

# E2

- a consulta submetida em 25/09/2020;
- País.nome chave candidata;
- bem menos emigrações do Brasil em toda a história mantida no banco de dados do que emigrações de quaisquer outras partes do mundo desde '01/01/2020';
- bem menos indíviduos emigrando do Brasil desde '01/01/2020' do que nascidos em todo o mundo desde '01/01/2000';
- muitas imigrações para certos países;
- índice primário para cada tabela (pelo respectivo campo id), índice secundário por cada chave estrangeira, I.dtNasc, M.dtMigra e País.nome.

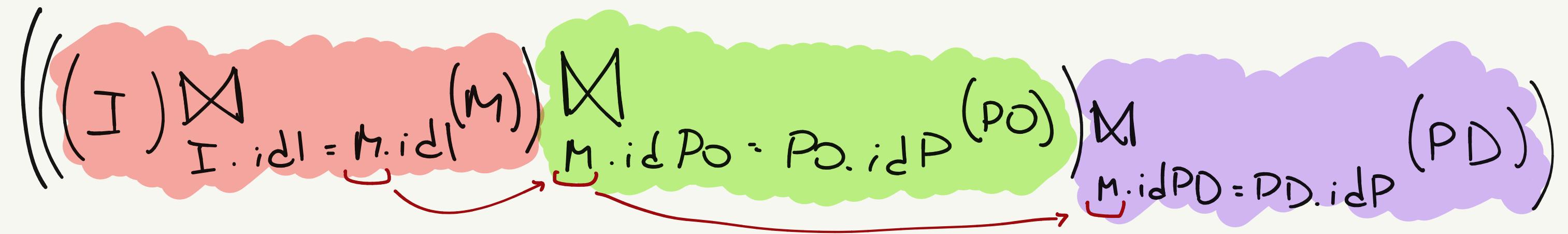
Indique algoritmos eficientes para executar (grupos de) operadores da estratégia proposta, em termos do número de tuplas em cada tabela ( $|Pais|$ ,  $|M|$  e  $|I|$ ) e dos tamanhos dos resultados de cada um desses algoritmos.

SELECT PD.\* , COUNT(M.\*) AS n

```
FROM Individuo I INNER JOIN Migra M ON I.idI = M.idI INNER JOIN Pais PO ON M.idPO = PO.idP INNER JOIN Pais AS PD ON M.idPD  
= PD.idP  
WHERE I.dtNasc >= '01/01/2000' AND M.dtMigra >= '01/01/2020' AND PO.nome = 'Brasil'  
GROUP BY PD.* HAVING n > 10000 ORDER BY n DESCENDING ;
```

Obs: Na tabela 'Migra':  
idPO = país origem  
idPD = país destino

"From":



→ Heurísticas a lembrar:

- Associação de  $\bowtie$

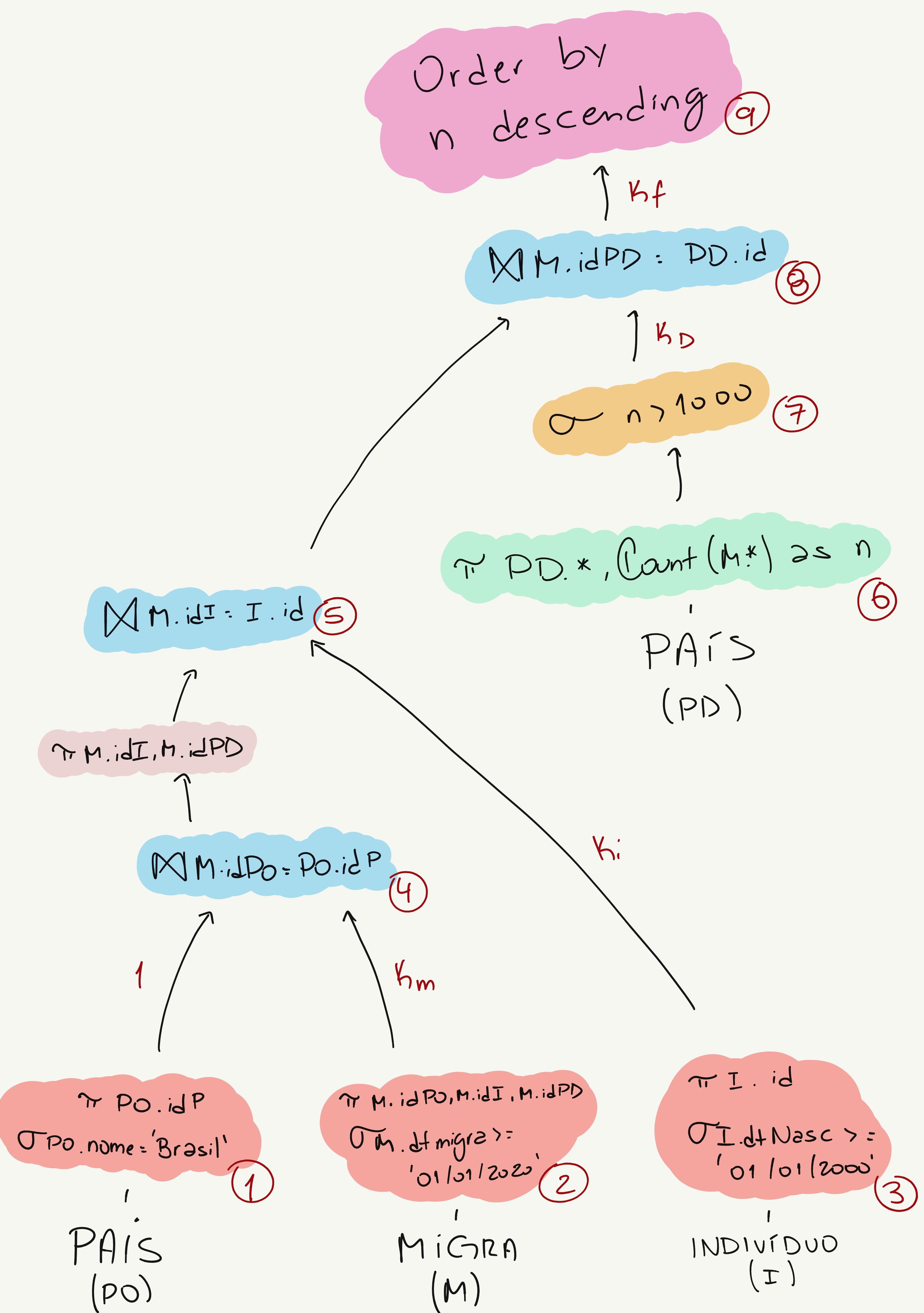
$$(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$$

- Comutatividade de  $\bowtie$  com  $\sigma$

$$\sigma_c(R \bowtie S) = \sigma_c(R) \bowtie S$$

→ Se  $c$  envolve apenas atributos de 1 tabela

# E2



Busca por índice  $[O(\log |N|)]$

1.  $O(\log |P_D|)$

2.  $O(\log |M|)$

3.  $O(\log |I|)$

Junção por igualdade (hash join)  
 $[O(\log R+S)]$

4.  $O(\log K_m+1)$

5.  $O(\log K_m + K_i)$

8.  $O(\log K_m + K_I + K_D)$

Contagem (passar por todos)  $[O(|R|)]$

6.  $O(|P_D|)$

Algoritmo semelhante = hash join  $[O(N)]$

7.  $O(|P_D|)$

Merge Sort  $[O(R \cdot \log R)]$

9.  $O(K_f \cdot \log K_f)$