

Operação de Divisão

- ▶ Propriedade
 - ▶ Seja $q = r \div s$
 - ▶ Então q é a maior relação satisfazendo $q \times s \subseteq r$.
- ▶ Definição em termos de operações básicas da álgebra rel.
Sejam $r(R)$ e $s(S)$ relações, com $S \subset R$

$$r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r))$$

Porquê?

- ▶ $\Pi_{R-S}(r) \times s$ dá os elementos de r com todos os valores de S .
- ▶ $\Pi_{R-S,S}(r)$ construi uma versão de r com os atributos da expressão anterior.
- ▶ $\Pi_{R-S}(\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r)$ dá os tuplos t em $\Pi_{R-S}(r)$ tal que para algum tuplo $u \in s$, $tu \notin r$.

MySQL — Exemplos de Consultas

- ▶ Que tipos de lojas estão presentes em todas as cidades?

```
SELECT DISTINCT store_type FROM stores s1
WHERE NOT EXISTS (
  SELECT * FROM cities WHERE NOT EXISTS (
    SELECT * FROM cities_stores
    WHERE cities_stores.city = cities.city
    AND cities_stores.store_type = stores.store_type));
```

Este último exame é uma consulta “NOT EXISTS” duplamente aninhada. Isto é, contém uma cláusula “NOT EXISTS” dentro de uma outra cláusula do mesmo tipo. Formalmente, responde à questão “existe uma cidade com uma loja que não está em Lojas?”. É mais fácil de ver que uma consulta “NOT EXISTS” duplamente aninhada responde à questão “é x verdade para todo o y ?”

MySQL — Exemplos de Consultas

Uma aproximação às consultas do tipo \div (do manual do MySQL, 13.2.8.6)

- ▶ Que tipos de lojas estão presentes em uma, ou mais, cidades?

```
SELECT DISTINCT tipos_lojas FROM Lojas
WHERE EXISTS (SELECT * FROM cidades_lojas
  WHERE cidades_lojas.tipo_loja = lojas.tipos_lojas);
```

- ▶ Que tipos de lojas estão presentes em nenhuma cidade?

```
SELECT DISTINCT tipo_loja FROM lojas
WHERE NOT EXISTS (SELECT * FROM cidades_lojas
  WHERE cidades_lojas.tipo_lojas = lojas.tipo_lojas);
```