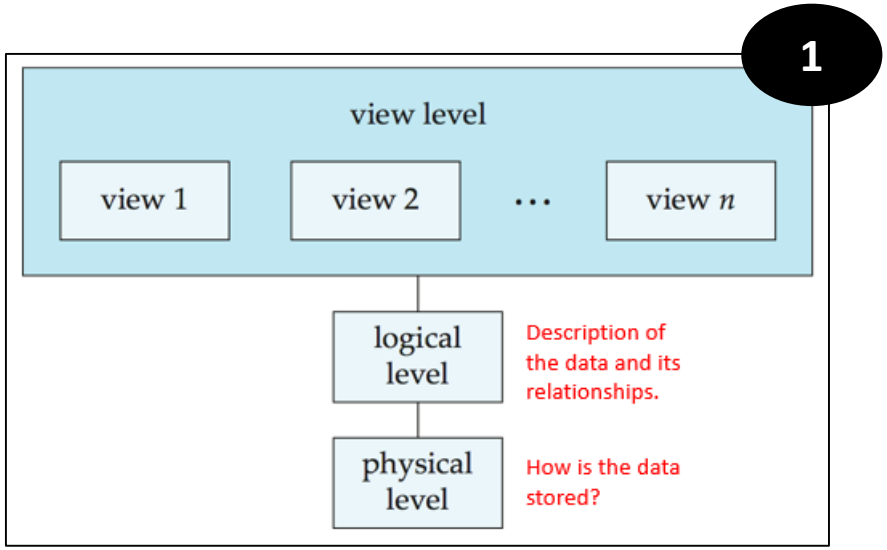
RDBMS and SQL
Relational Calculus

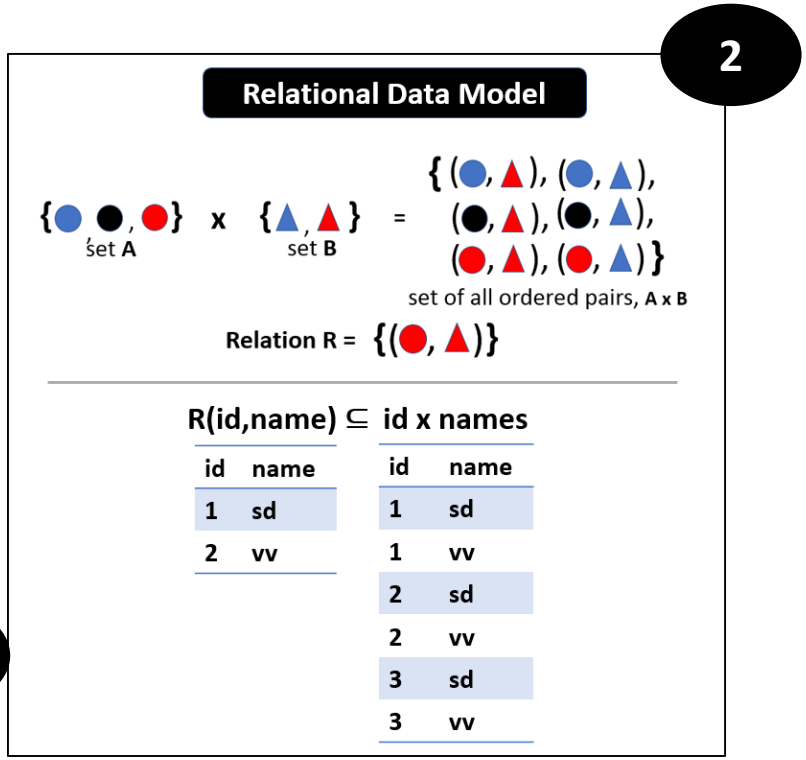
Venkatesh Vinayakarao
venkateshv@cmi.ac.in
http://vvtesh.co.in

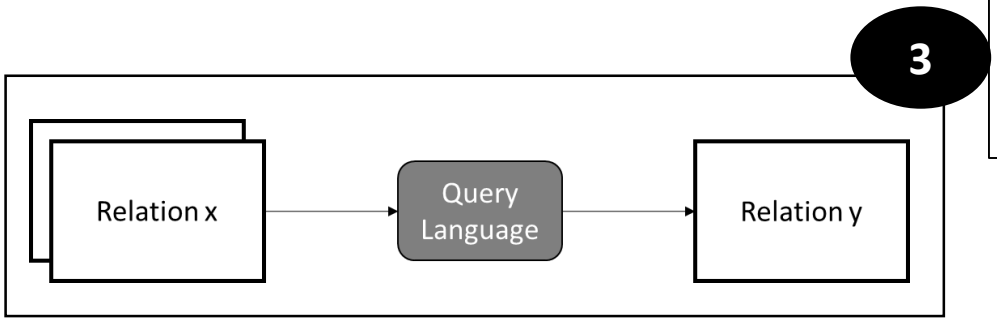
Chennai Mathematical Institute

Slide contents are borrowed from the course text. For the authors' original version of slides, visit: https://www.db-book.com/db6/slide-dir/index.html.

Story So Far…

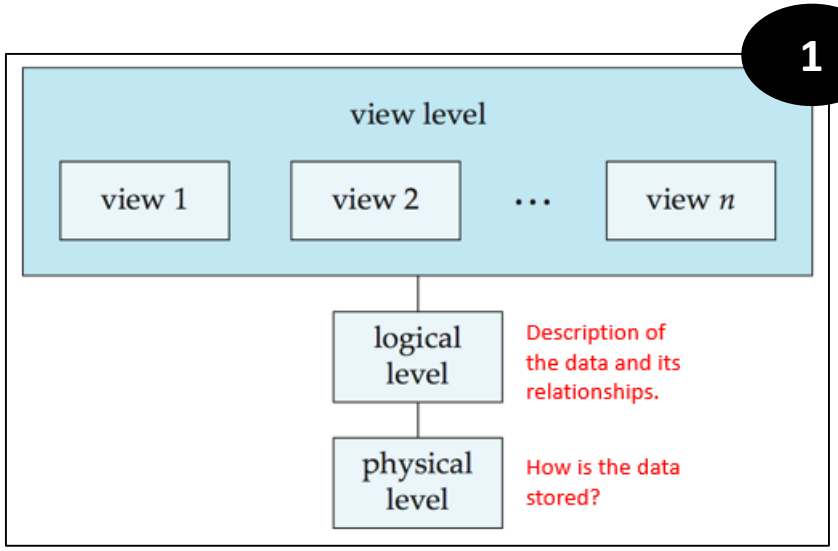


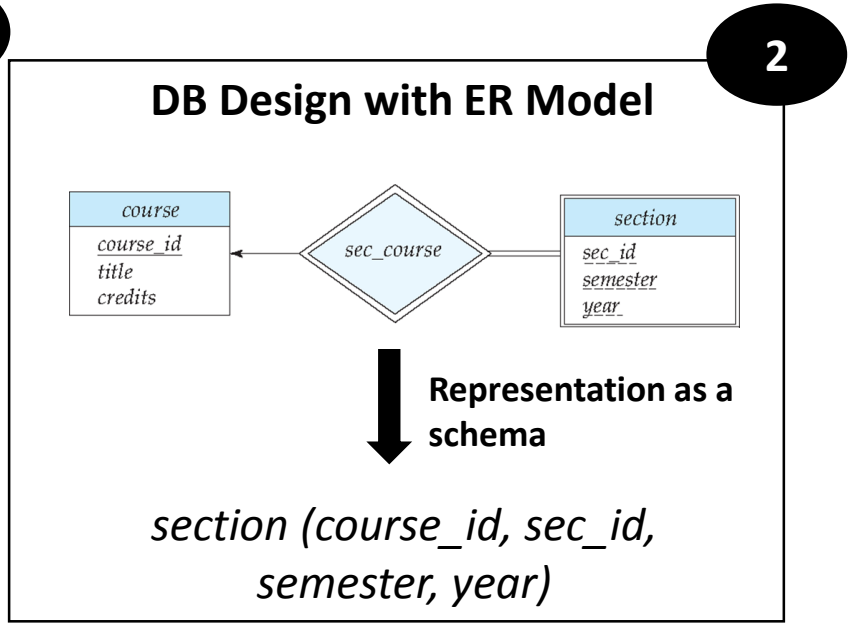




Relational Algebra
SQL

Story So Far…





Functional Dependency and Normal Forms

Review

Customer ID	First Name	Surname	Telephone Number
123	Pooja	Singh	555-861-2025, 192-122-1111
456	San	Zhang	(555) 403-1659 Ext. 53; 182-929-2929
789	John	Doe	555-808-9633

Not in 1NF

Manufacturer	Model	Model full name	Manufacturer country
Forte	X-Prime	Forte X-Prime	Italy
Forte	Ultraclean	Forte Ultraclean	Italy
Dent-o-Fresh	EZbrush	Dent-o-Fresh EZbrush	USA

Not in 2NF

Tournament	Year	Winner	Winner's date of birth
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 September 1968
Des Moines Masters	1999	Al Fredrickson	21 July 1975

Not in 3NF

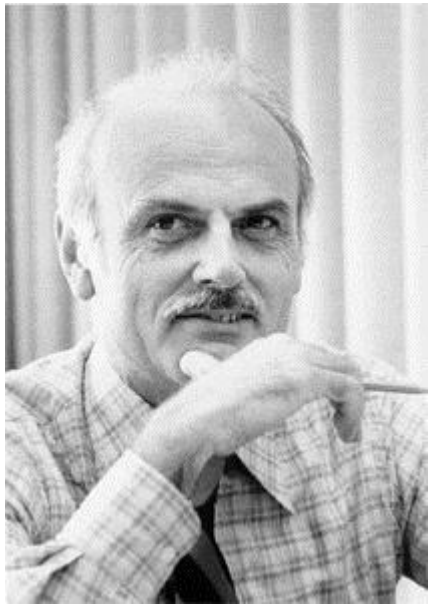
Court	Start time	End time	Rate type
1	09:30	10:30	SAVER
1	11:00	12:00	SAVER
1	14:00	15:30	STANDARD

Not in BCNF

Restaurant	Pizza Variety	Delivery Area
A1 Pizza	Thick Crust	Springfield
A1 Pizza	Thick Crust	Shelbyville
A1 Pizza	Thick Crust	Capital City

Not in 4NF

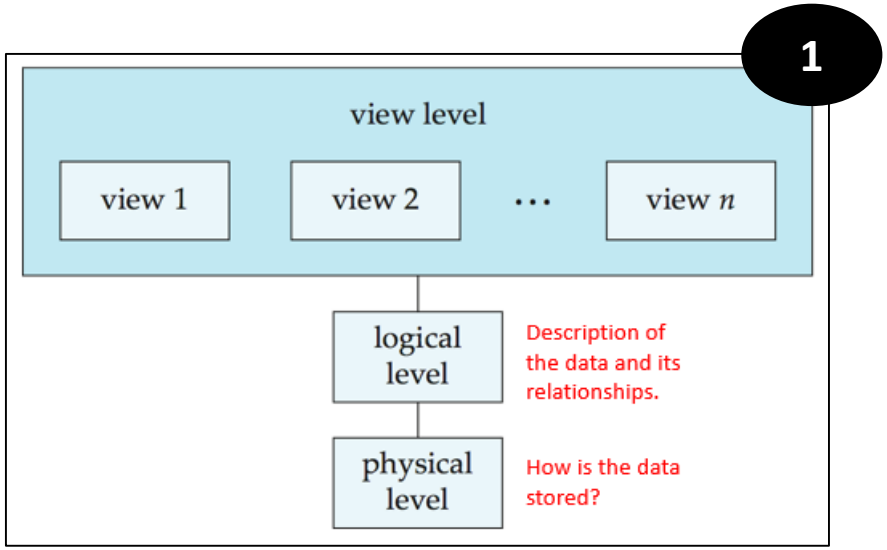
History

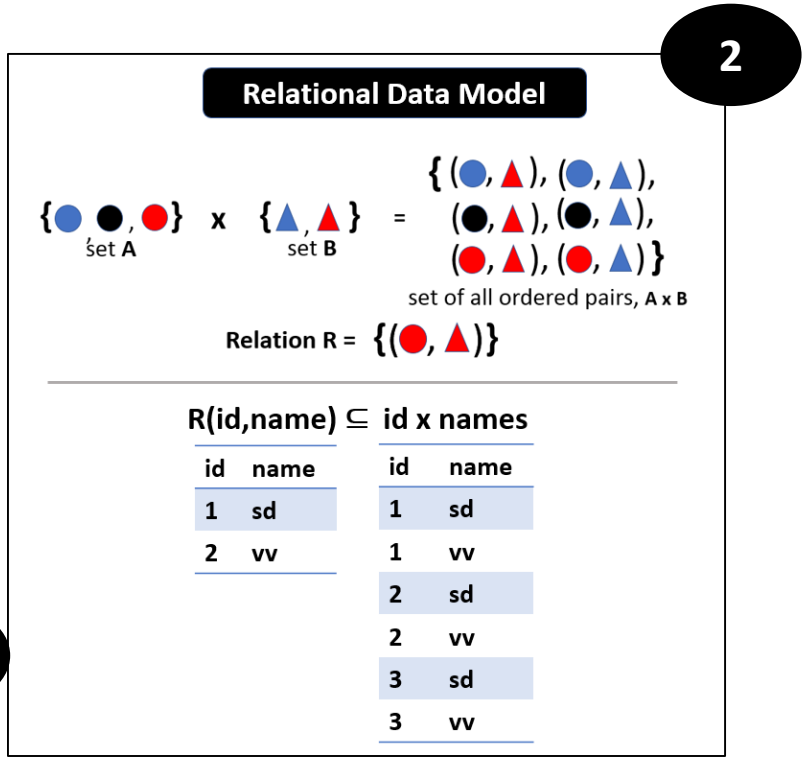


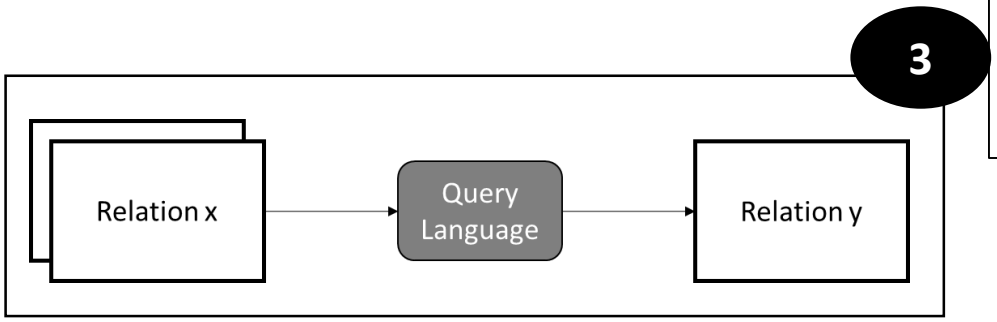
Edgar Frank "Ted" Codd
(19 August 1923 – 18 April 2003)
IBM's San Jose Research Lab
Turing Award, 1981.

Major Contributions
Boyce-Codd Normal Form
Online analytical processing (OLAP)
Tuple Relational Calculus
Codd's Theorem

Story So Far…

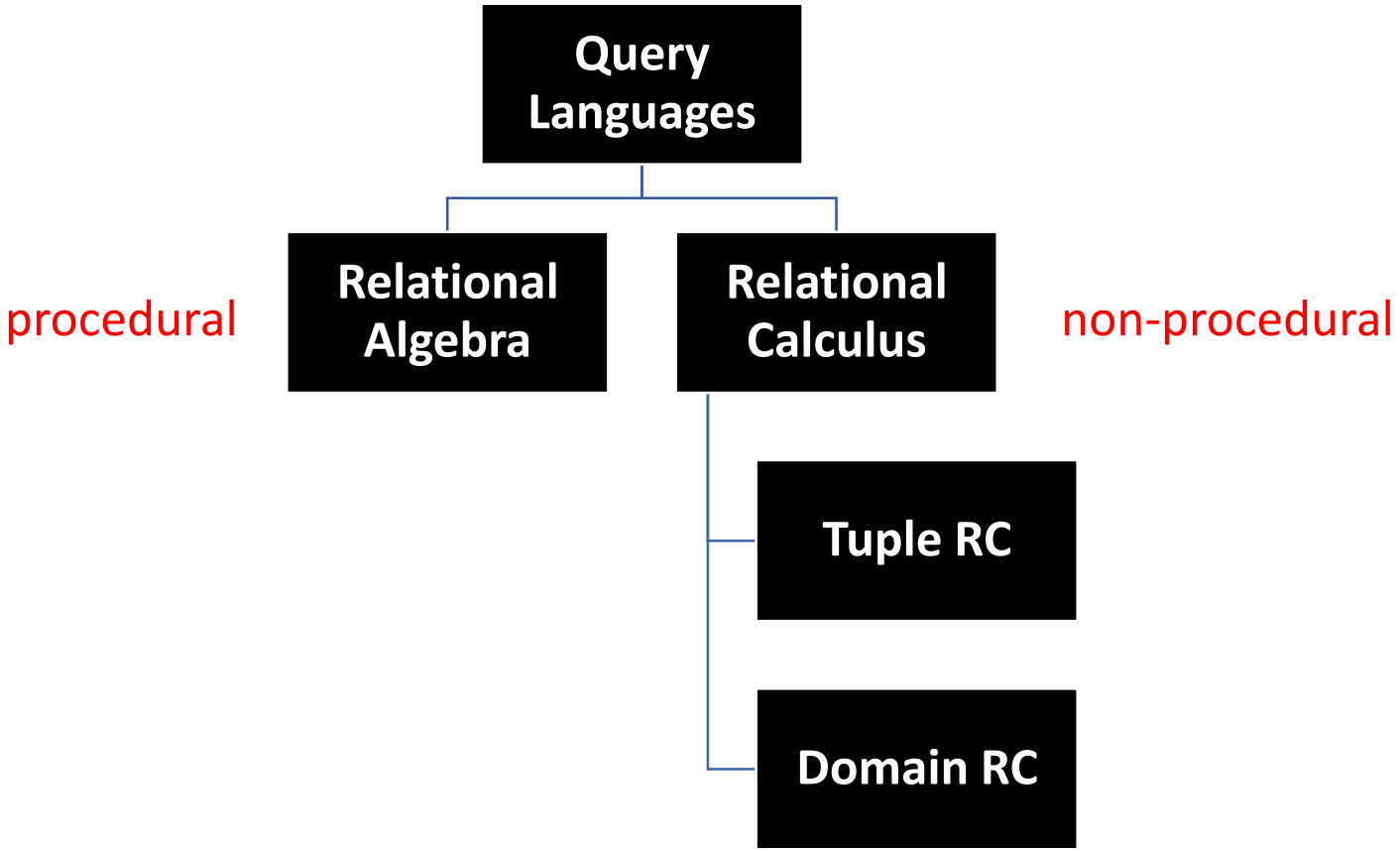




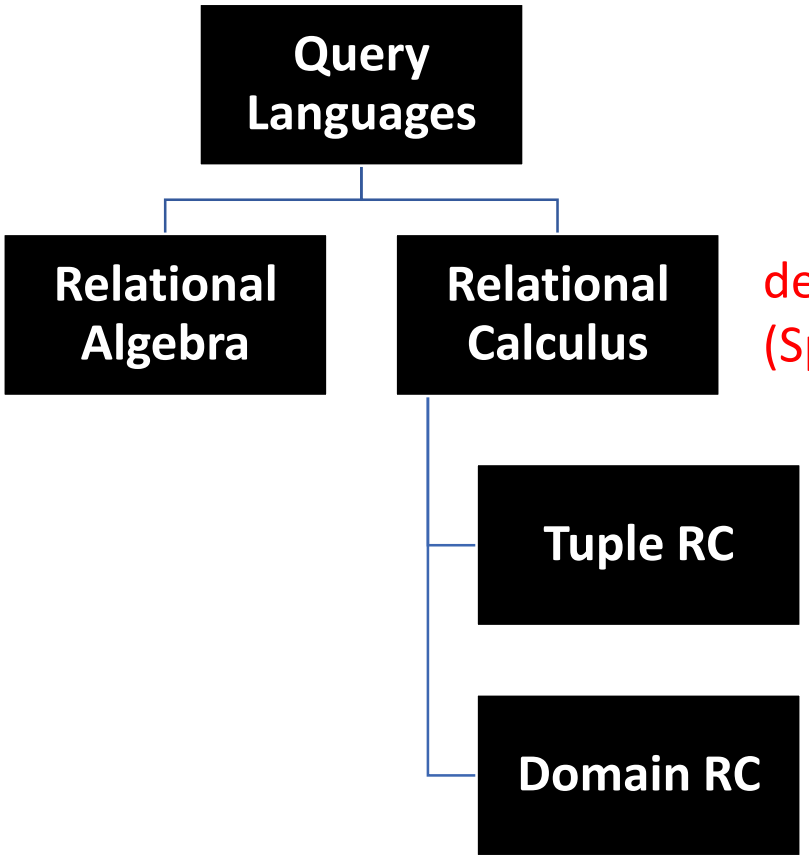


Relational Algebra
SQL

Query Languages



Query Languages



operational (real mechanics of how to get what we want)

descriptive (Specify what you want)

same expressive power

Calculus

- From Calculi, a Latin word for “stone”.
- History: We used stones for calculation.
- Today, it refers to methods of computation.

Source: Calculus Volume I: The Mathematics of Change and Variation.

Tuple Relational Calculus

- A nonprocedural query language, where each query is of the form

$$\{t \mid P(t)\}$$

- It is the set of all tuples *t* such that predicate *P* is true for *t*
- *P* is a *formula* similar to that of the predicate calculus

Example Query

- Find *all tuples from* instructor relation whose salary is greater than $80,000

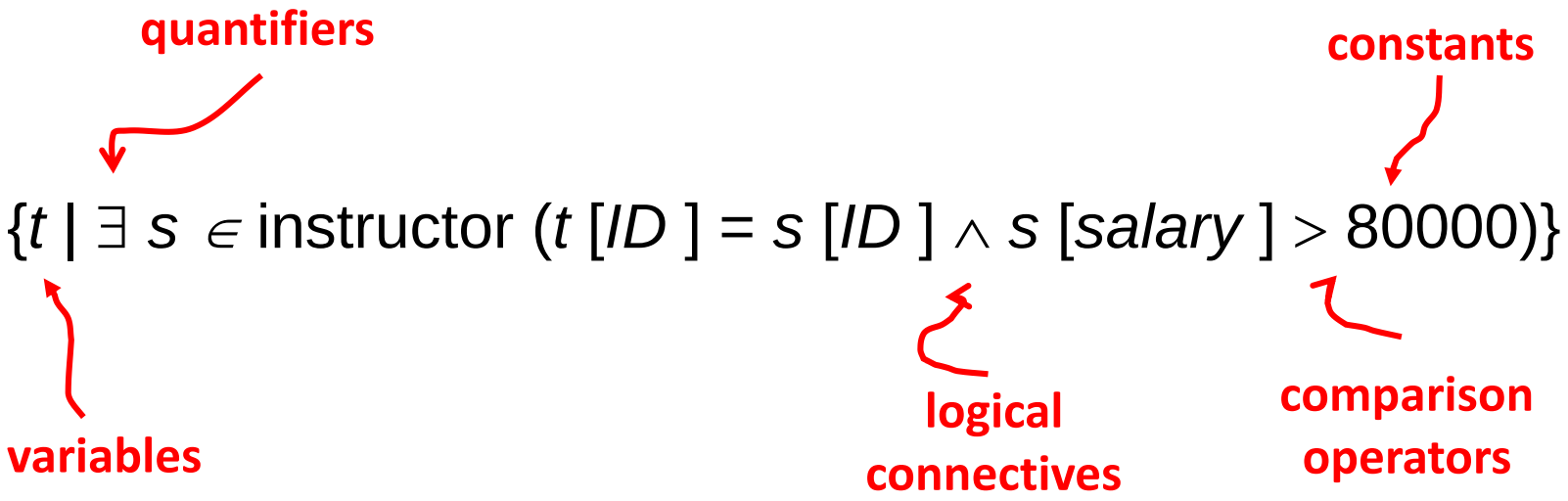
$$\{t \mid t \in instructor \wedge t\,[salary\,] > 80000\}$$

- As in the previous query, but output only the *ID* attribute value

$$\{t \mid \exists\, s \in \text{instructor}\ (t\,[ID\,] = s\,[ID\,] \wedge s\,[salary\,] > 80000)\}$$

Returns only ID attribute

A Tuple Relational Calculus Formula



TRC is Built with Atoms

- *Atom has one of the following forms:*
  - *s* $\in$ *r*
  - *s[x] op u[y] where op could be <,>,=,<=, >=, <>*
  - *s[x] op c, where c is a constant*
- *Formulae are built using the following rules:*
  - *Atom is a formula*
  - *If P is a formula, then so are* ¬P and (P)
  - If P(s) is a formula, $\exists$ *s* $\in$ *r(P(s)) and* $\forall$ s $\in$ *r(P(s)) are also formulae.*

Safety of Expressions

- Not all tuple-relational-calculus expressions are safe.
  - $\{t \mid \neg(t \in instructor)\}$ has infinite results!
- We introduce domain operator, dom() to handle this situation.
  - dom($\neg(t \in instructor)$) is set of all values appearing in instructor. Hence, safe!

Domain Relational Calculus

Domain Relational Calculus

- A nonprocedural query language equivalent in power to the tuple relational calculus
- Each query is an expression of the form:

$$\{ < x_1, x_2, \ldots, x_n > \mid P(x_1, x_2, \ldots, x_n) \}$$

  - $x_1, x_2, \ldots, x_n$ represent domain variables
  - P represents a formula similar to that of the predicate calculus

Example Queries

- Find the *ID, name, dept_name, salary* for instructors whose salary is greater than $80,000
  - $\{< i, n, d, s > \mid < i, n, d, s > \in instructor \wedge s > 80000\}$
- As in the previous query, but output only the *ID* attribute value
  - $\{< i > \mid < i, n, d, s > \in instructor \wedge s > 80000\}$
- Note that $\{<i> \mid \neg(<i> \in instructor)\}$ is unsafe

Relationally Complete

- Any query language as powerful as relational algebra is called relationally complete.
- Relational calculus (restricted to safe expressions) is relationally complete.

SQL is not relationally complete

SQL allows duplicate rows… so, not relational at all!
No zero-attribute relations possible.
Complex queries need loops/iterators.