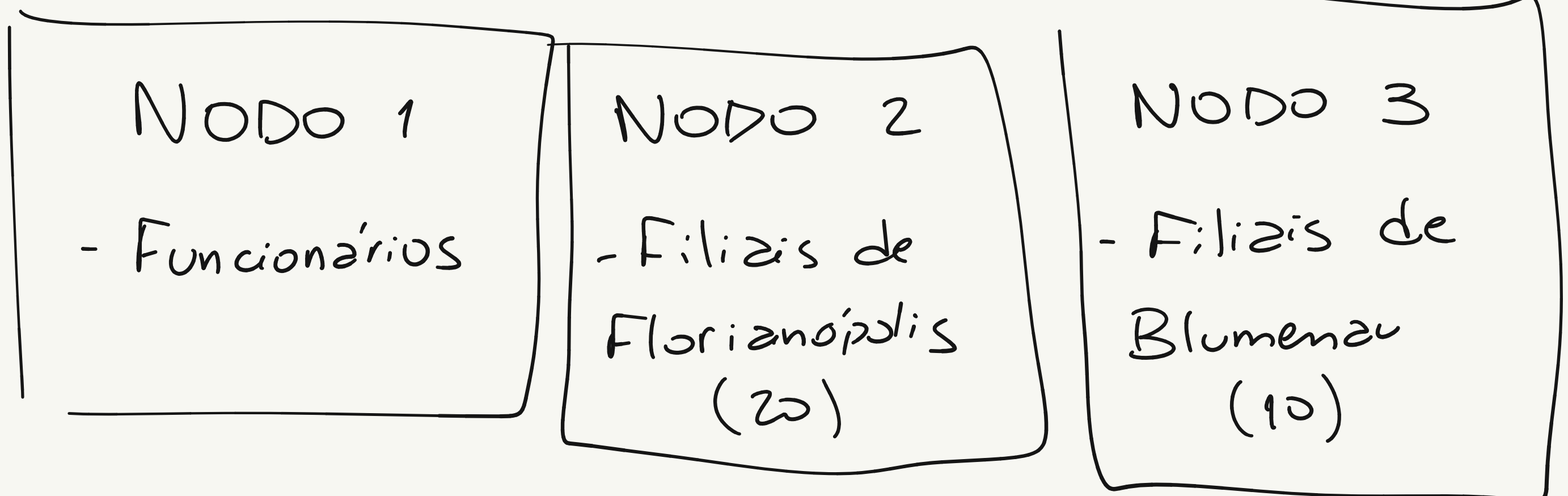
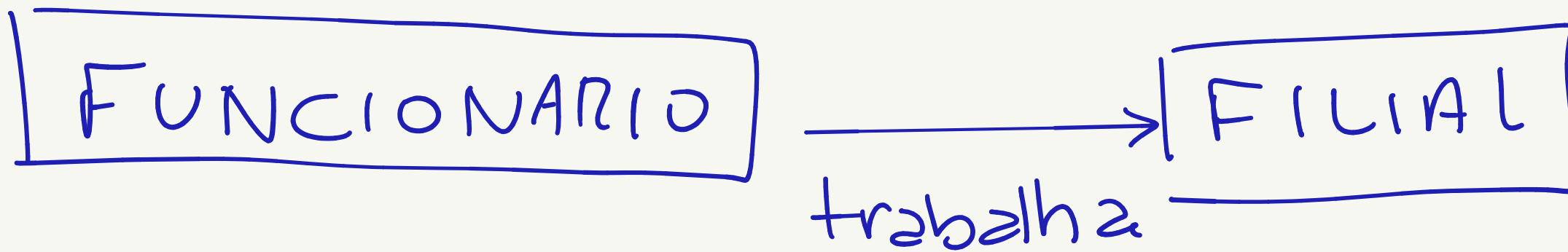


4.3



2)

Nodo 1.F ← Funcionário

Nodo 2.F ← $\sigma_{city: 'Florianópolis'}$ (Filial)

Nodo 3.F ← $\sigma_{city: 'Blumenau'}$ (Filial)

b.1) CREATE VIEW Nod1p2 AS
SELECT F.codF, F.nome, F.nroFilial

FROM Nod1.F

NATURAL JOIN Nod2.P

N
O
D
O

1 CREATE VIEW Nod1p3 AS

SELECT F.codF, F.nome, F.nroFilial

FROM Nod1.F

NATURAL JOIN Nod3.P

N
O
D
O
2

```
CREATE VIEW Nod2.P AS
SELECT Cdf
FROM Nod2.F
```

```
SELECT * FROM Nod1p2
UNION Nod1p3
```

N
O
D
3

```
CREATE VIEW Nod3.P AS
SELECT Cdf FROM
Nod3.F
```

b.2) CREATE VIEW Nod1.Func AS (Nod1)
SELECT Cdf, nome, noFilial
FROM Nod1.P

```
CREATE VIEW Nod2p1 AS
SELECT F.cdf, F.nome, F.noFilial
FROM Nod1.Func
NATURAL JOIN Nod2.P (Nod2)
```

```
CREATE VIEW Nod2p3 AS
```

```
SELECT F.codF, F.nome, F.noFilial (Nodo 2)
```

```
FROM Nod1.Func
```

```
NATURAL JOIN Nod3.F
```

```
CREATE VIEW Nod2.Filial AS
```

```
SELECT codF
```

```
FROM Nod3.P
```

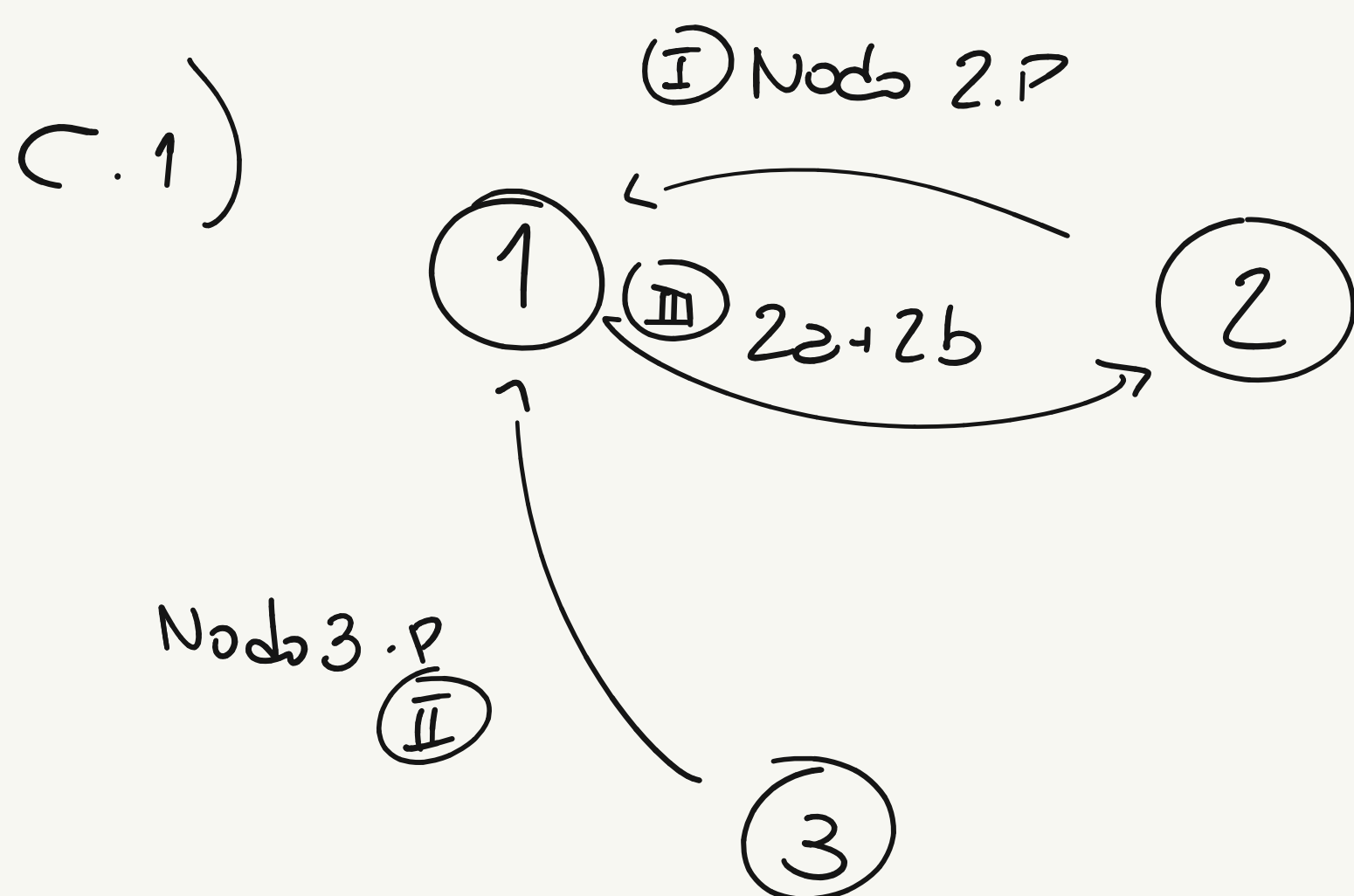
(Nodo 3)

```
CREATE VIEW Nod2.fim AS
```

```
SELECT *
```

```
FROM Nod2p1
```

```
UNION Nod2p3
```



I) 20 - |codF|

II) 10. |codF|

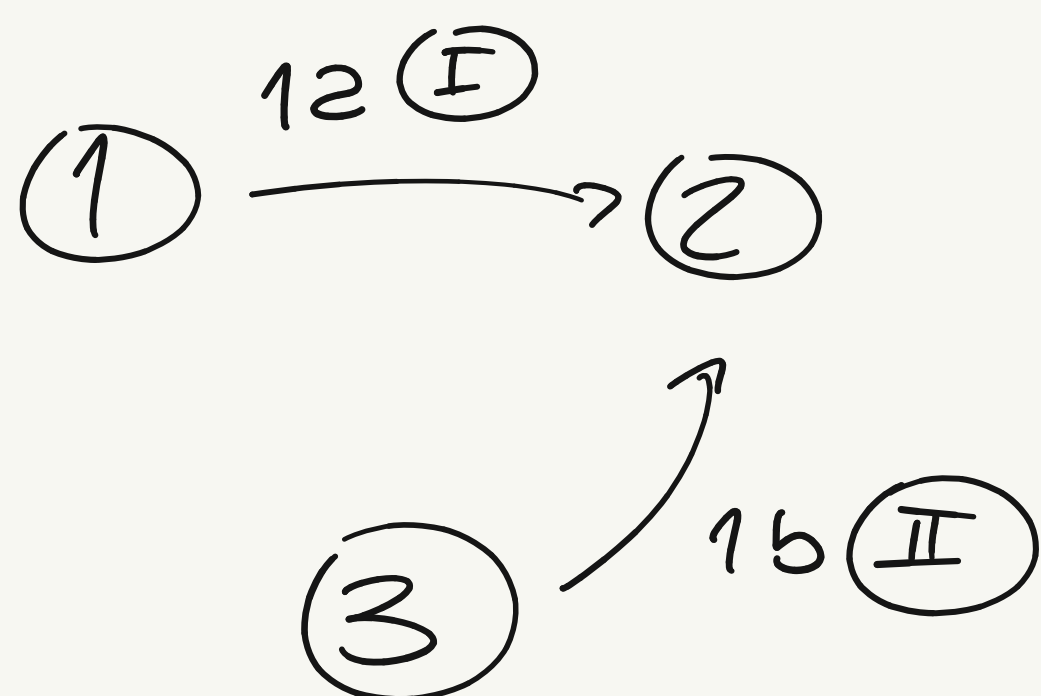
2a) K. florip2. (|codF| + |nome| + |noFilial|)

2b) K. blumenzu. (|codF| + (nome| + |noFilial|))

Total =

$$30 \cdot |\text{cod F}| + (K \cdot \text{floipa} + K \cdot \text{blumenau}) (|\text{cod F}| + |\text{nome}| + |\text{nr Filial}|)$$

C.2)



I) $K (|\text{cod F}| + |\text{nome}| + |\text{nr Filial}|)$

II) $10 \cdot |\text{cod F}|$

Total: $K \cdot (|\text{cod F}| + |\text{nome}| + |\text{nr Filial}|) + 10 \cdot |\text{cod F}|$

↓) Na realidade mostrada, de existirem filiais apenas em Flo e Blu, a alternativa 2 é a melhor.

Caso existam outras filiais, a alternativa 1 passa a ser mais eficiente, já que ele evita toda a transmissão dos dados dos funcionários.