Relational Database: The Relational Data Model; Operations on Database Relations

Greg Plaxton
Theory in Programming Practice, Spring 2005
Department of Computer Science
University of Texas at Austin
Overview

• Review of relations in mathematics
• Database relations
• Operations on database relations
Review of Relations in Mathematics

- Cross product
- Binary relation over a single set
- Generalization to $n$-ary relations
Cross Product

- Let $A$ and $B$ be two sets

- Then $A \times B$ is the set consisting of all pairs $(a, b)$ such that $a \in A$ and $b \in B$

- For finite sets $A$ and $B$, $|A \times B| = |A| \cdot |B|$
Binary Relation over a Set $S$

- Simply a subset of $S \times S$

- So there are $2^9 = 512$ different relations that one can define over the set $\{1, 2, 3\}$
  - Example: $\{(1, 2), (1, 3), (2, 3)\}$ is a relation over $\{1, 2, 3\}$
  - The preceding relation, call it $R$, happens to correspond to the operator $<$ in the sense that $(x, y)$ belongs to $R$ if and only if $x < y$

- Various properties of such relations are commonly defined, such as reflexivity, symmetry, transitivity
  - These concepts are not important in the context of database relations
Binary Relation

• A binary relation over a (ordered) pair of sets $A$ and $B$ is a subset of $A \times B$

• Example: If $A = \{1, 2, 3\}$ and $B = \{2, 5\}$ then any subset of $
\{(1, 2), (1, 5), (2, 2), (2, 5), (3, 2), (3, 5)\}$ is a relation over $A$ and $B$

Theory in Programming Practice, Plaxton, Spring 2005
Generalization to $k$-ary Relations

- Suppose we are given a sequence of sets $A_1, \ldots, A_k$

- A $k$-ary relation with respect to this sequence is any subset of $A_1 \times \cdots \times A_k$
Database Relations

● Consists of two parts:
  – A relational schema
  – A set of tuples
Relational Schema

- A set of attributes, each of which has an associated set called the \textit{domain} of the attribute
- Example: One attribute of a relation that contains information about students might be “birthdate”
  - The domain of this attribute is the set of all valid dates
The Set of Tuples of a Database Relation

• Each tuple specifies a value for each attribute of the relation
• A database relation is often represented as a table

• There is one column for each attribute
  – The order of the columns is unimportant, i.e., reordering the columns does not yield a different relation

• There is one row for each tuple
  – The order of the rows is also unimportant

• The entry in row $i$ and column $j$ is the value assigned by the $i$th tuple to the $j$th attribute
Relational Database

- A relational database is a set of database relations with distinct names.
- Typically, every database relation in a relational database $D$ has a common attribute with some other database relation in $D$. 
Relational Algebra

- An algebra consists of elements, operations, and identities
- Example: Algebra of basic arithmetic over the integers
  - Elements are the integers
  - Operations are $+, -, \times, \div$
  - Identities are equations such as $x + y = y + x$, $x \times (y + z) = x \times y + x \times z$, where $x$, $y$, and $z$ range over the elements (i.e., integers)
- In relational algebra, the elements are database relations
- We will now introduce a number of basic operations on database relations
Operations on Database Relations

- Union
- Intersection
- Difference
- Cross product (also called cartesian product)
- Projection
- Selection
- (Natural) Join
Operations on Compatible Database Relations

- Two database relations are said to be *union-compatible*, or simply *compatible*, if they have the same relational schema, i.e., the same set of attributes.

- The union, intersection, and difference operations are only defined over compatible database relations $R$ and $S$:
  - In each case, the resulting database relation is compatible with $R$ and $S$.
  - The set of tuples of $R \cup S$ consists of those tuples in either $R$ or $S$.
  - The set of tuples of $R \cap S$ consists of those tuples in both $R$ and $S$.
  - The set of tuples of $R - S$ consists of those tuples in $R$ but not $S$. 
Cross Product $R \times S$ of Database Relations $R$ and $S$

- First, assume that $R$ and $S$ have no common attributes
  - In this case, the set of attributes of $R \times S$ is the union of those of $R$ and $S$
  - The tuples of $R \times S$ are all tuples that can be formed by concatenating a tuple in $R$ with a tuple in $S$
  - So the number of tuples in $R \times S$ is equal to the number in $R$ times the number in $S$

- What if $R$ and $S$ have common attributes?
  - Rename the attributes to ensure that they are all distinct, and then proceed as above
Projection of a Database Relation $R$

- Specifies the subset of the attributes of $R$ to be retained
- When we drop the other attributes, we may get duplicates
- Such duplicates are removed
- Thus the number of tuples in any projection of $R$ is at most the number of tuples in $R$
- We write $\pi_{u_1, \ldots, u_k}(R)$ to denote the projection of database relation $R$ that retains attributes $u_1, \ldots, u_k$
Selection

• Selection from a database relation $R$ involves specifying a predicate $p$ defined over the tuples of $R$, i.e., $p$ maps each tuple of $R$ to a boolean value

• We write $\sigma_p(R)$ to denote the relation consisting of the subset of tuples of $R$ that satisfy predicate $p$

• The relational schema of $\sigma_p(R)$ is the same as that of $R$
(Natural) Join

- The join of database relations $R$ and $S$ is denoted $R \bowtie S$
- The join may be viewed as a more refined way of taking cross product
- As in the cross product, we consider each tuple $r$ in $R$ and $s$ in $S$
  - If $r$ and $s$ match in their common attributes, concatenate them keeping only one set of columns for the common attributes
- However, in the special case where $R$ and $S$ have no common attributes, we do not consider $R \bowtie S$ to be the same as $R \times S$
  - Instead, $R \bowtie S$ is defined to be the empty relation with the same relational schema as $R \times S$