**Sistemas de Banco de Dados 1**

**Junções em SQL (JOINs)**

Aluno:

Eduardo Júnio Veloso Rodrigues 14/0168192

### **Sobre o JOIN**

Joins em SQL são métodos ou cláusulas utilizada para fazer a combinação de dados oriundos de duas ou mais tabelas de uma base de dados, baseando-se em um relacionamento entre as colunas dessas tabelas, isto é, combina colunas de uma ou mais tabelas de uma base de dados relacional afim de obter um determinado conjunto de informações.

#### Vantagens

* + Permite a combinação entre dados de tabelas diferentes de forma rápida, tendo como requisito para a realização da junção um campo em comum entre as tabelas (geralmente é a chave primária de uma tabela e estrangeira da outra);

#### Desvantagens

* + As cláusulas JOINs podem ser cascateadas e o abuso no seu uso pode dificultar a leitura e entendimento da instrução;
  + As junções criam tabelas na memória volátil (ram), desta forma, ao se realizar junções muito grandes sem levar em consideração a capacidade de memória da máquina, pode ocasionar problemas;

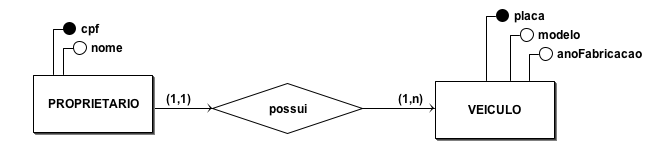
### **Tipos de JOINs**

As junções entre duas ou mais tabelas podem ser realizadas através das seguintes cláusulas:

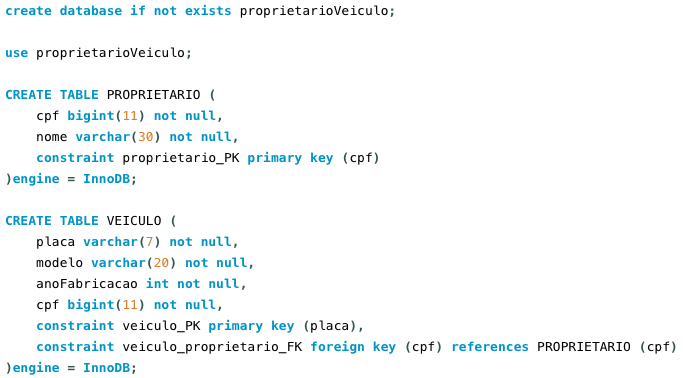
* CROSS JOIN
* INNER JOIN
* LEFT OUTER JOIN
* RIGHT OUTER JOIN
* FULL OUTER JOIN

Para a exemplificação destes tipos de joins, criou-se um projeto de banco de dados, para cadastrar proprietários de veículos e seus respectivos veículos.

#### **2.1 Diagrama Entidade Relacionamento**



#### **2.2 Script de definição da base de dados (DDL)**



#### 

#### **2.3 Script de Inserção de dados na base (DML)**

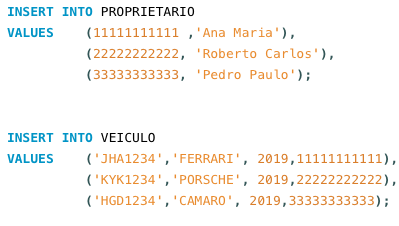
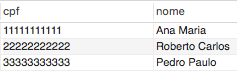
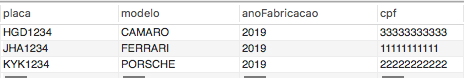
**2.4 Tabelas geradas**

Tabela de proprietarios Tabela de veiculos

#### **2.5 Cross Join**

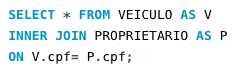
O uso do cross join se faz necessário quando se deseja juntar tabelas por cruzamento, isto é, para cada linha da tabela VEICULO queremos todos os proprietários ou vice-versa.

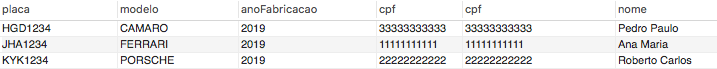
Esta ação nos dá todas as possibilidades de combinação entre as tabelas (nº de tuplas de VEICULO X nº de tuplas de PROPRIETARIO ), como mostrado abaixo.

* Comando
* Resultado 

#### **2.6 Inner Join**

O inner join faz a junção de duas ou mais tabelas por semelhança de atributos, isto é, a junção ocorre quando o atributo pelo qual se deseja fazer a junção da tabela X coincide com o atributo da tabela Y, utilizando o exemplo preparado para este estudo, queremos juntar as tabelas VEICULO e PROPRIETARIO pelo atributo chave primária de PROPRIETARIO (cpf) com o atributo chave estrangeira de VEICULO (cpf).

* **Comando**
* **Resultado**

Junção entre veículos e proprietários por cpf 

**2.7 Left Outer Join**

O left outer join faz a junção dos elementos de uma tabela X com os elementos da tabela Y, retornando também todos os outros elementos que estão contidos na tabela X mas que não estão na tabela Y, observe o diagrama de “Venn”. 

Considere o círculo verde como uma tabela X e o círculo vermelho como uma tabela Y, desta forma, X left outer join Y resultaria nos valores em comum entre X e Y, ou seja {2}, mas também resultaria em todos os valores que pertencem a X e que não pertencem a Y, ou seja, teríamos como resultado {1, 2}.

Para o exemplo projetado para este estudo, vamos inserir na tabela PROPRIETARIO uma nova tupla, mas não a relacionaremos com nenhum proprietário.

* **Inserção de um novo proprietário**

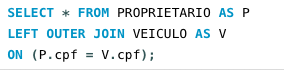


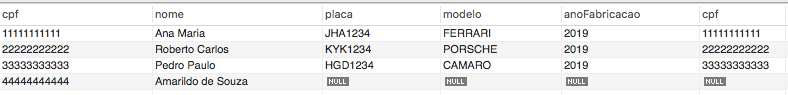
A tabela de proprietários fica assim:



Utilizando o **Left Outer Join**

* **Comando**



* **Resultado**

Note que a última tupla da tabela apresentada acima possui campos nulos referentes ao veículo, tal fato é esperado, uma vez que o proprietário não está vinculado a nenhum veículo.

#### **2.8 Right Outer Join**

#### Semelhante ao join apresentado no tópico 2.7, o right outer join faz a junção dos elementos de uma tabela Y com os elementos da tabela X, retornando também todos os outros elementos que estão contidos na tabela Y mas que não estão na tabela X, observe o diagrama de “Venn”.

Considere o círculo verde como uma tabela X e o círculo vermelho como uma tabela Y, desta forma, Y right outer join X resultaria nos valores em comum entre X e Y, ou seja {2}, mas também resultaria em todos os valores que pertencem a Y e que não pertencem a X, ou seja, teríamos como resultado { 2, 3}, diferente do left outer join que resultou em {1, 2}.

Para fins didáticos, mudou-se na tabela VEICULO a chave estrangeira cpf para aceitar valores nulos, com o objetivo de exemplificar este método, desta forma podemos inserir um veículo sem um proprietário associado.

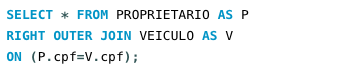
* **Comando para alterar a tabela veículo**



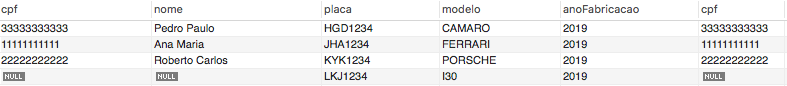
* **Inserção de um veículo sem associar um proprietário**



* **Comando**

****

* **Resultado**

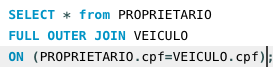
Note que a última tupla da tabela apresentada acima possui campos nulos referentes ao proprietário, tal fato é esperado, uma vez que o veículo não está vinculado a nenhum proprietário.

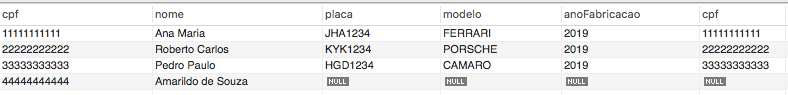
#### **2.9 Full Join ou Full Outer Join**

O outer full join ou somente full join faz a junção entre tabelas, isto é, dado uma tabela X full join Y o resultado é a união destas tabelas. Observe o diagrama de “Venn” abaixo:



Considere o círculo verde como uma tabela X e o círculo vermelho uma tabela Y, aplicando o método temos: X full join Y, resultando em todos os valores presentes em X e em Y, incluindo os valores que estão em X e não estão em Y e os valores que estão em Y e não estão em X. Aplicando ao exemplo proposto temos:

* **Comando**
* **Resultado**



**OBS**: O mysql não fornece suporte para o full outer join, mas o resultado esperado é apresentado na tabela acima e o comando que geraria esta tabela é apresentado no item “comando”. Para este exemplo dropou-se a base de dados e a criou novamente inserindo os dados iniciais. Inseriu-se também um proprietário sem associa-lo a nenhum veículo para que fosse possível obter a tabela acima.

#### 