Administração de Banco de Dados

Consulta de Dados - JOINS

Revisão!

Para dar continuidade aos conteúdos, faremos uma breve revisão sobre consultas utilizando o comando SELECT em MySQL.

Tabela teams

```
CREATE TABLE IF NOT EXISTS teams (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    creation_date DATE NOT NULL,
    city VARCHAR(200) NOT NULL,
    state VARCHAR(100) NOT NULL
);
```

```sql
CREATE TABLE IF NOT EXISTS teams (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    creation_date DATE NOT NULL,
    city VARCHAR(200) NOT NULL,
    state VARCHAR(100) NOT NULL
);
```

1. Informe o nome dos times da cidade de "Garanhuns"
2. Quantos times do estado da "Paraíba" estão cadastrados?
3. Informe o nome e a data de criação de todos os times.

Respostas

1. Informe o nome dos times da cidade de "Garanhuns"

```
SELECT name

FROM teams

WHERE city = "Garanhuns";
```

2. Quantos times do estado da "Paraíba" estão cadastrados?

```
SELECT count(id)

FROM teams

WHERE state = "Paraíba";
```

3. Informe o nome e a data de criação de todos os times.

```
SELECT nome,creation_date
FROM teams;
```

Tabela players

```sql
CREATE TABLE IF NOT EXISTS players (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    birth DATE NOT NULL,
    height DECIMAL(3,2) NOT NULL,
    weight DECIMAL(6,3) NOT NULL,
    team_id INT,
    FOREIGN KEY (team_id) REFERENCES teams(id)
);
```

```sql
CREATE TABLE IF NOT EXISTS players (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    birth DATE NOT NULL,
    height DECIMAL(3,2) NOT NULL,
    weight DECIMAL(6,3) NOT NULL,
    team_id INT,
    FOREIGN KEY (team_id) REFERENCES teams(id)
);
```

1. Informe o nome dos jogadores com peso superior a 80kg.
2. Quantos jogadores medem menos de 1.90?
3. Selecione a altura do menor jogador.

Respostas

1. Informe o nome dos jogadores com peso superior a 80kg.

```
SELECT name

FROM players

WHERE weight > 80;
```

2. Quantos jogadores medem menos de 1.90?

```
SELECT count(id)

FROM players

WHERE height < 1.90;
```

3. Selecione a altura do menor jogador.

```
SELECT min(height)
FROM players;
```

Informe a altura e data de nascimento de todos os jogadores do time Sport Club do Recife.

Consulta com duas tabelas!

Junção de Tabelas

- Com SQL você pode obter informações de colunas contidas em mais de uma tabela.
- Essa operação é chamada de operação de **junção (JOIN)**.
- As tabelas são juntas a partir de chaves que as relacionam, chave primária de uma com chave estrangeira de outra.
- Desta forma conseguimos acessar dados que necessitam de duas tabelas para fazerem sentido.

Joins em MySQL

- A cláusula JOIN é usada para combinar linhas de duas ou mais tabelas, com base em uma coluna relacionada entre elas.

- Analisaremos o seguinte exemplo:

teams

id	name	creation_da...	city	state
1	Sport Club do Recife	1905-05-13	Recife	Pernambuco
2	Ceará Sporting Club	1914-06-02	Fortaleza	Ceará
3	Esporte Clube Bahia	1931-01-01	Salvador	Bahia

players

id	name	birth	height	weight	team_id
10	Rodrigo Nunes	1986-10-10	1.81	77.200	1
11	Ana Rodrigues	1985-11-11	1.86	80.000	1
12	José Santos	1995-01-01	1.75	70.500	2
13	Ricardo Oliveira	1994-02-02	1.82	75.200	2
14	Amanda Souza	1993-03-03	1.68	68.000	2

Observe que a coluna "**teams_id**" na tabela "**players**" refere-se a "**id**" na tabela "**teams"**. A relação entre as duas tabelas acima é a coluna "**teams_id**.

Informe a altura e data de nascimento de todos os jogadores do time Sport Club do Recife.

Podemos criar a seguinte instrução SQL (que contém um INNER JOIN), que seleciona registros que têm valores correspondentes em ambas as tabelas:

```
SELECT height, birth    < ----- Retorno
FROM players            < ----- Tabela da Esquerda
INNER JOIN teams        < ----- Tabela da Direita
ON teams.id = players.team_id;  < --- Relacionamentos
```

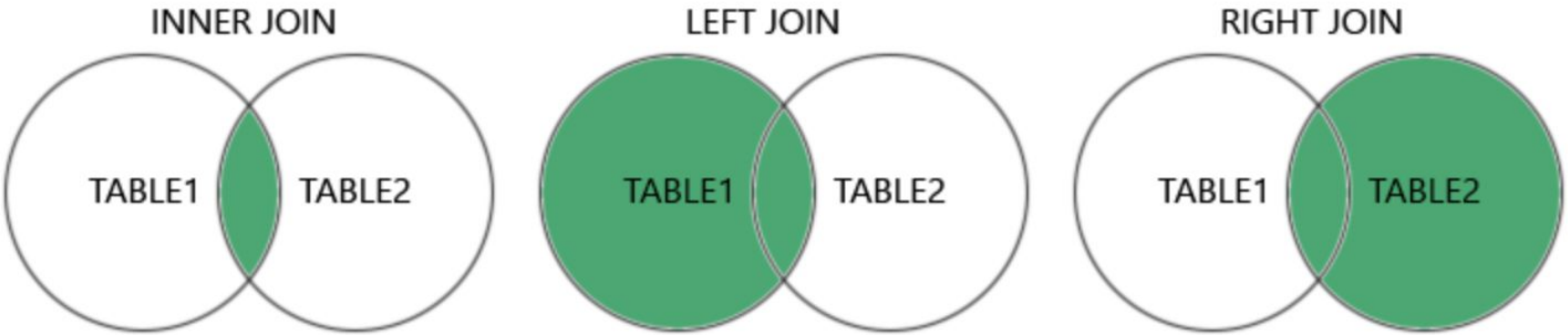
TIPOS IMPORTANTES DE JOIN SUPORTADAS NO MYSQL

INNER JOIN: Retorna registros que possuem valores correspondentes em ambas as tabelas

LEFT JOIN: Retorna todos os registros da tabela da esquerda e os registros correspondentes da tabela da direita

RIGHT JOIN: Retorna todos os registros da tabela à direita e os registros correspondentes da tabela à esquerda

TIPOS IMPORTANTES DE JOIN SUPORTADAS NO MYSQL



Consultar dados de duas tabelas

Informe o nome de todos os jogadores e nome e cidade de seus respectivos times:

```
SELECT players.name, teams.name, teams.city

FROM players

INNER JOIN teams

ON teams.id = players.team_id;
```