

## Conceitos de chaves e index

- Chave Primária (PRIMARY KEY)
  - Chave Estrangeira (FOREIGN KEY)
  - Índice (INDEX)
- 

## Conjunto de Tabelas: Sistema Escolar

### Tabelas

1. **alunos** – armazena informações dos alunos.
  2. **turmas** – representa as turmas ou classes.
  3. **matriculas** – registra quais alunos estão matriculados em quais turmas.
- 

## Criação das Tabelas com Explicações

### Tabela **alunos**

sql

CopiarEditar

```
CREATE TABLE alunos (  
  id INT AUTO_INCREMENT PRIMARY KEY, -- chave primária  
  nome VARCHAR(100),  
  data_nascimento DATE  
);
```

- **id** é a **chave primária**. Ela garante que cada aluno tenha um identificador único.
-

## Tabela turmas

sql

CopiarEditar

```
CREATE TABLE turmas (  
  id INT AUTO_INCREMENT PRIMARY KEY, -- chave primária  
  nome VARCHAR(50),  
  ano_letivo INT,  
  INDEX (ano_letivo) -- índice para acelerar buscas por ano  
);
```

- `id` é a **chave primária**.
  - O campo `ano_letivo` tem um **índice** criado com `INDEX (ano_letivo)`, o que ajuda a acelerar consultas que filtram por ano.
- 

## Tabela matriculas

sql

CopiarEditar

```
CREATE TABLE matriculas (  
  id INT AUTO_INCREMENT PRIMARY KEY, -- chave primária  
  aluno_id INT,  
  turma_id INT,  
  data_matricula DATE,  
  FOREIGN KEY (aluno_id) REFERENCES alunos(id), -- chave estrangeira  
  FOREIGN KEY (turma_id) REFERENCES turmas(id) -- chave estrangeira  
);
```

- `aluno_id` e `turma_id` são **chaves estrangeiras**, pois se referem às chaves primárias das tabelas `alunos` e `turmas`.
  - Isso garante a integridade referencial: uma matrícula só pode existir se o aluno e a turma existirem.
- 

## Explicação dos Conceitos

### Chave Primária (PRIMARY KEY)

- Identificador **único** e **obrigatório** de cada linha.
- Não permite valores duplicados ou nulos.
- Ex: `id` em `alunos`, `turmas`, `matriculas`.

## Chave Estrangeira (FOREIGN KEY)

- Um campo que **referencia** uma **chave primária** em outra tabela.
- **Garante integridade referencial**, impedindo dados inválidos.
- Ex: `aluno_id` referencia `alunos(id)`.

## Índice (INDEX)

- Utilizado para **melhorar a performance** das consultas.
- Funciona como um “atalho” para localizar dados mais rapidamente.
- Ex: índice em `ano_letivo` na tabela `turmas`.

---

## Exemplos de Consultas Práticas

### 1. Buscar alunos de uma turma específica

```
sql
CopiarEditar
SELECT a.nome
FROM alunos a
JOIN matriculas m ON m.aluno_id = a.id
WHERE m.turma_id = 2;
```

### 2. Contar número de alunos por turma

```
sql
CopiarEditar
SELECT t.nome, COUNT(*) AS total_alunos
FROM turmas t
JOIN matriculas m ON t.id = m.turma_id
GROUP BY t.id;
```

### 3. Buscar turmas do ano de 2025 (usando o índice)

sql

CopiarEditar

```
SELECT * FROM turmas WHERE ano_letivo = 2025;
```

---



## Exercícios Propostos

1. Crie a tabela **professores** com os campos **id**, **nome** e **disciplina**. Use **id** como chave primária.
  2. Adicione uma chave estrangeira na tabela **turmas** para vincular um professor responsável pela turma.
  3. Liste todos os alunos matriculados em turmas do ano letivo de 2025.
  4. Crie um índice no campo **nome** da tabela **alunos**. Por quê isso pode ser útil?
  5. Explique o que aconteceria se tentássemos inserir uma matrícula com um **aluno\_id** que não existe. Teste no MySQL e anote o erro.
-