

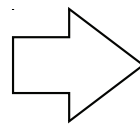
# SQL – Consultas Básicas

- Consulta a dados de uma tabela

```
select lista_atributos  
from tabela  
[where condição]
```

- Mapeamento para a álgebra relacional

```
select  $a_1, \dots, a_n$   
from  $t$   
where  $c$ 
```



$\pi_{a_1, \dots, a_n} (\sigma_c (t))$

# Comando SELECT

- Facilidades para **projeção** de informações
  - Não há eliminação de duplicatas no **Select**
    - **tabela**  $\equiv$  coleção
  - retorno de valores calculados
    - uso de operadores aritméticos (+, -, \*, /)
  - invocação de funções de agregação
    - **COUNT** (contador de ocorrências [de um atributo])
    - **MAX / MIN** (valores máximo / mínimo de um atributo)
    - **SUM** (somador de valores de um atributo)
    - **AVG** (média de valores de um atributo)

# Comando SELECT

- Eliminação de duplicatas

```
select [distinct] lista_atributos  
...
```

- Exemplo
  - buscar as especialidades dos médicos

```
select distinct especialidade  
from Médicos
```

# Comando SELECT

- Retorno de valores calculados - Exemplos
  - quantos grupos de 5 leitos podem ser formados em cada ambulatório?

```
select nroa, capacidade/5 as grupos5  
from Ambulatórios
```

$\equiv \rho_{(nroa, grupo5)} (\pi_{nroa, capacidade/5} (Ambulatórios))$

- qual o salário líquido dos funcionários (desc. 10%)?

```
select CPF, salário - (salário * 0.1) as  
líquido from Funcionários
```

# Comando SELECT

- Função COUNT - Exemplos

- informar o total de médicos ortopedistas

```
select count(*) as TotalOrtopedistas  
from Médicos  
where especialidade = 'ortopedia'
```

- total de médicos que atendem em ambulatorios

```
select count(nroa) as Total  
from Médicos
```

não conta nulos



# Comando SELECT

- Função SUM - Exemplo

- informar a capacidade total dos ambulatórios do primeiro andar

```
select sum(capacidade) as TotalAndar1  
from Ambulatórios  
where andar = 1
```

# Comando SELECT

- Função AVG - Exemplo

- informar a média de idade dos pacientes de Florianópolis

```
select avg(idade) as MediaPacFpolis  
from Pacientes  
where cidade = 'Florianópolis'
```

# Comando SELECT

- Funções MAX / MIN - Exemplo
  - informar o menor e o maior salário pagos aos Funcionários do departamento pessoal com mais de 50 anos

```
select min(salário) as mínimo,  
       max(salário) as máximo  
from Funcionários  
where depto = 'Pessoal'  
and idade > 50
```



# Comando SELECT

- Funções de Agregação com `distinct`
  - valores duplicados não são computados
  - exemplos

```
select count(distinct especialidade)
from Médicos
```

```
select avg(distinct salário)
from Funcionários
```

# Comando SELECT

- Observação sobre as funções de agregação
  - não podem ser combinadas a outros atributos da tabela no resultado da consulta

~~`select andar, COUNT (andar)  
from Ambulatórios`~~

# Cláusula WHERE

- Facilidades para **seleção** de dados
  - busca por padrões
    - cláusula **[NOT] LIKE**
  - teste de existência de valores nulos
    - cláusula **IS [NOT] NULL**
  - busca por intervalos de valores
    - cláusula **[NOT] BETWEEN *valor1* AND *valor2***
  - teste de pertinência elemento-conjunto
    - cláusula **[NOT] IN**

# Cláusula WHERE

- Busca por padrões

**where** atributo **like** '*padrão*'

% : casa com qq cadeia de caracteres

'\_' : casa com um único caractere

[a-f] : casa com qq caractere entre  
'a' e 'f' (SQL-Server)

- Exemplos

- buscar CPF e nome dos médicos com inicial M

```
select CPF, nome
```

```
from Médicos
```

```
where nome like 'M%'
```

# Cláusula WHERE

- Exemplos

- buscar nomes de pacientes cujo CPF termina com 20000 ou 30000

```
select nome  
from Pacientes  
where CPF like '%20000\  
or CPF like '%30000\'
```

- Observações

- em alguns dialetos SQL, '\*' é usado invés de '%'
- não é possível testar padrões em atributos *datetime* (SQL-Server)

# Cláusula WHERE

- Teste de valores nulos - Exemplo
  - buscar o CPF e o nome dos médicos que não dão atendimento em ambulatorios

```
select CPF, nome  
from Médicos  
where nroa is null
```

# Cláusula WHERE

- Busca por intervalos de valores - Exemplo
  - buscar os dados das consultas marcadas para o período da tarde

```
select *
```

```
from Consultas
```

```
where hora between '14:00' and '18:00'
```

# Cláusula WHERE

- Teste de pertinência elemento-conjunto -  
Exemplo
  - buscar os dados das médicos ortopedistas, traumatologistas e cardiologistas de Florianópolis

```
select *  
from Médicos  
where cidade = 'Florianópolis'  
and especialidade in ('cardiologia',  
                      'traumatologia',  
                      'cardiologia')
```



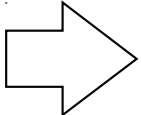
# SQL – DML

- Consultas envolvendo mais de uma tabela

```
select lista_atributos  
from tabela1, ..., tabelam  
[where condição]
```

- Mapeamento para a álgebra relacional

```
select a1, ..., an  
from tab1, ..., tabm  
where c
```



```
 $\pi_{a_1, \dots, a_n} (\sigma_c (tab_1 \times \dots \times tab_m))$ 
```

# Junção

- Sintaxe

```
select lista_atributos  
from tabela1 [inner] join tabela2 on  
    condição_junção [join tabela3 on ...]  
[where condição]
```

- Mapeamento para a álgebra relacional

```
select  $a_1, \dots, a_n$   
from  $tab_1$  join  $tab_2$   
on  $tab_1.x > tab_2.x$   
where  $c$ 
```

$\Rightarrow \pi_{a_1, \dots, a_n} (\sigma_c (tab_1 \theta X tab_2))$   
 $\theta = tab_1.x > tab_2.x$

# Junção Natural

- Sintaxe

```
select lista_atributos  
from tabela1 natural join tabela2  
[natural join tabela3 ...]  
[where condição]
```

- Mapeamento para a álgebra relacional

```
select  $a_1, \dots, a_n$   
from  $tab_1$  natural join  $tab_2$   
where  $c$ 
```

$\Rightarrow \pi_{a_1, \dots, a_n} (\sigma_c (tab_1 \bowtie tab_2))$

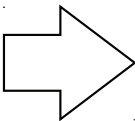
# Junções Externas (Não Naturais)

- Sintaxe

```
select lista_atributos  
from tabela1 left|right|full [outer] join  
    tabela2 on condição_junção  
    [join tabela3 on ...]  
[where condição]
```

- Mapeamento para a álgebra relacional

```
select a1, ..., an  
from t1 left join t2  
on t1.x > t2.x  
where c
```


$$\pi_{a_1, \dots, a_n} (\sigma_c (t_1 \theta \bowtie t_2))$$

$\theta = t_1.x > t_2.x$