

**BANCO DE DADOS HANDS-ON 12**  
**ANDRIEL MARK DA SILVA PINTO - 859093**

**1) CONSULTA SQL 1(SEM OTIMIZAR):**

```
SELECT A.CPF, A.Nome, B.Nome
FROM Funcionarios A, Clientes B, Aluguel C, Funcionarios D
WHERE A.CPF=B.CPF
AND B.CPF=C.CPF_Cliente
AND B.Sexo="M"
AND C.ValorPagar>50
AND A.CPF=D.CPF_Supervisor
```

**Algebra Relacional:**

```
 $\pi$  A.Nome, C.Nome
 $\sigma$  A.Codigo = B.CodFilme
 $\wedge$  B.CodAtor = C.Codigo
 $\wedge$  A.Genero = 'Aventura'
 $\wedge$  A.Codigo = D.CodFilme
 $\wedge$  D.PrecoDiaria > 10
(  $\rho$  A Filmes  $\times$   $\rho$  B AtoresEmFilmes )  $\times$   $\rho$  C Atores )  $\times$   $\rho$  D Midias
```

**1.1) Parsing em ordem natural:**

```
 $\pi$  (A.CPF, A.Nome, B.Nome)
|
 $\bowtie$  (A.CPF = D.CPF_Supervisor)
|
 $\bowtie$  (A.CPF = B.CPF)
|
 $\sigma$  (B.Sexo = "M")
|
 $\bowtie$  (B.CPF = C.CPF_Cliente)
|
 $\sigma$  (C.ValorPagar > 50)
|
```

-----

Funcionarios A	Clientes B	Aluguel C	Funcionarios D	

### 1.2) Parsing em ordem reversa:

$\pi$  (A.CPF, A.Nome, B.Nome)

|

$\bowtie$  (B.CPF = C.CPF\_Cliente)

|

$\sigma$  (C.ValorPagar > 50)

|

$\bowtie$  (A.CPF = B.CPF)

|

$\sigma$  (B.Sexo = "M")

|

$\bowtie$  (A.CPF = D.CPF\_Supervisor)

|

-----

|                    |                    |                    |  
Funcionarios A    Clientes B           Aluguel C           Funcionarios D

### 1.3) Árvore de consulta otimizada:

$\pi$  (A.CPF, A.Nome, B.Nome)

|

$\bowtie$  (A.CPF = D.CPF\_Supervisor)

|

$\bowtie$  (A.CPF = B.CPF)

|

$\bowtie$  (B.CPF = C.CPF\_Cliente)

|

-----

|                    |                    |  
 $\sigma$  (B.Sexo = "M")       $\sigma$  (C.ValorPagar > 50)  
|                    |                    |  
Clientes B    Aluguel C                    Funcionarios A, Funcionarios D

### 1.4) Consultas Reescritas:

#### 1.4.1) Consulta em ordem natural:

SELECT A.CPF, A.Nome, B.Nome

FROM Funcionarios A

JOIN Clientes B ON A.CPF = B.CPF

JOIN Aluguel C ON B.CPF = C.CPF\_Cliente

JOIN Funcionarios D ON A.CPF = D.CPF\_Supervisor

WHERE B.Sexo = "M"

AND C.ValorPagar > 50;

#### 1.4.2) Consulta em ordem reversa:

```
SELECT A.CPF, A.Nome, B.Nome
FROM Funcionarios D
JOIN Funcionarios A ON A.CPF = D.CPF_Supervisor
JOIN Clientes B ON A.CPF = B.CPF
JOIN Aluguel C ON B.CPF = C.CPF_Cliente
WHERE C.ValorPagar > 50
AND B.Sexo = "M";
```

#### 2) CONSULTA SQL 2(SEM OTIMIZAR):

```
SELECT A.Nome, C.Nome
FROM Filmes A, AtoresEmFilmes B, Atores C, Midias D
WHERE A.Codigo=B.CodFilme
AND B.CodAtor=C.Codigo
AND A.Genero="Aventura"
AND A.Codigo=D.CodFilme
AND D.PrecoDiaria>10
```

#### Algebra Relacional:

```
 $\pi$  A.Nome, C.Nome
 $\sigma$  A.Codigo = B.CodFilme
 $\wedge$  B.CodAtor = C.Codigo
 $\wedge$  A.Genero = 'Aventura'
 $\wedge$  A.Codigo = D.CodFilme
 $\wedge$  D.PrecoDiaria > 10
 $((\rho$  A Filmes  $\times \rho$  B AtoresEmFilmes)  $\times \rho$  C Atores )  $\times \rho$  D Midias
```

#### 2.1) Parsing em ordem natural:

```
 $\pi$  (A.Nome, C.Nome)
|
 $\bowtie$  (A.Codigo = D.CodFilme)
|
 $\sigma$  (D.PrecoDiaria > 10)
|
 $\bowtie$  (B.CodAtor = C.Codigo)
|
 $\sigma$  (A.Genero = "Aventura")
|
 $\bowtie$  (A.Codigo = B.CodFilme)
|
```

---

Filmes A	AtoresEmFilmes B	Atores C	Midias D	

## 2.2) Parsing em ordem reversa:

$\pi$  (A.Nome, C.Nome)

|

$\bowtie$  (A.Codigo = B.CodFilme)

|

$\sigma$  (A.Genero = "Aventura")

|

$\bowtie$  (B.CodAtor = C.Codigo)

|

$\sigma$  (D.PrecoDiaria > 10)

|

$\bowtie$  (A.Codigo = D.CodFilme)

|

-----

|                    |                    |                    |  
Filmes A    AtoresEmFilmes B    Atores C    Midias D

## 2.3) Árvore de consulta otimizada:

$\pi$  (A.Nome, C.Nome)

|

$\bowtie$  (B.CodAtor = C.Codigo)

|

$\bowtie$  (A.Codigo = B.CodFilme)

|

$\bowtie$  (A.Codigo = D.CodFilme)

|

$\sigma$  (A.Genero = "Aventura")    $\sigma$  (D.PrecoDiaria > 10)

|

-----

|                    |  
Filmes A            Midias D

## 2.4) Consultas Reescritas:

### 2.4.1) Consulta em ordem natural:

SELECT A.Nome, C.Nome

FROM Filmes A

JOIN AtoresEmFilmes B ON A.Codigo = B.CodFilme

JOIN Atores C ON B.CodAtor = C.Codigo

JOIN Midias D ON A.Codigo = D.CodFilme

WHERE A.Genero = "Aventura"

AND D.PrecoDiaria > 10;

#### 2.4.2) Consulta em ordem reversa:

```
SELECT A.Nome, C.Nome
FROM Midias D
JOIN Filmes A ON A.Codigo = D.CodFilme
JOIN AtoresEmFilmes B ON A.Codigo = B.CodFilme
JOIN Atores C ON B.CodAtor = C.Codigo
WHERE D.PrecoDiaria > 10
AND A.Genero = "Aventura";
```

#### 3) CONSULTA SQL 3(SEM OTIMIZAR):

```
SELECT A.CPF, A.Nome, B.Nome
FROM Funcionarios A, Clientes B, Aluguel C, Pagamentos D
WHERE A.CPF=B.CPF
AND C.ValorPagar>100
AND B.CPF=C.CPF_Cliente
AND D.Valor<50
AND A.CPF_Supervisor IS NULL
AND A.CPF=C.CPF_Funcionario
```

#### Algebra Relacional:

```
 $\pi$  A.CPF, A.Nome, B.Nome
 $\sigma$  A.CPF = B.CPF
AND C.ValorPagar > 100
AND B.CPF = C.CPF_Cliente
AND D.Valor < 50
AND A.CPF_Supervisor = NULL
AND A.CPF = C.CPF_Funcionario
((  $\rho$  A Funcionarios  $\times$   $\rho$  B Clientes )  $\times$   $\rho$  C Aluguel )  $\times$   $\rho$  D Pagamento)
```

#### 3.1) Parsing em ordem natural:

```
 $\pi$  (A.CPF, A.Nome, B.Nome)
|
 $\bowtie$  (A.CPF = C.CPF_Funcionario)
|
 $\sigma$  (A.CPF_Supervisor IS NULL)
|
 $\bowtie$  (A.CPF = B.CPF)
|
 $\sigma$  (D.Valor < 50)
|
 $\bowtie$  (B.CPF = C.CPF_Cliente)
|
 $\sigma$  (C.ValorPagar > 100)
```

-----				
Funcionarios A	Clientes B	Aluguel C	Pagamentos D	

### 3.2) Parsing em reversa:

$\pi$  (A.CPF, A.Nome, B.Nome)

|

$\bowtie$  (B.CPF = C.CPF\_Cliente)

|

$\sigma$  (C.ValorPagar > 100)

|

$\bowtie$  (A.CPF = B.CPF)

|

$\sigma$  (D.Valor < 50)

|

$\bowtie$  (A.CPF = C.CPF\_Funcionario)

|

$\sigma$  (A.CPF\_Supervisor IS NULL)

|

-----

Funcionarios A	Clientes B	Aluguel C	Pagamentos D	

### 3.3)Árvore de consulta otimizada:

$\pi$  (A.CPF, A.Nome, B.Nome)

|

$\bowtie$  (A.CPF = C.CPF\_Funcionario)

|

$\bowtie$  (A.CPF = B.CPF)

|

$\bowtie$  (B.CPF = C.CPF\_Cliente)

|

$\sigma$  (A.CPF\_Supervisor IS NULL)  $\sigma$  (C.ValorPagar > 100)  $\sigma$  (D.Valor < 50)

|

-----

Funcionarios A	Aluguel C	Pagamentos D	

### **3.4) Consultas Reescritas:**

#### **3.4.1) Consulta em ordem natural:**

```
SELECT A.CPF, A.Nome, B.Nome  
FROM Funcionarios A  
JOIN Clientes B ON A.CPF = B.CPF  
JOIN Aluguel C ON B.CPF = C.CPF_Cliente  
JOIN Pagamentos D ON A.CPF = C.CPF_Funcionario  
WHERE C.ValorPagar > 100  
      AND D.Valor < 50  
      AND A.CPF_Supervisor IS NULL;
```

#### **3.4.2) Consulta em ordem reversa:**

```
SELECT A.CPF, A.Nome, B.Nome  
FROM Pagamentos D  
JOIN Aluguel C ON D.CPF_Cliente = C.CPF_Funcionario  
JOIN Clientes B ON C.CPF_Cliente = B.CPF  
JOIN Funcionarios A ON A.CPF = B.CPF  
WHERE D.Valor < 50  
      AND C.ValorPagar > 100  
      AND A.CPF_Supervisor IS NULL;
```

