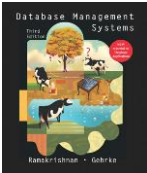


# Sumário e Referências

---

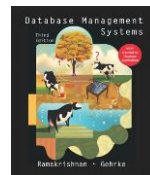
- Sumário
  - Operadores de agregação  
COUNT, SUM, AVG, MAX, MIN  
Exemplos
  - Sub-Interrogações com operadores de agregação
  - Agrupamento de linhas e Filtragem sobre grupos  
GROUP BY e HAVING  
Exemplos e Regras do GROUP BY
- Referências
  - R. Ramakrishnan (**capítulo 5, secção 5.5**)



The image shows the front cover of the textbook 'Database Management Systems' by Ramakrishnan and Gorthi. The cover is dark with a colorful illustration of a rural scene featuring a tree, a cow, and a house. The title is at the top, and the authors' names are at the bottom. A red circular badge on the right side of the cover indicates '5th Edition'.

© 2015 - Docentes SI - DI/FCUL

2



## 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

© 2015 - Docentes SI - DI/FCUL

3

## Interrogações com COUNT

- Número total de marinheiros

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

- Número de nomes distintos de marinheiros

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

Observação

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

© 2015 - Docentes SI - DI/FCUL

4

## Interrogações com AVG

---

- Média de idades dos marinheiros

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

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

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

## 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 )
```

## 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)
```

- 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)
```

- Observação: > ALL pode ser menos claro que MAX

© 2015 - Docentes SI - DI/FCUL

7

## 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 **group-qualification**
  - Tem de incluir colunas da **grouping-list** ou operadores de agregação
  - **HAVING (age > 15)** inclui grupos cujos marinheiros têm idade > 15

© 2015 - Docentes SI - DI/FCUL

8

# Interrogações com GROUP BY

- Idade do mais novo marinheiro para cada rating
- 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)
FROM Sailors S
GROUP BY S.rating
```

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

# Passos de Execução

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

Sailors

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
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

rating	age
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

rating	age
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

rating	minage
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

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

Criação de grupos para cada rating

(fonte: António Ferreira, SI8D15)

## 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' -- Incorreto!
```

- 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)
```