The Relational Algebra
Part 2

Chapter 5b

Outline

- Additional Relational Operations
- Aggregate functions
- Recursive Closure
- Outer join
- Outer union
- Examples of Queries in Relational Algebra

Additional Relational Operations

Aggregate Functions and Grouping

- A type of request that cannot be expressed in the basic relational algebra is to specify mathematical **aggregate functions** on collections of values from the database.

- Examples of such functions include retrieving the average or total salary of all employees or the total number of employee tuples. These functions are used in simple statistical queries that summarize information from the database tuples.

Aggregate functions

- Common functions applied to collections of numeric values include AVERAGE, SUM, MAX, MIN, COUNT.
- Each of these function can be applied to a collection of tuples (or to a group of tuples that verify some properties):
- $_{\textit{<grouping attributes>}}\mathfrak{I}_{\textit{<function list>}}$ <RelationName>,
- where <grouping attributes> is a list of
- attributes specified in <relation name>,
- <function list is> a list of (<function>, <attribute>).

Example

$\mathcal{F}_{\text{MAX } Salary}$ (Employee)

retrieves the maximum salary value from the Employee relation

$\mathcal{F}_{\text{MIN } (Salary),\ AVERAGE(Salary)}$ (Employee)

retrieves the minimum Salary value and its average from the Employee relation

$\mathcal{F}_{\text{SUM } (Salary)\ ,COUNT\ (SSN)}$ (Employee)

retrieves the sum of the Salary and the number of employees from the Employee relation

Example

- for each D# we would like to know the number
- of employees and their average salary:

$$_{D\#}\,\Im\,_{count(SSN),\ Average(Salary)}\ (\text{EmPloyee})$$

Use of the Functional operator $\mathcal{F}$

- DNO $\mathcal{F}_{COUNT\ (SSN),\ AVERAGE\ (Salary)}$ **(Employee)**

  gives the number of employees and their average salary per department

- SuperSSN $\mathcal{F}_{COUNT\ (SSN),\ AVERAGE\ (Salary)}$ **(Employee)**

  gives the number of employees and their average salary per supervisor

Recursive Closure Operations

- Another type of operation that, in general, cannot be specified in the basic original relational algebra is **recursive closure.** This operation is applied to a **recursive relationship**.
- An example of a recursive operation is to retrieve all SUPERVISEES of an EMPLOYEE e at all levels—that is, all EMPLOYEE e' directly supervised by e; all employees e'' directly supervised by each employee e'; all employees e''' directly supervised by each employee e''; and so on

Recursive Closure Operations

- Although it is possible to retrieve employees at each level and then take their union, we cannot, in general, specify a query such as "retrieve the supervisees of '*Ahmad alShahrani*' at all levels without utilizing a looping mechanism.
- The SQL3 standard includes syntax for recursive closure.

The OUTER JOIN Operation

- In NATURAL JOIN tuples without a *matching* (or *related*) tuple are eliminated from the join result. Tuples with null in the join attributes are also eliminated. This amounts to loss of information.

- A set of operations, called outer joins, can be used when we want to keep all the tuples in R, or all those in S, or all those in both relations in the result of the join, regardless of whether or not they have matching tuples in the other relation.

OUTER JOIN Operation

- The left outer join operation keeps every tuple in the *first* or *left* relation R in R ⟕ S; if no matching tuple is found in S, then the attributes of S in the join result are filled or “padded” with null values.

- A similar operation, right outer join, keeps every tuple in the *second* or right relation S in the result of R ⟖ S.

- A third operation, full outer join, denoted by ⟗ keeps all tuples in both the left and the right relations when no matching tuples are found, padding them with null values as needed.

OUTER UNION Operations

- The outer union operation was developed to take the union of tuples from two relations if the relations are *not union compatible*.
- This operation will take the union of tuples in two relations R(X, Y) and S(X, Z) that are **partially compatible**, meaning that only some of their attributes, say X, are union compatible.
- The attributes that are union compatible are represented only once in the result, and those attributes that are not union compatible from either relation are also kept in the result relation T(X, Y, Z).