Relations and Relational Algebra

Database Systems

Michael Pound

www.cs.nott.ac.uk/~mpp/G51DBS
mpp@cs.nott.ac.uk

This Lecture

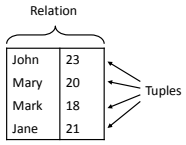
- The Relational Model
  - Relational data structures
- Relational algebra
  - Union, Intersection and Difference
  - Product of Relations
  - Projection, Selection
- Further reading
  - Database Systems, Connolly & Begg, 4.2 and 5.1
  - The Manga Guide to Databases, Chapter 2

The Relational Model

- Introduced by E.F. Codd in his paper "A Relational Model of Data for Large Shared Databanks", 1970
- The foundation for most (but not all) modern database systems

Relational Data Structure

- Data is stored in *relations* (tables)
- Data takes the form of *tuples* (rows)
  - The order of tuples is not important
  - There must not be duplicate tuples



Relations

- We will use tables to represent relations
- This is an example relation between people and email addresses:

Anne	aaa@cs.nott.ac.uk
Bob	bbb@cs.nott.ac.uk
Chris	ccc@cs.nott.ac.uk

Relations

- In general, each column has a *domain*, a set from which all possible values for that column can come
- For example, each value in the first column below comes from the set of first names

Anne	aaa@cs.nott.ac.uk
Bob	bbb@cs.nott.ac.uk
Chris	ccc@cs.nott.ac.uk

Relations

- A mathematical relation is a set of tuples: sequences of values. Each tuple represents a row in the table:

Anne	aaa@cs.nott.ac.uk	0115 911 1111
Bob	bbb@cs.nott.ac.uk	0115 922 2222
Chris	ccc@cs.nott.ac.uk	0115 933 3333

- {<Anne, aaa@cs.nott.ac.uk, 01159111111>,
  <Bob, bbb@cs.nott.ac.uk, 01159222222>,
  <Chris, ccc@cs.nott.ac.uk, 01159333333>}

Terminology

- **Degree of a relation**: how long each tuple is, or how many columns the table has
  - In the first example (name, email), the degree of the relation is 2
  - In the second example (name, email, phone) the degree of the relation is 3
  - Degrees of 2, 3, … are often called Binary, Ternary, etc.
- **Cardinality of a relation**: how many different tuples there are, or how many rows a table has

Mathematical Definition

- The mathematical definition of a relation R of degree n, where values come from domains A_1, …, A_n:

  $R \subseteq A_1 \times A_2 \times \ldots \times A_n$

  (a relation is a subset of the Cartesian product of domains)

  Cartesian product:

  $A_1 \times A_2 \times \ldots \times A_n =$

  $\{<a_1, a_2, \ldots, a_n>: a_1 \in A_1, a_2 \in A_2, \ldots, a_n \in A_n\}$

Data Manipulation

- Data is represented as relations
- Manipulation of this data (through updates and queries) corresponds to operations on relations
- Relational algebra describes those operations. These take relations as arguments, and produce new relations
- Relational algebra contains two types of operators. Common, set-theoretic operators and those specific to relations

Union

- Standard set-theoretic definition of union:

  $A \cup B = \{x: x \in A \text{ or } x \in B\}$
- For example, $\{a,b,c\} \cup \{a,d,e\} = \{a,b,c,d,e\}$
- For relations, we require the results to be in the form of another relation.
- In order to take a union of relations R and S, R and S must have the same number of columns and corresponding columns must have the same domains

Union-compatible Relations

- Two relations R and S are **union-compatible** if:
  - They have the same number of columns
  - Corresponding columns have the same domains

Union-compatible Example

Same number of columns and matching domains

Anne	1970
Bob	1971
Chris	1972

Tom	1980
Sam	1985
Steve	1986

Not union-compatible Example

Different numbers of columns

Anne	1970	NG7
Bob	1971	NG16
Chris	1972	NG21

Tom	1980
Sam	1985
Steve	1986

Not union-compatible Example

Corresponding columns have different domains

Anne	NG7
Bob	NG16
Chris	NG21

Tom	1980
Sam	1985
Steve	1986

Unions of Relations

- Let R and S be two union-compatible relations. The Union $R \cup S$ is a relation containing all tuples from both relations:
  $R \cup S = \{x: x \in R \text{ or } x \in S\}$
- Note that union is a partial operation on relations. That is, it is only defined for some (compatible) relations
- This is similar in principle to division of numbers. Division by zero is undefined

Union Example

R

Cheese	1.34
Milk	0.80
Bread	0.60
Eggs	1.20
Soap	1.00

S

Cream	2.00
Soap	1.00

$R \cup S$

Cheese	1.34
Milk	0.80
Bread	0.60
Eggs	1.20
Soap	1.00
Cream	2.00

Difference of Relations

- Let R and S be two union-compatible relations. The **difference** R - S is a relation containing all tuples from R that are not in S:
  $R - S = \{x: x \in R \text{ and } x \notin S\}$
- This is also a partial operation on relations

Difference Example

R

Cheese	1.34
Milk	0.80
Bread	0.60
Eggs	1.20
Soap	1.00

S

Cream	2.00
Soap	1.00

R - S

Cheese	1.34
Milk	0.80
Bread	0.60
Eggs	1.20

Intersection of Relations

- Let R and S be two union-compatible relations. The **intersection** $R \cap S$ is a relation containing all tuples that are in both R and S:

  $R \cap S = \{x: x \in R \text{ and } x \in S\}$
- This is also a partial operation on relations

Intersection Example

R

Cheese	1.34
Milk	0.80
Bread	0.60
Eggs	1.20
Soap	1.00

S

Cream	2.00
Soap	1.00

$R \cap S$

Soap	1.00

Cartesian Product

- Cartesian product is a total operation on relations.
  - Can be applied to relations of any relative size
- Set-theoretic definition of product:

  $R \times S = \{\langle x, y\rangle: x \in R, y \in S\}$
- For example, if <Cheese, 1.34> $\in$ R and <Soap, 1.00> $\in$ S then

  <<Cheese,1.34>,<Soap,1.00>> $\in R \times S$

Extended Cartesian Product

- **Extended** Cartesian product flattens the result into a single tuple. For example:

  <Cheese, 1.34, Soap, 1.00>
- This is more useful for relational databases
- For the rest of this course, "product" will mean extended Cartesian product

Extended Cartesian Product of Relations

- Let R be a relation with column domains $\{A_1,...,A_n\}$ and S a relation with column domains $\{B_1,...,B_m\}$. Their extended Cartesian product R x S is a relation:

  $R \times S = \{\langle c_1, ..., c_n, c_{n+1}, ..., c_{n+m}\rangle:$

  $\langle c_1, ..., c_n\rangle \in R, \langle c_{n+1}, ..., c_{n+m}\rangle \in S\}$

Product Example

R

Cheese	1.34
Milk	0.80
Bread	0.60
Eggs	1.20
Soap	1.00

S

Cream	2.00
Soap	1.00

R x S

Cheese	1.34	Cream	2.00
Milk	0.80	Cream	2.00
Bread	0.60	Cream	2.00
Eggs	1.20	Cream	2.00
Soap	1.00	Cream	2.00
Cheese	1.34	Soap	1.00
Milk	0.80	Soap	1.00
Bread	0.60	Soap	1.00
Eggs	1.20	Soap	1.00
Soap	1.00	Soap	1.00

Projection

- Sometimes using all columns in a relation is unnecessary
- Let R be a relation with n columns, and X be a set of column identifiers. The projection of R on X is a new relation $\pi_X(R)$ that only has columns in X
- For example, $\pi_{1,2}(R)$ is a table that contains only the 1st and 2nd columns of R
- We can use numbers or names to index columns (naming columns will be discussed in the next lecture)

Projection Example

R

1	2	3
Anne	aaa@cs.nott.ac.uk	0115 911 1111
Bob	bbb@cs.nott.ac.uk	0115 922 2222
Chris	ccc@cs.nott.ac.uk	0115 933 3333

$\pi_{1,3}(R)$

Anne	0115 911 1111
Bob	0115 922 2222
Chris	0115 933 3333

Selection

- Sometimes we want to select tuples based on one or more criteria
- Let R be a relation with n columns, and α is a property of tuples
- **Selection from R subject to condition α** is defined as:

$\sigma_\alpha(R) = \{<a_1,\ldots,a_n> \in R: \alpha(a_1,\ldots,a_n)\}$

Comparison Properties

- We assume that properties are written using {and, or, not} and expressions of the form col(i) Θ col(j), where i, j are column numbers, or col(i) Θ v, where v is a value from domain A_i
- Θ is a comparator which makes sense when applied to values from columns i and j. Often these will be = , ≠, ≤, ≥, <, >

Meaningful Comparisons

- Comparisons between values can only take place where it makes sense to compare them
  - We can always perform an equivalence test between two values in the same domain
  - In some cases you can compare values from different domains, e.g. if both are strings
- For example, 1975 < 1987 is a meaningful comparison, "Anne" = 1981 is not
- We can only use a comparison in a selection if its result is true or false, never undefined

Selection Example

- $\sigma_{col(3) < 2002 \text{ and } col(2) = Nolan}(R)$

R

Insomnia	Nolan	2002
Magnolia	Anderson	1999
Insomnia	Skjoldbjaerg	1997
Memento	Nolan	2000
Gattaca	Niccol	1997

Selection Example

- $\sigma_{col(3) < 2002 \text{ and } col(2) = Nolan}(R)$

R

Insomnia	Nolan	2002
Magnolia	Anderson	1999
Insomnia	Skjoldbjaerg	1997
Memento	Nolan	2000
Gattaca	Niccol	1997

Selection Example

- $\sigma_{col(3) < 2002 \text{ and } col(2) = Nolan}(R)$

R

Memento	Nolan	2000

Other Operations

- Not all SQL queries can be translated into relational algebra operations defined in this lecture
- Extended relational algebra includes counting, joins and other additional operations

This Lecture in Exams

What is the result of the following operation $\pi_{1,3}(\sigma_{col(2) = col(4)}(R \times S))$, where R and S are:

R

Anne	111111
Bob	222222

S

Chris	111111
Dan	222222

Next Lecture

- The Relational Model
  - Relational data structures
  - Relational data integrity
- Further reading
  - Database Systems, Connolly & Begg, Chapter 4.2
  - The Manga Guide to Databases, Chapter 2