

SQL

Funções de Agregação e Grupos



INFORMÁTICA
1º ANO – 2023/2024

ARTUR JORGE DA SILVA ROCHA
E-MAIL: D001457@UMAIA.PT

Funções de Agregação



Funções de Agregação



- Funções de agregação: Calculam um resultado tendo como argumento uma série de valores, e não apenas um.
- Apesar de a entrada ser um vector de vários elementos, o resultado é apenas um valor escalar
- Funções mais usuais:
 - COUNT
 - AVG
 - SUM
 - MAX
 - MIN

Sintaxe



- Funções de agregação exigem uma coluna como argumento
- Resultado é calculado usando todos os valores da coluna respectiva
- Recebem um único argumento, mas este pode ser uma expressão complicada usando uma ou mais colunas
- Valor da expressão é avaliado para cada linha da tabela e a função de agregação é calculada sobre o total destes valores (um por cada linha).
- Se uma expressão é avaliada *null* para uma linha, esse valor não é considerado
- A expressão pode ser precedida de DISTINCT
- Caso especial: o COUNT admite o argumento *, o que significa ‘contar todas as linhas’, incluindo linhas com valores *null*

Exemplos



- Contar o total de pilotos
 - `SELECT COUNT(*) FROM piloto` ou
 - `SELECT COUNT(id) FROM piloto`
- Mas note-se o resultado de
 - `SELECT COUNT(anoNasc) FROM piloto`

Sintaxe



- As funções AVG e SUM só aceitam valores de tipos numéricos.
- COUNT, MIN e MAX aceitam valores de tipos numéricos, de data e alfanuméricos.
- É possível filtrar as linhas a incluir na função com a cláusula WHERE, como normalmente.
- É conveniente dar um nome à coluna assim gerada (usando AS).

Exemplo



- Total de pontos de todos os pilotos brasileiros em 1975

```
SELECT SUM(pontos) AS pontos  
FROM piloto JOIN epoca ON piloto.id = epoca.piloto_id  
WHERE pais = 'BRA'  
AND ano = 1975
```

Número de Colunas e Linhas



- Em cada coluna do resultado da query, apenas se pode usar uma função de agregação: não é permitido usar funções de agregação como argumentos de outras
- Diferentes colunas do resultado podem ter diferentes funções de agregação ou até repetições da mesma função
- Quando um SELECT contém funções de agregação, só se podem incluir colunas não agregadas (sem funções agregadas) se se definirem grupos
- Se não se usarem grupos, o resultado de um SELECT com agregação é uma única linha.

Grupos



Grupos



- É possível dividir as linhas da tabela (depois do filtro do *WHERE*) em grupos, e calcular o resultado de uma agregação para cada grupo
- Grupos são especificados com a cláusula *GROUP BY*
- Intuitivamente: a tabela é ordenada pelas colunas especificados na cláusula *GROUP BY* e todas as linhas que tiverem o mesmo valor nessas colunas são consideradas um grupo
- Resultado da agregação devolve uma linha para cada grupo
- As colunas especificadas na cláusula *GROUP BY* podem ser incluídas no *SELECT*

Exemplo



- Total de pontos que um piloto fez na sua carreira

```
SELECT pnome,unome, SUM(pontos) as pontos  
FROM epoca JOIN piloto ON epoca.piloto_id = piloto.id  
GROUP BY piloto_id, pnome, unome  
ORDER BY pontos desc
```

HAVING



- Com a cláusula WHERE podemos filtrar as linhas que queremos incluir no resultado final
- Mas podemos também filtrar os grupos que queremos que apareçam
- Para isso, usa-se a cláusula HAVING
- A condição em HAVING é semelhante ao WHERE e só pode incluir colunas no GROUP BY ou argumentos de uma função de agregação

Exemplo



- Mostrar o total de vitórias conseguidas, em toda a sua vida, por cada construtor, mas apenas para aqueles que tenham ganho por mais de 5 vezes.

```
SELECT construtor.nome, SUM(vitorias)
```

```
FROM equipa JOIN construtor ON construtor_id =  
construtor.id
```

```
JOIN epoca ON ((epoca.piloto_id IN (equipa.piloto1_id,  
equipa.piloto2_id))
```

```
AND (epoca.ano = equipa.ano))
```

```
GROUP BY construtor_id, construtor.nome
```

```
HAVING SUM(vitorias) >= 5
```



- Para todos os pilotos que participaram em pelo menos 15 corridas, mostrar o total de pontos em toda a sua carreira, o número de anos em que correram e a média de pontos por ano.

```
SELECT pnome,unome, AVG(pontos) AS media,  
COUNT(DISTINCT ano) AS anos, SUM(pontos) AS total  
FROM epoca JOIN piloto ON epoca.piloto_id = piloto.id  
GROUP BY piloto_id, pnome, unome  
HAVING SUM(partidas) >= 15  
ORDER BY media DESC
```

Exercícios (I)



- Devolver o total de pontos de cada equipa em cada ano
- Apresentar uma lista de construtores ordenada por ordem decrescente do número de corridas efectuadas em toda a História
- Para cada marca de motor, apresentar o número de vitórias que conseguiu em toda a história
- Para cada ano, liste o total de pilotos presentes no início de cada grande prémio, o total de pilotos inscritos e o total de pilotos que terminaram a corrida.
 - Nota: Estes dados podem ser encontrados na tabela resultados.
 - Um piloto que tenha um valor não-nulo na coluna posicao correu e terminou a corrida; caso contrário, a coluna resultado indica porque é que não terminou:
 - ✦ RET - desistência durante a corrida;
 - ✦ DNQ - não se qualificou e por isso não iniciou;
 - ✦ DNS - alinhou para a partida, mas o carro não partiu;
 - ✦ DSQ - desqualificado durante a corrida;
 - ✦ NC - terminou mas não foi classificado;
 - ✦ WD - retirou a inscrição antes da corrida.
 - ✦ Um piloto com NULL nas colunas resultado e posicao não estava inscrito para a corrida.

Exercícios (II)



- Para cada país, liste as pistas onde já se realizaram provas de F1 e quantas vezes em cada circuito.
- Conte o número de circuitos existentes em cada continente
- Indique para cada piloto quais foram a melhor e a pior posição que alguma vez alcançou
- Para cada construtor, indique o número de pódios conseguidos durante cada ano.
 - Nota: um pódio é uma classificação num dos 3 primeiros lugares
- Para cada ano, indique o número médio de pilotos que terminaram cada corrida.
- Para cada circuito, indique o número total de pilotos (distintos) que aí ficaram nos seis primeiros lugares.