

Consulta 1 (SQL)

```
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
```

1. Árvore de Consulta Inicial (Ordem Natural)

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

2. Árvore de Consulta Inicial (Ordem Reversa)

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

3. Árvore de Consulta Otimizada

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

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

(Funcionarios D)

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

(Funcionarios A)

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

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

$(\sigma(C.ValorPagar>50))(Aluguel C)$

4. Consulta Reescrita (Ordem Natural) (SQL)

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

FROM (SELECT * FROM Funcionarios) A

JOIN (SELECT * FROM Clientes WHERE Sexo = 'M') B ON A.CPF = B.CPF

JOIN (SELECT * FROM Aluguel WHERE ValorPagar > 50) C ON B.CPF =
C.CPF_Cliente

JOIN (SELECT * FROM Funcionarios) D ON A.CPF = D.CPF_Supervisor

5. Consulta Reescrita (Ordem Reversa) (SQL)

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

FROM (SELECT * FROM Funcionarios) D

JOIN (SELECT * FROM Funcionarios) A ON A.CPF = D.CPF_Supervisor

JOIN (SELECT * FROM Clientes WHERE Sexo = 'M') B ON A.CPF = B.CPF

JOIN (SELECT * FROM Aluguel WHERE ValorPagar > 50) C ON B.CPF =
C.CPF_Cliente

6. Plano de Execução Otimizado

1. Executar seleção em Clientes (Sexo='M') usando índice secundário em Sexo
2. Executar seleção em Aluguel (ValorPagar>50) usando pesquisa linear (sem índice adequado)
3. Realizar junção entre resultados de 1 e 2 usando índice secundário em CPF_Cliente (Aluguel)
4. Realizar junção com Funcionarios A usando índice primário em CPF
5. Realizar junção com Funcionarios D usando índice secundário em CPF_Supervisor
6. Projetar os campos solicitados

Consulta 2 (SQL_=)

```
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
```

1. Árvore de Consulta Inicial (Ordem Natural)

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

2. Árvore de Consulta Inicial (Ordem Reversa)

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

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

(Mídias D)

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

(Atores C)

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

(AtoresEmFilmes B)

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

3. Árvore de Consulta Otimizada

Copy

Download

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

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

$(\sigma(D.PrecoDiaria>10))(Mídias D))$

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

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

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

(AtoresEmFilmes B)

(Atores C)

4. Consulta Reescrita (Ordem Natural) (SQL)

SELECT A.Nome, C.Nome

FROM (SELECT * FROM Filmes WHERE Genero = 'Aventura') A

JOIN AtoresEmFilmes B ON A.Codigo = B.CodFilme

JOIN Atores C ON B.CodAtor = C.Codigo

JOIN (SELECT * FROM Mídias WHERE PrecoDiaria > 10) D ON A.Codigo = D.CodFilme

5. Consulta Reescrita (Ordem Reversa) (SQL)

```
SELECT A.Nome, C.Nome  
FROM (SELECT * FROM Midias WHERE PrecoDiaria > 10) D  
JOIN (SELECT * FROM Filmes WHERE Genero = 'Aventura') A ON A.Codigo =  
D.CodFilme  
JOIN AtoresEmFilmes B ON A.Codigo = B.CodFilme  
JOIN Atores C ON B.CodAtor = C.Codigo
```

6. Plano de Execução Otimizado

1. Executar seleção em Filmes (Genero='Aventura') usando índice secundário em Genero
2. Executar seleção em Midias (PrecoDiaria>10) usando pesquisa linear (sem índice adequado)
3. Realizar junção entre resultados de 1 e 2 usando índice secundário em CodFilme (Midias)
4. Realizar junção com AtoresEmFilmes usando índice secundário em CodFilme
5. Realizar junção com Atores usando índice primário em Codigo
6. Projetar os campos solicitados

Consulta 3 (SQL)

```
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
```

1. Árvore de Consulta Inicial (Ordem Natural)

$\pi(A.CPF, A.Nome, B.Nome)$
 $\bowtie(A.CPF=C.CPF_Funcionario)$
 $\bowtie(D.Valor<50)$
 $\bowtie(B.CPF=C.CPF_Cliente)$
 $\bowtie(A.CPF=B.CPF)$
 $(\sigma(A.CPF_Supervisor \text{ IS NULL})(Funcionarios A))$
 $(Clientes B)$
 $(\sigma(C.ValorPagar>100)(Aluguel C))$
 $(Pagamentos D)$

2. Árvore de Consulta Inicial (Ordem Reversa)

$\pi(A.CPF, A.Nome, B.Nome)$
 $\bowtie(A.CPF=C.CPF_Funcionario)$
 $(\sigma(C.ValorPagar>100)(Aluguel C))$
 $\bowtie(D.Valor<50)$
 $(Pagamentos D)$
 $\bowtie(B.CPF=C.CPF_Cliente)$
 $\bowtie(A.CPF=B.CPF)$
 $(Clientes B)$
 $(\sigma(A.CPF_Supervisor \text{ IS NULL})(Funcionarios A))$

3. Árvore de Consulta Otimizada

$\pi(A.CPF, A.Nome, B.Nome)$
 $\bowtie(A.CPF=C.CPF_Funcionario)$
 $(\sigma(A.CPF_Supervisor \text{ IS NULL})(Funcionarios A))$
 $\bowtie(B.CPF=C.CPF_Cliente)$
 $(Clientes B)$
 $\bowtie(A.CPF=B.CPF)$
 $(\sigma(C.ValorPagar>100)(Aluguel C))$

$(\sigma(D.Valor < 50)(Pagamentos D))$

4. Consulta Reescrita (Ordem Natural) (SQL)

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

5. Consulta Reescrita (Ordem Reversa) (SQL)

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

6. Plano de Execução Otimizado

1. Executar seleção em Funcionarios (CPF_Supervisor IS NULL) usando pesquisa linear (sem índice)
2. Executar seleção em Aluguel (ValorPagar>100) usando pesquisa linear (sem índice adequado)
3. Executar seleção em Pagamentos (Valor<50) usando pesquisa linear (sem índice adequado)
4. Realizar junção entre Clientes e Funcionarios usando índice primário em CPF
5. Realizar junção com Aluguel usando índices secundários em CPF_Cliente e CPF_Funcionario
6. Realizar junção com Pagamentos (considerando que não há condição direta de junção)
7. Projetar os campos solicitados