

Banco de Dados

Linguagem SQL

DQL – Data Query Language – Select com 2 tabelas, Funções de Agregação, Group By, HAVING e Junções

Prof. Clóvis Ferraro
cferraro@unicid.edu.br



Consultas em duas ou mais Tabelas

DQL – Select com 2 tabelas

- É possível verificar a estrutura da sua tabela através do comando **DESCRIBE**.

```
DESCRIBE FABRICANTE;
```

Field	Type	Null	Key	Default	Extra
COD_FABRICANTE	int	NO	PRI	NULL	auto_increment
NOM_FABRICANTE	varchar(100)	NO		NULL	
SITE_FABRICANTE	varchar(250)	YES		NULL	
IDF_ATIVO	varchar(1)	YES		S	

DQL – Select com 2 tabelas

- Selecionando registros de duas tabelas:

```
use loja;
```

```
SELECT FAB.NOM_FABRICANTE, PROD.NOM_PRODUTO  
      FROM FABRICANTE AS FAB,  
           PRODUTO AS PROD  
      WHERE FAB.COD_FABRICANTE = PROD.COD_FABRICANTE;
```

DQL – Select com 2 tabelas

- Selecionando registros de duas tabelas:

USE LOJA;

Para saber se a tabela contém registros:

```
SELECT * FROM FABRICANTE;
```

```
SELECT * FROM PRODUTO;
```

```
SELECT FAB.NOM_FABRICANTE, PROD.NOM_PRODUTO  
FROM FABRICANTE AS FAB,  
      PRODUTO AS PROD  
WHERE FAB.COD_FABRICANTE = PROD.COD_FABRICANTE;
```

DQL – Select com 2 tabelas

- Selecionando registros de duas tabelas e **ordenando**:

```
SELECT FAB.NOM_FABRICANTE, PROD.NOM_PRODUTO  
FROM FABRICANTE AS FAB  
      , PRODUTO AS PROD  
WHERE FAB.COD_FABRICANTE = PROD.COD_FABRICANTE  
ORDER BY NOM_FABRICANTE;
```

Acrescentar o ORDER BY e verificar o resultado.

nom_fabricante	nom_produto
ACER	Ultrabook Acer Aspire M5-481T-6650 com Intel® Core™ i3-3227U
APPLE	Ultrabook Acer Aspire M5-481T-6417 com Intel® Core™ i5-3317U
ASUS	Ultrabook Acer Aspire M5-481T-6195 com Intel® Core™ i5-3317U
DELL	Ultrabook Acer Aspire com Intel® Core™ i5
HP	Computador HP 402 G1 SFF com Intel Core i3-4130, 4GB, 500GB, Gravador de ...
KINGSTON	Notebook Lenovo G400S com Intel® Core™ i5-3230M, 4GB, 1TB, Gravador de ...

DQL – Select com 3 tabelas

Selecionando registros de três tabelas:

```
use loja;
```

```
desc loja.produto;
```

```
desc loja.categoria;
```

```
desc loja.departamento;
```

```
select cod_produto, nom_produto, nom_categoria,  
       nom_departamento from produto as prod, categoria as  
       cat, departamento as dep where  
       prod.cod_Categoria=cat.COD_CATEGORIA and  
       cat.cod_departamento=dep.COD_DEPARTAMENTO;
```



DQL – Funções de **Agregação**

- Segue algumas das funções de agregação:

COUNT	Retornar a quantidade de registros existentes
AVG	Calcular a média dos valores de um campo determinado
SUM	Calcular a soma dos valores de um determinado campo
MAX	Retornar o maior valor existente para um campo especificado
MIN	Retornar o menor valor existente para um campo especificado

- Todas essas funções são aplicadas apenas aos registros que atenderem os critérios estabelecidos na cláusula WHERE da query onde estiverem sendo usadas. Fonte: <http://www.bosontreinamentos.com.br/mysql/mysql-funcoes-de-agregacao-max-min-avg-count-sum-18/>

DQL – Funções de Agregação

- Vamos iniciar contando os registros da nossa tabela, sem estabelecer filtros na clausula WHERE:

-- Contar os registros da tabela PRODUTO

```
SELECT COUNT(*) FROM PRODUTO;
```

COUNT(*)
25

DQL – Funções de Agregação

- Utilizando a Função de Agregação AVG, vamos consultar a Média dos preços dos produtos:

-- Verificar a Média de preço para todos os produtos

```
SELECT AVG(PCO_PRODUTO) FROM PRODUTO;
```

AVG(PCO_PRODUTO)
1353.400400

DQL – Funções de Agregação

- Neste comando vamos aplicar a mesma função, porém estabelecendo um filtro na clausula **WHERE**:

-- Verificar a Média de preço para todos os produtos acima de 2.000,00

```
SELECT AVG(PCO_PRODUTO)
FROM PRODUTO
WHERE PCO_PRODUTO > 2000;
```

Pode ser >, <, >=, <=, =

AVG(PCO_PRODUTO)
2775.488333

DQL – Funções de Agregação

- Neste comando vamos aplicar a mesma função, porém estabelecendo um filtro na cláusula WHERE:

-- Verificar a Média de preço para todos os produtos acima de 2.000,00

```
SELECT AVG(PCO_PRODUTO)
FROM PRODUTO
WHERE PCO_PRODUTO > 2000 AND EST_PRODUTO = 0;
```

Pode ser >, <, >=, <=, =

AVG(PCO_PRODUTO)
NULL

DQL – Funções de Agregação

- Podemos somar os valores de preço dos produtos; para isso vamos utilizar a função SUM:

-- Somar os valores de preço dos produtos

```
SELECT SUM(PCO_PRODUTO)  
FROM PRODUTO;
```

SUM(PCO_PRODUTO)
33835.01

DQL – Funções de Agregação

- Podemos verificar qual é o maior preço; para isso vamos utilizar a função MAX:

--Selecionar o maior valor do campo PCO_PRODUTO na tabela produto = mostrar o produto mais caro!

```
SELECT MAX(PCO_PRODUTO)
FROM PRODUTO;
```

MAX(PCO_PRODUTO)
3959.00

DQL – Funções de Agregação

- Podemos verificar qual é o menor preço; para isso vamos utilizar a função MIN:

--Selecionar o menor valor do campo PCO_PRODUTO na tabela produto = mostrar o produto mais barato!

```
SELECT MIN(PCO_PRODUTO)
FROM PRODUTO;
```

MIN(PCO_PRODUTO)
27,30

DQL – GROUP BY

- A cláusula **GROUP BY** tem por função agrupar um conjunto de registros que possuam características idênticas, apresentando um resultado sintético das informações encontradas, ou seja, retorna em uma linha para cada grupo, reduzindo o número de linhas no conjunto de resultados.

```
SELECT COD_CATEGORIA  
       , DSC_PRODUTO  
FROM PRODUTO  
GROUP BY COD_CATEGORIA;
```

COD_CATEGORIA	DSC_PRODUTO
2	Ultrabook Acer Aspire M5-481T-6417 com Intel® Core™ i5-3317U, 6GB, 500GB, 20GB SSD, Gravador de ...
3	Notebook Acer Aspire E1-572-6_BR648 com Intel® Core™ i5-4200U, 6GB, 500GB, Gravador de DVD, Leit...
5	Computador HP 402 G1 SFF com Intel Core i3-4130, 4GB, 500GB, Gravador de DVD e FreeDOS 2.0
8	Impressora Monocromática Laser - Samsung ML-2165
10	Tablet Samsung Galaxy Tab 3 Lite SMT111M Branco com Tela 7", Wi-Fi, 3G, 8GB, Processador Dual Core ...
14	Pen Drive SanDisk Cruzer Blade 16GB

Fonte: <https://dev.mysql.com/doc/refman/8.0/en/group-by-modifiers.html>

DQL – GROUP BY

- A cláusula **GROUP BY** tem por função agrupar um conjunto de registros que possuam características idênticas, apresentando um resultado sintético das informações encontradas, ou seja, retorna em uma linha para cada grupo, reduzindo o número de linhas no conjunto de resultados.

```
SET GLOBAL sql_mode=(SELECT  
  REPLACE (@@sql_mode, 'ONLY_FULL_GROUP_BY', '')) ;
```

```
SELECT COD_CATEGORIA  
      , DSC_PRODUTO  
FROM PRODUTO  
GROUP BY COD_CATEGORIA;
```

COD_CATEGORIA	DSC_PRODUTO
2	Ultrabook Acer Aspire M5-481T-6417 com Intel® Core™ i5-3317U, 6GB, 500GB, 20GB SSD, Gravador de ...
3	Notebook Acer Aspire E1-572-6_BR648 com Intel® Core™ i5-4200U, 6GB, 500GB, Gravador de DVD, Leit...
5	Computador HP 402 G1 SFF com Intel Core i3-4130, 4GB, 500GB, Gravador de DVD e FreeDOS 2.0
8	Impressora Monocromática Laser - Samsung ML-2165
10	Tablet Samsung Galaxy Tab 3 Lite SMT111M Branco com Tela 7", Wi-Fi, 3G, 8GB, Processador Dual Core ...
14	Pen Drive SanDisk Cruzer Blade 16GB

Fonte: <https://dev.mysql.com/doc/refman/8.0/en/group-by-modifiers.html>

DQL – GROUP BY

- A GROUP BY deve aparecer após as cláusulas FROM e WHERE; esta, por sua vez, admite uma lista de colunas ou expressões, separadas por vírgula, como critério para o agrupamento.
- Comparativamente, a cláusula GROUP BY é semelhante ao operador DISTINCT:

```
SELECT DISTINCT COD_CATEGORIA  
FROM PRODUTO;
```

OU

```
SELECT COD_CATEGORIA  
FROM PRODUTO  
GROUP BY COD_CATEGORIA;
```

COD_CATEGORIA
2
3
5
8
10
14

Fonte: <https://dev.mysql.com/doc/refman/8.0/en/group-by-modifiers.html>

DQL – GROUP BY

- Podemos também utilizar a cláusula GROUP BY com funções agregadas (SUM, AVG, MAX, MIN, e COUNT).
- Exemplo: estou contando quantos produtos tem por categoria.

```
SELECT COD_CATEGORIA, COUNT(*)  
      FROM PRODUTO  
      GROUP BY COD_CATEGORIA;
```

COD_CATEGORIA	COUNT(*)
2	4
3	5
5	4
8	4
10	4
14	4


Fonte: <https://dev.mysql.com/doc/refman/8.0/en/group-by-modifiers.html>

DQL – GROUP BY

- Para sabermos a qual categoria e produto pertence porque até agora só temos código: Tenho que fazer o select em duas tabelas

```
SELECT PROD.COD_CATEGORIA
      , CAT.NOM_CATEGORIA
      , PROD.NOM_PRODUTO
      , COUNT (*)
FROM PRODUTO AS PROD, CATEGORIA AS CAT
WHERE PROD.COD_CATEGORIA = CAT.COD_CATEGORIA
GROUP BY PROD.COD_CATEGORIA;
```

COD_CATEGORIA	COUNT(*)
2	4
3	5
5	4
8	4
10	4
14	4



COD_CATEGORIA	NOM_CATEGORIA	NOM_PRODUTO	COUNT
2	ULTRABOOK	Ultrabook Acer Aspire M5-481T-6650 com Intel...	4
5	COMPUTADORES	Computador HP 402 G1 SFF com Intel Core i3-4...	4
3	NOTEBOOK	Notebook Acer Aspire E1-572-6_BR648 com Int...	5
8	IMPRESSORAS	Impressora Monocromática Laser - Samsung ML...	4
14	PEN DRIVES	Pen Drive SanDisk Cruzer Blade 16GB	4
10	IPOS E ACESSÓRIOS	Tablet Samsung Galaxy Tab 3 Lite SMT111M Bra...	4

Fonte: <https://dev.mysql.com/doc/refman/8.0/en/group-by-modifiers.html>

DQL – HAVING

- Nem sempre queremos realizar agrupamentos em todos os dados em uma determinada tabela. Neste caso podemos utilizar a cláusula **HAVING** para filtrar os resultados dos grupos formados pela consulta.
- É utilizado quando se deseja o filtro depois do agrupamento e não antes.

```
SELECT PROD.COD_CATEGORIA , CAT.NOM_CATEGORIA
      , PROD.NOM_PRODUTO
      , COUNT (*)
FROM PRODUTO AS PROD,
      CATEGORIA AS CAT
WHERE PROD.COD_CATEGORIA = CAT.COD_CATEGORIA
GROUP BY PROD.COD_CATEGORIA
HAVING PROD.COD_CATEGORIA <> 10;
```

COD_CATEGORIA	NOM_CATEGORIA	NOM_PRODUTO	COUNT(*)
2	ULTRABOOK	Ultrabook Acer Aspire M5-481T-6650 com Intel...	4
5	COMPUTADORES	Computador HP 402 G1 SFF com Intel Core i3-4...	4
3	NOTEBOOK	Notebook Acer Aspire E1-572-6_BR648 com Int...	5
8	IMPRESSORAS	Impressora Monocromática Laser - Samsung ML...	4
14	PEN DRIVES	Pen Drive SanDisk Cruzer Blade 16GB	4
10	IPOS E ACESSÓRIOS	Tablet Samsung Galaxy Tab 3 Lite SMT111M Bra...	4

COD_CATEGORIA	NOM_CATEGORIA	NOM_PRODUTO	COUNT(*)
2	ULTRABOOK	Ultrabook Acer Aspire M5-481T-6650 com Intel...	4
5	COMPUTADORES	Computador HP 402 G1 SFF com Intel Core i3-4...	4
3	NOTEBOOK	Notebook Acer Aspire E1-572-6_BR648 com Int...	5
8	IMPRESSORAS	Impressora Monocromática Laser - Samsung ML...	4
14	PEN DRIVES	Pen Drive SanDisk Cruzer Blade 16GB	4

DQL – GROUP BY

- Sem Group By



- Com Group By



Tipos de Joins

SQL JOINS

INNER JOIN

LEFT [outer] JOIN

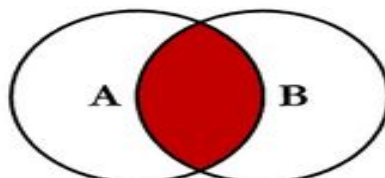


```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```

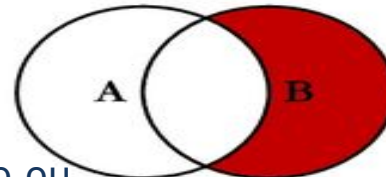
RIGHT [outer] JOIN



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```

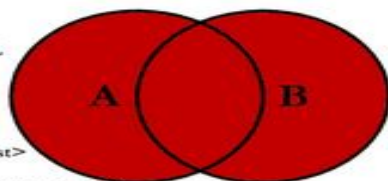


```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



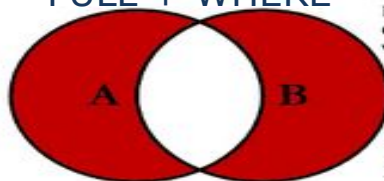
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```

FULL [outer] JOIN



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```

FULL Exclusivo ou
FULL + WHERE



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

© C.L. Moffitt, 2008

RIGHT
Exclusivo ou
RIGHT +
WHERE

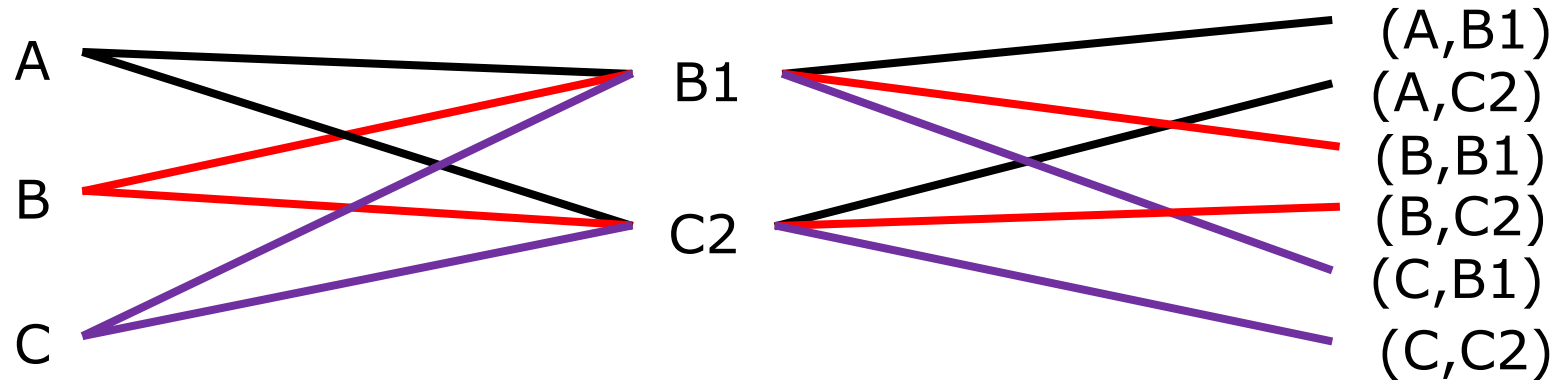
Junção Cruzada

CROSS JOIN

Cross Join – Junção cruzada

- Tipo mais simples de junção.
 - **Geralmente o menos utilizado.**
- Executa o **produto cartesiano** de duas tabelas.
 - Resultado: Uma linha para cada combinação de linhas das duas tabelas.
 - Se existem m linhas em uma tabela T1 e n linhas em uma tabela T2, o resultado de uma junção cruzada entre T1 e T2 é uma tabela virtual com $m \times n$ linhas.

Cross Join – Junção cruzada



Cross Join – Junção cruzada

DIA_SEMANA

<u>Dia_da_semana</u>	Dia_util
Segunda-feira	1
Terça-feira	1
Quarta-feira	1
Quinta-feira	1
Sexta-feira	1
Sábado	0
Domingo	0

7 linhas

PERIODO_DIA

<u>Periodo_dia</u>	Observacao
Manhã	Sono
Tarde	Preguiça
Noite	Sem sono

3 linhas

Resultado: 21 linhas (7 * 3)

<u>Dia da semana</u>	<u>Dia_util</u>	<u>Periodo_dia</u>	<u>Observacao</u>	<u>Linhas</u>
Segunda-feira	1	Manhã	Sono	1
Segunda-feira	1	Tarde	Preguiça	2
Segunda-feira	1	Noite	Sem sono	3
Terça-feira	1	Manhã	Sono	4
Terça-feira	1	Tarde	Preguiça	5
Terça-feira	1	Noite	Sem sono	6
Quarta-feira	1	Manhã	Sono	7
Quarta-feira	1	Tarde	Preguiça	8
Quarta-feira	1	Noite	Sem sono	9
Quinta-feira	1	Manhã	Sono	10
Quinta-feira	1	Tarde	Preguiça	11
Quinta-feira	1	Noite	Sem sono	12
Sexta-feira	1	Manhã	Sono	13
Sexta-feira	1	Tarde	Preguiça	14
Sexta-feira	1	Noite	Sem sono	15
Sábado	0	Manhã	Sono	16
Sábado	0	Tarde	Preguiça	17
Sábado	0	Noite	Sem sono	18
Domingo	0	Manhã	Sono	19
Domingo	0	Tarde	Preguiça	20
Domingo	0	Noite	Sem sono	21

Junção Interna
INNER JOIN = INTERSECÇÃO DE
CONJUNTOS = REGISTROS
IGUAIS NAS DUAS TABELAS



Inner Join – Junção interna

- Com uma junção interna você faz a correspondência de duas tabelas com base sobre um predicado.
 - Predicado geralmente compara um valor de chave primária em um lado com uma chave estrangeira no outro lado.
 - O predicado de correspondência é especificado na cláusula **ON**

Select colunas

from tabela 1

[INNER] JOIN tabela 2

on tabela1.coluna = tabela2.coluna

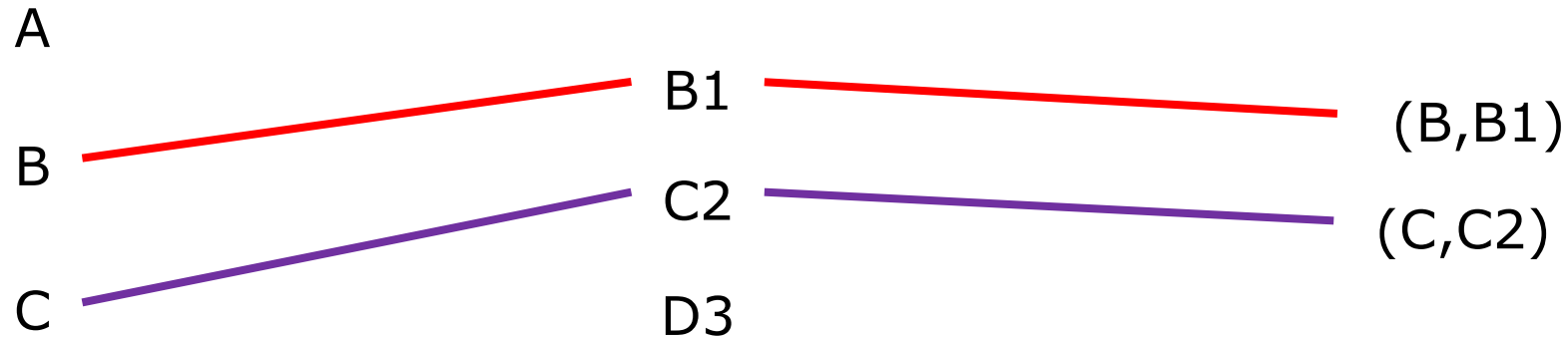
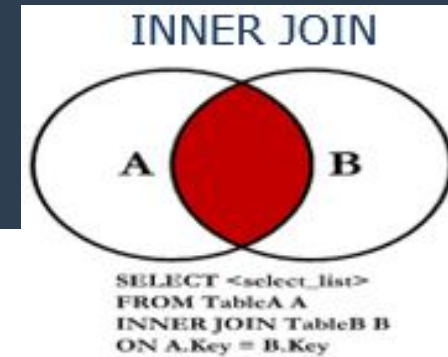
[INNER] JOIN tabela 3

on tabela1.coluna = tabela3.coluna

[where condição];



Inner Join – Junção interna



Predicado: Letra do elemento do 1º conjunto = Primeira letra do elemento do 2º conjunto

Inner Join – Junção interna

CARRO

Registro	Modelo	Cor
AAA1234B	Gol	Preta
ABC9999C	Corsa	Prata
HBC4789D	Palio	Vermelho
ASD9876Q	Uno	Azul

PESSOA

CPF	Nome	Reg_carro
123456789	João	AAA1234B
333444555	Pedro	HBC4789D
987654321	Ana	ASD9876Q

Para cada pessoa, liste o nome e a cor do carro dessa pessoa.

Inner Join – Junção interna

CARRO			PESSOA		
Registro	Modelo	Cor	CPF	Nome	Reg_carro
AAA1234B	Gol	Preta	123456789	João	AAA1234B
ABC9999C	Corsa	Prata	333444555	Pedro	HBC4789D
HBC4789D	Palio	Vermelho	987654321	Ana	ASD9876Q
ASD9876Q	Uno	Azul			

Para cada pessoa, liste o nome e a cor do carro dessa pessoa.

```
SELECT Nome, Cor  
FROM PESSOA INNER JOIN CARRO ON (Reg_carro = Registro);
```

Inner Join – Junção interna

```
SELECT FAB.NOM_FABRICANTE  
       , PROD.NOM_PRODUTO  
FROM FABRICANTE AS FAB  
INNER JOIN PRODUTO AS PROD ON  
       FAB.COD_FABRICANTE = PROD.COD_FABRICANTE  
ORDER BY NOM_FABRICANTE;
```

NOM_FABRICANTE	NOM_PRODUTO
ACER	Notebook Acer Aspire E1-572-6_BR648 com Intel® Core™ i5-42...
APPLE	iPad Air com Tela Retina Apple Wi-Fi com 32GB, Bluetooth 4.0, C...
APPLE	iPad Air com Tela Retina Apple Wi-Fi + 3G/4G* com 32GB, Blueto...
ASUS	Notebook Asus X551MA-BRAL-SX207H com Intel® Quad Core, 4...
DELL	Ultrabook Acer Aspire M5-481T-6650 com Intel® Core™ i3-3227U
DELL	Ultrabook Acer Aspire M5-481T-6417 com Intel® Core™ i5-3317U

Inner Join – Junção interna

```
select cod_produto, nom_produto, nom_categoria,  
       nom_fabricante  
from produto as prod  
inner join categoria as cat  
on prod.cod_Categoria = cat.COD_CATEGORIA  
inner join fabricante as fab on  
   prod.cod_fabricante=fab.COD_fabricante;
```

	cod_produto	nom_produto	nom_categoria	nom_fabricante
►	1	Ultrabook Acer Aspire M5-481T-6650 com Intel...	ULTRABOOK	DELL
	2	Ultrabook Acer Aspire M5-481T-6417 com Intel...	ULTRABOOK	DELL
	3	Ultrabook Acer Aspire M5-481T-6195 com Intel...	ULTRABOOK	DELL
	4	Ultrabook Acer Aspire com Intel® Core™ i5	ULTRABOOK	DELL
	5	Computador HP 402 G1 SFF com Intel Core i3-4...	COMPUTADORES	HP

Inner Join – Junção interna

CARRO


Registro	Modelo	Cor
AAA1234B	Gol	Preta
ABC9999C	Corsa	Prata
HBC4789D	Palio	Vermelho
ASD9876Q	Uno	Azul

PESSOA

CPF	Nome	Reg_carro
123456789	João	AAA1234B
333444555	Pedro	HBC4789D
987654321	Ana	ASD9876Q

Quais são os nomes das pessoas que tem um carro da cor preta?

Inner Join – Junção interna



CARRO			PESSOA		
Registro	Modelo	Cor	CPF	Nome	Reg_carro
AAA1234B	Gol	Preta	123456789	João	AAA1234B
ABC9999C	Corsa	Prata	333444555	Pedro	HBC4789D
HBC4789D	Palio	Vermelho	987654321	Ana	ASD9876Q
ASD9876Q	Uno	Azul			

Quais são os nomes das pessoas que tem um carro da cor preta?

```
SELECT  Nome, Cor
FROM    PESSOA INNER JOIN CARRO ON (Reg_carro = Registro)
WHERE   Cor = 'Preta';
```

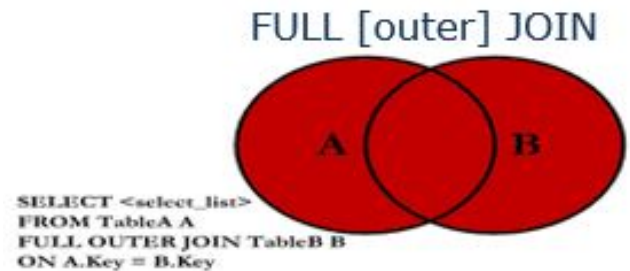
DML – Select com 2 tabelas

Exemplo

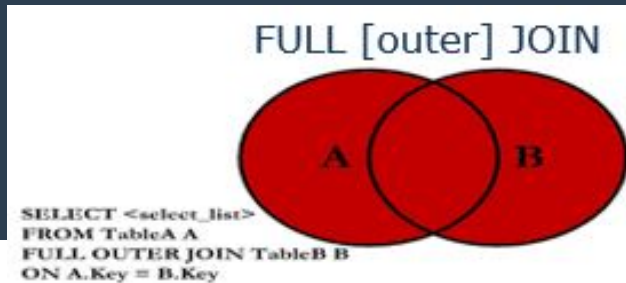
```
SELECT DPT.COD_DEPARTAMENTO
      ,DPT.NOM_DEPARTAMENTO
      ,DPT.IDF_ATIVO
      ,DPT.COD_GRUPO
      ,GRU.NOM_GRUPO
FROM DEPARTAMENTO DPT
JOIN GRUPO GRU ON DPT.COD_GRUPO = GRU.COD_GRUPO
WHERE DPT.COD_DEPARTAMENTO = 1;
```

COD_DEPARTAMENTO	NOM_DEPARTAMENTO	IDF_ATIVO	COD_GRUPO	NOM_GRUPO
1	INFORMATICA	S	2	INFORMÁTICA E TECNOLOGIA

Outer JOINS = união



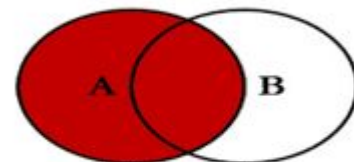
DML – Select com 2 tabelas



- Em alguns casos, podemos **não ter um registro** relacionado em **outra tabela** mas, mesmo assim, **gostaríamos** de listar da tabela principal.
- Para resolver este problemas temos o OUTER JOIN.

DML – Select com 2 tabelas

LEFT [outer] JOIN

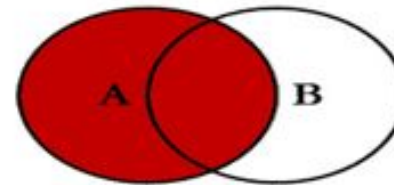


```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```

- Considerando o sinal de igualdade “=”, quando temos dados na tabela do lado esquerdo e podemos não ter do lado direito, usamos **LEFT JOIN** e, o oposto usamos **RIGHT JOIN**.

LEFT OUTER JOIN

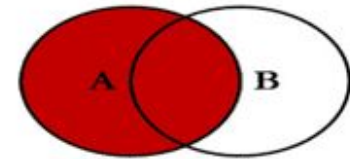
LEFT [outer] JOIN



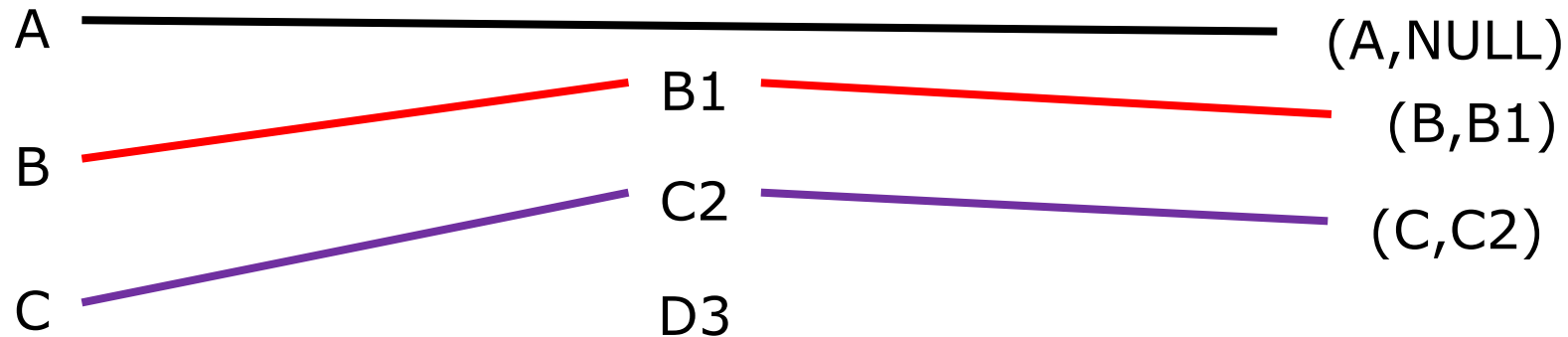
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```

LEFT OUTER JOIN – Junção externa à esquerda

LEFT [outer] JOIN



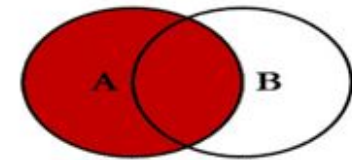
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



Predicado: Letra do elemento do 1º conjunto = Primeira letra do elemento do 2º conjunto

LEFT OUTER JOIN – Junção externa à esquerda

LEFT [outer] JOIN



```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```

CARRO

Registro	Modelo	Cor
AAA1234B	Gol	Preta
ABC9999C	Corsa	Prata
HBC4789D	Palio	Vermelho
ASD9876Q	Uno	Azul

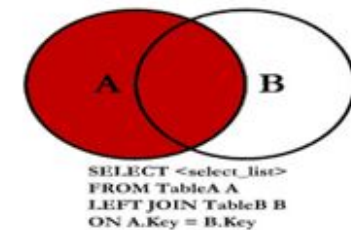
PESSOA

CPF	Nome	Reg_carro
123456789	João	AAA1234B
333444555	Pedro	HBC4789D
987654321	Ana	ASD9876Q

Para todos os carros, devolva o modelo do carro e o nome do dono, se esse tiver dono.

LEFT OUTER JOIN – Junção externa à esquerda

LEFT [outer] JOIN



CARRO

Registro	Modelo	Cor
AAA1234B	Gol	Preta
ABC9999C	Corsa	Prata
HBC4789D	Palio	Vermelho
ASD9876Q	Uno	Azul

PESSOA

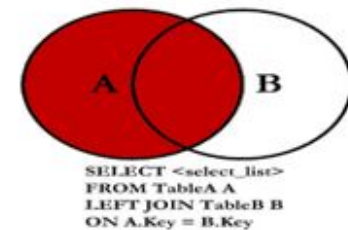
CPF	Nome	Reg_carro
123456789	João	AAA1234B
333444555	Pedro	HBC4789D
987654321	Ana	ASD9876Q

Para todos os carros, devolva o modelo do carro e o nome do dono, se esse tiver dono.

```
SELECT Modelo, Nome  
FROM CARRO LEFT JOIN PESSOA ON (Registro = Reg_carro);
```

LEFT OUTER JOIN – Junção externa à esquerda

LEFT [outer] JOIN



CARRO

Registro	Modelo	Cor
AAA1234B	Gol	Preta
ABC9999C	Corsa	Prata
HBC4789D	Palio	Vermelho
ASD9876Q	Uno	Azul

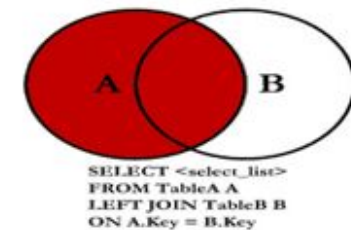
PESSOA

CPF	Nome	Reg_carro
123456789	João	AAA1234B
333444555	Pedro	HBC4789D
987654321	Ana	ASD9876Q

Registro	Modelo	Cor	CPF	Nome	Reg_carro
AAA1234B	Gol	Preta	123456789	João	AAA1234B
ABC9999C	Corsa	Prata	NULL	NULL	NULL
HBC4789D	Palio	Vermelho	333444555	Pedro	HBC4789D
ASD9876Q	Uno	Azul	987654321	Ana	ASD9876Q

DML – Select com 2 tabelas

LEFT [outer] JOIN



Exemplo

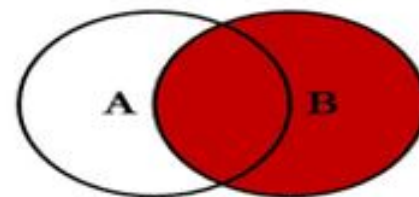
```
SELECT FAB.NOM_FABRICANTE  
       , PROD.NOM_PRODUTO  
FROM FABRICANTE FAB  
LEFT JOIN PRODUTO PROD ON  
          FAB.COD_FABRICANTE = PROD.COD_FABRICANTE  
ORDER BY NOM_FABRICANTE;
```

NOM_FABRICANTE	NOM_PRODUTO
SAMSUNG	Tablet Samsung Galaxy Tab 4 7.0 com Tela 7", com TV Digital, 8...
SANDISK	Pen Drive SanDisk Cruzer Blade 16GB
SANDISK	Pen Drive SanDisk Ultra® Dual Drive com USB 2.0 e Micro-USB 3...
SEMP-TOSHIBA	NULL
SONY	NULL



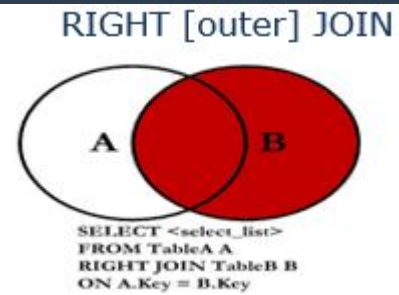
RIGHT OUTER JOIN

RIGHT [outer] JOIN



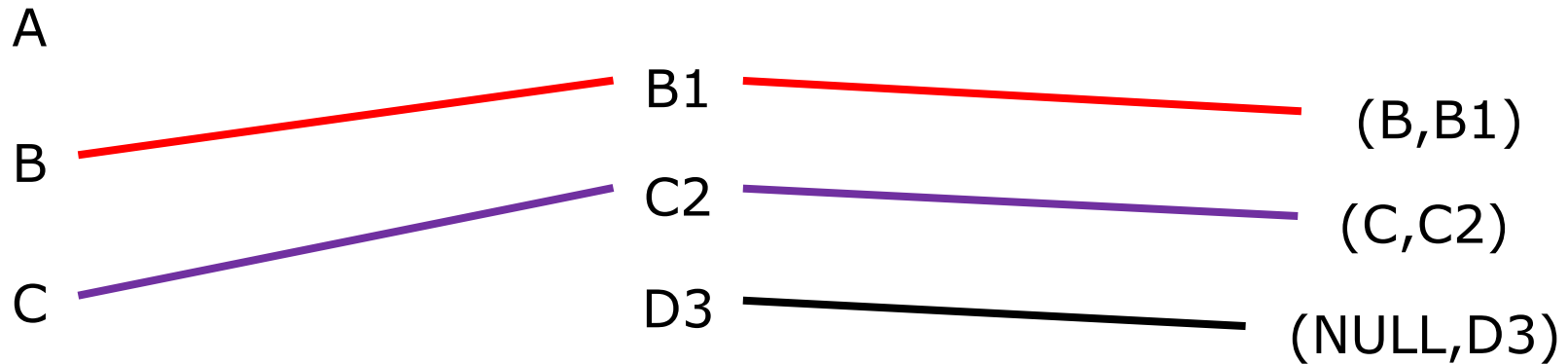
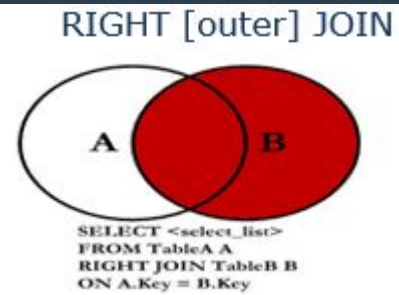
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```

Outers Join – Junções externas



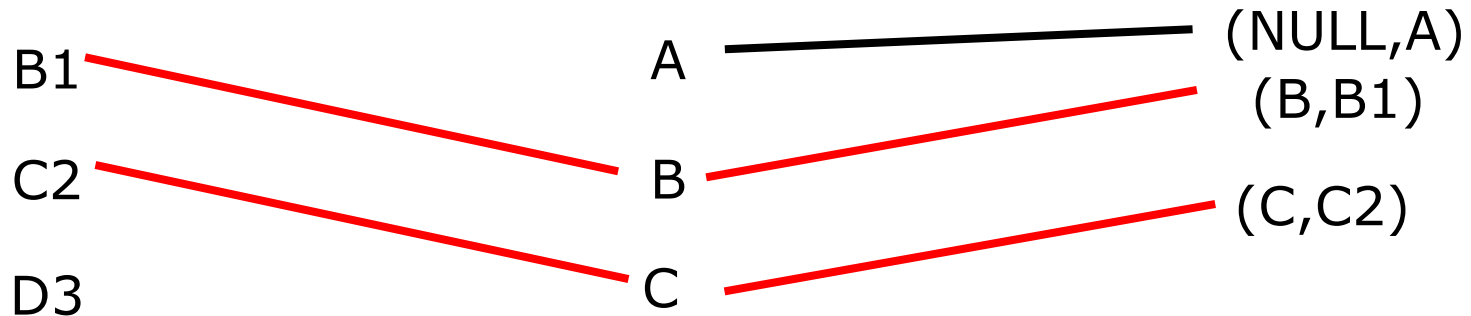
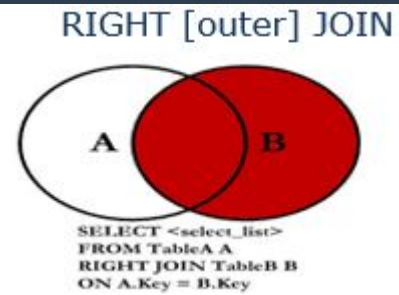
- Usando **RIGHT OUTER JOIN** (ou simplesmente **RIGHT JOIN**):
 - Preserva a tabela da direita.
 - A junção devolve o que a junção interna devolve mais as linhas da tabela da direita que não possuem correspondência na tabela da esquerda.

RIGHT OUTER JOIN – Junção externa à direita



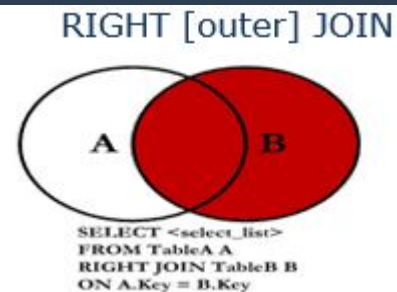
Predicado: Letra do elemento do 1º conjunto = Primeira letra do elemento do 2º conjunto

RIGHT OUTER JOIN – Junção externa à direita



Predicado: Letra do elemento do 1º conjunto = Primeira letra do elemento do 2º conjunto

RIGHT OUTER JOIN – Junção externa à direita



CARRO

Registro	Modelo	Cor
AAA1234B	Gol	Preta
ABC9999C	Corsa	Prata
HBC4789D	Palio	Vermelho
ASD9876Q	Uno	Azul

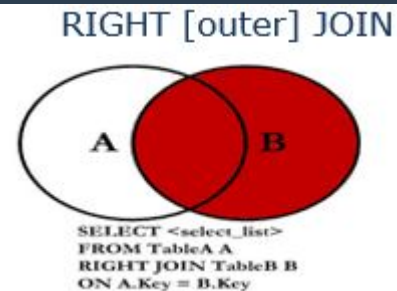
PESSOA

CPF	Nome	Reg_carro
123456789	João	AAA1234B
333444555	Pedro	HBC4789D
987654321	Ana	ASD9876Q

Para todos os carros, devolva o modelo do carro e o nome do dono, se esse tiver dono.

```
SELECT Modelo, Nome  
FROM PESSOA RIGHT JOIN CARRO ON (Reg_carro = Registro);
```

DML – Select com 2 tabelas



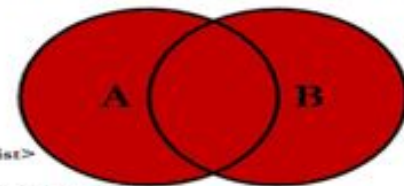
Exemplo

```
SELECT FAB.NOM_FABRICANTE  
       , PROD.NOM_PRODUTO  
FROM FABRICANTE FAB  
RIGHT JOIN PRODUTO PROD ON  
           FAB.COD_FABRICANTE = PROD.COD_FABRICANTE  
ORDER BY NOM_FABRICANTE;
```

Na nossa loja, não existe esta situação, de haver produto sem fabricante, logo o resultado a tabela de produto inteira sem nenhum NULL na tabela fabricante.

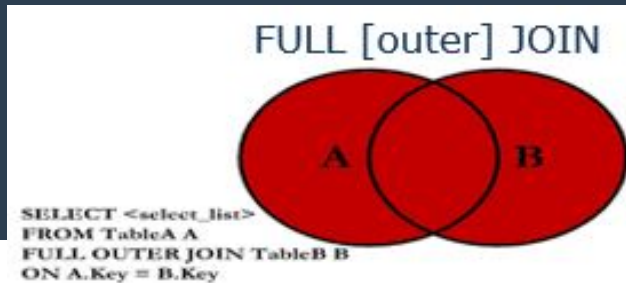
FULL OUTER JOIN

FULL [outer] JOIN



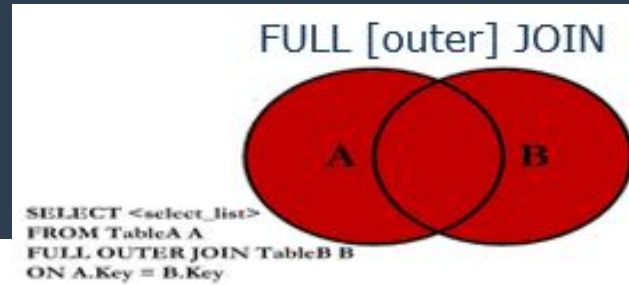
```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```

DML – Select com 2 tabelas



- Em algumas situações, as junções podem trazer linhas repetidas.
- Isso também pode acontecer por falta de uma junção.
- É o chamado **PLANO CARTESIANO**, onde todas as linhas de uma tabela são relacionadas com todas as linhas das demais tabelas.

Outers Join – Junções externas



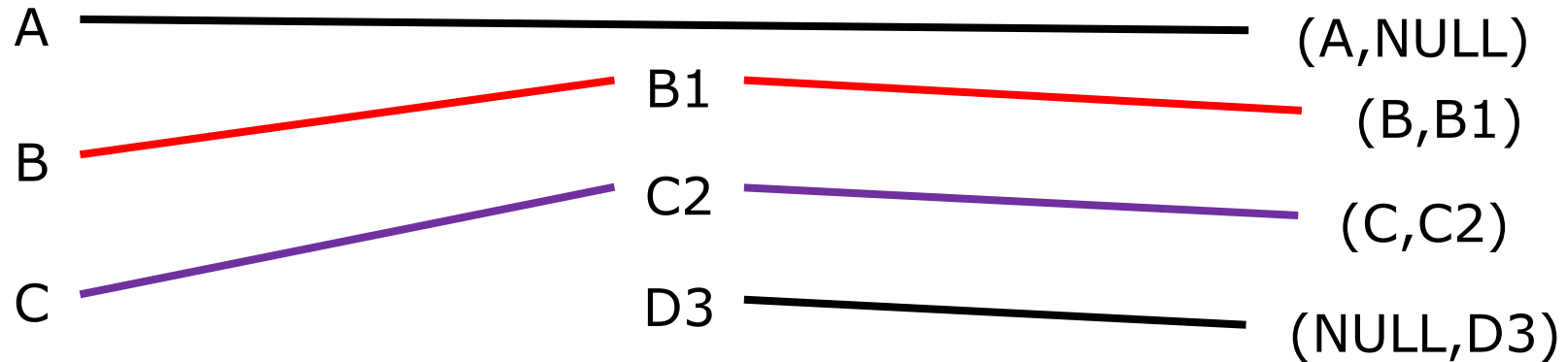
- Usando **FULL OUTER JOIN** (ou simplesmente **FULL JOIN**):
 - Preserva as duas tabelas.
 - A junção devolve o que a junção interna devolve mais:
 - As linhas da tabela da esquerda que não possuem correspondência na tabela da direita.
 - As linhas da tabela da direita que não possuem correspondência na tabela da esquerda.

FULL OUTER JOIN – Junção externa completa

FULL [outer] JOIN



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



Predicado: Letra do elemento do 1º conjunto = Primeira letra do elemento do 2º conjunto

FULL OUTER JOIN – Junção externa completa

FULL [outer] JOIN



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```

CARRO

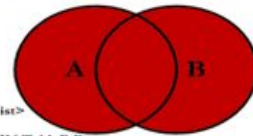
Registro	Modelo	Cor
AAA1234B	Gol	Preta
ABC9999C	Corsa	Prata
HBC4789D	Palio	Vermelho
ASD9876Q	Uno	Azul

PESSOA

CPF	Nome	Reg_carro
123456789	João	AAA1234B
333444555	Pedro	HBC4789D
987654321	Ana	ASD9876Q
111222333	Maria	NULL

Faça um relatório contendo todos os modelos de carro e todas as pessoas.
Deve ser devolvido o nome da pessoa e o modelo do carro. Devem ser listados todas as pessoas e todos os modelos.

```
SELECT Modelo, Nome  
FROM PESSOA FULL JOIN CARRO ON (Reg_carro = Registro);
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```

FULL OUTER JOIN – Junção externa completa

CARRO

Registro	Modelo	Cor
AAA1234B	Gol	Preta
ABC9999C	Corsa	Prata
HBC4789D	Palio	Vermelho
ASD9876Q	Uno	Azul

PESSOA

CPF	Nome	Reg_carro
123456789	João	AAA1234B
333444555	Pedro	HBC4789D
987654321	Ana	ASD9876Q
111222333	Maria	NULL

CPF	Nome	Reg_carro	Registro	Modelo	Cor
123456789	João	AAA1234B	AAA1234B	Gol	Preta
333444555	Pedro	HBC4789D	HBC4789D	Palio	Vermelho
987654321	Ana	ASD9876Q	ASD9876Q	Uno	Azul
111222333	Maria	NULL	NULL	NULL	NULL
NULL	NULL	NULL	ABC9999C	Corsa	Prata

Múltiplas Junções

- Conceitualmente a junção entre duas tabelas é feita entre duas tabelas.
- Com junções múltiplas: avaliada da esquerda para a direita.

Múltiplas Junções

- Para todo empregado que trabalha no projeto 'Produto Y', faça um relatório contendo:
 - Nome do empregado, completo em uma coluna.
 - Número de horas semanais trabalhada no projeto.
 - Parcela do salário devido ao projeto, isso é, $40 / \text{horas trabalhadas} * \text{Salario}$.
 - Caso tenha, o nome de seus dependentes e o relacionamento com o empregado.

Tratando campos com valor NULL

DML – Pesquisa (Select)

NVL() - Oracle

Retorna um valor (especificado pelo programador) se o valor em teste for NULL.

Exemplos

```
1  SELECT NOM_PROGRAMA
2         , NVL(IDF_ATIVO_SN, 'N') IDF_ATIVO_SN
3  FROM PROGRAMA;
4
5  SELECT NOM_PRODUTO
6         , NVL(PCO_PRODUTO, 0) PCO_PRODUTO
7  FROM PRODUTO
```


DML – Pesquisa (Select)

IFNULL() - MySQL

Retorna um valor (especificado pelo programador) se o valor em teste for NULL.

Exemplos

```
1  SELECT NOM_PROGRAMA
2      , IFNULL(IDF_ATIVO_SN, 'N') IDF_ATIVO_SN
3  FROM PROGRAMA;
4
5  SELECT NOM_PRODUTO
6      , IFNULL(PCO_PRODUTO, 0) PCO_PRODUTO
7  FROM PRODUTO
```

DML – Pesquisa (Select)

COALESCE() - Firebird / PostgreSQL

Retorna um valor (especificado pelo programador) se o valor em teste for NULL.

Exemplos

```
1 SELECT NOM_PROGRAMA
2       , COALESCE (IDF_ATIVO_SN, 'N') IDF_ATIVO_SN
3 FROM PROGRAMA;
4
5 SELECT NOM_PRODUTO
6       , COALESCE (PCO_PRODUTO, 0) PCO_PRODUTO
7 FROM PRODUTO
```

Seleção

DML – Pesquisa (Select)

DECODE – Oracle e Firebird

Funciona como um comando **SE..ENTÃO..SENÃO** em linha
Sintaxe

```
DECODE (VALOR, OPC1, VLR1, [VLR2 | OPCn, VLRn])
```

Exemplos

```
DECODE (IDF_ATIVO, 'S', 'SIM', 'NÃO')
```

```
DECODE (EST_CIVIL, 'C', CASADO, 'V', VIUVO, 'SOLTEIRO')
```

DML – Pesquisa (Select)

IF – MySQL ou IIF – Firebird

Funciona como um comando SE..ENTÃO..SENÃO em linha, entretanto, permite apenas uma comparação, sendo necessário fazer outro IF/IIF neles. Assemelha-se ao SE() do EXCEL.

Sintaxe: `IF (VALOR=OPC, VLR1, VLR2)`

`IIF (VALOR=OPC, VLR1, VLR2)`

Exemplos

`IF (IDF_ATIVO='S', 'SIM', 'NÃO')`

`IIF (IDF_ATIVO='S', 'SIM', 'NÃO')`

DML – Pesquisa (Select)

IF – MySQL ou IIF – Firebird

Funciona como um comando SE..ENTÃO..SENÃO em linha, entretanto, permite apenas uma comparação, sendo necessário fazer outro IF/IIF neles. Assemelha-se ao SE() do EXCEL.

Sintaxe: IF (VALOR=OPC, VLR1, VLR2)

IIF (VALOR=OPC, VLR1, VLR2)

Exemplos

```
IF (IDF_ATIVO='S', 'SIM', IF (IDF_ATIVO='N', 'NÃO', 'XXX'))
```

```
IIF (IDF_ATIVO='S', 'SIM', 'NÃO')
```

DML – Pesquisa (Select)

CASE (Todos os SGBDs)

Semelhante ao DECODE, porém, utiliza mais linhas.

Sintaxe

```
SELECT CASE valor_em_teste
        WHEN OPC1 THEN VLR1
        WHEN OPC2 THEN VLR2
        (...)
        WHEN OPCn THEN VLRn
        ELSE valor_padrão
        END alias_coluna
FROM TABELA
```

DML – Pesquisa (Select)

```
1  SELECT NOM_FUNCIONARIO
2     , CASE EST_CIVIL
3         WHEN 'C' THEN 'CASADO'
4         WHEN 'D' THEN 'DIVORCIADO'
5         WHEN 'A' THEN 'AMAZIADO'
6         WHEN 'V' THEN 'VIUVO'
7     ELSE 'SOLTEIRO' ESTADO_CIVIL
8  FROM FUNCIONARIO
```


DML – Pesquisa (Select)

```
SELECT COD_DEPARTAMENTO
       ,NOM_DEPARTAMENTO
       ,CASE IDF_ATIVO
           WHEN 'S' THEN 'SIM'
           WHEN 'N' THEN 'NÃO'
       END SITUACAO
FROM DEPARTAMENTO
WHERE COD_GRUPO = 3;
```

COD_DEPARTAMENTO	NOM_DEPARTAMENTO	SITUACAO
7	ELETRODOMÉSTICOS	SIM
8	MÓVEIS	SIM
9	ELETROPORTÁTEIS	SIM
10	UTILIDADES DOMÉSTICAS	SIM
11	CAMA, MESA E BANHO	SIM
12	AR E VENTILAÇÃO	SIM

Prática

Loja

Continuação....

Referências

Referências

- ELMASRI, Ramez; NAVATHE, Shamkant B.. Parte 2: Modelo de Dados Relacional e SQL: SQL Básica. In: ELMASRI, Ramez; NAVATHE, Shamkant B.. **Sistema de Banco de Dados**. 6. ed. São Paulo: Pearson Education, 2011. Cap. 4. p. 57-75.
- MICROSOFT. **Quais são as funções do banco de dados SQL?** 2017. Disponível em: <<https://docs.microsoft.com/pt-br/sql/t-sql/functions/functions?view=sql-server-2017>>. Acesso em: 27 jun. 2019.
- RAMAKRISHNAN, Raghu; GEHRKE, Johannes. SQL, Consultas, Restrições, Gatilhos. In: RAMAKRISHNAN, Raghu; GEHRKE, Johannes. **Sistemas de Gerenciamento de Banco de Dados**. 3. ed. São Paulo: Mc Graw Hill, 2008. Cap. 5. p. 110-147. Tradução da Terceira Edição.

Referências

- SILBERSCHATZ, Abraham; KORTH, Henry F.; SUNDARSHAN, S.. Introdução à SQL. In: SILBERSCHATZ, Abraham; KORTH, Henry F.; SUNDARSHAN, S.. **Sistemas de Banco de Dados**. 5. ed. Rio de Janeiro: Campus, 2006. Cap. 3. p. 37-67.
- W3SCHOOLS. **SQL INSERT INTO Statement**. Disponível em: <https://www.w3schools.com/sql/sql_insert.asp>. Acesso em: 19 jun. 2019.
- _____. **SQL INSERT INTO SELECT Statement**. Disponível em: <https://www.w3schools.com/sql/sql_insert_into_select.asp>. Acesso em: 19 jun. 2019.
- _____. **SQL Server Functions**. Disponível em: <https://www.w3schools.com/sql/sql_ref_sqlserver.asp>. Acesso em: 27 jun. 2019.
- W3SCHOOLS. **SQL SELECT Statement**. Disponível em: <https://www.w3schools.com/sql/sql_select.asp>. Acesso em: 27 jun. 2019.

Obrigado!

Prof. Clóvis Ferraro

(Adaptado de Prof. Dr. Alexandre Rangel)