CS 1555

Lecture 9

**Formal Query Languages: Relational Algebra (continued)**

Outer union

- defined on partially union compatible relations

- non union-compatible attributes are kept in r OUTER UNION s

- non-union compatible attributes without value are set to NULL

- Tuples are “matched” over common named attributes like in natural join

- Outer intersection/outer difference don’t make sense

Outer join

- Selects only tuples satisfying the join condition

- Left outer join keeps every tuple in left relation

- Right outer join keeps every tuple in right relation

- Full outer join keeps every tuple

- Attributes of tuples with no matching tuples are set to NULL

Aggregate functions

- Mathematical and statistical aggregate functions on collections of values

- SUM, MAXIMUM, MINIMU, AVERAGE

- COUNT number of tuples (cardinality)

- Aggregation functions always return a table, not a scalar value

Grouping

- Grouping the tuples in a relation

- Tuples are grouped based on the values of <grouping attributes>

Recursive closure

- Applied to a recursive relationship between tuples of same relation

- E.g. find all ancestors or descendants

- Need control statements… iteration

Write queries in relational algebra

- Deletion: r 🡨 r – *Relational\_expression*

- Student 🡨 Student – (SIGMADept=’CSD’ and QPA < 2.5 (STUDENT))

- Insertion: r 🡨 r UNION Relational\_expression

- Student 🡨 Student UNION {(365, ‘Smith’, ‘John’)}

- Updating: r 🡨 PROJECTattributes-to-be-updated(r)

- Student 🡨 PROJECTDept=’CSD’(SELECTDept=’CS’(STUDENT))

Discussion

- The relational algebra is procedural

- The queries in relational algebra specify how to produce a result, but the how should be the responsibility of the system

- User queries should be declarative specifying what is to be retrieved