

1) O que é e pra que serve o JOIN ?

R: tendo o significado de junção o JOIN é um meio de organizar colunas de uma ou mais tabelas, usando os valores em comum de cada uma das tabelas no banco de dados. Sendo assim facilita nas consultas e nos relacionamentos entre as tabelas. Ademais o JOIN serve para melhorar a compreensão sobre cada tabela em específico, e por conseguinte ajudam a tomada de decisão e são úteis para empresas em diversos contextos

2) Quando devemos utilizar ?

R: O JOIN serve para manipulação de tabelas que serve para obter uma tabela mais organizada que vai ver as necessidades do que se quer adquirir.

Existem vários tipos de JOINS são eles:

INNER JOIN traz uma ideia bem intuitiva: a de buscar a interseção entre as tabelas ($A \cap B$), traz os valores que estão em ambas as tabelas. É importante ressaltar que os dados que se encontram no INNER JOIN fazem parte de ambas.

LEFT JOIN traz uma ideia mais complexa: ela exibe os itens de uma das tabelas (uma à esquerda e outra à direita) exibindo os itens de uma das tabelas, incluindo os itens em interseção com a outra tabela.

Exemplificando “pegar as vendas da loja 1, em uma listagem completa dessas vendas, o que inclui os produtos unicamente vendidos lá e os produtos que possam ter sido vendidos na loja 2”.

RIGHT JOIN segue a mesma ideia do LEFT, só que para a outra tabela. Pega os dados completos da tabela da direita, incluindo os que também estão na tabela da esquerda.

FULL JOIN é o comando que pega todos os valores de ambas as tabelas e une em uma única resultante, também representado por ($A \cup B$).

Como pegar produtos de duas empresas diferentes e juntar esses registros para facilitar o controle e manipulação.

CROSS JOIN a execução do CROSS é complexa, onde todos os itens de uma tabela estão associados a cada item da outra tabela.

Assim, se a tabela 2 tem 7 itens e a tabela 1 tem apenas 2, cada um dos dois itens da tabela 1 será representado 7 vezes na nova tabela (com relação com todos os itens da tabela 2).

3) Sintaxe JOIN.

R: Com o intuito de facilitar a visualização final, para um usuário por exemplo, o join auxilia no fácil entendimento, visto que, sem ele seria exibido para o usuário apenas o id. Existem basicamente duas formas de se realizar a junção entre tabelas, utilizando o padrão sql 89 e sql 92, como mostrado no exemplo abaixo:

```
SQL 89
SELECT *
FROM estado, cidade;
```

```
SQL 92
SELECT *
FROM estado
INNER JOIN cidade;
– JOIN cidade; Também é possível utilizar a sintaxe sem o INNER.
```

Vale ressaltar que a única diferença entre ambas versões é apenas na sintaxe, elas possuem o mesmo desempenho e geram o mesmo resultado. Durante a consulta é possível especificar quais colunas serão apresentadas, por exemplo:

```
SELECT
    cidade.*
    ,estado.*
FROM estado, cidade;
```

Nesse exemplo, seria exibido apenas todas as colunas das tabelas estado e cidade. Também é possível especificar algumas colunas, exemplo:

```
SELECT
    cidade.nome*
    ,estado.nome*
FROM estado, cidade; (sql89)
```

```
SELECT
    estado.nome
    ,cidade.nome
```

```
FROM estado  
JOIN cidade; (sql92)
```

Nesse caso seria mostrado apenas o nome do estado e cidade.

Para que as informações sejam mostradas corretamente, sem que haja dados incorretos, utiliza-se:

```
SQL 89  
SELECT *  
FROM estado, cidade;  
WHERE cidade.estado_id = estado.id
```

```
SQL 92  
SELECT *  
FROM estado  
INNER JOIN cidade ON cidade.estado_id = estado.id;  
– JOIN cidade ON cidade.estado_id = estado.id; Também é possível utilizar a  
sintaxe sem o INNER.
```

Realizando tais comandos, haverá a comparação entre os IDS, e só haverá junção se os IDS forem iguais.