SQL: The Bad Cont'd

Ordering Results:

- historically just an extra can be added to the top level query
- now: for some years, SQL had a LIMIT k clause, often used with ORDER BY (usually importance / relevance) "top k" (e.g. top10 salary) due to non-determinism

Null Values

value inapplicable: bad schema design (compromise)

- better design exists, but poor performance

value unknown: can be replaced by any domain value satisfying integrity constraints

→ many possibilities (possible worlds)

how to answer queries?

→ true in all worlds W of an incomplete D

Certain Answer $$Q(D) = \bigcap_{W \text{ world of } D} Q(W)$$

→ answer common to all possible worlds

computationally infeasible in general!

SQL gives a crude approximation.

expressions: NULL as param to op makes result NULL
pred/comp. : 3-valued logic (crude approx. of V-Unk.)
set ops: unique special value for duplicates
aggregate op: doesn't count (value inapplicable)

Comps w/ NULL return UNKNOWN

∧	T	U	F
T	T	U	F
U	U	U	F
F	F	F	F

∨	T	U	F
T	T	T	T
U	T	U	U
F	T	U	F

¬	
T	F
U	U
F	T

x null

$\underbrace{(x = 0)}_{\text{Unk}}$ OR $\underbrace{(x <> 0)}_{\text{Unk}}$ = UNKNOWN ≠ TRUE

IS TRUE / IS FALSE / IS UNKNOWN

WHERE <cond> is shorthand for WHERE <cond> IS TRUE

select aid, name from author where url is NULL

count(URL) counts only non-NULL URLs

count(*) counts rows

Outer Joins

Allows NULL-padded answers that fail to satisfy a conjunct in a conjunction

FULL/LEFT/RIGHT/INNER

FROM R <j-type> JOIN S on C

$$\{(x,y,z) \mid R(x,y) \wedge S(y,z)$$

$$\vee (z = \text{NULL} \wedge R(x,y) \wedge \neg(\exists z \cdot S(y,z))) \quad \text{(LEFT/FULL)}$$

$$\vee (x = \text{NULL} \wedge S(y,z) \wedge \neg(\exists x \cdot R(x,y))) \quad \text{(RIGHT/FULL)} \}$$

select aid, pub from author left join wrote on aid=author

1	ChTo98
5	—

select aid, count(publication) as pubs from author left join wrote on aid=author group by aid

#pub per author

NULLs are a necessary evil

Can always be avoided, but possibly at cost of efficiency can't escape NULLS in practice (schema evolution)

'connect to cs 348'