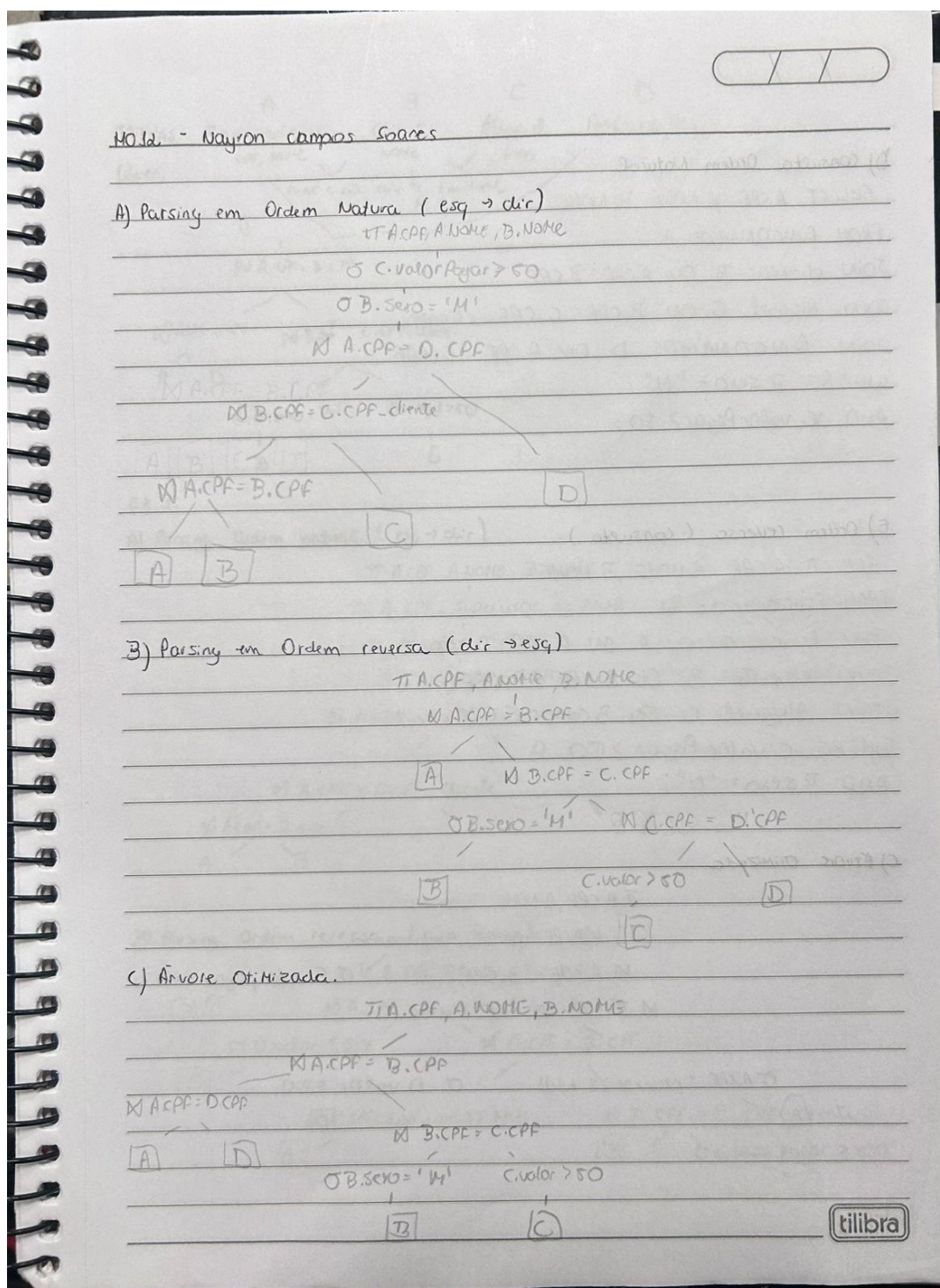


Ho 12 -

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 (sub = pag.)  
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.valorPagor > 50;
```

E) Ordem reversa (consulta)

```
SELECT A.CPF, A.NOME, B.NOME  
FROM funcionarios D  
JOIN funcionarios A ON A.CPF = D.CPF > Supervisor (E)  
JOIN cliente B ON A.CPF = B.CPF  
JOIN Aluguel C ON B.CPF = C.CPF_cliente  
WHERE C.valorPagor > 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:

A) Parsing Natural (esq \rightarrow dir)

π A.NOME, C.NOME

σ D.Preco > 10

\bowtie A.codigo = D.codigo

σ A.genero = "Aventura"

\bowtie B.codigo = C.codigo

\bowtie A.codigo = B.codigo

B) Parsing Natural reverso (dir \rightarrow esq)

π A.NOME, C.NOME

\bowtie B.codigo = C.codigo

\bowtie A.codigo = B.codigo

\bowtie A.codigo = D.codigo

σ A.genero = "Aventura"

σ D.Preco > 10

C) Árvore Optimizada

π A.NOME, C.NOME

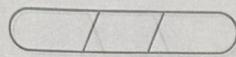
\bowtie B.codigo = C.codigo

\bowtie A.codigo = B.codigo

\bowtie A.codigo = D.codigo

σ A.genero = "Aventura"

σ D.Preco > 10



D) Consulta Ordem Natural

```
SELECT A.NOME, C.NOME  
FROM FILMES A  
JOIN Atores em filmes B ON A.codigo = B.codfilme  
JOIN Atores C ON B.codautor = C.codigo  
JOIN MIDIAS D ON A.codigo = D.codfilme  
WHERE A.Genero = "Aventura"  
AND D.Preco diaria > 10;
```

E) Ordem reversa (consulta)

```
SELECT A.NOME, C.NOME  
FROM Midias D  
JOIN Filmes A ON A.codigo = D.codfilme  
JOIN Atores em filmes B ON A.codigo = B.codfilme  
JOIN Atores C ON B.codautor = C.codigo  
WHERE D.Preco diaria > 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

σ C.valor pagar > 100

\bowtie A.CPF = C.CPF = funcionario

\bowtie B.CPF = C.CPF = cliente

\bowtie A.CPF = B.CPF

A B

B) Parsing Ordem reversa (dir → esq)

π A.CPF, A.NOME, B.NOME

\bowtie A.CPF = C.CPF = funcionario

σ D.valor < 50

\bowtie A.CPF = B.CPF

D

σ A.CPF = supervisor IS NULL

\bowtie B.CPF = C.CPF = cliente

A

B

σ C.valor pagar > 100

C

tilibra

C) Alvore Otimizada

π A.CPF, A.NOME, B.NOME

\bowtie A.CPF = C.CPF = funcionario

\bowtie A.CPF = B.CPF

σ C.valor pagar > 100

\bowtie B.CPF = C.CPF = cliente

σ A.CPF = supervisor IS NULL

σ D.valor < 50

A

D

tilibra

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 pagamento D ON  
A.CPF = C.CPF_funcionario  
WHERE C.valor_pagar > 100  
AND D.valor < 50  
AND A.CPF_supervisor IS NULL;
```

tilibra

E) Ordem reversa (consulta)

```
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.valor_pagar > 100  
AND A.CPF_supervisor IS NULL;
```

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