

4.5

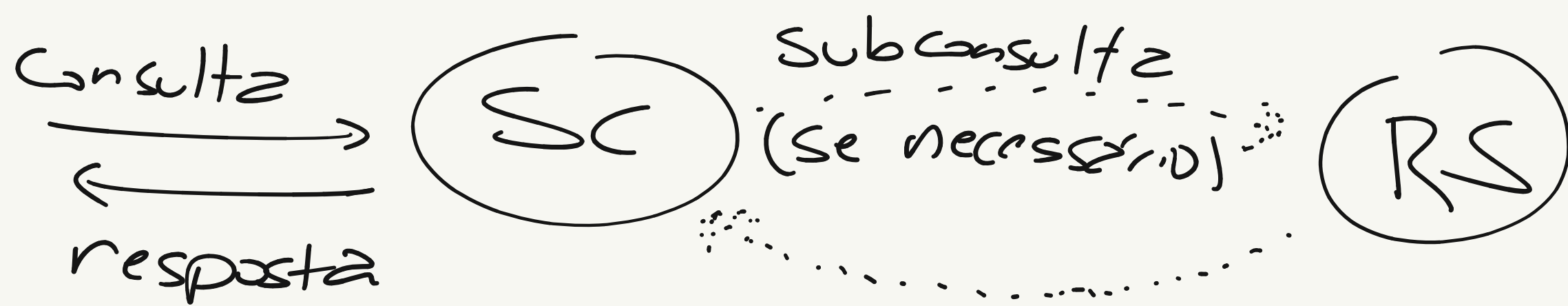
2) SC e RS

- 1) SC.Alunos $\leftarrow \sigma_{UF='SC'} (Aluno)$
- 2) SC.Cursa $\leftarrow \pi_{Cursa.*} (SC.Aluno \times Cursa)$
- 3) SC.Disciplina $\leftarrow Disciplina$
RS.Disciplina $\leftarrow Disciplina$
RJ.Disciplina $\leftarrow Disciplina$
PE.Disciplina $\leftarrow Disciplina$
DF.Disciplina $\leftarrow Disciplina$

- 4) RS.Alunos $\leftarrow Aluno$
RJ.Alunos $\leftarrow Aluno$
PE.Alunos $\leftarrow Aluno$
DF.Alunos $\leftarrow Aluno$
RS.Cursa $\leftarrow Cursa$
RJ.Cursa $\leftarrow Cursa$
PE.Cursa $\leftarrow Cursa$
DF.Cursa $\leftarrow Cursa$

b) Consultas submetidas no nó SC.

Deve acionar apenas SC e RS.



```
SELECT D.*, COUNT(DISTINCT A.*) AS  
nAlunos, AVG(notas) AS mNotas  
FROM Aluno A NATURAL JOIN  
DISCIPLINA D
```

```
GROUP BY D.*
```

```
HAVING nAlunos > 1000 AND mNotas < 7
```

```
ORDER BY mNotas ASCENDING  
nAlunos DESCENDING
```

Nó RS

→ Pegar todas as infos que não estejam
no nó SC (pl diminuir o payload)

Nó SC

→ "Mesclar" (join) com o que for recebido
de RS

(RS)

```
CREATE VIEW RS.Result AS  
SELECT D.*, COUNT(DISTINCT A.*) AS  
    nAlunos, AVG(nota) as mNotas,  
    D.idD → Necessário pro JOIN em SC.
```

```
FROM Aluno A NATURAL JOIN  
CURSA C NATURAL JOIN  
DISCIPLINA D
```

```
GROUP BY D.idD
```

```
HAVING nAlunos > 1000 AND mNotas < 7
```

```
ORDER BY mNotas ASCENDING  
nAlunos DESCENDING
```

(SC)

```
CREATE VIEW SC.Result AS
```

```
SELECT *
```

```
FROM SC.Disciplina SCD
```

```
INNER JOIN RS.Result RSR
```

```
ON SCD.idD = RSR.idD
```

```
ORDER BY mNotas ASCENDING  
nAlunos DESCENDING
```