

EX3: all countries (names) with

- > 1.000.000 inhabitants
- not member of the 'EU'

$$F(N) \equiv \exists C, A, P, Gp, GpProv :$$

$$country(C, N, A, P, Gp, GpProv) \wedge P > 1000000$$

- safe ✓  
- RANF no!

$\wedge \neg \exists T : isMember(C, 'EU', T)$

free = {C, T}  
π = {C, T}

free = {C}  
π = ∅  
⇒ C is missing

⇒ push into  $\neg \exists$

sufficient:  
C is in the active domain:

Mai 27-10:15

$$F(N) \equiv \exists C, A, P, Gp, GpProv :$$

$$country(C, N, A, P, Gp, GpProv) \wedge P > 1000000$$

$\pi[Name] \wedge \neg \exists T : isMember(C, 'EU', T)$

$\sigma[pop > 1000000]$

→ Alpha

adom(C) ∧

$\sigma[\$2 \rightarrow name]$

Country

$\sigma[\$2]$

$\pi[\$2]$

$\sigma[\$2 = 'EU']$

$\sigma[\$2 = 'EU']$

$\pi[country](country)$

isMember

$\pi[name]$

inernal evaluation

Country

$\sigma[ag = 'EU']$

isMember

Country

$\sigma[ag = 'EU']$

isMember

"Anti-join-operator"  
"∃"

Mai 27-10:34

Exercise 4:  $\tau(A|B), s(AB)$

$$\tau := \{ \mu \in \text{Type}(A) \mid \{ \mu \} \times S \subseteq \tau \}$$

$$F(x) \equiv \text{adom}(x) \wedge \forall y: (s(y) \rightarrow \tau(x, y))$$

instead of adom, take first col of  $\tau$

$$\equiv \exists z: r(x, z) \wedge \forall y: (s(y) \rightarrow \tau(x, y))$$

... translate it to Relational Algebra via SRNF and RANF

$$\equiv \exists z: r(x, z) \wedge \forall y: (\neg s(y) \vee \tau(x, y))$$

$$\equiv \exists z: r(x, z) \wedge \neg \exists y: (s(y) \wedge \neg \tau(x, y))$$

is in SRNF  $\leftarrow$  cf. SQL: not exists.. not exists

Free = {x}  $\rightarrow$  not in RANF  
 $\tau = \beta$  the x is not  $\tau$ .

"push it to not exists"

$$\equiv \exists z: r(x, z) \wedge \neg \exists y: (\exists z: r(x, z) \wedge s(y) \wedge \neg \tau(x, y))$$

SRNF  $\uparrow$

Mai 27-10:45

Ex. 5

```

select country.name, sum(city.population)
from city, country
where city.country=country.code
group by city.country, country.name
    
```

need: - a new variable, bound to the sum of city populations in that country

$$F(N, S) \equiv S = \text{sum}(\dots) \wedge \text{country}(C, N, CP, A, GP, GP2)$$

$$\equiv S = \text{sum}\{\text{values}\} \wedge \dots$$

$$\equiv S = \text{sum}\{P \mid P \text{ is a pop of a city in } C\}$$

$$\equiv S = \text{sum}\{P \mid \text{city}(C, N, Pop, P, \dots)\} \wedge \text{country}(C, N, CP, A, GP, GP2)$$

Relay:  $S$  is  $\text{sum}\{\dots\}$

in general: Definition of Aggregation Operators

- aggregation function (sum, max, count, avg, ...)
- applied to a set/collection of values/tuples

$\rightarrow$  defined inductively

$$\text{op}\{\dots\} := \dots$$

$$\text{sum}\{\dots\} := \dots$$

$$\text{sum}(\emptyset) = 0$$

$$\text{sum}\{x\} \cup S = x + \text{sum}(S)$$

in De log/Prolog: Lists: head.tail

$$[a [b [c []]]]$$

... soon next slides

Mai 27-10:54

Slide 363

↑ RESULTS T[org]  
6[Pop > 1000000] ist member  
6[Org = 'EU' & Code = '9Country']

