

Notes

Sunday, April 28, 2019

11:06 PM

- Three value logic: true, false, unknown
 - where, having requires true value
 - unknown or true \Rightarrow true (null=1 \Rightarrow unknown)
 - unknown or false \Rightarrow unknown
 - unknown and true \Rightarrow unknown
 - unknown and false \Rightarrow false
 - Don't allow null in not in, or it will cause unknowns
 - Try use not exist instead of not in, or use where to remove null.

Concepts

- Physical data independence: The logical definition of data remain unchanged.
 - Relational database systems support it better than NoSQL did.
- first normal form = not nested = flat
- Monotone when ever we add tuples to one or more input tables, the answer to the query will not lose any output table.
 - If Q is a select-from-where query that does not have subqueries or aggregates, then it's monotone.
 - If a query is monotone, then we can write it without nested subqueries.

- SQL & RA can express same queries
- facts : tuples
- rules : queries
- stratified A datalog program is stratified if it can be partitioned into strata.
 - automatically stratified if no aggregation or negation.
- { OLTP - NoSQL
 OLAP no updates }
 - partitioning
 - replication
- semi-structured data : good for data exchange.

• SQL Datatype

text, char(n), varchar(n), float, real, date, time ...

• Declare table

create table table-name (

name1 type (primary key),

name2 type (references) table2_name

);

▪ If name2 value inserted into table 2 not in table1, report "foreign key constraint failed"

o aggregation: count, max, min, sum, avg

▪ Count(*) will not ignore nulls

o no aggregates are allowed in where clause

▪ Everything in select must be in group by or be aggregates

o FWHOS

o with temp-table-name as (subquery)

• Relational Algebra Operators

union \cup difference $-$

selection σ projection π cartesian product \times join \bowtie left outer join \ltimes
 duplicate elimination δ grouping γ sorting τ

- Datalog aggregates: count, min, max, sum

$\alpha(c) :- c = \text{count} : \{ \dots \}.$

- $R(A, B) \bowtie S(B, C)$ \rightarrow size m \rightarrow size n
 \circ no keys: max length: $m \cdot n$

\circ B foreign key in R: max length: m .

\circ B foreign key in S: max length: n .

- Check: I. What semantic
 II. If there's NULL
 III. distinct
 IV. left outer join
 V. When having same schema
 consider union, intersection