

– SQL –
Comandos para Relatórios
e Formulários

Laboratório de Bases de Dados
Profa. Dra. Cristina Dutra de Aguiar Ciferri

SELECT

```
SELECT <lista de atributos e funções>  
FROM <lista de tabelas>  
[ WHERE predicado ]  
[ GROUP BY <atributos de agrupamento> ]  
[ HAVING <condição para agrupamento> ]  
[ ORDER BY <lista de atributos> ] ;
```

Funções de Agregação

- Funções
 - Média: AVG()
 - Mínimo: MIN()
 - Máximo: MAX()
 - Total: SUM()
 - Contagem: COUNT()
- Observação
 - DISTINCT: não considera valores duplicados
 - ALL: inclui valores duplicados

Funções de Agregação

- Características
 - recebem uma coleção de valores como entrada
 - retornam um único valor
- Entrada
 - sum() e avg(): conjunto de números
 - demais funções: tipos de dados numéricos e não-numéricos

Funções de Agregação

vinho (vinho_id, nome_vinho, tipo_vinho, preço, vinícola_id)

vinho_id	nome_vinho	tipo_vinho	preço	vinícola_id
10	Amanda	tinto	100,00	1
09	Belinha	branco	200,00	1
05	Camila	rosê	300,00	1
15	Daniela	branco	250,00	2
27	Eduarda	branco	150,00	2
48	Fernanda	tinto	7,00	2
13	Gabriela	tinto	397,00	3
12	Helena	branco	333,00	3

Exemplos

- Qual a *média* dos preços?

```
SELECT AVG (preço)  
FROM vinho
```

217,125

- Qual a *soma* dos preços?

```
SELECT SUM (preço)  
FROM vinho
```

1737,00

- Qual o preço mais *baixo*?

```
SELECT MIN (preço)  
FROM vinho
```

7,00

- Qual o preço mais *alto*?

```
SELECT MAX (preço)  
FROM vinho
```

397,00

Exemplos

- *Quantos* vinhos existem na relação vinho?

```
SELECT COUNT (vinho_id)
```

```
FROM vinho      8
```

- Quantos tipos de vinho *diferentes* existem na relação vinho?

```
SELECT COUNT (DISTINCT tipo_vinho)
```

```
FROM vinho      3
```

Cláusula GROUP BY

- Funcionalidade
 - permite aplicar uma função de agregação não somente a um conjunto de tuplas, mas também a um grupo de conjunto de tuplas
- Grupo de conjunto de tuplas
 - conjunto de tuplas que possuem o mesmo valor para os atributos de agrupamento
- Semântica da respostas
 - atributos de agrupamento no GROUP BY também devem aparecer no SELECT

Exemplo

- Qual o preço mais alto e a *média* dos preços *por tipo de vinho*?

```
SELECT tipo_vinho,  
       MAX (preço) AS "maior preço",  
       AVG (preço) AS "preço médio"  
FROM vinho  
GROUP BY tipo_vinho
```

Solução

- As tuplas da tabela vinho são divididas em grupo, cada grupo contendo o mesmo tipo de valor para o atributo de agrupamento tipo_vinho

vinho_id	nome_vinho	tipo_vinho	preço	vinícola_id
10	Amanda	tinto	100,00	1
09	Belinha	branco	200,00	1
05	Camila	rosê	300,00	1
15	Daniela	branco	250,00	2
27	Eduarda	branco	150,00	2
48	Fernanda	tinto	7,00	2
13	Gabriela	tinto	397,00	3
12	Helena	branco	333,00	3

Solução

- Considerações adicionais
 - a função MAX e a função AVG são aplicadas a cada grupo de tuplas separadamente
 - a cláusula SELECT inclui somente os atributos de agrupamento e as funções a serem aplicadas a cada grupo de tuplas
 - o comando SELECT pode possuir cláusula WHERE de qualquer complexidade

Solução

tipo_vinho	maior preço	preço médio
branco	333	233,25
rosê	300	300
tinto	397	168

Cláusula HAVING

- Funcionalidade
 - permite especificar uma condição de seleção para grupos, melhor do que para tuplas individuais
- Resposta
 - recupera os valores para as funções somente para aqueles grupos que satisfazem à condição imposta na cláusula HAVING

Exemplo

- Qual o preço mais alto e a *média* dos preços *por tipo de vinho*, para médias de preços superiores a R\$200,00

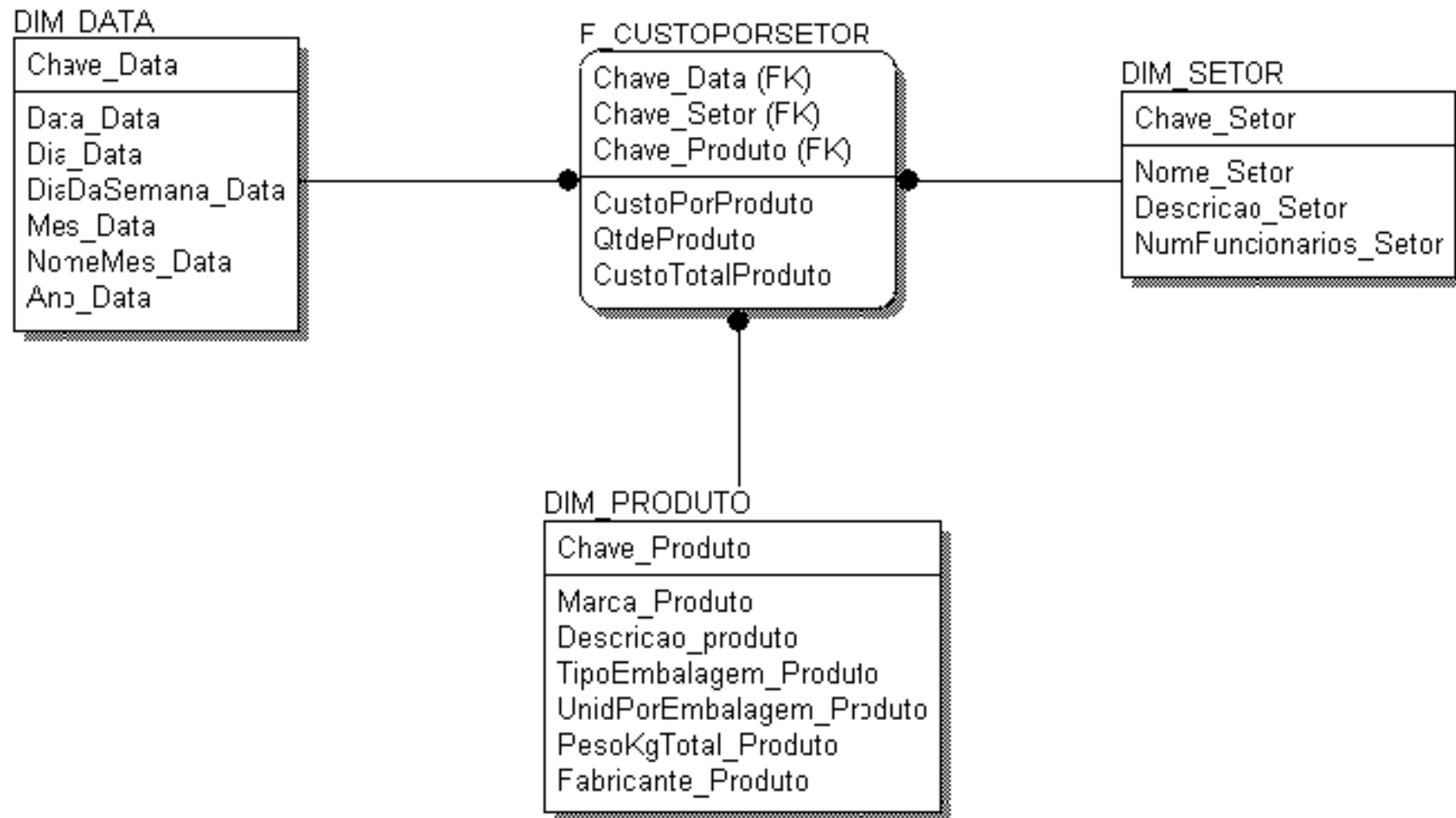
```
SELECT tipo_vinho, MAX (preço), AVG (preço)
FROM vinho
GROUP BY tipo_vinho
HAVING AVG (preço) > 200
```

tipo_vinho	max(preço)	avg(preço)
branco	333	233,25
rosê	300	300

Processamento da Consulta

- Passos
 - aplica-se o predicado que aparece na cláusula WHERE
 - coloca-se as tuplas que satisfazem a cláusula WHERE em grupos por meio da cláusula GROUP BY
 - aplica-se a cláusula HAVING a cada grupo
 - remove-se os grupos que não satisfazem o predicado da cláusula HAVING
 - exhibe-se as colunas listadas na cláusula SELECT

Esquema: Custo por Setor



Funções de Agregação Estendidas

- ROLLUP e CUBE
 - permitem aos usuários elaborarem consultas SQL que executem funções semelhantes à cláusula GROUP BY
 - geram agregação dos dados
 - podem ser utilizadas para a construção de vários níveis de agregação a partir de níveis subjacentes já existentes

GROUP BY

- Quais as despesas do hospital em cada ano, separado pelos diferentes setores?

```
SELECT d.ano_data Ano, s.nome_setor Setor,  
       sum(f.CustoTotalProduto) DespesaTotal  
FROM f_custoporsetor f JOIN dim_data d  
     ON d.chave_data=f.chave_data  
     JOIN dim_setor s  
     ON s.chave_setor=f.chave_setor  
GROUP BY d.ano_data, s.nome_setor
```

Resultado GROUP BY

ANO	SETOR	DESPESA TOTAL
2000	Limpeza	83618
2000	Lavanderia	92473,5
2000	Secretaria	74631
2000	Alimentação	135529,8
2000	Medicamentos	329780
2001	Limpeza	80571
2001	Lavanderia	126022,5
2001	Secretaria	87622,5
2001	Alimentação	141860,8
2001	Medicamentos	366360
2002	Limpeza	90396
2002	Lavanderia	108843
2002	Secretaria	66722
2002	Alimentação	146340,6
2002	Medicamentos	327750
2003	Limpeza	83492,5
2003	Lavanderia	95205
2003	Secretaria	80564,5
2003	Alimentação	120127,8
2003	Medicamentos	254150

ROLLUP

- Funcionalidade
 - criação de subtotais que envolvem desde o nível mais detalhado até um total geral, seguindo uma lista de agrupamento especificada na cláusula ROLLUP
- Argumento
 - lista ordenada de agrupamento de colunas

ROLLUP

- **Processamento**
 - cálculo dos valores agregados padrões especificados na cláusula GROUP BY
 - criação, de forma progressiva, de subtotais de nível mais alto, da esquerda para a direita na lista de agrupamento de colunas
 - criação de um total geral
- **Resultado**
 - $n+1$ níveis, sendo n o número de agrupamento de colunas

ROLLUP

- Quais as despesas do hospital em cada ano, separado pelos diferentes setores?

```
SELECT d.ano_data Ano, s.nome_setor Setor,  
       sum(f.CustoTotalProduto) DespesaTotal  
FROM f_custoporsetor f JOIN dim_data d  
     ON d.chave_data=f.chave_data  
     JOIN dim_setor s  
     ON s.chave_setor=f.chave_setor  
GROUP BY ROLLUP (d.ano_data, s.nome_setor)
```

Resultado ROLLUP

ANO	SETOR	DESPESA TOTAL
2000	Limpeza	83618
2000	Lavanderia	92473,5
2000	Secretaria	74631
2000	Alimentação	135529,8
2000	Medicamentos	329780
2000		716032,3
2001	Limpeza	80571
2001	Lavanderia	126022,5
2001	Secretaria	87622,5
2001	Alimentação	141860,8
2001	Medicamentos	366360
2001		802436,8
2002	Limpeza	90396
2002	Lavanderia	108843
2002	Secretaria	66722
2002	Alimentação	146340,6
2002	Medicamentos	327750
2002		740051,6

Parte 1

Resultado ROLLUP

ANO	SETOR	DESPESA TOTAL
2003	Limpeza	83492,5
2003	Lavanderia	95205
2003	Secretaria	80564,5
2003	Alimentação	120127,8
2003	Medicamentos	254150
2003		633539,8
		2892060,5

Parte 2

CUBE

- Funcionalidade
 - criação de subtotais para todas as combinações da lista de agrupamento especificada na cláusula CUBE
 - criação do total geral
- Resultado
 - $2n$ níveis, sendo n o número de agrupamento de colunas

CUBE

- Processamento
 - cálculo dos valores agregados padrões especificados na cláusula GROUP BY
 - criação, de forma progressiva, de subtotais de nível mais alto, para todas as combinações de dimensões na lista de agrupamento de colunas
 - criação de um total geral

CUBE

- Quais as despesas do hospital em cada ano, separado pelos diferentes setores?

```
SELECT d.ano_data Ano, s.nome_setor Setor,  
       sum(f.CustoTotalProduto) DespesaTotal  
FROM f_custoporsetor f JOIN dim_data d  
     ON d.chave_data=f.chave_data  
     JOIN dim_setor s  
     ON s.chave_setor=f.chave_setor  
GROUP BY CUBE (d.ano_data, s.nome_setor)
```

Resultado CUBE

ANO	SETOR	DESPESA TOTAL
		2892060,5
	Limpeza	338077,5
	Lavanderia	422544
	Secretaria	309540
	Alimentação	543884
	Medicamentos	1278040
2000		716032,3
2000	Limpeza	83618
2000	Lavanderia	92473,5
2000	Secretaria	74631
2000	Alimentação	135529,8
2000	Medicamentos	329780
2001		802436,8
2001	Limpeza	80571
2001	Lavanderia	126022,5
2001	Secretaria	87622,5
2001	Alimentação	141860,8
2001	Medicamentos	366360

Parte 1

Resultado CUBE

ANO	SETOR	DESPESA TOTAL
2002		740051,6
2002	Limpeza	90396
2002	Lavanderia	108843
2002	Secretaria	66722
2002	Alimentação	146340,6
2002	Medicamentos	327750
2003		633539,8
2003	Limpeza	83492,5
2003	Lavanderia	95205
2003	Secretaria	80564,5
2003	Alimentação	120127,8
2003	Medicamentos	254150

Parte 2