

# LINGUAGEM DE PROGRAMAÇÃO DE BANCO DE DADOS

**AULA 04**

**Profº Me. Sergio Eduardo Nunes**

- ✓ FAZER CONSULTAS UTILIZANDO:
  - ✓ *INNER JOIN*;
  - ✓ *LEFT JOIN*;
  - ✓ *RIGHT JOIN*.

## Tabela Categoria:

**Numero:** inteiro, não nulo, auto incremento e chave primaria.;

**Tipo,** recebe nomes com tamanho 50, não nulo.

## Tabela Produtos:

**id:** inteiro, não nulo, auto incremento e chave primaria.;

**Nome:** recebe nomes com tamanho 50, não nulo;

**Preço:** decimal com 6 números e 2 após a virgula, não nulo.

**CatNum\_FK,** inteiro, não nulo, chave estrangeira.

Cadastre duas categorias: Livros e DVD.

```
+-----+-----+
| Numero | Tipo  |
+-----+-----+
|      1 | Livros|
|      2 | Dvd   |
+-----+-----+
2 rows in set (0.00 sec)
```

Cadastre dois livros e dois DVDs a sua escolha.

```
+-----+-----+-----+-----+
| Id | Nome                | Preço | CatNum |
+-----+-----+-----+-----+
|  1 | Codigo da Vinci    | 39.99 |      1 |
|  2 | Diario de um mago  | 19.99 |      1 |
|  3 | Hancock            | 89.99 |      2 |
|  4 | Eu sou a lenda     | 79.99 |      2 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Esta instrução SQL serve para unir os resultados do relacionamento de duas ou mais tabelas. A palavra JOIN pode ser traduzida como UNIR. Conhecida como conexão interna.

## **INNER JOIN**

```
SELECT Tabela1.Coluna1
```

```
FROM Tabela1 INNER JOIN Tabela2
```

```
ON Tabela1.ChaveEstrangeira = Tabela2.ChavePrimaria;
```

# INNER JOIN

```
SELECT Categoria.Tipo, Produto.Nome, Produto.Preco  
FROM Produto INNER JOIN Categoria  
ON produto.CatNum_FK = Categoria.Numero;
```

```
+-----+-----+-----+  
| Tipo  | Nome                | Preço |  
+-----+-----+-----+  
| Livros |Codigo da Vinci     | 39.99 |  
| Livros |Diario de um mago   | 19.99 |  
| Dvd    |Hancock             | 89.99 |  
| Dvd    |Eu sou a lenda      | 79.99 |  
+-----+-----+-----+  
4 rows in set (0.00 sec)
```

Traduzindo...

Faça a união da **tabela produto** com a **tabela categoria**

onde a chave **Produto.CatNum** seja igual a chave

**Categoria.Numero;**

# VAMOS PRATICAR ?



- 1) Insira duas novas categorias, chamadas “escritório” e “Doces”
  - a) Se utilizarmos o JOIN do exemplo, porque não exibe as novas categorias na consulta?
  
- 2) Cadastre uma caneta na tabela Produtos.
  - a) Porque agora exibiu?
  
- 3) Faça uma consulta que retorne as seguintes colunas, na ordem:  
Id do Produto, Nome do produto, Preço do produto e Tipo da Categoria.



A instrução SQL LEFT JOIN retorna a tabela1 inteira e apenas os registros que coincidem com igualdade do join da Tabela2.

```
SELECT Tabela1.*, Tabela2.*  
FROM Tabela1 LEFT JOIN Tabela2  
ON Tabela1.ChaveEstrangeira = Tabela2.chavePrimaria;
```

# LEFT JOIN

Exemplo:

```
SELECT Categoria.*, Produtos.*
```

```
FROM Categoria LEFT JOIN Produto
```

```
ON Categoria.Numero = Produto.CatNum_FK;
```

Numero	Tipo	Id	Nome	Preco	CatNum
1	Livros	1	Codigo da Vinci	39.99	1
1	Livros	2	Diario de um mago	19.99	1
2	Dvd	3	Hancock	89.99	2
2	Dvd	4	Eu sou a lenda	79.99	2
3	Escritorio	5	Caneta	1.99	3
4	Doces	NULL	NULL	NULL	NULL

6 rows in set (0.04 sec)

# VAMOS PRATICAR ?



4) Repare que na instrução utilizada no exemplo da consulta LEFT JOIN, relacionou as tabelas Categoria X Produtos.

**SELECT Categoria.\*, Produto.\***

**FROM Categoria LEFT JOIN Produto**

**ON Categoria.Numero = Produto.CatNum\_FK;**

Faça uma consulta utilizando LEFT JOIN, só que agora inverta a relação, ou seja, Produtos X Categoria.

O RIGHT JOIN segue o mesmo raciocínio do LEFT JOIN. Ou seja, retorna a tabela2 inteira, e apenas os registros que coincidem com igualdade do JOIN na Tabela1.

```
SELECT Tabela1.*, Tabela2.*  
FROM Tabela1 RIGHT JOIN Tabela2  
ON Tabela1.ChaveEstrangeira = Tabela2.chavePrimaria;
```

# RIGHT JOIN

Exemplo:

```
SELECT Categoria.*, Produto.*
```

```
FROM Categoria RIGHT JOIN Produto
```

```
ON Categoria.Numero = Produto.CatNum_FK;
```

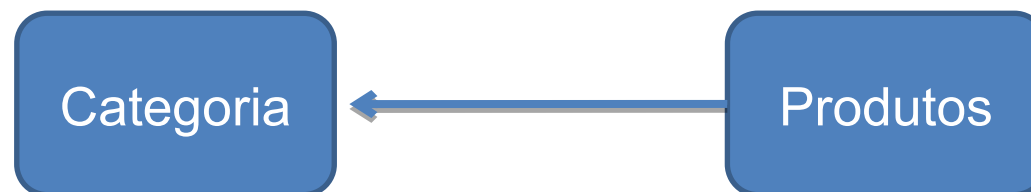
```
+-----+-----+-----+-----+-----+-----+
| Numero | Tipo      | Id  | Nome                | Preço | CatNum |
+-----+-----+-----+-----+-----+-----+
|      1 | Livros    | 1   | Codigo da Vinci     | 39.99 |      1 |
|      1 | Livros    | 2   | Diario de um mago   | 19.99 |      1 |
|      2 | Dvd       | 3   | Hancock             | 89.99 |      2 |
|      2 | Dvd       | 4   | Eu sou a lenda      | 79.99 |      2 |
|      3 | Escritorio| 5   | Caneta              | 1.99  |      3 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

# LEFT JOIN x RIGHT JOIN

Ao utilizar o comando LEFT ou RIGHT, o SQL observa o sentido para a Direita ou para a Esquerda.

- No LEFT JOIN o BD analisa a estrutura da tabela da Direita e compara com a tabela da Esquerda.
- No RIGHT JOIN o BD analisa a estrutura da tabela da Esquerda e compara com a tabela da Direita.

LEFT JOIN



RIGHT JOIN



# VAMOS PRATICAR ?





Create database LojaMotos;

Use LojaMotos;

Create table Marca(

    Codigo int primary key auto\_increment,

    Nome varchar (30) not null

);

INSERT Marca VALUES (0, "Honda"),

(0, "Yamaha"),

(0, "Kawasaki"),

(0, "Suzuki");

Create table Moto(

    Chassi int primary key,

    Nome varchar(30) not null,

    Cor varchar(20) not null,

    MarcaCodigo int not null,

    foreign key(MarcaCodigo) references Marca(Codigo)

);

INSERT Moto VALUES (12356789, "Titan", "Azul", 1),

(321654987, "Twister", "Preta", 1),

(147258369, "Factor", "Prata", 2),

(852258022, "Fazer", "Vermelha", 2),

(996655887, "Ninja 250", "Preta", 3),

(753951254, "Ninja 650", "Preta", 3),

(102030550, "CBR", "Prata", 1),

(805099604, "R1", "Amarela", 2);

```
create table Cliente(  
    Numero int(3) primary key auto_increment,  
    Nome varchar(30) not null,  
    Celular varchar(13) not null  
);  
  
INSERT Cliente VALUES (0, "Corey Taylor", "(19)988552266"),  
(0, "Joey Jordison", "(19)977441122"),  
(0, "Paul Gray", "(19)992365417"),  
(0, "Shawn Crahan", "(19)998004433"),  
(0, "James Root", "(19)920360055"),  
(0, "Mick Thomson", "(19)914523688"),  
(0, "Sid Wilson", "(19)977553344");
```

```
create table Servicos(  
    Cod int(3) primary key auto_increment,  
    Nome varchar(30) not null,  
    Preco decimal (6,2) not null  
);  
  
INSERT Servicos VALUES (0, "Troca de óleo", 120.00),  
(0, "Troca pastilha", 100.00),  
(0, "Troca transmissao", 150.00),  
(0, "Eletrica", 80.00),  
(0, "Troca cabo freio", 50.00),  
(0, "Troca cabo do acelerador", 70.00);
```

create table Produtos(

Numero int(3) primary key auto\_increment,

Nome varchar(30) not null,

Preco decimal (6,2) not null

);

INSERT Produtos VALUES (0, "Oleo", 30.00),

(0, "Pastilha de freio", 50.00),

(0, "Transmissao", 120.00),

(0, "Fio", 20.00),

(0, "Cabo freio", 25.00),

(0, "Cabo do acelerador", 30.00);

create table Vendas(

Numero int(3) primary key auto\_increment,

MotoChassi int not null,

ClieNumero int(3) not null,

ServCod int(3) not null,

ProdNumero int(3) not null,

foreign key (MotoChassi) references Moto(Chassi),

foreign key (ClieNumero) references Cliente(Numero),

foreign key (ServCod) references Servicos(Cod),

foreign key (ProdNumero) references Produtos(Numero)

);

INSERT Vendas VALUES (0, 12356789, 1, 2, 2),

(0, 321654987, 2, 4, 4),

(0, 147258369, 3, 6, 6),

(0, 852258022, 6, 1, 1),

(0, 996655887, 7, 3, 3),

(0, 753951254, 5, 5, 3),

(0, 102030550, 4, 2, 2),

(0, 805099604, 3, 3, 3);

- 5) Com base no script desenvolvido, faça as seleções de dados a seguir.
- a) Selecione o nome das motos e a sua respectiva marca, em ordem crescente pelo nome da moto.
  - b) Utilize um JOIN para demonstrar a marca que possui nenhuma moto relacionada.
  - c) Selecione o nome dos clientes e a sua respectiva moto(nome da moto, e o numero do chassi), ordene a pesquisa em ordem crescente pelo nome do cliente.

- d) Selecione o nome das motos, chassis, e o produto consumido, somente das motos da marca Honda.
- e) Selecione o nome dos clientes, e o total a pagar (produtos + serviços consumidos). Ordene em ordem crescente, como primeiro critério o total a pagar, e o segundo o nome do cliente
- f) Selecione o nome das motos, que sejam da cor preta OU prata, E da Marca Honda OU Kawasaki.
- g) Demonstre o valor médio gasto (produtos + serviços consumidos).

- h) Selecione o nome do cliente, o número do celular, o nome da moto, somente dos clientes com moto da marca Yamaha.
- i) Selecione o nome das motos e chassi das motos que trocaram óleo.
- j) Selecione o nome da moto, o chassi da moto, o nome do cliente, o nome do serviço feito, o valor do serviço, o nome do produto utilizado, o valor do produto utilizado, o total a pagar (produtos + serviços). Ordene a busca em ordem decrescente pelo nome do cliente.





a)

Nome	Nome
CBR	Honda
Factor	Yamaha
Fazer	Yamaha
Ninja 250	Kawasaki
Ninja 650	Kawasaki
R1	Yamaha
Titan	Honda
Twister	Honda

8 rows in set (0.00 sec)

b)

Marca	Moto
Honda	Titan
Honda	CBR
Honda	Twister
Yamaha	Factor
Yamaha	R1
Yamaha	Fazer
Kawasaki	Ninja 650
Kawasaki	Ninja 250
Suzuki	NULL

9 rows in set (0.04 sec)

c)

Cliente	Moto	Chassi
Corey Taylor	Titan	12356789
James Root	Ninja 650	753951254
Joey Jordison	Twister	321654987
Mick Thomson	Fazer	852258022
Paul Gray	Factor	147258369
Paul Gray	R1	805099604
Shawn Crahan	CBR	102030550
Sid Wilson	Ninja 250	996655887

8 rows in set (0.00 sec)

d)

Moto	Chassi	Produto
Titan	12356789	Pastilha de freio
CBR	102030550	Pastilha de freio
Twister	321654987	Fio

3 rows in set (0.07 sec)

e)

Cliente	Total a Pagar
Joey Jordison	100.00
Paul Gray	100.00
Corey Taylor	150.00
Mick Thomson	150.00
Shawn Crahan	150.00
James Root	170.00
Paul Gray	270.00
Sid Wilson	270.00

8 rows in set (0.00 sec)

f)

Moto	Cor
CBR	Prata
Twister	Preta
Ninja 650	Preta
Ninja 250	Preta

4 rows in set (0.00 sec)

g)

```
+-----+
| Valores Medios |
+-----+
|      170.000000 |
+-----+
1 row in set (0.00 sec)
```

h)

```
+-----+-----+-----+
| Cliente      | Celular      | Moto      |
+-----+-----+-----+
| Paul Gray    | (19)992365417 | Factor    |
| Paul Gray    | (19)992365417 | R1        |
| Mick Thomson | (19)914523688 | Fazer     |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

i)

```
+-----+-----+
| Moto   | Chassi   |
+-----+-----+
| Fazer  | 852258022 |
+-----+-----+
1 row in set (0.00 sec)
```

j)

```
+-----+-----+-----+-----+-----+-----+-----+
| Moto   | Chassi   | Cliente   | Servicos   | Valor Servicos | Produto   | Valor do Produto | Total a Pagar |
+-----+-----+-----+-----+-----+-----+-----+
| Ninja 250 | 996655887 | Sid Wilson | Troca transmissao | 150.00 | Transmissao | 120.00 | 270.00 |
| CBR      | 102030550 | Shawn Crahan | Troca pastilha | 100.00 | Pastilha de freio | 50.00 | 150.00 |
| Factor   | 147258369 | Paul Gray | Troca cabo do acelerador | 70.00 | Cabo do acelerador | 30.00 | 100.00 |
| R1       | 805099604 | Paul Gray | Troca transmissao | 150.00 | Transmissao | 120.00 | 270.00 |
| Fazer    | 852258022 | Mick Thomson | Troca de óleo | 120.00 | Oleo | 30.00 | 150.00 |
| Twister  | 321654987 | Joey Jordison | Eletrica | 80.00 | Fio | 20.00 | 100.00 |
| Ninja 650 | 753951254 | James Root | Troca cabo freio | 50.00 | Transmissao | 120.00 | 170.00 |
| Titan    | 12356789 | Corey Taylor | Troca pastilha | 100.00 | Pastilha de freio | 50.00 | 150.00 |
+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

- DATE, C. **Introdução a sistemas de banco de dados.** Rio de Janeiro: Campus, 2000.
- SILBERSCHATZ, A; KORTH, H.; SUDARSHAN, S. **Sistemas de banco de dados.** São Paulo: Makron Books, 1999.
- FERRARI, A. **Crie banco de dados em MySQL.** São Paulo: Digerati Books, 2007.

- 6) Com base no sistema de consulta de notas da faculdade. Desenvolva um banco de dados, com informações dos professores, alunos, curso, disciplina, presença e notas.
- a) Desenvolva seleções de dados (Livre), em que ocorra relacionamento entre as tabelas (JOIN).