

### 1.1. Árvore de Consulta Inicial (não otimizada)

```
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;
```

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

#### a) Parsing em Ordem Natural (da Esquerda para a Direita)

```
 $\pi(A.CPF, A.Nome, B.Nome)$ 
|
 $\bowtie(A.CPF = D.CPF\_Supervisor)$ 
|
 $\bowtie(A.CPF = B.CPF)$ 
|
 $\sigma(B.Sexo = "M")$ 
|
 $\bowtie(B.CPF = C.CPF\_Cliente)$ 
|
 $\sigma(C.ValorPagar > 50)$ 
|
Funcionarios A, Clientes B, Aluguel C, Funcionarios D
```

**b) Parsing em Ordem Reversa (da Direita para a Esquerda)**

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

|

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

|

$\sigma(C.ValorPagar > 50)$

|

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

|

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

|

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

|

Funcionarios A, Clientes B, Aluguel C, Funcionarios D

**2.1. 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

### 3.1. Consulta Reescrita

#### d) Consulta Otimizada 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;
```

#### e) Consulta Otimizada 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";
```

----

## 1.2. Árvore de Consulta Inicial (não otimizada)

```
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;
```

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

### a) Parsing em Ordem Natural (da Esquerda para a Direita)

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

**b) Parsing em Ordem Reversa (da Direita para a Esquerda)**

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

|

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

|

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

|

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

|

$\sigma(D.PrecoDiaria > 10)$

|

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

|

Filmes A, AtoresEmFilmes B, Atores C, Midias D

**2.2. Árvore de Consulta Otimizada**

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

|

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

|

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

|

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

|

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

Filmes A

$\sigma(D.PrecoDiaria > 10)$

Midias D

### 3.2. Consulta Reescrita

#### d) Consulta Otimizada 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;
```

#### e) Consulta Otimizada 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";
```

---

### 1.3. Árvore de Consulta Inicial (não otimizada)

```
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;
```

$\pi A.CPF, A.Nome, B.Nome \sigma A.CPF = B.CPF \text{ and } C.ValorPagar > 100 \text{ and } B.CPF = C.CPF\_Cliente \text{ and } D.Valor < 50 \text{ and } A.CPF\_Supervisor = null \text{ and } A.CPF = C.CPF\_Funcionario ( ( ( \rho A Funcionarios \times \rho B Clientes ) \times \rho C Aluguel ) \times \rho D Pagamentos )$

#### a) Parsing em Ordem Natural (da Esquerda para a Direita)

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

|

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

|

$\sigma(A.CPF\_Supervisor \text{ IS NULL})$

|

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

|

$\sigma(D.Valor < 50)$

|

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

|

$\sigma(C.ValorPagar > 100)$

|

$\_ Funcionarios A, Clientes B, Aluguel C, Pagamentos D$

**b) Parsing em Ordem Reversa (da Direita para a Esquerda)**

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

|

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

|

$\sigma(C.ValorPagar > 100)$

|

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

|

$\sigma(D.Valor < 50)$

|

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

|

$\sigma(A.CPF\_Supervisor \text{ IS NULL})$

|

Funcionarios A, Clientes B, Aluguel C, Pagamentos D

**2.3. Árvore de Consulta Otimizada**

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

|

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

|

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

|

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

|

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

Funcionarios A

Aluguel C

Pagamentos D



### 3.3. Consulta Reescrita com Parsing

#### d) Parsing da Consulta Otimizada 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;
```

#### e) Parsing da Consulta Otimizada 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;
```

---

**FIM**