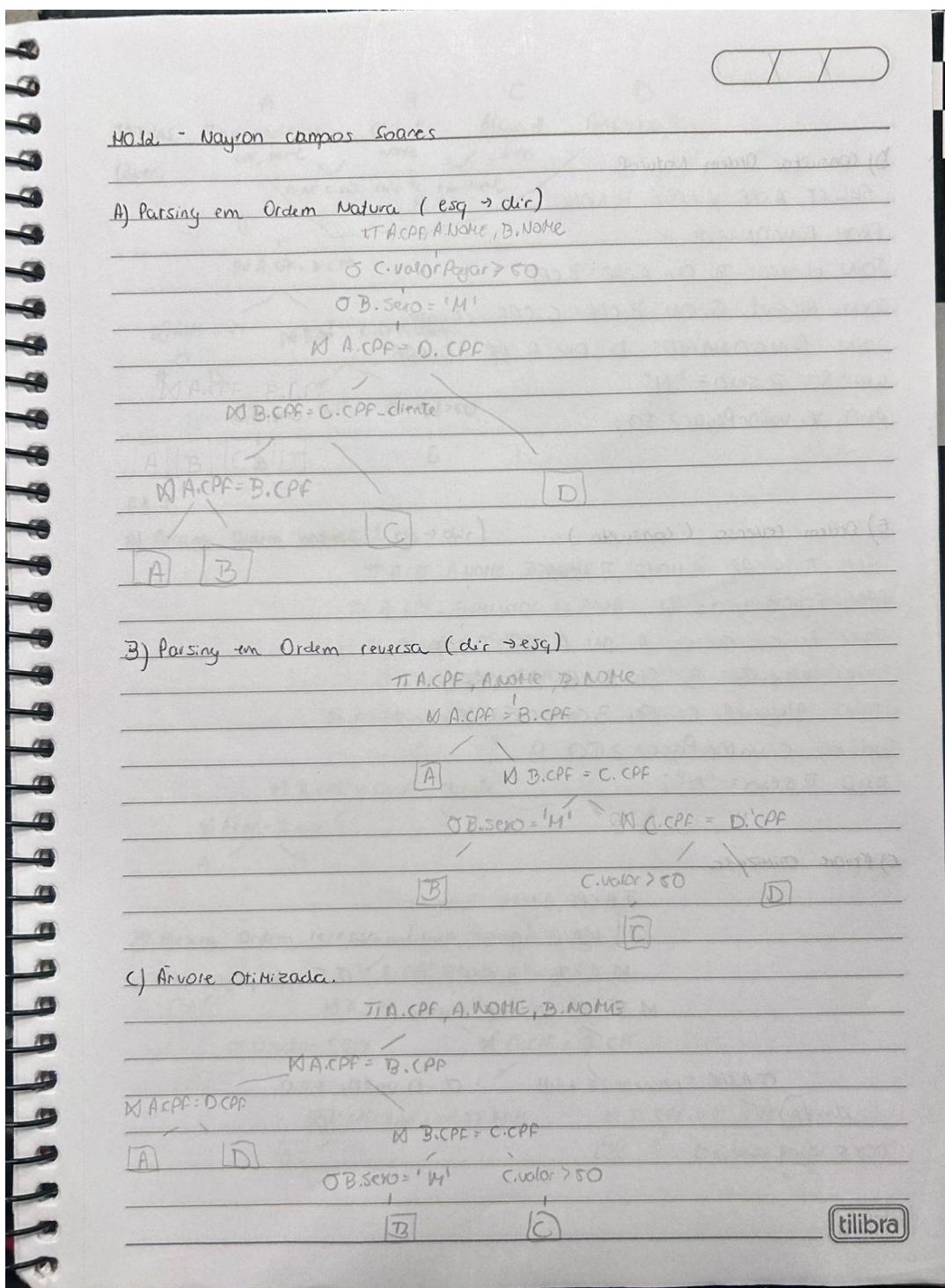


Ho 12 -

Nayron Campos Soares

Para desenho das árvores resolvi fazer no papel para facilitar o entendimento.

- 1- SELECT A.CPF, A.Nome, B.Nome
- 2- FROM Funcionarios A, Clientes B, Aluguel C, Funcionarios D
- 3- WHERE A.CPF=B.CPF
- 4- AND B.CPF=C.CPF_Cliente
- 5- AND B.Sexo="M"
- 6- AND C.ValorPagar>50
- 7- AND A.CPF=D.CPF_Supervisor



D) consulta 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  
WHERE B.Sexo = "M"  
AND C.ValorPagar > 50;
```

E) Ordem reversa (consueta)

```
SELECT A.CPF, A.NOME, B.NOME  
FROM FUNCIONARIOS D  
JOIN FUNCIONARIOS A ON A.CPF = D.CPF_supervisor  
JOIN CLIENTE B ON A.CPF = B.CPF  
JOIN ALUGUEL C ON B.CPF = C.CPF_cliente  
WHERE C.ValorPagar > 50  
AND B.Sexo = "M";
```

Ex 2:

1. SELECT A.Nome, C.Nome
2. FROM Filmes A, AtoresEmFilmes B, Atores C, Midias D
3. WHERE A.Codigo=B.CodFilme
4. AND B.CodAtor=C.Codigo
5. AND A.Genero="Aventura"
6. AND A.Codigo=D.CodFilme
7. AND D.PrecoDiaria>10

Ex 2:

Sentença: Mário é meu

A) Parsing natural (esq \rightarrow dir)

T A. NOME, C. NOME

O D. Presodaria > 10

M A. código = D. cod/f/me

/ \ [D]

O A. gênero = "Aventura"

M B. cod/f/me = C. código

/ \ [C]

(meu) é meu

M A. cod/f/me = B. cod/f/me

[A] / \ [B]

B) Parsing natural reverso (dir \rightarrow esq)

T A. NOME, C. NOME

M B. cod/f/me = C. código

M A. cod = B. cod

M A. cod = D. cod

O A. gênero = "Aventura" [B]

[A] O D. Presodaria > 10

(meu) é meu

é meu é meu

C) Árvore Optimizada

T A. NOME, C. NOME

M B. cod/f/me = C. código

/ \ [C]

M A. cod/f/me = B. cod/f/me

[B]

M A. cod/f/me = D. cod/f/me

/ \ [D]

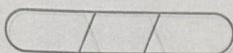
O A. gênero = "Aventura"

[A]

O D. Presodaria > 10

[D]

tilibra



D) consulta 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;
```

c) Ordem reversa (consueta)

```
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"
```

Ex3:

1. SELECT A.CPF, A.Nome, B.Nome
2. FROM Funcionarios A, Clientes B, Aluguel C, Pagamentos D
3. WHERE A.CPF=B.CPF
4. AND C.ValorPagar>100
5. AND B.CPF=C.CPF_Cliente
6. AND D.Valor<50
7. AND A.CPF_Supervisor IS NULL
8. AND A.CPF=C.CPF_Funcionario

Ex 3

A) Parsing Ordem natural (esq → dir)

π A.CPF, A.NOME, B.NOME

σ A.CPF = supervisor IS NULL

σ D.valor < 50

β (σ A.CPF = supervisor IS NULL) AND σ A.NOME = B.NOME

κ A.CPF = C.CPF - Funcionario

/ \ 0

κ B.CPF = C.CPF - Cliente

κ A.CPF = B.CPF / \ C

A / \ B

B) Parsing Ordem reversa (dir → esq)

π A.CPF, A.NOME, B.NOME

κ A.CPF = C.CPF - Funcionario

σ D.valor < 50 / \ κ A.CPF = B.CPF

D / \

σ A.CPF = supervisor IS NULL

A / \

κ B.CPF = C.CPF - Cliente

B / \ σ C.valor > 100

C / \

tilibra

C) Árvore otimizada.

π A.CPF, A.NOME, B.NOME

κ A.CPF = C.CPF - Funcionario

κ A.CPF = B.CPF

σ C.valor > 100 / \

κ B.CPF = C.CPF - Cliente

C / \

σ A.CPF = supervisor IS NULL

A / \

σ D.valor < 50

D / \

tilibra

D) consulta Ordem natural:	E) Ordem reversa (consulta)
<pre> 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 Pagamento D ON A.CPF = C.CPF_funcionario WHERE C.valor_pagar > 100 AND D.valor < 50 AND A.CPF_supervisor IS NULL; </pre>	<pre> SELECT A.CPF, A.NOME, B.NOME, D.valor FROM Pagamento 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.valor_pagar > 100 AND A.CPF_supervisor IS NULL; </pre>
tilibra	

OBS: Não consegui entender muito e não sei se fiz corretamente!