

Ex 1.5

$$\pi[\dots]$$

$$\sigma[\text{pop} < \text{sysquery}]$$

county city (on capital)

not allowed

$$\pi[\text{county.name}]$$

$$\sigma[\dots \text{city is capital} \dots \wedge \text{city.pop} > 0.25 \text{ county.pop}]$$

county city

DO NOT touch

Jun 4-10:08

county, NO city with > 0.25 pop.

$\exists X, XP, C, CapProv:$

$Q(XN) :- \text{county}(X, XN, XP, \dots, C, CapProv)$

$\wedge \neg \exists CP : \text{city}(C, CapProv, X, CP, \dots)$

$\wedge CP > 0.25 XP$

Query is safe ✓

Query is RANF? No

Push-into-not-exists

not self contained

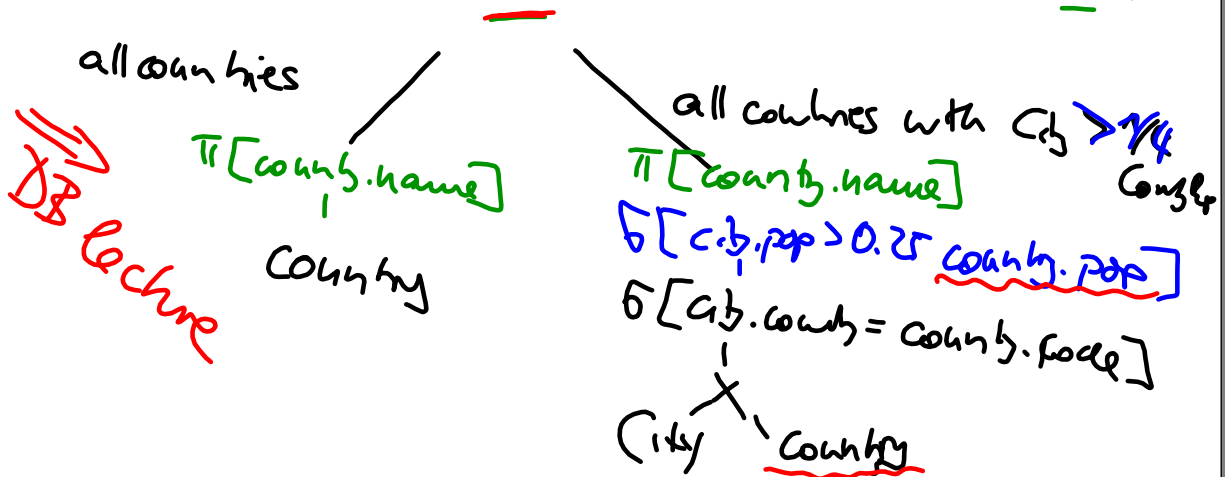
$\neg \exists (\text{county}(XP) \wedge \text{city}(\dots))$

Jun 4-10:37

Same in the algebra :

NOT EXISTS \rightsquigarrow MINUS

" \ "
" _ "



Jun 4-10:44

Ex 1.3 outer join

$$A \text{ OPT } B \approx A \bowtie B$$

? full outer join $A \bowtie B$ in SQL

$$\rightarrow (A \bowtie B) \cup (B \bowtie A)$$

\rightarrow Duplicates? = inner join \rightarrow set algebra \rightarrow removes Duplicates

SQL: eliminate Dups explicitly

Select DISTINCT ...
 where {
 $P_A(X)$ optional $P_B(Y)$
 UNION
 $P_B(X)$ optional $P_A(X)$
 }

Jun 4-10:50

$(A \bowtie B) \cup (B \bowtie A) \setminus (B \bowtie S)$
 removes Duplicates

Jun 4-10:58

Select ~~DISTINCT~~
 where { $P_A(X)$ OR $P_B(Y)$ } ✓
 UNION
 { $(P_B(X) \text{ OR } P_A(X))$ } " " remove them
 FILTER (!bound(X₀)) here
 some non-null X₀

Jun 4-11:01

b) pattern with union

```

select distinct ?XN ?CN ?RN
where {
  { ?X a:City; :name ?XN.
    ?C a:Country; :cap ?X; :name ?CN }
  OPTIONAL
  { ?X :locatedAt ?R.
    ?R a:River; :name ?RN } }
  UNION
  { ?X a:City; :name ?XN;
    :locatedAt ?R.
    ?R a:River; :name ?RN }
  OPTIONAL
  { ?C a:Country; :cap ?X;
    :name ?CN } }
  Filter ( !bound(?CN) )
  
```

Annotations:
 } Capitals
 } (at Rivers)
 at River

Jun 4-11:04

alternable:

```

select ?XN ?CN ?RN
where {
  { ?X a:City; :name ?XN }
  OPTIONAL
  { ?C a:Country; :cap ?X;
    :name → ?CN } }
  OPTIONAL
  { ?X :locatedAt ?R.
    ?R a:River; :name ?RN }
  Filter ( bound(?CN) || bound(?RN) )
  
```

Annotations:
 all other

Jun 4-11:12

Relational join

A		B	
a ₁	a ₂	b ₁	b ₂
1	null	null	2

Relational

$$A \bowtie B = \{ \}$$

SQL

$$A \text{ JOIN } B = \{ \{ x \rightarrow 1, z \rightarrow 2 \} \}$$

Jun 4-11:17

null-tolerant join in SQL

considers R(A,B,C)
S(A,B,D)

A is non-null,
B may contain nulls.

⇒ join cond:
select R.A, coalesce(R.b, S.b)

... where R.a = S.a and
(R.b = S.b or R.b is null or S.b is null)

→ Result Format
R.A, R.B, R.C, S.A, S.B, S.D

SQL does not have the coalesce-problem because it takes the UNION OF MAGNUS!

Jun 4-11:22

"\ " in the SQL-Algebra

NOT EXISTS

$R(A, B, C)$, $S(A, B, D)$ as before

$R \setminus S \rightarrow$

select A, B, C
 from R
 where NOT EXISTS (

" -- any D"

select *
 from S
 where (R.A = S.A and
 (R.B = S.B or R.B is null
 or S.B is null))

Jun 4-11:29

"\ " is there in the Algebra,
 but not in SQL !

\Rightarrow how to write it in SQL (1.0)

Consider SQL-subqueries

$R(?A, ?B, ?C)$
 $S(?A, ?B, ?D)$

select ?A ?B ?C
 where (?A, ?B, ?C is an answer to R)

OPTIONAL (?A ?B ?D is an answer to S)

Filled (!bound(D))

^ (?X ?Y ?Z ?Z)

add some "flag"

Consider $\{A \rightarrow 1\} \in [R]$ Dismiss!

$\{A \rightarrow 1, B \rightarrow 2\} \in [S]$

\Rightarrow result contains $\{A \rightarrow 1, B \rightarrow 2\} \notin [R]!$

!WRONG!

Jun 4-11:36

Example: all ^{parts of} countries that are
not neighbors

Select ?C1 ?C2

Where { ?C1 a:country.
?C2 a:country.

optional { ?C1 :neighbor ?C2 .

?X ?Y ?Z

Filter (!bound ?X ?Z)

Yes/No!