

SQL DQL: GROUP BY AGRUPAMENTO E AGREGAÇÃO

BANCO DE DADOS

Prof. Luciano Xiscatti

AGRUPAMENTO

- Utilizando o comando **SELECT**, é possível agrupar tuplas por um ou mais atributos;
- O agrupamento é feito por meio da cláusula **GROUP BY**;
- Isso é útil principalmente quando precisamos contar, somar, tirar média *etc.* de um conjunto de registros agrupados;
- Primeiramente vamos ver como funciona o agrupamento:

AGRUPAMENTO

- O agrupamento sempre é feito por um ou mais campos de seleção de registros que podem se repetir em grupos.
- Antes de agrupar, vamos começar verificando todos os registros da tabela **Quarto**, considerando a estrutura a seguir:

Quarto	
PK	NroQuarto: INT
	Andar: TINYINT
	Tipo: CHAR(2)
	Descricao: VARCHAR(200)
	VlrDiaria: DECIMAL(7,2)

```
SELECT *  
FROM Quarto;
```

AGRUPAMENTO

- Considere os registros (hipotéticos) resultantes da consulta:

Quarto	NroQuarto	Andar	Tipo	Descricao	VlrDiaria
	11	1	CL	Casal, luxo, janela	300.00
	12	1	SS	Solteiro, standard, corredor	150.00
	13	1	SL	Solteiro, luxo, corredor	200.00
	14	1	SL	Solteiro, luxo, janela	250.00
	51	5	CL	Casal, luxo, janela	400.00
	52	5	SS	Solteiro, standard, corredor	200.00
	53	5	SL	Solteiro, luxo, corredor	250.00
	54	5	SL	Solteiro, luxo, janela	300.00

AGRUPAMENTO

- A cláusula **GROUP BY** pode, sintaticamente, ser aplicada para qualquer combinação de campos resultantes do **SELECT**, porém, ela só terá efeito se a consulta tiver resultados que tragam apenas registros repetidos;
- Observe as comparações de consulta sobre a tabela

Quarto:

AGRUPAMENTO

- Para: `SELECT * FROM Quarto GROUP BY NroQuarto;`

- O resultado será:

Observe que não há condições de agrupamento, pois cada tupla é diferente

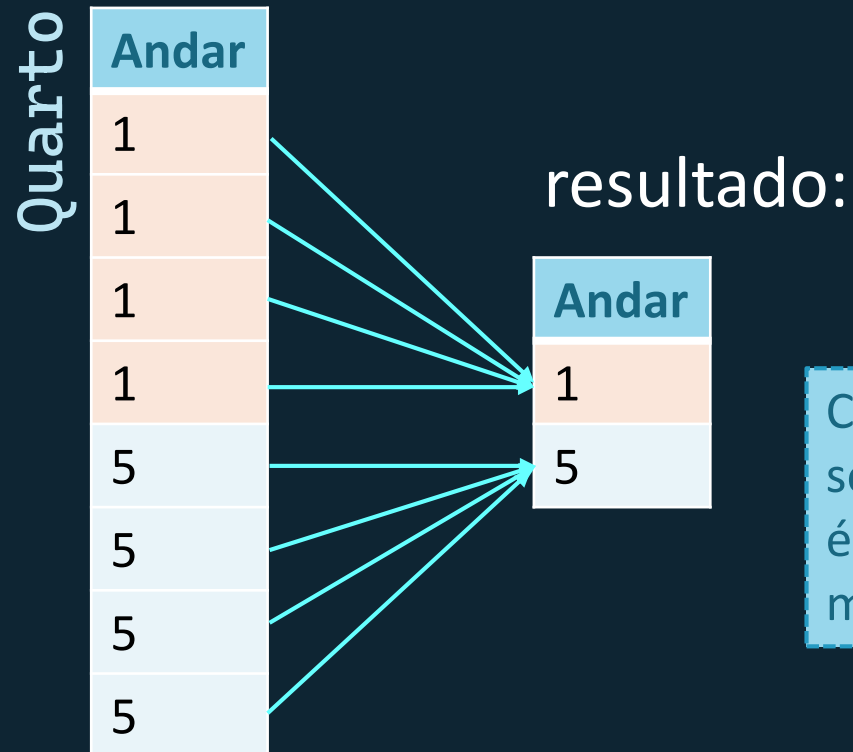
Quarto

NroQuarto	Andar	Tipo	Descricao	VlrDiaria
11	1	CL	Casal, luxo, janela	300.00
12	1	SS	Solteiro, standard, corredor	150.00
13	1	SL	Solteiro, luxo, corredor	200.00
14	1	SL	Solteiro, luxo, janela	250.00
51	5	CL	Casal, luxo, janela	400.00
52	5	SS	Solteiro, standard, corredor	200.00
53	5	SL	Solteiro, luxo, corredor	250.00
54	5	SL	Solteiro, luxo, janela	300.00

AGRUPAMENTO

- Para: `SELECT Andar FROM Quarto GROUP BY Andar;`

- O agrupamento será:



Considerando que foi solicitado somente a coluna **Andar** na consulta, é possível agrupar por ela, e são mostrados os valores sem repetição

AGRUPAMENTO

- Para: `SELECT Andar, Tipo FROM Quarto GROUP BY Andar, Tipo;`

- O agrupamento será:

Quarto

Andar	Tipo
1	CL
1	SS
1	SL
1	SL
5	CL
5	SS
5	SL
5	SL

resultado:

Andar	Tipo
1	CL
1	SS
1	SL
5	CL
5	SS
5	SL

Observe que as linhas de valores exatamente iguais foram agrupadas em uma única linha;
é o caso dos dois quartos de tipo 'SL' em cada um dos andares

FUNÇÕES DE AGREGAÇÃO

- O agrupamento de registros, tem, por si só, sua utilidade;
- Porém, na grande maioria das vezes, o agrupamento é feito sobre um ou mais campos para que sejam aplicadas **funções de agregação** nos campos subsequentes;
- Veja a seguir algumas funções de agregação, primeiramente sem o agrupamento e depois combinadas com agrupamento.

FUNÇÃO COUNT()

- Utilizada para contar registros.
Ex.: “Conte quantos quartos existem no hotel”.

Consulta: `SELECT Count(*) FROM Quarto;`

Resultado:

Count(*)
8

Consulta: `SELECT Count(*) TotQuartos FROM Quarto;`

Resultado:

TotQuartos
8

Aqui foi aplicado um *alias* para o campo resultante da contagem

FUNÇÃO COUNT()

“Conte quantos quartos existem por andar no hotel”

Consulta:

```
SELECT Andar, Count(*) TotQuartos
FROM Quarto
GROUP BY Andar;
```

Resultado:

Andar	TotQuartos
1	4
5	4

Neste caso, a contagem foi agrupada por Andar, isto é: existem 4 registros de quartos com Andar = 1, e 4 registros de quartos com Andar = 5

FUNÇÃO COUNT()

“Conte quantos quartos existem em cada andar, agrupando por tipo”

Consulta:

```
SELECT Andar, Tipo, Count(*) TotQuartos  
FROM Quarto GROUP BY Andar, Tipo;
```

Resultado:

Andar	Tipo	TotQuartos
1	CL	1
1	SS	1
1	SL	2
5	CL	1
5	SS	1
5	SL	2

Todos os campos usados no SELECT, que não são função de agregação, devem ser mencionados no GROUP BY

FUNÇÃO SUM()

- Retorna a soma de valores de campos de um grupo de registros.
Ex.: “Some o valor total de pagamentos feitos hoje”.

Consulta:

```
SELECT Sum(ValorTotal) ValorDeHoje
FROM PagamentoOcupacao
WHERE DataHora = CURDATE();
```

Resultado:

ValorDeHoje

8950.00

PagamentoOcupacao	
PK, FK	NroReserva: INT
FK	IdTipoPagamento: TINYINT DataHora: DATETIME ValorTotal: DECIMAL(10,2)

FUNÇÃO SUM()

“Some o valor total de pagamentos feitos hoje agrupados por andar”.

Consulta:

```
SELECT Q.Andar, Sum(P0.ValorTotal) ValorDeHoje
FROM PagamentoOcupacao P0
INNER JOIN Ocupacao O ON O.NroReserva = P0.NroReserva
INNER JOIN Quarto Q ON Q.NroQuarto = O.NroQuarto
WHERE P0.DataHora = CURDATE();
```

Resultado:

Andar	ValorDeHoje
1	5000.00
5	3950.00

FUNÇÃO AVG()

- Retorna a média de valores de campos de um grupo de registros.
Ex.: “Apresente a média de valores das diárias dos quartos do hotel”.

Consulta:

```
SELECT Avg(VlrDiaria) MediaDiarias
FROM Quarto;
```

Resultado:

MediasDiarias

256.25

Quarto		NroQuarto	VlrDiaria
PK	NroQuarto: INT	11	300.00
		12	150.00
		13	200.00
		14	250.00
		51	400.00
	Andar: TINYINT	52	200.00
	Tipo: CHAR(2)	53	250.00
	Descricao: VARCHAR(200)	54	300.00
	VlrDiaria: DECIMAL(7,2)		

FUNÇÃO AVG()

Ex.: “Apresente a média de valores das diárias dos quartos do hotel, agrupadas por andar”.

Consulta:

```
SELECT Andar, Avg(VlrDiaria) MediaDiarias
FROM Quarto
GROUP BY Andar;
```

Resultado:

Andar	MediaDiarias
1	225.00
5	287.50

Quarto	
PK	NroQuarto: INT
	Andar: TINYINT
	Tipo: CHAR(2)
	Descricao: VARCHAR(200)
	VlrDiaria: DECIMAL(7,2)
	11
	12
	13
	14
	51
	52
	53
	54

FUNÇÃO MIN()

- Retorna o menor valor de um campo em um grupo de registros.
Ex.: “Apresente valor da diária mais barata do hotel”.

Consulta:

```
SELECT Min(VlrDiaria) MenorDiaria
FROM Quarto;
```

Resultado:

MenorDiaria

150.00

Quarto		NroQuarto	VlrDiaria
PK	NroQuarto: INT	11	300.00
		12	150.00
		13	200.00
		14	250.00
		51	400.00
	Andar: TINYINT	52	200.00
	Tipo: CHAR(2)	53	250.00
	Descricao: VARCHAR(200)	54	300.00
	VlrDiaria: DECIMAL(7,2)		

FUNÇÃO MIN()

Ex.: “Apresente o número do quarto e o valor da diária mais barata do hotel”.

Consulta:

```
SELECT NroQuarto, VlrDiaria
FROM Quarto
WHERE VlrDiaria =
      (SELECT Min(VlrDiaria) FROM Quarto);
```

Resultado:

NroQuarto	VlrDiaria
12	150.00

FUNÇÃO MAX()

- Retorna o maior valor de um campo em um grupo de registros.
Ex.: “Apresente valor da diária mais cara do hotel”.

Consulta: **SELECT Max(VlrDiaria) MaiorDiaria
FROM Quarto;**

Resultado: **MaiorDiaria**
400.00

Quarto		NroQuarto	VlrDiaria
PK	NroQuarto: INT	11	300.00
		12	150.00
		13	200.00
		14	250.00
		51	400.00
	Andar: TINYINT	52	200.00
	Tipo: CHAR(2)	53	250.00
	Descricao: VARCHAR(200)	54	300.00
	VlrDiaria: DECIMAL(7,2)		

FUNÇÃO MAX()

Ex.: “Apresente o número do quarto e o valor da diária mais cara do hotel”.

Consulta:

```
SELECT NroQuarto, VlrDiaria
FROM Quarto
WHERE VlrDiaria =
      (SELECT Max(VlrDiaria) FROM Quarto);
```

Resultado:

NroQuarto	VlrDiaria
51	400.00



ATIVIDADE PRÁTICA

REFERÊNCIAS

- MySQL. Disponível em <https://mysql.com>. Acesso em: 26 ago. 2023.
- SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. ***Database System Concepts***. 6th Ed. New York: McGraw-Hill, 2011.