
Bases de Dados

Interrogações SQL – III (Operadores de Agregação)

FCUL, Departamento de Informática

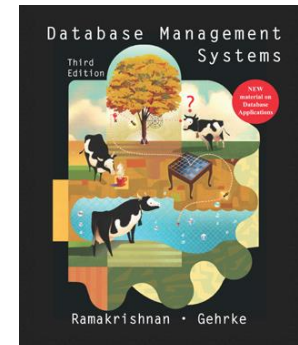
Ano Letivo 2021/2022

Ana Paula Afonso

Sumário e Referências

- Sumário
 - Operadores de agregação
COUNT, SUM, AVG, MAX, MIN
Exemplos
 - Sub-Interrogações com operadores de agregação
 - Agrupamento de linhas e Filtragem sobre grupos
GROUP BY e HAVING
Exemplos e Regras do GROUP BY
 - Operador de divisão

- Referências
 - R. Ramakrishnan (**capítulo 5, secção 5.5**)



Operadores de Agregação

- Os operadores de agregação
 - produzem sumários de dados tipicamente referentes a uma coluna da tabela
 - devolvem um valor único como resultado
- Operadores de agregação em SQL
 - COUNT ([DISTINCT] coluna)
Número de valores na coluna da tabela
 - SUM ([DISTINCT] coluna)
Soma dos valores na coluna A
 - AVG ([DISTINCT] coluna)
Média dos valores na coluna A
 - MAX (coluna) e MIN(coluna)
Máximo e mínimo valor na coluna

Observação: DISTINCT considera apenas os valores únicos, sem repetições

Tabelas de Exemplo

Sailors

<u>sid</u>	sname	rating	age
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
61	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

Boats

<u>bid</u>	bname	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

Reserves

<u>sid</u>	<u>bid</u>	<u>day</u>
22	101	10/10/96
58	103	11/12/96

Interrogações com COUNT

- Número total de marinheiros

```
SELECT COUNT(*)  
FROM Sailors S
```

COUNT(*)
10

<u>sid</u>	sname	rating	age
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
61	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

- Número de nomes distintos de marinheiros

```
SELECT COUNT(DISTINCT S.sname)  
FROM Sailors S
```

COUNT(DISTINCT S.sname)

9

Observação

- A utilização do * no COUNT é recomendada
- Quando estão a ser contadas linhas e não colunas específicas

Interrogações com AVG

- Média de idades dos marinheiros

```
SELECT AVG(S.age)
FROM Sailors S
```

AVG(S.age)
36.90000

<u>sid</u>	sname	rating	age
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
61	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

- Média de idades dos marinheiros com *rating* de 10

```
SELECT AVG(S.age)
FROM Sailors S
WHERE S.rating = 10
```

AVG(S.age)
25.50000

Interrogações com MAX

- Idade do marinheiro mais velho

```
SELECT MAX(S.age)
FROM Sailors S
```

- Nome e idade do marinheiro mais velho

```
SELECT S.sname, MAX(S.age)
FROM Sailors S
```

- Não é um comando válido

O operador de agregação MAX agrega num só valor
os valores das outras colunas não estão acessíveis
Excepto se GROUP BY (mais à frente)

```
SELECT S.sname, S.age
FROM Sailors S
WHERE S.age = ( SELECT MAX(S2.age)
                FROM Sailors S2 )
```

sid	sname	rating	age
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
61	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

sname	MAX(S.age)
Dustin	63.5



sname	age
Bob	63.5



Sub-Interrogações com Operadores de Agregação

- Nomes dos marinheiros que são mais velhos que o **marinheiro mais velho com *rating* de 10?**

```
SELECT S.sname
FROM Sailors S
WHERE S.age > ( SELECT MAX(S2.age)
                FROM Sailors S2
                WHERE S2.rating = 10)
```

sid	sname	rating	age
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
61	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

- Outra alternativa, com operador ALL

```
SELECT S.sname
FROM Sailors S
WHERE S.age > ALL( SELECT S2.age
                   FROM Sailors S2
                   WHERE S2.rating = 10)
```

sname

Dustin

Lubber

Bob

- Observação: > ALL pode ser menos claro que MAX

Agrupamento e Filtragem

- Sintaxe do comando SELECT

```
SELECT [DISTINCT] select-list  
FROM from-list  
[WHERE qualification]  
GROUP BY grouping-list  
[HAVING group-qualification]
```

- GROUP BY permite criar grupos de linhas
 - Cada grupo de linhas tem o mesmo valor nas colunas do **grouping-list**
 - GROUP BY (age) cria tantos grupos quantas as idades existentes
- HAVING elimina os grupos que não satisfazem a condição da **group-qualification**
 - Tem de incluir colunas da **grouping-list** ou operadores de agregação
 - HAVING (age > 15) inclui grupos cujos marinheiros têm idade > 15

Interrogações com GROUP BY

sid	sname	rating	age
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
61	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

- Idade do mais novo marinheiro para cada *rating*

```
SELECT S.rating, MIN(S.age)
FROM Sailors S
GROUP BY S.rating
```

rating	MIN(S.age)
1	33.0
3	25.5
7	35.0
8	25.5
9	35.0
10	16.0

- Idade do marinheiro mais novo com mais de 18 anos para cada *rating* com pelo menos 2 marinheiros (com mais de 18 anos)

```
SELECT S.rating, MIN(S.age) AS "Idade mínima"
FROM Sailors S
WHERE S.age > 18
GROUP BY S.rating
HAVING COUNT(*) >= 2
```

rating	Idade mínima
3	25.5
7	35.0
8	25.5

Passos de Execução

```
SELECT S.rating, MIN(S.age) AS "Idade minima"  
FROM Sailors S  
WHERE S.age > 18  
GROUP BY S.rating  
HAVING COUNT(*) >= 2
```

Sailors

<i>sid</i>	<i>sname</i>	<i>rating</i>	<i>age</i>
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

<i>rating</i>	<i>age</i>
7	45.0
1	33.0
8	55.5
8	25.5
10	35.0
7	35.0
9	35.0
3	25.5
3	63.5

Exclusão de marinheiros com idade não superior a 18 anos (e eliminação de colunas desnecessárias)

<i>rating</i>	<i>age</i>
1	33.0
3	25.5
3	63.5
7	45.0
7	35.0
8	55.5
8	25.5
9	35.0
10	35.0

Criação de grupos para cada *rating*

<i>rating</i>	<i>minage</i>
3	25.5
7	35.0
8	25.5

Exclusão de grupos com menos de 2 marinheiros e cálculo da idade mínima em cada grupo sobrevivente

(fonte: António Ferreira, SIBD19)

Agrupamento e Filtragem: Regras

- Sintaxe do comando SELECT

```
SELECT [DISTINCT] select-list  
FROM from-list  
[WHERE qualification]  
GROUP BY grouping-list  
[HAVING group-qualification]
```

- **Cada coluna na select-list** só pode ter um valor único por grupo
 - Podem ser colunas do grouping-list
Ou com dependência funcional
 - Podem ser operadores de agregação

Interrogações com GROUP BY e HAVING

- Número de reservas para cada barco vermelho

```
SELECT B.bid, COUNT (*) AS rescount
FROM Boats B, Reserves R
WHERE R.bid = B.bid
GROUP BY B.bid
HAVING B.color = 'red'
```



- Solução correta

```
SELECT B.bid, COUNT (*) AS rescount
FROM Boats B, Reserves R
WHERE R.bid = B.bid
      AND B.color = 'red'
GROUP BY B.bid
```



Interrogações com GROUP BY e HAVING

- Idade do marinheiro mais novo com **mais de 18 anos** para cada *rating* cuja **média de idades dos marinheiros** (com mais de 18 anos) **seja superior** à **média de idade de todos os marinheiros**?

```
SELECT S.rating, MIN(S.age) AS minage
FROM Sailors S
WHERE S.age >= 18
GROUP BY S.rating
HAVING AVG(S.age) > ( SELECT AVG(S2.age)
                      FROM Sailors S2)
```

rating	minage
3	25.5
7	35.0
8	25.5

Operador de divisão

- Utilidade
 - Permite expressar interrogações de um determinado tipo
 - p.e., "Nome dos marinheiros que reservaram todos os barcos"
- Exemplo

A

<i>sno</i>	<i>pno</i>
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

B1

<i>pno</i>
p2

B2

<i>pno</i>
p2
p4

B3

<i>pno</i>
p1
p2
p4

A / B1

<i>sno</i>
s1
s2
s3
s4

A / B2

<i>sno</i>
s1
s4

A / B3

<i>sno</i>
s1

Interrogações com GROUP BY e HAVING

Exemplo de Divisão

Nomes dos marinheiros que reservaram **todos** os barcos

```
SELECT S.sname
FROM Sailors S, Reserves R
WHERE R.sid = S.sid
GROUP BY S.sname
HAVING COUNT(DISTINCT R.bid) = ( SELECT COUNT(*)
                                FROM Boats B)
```


Interrogações com NOT EXISTS

Exemplo de Divisão

Nomes dos marinheiros que reservaram **todos** os barcos

```
SELECT S.sname
FROM Sailors S
WHERE NOT EXISTS (
    SELECT B.bid
    FROM Boats B
    WHERE NOT EXISTS (
        SELECT R. bid
        FROM Reserves R
        WHERE R.bid = B.bid
            AND R.sid = S.sid ))
```

Obs. All x: $p(x) \iff \text{Not Exists } x: \text{Not } p(x)$

Exemplo de Divisão

1. Nome dos marinheiros, para os quais

- Qualquer que seja o barco
- Existe uma reserva para esse barco e para esse marinheiro

2. Nome dos marinheiros:

$\forall \text{ barco: } (\exists \text{ reserva: } R.\text{bid} = B.\text{bid} \text{ AND } R.\text{sid} = S.\text{Sid})$

Nota: $\forall x: p(x) \Leftrightarrow \sim \exists x: \sim p(x)$

3. Nomes dos marinheiros:

$\sim \exists \text{ barco: } \sim \exists \text{ reserva: } R.\text{bid} = B.\text{bid} \text{ AND } R.\text{sid} = S.\text{sid}$

```
SELECT S.sname
FROM Sailors S
WHERE NOT EXISTS (SELECT B.bid
                   FROM Boats B
                   WHERE NOT EXISTS (SELECT R. bid
                                     FROM Reserves R
                                     WHERE R.bid = B.bid AND R.sid = S.sid ))
```

Exemplo de Divisão com Restrição

- Nomes dos marinheiros que reservaram **todos** os barcos **vermelhos**

```
SELECT S.sname
FROM Sailors S
WHERE NOT EXISTS (
    SELECT B.bid
    FROM Boats B
    WHERE B.color = 'red'
        AND NOT EXISTS (
            SELECT R.bid
            FROM Reserves R
            WHERE R.bid = B.bid
                AND R.sid = S.sid ))
```