

Ex 2 Ex 1:

null-tolerant join

eg. $R(A, B, C)$, $S(A, B, D)$

R:

A	B	C
a_1	b_1	c_1
a_2	null	c_2

S:

A	B	D
a_1	b_1	d_1
a_2	b_2	d_2
a_3	null	d_3

Rel. Alg

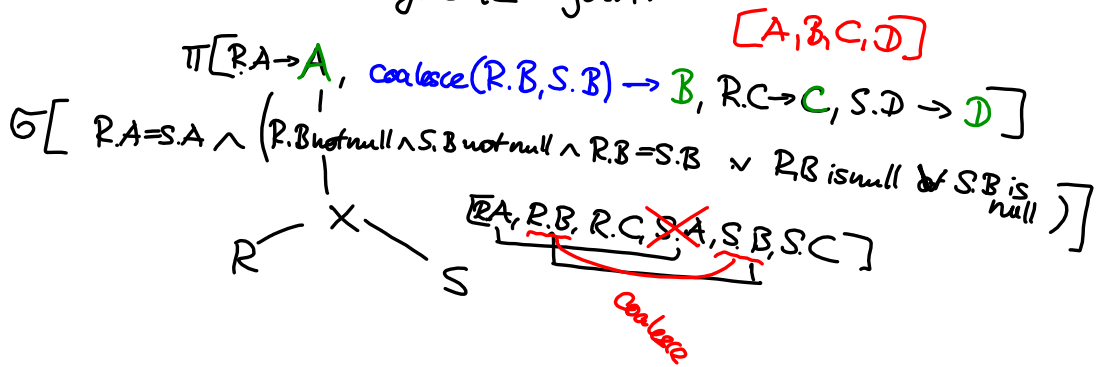
$\Rightarrow (a_1, b_1, c_1, d_1)$
and not more : $(a_2, \text{null}) \neq (a_2, b_2)$

SPARQL:

M_1 and v_1 compatible
 M_2 and v_2 are also compatible $\rightarrow (a_2, b_2, c_2, d_2)$

Dez 4-14:17

\Rightarrow null tolerant relAlg/SQL join:



Dez 4-14:29

b) $R \setminus_{\text{SQL}} S \equiv R$ where not exists

SQL:
 (Select *
 from S
 where tuple in r is compatible
 with tuple in S)

rel Algebra: $\hat{=}$ minus
 $R(X), S(Y), \bar{Z} := \bar{X} \cap \bar{Y}$

$R \setminus X \xrightarrow{\text{minus (rel)}} \pi[\bar{Z}](R) \quad \pi[\bar{Z}](S)$

$\hat{=} \text{SPARQL} \quad R \setminus S$

consider null values!!

Dez 4-14:36

c) SPARQL Algebra: $P_2 \text{ MINUS } P_1$

$\hat{=} \text{SPARQL 1.1}$:
 select ...
 from ...
 where { P_1
Filter not exists P_2 }

SPARQL 1.0:
 select ...
 where P_1 opt P_2
 filter (!bound(y))

what if P_1 and P_2 have the same variables

prefix mon: <<http://www.semwebtech.org/mondial/10/meta#>>

```

SELECT ?C ?X
FROM <file:mondial-europe.n3>
WHERE { ?C a mon:Country; mon:hasCity ?X
  optional { ?C mon:capital ?X ; mon:name ?Y }
  filter (!bound(?Y))
}
    
```

Dez 4-14:49

$$d) R \bowtie S = \underbrace{R \bowtie S}_{\text{inner join}} \cup R \setminus S$$

Ex2: Full outer join in SPARQL

$$R \bowtie S = R \bowtie S \cup R \setminus S \cup (R \bowtie S) \cup S \setminus R$$

Setsquaries → no duplicates ✓
Union
⇒ SPARQL ?

prefix mon: <<http://www.semwebtech.org/mondial/10/meta#>>

SELECT ?CN ?RN

FROM <file:mondial.n3>

WHERE {{ ?C a mon:Country; mon:capital ?X . ?X mon:name ?CN

optional {?X mon:locatedAt ?W . ?W a mon:River; mon:name ?RN} }

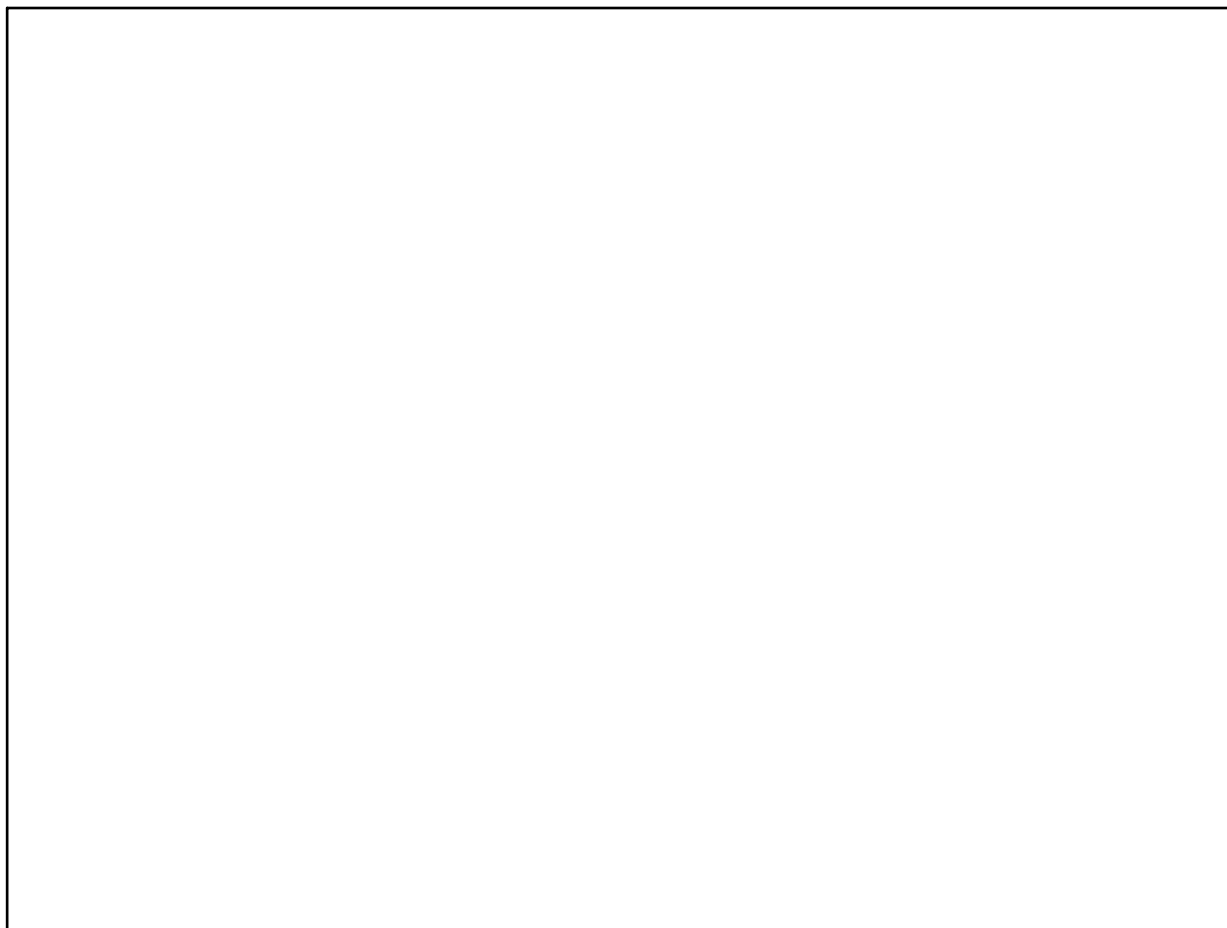
union

{ {?X a mon:City; mon:name ?CN; mon:locatedAt ?W . ?W a mon:River; mon:name ?RN}

filter not exists {?C2 a mon:Country ; mon:capital ?X} }}

order by ?CN

Dez 4-14:59



Dez 4-15:07