By Marina Barsky

Structured Query Language SQL

Summary

3-valued logic

TRUE = 1, FALSE = 0, UNKNOWN (NULL) = ½

AND: min, OR: max, NOT: 1-x

Any comparison with NULL yields UNKNOWN (NULL)

Any arithmetic operation on NULL results in NULL

`(p or (NOT p))` might not be `TRUE`!

`(0*x)` might not be `0`!

NULL in aggregations

- Computing aggregates AGG in column A: null is not included
- If all values in A are NULL – AGG returns NULL (except for count which returns 0)
- Count (*) returns total number of rows

NULL in joins

- Two nulls are not equal and tuples are not joined on nulls

- If you want to include in the result the tuples that did not have a match in the joined table, use LEFT or FULL outer join: the missing attributes from the second table will be padded with NULLs

Subqueries can be used

- As a relation in a FROM clause.
- As a value in a WHERE clause.
- With ANY, ALL, IN or EXISTS in a WHERE clause.
- As operands to UNION, INTERSECT or EXCEPT.

Subqueries

- If subquery returns 1 value – can compare using regular operators: =, > , <, <>

- If subquery returns a list of values (unary relation), can use same operators applied to
  - ANY
  - ALL
  - IN

Subqueries

- If subquery returns a list of tuples (not unary relation), can only use
  - IN
- If we only need to know the count of tuples in the subquery – use
  - EXISTS
- The subquery can be executed for each value in the outer query – correlated subquery, expensive

Controlling duplicate elimination

- For projections use SELECT DISTINCT to get a set. Bag is default

- For set operators use UNION ALL (INTERSECT ALL, EXCEPT ALL) to get a bag. Set is default

Aggregation queries

- Include in SELECT only aggregate functions (AGG) and grouping attributes
- Grouping attributes have to appear in the GROUP BY clause

Difference between WHERE clause and HAVING clause

- HAVING applies to results of aggregations
- WHERE applies to a single tuple