
Introdução às Bases de Dados

Interrogações SQL – IV

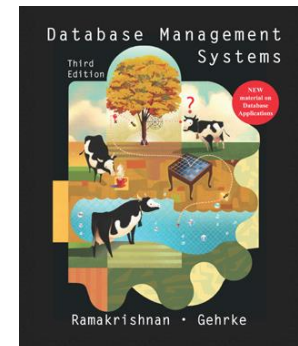
FCUL, Departamento de Informática

Ano Letivo 2021/2022

Ana Paula Afonso

Sumário e Referências

- Sumário
 - Ordenação dos resultados
ORDER BY ASC, ORDER BY DESC
 - Valores Nulos
Comparações e Operações
Verificação de existência de valores Nulos
Impacto nos comandos SQL
Operadores de Agregação
Junções Exteriores
- Referências
 - R. Ramakrishnan (**capítulo 5, secção 5.6**)



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	NULL
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

Nota: Zorba tem idade desconhecida

Ordenação de Resultados de Interrogações

- Sintaxe do comando SELECT

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

- ORDER BY permite ordenar os resultados de uma interrogação
 - Sem ORDER BY a ordem das linhas no resultado é arbitrária
 - As colunas na `order-list` devem constar na `select-list`
Para facilitar a compreensão do(s) critério(s) de ordenação
 - Ordenação ascendente (ASC) ou descendente (DESC)
Aplica-se a cada coluna da `order-list`

Ordenação de Resultados de Interrogações

- Todos os marinheiros ordenados alfabeticamente e depois por *rating* (do maior para o menor)

```
SELECT S.sname, S.rating
FROM Sailors S
ORDER BY S.sname ASC, S.rating DESC
```

<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	NULL
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

sname	rating
Andy	8
Art	3
Bob	3
Brutus	1
Dustin	7
Horatio	9
Horatio	7
Lubber	8
Rusty	10
Zorba	10

Valores Nulos

- Usados quando se **desconhece** o valor de uma coluna ou quando **não aplicável**
 - Em SQL são representados por **NULL**
 - Ex. desconhece-se o telemóvel de um cliente, mas pode vir a saber-se
 - Ex. nome de solteira só é relevante para mulheres casadas (*maiden name*)
- Aplicabilidade
 - Podem ser usados em colunas com qq domínio de dados
 - Não podem ser usados em chaves candidatas, chave primária
 - Não podem ser usados em colunas NOT NULL

Operações com Valores Nulos

- Operações de comparação (<, <=, =, <>, >=, >) e aritméticas (+, -, *, /)
 - Basta um dos argumentos ser NULL, para resultado ser *unknown*
 - arg1 Op arg2 , se arg1 ou arg2 NULL => resultado *unknown*
- Operações conjunção e disjunção (AND, OR)
 - NULL **AND** FALSE = FALSE, cc. resultado é sempre *unknown*
 - NULL **OR** TRUE = TRUE, cc. resultado é sempre *unknown*
- Verificação de valores nulos (IS NULL)
 - NULL **IS NULL** = TRUE (ex. s.age IS NULL)
 - NULL **IS NOT NULL** = FALSE (ex. s.age IS NOT NULL)

Impacto nos Comandos SQL

- A condição na cláusula **WHERE**
 - Só aparecem no resultado as linhas que verificam a condição ou seja, para a qual a condição seja TRUE
 - Elimina as linhas cujo resultado seja FALSE ou *unknown*
- Ex. Marinheiros com idade superior a 18
 - O marinheiro Zorba aparece no resultado?

```
SELECT *
```

```
FROM Sailors
```

```
WHERE age > 18
```

<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	NULL
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

Impacto nos Comandos SQL

- A condição na cláusula **WHERE**
 - Só aparecem no resultado as linhas que verificam a condição ou seja, para a qual a condição seja TRUE
 - Elimina as linhas cujo resultado seja FALSE ou *unknown*

- Ex. Marinheiros com idade superior a 18

```
SELECT *  
FROM Sailors  
WHERE age > 18
```

- O marinheiro Zorba aparece no resultado?
- `NULL > 18 = unknown`, pelo que não é selecionado

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
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

Impacto nos Comandos SQL

- Operadores de agregação (resultados inesperados)
 - COUNT(*) , na contagem inclui os valores NULL
 - Restantes operadores (com ou sem DISTINCT) ignoram NULL
COUNT(coluna), SUM(coluna), AVG(coluna), MAX/MIN(coluna)
 - Se todos os valores na coluna forem nulos, resultado é NULL
Exceto COUNT(coluna) que devolve 0

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

count(*)
10

```
SELECT COUNT (age)  
FROM Sailors
```

COUNT(age)
9

<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	NULL
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

Junções Naturais (*natural join*)

- Marinheiros e as suas reservas (junção interior)

```
SELECT *  
FROM Sailors S, Reserves R  
WHERE S.sid = R.sid
```

```
SELECT *  
FROM Sailors S INNER JOIN Reserves R  
ON (S.sid = R.sid)
```

<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	NULL
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

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

<u>sid</u>	<u>sname</u>	<u>rating</u>	<u>age</u>	<u>sid</u>	<u>bid</u>	<u>day</u>
22	Dustin	7	45.0	22	101	1996-10-10
58	Rusty	10	35.0	58	103	1996-12-11

Observação: E os marinheiros que não fizeram reservas?
Não devolve marinheiros sem reservas

Junções Exteriores (*outer join*)

- As junções exteriores
 - Devolvem as linhas que satisfazem a condição de junção e as linhas numa das tabelas sem correspondência com as da outra tabela
 - FULL OUTER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN
- Ex. Marinheiros e as suas reservas

```
SELECT *  
FROM Sailors S FULL OUTER JOIN Reserves R  
ON (S.sid = R.sid)
```

- Devolve todos os marinheiros mesmo os que não fizeram reservas
- Devolve também as reservas, mesmo as que não tiverem marinheiros (...)

Junções Exteriores

- Qual o número de barcos reservados para cada marinheiro?

```
SELECT S.sname, COUNT(R.bid)
FROM Sailors S LEFT OUTER JOIN Reserves R
    ON (S.sid = R.sid)
GROUP BY S.sname
```

sname	COUNT(R.bid)
Andy	0
Art	0
Bob	0
Brutus	0
Dustin	1
Horatio	0
Lubber	0
Rusty	1
Zorba	0

- LEFT indica que todas as linhas de Sailors vão aparecer mesmo as que não têm reserva essas são juntas com valores nulos nas colunas de R

Junções exteriores

- Variantes - Considerando a junção de T1 com T2, por esta ordem
 - Junção exterior esquerda: inclui todos os valores de T1
 - Junção exterior direita: inclui todos os valores de T2
 - Junção exterior completa: inclui todos os valores de T1 e T2
- Exemplos
 - Todos os marinheiros e suas reservas, mesmo os que não fizeram reservas
`SELECT * FROM sailors S LEFT OUTER JOIN reserves R ON (S.sid = R.sid)`
 - Todas as reservas e seus marinheiros, mesmo as que não têm marinheiros
`SELECT * FROM sailors S RIGHT OUTER JOIN reserves R ON (S.sid = R.sid)`

Como todas as reservas têm de ter um marinheiro (ver chave primária) esta junção exterior direita é equivalente à junção interior

MySQL exemplos: tabelas t1 e t2

```
-- t1
id  name
1   Tim
2   Marta
```

```
-- t2
id  name
1   Tim
3   Katarina
```

```
SELECT *
FROM `t1`
INNER JOIN `t2` ON `t1`.`id` = `t2`.`id`;
```

```
1 Tim  1 Tim
```

```
SELECT *
FROM `t1`
LEFT OUTER JOIN `t2` ON `t1`.`id` = `t2`.`id`;
```

```
1 Tim  1   Tim
2 Marta NULL NULL
```

```
SELECT *
FROM `t1`
RIGHT OUTER JOIN `t2` ON `t1`.`id` = `t2`.`id`;
```

```
1   Tim  1   Tim
NULL NULL 3 Katarina
```

```
SELECT *
FROM `t1`
LEFT OUTER JOIN `t2` ON `t1`.`id` = `t2`.`id`

UNION

SELECT *
FROM `t1`
RIGHT OUTER JOIN `t2` ON `t1`.`id` = `t2`.`id`;
```

```
1   Tim  1   Tim
2   Marta NULL NULL
NULL NULL 3   Katarina
```