Joins e Having

Certified Developer
The Ultimate Tech Degree

DigitalHouse >
Coding School

Joins

Certified Developer
The Ultimate Tech Degree

DigitalHouse >
Coding School

Por que usar JOINS?

Além de fazer consultas em uma tabela ou em muitas tabelas por meio de **referência de tabela**, também é possível e necessário consultar **diferentes tabelas** e unir esses resultados com **JOINS.**

Embora tenham a mesma função que a **referência da tabela**, o **JOINS**:

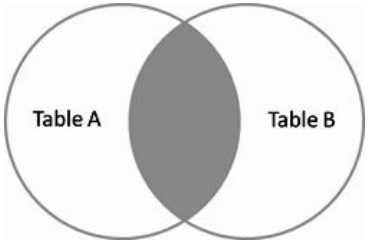
- Eles fornecem certas flexibilidades adicionais.
- Sua sintaxe é muito mais usada.
- Apresentam uma melhor performace.

INNER JOIN

O **INNER JOIN** fará um **cruzamento** entre duas tabelas. Se cruzássemos as tabelas de clientes e vendas e houvesse um cliente sem vendas, o INNER JOIN **não traria** esse cliente como resultado.

CLIENTES		
id	**nombre**	**apellido**
1	Juan	Perez
2	Clara	Sanchez
3	Marta	García

INNER JOIN



VENDAS		
id	**cliente_id**	**data**
1	2	12/03/2019
2	2	22/08/2019
3	1	04/09/2019

Criando um INNER JOIN

Antes escrevíamos:

```sql
SELECT clientes.id AS id, clientes.nome, vendas.data
FROM clientes, vendas
```

Agora escrevemos:

```sql
SELECT clientes.id AS id, clientes.nome, vendas.data
FROM clientes
INNER JOIN vendas
```

Anotações

“Embora já tenhamos dado o primeiro passo, que é **cruzar** as duas tabelas, ainda precisamos esclarecer **onde** fica esse cruzamento.
Ou seja, qual **chave primária (PK)** será cruzada com qual **chave estrangeira (FK).**”

Criando um INNER JOIN (cont.)

A sintaxe JOIN não usa **WHERE**, mas requer a palavra **ON**. É aí onde indicaremos o **filtro** a ter em conta para efetuar a travessia.

Ou seja, o que escrevíamos no **WHERE** agora vamos escrever no **ON**.

```sql
SELECT clientes.id AS id, clientes.nome, vendas.data
FROM clientes
INNER JOIN vendas
ON clientes.id = vendas.cliente_id
```

Having

Certified Developer
The Ultimate Tech Degree

DigitalHouse >
Coding School

HAVING Sintaxe

Ele cumpre a mesma função de **WHERE**, ao contrário de **HAVING** sendo usado em conjunto com as **funções de agregação** para filtrar **dados agregados.**

```sql
SELECT coluna_1
FROM nome_tabela
INNER JOIN condition
ON condition = condition
HAVING condition_Group
ORDER BY coluna_1;
```

HAVING Sintaxe

Esta consulta retornará o número de clientes por país (agrupados por país). Apenas os países com **pelo menos** 3 clientes serão incluídos no resultado.

```SQL
SELECT pais, COUNT(clienteId)
FROM clientes
HAVING COUNT(clienteId)>3;
```

Exercícios

Usando a base musimundos_V2, execute as seguintes consultas:

-- 1 - Selecione o nome dos artistas(tabela artistas) com o nome de seus albuns (tabela albuns) ordenado por id do artista.

-- 2 - Selecione o nome do cliente (clientes) e suas faturas (faturas), porém somente as faturas com valor maior de 5 ordenados pelo valor da fatura decrescente.

-- 3 Selecione o nome das cancoes (tabela cancoes) e o tipo de arquivo (tabela tipos_de_arquivos) e exiba o nome da canção e o tipo ordenado pelo id da canção.


DigitalHouse>
Coding School