoa_id* (FK).

PESSOA (Pessoa):

- **Atributos**:
 - *id*: BINT (identificador da pessoa)
 - *nome*: VCHR (nome da pessoa)
- **Chave Primária (PK)**: *id*
- **Subtipos**: *ALUNO*, *INSTRUTOR*, *ASSISTENTES*

DIVIDA_ESTUDANTIL (Dívida do Aluno):

- **Atributos**:
 - *aluno_pessoa_id*: BINT, FK (referencia *ALUNO*)
 - *pago*: BOOL (indica se a dívida foi paga -> true ou não -> false)
- **Chave Primária (PK)**: *aluno_pessoa_id*
- **Relações**: Cada dívida está ligada a um aluno (*ALUNO*) através de *aluno_pessoa_id* (FK).

EXTRACURRICULAR (Atividades Extracurriculares):

- **Atributos**:
 - *tipo*: BOOL (indica se é clube -> true ou esporte -> false)
 - *custo*: DBLPRC (custo da atividade)
 - *nome*: VCHR (nome da atividade)

Realização:

Teodoro Marques - 2023211717 - teomarques@gmail.com

Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

- **Chave Primária (PK):** *nome*
- **Relações:** Cada atividade extracurricular pode estar associada a vários alunos (*ALUNO*) através da tabela *ALUNO_EXTRACURRICULAR*, referenciada por *extracurricular_nome* (FK).

HISTORICO_ACADEMICO (Histórico Acadêmico):

- **Atributos:**
 - *aluno_pessoa_id*: BINT, FK (referencia *ALUNO*)
 - *grade*: INT (nota do aluno)
- **Chave Primária (PK):** *aluno_pessoa_id*
- **Relações:** Cada histórico está ligado a um e apenas um aluno (*ALUNO*) através de *aluno_pessoa_id* (FK).

Tabelas de Relação:

- **DEGREE_COURSES:** Relaciona *DEGREE* e *COURSE* (*degree_id*: BINT, PK, FK; *course_id*: BINT, PK, FK).
- **ALUNO_DEGREES:** Relaciona *ALUNO* e *DEGREE* (*aluno_pessoa_id*: BINT, PK, FK; *degree_id*: BINT, PK, FK).
- **COURSE_ED_ALUNO:** Relaciona *COURSE_ED* e *ALUNO* (*course_ed_ano*: BINT, PK, FK; *course_ed_course_id*: BINT, PK, FK; *course_ed_instr_pessoa_id*: BINT, PK, FK; *aluno_pessoa_id*: BINT, PK, FK).
- **ASSISTENTES_ED:** Relaciona *ASSISTENTES* e *COURSE_ED* (*course_ed_ano*: BINT, PK, FK; *course_ed_course_id*: BINT, PK, FK; *course_ed_instr_pessoa_id*: BINT, PK, FK; *assistentes_pessoa_id*: BINT, PK, FK).
- **ALUNO_EXTRACURRICULAR:** Relaciona *ALUNO* e *EXTRACURRICULAR* (*aluno_pessoa_id*: BINT, PK, FK; *extracurricular_nome*: VCHR, PK, FK).

Este modelo físico reflete a estrutura relacional do banco de dados, com chaves primárias (PK) e estrangeiras (FK) definidas para garantir a integridade das relações entre as tabelas.

Realização:

Teodoro Marques - 2023211717 - teomarques@gmail.com

Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

Explicação das Relações no Modelo Conceitual

Descrição sobre as tabelas do Conceptual Model

Entidade	Atributos	Relações
AULAS (Classes)	ID, Data, Hora de Início, Duração, Tipo	1 AULAS para 1 course; 1 AULAS para 1 LOCALIZACAO
LOCALIZACAO (Localização)	pint (ID), new_relation, Capacidade_Sala, VChr	1 LOCALIZACAO para muitas AULAS
course (course)	Nome, Ed_Curso	1 course para 1 Degree_courses; 1 course para muitas AULAS
Degree_courses (Cursos de Grau)	Nome, Degree_ID	1 Degree_courses para muitas courseS; Muitas Degree_courses para muitos Alunos_inscritos
Alunos_inscritos (Alunos Matriculados)	-	Tabela de junção entre Degree_courses e Aluno
course_ed (Edição da course)	Apo, Assistentes_ed	1 course_ed para 1 Coordenador; 1 course_ed para muitos Assistentes_da_Ed
Coordenador (Coordenador)	-	1 Coordenador para 1 course_ed
Assistentes_da_Ed (Assistentes de Edição)	-	Muitos Assistentes_da_Ed para 1 course_ed
PESSOA (Pessoa)	ID, Nome	Subtipos: Assistentes, Aluno, Instrutor
Aluno (Aluno)	Média Estudantil	Muitos Aluno para muitos Alunos_inscritos; 1 Aluno para muitas aluno_divida, aluno_extracurric, grades_aluno
aluno_divida (Dívida do Aluno)	-	Muitas aluno_divida para 1 Aluno; 1 aluno_divida para 1 DIVIDA_ESTUDANTIL
aluno_extracurric (Atividades Extracurriculares)	Tipo, Custo, Nome	Muitas aluno_extracurric para 1 Aluno
grades_aluno (Notas do Aluno)	-	Muitas grades_aluno para 1 Aluno; Muitas grades_aluno para muitos Historico_Acad
Historico_Acad (Histórico Acadêmico)	Nota	Muitos Historico_Acad para muitas grades_aluno
DIVIDA_ESTUDANTIL (Dívida Estudantil)	PAGO	1 DIVIDA_ESTUDANTIL para muitas aluno_divida

Realização:

Teodoro Marques - 2023211717 - teomarques@gmail.com

Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

Descrição detalhada:

- **AULAS e Course_ed:** Cada aula está vinculada a uma edição específica de um curso (Course_ed), organizando as aulas por edições. Exemplo: uma aula de "Análise III" de 2024 pertence à edição correspondente.
- **AULAS e Localizacao:** Cada aula acontece em um local específico, como uma sala, com detalhes como capacidade e prédio.
- **Course e Degree:** Um curso (Course) pode integrar vários programas de graduação (Degree), como Engenharia Informática ou Física, indicando sua abrangência acadêmica.
- **Course e Course_ed:** Cada curso (Course) possui várias edições (Course_ed), representando ofertas em diferentes períodos.
- **Course_ed e Instrutor (Coordenador):** Cada edição de curso tem um instrutor como coordenador, responsável pela sua gestão.
- **Course_ed e Assistentes:** Uma edição de curso pode contar com vários assistentes, apoiando o coordenador.
- **Aluno e Course_ed (via Matrículas):** Alunos podem se matricular em várias edições de cursos, e cada edição pode ter múltiplos alunos.
- **Aluno e Degree (via aluno_degrees):** Um aluno pode estar em vários programas de graduação, e cada programa pode ter vários alunos.
- **Aluno e AULAS (via Alunos_Inscritos):** Alunos se inscrevem em várias aulas, e cada aula pode ter múltiplos inscritos.
- **Aluno e Extracurricular (via aluno_extracurriculars):** Um aluno participa de várias atividades extracurriculares, e cada atividade pode incluir vários alunos.
- **Aluno e Divida_estudantil (via aluno_divida):** Cada aluno pode ter uma dívida estudantil associada, indicando seu status de pagamento.
- **Aluno e Historico_Academico (via Grades_aluno):** As notas de um aluno são registradas no histórico acadêmico, rastreando seu desempenho.
- **Course e Course (via preq):** Um curso pode ter pré-requisitos, outros cursos necessários antes de sua realização. Ou seja, para se fazer a cadeira X do segundo ano, deve se ter concluído a cadeira Y do primeiro ano de antemão (não sei exemplos do nosso curso para isso).
- **Pessoa e subtipos (Aluno, Instrutor, Assistentes):** Pessoa é uma entidade geral que abrange alunos, instrutores e assistentes, cada um com atributos próprios.

Realização:

Teodoro Marques - 2023211717 - teomarques@gmail.com

Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

Diagrama de Relações e Modelo Físico

Relações no Modelo Físico

As relações são implementadas por meio de chaves primárias (PK) e chaves estrangeiras (FK). A tabela abaixo mapeia cada tabela física, a entidade conceitual que lhe deu origem, suas FKs (quando existirem) e os principais atributos (incluindo chaves primárias).

Tabela de Relações do Modelo Físico

Tabela Física	Entidade Conceitual Correspondente	Foreign Keys (FK)	Atributos Principais
aulas	AULAS (classes)	-localizacao_building (FK p/ localizacao) - localizacao_sala (FK p/ localizacao) - disciplina_ed_id (FK p/ course_ed)	- data (PK) - start_time (PK) - duration - inscritos - tipo -localizacao_building (FK) - localizacao_sala (FK) - disciplina_ed_id (FK)
localizacao	LOCALIZACAO	(não possui FKs)	- building (PK) - sala (PK) - capacidade
course	COURSE	(não possui FKs)	- discipline_id (PK) - nome
degree	DEGREE	(não possui FKs)	- degree_id (PK)

Realização:

Teodoro Marques - 2023211717 - teomarques@gmail.com

Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

			- nome
course_ed	COURSE_ed	- course_id (FK p/ course) - coordenador_id (FK p/ instrutor)	- course_ed_id (PK) - ano - assistentes - course_id (FK) - coordenador_id (FK)
pessoa	PESSOA	(não possui FKs)	- pessoa_id (PK) - nome
aluno	Aluno (Alunos Matriculados)	- pessoa_id (FK p/ pessoa)	- pessoa_id (PK, FK) - media - divida_estudantil
instrutor	Instrutor (Coordenador)	- pessoa_id (FK p/ pessoa)	- pessoa_id (PK, FK)
assistentes	Assistentes (Assistentes de Edição)	- pessoa_id (FK p/ pessoa) - instrutor_id (FK p/ instrutor) - disciplina_ed_id (FK p/ course_ed)	- pessoa_id (PK, FK) - instrutor_id (FK) - disciplina_ed_id (FK)
extracurricular	Extracurricular (student_extracurriculares)	(não possui FKs)	- extracurricular_id (PK) - tipo - custo - nome

Realização:

Teodoro Marques - 2023211717 - teomarques@gmail.com

Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

grades_aluno	Historico_Academico (Grades_aluno)	- peessoa_id (FK p/ aluno)	- peessoa_id (PK, FK) - grade
degree_course	Relação DISCIPLINA ↔ Degree	- disciplina_id (FK p/ course) - degree_id (FK p/ degree)	- disciplina_id (PK, FK) - degree_id (PK, FK)
aluno_aulas	Relação Aluno ↔ AULAS	- peessoa_id (FK p/ aluno) - aula_data (FK p/ aulas) - aula_start_time (FK p/ aulas)	- peessoa_id (PK, FK) - aula_data (PK, FK) - aula_start_time (PK, FK)
aluno_course_ed	Relação Aluno ↔ DISCIPLINA_ed	- peessoa_id (FK p/ aluno) - disciplina_ed_id (FK p/ course_ed)	- peessoa_id (PK, FK) - disciplina_ed_id (PK, FK)
aluno_degree	Relação Aluno ↔ Degree	- peessoa_id (FK p/ aluno) - degree_id (FK p/ degree)	- peessoa_id (PK, FK) - degree_id (PK, FK)
aluno_extracurriculares	Relação Aluno ↔ Extracurricular	- peessoa_id (FK p/ aluno) - extracurricular_id (FK p/ extracurricular)	- peessoa_id (PK, FK) - extracurricular_id (PK, FK)

Realização:

Teodoro Marques - 2023211717 - teomarques@gmail.com

Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

Explicações das Relações no Modelo Físico

A seguir, detalhamos por que essas relações são definidas dessa forma:

1. **aulas** → **localizacao** e **course_ed**

- Cada aula é ministrada em um local específico do campus, por isso **aulas** possui FKs para **localizacao** (building e sala).
- Além disso, a aula está vinculada a uma edição de disciplina (**course_ed**), pois é necessário saber a qual turma/edição a aula pertence.

2. **localizacao**

- É uma tabela independente, pois cada localização (edifício e sala) é única e pode ser referenciada por várias aulas. Não possui FK, mas seus campos (building e sala) formam a PK.

3. **course** e **course_ed**

- **course** armazena informações sobre a disciplina em si (por exemplo, nome e um identificador único).
- **course_ed** representa uma edição específica dessa disciplina (ano, assistentes, coordenador etc.), tendo FK para **course** (pois uma edição pertence a apenas uma disciplina) e FK para **instrutor** (que atua como coordenador).

4. **pessoa**, **aluno** e **instrutor**

- **pessoa** é a entidade genérica que contém os dados básicos (ID, nome).
- **aluno** e **instrutor** são especializações que referenciam **pessoa** via FK. Isso permite que o mesmo registro em **pessoa** possa representar diferentes papéis (quando aplicável).

5. **assistentes**

- Tabela que relaciona uma pessoa (FK para **pessoa**) que atua como assistente a um instrutor responsável (FK para **instrutor**) e a edição de disciplina em que auxilia (FK para **course_ed**).

6. **extracurricular**

- Representa as atividades extracurriculares (esportes, clubes etc.). Não possui FK porque cada atividade é cadastrada independentemente.

7. **grades_aluno**

- Vincula o histórico de notas ao **aluno** (FK para a tabela **aluno**), armazenando a nota (grade) de cada disciplina cursada.

8. **degree_course**

Realização:

Teodoro Marques - 2023211717 - teomarques@gmail.com

Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

- Relaciona as disciplinas (**course**) a um curso de graduação (**degree**), mostrando quais disciplinas pertencem a qual grau (curso superior). Cada par (disciplina_id, degree_id) é único (PK composta).

9. aluno_aulas

- Relaciona o **aluno** (via pessoa_id) às **aulas** (via data e start_time), indicando em quais aulas o aluno está inscrito ou participa. A PK é composta por (pessoa_id, aula_data, aula_start_time).

10. aluno_course_ed

- Relaciona o **aluno** às edições de disciplina (**course_ed**). A PK é composta por (pessoa_id, disciplina_ed_id), representando a matrícula do aluno naquela edição.

11. aluno_degree

- Relaciona o **aluno** a um curso de graduação (**degree**). Isso possibilita que um aluno esteja associado a um ou mais cursos ao longo do tempo.

12. aluno_extracurriculares

- Relaciona o **aluno** às atividades extracurriculares. Assim, o sistema consegue registrar quais alunos estão em quais atividades, permitindo o controle de custos e participação.

Por que essas relações são necessárias

- **Integridade Referencial:** Cada FK garante que os dados em tabelas dependentes (como aulas, aluno_aulas, assistenes etc.) estejam sempre associados a registros válidos em tabelas de referência (por exemplo, localizacao, pessoa, course_ed).
 - **Normalização:** Separar as entidades (pessoa, aluno, instrutor) reduz redundâncias. A criação de tabelas de relacionamento (aluno_course_ed, aluno_extracurriculares, degree_course etc.) evita duplicação de dados e permite gerenciar relações muitos-para-muitos de forma consistente.
 - **Manutenção de Consistência:** Várias entidades requerem remoção em cascata (por exemplo, ao excluir um aluno, remover suas matrículas e histórico). As FKs e constraints de ON DELETE CASCADE ou triggers facilitam essa tarefa.
 - **Escalabilidade e Evolução:** Definir essas relações de forma clara facilita a evolução do modelo, permitindo adicionar novos campos e relacionamentos sem comprometer a integridade dos dados.
-

Realização:

Teodoro Marques - 2023211717 - teomarques@gmail.com

Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

Transações

As transações garantem que operações críticas no banco de dados sejam executadas de forma **atômica** (tudo ou nada). Isso é essencial para manter a consistência dos dados, especialmente em sistemas universitários onde múltiplas operações dependem umas das outras.

Exemplo Prático: Matrícula em uma Edição de Curso

Suponha que um aluno deseja se matricular em uma edição de curso (*course_ed*). Essa ação envolve:

1. **Verificar vagas disponíveis** na edição.
2. **Inserir o aluno** na tabela de matrículas (*course_ed_aluno*).
3. **Atualizar o número de inscritos** na tabela *AULAS*.

Se qualquer etapa falhar (ex.: vagas esgotadas), a transação deve ser **cancelada** (ROLLBACK). Caso contrário, todas as alterações são **confirmadas** (COMMIT).

Exemplo para gestão das transações:

sql

Copy

```
BEGIN TRANSACTION;
```

-- selecionar inscritos em uma edição

```
SELECT inscritos FROM aulas WHERE course_ed_ano = 2024;
```

-- se houver vagas, cadastrar aluno

```
INSERT INTO course_ed_aluno (course_ed_ano, aluno_id) VALUES (2024, 123);
```

-- dar update na quantidade de alunos na edição

```
UPDATE aulas SET inscritos = inscritos + 1 WHERE course_ed_ano = 2024;
```

```
COMMIT; -- Ou ROLLBACK em caso de erro
```

Concorrência

A concorrência ocorre quando múltiplos usuários acessam ou modificam os mesmos dados simultaneamente. Se a gestão for mal feita, podem ter conflitos. Por exemplo:

- **Leitura suja:** Ler dados não confirmados.
- **Atualizações perdidas:** Duas transações sobrescrevem uma à outra.

Uma estratégia para evitar os conflitos são os **bloqueios**:

Realização:

Teodoro Marques - 2023211717 - teomarques@gmail.com

Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

- Bloqueia um registro durante uma transação (ex.: *SELECT ... FOR UPDATE*).
- Exemplo: Bloquear a edição de curso durante uma matrícula para evitar duplicidade.

Cenário de Conflito:

Dois administradores tentam atualizar o número de inscritos na mesma aula.

- **Solução:** Usar transações curtas e bloquear a tabela temporariamente.
-

Refinamento Técnico: Chaves Primárias Compostas

- **Por que a tabela *AULAS* tem uma PK com 7 atributos?**

A chave primária composta (*data*, *start_time*, *localizacao_building*, *localizacao_sala*, *course_ed_ano*, *course_ed_course_id*, *course_ed_instrutor_pessoa_id*) garante que:

- Não haja aulas duplicadas no mesmo local, horário e edição de curso.
- Cada aula seja única em seu contexto acadêmico.

Problemas Potenciais:

- **Complexidade:** Consultas podem ficar lentas devido ao tamanho da PK.
- **Manutenção:** Alterar a estrutura da PK afeta todas as tabelas relacionadas.

Alternativa Mais Eficiente

Usar um **ID único autoincrementado** como PK:

```
sql
Copy
CREATE TABLE aulas (
  aula_id SERIAL PRIMARY KEY, -- ID único
  data DATE,
  start_time TIME,
  ... -- atributos restantes
);
```

Vantagens:

- Simplifica consultas e relacionamentos.
- Reduz o tamanho das FKs em tabelas associativas.

Quando Usar PK Composta?

Realização:

Teodoro Marques - 2023211717 - teomarques@gmail.com

Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

- Quando a unicidade depende naturalmente de múltiplos atributos (ex.: horário + local).
- Evitar quando a eficiência é crítica.

Data Limite: 22 de maio de 2025

Tabela de Tarefas

Tarefa	Descrição	Deadline
1. Definição de Escopo	Revisar requisitos funcionais e identificar casos de uso críticos.	20/02/2025 
2. Modelagem Conceitual (ER)	Criar diagrama ER no ONDA, definindo entidades, atributos e relações.	25/02/2025 
3. Modelagem Física	Converter ER em tabelas, definir PKs, FKs e constraints no ONDA.	20/03/2025 
4. Configuração do PostgreSQL	Criar scripts SQL para tabelas, roles, triggers e funções PL/pgSQL.	10/03/2025
5. Desenvolvimento da REST API	Implementar endpoints básicos (GET, POST, PUT, DELETE) em Python.	15/04/2025
6. Integração API-Banco de Dados	Conectar API ao PostgreSQL usando psycopg3 e validar operações CRUD.	20/04/2025
7. Implementação de Transações	Garantir atomicidade em operações críticas (ex: matrículas, remoção de alunos).	25/04/2025

Realização:
Teodoro Marques - 2023211717 - teomarques@gmail.com
Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com

8. Gestão de Concorrência	Definir estratégias (ex: bloqueios, versionamento) para evitar conflitos.	30/04/2025
9. Testes Funcionais	Testar todos os endpoints, validar fluxos e corrigir bugs.	05/05/2025
10. Redação do Relatório	Incluir seções: transações, concorrência, modelos ER/físico, desenvolvimento.	10/05/2025
11. Revisão Final	Validar consistência do código, dados e documentação.	15/05/2025
12. Submissão no Inforestudante	Upload do relatório, modelos ONDA, JSON, e scripts SQL.	22/05/2025

Conclusão

O modelo físico aqui descrito implementa, de forma fiel, o modelo conceitual proposto, garantindo integridade referencial, consistência e clareza no gerenciamento de alunos, cursos, disciplinas, localizações e o restante das entidades acadêmicas. As tabelas e suas foreign keys refletem as necessidades de um sistema universitário, onde a rastreabilidade de informações (matrículas, aulas, notas, atividades, dívidas estudantis etc.) é fundamental para a operação confiável da aplicação.

Realização:

Teodoro Marques - 2023211717 - teomarques@gmail.com

Matheus Coutinho D'Oliveira - 2023105804 - matheuscoutinho20@hotmail.com