

Primary Key

A Primary Key (PK) ou Chave Primária é um campo (ou conjunto de campos) que identifica unicamente cada registro em uma tabela do banco de dados.

Características da PK:

- Valor único: Não pode haver dois registros com o mesmo valor.
- Não pode ser NULL: Sempre deve ter um valor definido.
- Uma por tabela: Cada tabela pode ter apenas uma Primary Key.



Coluna que possui
regra de chave primária

cliente	id_cliente	nome	data_nasc	sexo
	1	José	1978-04-21	m
	2	Maria	1980-10-17	f
	3	João	1995-08-12	m
	4	Pedro	1990-03-18	m

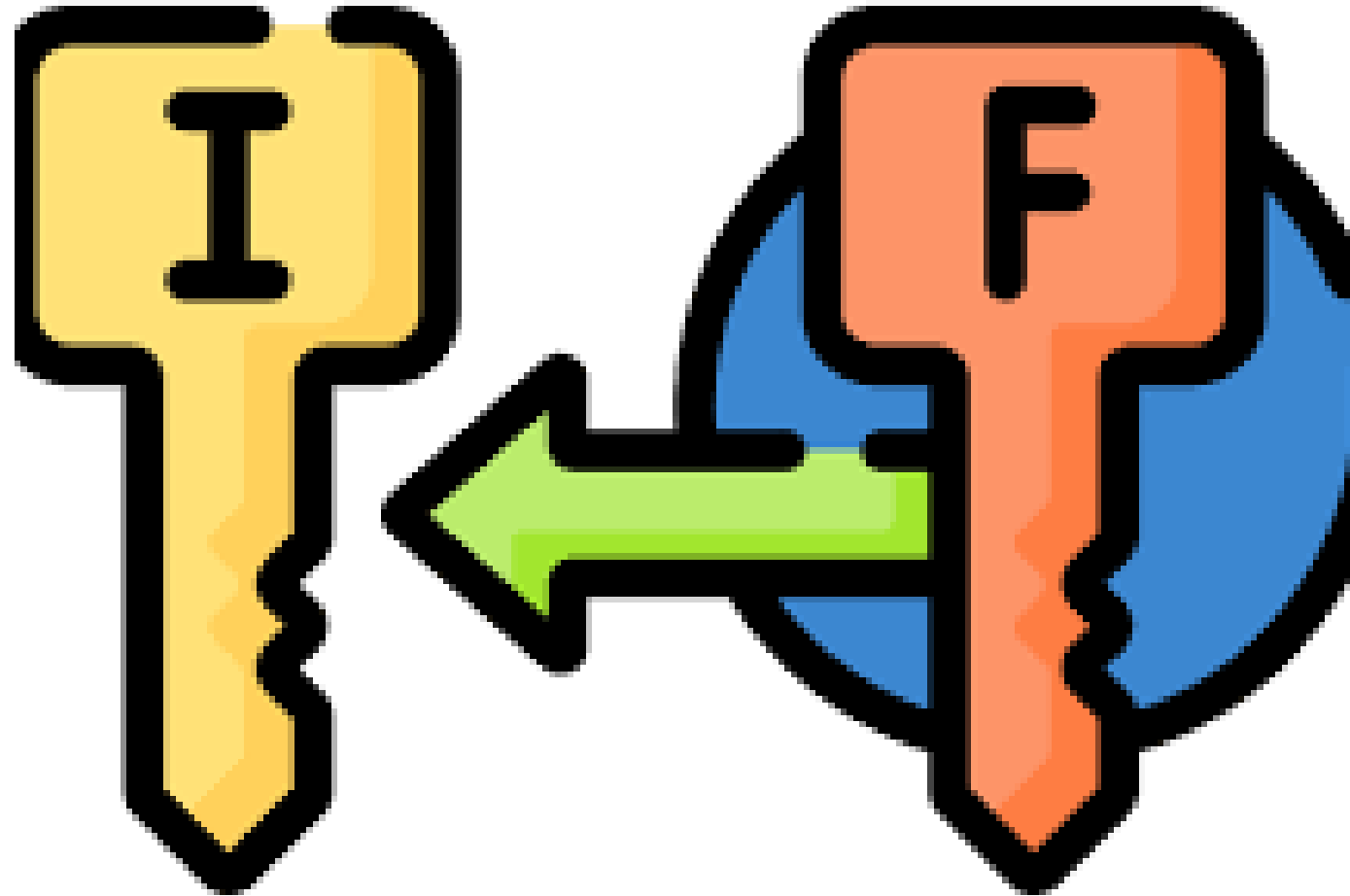
Foreign Key

A Foreign Key (FK) ou Chave Estrangeira é um campo em uma tabela que referencia a Primary Key (PK) de outra tabela.

Ela estabelece um relacionamento entre os dados, garantindo a integridade referencial no banco de dados.

Principais Características da Foreign Key:

- Relaciona duas tabelas.



Chave
Primária

Chave
Estrangeira

Chave
Primária

MATRÍCULA (dependent)

DEPTO (parent)

<u>Matricula</u>	Nome	Endereço	<u>Depto.</u>
1001	João	Rua 5	108
1003	Pedro	Rua 8	123
1004	Manoel	Rua 6	120
1005	Pedro	Rua 7	123
1007	Maria	Rua6	132

<u>Depto.</u>	Descrição
105	Vendas
108	Cobrança
120	Fiscal
123	Contábil
132	Atendimento

INTEGRIDADE
REFERENCIAL

PK: Identifica um registro de forma única na tabela.

FK: Relaciona duas tabelas, referenciando uma PK.

TABELA CLIENTES

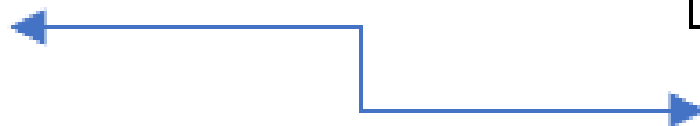
ID_CLIENTE	NOME
1	DANIEL
2	JOAO
3	PAULO

PK

TABELA VENDAS

IDCLIENTE	PRODUTO	QUANTIDADE
1	AGUA	1
1	BISCOITO	2
1	CHOCOLATE	1
2	AGUA	2

FK



Como cria uma FK?

O comando **CONSTRAINT** é usado no MySQL para adicionar restrições (constraints) em uma tabela.

- **CONSTRAINT**: Este comando é usado para definir uma **restrição** (ou regra) em uma tabela. No caso de chaves estrangeiras, essa restrição assegura que os dados em uma coluna de uma tabela correspondam aos dados de uma coluna em outra tabela.

CONSTRAINT **fk_alunos_cursos** **FOREIGN KEY** (**curso_id**) **REFERENCES** **cursos**(**curso_id**)

- **CONSTRAINT:** usado para definir uma **restrição** (ou regra) em uma tabela.
- **fk_alunos_cursos:** Este é o **nome da chave estrangeira**.
- **FOREIGN KEY:** Isso indica que estamos criando uma chave estrangeira.
- **(curso_id):** Dentro dos parênteses, especificamos qual coluna da tabela atual será a **chave estrangeira**.
- **REFERENCES:** Essa palavra-chave indica que estamos especificando a tabela e a coluna que a chave estrangeira (**curso_id** da tabela **alunos**) irá referenciar.
- **cursos:** Este é o nome da tabela referenciada
- **(curso_id):** Esse é o nome da coluna na tabela cursos que contém os valores que serão referenciados pela chave estrangeira na tabela alunos

Hands ON

Regra de Negócio

- Um aluno pertence a um curso (1 curso → vários alunos).
- Um curso tem várias matérias (1 curso → várias matérias).
- Uma matéria pertence a um único curso (1 matéria → 1 curso).



Criando DB

Vamos criar o banco de dados para a aula e usá-lo.

Banco: **db_aula_pk_fk**

The screenshot shows a database management tool interface. On the left is the 'Navigator' pane with a 'SCHEMAS' section. It contains a search bar 'Filter objects' and a tree view where 'db_aula_pk_fk' is expanded, showing sub-items for 'Tables', 'Views', 'Stored Procedures', and 'Functions'. Below these are 'phpmyadmin' and 'test' databases. The main area is titled 'Query 1' and contains two SQL statements: '1 • CREATE DATABASE db_aula_pk_fk;' and '2 • USE db_aula_pk_fk;'. Below the query editor is the 'Output' pane, which is set to 'Action Output'. It displays a table with the results of the executed queries.

#	Time	Action	Message
✓ 1	15:52:28	CREATE DATABASE db_aula_pk_fk	1 row(s) affected
✓ 2	15:52:36	USE db_aula_pk_fk	0 row(s) affected

Tabela Curso

```
3
4 ● ⊖ CREATE TABLE cursos(
5     id_curso INT AUTO_INCREMENT PRIMARY KEY,
6     nome VARCHAR (100) NOT NULL,
7     descricao VARCHAR (255)
8
9     );
10
```

Tabela Alunos

```
11 • ⊖ CREATE TABLE alunos (  
12     id_aluno INT AUTO_INCREMENT PRIMARY KEY,  
13     nome VARCHAR(100) NOT NULL,  
14     idade INT NOT NULL,  
15     email VARCHAR (100) UNIQUE NOT NULL,  
16     data_nascimento DATE NOT NULL,  
17     id_curso INT,  
18     telefone VARCHAR (20),  
19     CONSTRAINT fk_aluno_curso FOREIGN KEY (id_curso) REFERENCES cursos(id_curso)  
20 );
```

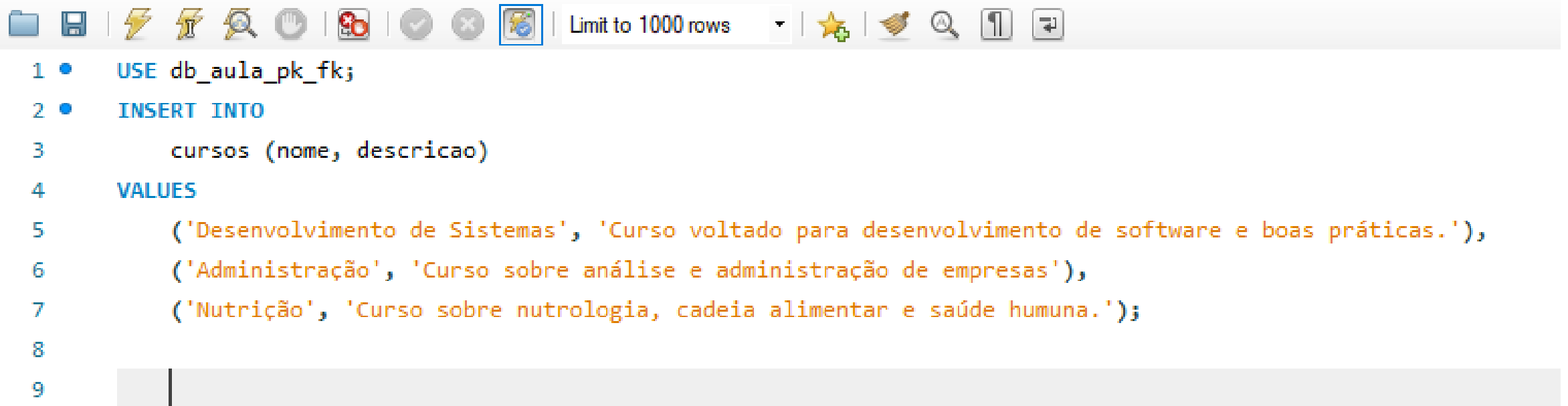
Tabela Materias

```
21
22 ● ⊖ CREATE TABLE materias (
23     materia_id INT AUTO_INCREMENT PRIMARY KEY,
24     nome VARCHAR(100) NOT NULL,
25     id_curso INT,
26     professor VARCHAR(100),
27     carga_horaria INT NOT NULL,
28     CONSTRAINT fk_materias_cursos FOREIGN KEY (id_curso) REFERENCES cursos(id_curso)
29 );
30
```

DML's



Inserindo na tabela cursos



The image shows a screenshot of a SQL IDE interface. The top toolbar contains various icons for file operations, execution, and navigation. A dropdown menu is open, showing 'Limit to 1000 rows'. The main editor area displays a SQL query with line numbers 1 through 9 on the left. The query is as follows:

```
1 • USE db_aula_pk_fk;
2 • INSERT INTO
3     cursos (nome, descricao)
4 VALUES
5     ('Desenvolvimento de Sistemas', 'Curso voltado para desenvolvimento de software e boas práticas.'),
6     ('Administração', 'Curso sobre análise e administração de empresas'),
7     ('Nutrição', 'Curso sobre nutrologia, cadeia alimentar e saúde humuna.');
```

Validando registros na tabela cursos

14 •

```
SELECT * FROM cursos
```



Result Grid



Filter Rows:

Edit:



Export/Import:



Wrap Cell Content:



	id_curso	nome	descricao
▶	1	Desenvolvimento de Sistemas	Curso voltado para desenvolvimento de software e boas práticas.
	2	Administração	Curso sobre análise e administração de empresas
	3	Nutrição	Curso sobre nutrologia, cadeia alimentar e saúde humana.
⊙	NULL	NULL	NULL

Inserindo Registros na tabela Materias

```
10
11 • USE db_aula_pk_fk;
12 • INSERT INTO
13     materias (nome, id_curso, professor, carga_horaria)
14     VALUES
15     ('Banco de Dados II', 1, 'Cleiton da Silva', 60),
16     ('Gestão e Administração de Empresas I', 2, 'Thiago Pascoto', 50),
17     ('Cálculos de Biopedância', 3, 'Miriam Machado', 55);
18
```

Validando registros na tabela materias

19 • `select * from materias;`



Result Grid



Filter Rows:

Edit:



Export/Import:



	materia_id	nome	id_curso	professor	carga_horaria
▶	1	Banco de Dados II	1	Cleiton da Silva	60
	2	Gestão e Administração de Empresas I	2	Thiago Pascoto	50
	3	Cálculos de Biopedância	3	Miriam Machado	55
⊙	NULL	NULL	NULL	NULL	NULL

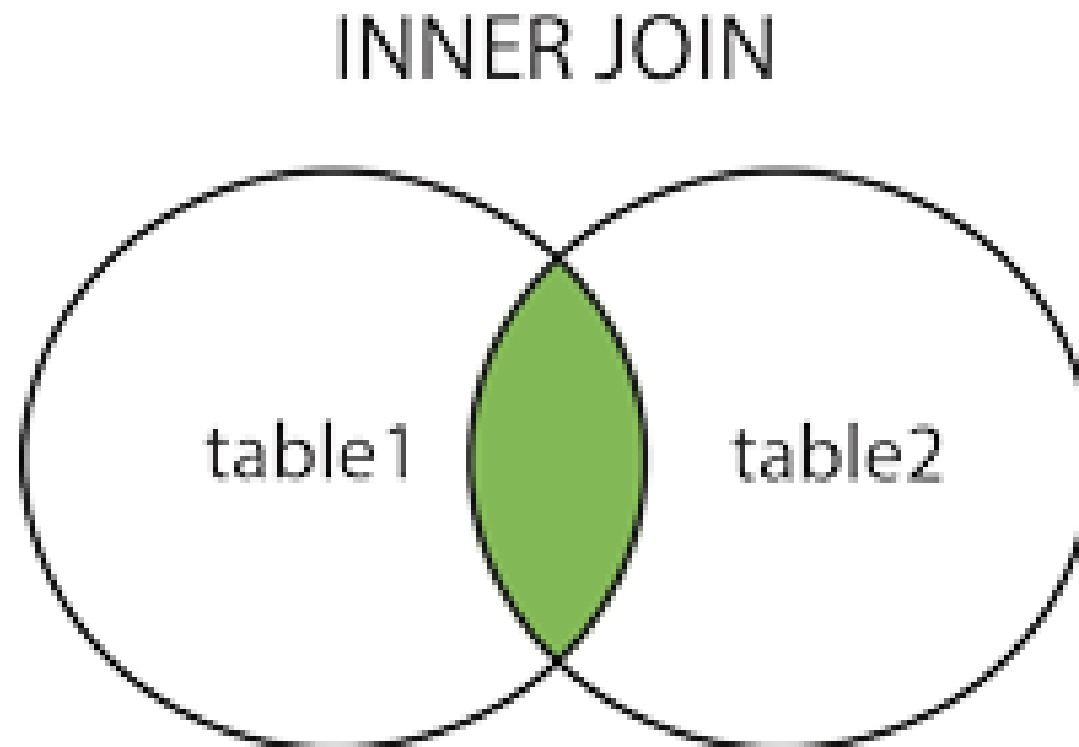
Inserindo registros na tabela alunos

```
22 • USE db_aula_pk_fk;
23 • INSERT INTO
24     alunos (nome, idade, email, data_nascimento, id_curso, telefone)
25 VALUES
26     ('Geovanna Passos', 23, 'geovanna.passos@etec.gov.br', '2001-05-10', 1, '(11) 91234-5678'),
27     ('Michael Jackson', 25, 'michael.jackson@etec.gov.br', '1999-08-22', 2, '(21) 99876-5432'),
28     ('Renata Fan', 24, 'renata.fan@etec.gov.br', '2000-12-03', 3, '(31) 98765-4321');
29
```


INNER JOIN

Combina registros de duas ou mais tabelas **somente quando há correspondência** entre elas, com base em uma chave comum (**chave primária e chave estrangeira**).

Geralmente utilizamos para exibir **apenas registros que possuem correspondência** em ambas as tabelas.



```
SELECT coluna1, coluna2, ...  
FROM tabelaA  
INNER JOIN tabelaB ON tabelaA.chave_primaria = tabelaB.chave_estrangeira;
```

Sintaxe

Verificando
todos alunos
matriculados
em um curso

```
32 • SELECT
33     alunos.nome, cursos.nome
34 FROM
35     alunos
36 INNER JOIN
37     cursos
38 ON
39     alunos.id_curso = cursos.id_curso;
```

	nome	nome
▶	Geovanna Passos	Desenvolvimento de Sistemas
	Michael Jackson	Administração
	Renata Fan	Nutrição



Desafio 1

Exibir nome da
matéria, o
professor
responsável e o
curso ao qual
pertence.


```

41      /*desafio*/
42      •  SELECT
43          materias.nome AS Matéria,
44          materias.professor,
45          cursos.nome AS Curso
46      FROM
47          materias
48      INNER JOIN
49          cursos
50      ON materias.id_curso = cursos.id_curso;

```



Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	Matéria	professor	Curso
▶	Banco de Dados II	Cleiton da Silva	Desenvolvimento de Sistemas
	Gestão e Administração de Empresas I	Thiago Pascoto	Administração
	Cálculos de Biopedância	Miriam Machado	Nutrição

Desafio 2

Exibir o **nome do curso** e a **quantidade de matérias** que ele possui

- Count
- Group by



```

52
53      /*desafio 2*/
54  •   SELECT
55          cursos.nome AS curso,
56          COUNT(materias.materia_id) AS total_materias
57  FROM
58          cursos
59  INNER JOIN
60          materias
61  ON
62          cursos.id_curso = materias.id_curso
63  GROUP BY
64          cursos.nome;

```

<

Result Grid



Filter Rows:

Export:



Wrap Cell Content:

	curso	total_materias
▶	Administração	1
	Desenvolvimento de Sistemas	1
	Nutrição	1

Tem muitas coisas
que podemos
aprender...



Professores

- Cleiton S Dias
- Thiago G Pascotto

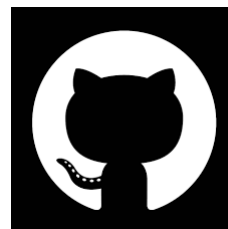




Contatos



cleitondsd



cleitondsd



(11) 9 3029-0421