

Ex 1:
 null-tolerant join:

$R(A, B, C) \bowtie_{\text{nulls}} S(A, B, D)$
maybe null

- Cartesian product:
 $\Rightarrow (A_1, B_1, C_1, A_2, B_2, D)$

- with condition
 $R.A = S.A \wedge (R.B = S.B \vee R.B \text{ is null} \vee S.B \text{ is null})$
- drop column A2
- SQL: coalesce (X, Y):
 if X is not null: $\rightarrow X$
 if X is null: $\rightarrow Y$
 (i.e. if both are null \rightarrow null)
 $\sim B := \text{coalesce}(R.B, S.B)$

$\Rightarrow \pi [R.A, B \leftarrow \text{coalesce}(R.B, S.B), R.C, S.D]$
 $\sigma [R.A = S.A \wedge (R.B = S.B \vee R.B \text{ is null} \vee S.B \text{ is null})]$

$\begin{matrix} X \\ / \quad \backslash \\ R \quad S \end{matrix}$
 Hint: A, B, C, D columns

SQL: select R.A as A, coalesce(R.B, S.B) as B,
 R.C as C, S.D as D
 from R, S
 where R.A = S.A
 and (R.B = S.B or R.B is null or S.B is null)

Jun 14-10:17

b) $\begin{matrix} \text{''} \\ \text{S} \times \text{R} \times \text{Q} \end{matrix}$

$A \text{ } \swarrow_{\text{S} \times \text{R} \times \text{Q}} \text{ B} \hat{=} \text{Select } * \text{ from } A \text{ where } \boxed{\text{not exists}} \text{ (select } * \text{ from } B \text{ where } \dots \text{ "that matches A")}$

Rel Algebra: $A \text{ } \swarrow_{\text{rel}} (A \text{ } \bowtie \text{ B})$
 \uparrow left semi join

Jun 14-10:35

Λc) Consider SPARQL

$$\{ P_1 \} \text{ MINUS } \{ P_2 \}$$

does not exist in SPARQL (1.0)

expression in SPARQL 1.0:

P_1 OPTIONAL P_2
 Filter (!BOUND(V))

V must be a variable that occurs in P_2 but not in P_1

Problem: if such a variable does not exist :-)

⇒ extend P_2 - new variable

P_1 opt ($P_2 \wedge \text{Triple}$)
 Filter !BOUND(W)

↳ Example

Jun 14-10:39

- all county names x city names in this country, s.t. not capital

$$\{ ?X \text{ a :County ; :name ?XV ; :hasCity ?C . ?C :name ?CV } \}$$

(OPTIONAL)

$$\{ ?X \text{ a :County ; :capital ?C ; :name ?Z } \}$$

(Filter (!BOUND(?Z)))

SPARQL 1.1:

FILTER NOT EXISTS

P_1 :
 ?C, ?CV
 ?X, ?XV

P_2 :
 ?X, ?C

Jun 14-10:45

1d): \bowtie_{rel} : $R(A,B)$
 $S(B,C)$

$$R \bowtie S := R \bowtie S \cup R \setminus (R \bowtie S) \times \{c: null\}$$

A, B, C A, B, C ←

\bowtie_{SPARQL} in the same way: B in rel. Alg. ✓

\cup_{SPARQL} allows for different formats

$$\bowtie_{SPARQL} := R \bowtie S \cup_{SPARQL} R \setminus_{rel} (R \bowtie S)$$

Jun 14-10:52

Ex 2: Full outer join for SPARQL:

$$\begin{matrix} (R \bowtie S) \cup (R \bowtie S) \\ \cup \\ R \bowtie S \end{matrix} \cup \begin{matrix} R \cup \\ R \bowtie S \end{matrix}$$

Duplicates: Set semantics removes them :)

SPARQL: multiset semantics \Rightarrow DISTINCT

or

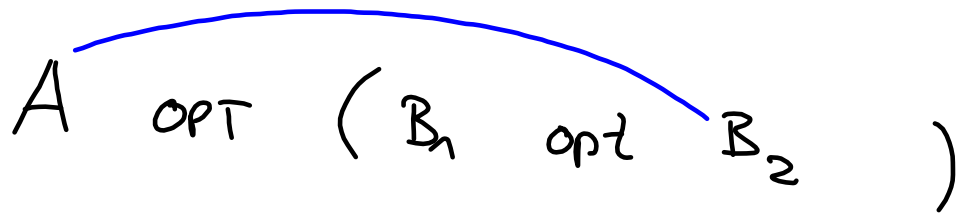
$$(R \bowtie S) \cup (R \bowtie S \setminus R \bowtie S)$$

Jun 14-10:56

Ex. 3

"guess"

A OPT (B₁ opt B₂)



Jun 14-11:24

Q. 223

"42" ^^xsd:int ≡ 42

Jun 14-11:27