

**UFVJM – UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI**  
**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**  
**BANCO DE DADOS I**

**TRABALHO DE BANCO DE DADOS I**  
**ESPECIFICAÇÃO CONCEITUAL, LÓGICA E IMPLEMENTAÇÃO DE UMA BASE**  
**DE DADOS RELACIONAL**

**Alunos:**

Ana Clara Rolim de Azevedo  
Gabriel de Jesus Moraes Maurício  
Janaína Alves Cordeiro  
Nádson Nascimento Santos  
Riquelme Moreira Campos

## SUMÁRIO

<b>1 – APRESENTAÇÃO DO PROBLEMA.....</b>	<b>3</b>
<b>2 – MODELAGEM CONCEITUAL.....</b>	<b>3</b>
<b>3 – MODELO LÓGICO DA BASE DE DADOS.....</b>	<b>4</b>
<b>4 – NORMALIZAÇÃO DO MODELO LÓGICO.....</b>	<b>6</b>
4.1- 1ª FORMA NORMAL (1FN).....	7
4.2- 1ª FORMA NORMAL (2FN).....	8
4.3- 3ª FORMA NORMAL (3FN).....	10
<b>5 – SCRIPTS DE IMPLEMENTAÇÃO DA BASE DE DADOS.....</b>	<b>10</b>
5.1 – COMANDOS CREATE TABLE.....	10
5.2 – COMANDOS INSERT.....	12
5.3 – COMANDOS SELECT.....	18
<b>6 – CONCLUSÃO.....</b>	<b>19</b>
<b>7 – REFERÊNCIAS.....</b>	<b>20</b>

## 1 – APRESENTAÇÃO DO PROBLEMA

### Escola de Idiomas

A escola de idiomas 'Global Language Center' oferece cursos de diferentes idiomas e níveis, visando atender alunos de diversas faixas etárias e perfis. Cada curso é identificado por um código único e possui um idioma específico (como inglês, espanhol, francês e italiano), além do nível (básico, intermediário, avançado), carga horária total e preço.

Os alunos matriculados na escola têm um cadastro que inclui nome, data de nascimento (AAAA-MM-DD), endereço (rua, bairro, número e cidade), email (podendo haver mais de um) e telefone (aceitando múltiplos números). Cada aluno possui um número de registro único. Além disso, cada aluno pode estar matriculado em um ou mais cursos, sendo que cada matrícula armazena a data de início e término do curso.

A escola conta com um quadro de professores especializados em diferentes idiomas. Sobre cada professor, são registrados nome, especialidade (idiomas que leciona), e disponibilidade (dias e horários).

As aulas são agendadas de acordo com o cronograma de cada curso, contendo data, hora, professor responsável e o conteúdo que será abordado. Cada aula está vinculada a um curso específico com uma determinada carga horária e pode ser lecionada por um ou mais professores, dependendo da necessidade da escola.

A base de dados do negócio deve prover informações para o controle de alunos matriculados, cursos oferecidos, professores disponíveis, aulas agendadas e histórico de matrículas, além de possibilitar a emissão de relatórios financeiros relacionados ao pagamento dos cursos e controle de inadimplência.

## 2 – MODELAGEM CONCEITUAL

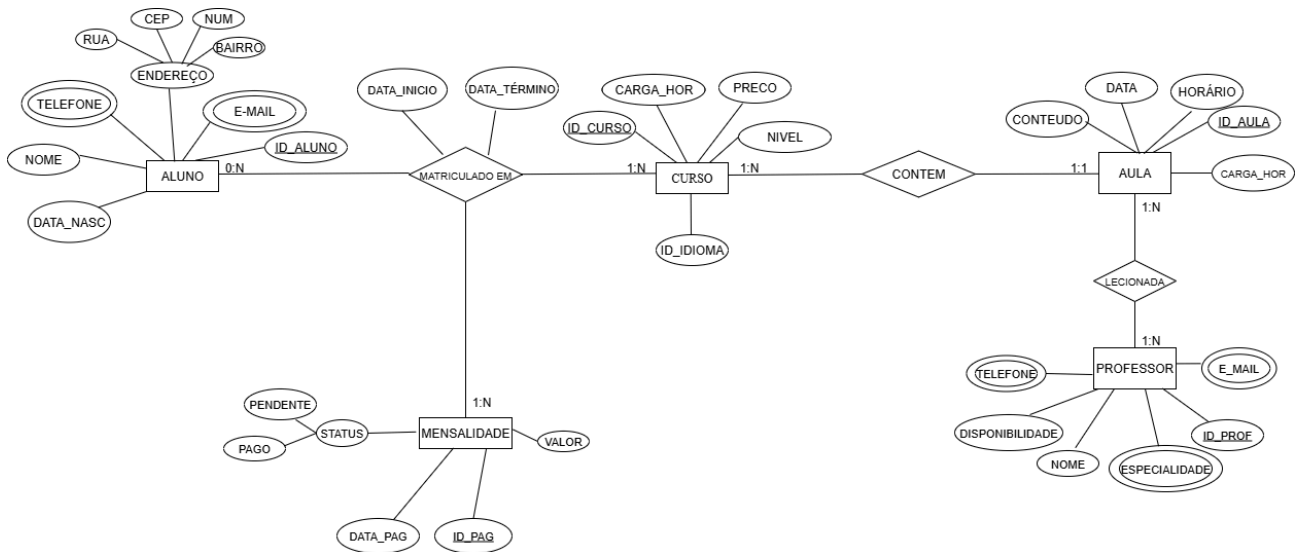
O diagrama de modelagem conceitual apresentado representa a estrutura de dados de uma escola de idiomas, detalhando as entidades, atributos e os principais relacionamentos entre elas. A entidade Aluno possui atributos como identificador (ID\_Aluno), nome, data de nascimento, e um endereço composto (rua, bairro, número, CEP), além de permitir o cadastro de múltiplos telefones e e-mails para cada aluno. Alunos podem se matricular em diferentes cursos através do relacionamento Matriculado em, que armazena informações como data de início e término da matrícula.

A entidade Curso é identificada por um código (ID\_Curso) e está associada a um idioma específico, nível (básico, intermediário, avançado), carga horária total e preço. Cada curso pode ter diversas aulas, detalhadas na entidade Aula, que armazena dados como identificador (ID\_Aula), data, horário, conteúdo e carga horária. As aulas são ministradas por professores, representados na entidade Professor, que contém dados como identificador (ID\_Prof), nome, especialidade (idiomas que leciona), disponibilidade, além de telefones e e-mails.

O sistema de pagamentos é modelado pela entidade Mensalidade, relacionada à matrícula do aluno, registrando informações como identificador (ID\_Pag), valor, data de pagamento e status (pago ou pendente).

Os relacionamentos presentes no diagrama demonstram que um aluno pode estar matriculado em vários cursos, cada curso pode ter várias aulas e um professor pode lecionar diversas aulas. Dessa forma, o modelo captura todas as informações essenciais para o gerenciamento acadêmico, administrativo e financeiro da

escola de idiomas, permitindo um controle integrado dos dados de alunos, cursos, professores, aulas e pagamentos.



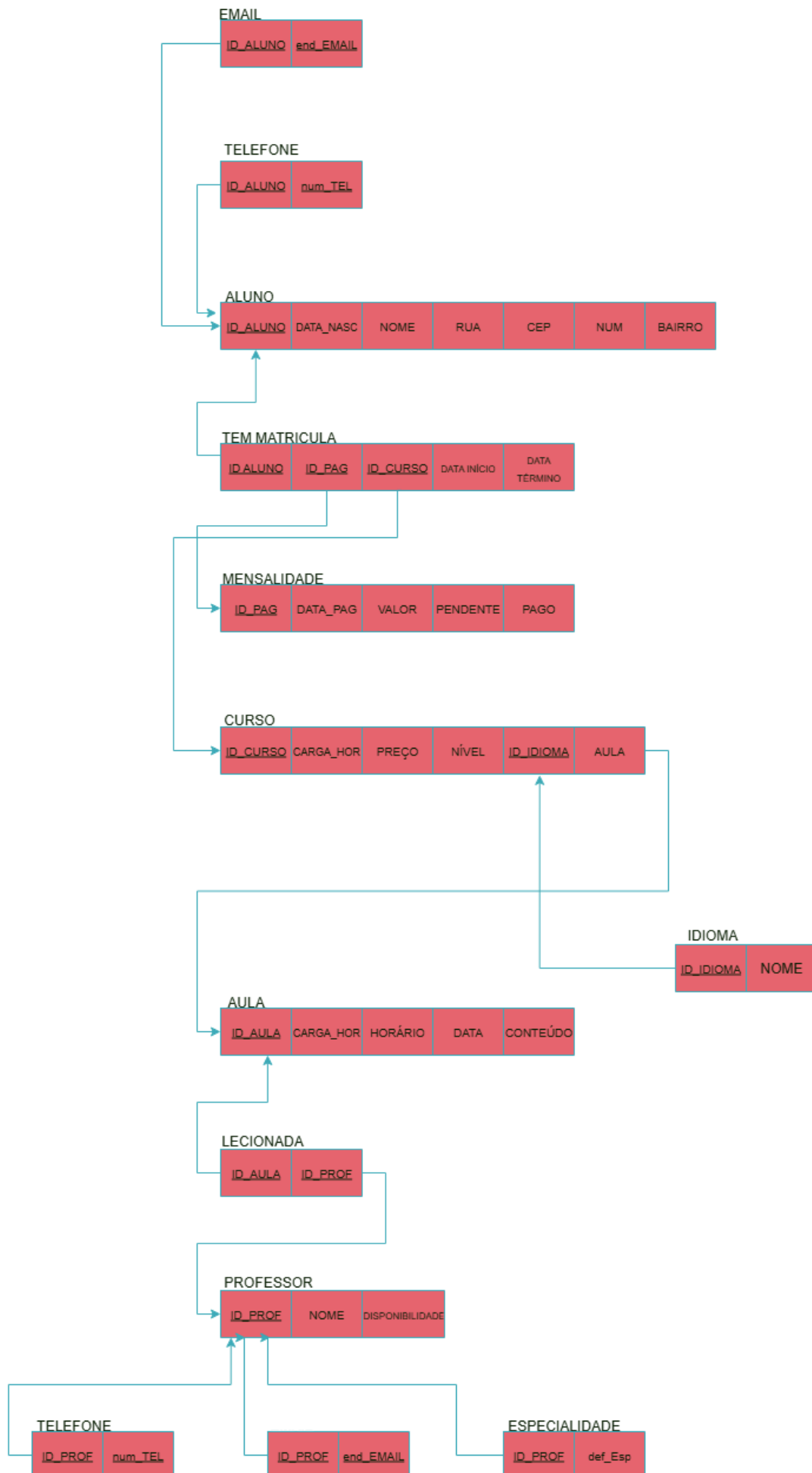
### 3 – MODELO LÓGICO DA BASE DE DADOS

A modelagem lógica é a etapa intermediária do projeto de um banco de dados, responsável por transformar o modelo conceitual em uma representação mais próxima do ambiente de banco de dados relacional, porém ainda independente do Sistema Gerenciador de Banco de Dados (SGBD) utilizado. Neste momento, as entidades, atributos e relacionamentos definidos na modelagem conceitual são convertidos em tabelas, colunas e chaves, respeitando as regras do modelo relacional.

Durante a modelagem lógica, são especificadas as chaves primárias, chaves estrangeiras, tipos de dados, restrições de integridade e cardinalidades, sem considerar particularidades de implementação específicas de um SGBD. O modelo lógico detalha como os dados serão estruturados e relacionados entre si, servindo como base para a próxima etapa, a modelagem física.

Essa fase é fundamental para garantir que o banco de dados seja eficiente, íntegro e atenda aos requisitos identificados na modelagem conceitual, promovendo a padronização e facilitando a futura manutenção do sistema.

A seguir, apresenta-se o modelo lógico da base de dados desenvolvida para a escola de idiomas *Global Language Center*. Este modelo traduz a estrutura conceitual previamente definida em um formato mais próximo do que será implementado no sistema de gerenciamento de banco de dados, detalhando as tabelas, atributos, chaves primárias e estrangeiras, além dos relacionamentos entre as entidades.



## 4 – NORMALIZAÇÃO DO MODELO LÓGICO

A normalização de banco de dados é um processo essencial no projeto lógico de sistemas que utilizam o modelo relacional. Seu objetivo é organizar os dados de modo a minimizar redundâncias, evitar inconsistências e facilitar a manutenção da integridade dos dados. Através da aplicação de regras formais denominadas formas normais, o processo de normalização orienta a decomposição de relações complexas em estruturas mais simples e coerentes.

A normalização surgiu a partir dos estudos de Edgar F. Codd, criador do modelo relacional, e é aplicada em diferentes níveis chamados de formas normais (1ª Forma Normal, 2ª Forma Normal, 3ª Forma Normal, entre outras). Cada forma normal impõe restrições específicas que devem ser atendidas para garantir um nível adequado de organização dos dados.

A **Primeira Forma Normal (1FN)** estabelece que todos os atributos de uma relação devem possuir valores atômicos, ou seja, indivisíveis. Essa forma normal elimina grupos repetitivos e atributos multivalorados. Por exemplo, uma tabela que armazena múltiplos e-mails ou telefones em uma única célula viola a 1FN, pois contém conjuntos de dados que não são atômicos.

A **Segunda Forma Normal (2FN)** exige que a relação esteja em 1FN e que todos os atributos não-chave sejam dependentes funcionalmente da chave primária inteira. Isso significa que, em tabelas com chaves compostas, não deve haver atributos que dependam apenas de parte da chave. A 2FN trata, portanto, de eliminar dependências parciais, reestruturando os dados para que cada atributo descreva exclusivamente a entidade representada pela chave primária completa.

A **Terceira Forma Normal (3FN)** requer que a relação esteja na 2FN e, adicionalmente, que não existam dependências transitivas entre atributos não-chave. Ou seja, um atributo não-chave não deve depender de outro atributo que também não é chave. A 3FN visa garantir que cada atributo contenha apenas informações diretamente relacionadas à chave primária da tabela.

A aplicação das três primeiras formas normais é geralmente suficiente para garantir uma estrutura de dados eficiente e livre de redundâncias. No entanto, em ambientes em que o desempenho é uma prioridade, como em sistemas de leitura intensiva ou data warehouses, pode ser necessário realizar a **desnormalização**. Esse processo consiste em reintroduzir alguma redundância controlada para evitar junções excessivas e melhorar o desempenho de determinadas consultas, sempre com cautela para não comprometer a integridade dos dados.

Em suma, a normalização é uma prática indispensável para garantir a qualidade, consistência e eficiência dos dados em sistemas relacionais. Ao aplicar as formas normais corretamente, obtém-se um modelo de dados mais limpo, modular e preparado para futuras evoluções do sistema.

A seguir, apresenta-se o processo de normalização aplicado à base de dados da escola de idiomas Global Language Center. A normalização consiste em uma série de etapas destinadas a organizar as estruturas do banco de dados, eliminando redundâncias, minimizando inconsistências e promovendo a integridade das informações. Por meio da aplicação das formas normais, as tabelas são ajustadas para garantir uma modelagem eficiente, facilitando a manutenção e o uso seguro dos dados ao longo do tempo.

#### 4.1- 1ª FORMA NORMAL (1FN)

A 1FN exige que todos os atributos das tabelas contenham apenas valores atômicos, ou seja, indivisíveis. Também impede a existência de grupos repetitivos ou multivalorados. Para atingir a 1FN, é necessário criar tabelas separadas para atributos que podem ocorrer múltiplas vezes, como telefones ou e-mails.

##### **Tabela Aluno**

<b>Campo</b>	<b>Tipo</b>
ID_ALUNO (PK)	Inteiro
NOME	Texto
DATA_NASC	Data
RUA	Texto
NUM	Texto
BAIRRO	Texto
CEP	Texto

##### **Tabela Email\_Aluno**

<b>Campo</b>	<b>Tipo</b>
ID_ALUNO (FK)	Inteiro
END_EMAIL	Texto

##### **Tabela Telefone\_Aluno**

<b>Campo</b>	<b>Tipo</b>
ID_ALUNO (FK)	Inteiro
NUM_TEL	Texto

##### **Tabela Professor**

<b>Campo</b>	<b>Tipo</b>
ID_PROF (PK)	Inteiro
NOME	Texto

DISPONIBILIDADE	Texto
-----------------	-------

### Tabela Email\_Prof

Campo	Tipo
ID_PROF (FK)	Inteiro
END_EMAIL	Texto

### Tabela Telefone\_Prof

Campo	Tipo
ID_PROF (FK)	Inteiro
NUM_TEL	Texto

### Tabela Especialidade\_Prof

Campo	Tipo
ID_PROF (FK)	Inteiro
DESC_ESP	Texto

## 4.2- 1ª FORMA NORMAL (2FN)

A 2FN exige que a tabela esteja na 1FN e que nenhum atributo não-chave dependa apenas de uma parte da chave primária (no caso de chaves compostas). Aqui, eliminamos dependências parciais, ou seja, quando um atributo depende de parte da chave composta.

### Tabela Matrícula

Campo	Tipo
ID_MATRICULA (PK)	Inteiro
ID_ALUNO (FK)	Inteiro
ID_CURSO (FK)	Inteiro
DATA_INICIO	Data
DATA_TERMINO	Data



**Tabela Curso**

<b>Campo</b>	<b>Tipo</b>
ID_CURSO (PK)	Inteiro
NIVEL	Texto
CARGA_HOR	Inteiro
PREÇO	Decimal

**Tabela Aula**

<b>Campo</b>	<b>Tipo</b>
ID_ALUNO (PK)	Inteiro
ID_CURSO (FK)	Inteiro
CARGA_HOR	Inteiro
HORÁRIO	Texto
DATA	Data
CONTEÚDO	Texto

**Tabela Leciona**

<b>Campo</b>	<b>Tipo</b>
ID_PROF (FK)	Inteiro
ID_AULA (FK)	Inteiro

#### 4.3- 3ª FORMA NORMAL (3FN)

A 3FN exige que a tabela esteja na 2FN e que nenhum atributo não-chave dependa de outro atributo não-chave. Em outras palavras, todo atributo não-chave deve depender somente da chave primária, eliminando assim dependências transitivas.

#### Tabela Pagamento

Campo	Tipo
ID_PAG (PK)	Inteiro
ID_MATRÍCULA (FK)	Inteiro
DATA_PAG	Data
VALOR	Decimal
PENDENTE	Boolean
PAGO	Boolean

### 5 – SCRIPTS DE IMPLEMENTAÇÃO DA BASE DE DADOS

Nesta seção deverão ser inseridos os scripts necessários para a criação e manipulação da base de dados, divididos em 3 partes:

#### 5.1 – COMANDOS CREATE TABLE

-- Tabela ALUNO

```
CREATE TABLE ALUNO (
  ID_ALUNO INT PRIMARY KEY,
  DATA_NASC DATE NOT NULL,
  NOME VARCHAR(100) NOT NULL,
  RUA VARCHAR(100) NOT NULL,
  CEP VARCHAR(10) NOT NULL,
  NUM INT NOT NULL,
  BAIRRO VARCHAR(50) NOT NULL
);
```

-- Tabela EMAIL do aluno

```
CREATE TABLE EMAIL (
  ID_ALUNO INT NOT NULL,
  END_EMAIL VARCHAR(100) NOT NULL,
  FOREIGN KEY (ID_ALUNO) REFERENCES ALUNO(ID_ALUNO)
);
```

```

-- Tabela TELEFONE do aluno
CREATE TABLE TELEFONE (
    ID_ALUNO INT NOT NULL,
    NUM_TEL VARCHAR(20) NOT NULL,
    FOREIGN KEY (ID_ALUNO) REFERENCES ALUNO(ID_ALUNO)
);

-- Tabela IDIOMA
CREATE TABLE IDIOMA (
    ID_IDIOMA INT PRIMARY KEY,
    NOME VARCHAR(50) NOT NULL
);

-- Tabela CURSO
CREATE TABLE CURSO (
    ID_CURSO INT PRIMARY KEY,
    ID_IDIOMA INT NOT NULL,
    NIVEL VARCHAR(20) NOT NULL,
    CARGA_HOR INT NOT NULL,
    PRECO DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (ID_IDIOMA) REFERENCES IDIOMA(ID_IDIOMA)
);

-- Tabela AULA
CREATE TABLE AULA (
    ID_AULA INT PRIMARY KEY,
    CARGA_HOR INT NOT NULL,
    HORARIO TIME NOT NULL,
    DATA DATE NOT NULL,
    CONTEUDO TEXT NOT NULL
);

-- Tabela PROFESSOR
CREATE TABLE PROFESSOR (
    ID_PROF INT PRIMARY KEY,
    NOME VARCHAR(100) NOT NULL,
    DISPONIBILIDADE VARCHAR(100) NOT NULL
);

-- Tabela EMAIL_PROF
CREATE TABLE EMAIL_PROF (
    ID_PROF INT NOT NULL,
    END_EMAIL VARCHAR(100) NOT NULL,
    FOREIGN KEY (ID_PROF) REFERENCES PROFESSOR(ID_PROF)
);

-- Tabela TELEFONE_PROF
CREATE TABLE TELEFONE_PROF (
    ID_PROF INT NOT NULL,
    NUM_TEL VARCHAR(20) NOT NULL,
    FOREIGN KEY (ID_PROF) REFERENCES PROFESSOR(ID_PROF)
);

```

```
-- Tabela ESPECIALIDADE
CREATE TABLE ESPECIALIDADE (
    ID_PROF INT NOT NULL,
    DEF_ESP VARCHAR(100) NOT NULL,
    FOREIGN KEY (ID_PROF) REFERENCES PROFESSOR(ID_PROF)
);

-- Tabela LECIONADA
CREATE TABLE LECIONADA (
    ID_AULA INT NOT NULL,
    ID_PROF INT NOT NULL,
    PRIMARY KEY (ID_AULA, ID_PROF),
    FOREIGN KEY (ID_AULA) REFERENCES AULA(ID_AULA),
    FOREIGN KEY (ID_PROF) REFERENCES PROFESSOR(ID_PROF)
);

-- Tabela MENSALIDADE
CREATE TABLE MENSALIDADE (
    ID_PAG INT PRIMARY KEY,
    DATA_PAG DATE NOT NULL,
    VALOR DECIMAL(10,2) NOT NULL,
    PENDENTE BOOLEAN NOT NULL,
    PAGO BOOLEAN NOT NULL
);

-- Tabela TEM_MATRICULA
CREATE TABLE TEM_MATRICULA (
    ID_ALUNO INT NOT NULL,
    ID_PAG INT NOT NULL,
    ID_CURSO INT NOT NULL,
    DATA_INICIO DATE NOT NULL,
    DATA_TERMINO DATE NOT NULL,
    PRIMARY KEY (ID_ALUNO, ID_CURSO),
    FOREIGN KEY (ID_ALUNO) REFERENCES ALUNO(ID_ALUNO),
    FOREIGN KEY (ID_PAG) REFERENCES MENSALIDADE(ID_PAG),
    FOREIGN KEY (ID_CURSO) REFERENCES CURSO(ID_CURSO)
);
```

## 5.2 – COMANDOS INSERT

A seguir, são apresentados os comandos SQL do tipo INSERT utilizados para povoar o banco de dados da escola de idiomas **Global Language Center** com dados fictícios. Esses dados simulam registros reais de alunos, seus endereços, emails e telefones, permitindo realizar testes e consultas no sistema de forma realista.

Os comandos foram organizados por tabela, respeitando as chaves primárias e estrangeiras previamente definidas no modelo relacional.

```
-- Inserção dos idiomas
INSERT INTO IDIOMA (ID_IDIOMA, NOME) VALUES (1, 'Inglês');
INSERT INTO IDIOMA (ID_IDIOMA, NOME) VALUES (2, 'Francês');
INSERT INTO IDIOMA (ID_IDIOMA, NOME) VALUES (3, 'Espanhol');
```

-- Inserção dos cursos

```
INSERT INTO CURSO (ID_CURSO, ID_IDIOMA, NIVEL, CARGA_HOR, PRECO)
VALUES (1, 1, 'Básico', 60, 1200.00);
INSERT INTO CURSO (ID_CURSO, ID_IDIOMA, NIVEL, CARGA_HOR, PRECO)
VALUES (2, 1, 'Intermediário', 60, 1200.00);
INSERT INTO CURSO (ID_CURSO, ID_IDIOMA, NIVEL, CARGA_HOR, PRECO)
VALUES (3, 1, 'Avançado', 60, 1200.00);
INSERT INTO CURSO (ID_CURSO, ID_IDIOMA, NIVEL, CARGA_HOR, PRECO)
VALUES (4, 2, 'Básico', 60, 1200.00);
INSERT INTO CURSO (ID_CURSO, ID_IDIOMA, NIVEL, CARGA_HOR, PRECO)
VALUES (5, 2, 'Intermediário', 60, 1200.00);
INSERT INTO CURSO (ID_CURSO, ID_IDIOMA, NIVEL, CARGA_HOR, PRECO)
VALUES (6, 2, 'Avançado', 60, 1200.00);
INSERT INTO CURSO (ID_CURSO, ID_IDIOMA, NIVEL, CARGA_HOR, PRECO)
VALUES (7, 3, 'Básico', 60, 1200.00);
INSERT INTO CURSO (ID_CURSO, ID_IDIOMA, NIVEL, CARGA_HOR, PRECO)
VALUES (8, 3, 'Intermediário', 60, 1200.00);
INSERT INTO CURSO (ID_CURSO, ID_IDIOMA, NIVEL, CARGA_HOR, PRECO)
VALUES (9, 3, 'Avançado', 60, 1200.00);
```

-- Inserção dos alunos

```
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (1, '1974-10-15', 'Ana Clara Rolim de Azevedo', 'Trevo Silveira, Diamantina', '04548542', 271, 'Parque São José');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (2, '2008-08-15', 'Gabriel de Jesus Moraes Maurício', 'Ladeira de Viana, Santa Maria do Suaçuí', '82680928', 211, 'Havaí');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (3, '1995-09-01', 'Janaina Alves Cordeiro', 'Quadra de Silveira, Capelinha', '43991-648', 301, 'Vila Satélite');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (4, '1979-10-21', 'Nádson Nascimento Santos', 'Distrito Ana Júlia Cardoso, Guanhões', '19391-443', 213, 'Cidade Nova');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (5, '1980-01-08', 'Riquelme Moreira Campos', 'Parque Pinto, Sabinópolis', '15561093', 179, 'Maria Helena');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (6, '1989-12-27', 'Srta. Marina das Neves', 'Vereda da Paz, Peçanha', '49129723', 148, 'João Pinheiro');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (7, '1975-10-10', 'Dra. Helena Ferreira', 'Campo de da Mata, Itamarandiba', '87476-647', 386, 'Vila Da Ária');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (8, '2001-12-01', 'João Gabriel Novaes', 'Quadra Luiz Otávio Freitas, Rio Vermelho', '33773-096', 215, 'Canaa');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (9, '1977-12-24', 'Srta. Beatriz Teixeira', 'Lago de Santos, Diamantina', '94611238', 25, 'Estoril');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (10, '1980-01-27', 'Pietra Lima', 'Sítio Eloah das Neves, Santa Maria do Suaçuí', '07212-798', 99, 'Vila Primeiro De Maio');

INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
```

```

VALUES (11, '1978-02-25', 'Levi Peixoto', 'Jardim Bruno Silva, Capelinha', '30033-554', 400, 'Conjunto
Bonsucesso');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (12, '2008-03-01', 'Dr. Heitor Costela', 'Distrito Guilherme Correia, Guanhães', '19554589', 348,
'Vila União');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (13, '1989-07-30', 'Emilly Fernandes', 'Largo Maria Fernanda Farias, Sabinópolis', '10407-689', 10,
'Padre Eustáquio');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (14, '1989-02-24', 'Vicente Ferreira', 'Pátio Maria Sophia Alves, Peçanha', '94586344', 263,
'Conjunto Jardim Filadélfia');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (15, '1991-12-03', 'Bruno Cunha', 'Rua Fernando da Rosa, Itamarandiba', '17122884', 367,
'Aparecida 7ª Seção');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (16, '1983-05-05', 'Maria Luiza Alves', 'Jardim de Farias, Rio Vermelho', '51316152', 240, 'Nova
Esperança');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (17, '1975-09-17', 'Lucas Novaes', 'Núcleo Francisco Nunes, Diamantina', '00812-644', 41, 'Nossa
Senhora Da Aparecida');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (18, '2009-06-08', 'Cecília da Paz', 'Jardim Kevin Duarte, Santa Maria do Suaçuí', '93123-899',
282, 'Nossa Senhora Aparecida');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (19, '2009-11-22', 'Sra. Laís Ferreira', 'Campo Thomas Pinto, Capelinha', '42405183', 3,
'Independência');
INSERT INTO ALUNO (ID_ALUNO, DATA_NASC, NOME, RUA, CEP, NUM, BAIRRO)
VALUES (20, '1994-09-22', 'Maria Cardoso', 'Vila da Paz, Guanhães', '36712-306', 141, 'Padre Eustáquio');

```

-- Inserção das mensalidades

```

INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (1, '2025-01-16', 300.0, FALSE, TRUE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (2, '2025-04-16', 300.0, FALSE, TRUE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (3, '2025-01-09', 300.0, FALSE, TRUE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (4, '2025-04-03', 400.0, TRUE, FALSE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (5, '2025-02-05', 300.0, TRUE, FALSE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (6, '2025-04-16', 400.0, FALSE, TRUE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (7, '2025-04-14', 500.0, TRUE, FALSE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (8, '2025-04-11', 400.0, FALSE, TRUE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (9, '2025-05-03', 300.0, TRUE, FALSE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (10, '2025-06-30', 400.0, TRUE, FALSE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (11, '2025-04-30', 400.0, FALSE, TRUE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)

```

```

VALUES (12, '2025-01-21', 500.0, FALSE, TRUE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (13, '2025-01-13', 500.0, TRUE, FALSE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (14, '2025-02-12', 300.0, TRUE, FALSE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (15, '2025-01-11', 500.0, TRUE, FALSE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (16, '2025-03-29', 400.0, TRUE, FALSE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (17, '2025-02-03', 500.0, FALSE, TRUE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (18, '2025-03-09', 500.0, TRUE, FALSE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (19, '2025-03-13', 400.0, TRUE, FALSE);
INSERT INTO MENSALIDADE (ID_PAG, DATA_PAG, VALOR, PENDENTE, PAGO)
VALUES (20, '2025-04-11', 400.0, FALSE, TRUE);

```

-- Inserção das matrículas

```

INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (1, 1, 4, '2025-07-12', '2025-10-10');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (2, 2, 3, '2025-07-15', '2025-09-13');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (3, 3, 8, '2025-07-18', '2025-10-01');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (4, 4, 5, '2025-07-21', '2025-09-19');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (5, 5, 8, '2025-07-24', '2025-10-07');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (6, 6, 6, '2025-07-27', '2025-10-10');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (7, 7, 8, '2025-07-30', '2025-09-28');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (8, 8, 9, '2025-08-02', '2025-10-31');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (9, 9, 5, '2025-08-05', '2025-10-19');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (10, 10, 8, '2025-08-08', '2025-10-07');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (11, 11, 3, '2025-08-11', '2025-11-09');

```

```

INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (12, 12, 8, '2025-08-14', '2025-11-12');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (13, 13, 5, '2025-08-17', '2025-11-15');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (14, 14, 1, '2025-08-20', '2025-11-18');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (15, 15, 6, '2025-08-23', '2025-11-21');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (16, 16, 7, '2025-08-26', '2025-11-24');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (17, 17, 4, '2025-08-29', '2025-11-12');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (18, 18, 8, '2025-09-01', '2025-11-30');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (19, 19, 8, '2025-09-04', '2025-11-03');
INSERT INTO TEM_MATRICULA (ID_ALUNO, ID_PAG, ID_CURSO, DATA_INICIO,
DATA_TERMINO)
VALUES (20, 20, 2, '2025-09-07', '2025-11-06');

```

-- Inserção dos professores

```

INSERT INTO PROFESSOR (ID_PROF, NOME, DISPONIBILIDADE)
VALUES (1, 'Igor da Mata', 'Seg-Qua - 13h-17h');
INSERT INTO PROFESSOR (ID_PROF, NOME, DISPONIBILIDADE)
VALUES (2, 'Juan Almeida', 'Seg-Qua - 13h-17h');
INSERT INTO PROFESSOR (ID_PROF, NOME, DISPONIBILIDADE)
VALUES (3, 'João Moreira', 'Seg-Sex - 13h-17h');
INSERT INTO PROFESSOR (ID_PROF, NOME, DISPONIBILIDADE)
VALUES (4, 'Heloísa Nascimento', 'Seg-Qua - 13h-17h');
INSERT INTO PROFESSOR (ID_PROF, NOME, DISPONIBILIDADE)
VALUES (5, 'João Miguel Caldeira', 'Seg-Qua - 13h-17h');
INSERT INTO PROFESSOR (ID_PROF, NOME, DISPONIBILIDADE)
VALUES (6, 'Manuela Novaes', 'Seg-Sex - 8h-12h');
INSERT INTO PROFESSOR (ID_PROF, NOME, DISPONIBILIDADE)
VALUES (7, 'Laís Moreira', 'Seg-Sex - 13h-17h');
INSERT INTO PROFESSOR (ID_PROF, NOME, DISPONIBILIDADE)
VALUES (8, 'Kamilly Barbosa', 'Seg-Sex - 8h-12h');
INSERT INTO PROFESSOR (ID_PROF, NOME, DISPONIBILIDADE)
VALUES (9, 'Alexia Azevedo', 'Seg-Sex - 13h-17h');
INSERT INTO PROFESSOR (ID_PROF, NOME, DISPONIBILIDADE)
VALUES (10, 'Marcelo da Costa', 'Seg-Sex - 13h-17h');

```

-- Inserção dos emails, telefones e especialidades dos professores



```

INSERT INTO EMAIL_PROF (ID_PROF, END_EMAIL) VALUES (1, 'renandas-neves@ig.com.br');
INSERT INTO EMAIL_PROF (ID_PROF, END_EMAIL) VALUES (2, 'fpereira@ig.com.br');
INSERT INTO EMAIL_PROF (ID_PROF, END_EMAIL) VALUES (3, 'jpereira@gmail.com');
INSERT INTO EMAIL_PROF (ID_PROF, END_EMAIL) VALUES (4, 'ceciliaazevedo@yahoo.com.br');
INSERT INTO EMAIL_PROF (ID_PROF, END_EMAIL) VALUES (5, 'emanuely79@monteiro.com');
INSERT INTO EMAIL_PROF (ID_PROF, END_EMAIL) VALUES (6, 'qjesus@yahoo.com.br');
INSERT INTO EMAIL_PROF (ID_PROF, END_EMAIL) VALUES (7, 'mariada-mota@viana.com');
INSERT INTO EMAIL_PROF (ID_PROF, END_EMAIL) VALUES (8, 'clara19@uol.com.br');
INSERT INTO EMAIL_PROF (ID_PROF, END_EMAIL) VALUES (9, 'fpereira@moraes.net');
INSERT INTO EMAIL_PROF (ID_PROF, END_EMAIL) VALUES (10, 'nascimentoesther@hotmail.com');

```

```

INSERT INTO TELEFONE_PROF (ID_PROF, NUM_TEL) VALUES (1, '(071) 2266 2040');
INSERT INTO TELEFONE_PROF (ID_PROF, NUM_TEL) VALUES (2, '+55 21 4221-6539');
INSERT INTO TELEFONE_PROF (ID_PROF, NUM_TEL) VALUES (3, '+55 (061) 1759 9280');
INSERT INTO TELEFONE_PROF (ID_PROF, NUM_TEL) VALUES (4, '61 5354 9864');
INSERT INTO TELEFONE_PROF (ID_PROF, NUM_TEL) VALUES (5, '0900-311-2532');
INSERT INTO TELEFONE_PROF (ID_PROF, NUM_TEL) VALUES (6, '84 3709 8967');
INSERT INTO TELEFONE_PROF (ID_PROF, NUM_TEL) VALUES (7, '+55 (081) 7700 2164');
INSERT INTO TELEFONE_PROF (ID_PROF, NUM_TEL) VALUES (8, '+55 11 1046 1032');
INSERT INTO TELEFONE_PROF (ID_PROF, NUM_TEL) VALUES (9, '71 2900-9237');
INSERT INTO TELEFONE_PROF (ID_PROF, NUM_TEL) VALUES (10, '84 4482-5104');

```

```

INSERT INTO ESPECIALIDADE (ID_PROF, DEF_ESP) VALUES (1, 'Inglês');
INSERT INTO ESPECIALIDADE (ID_PROF, DEF_ESP) VALUES (2, 'Inglês');
INSERT INTO ESPECIALIDADE (ID_PROF, DEF_ESP) VALUES (3, 'Espanhol');
INSERT INTO ESPECIALIDADE (ID_PROF, DEF_ESP) VALUES (4, 'Espanhol');
INSERT INTO ESPECIALIDADE (ID_PROF, DEF_ESP) VALUES (5, 'Francês');
INSERT INTO ESPECIALIDADE (ID_PROF, DEF_ESP) VALUES (6, 'Espanhol');
INSERT INTO ESPECIALIDADE (ID_PROF, DEF_ESP) VALUES (7, 'Inglês');
INSERT INTO ESPECIALIDADE (ID_PROF, DEF_ESP) VALUES (8, 'Francês');
INSERT INTO ESPECIALIDADE (ID_PROF, DEF_ESP) VALUES (9, 'Inglês');
INSERT INTO ESPECIALIDADE (ID_PROF, DEF_ESP) VALUES (10, 'Francês');

```

-- Inserção das aulas e associação com professores

```

INSERT INTO AULA (ID_AULA, CARGA_HOR, HORARIO, DATA, CONTEUDO) VALUES (1, 2,
'10:00:00', '2025-01-17', 'Aula de gramática');
INSERT INTO AULA (ID_AULA, CARGA_HOR, HORARIO, DATA, CONTEUDO) VALUES (2, 2,
'16:00:00', '2024-11-27', 'Aula de vocabulário');
INSERT INTO AULA (ID_AULA, CARGA_HOR, HORARIO, DATA, CONTEUDO) VALUES (3, 1,
'10:00:00', '2024-09-01', 'Aula de vocabulário');
INSERT INTO AULA (ID_AULA, CARGA_HOR, HORARIO, DATA, CONTEUDO) VALUES (4, 2,
'8:00:00', '2024-07-21', 'Aula de gramática');
INSERT INTO AULA (ID_AULA, CARGA_HOR, HORARIO, DATA, CONTEUDO) VALUES (5, 3,
'17:00:00', '2025-03-26', 'Aula de vocabulário');
INSERT INTO AULA (ID_AULA, CARGA_HOR, HORARIO, DATA, CONTEUDO) VALUES (6, 2,
'14:00:00', '2025-05-25', 'Aula de gramática');
INSERT INTO AULA (ID_AULA, CARGA_HOR, HORARIO, DATA, CONTEUDO) VALUES (7, 3,
'17:00:00', '2024-12-06', 'Aula de gramática');
INSERT INTO AULA (ID_AULA, CARGA_HOR, HORARIO, DATA, CONTEUDO) VALUES (8, 3,
'20:00:00', '2025-07-09', 'Aula de conversação');
INSERT INTO AULA (ID_AULA, CARGA_HOR, HORARIO, DATA, CONTEUDO) VALUES (9, 3,
'10:00:00', '2024-10-06', 'Aula de conversação');
INSERT INTO AULA (ID_AULA, CARGA_HOR, HORARIO, DATA, CONTEUDO) VALUES (10, 3,

```

'20:00:00', '2024-10-29', 'Aula de conversação');

```
INSERT INTO LECIONADA (ID_AULA, ID_PROF) VALUES (1, 10);
INSERT INTO LECIONADA (ID_AULA, ID_PROF) VALUES (2, 7);
INSERT INTO LECIONADA (ID_AULA, ID_PROF) VALUES (3, 8);
INSERT INTO LECIONADA (ID_AULA, ID_PROF) VALUES (4, 6);
INSERT INTO LECIONADA (ID_AULA, ID_PROF) VALUES (5, 4);
INSERT INTO LECIONADA (ID_AULA, ID_PROF) VALUES (6, 8);
INSERT INTO LECIONADA (ID_AULA, ID_PROF) VALUES (7, 2);
INSERT INTO LECIONADA (ID_AULA, ID_PROF) VALUES (8, 3);
INSERT INTO LECIONADA (ID_AULA, ID_PROF) VALUES (9, 9);
INSERT INTO LECIONADA (ID_AULA, ID_PROF) VALUES (10, 10);
```

### 5.3 – COMANDOS SELECT

Os comandos SELECT são utilizados para **consultar e recuperar informações** armazenadas no banco de dados. Eles permitem visualizar dados de uma ou mais tabelas, com ou sem filtros, possibilitando análises, geração de relatórios e suporte à tomada de decisão.

Com SELECT, é possível exibir colunas específicas, aplicar condições (WHERE), ordenar resultados (ORDER BY), agrupar dados (GROUP BY) e combinar tabelas (JOIN), tornando-o um dos comandos mais importantes do SQL.

-- 1. Listar nome dos alunos, endereço, nível e idioma do curso

```
SELECT A.NOME AS Aluno, A.RUA AS Endereco, C.NIVEL, I.NOME AS Idioma
FROM ALUNO A
JOIN TEM_MATRICULA M ON A.ID_ALUNO = M.ID_ALUNO
JOIN CURSO C ON M.ID_CURSO = C.ID_CURSO
JOIN IDIOMA I ON C.ID_IDIOMA = I.ID_IDIOMA;
```

-- 2. Listar nome dos professores e os idiomas que lecionam

```
SELECT P.NOME AS Professor, E.DEF_ESP AS Idioma
FROM PROFESSOR P
JOIN ESPECIALIDADE E ON P.ID_PROF = E.ID_PROF;
```

-- 3. Listar nomes dos alunos e status de pagamento

```
SELECT A.NOME AS Aluno, M.PAGO, M.PENDENTE
FROM ALUNO A
JOIN TEM_MATRICULA TM ON A.ID_ALUNO = TM.ID_ALUNO
JOIN MENSALIDADE M ON TM.ID_PAG = M.ID_PAG;
```

-- 4. Listar nome, e-mail e telefone dos professores

```
SELECT P.NOME, EP.END_EMAIL, TP.NUM_TEL
FROM PROFESSOR P
LEFT JOIN EMAIL_PROF EP ON P.ID_PROF = EP.ID_PROF
LEFT JOIN TELEFONE_PROF TP ON P.ID_PROF = TP.ID_PROF;
```

-- 5. Listar aulas com data, conteúdo e professor responsável

```
SELECT A.DATA, A.CONTEUDO, P.NOME AS Professor
FROM AULA A
JOIN LECIONADA L ON A.ID_AULA = L.ID_AULA
JOIN PROFESSOR P ON L.ID_PROF = P.ID_PROF;
```

-- 6. Listar alunos matriculados em cursos de Espanhol

```
SELECT AL.NOME AS Aluno, I.NOME AS Idioma
FROM ALUNO AL
JOIN TEM_MATRICULA TM ON AL.ID_ALUNO = TM.ID_ALUNO
JOIN CURSO C ON TM.ID_CURSO = C.ID_CURSO
JOIN IDIOMA I ON C.ID_IDIOMA = I.ID_IDIOMA
WHERE I.NOME = 'Espanhol';
```

-- 7. Listar professores e a quantidade de aulas que lecionam

```
SELECT P.NOME AS Professor, COUNT(L.ID_AULA) AS Total_Aulas
FROM PROFESSOR P
JOIN LECIONADA L ON P.ID_PROF = L.ID_PROF
GROUP BY P.NOME;
```

-- 8. Listar cursos com número de alunos matriculados

```
SELECT I.NOME AS Idioma, C.NIVEL, COUNT(TM.ID_ALUNO) AS Alunos_Matriculados
FROM CURSO C
JOIN IDIOMA I ON C.ID_IDIOMA = I.ID_IDIOMA
JOIN TEM_MATRICULA TM ON C.ID_CURSO = TM.ID_CURSO
GROUP BY I.NOME, C.NIVEL;
```

-- 9. Listar nome, email e telefone dos alunos

```
SELECT A.NOME, E.END_EMAIL, T.NUM_TEL
FROM ALUNO A
LEFT JOIN EMAIL E ON A.ID_ALUNO = E.ID_ALUNO
LEFT JOIN TELEFONE T ON A.ID_ALUNO = T.ID_ALUNO;
```

-- 10. Listar pagamentos com nome do aluno e valor

```
SELECT A.NOME AS Aluno, M.VALOR, M.DATA_PAG, M.PAGO
FROM ALUNO A
JOIN TEM_MATRICULA TM ON A.ID_ALUNO = TM.ID_ALUNO
JOIN MENSALIDADE M ON TM.ID_PAG = M.ID_PAG;
```

## 6 – CONCLUSÃO

O desenvolvimento da base de dados para a escola de idiomas *Global Language Center* permitiu compreender e aplicar de forma prática os principais conceitos estudados na disciplina de Banco de Dados I. Através da modelagem conceitual, foi possível representar com clareza as entidades e os relacionamentos fundamentais do sistema, garantindo uma estrutura coerente com as necessidades da instituição. Na modelagem lógica, organizamos as informações em tabelas relacionais com chaves primárias, estrangeiras e

tipos de dados apropriados, respeitando as regras de integridade referencial.

A normalização, por sua vez, foi essencial para eliminar redundâncias e garantir a consistência dos dados, assegurando que a base estivesse bem estruturada até a Terceira Forma Normal (3FN). A implementação, realizada por meio de comandos SQL CREATE TABLE, INSERT e SELECT, demonstrou a capacidade de estruturar e consultar eficientemente os dados, apoiando o controle de matrículas, professores, cursos, aulas e pagamentos.

Com a aplicação prática deste projeto, foi possível consolidar o conhecimento teórico, desenvolver habilidades técnicas e perceber a importância de um bom projeto de banco de dados na organização, análise e segurança das informações. Essa base servirá como suporte confiável para a gestão escolar e também como fundamento para o desenvolvimento de aplicações futuras mais completas e robustas.

## 7 – REFERÊNCIAS

ELMASRI, R.; NAVATHE, S. B. *Sistemas de Banco de Dados*. 7. ed. São Paulo: Pearson, 2017.

UFVJM – Universidade Federal dos Vales do Jequitinhonha e Mucuri. Material didático da disciplina Banco de Dados I. Diamantina, 2025.

MySQL Documentation. Disponível em: <https://dev.mysql.com/doc/>. Acesso em: 09 jul. 2025.