



Linguagem SQL

09 – Agregações e Sumarização

Sand Onofre

Sand.Onofre@FaculdadeImpacta.com.br

Sumário

- Data Query Language – DQL
 - Funções de Agregação (MIN, MAX, COUNT, AVG, SUM, ...)
 - GROUP BY
 - HAVING
- Exercícios

Funções de Agregação

Funções de agregação são funções que estão embutidas (bulti-in) no banco de dados e utilizamos quando precisamos calcular valores, contabilizar número de registros ou retornar os maiores e menores valores dentro de uma coluna.

Possuem as seguintes características:

- ✓ Retornam valores escalares
- ✓ Retornam a coluna sem nome
- ✓ Ignoram colunas NULL (exceção a COUNT(*))
- ✓ Podem ser usadas nas cláusulas:

SELECT, HAVING e ORDER BY

Funções de Agregação

Iremos focar nas funções de agregação de uso comum, mas é importante saber que existem outras categorias de funções agregadas como estatísticas e outras.

Uso Comum

- SUM
- MIN
- MAX
- AVG
- COUNT
- COUNT_BIG

Estatísticas

- STDEV
- STDEVP
- VAR
- VARP

Funções de Agregação

São frequentemente utilizadas com a cláusula GROUP BY, mas não são restritas ao uso sem este comando. O exemplo abaixo mostra funções de agregação sem o uso do GROUP BY.

```
SELECT  AVG(unitprice) AS avg_price,
        MIN(qty) AS min_qty,
        MAX(discount) AS max_discount
FROM    Sales.OrderDetails;
```

avg_price	min_qty	max_discount
-----	-----	-----
26.2185	1	0.250

Funções de Agregação

Podemos combinar a cláusula DISTINCT com funções de agregação para sumarizar somente valores ÚNICOS.

A agregação com a cláusula DISTINCT elimina valores duplicados, não linhas (exceto se utilizarmos SELECT DISTINCT).

Compare os resultados parciais do exemplo abaixo, com e sem o uso da cláusula DISTINCT:

```
SELECT  COUNT(custid) AS all_custs,
        COUNT(DISTINCT custid) AS unique_custs
FROM Sales.Orders
```

all_custs	unique_custs
26	22

Funções de Agregação

A maioria das funções de agregação, simplesmente ignoram o NULL e não geram nenhum erro. As funções a seguir Ignoram o NULL:

AVG(<coluna>), COUNT (<coluna>), ...

A função COUNT usada com *, é uma EXCEÇÃO a regra acima, contabilizando TODAS AS LINHAS:

COUNT(*)

Esse tipo de comportamento das funções agregadas perante o NULL, pode produzir resultados INCORRETOS, como é o caso abaixo, utilizando a função AVG. Se quisermos contabilizar os registros nulos, podemos ajustar os dados com a função ISNULL.

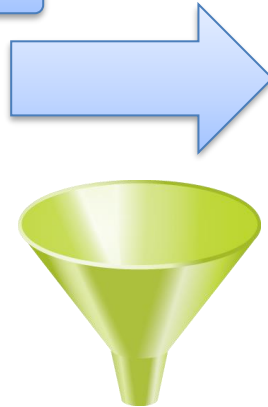
```
SELECT  AVG(c2) AS AvgWithNULLs,
        AVG(ISNULL(c2,0)) AS AvgWithNULLReplace
FROM    dbo.t2;
```

Demonstração de Funções de Agregação

Cláusula GROUP BY

```
SELECT orderid, empid, custid
FROM Sales.Orders;
```

orderid	empid	custid
10643	6	1
10692	4	1
10926	4	2
10625	3	2
10365	3	3



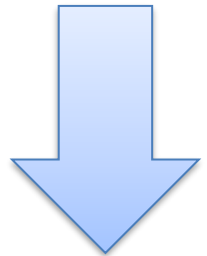
WHERE custid IN(1,2)

orderid	empid	custid
10643	6	1
10692	4	1
10926	4	2
10625	3	2

SELECT output

empid	COUNT(*)
6	1
4	2
3	1

GROUP BY empid



Cláusula GROUP BY

GROUP BY cria grupos no retorno das linhas de acordo com a combinação da(s) coluna(s) escritas na cláusula GROUP BY.

```
SELECT <select_list>
FROM <table_source>
WHERE <search_condition>
GROUP BY <group_by_list>;
```

O GROUP BY retira os “detalhes” das linhas, fazendo um cálculo com a função de agregação selecionada para a coluna escolhida.

```
SELECT empid, SUM(freight) AS freight
FROM Sales.Orders
GROUP BY empid;
```

```
SELECT empid, COUNT(*) AS cnt
FROM Sales.Orders
GROUP BY empid;
```

Cláusula GROUP BY

Funções de agregação são comumente usadas em cláusula SELECT, resumindo a(s) coluna(s) colocada(s) no GROUP BY.

```
SELECT custid, COUNT(*) AS cnt
FROM Sales.Orders
GROUP BY custid;
```

Funções de agregação podem referir qualquer coluna, não apenas as que estiverem escritas no GROUP BY.

```
SELECT productid, MAX(qty) AS largest_order
FROM Sales.OrderDetails
GROUP BY productid;
```

Cláusula GROUP BY

Ordem Lógica	Fase	Comentário
5	SELECT	
1	FROM	
2	WHERE	
3	GROUP BY	Cria Grupos
4	HAVING	Opera na Filtragem dos Grupos
6	ORDER BY	

- Se a consulta (query) usa GROUP BY, todas as fases subsequentes irão operar nos grupos.
- HAVING, SELECT e ORDER BY precisam NECESSARIAMENTE que retornar apenas um valor por grupo.
- Todas as colunas que aparecerem no SELECT, HAVING e ORDER BY, precisam OBRIGATORIAMENTE estar ou no GROUP BY ou numa Função de Agregação.

Cláusula HAVING

- HAVING filtra os dados obtidos através do GROUP BY.
- HAVING fornece condição de pesquisa que precisa ser satisfeita para cada grupo.
- HAVING é processado após a execução do GROUP BY.

```
SELECT custid, COUNT(*) AS count_orders
FROM Sales.Orders
GROUP BY custid
HAVING COUNT(*) > 10;
```

WHERE x HAVING

WHERE

- Filtra linhas ANTES dos grupos serem criados
- Controla quais linhas serão passadas para o GROUP BY

HAVING

- Filtra GRUPOS
- Controla quais GRUPOS serão passados para próxima fase lógica

WHERE x HAVING

A utilização da expressão COUNT(*) combinado com a cláusula HAVING é muito útil para solucionar problemas comuns de negócios. Exemplos:

- Mostre apenas os clientes que fizeram mais de um pedido

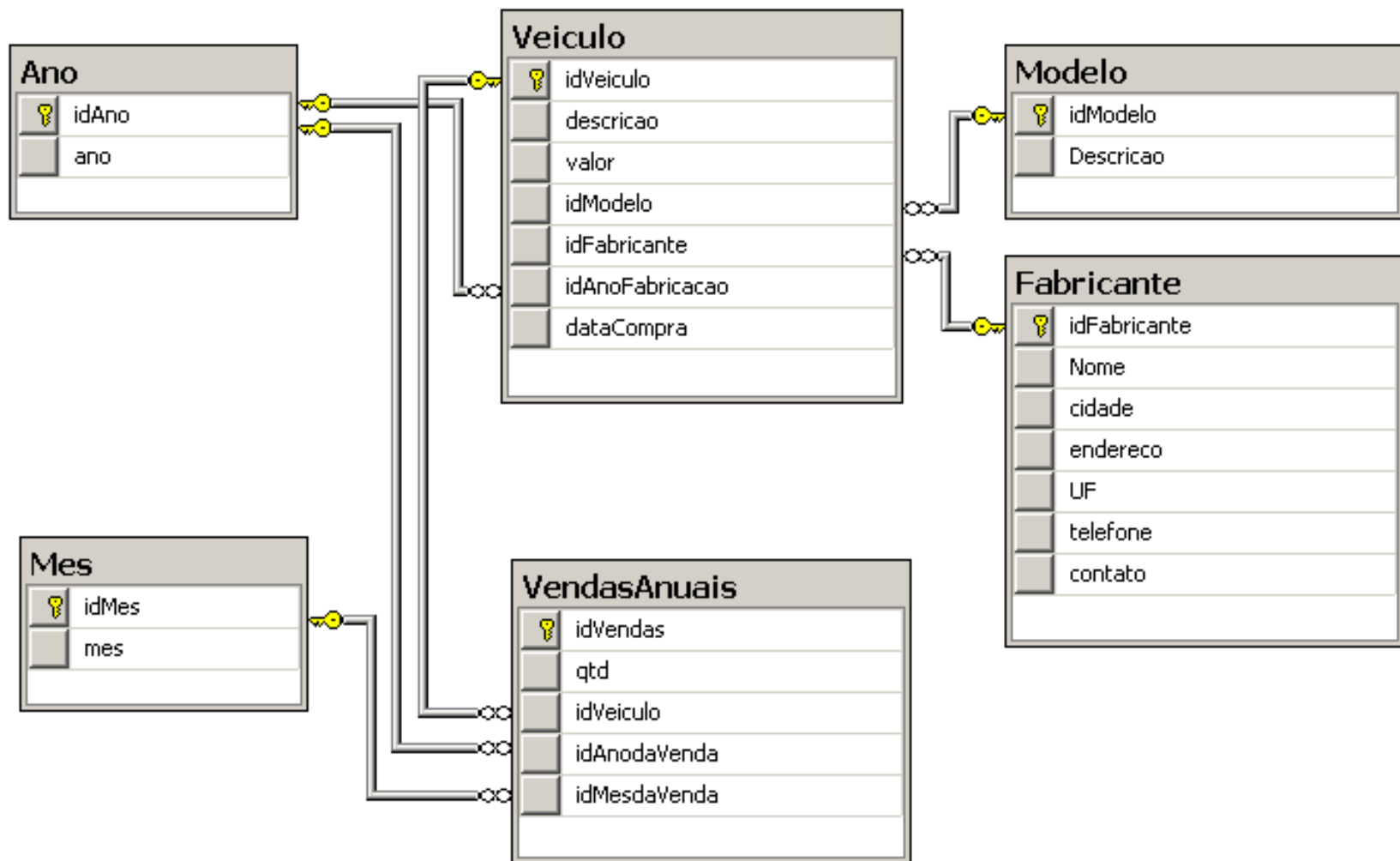
```
SELECT c.contactname, COUNT(*) AS cnt
FROM Sales.Customers AS c
      JOIN Sales.Orders AS o ON c.custid = o.custid
GROUP BY c.contactname
HAVING COUNT(*) > 1;
```

- Retorne somente produtos que aparecem mais de 10 vezes nos pedidos

```
SELECT p.productname, sum(qty) AS qty
FROM Production.Products AS p JOIN Sales.OrderDetails
      AS od ON p.productid = od.productid
WHERE p.productname in ('Product A', 'Product B')
GROUP BY p.productname
HAVING COUNT(*) > 10;
```

Demonstração de Agrupamento

Exercícios



Exercícios

Na base Concessionária, utilize funções de agregação para solucionar as seguintes questões:

- 1) Qual o número de registros existentes na tabela VendasAnuais ?
- 2) Qual o acumulado da quantidade de vendas Totais até o momento ?
- 3) Quais as quantidades de vendas Totais até o momento para: menor venda, média de vendas e a maior venda?
- 4) Extraia a soma das vendas anuais por ano em ordem descendente.
- 5) Traga a soma das vendas anuais para o veículo/modelo CG 125 STD (as informações do veículo devem constar na query).
- 6) Traga as primeiras datas (ANOS) de FABRICAÇÃO de todos os veículos e modelos, ordenados pelo nome do fabricante (ascendente), ano (descendente), Veículo (ascendente) e Modelo (descendente) Toda as informações solicitadas, inclusive ordenação, devem constar na query.
- 7) Extraia a menor, maior, média e a soma das vendas de cada mês do ano de 2000, em ordem ascendente.
- 8) Retorne a mesma consulta anterior, mas somente os registros que tiverem média de vendas superior a 500.



Obrigado !

Sand Onofre
Sand.Onofre@FaculdadeImpacta.com.br