

# Similarity-Based Databases: An Introduction

Vilem Vychodil (Palacky University, Olomouc)

# DAMOL

DATA ANALYSIS AND MODELING LAB

Palacky University, Olomouc, Czech Republic



europa  
european  
social fund in the  
czech republic



EUROPEAN UNION



MINISTRY OF EDUCATION,  
YOUTH AND SPORTS



OP Education  
for Competitiveness

INVESTMENTS IN EDUCATION DEVELOPMENT

# Problem setting

## Extension of Codd's relational model:

- extension of data tables and related agenda
- **similarity** relations on domains
- **ranks** assigned to tuples

## Reason:

- to introduce formal (logical) model for
  - 1 *similarity-based queries*  
"Show all houses that are sold for \$400,000."
  - 2 *approximate dependencies* in data  
"Do houses in similar locations have similar prices?"

## Not Discussed:

- physical model (logical model is independent)
- algorithms (future research), . . .

## Our extension of Codd's model

**(ranked) data tables over domains with similarities**

<u><i>name</i></u>	<u><i>age</i></u>	<u><i>education</i></u>
Adams	30	Comput. Sci.
Black	30	Comput. Eng.
Chang	28	Accounting
Davis	27	Comput. Eng.
Enke	36	Electric. Eng.
Francis	39	Business

# Our extension of Codd's model

(ranked) data tables over domains with similarities

<u><i>name</i></u>	<u><i>age</i></u>	<u><i>education</i></u>
Adams	30	Comput. Sci.
Black	30	Comput. Eng.
Chang	28	Accounting
Davis	27	Comput. Eng.
Enke	36	Electric. Eng.
Francis	39	Business

$$n_1 \approx_n n_2 = \begin{cases} 1 & \text{if } n_1 = n_2 \\ 0 & \text{if } n_1 \neq n_2 \end{cases}$$

$$a_1 \approx_a a_2 = s_a(|a_1 - a_2|)$$

with scaling  $s_a : \mathbb{Z}^+ \rightarrow [0, 1]$

$\approx_e$	A	B	CE	CS	EE
A	1	.7			
B	.7	1			
CE			1	.9	.7
CS			.9	1	.6
EE			.7	.6	1

# Our extension of Codd's model

(ranked) data tables over domains with similarities

$\mathcal{D}(t)$	<u><i>name</i></u>	<u><i>age</i></u>	<u><i>education</i></u>
1.0	Adams	30	Comput. Sci.
1.0	Black	30	Comput. Eng.
0.9	Chang	28	Accounting
0.8	Davis	27	Comput. Eng.
0.4	Enke	36	Electric. Eng.
0.3	Francis	39	Business

$$n_1 \approx_n n_2 = \begin{cases} 1 & \text{if } n_1 = n_2 \\ 0 & \text{if } n_1 \neq n_2 \end{cases}$$

$$a_1 \approx_a a_2 = s_a(|a_1 - a_2|)$$

with scaling  $s_a : \mathbb{Z}^+ \rightarrow [0, 1]$

$\approx_e$	A	B	CE	CS	EE
A	1	.7			
B	.7	1			
CE			1	.9	.7
CS			.9	1	.6
EE			.7	.6	1

ranked table  $\Rightarrow$  **answer to similarity-based query**

## Related work (1 of 2)

### Extensions of Codd's model employing fuzzy logic

- several approaches, many papers
- Raju, Majumdar, Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems.  
*ACM Trans. Database Systems* Vol. 13, No. 2, 1988, pp. 129–166.

### Extensions of Codd's model employing probability

- **different** both semantically and technically (probability  $\neq$  fuzzy logic)
- Fuhr, Rölleke, A probabilistic relational algebra for the integration of information retrieval and database systems.  
*ACM Trans. Information Systems* 15:32–66, 1997.
- D. Dey and S. Sarkar S. A probabilistic relational model and algebra.  
*ACM Trans. Dat. Syst.* 21:339–369, 1996.

## Related work (2 of 2)







### Fagin et al.

- R. Fagin. Combining fuzzy information: an overview. *ACM SIGMOD Record* 31(2):109–118, 2002.
- Natsev, Chang, Smith, Li, Vitter: Supporting incremental join queries on ranked inputs. *VLDB 2001*, pp. 281–290.
- Cohen, Sagiv: An incremental algorithm for computing ranked full disjunctions. *PODS 2005*, pp. 98–107.

### RankSQL + related research

- Li, Chang, Ilyas, Song: RanSQL: Query Algebra and Optimization for Relational top- $k$  queries. *ACM SIGMOD* 2005, pages 131–142, 2005.
- Ilyas, Aref, Elmagarmid: Supporting top- $k$  join queries in relational databases. *The VLDB Journal* 13:207–221, 2004.

## Previous Results

-  R. B., V. V.: Data tables with similarity relations: functional dependencies, complete rules, and non-redundant bases. In: *Proc. DASFAA 2006, LNCS 3882*, 2006, pp. 644–658.
-  R. B., S. O., V. V.: Relational algebra for ranked tables with similarities: properties and implementation. In: *Proc. IDA VII, LNCS 4723*, 2007, pp. 140–151.
-  R. B., V. V.: Data Dependencies in Codd's Relational Model with Similarities. In: *Handbook of Research on Fuzzy Information Processing in Databases*, 2008, pp. 634–657.
-  R. B., V. V.: Logical foundations for similarity-based databases. In: *Proc. DASFAA 2009, Workshops, LNCS 5667*, 2009, pp. 137–151.
-  R. B., V. V.: Query systems in similarity-based databases: logical foundations, expressive power, and completeness. In: *Proc. ACM SAC 2010*, pp. 1648–1655.
-  R. B., V. V.: Codd's relational model from the point of view of fuzzy logic. *Journal of Logic and Computation*, **21**(5)2011, 851–862.

# What is Fuzzy Logic?

Term **fuzzy logic** is used in

- **broad sense**: any application of fuzzy approach in modeling
  - Zadeh L. A.: Fuzzy sets. *Inf. Control* (1965)
  - simple observations on handling of vagueness
- **narrow sense**: mathematical fuzzy logic
  - Hájek P.: *Metamathematics of Fuzzy Logic*. (1998)
  - Basic Logic (BL-logic), propositional/predicate; *logic of continuous t-norms*
  - Höhle, Esteva, Godo, Gottwald, Montagna, ...
  - various logical calculi (MTL-logic)

Basic **principle**: we allow for propositions to have **intermediate truth degrees** instead of just 0 (false) and 1 (true), e.g.

$$||\text{value } x \text{ is similar to value } y.||_{\mathbf{M}} = 0.9$$

intuitive meaning “values  $x$  and  $y$  are very similar/almost equal, ...”.

## Our Approach:

- our (extended) model  $\sim$  predicate fuzzy logic(s)

## Preliminaries: structures of truth degrees

**(Complete) residuated lattice** – basic structure of truth degrees

$\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1 \rangle$ , where

$\langle L, \wedge, \vee, 0, 1 \rangle \dots$  (complete) lattice,

$\langle L, \otimes, 1 \rangle \dots$  commutative monoid,

$\langle \otimes, \rightarrow \rangle \dots$  adjoint pair ( $a \otimes b \leq c$  iff  $a \leq b \rightarrow c$ ).

**Structures on  $[0, 1]$**  (based on t-norms)

$\mathbf{L} = \langle [0, 1], \min, \max, \otimes, \rightarrow, 0, 1 \rangle$  given by left-continuous (continuous)  $\otimes$ .

Łukasiewicz:

$$a \otimes b = \max(a + b - 1, 0),$$

$$a \rightarrow b = \min(1 - a + b, 1),$$

Gödel (minimum):

$$a \otimes b = \min(a, b),$$

$$a \rightarrow b = \begin{cases} 1 & \text{if } a \leq b, \\ b & \text{otherwise,} \end{cases}$$

**Finite structures of truth degrees**

$L = \{a_0 = 0, a_1, \dots, a_n = 1\} \subseteq [0, 1] \dots$  finite subset of  $[0, 1]$

finite Łukasiewicz chain / Gödel chain  $\dots$  restrictions of  $\otimes, \rightarrow$  on finite  $L$

## Preliminaries: truth-stressing hedges

**Truth-stressing hedges** (Takeuti+Titani, Baaz, Hájek, ...)

(idempotent) truth-stressing **hedge** ... mapping  $*$ :  $L \rightarrow L$  satisfying

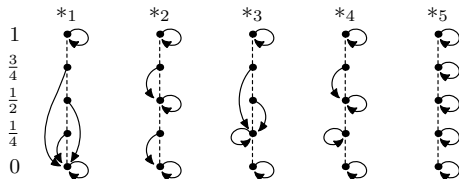
$$1^* = 1, \quad a^* \leq a, \quad (a \rightarrow b)^* \leq a^* \rightarrow b^*, \quad a^{**} = a^*,$$

meaning of  $*$ : truth function of logical connective “very true”

### Two boundary hedges

① **identity**, i.e.  $a^* = a$  ( $a \in L$ );

② **globalization**:  $a^* = \begin{cases} 1 & \text{if } a = 1, \\ 0 & \text{otherwise.} \end{cases}$  ... interp. “fully true”



# Preliminaries: fuzzy sets and fuzzy relations

## Residuated structure of truth degrees

$$\mathbf{L} = \langle L, \vee, \wedge, \rightarrow, \otimes, *, 0, 1 \rangle$$

## Fuzzy sets (**L**-sets)

**L-set**  $A$  in universe  $U$  ... mapping  $A: U \rightarrow L$ ,  $A = \{\dots, A(u)/u, \dots\}$

$A(u)$ : “degree to which  $u$  belongs to  $A$ ”

## Fuzzy relations (**L**-relations)

**binary L-relation**  $R$  between  $U$  and  $V$  ... mapping  $R: U \times V \rightarrow L$ ,

$R(u, v)$ : “degree to which  $u \in U$  and  $v \in V$  are  $R$ -related”

## Operations with **L**-sets

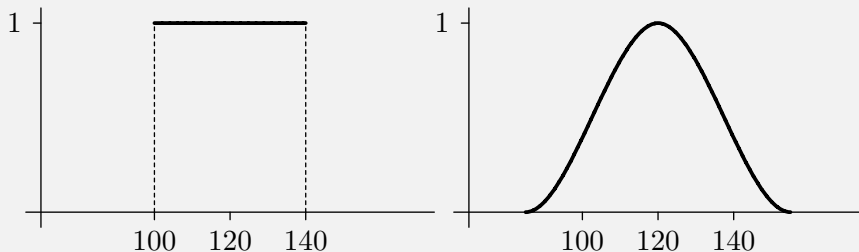
$\mathbf{L}^U$  ... collection of all **L**-sets in universe  $U$ ,

operations defined componentwise, e.g.  $(A \cap B)(u) = A(u) \wedge B(u)$ , ...

## Application

Example (Formalization of “Normal Blood Pressure”)

What is *normal systolic blood pressure*?



Fuzzy set (unary fuzzy relation) in universe  $[0, \infty)$ .

- warning: degrees of truth  $\neq$  degrees of belief
- compare: “Next patient will have systolic blood pressure  $120 \pm 10$ ”.

## More on Mathematical Fuzzy Logic (1 of 2)

### Syntax of Fuzzy Logics (predicate calculi)

- object variables, function and relation symbols (with arities)
- logical connectives:  $\otimes$  (strong conjunction),  $\Rightarrow$  (implication)  $\wedge$  (conjunction)  $\vee$  (disjunction).
- constants for truth degrees:  $\bar{0}$ ,  $\bar{1}$ , in general:  $\bar{a}$
- **formulas:** defined inductively as in classical logic
  - $r(t_1, \dots, t_n)$  – atomic formula,
  - if  $\varphi$  and  $\psi$  are formulas, then  $(\varphi \otimes \psi)$ ,  $(\varphi \Rightarrow \psi)$ , ... are formulas
  - if  $\varphi$  is formula and  $x$  is object variable then  $(\forall x)\varphi$  and  $(\exists x)\psi$  are formulas

### Example (Example of Formulas)

- $(\forall x)(r(x) \Rightarrow s(x)), r(f(x)) \Rightarrow \bar{0}$ ,
- additional connectives:  $\wedge, \vee, \Leftrightarrow, \dots$
- new nontrivial connectives: e.g.,  $\Delta\varphi$  reads “ $\varphi$  is very true”

## More on Mathematical Fuzzy Logic (2 of 2)

### Semantics of Formulas

- semantic components: fuzzy structures + evaluations of (free) variables

### Fuzzy Structure (analogy of first-order structure)

- $M =$  universe (nonempty)
- functions  $f^{\mathbf{M}}: M^n \rightarrow M$  interpreting function symbols
- fuzzy relations  $r^{\mathbf{M}}: M^n \rightarrow L$  interpreting relation symbols  
 $L \dots$  support of a (complete) residuated lattice

**Degrees of Truth:**  $\|r(t_1, \dots, t_n)\|_{\mathbf{M},v} = r^{\mathbf{M}}(\|t_1\|_{\mathbf{M},v}, \dots, \|t_n\|_{\mathbf{M},v})$

$$\|\varphi \otimes \psi\|_{\mathbf{M},v} = \|\varphi\|_{\mathbf{M},v} \otimes \|\psi\|_{\mathbf{M},v}, \quad \|\varphi \Rightarrow \psi\|_{\mathbf{M},v} = \|\varphi\|_{\mathbf{M},v} \rightarrow \|\psi\|_{\mathbf{M},v},$$

$$\|(\forall x)\varphi\|_{\mathbf{M},v} = \bigwedge_{v' \equiv_x v} \|\varphi\|_{\mathbf{M},v'}, \quad \|(\exists x)\varphi\|_{\mathbf{M},v} = \bigvee_{v' \equiv_x v} \|\varphi\|_{\mathbf{M},v'},$$

$$\|\Delta\varphi\|_{\mathbf{M},v} = (\|\varphi\|_{\mathbf{M},v})^*, \quad \|\bar{a}\|_{\mathbf{M},v} = a.$$

## Application

### Example (Obtaining “Graded Subsethood”)

Subsethood of sets  $A$  and  $B$ :

$$(\forall x)(x \in A \Rightarrow x \in B),$$

$$(\forall x)(r_A(x) \Rightarrow r_B(x)).$$

which is true in a structure  $\mathbf{M}$  with  $r_A^{\mathbf{M}} = A$  and  $r_B^{\mathbf{M}} = B$  iff, for each  $x$ , if  $x$  is in  $A$  then  $x$  is in  $B$ .

In graded setting, the same formula:

$$(\forall x)(r_A(x) \Rightarrow r_B(x))$$

evaluated in a **fuzzy structure**  $\mathbf{M}$  with  $r_A^{\mathbf{M}}: X \rightarrow L$  and  $r_B^{\mathbf{M}}: X \rightarrow L$ :

$$\|(\forall x)(r_A(x) \Rightarrow r_B(x))\|_{\mathbf{M}} = \bigwedge_{x \in X} (r_A^{\mathbf{M}}(x) \rightarrow r_B^{\mathbf{M}}(x)) =$$

**degree to which  $A: X \rightarrow L$  is a subset of  $B: X \rightarrow L$ .**

## Recall: Our Extension

$\mathcal{D}(t)$	<u><i>name</i></u>	<u><i>age</i></u>	<u><i>education</i></u>
1.0	Adams	30	Comput. Sci.
1.0	Black	30	Comput. Eng.
0.9	Chang	28	Accounting
0.8	Davis	27	Comput. Eng.
0.4	Enke	36	Electric. Eng.
0.3	Francis	39	Business

$$n_1 \approx_n n_2 = \begin{cases} 1 & \text{if } n_1 = n_2 \\ 0 & \text{if } n_1 \neq n_2 \end{cases}$$

$$a_1 \approx_a a_2 = s_a(|a_1 - a_2|)$$

with scaling  $s_a : \mathbb{Z}^+ \rightarrow [0, 1]$

$\approx_e$	A	B	CE	CS	EE
A	1	.7			
B	.7	1			
CE			1	.9	.7
CS			.9	1	.6
EE			.7	.6	1

**Note:** “Job Applicants” (similarity of names is trivial)

# Our Extension Formalized: Basic Notions

**Attributes** = names for columns of ranked data tables

- $Y$ : set of all attributes
- attributes denoted  $y, y', y_1, y_2, \dots$

**Relation Schemes** = subsets  $R \subseteq Y$

**Domains** = sets of all possible values of attributes

- $D_y$  is domain of attribute  $y \in Y$ ;
- nonempty (at most) enumerable set of all possible values  $y$

**Tuples:** for  $R \subseteq Y$  and domains  $D_y$ :

$$\text{Tupl}(R) = \prod_{y \in R} D_y \quad (\text{Cartesian product of domains})$$

- each  $r \in \text{Tupl}(R)$  is tuple over  $R$ ;
- $r(y) \in D_y$  is  $y$ -value of  $r$ .

# Domains with Similarities

**Similarity relations on domains** (needed for approximate matches)

each domain  $D_y$  equipped with binary  $\mathbf{L}$ -relation  $\approx_y$  satisfying:

(Ref) for each  $u \in D_y$ :  $u \approx_y u = 1$ ,

(Sym) for each  $u, v \in D_y$ :  $u \approx_y v = v \approx_y u$ , and (optionally):

(Sep) for each  $u, v \in D_y$ :  $u \approx_y v = 1$  iff  $u$  equals  $v$ , and

(Tra) for each  $u, v, w \in D_y$ :  $u \approx_y v \otimes v \approx_y w \leq u \approx_y w$ .

so-called **similarity relation**

**Domain with similarity** =  $\langle D_y, \approx_y \rangle$ , where

- $D_y$  is domain of attribute  $y \in Y$ ;
- $\approx_y$  is similarity on  $D_y$ .

Note:

- interpretation:  $u \approx_y v =$  degree to which  $u$  and  $v$  are similar
- boundary case: strict identity

# Ranked Data Tables over Domains with Similarities

central notion to our model:

- formal counterpart to *relations* from Codd's model
- in mathematical fuzzy logic: first-order fuzzy structure

## Definition

Let  $R \subseteq Y$  be a relation scheme and  $\langle D_y, \approx_y \rangle$  be a domain with similarity (for each  $y \in R$ ). A **ranked data table on  $R$  over**  $\{\langle D_y, \approx_y \rangle \mid y \in R\}$  is any (finite)  $L$ -set in  $\text{Tupl}(R)$ .

Notes:

- RDTs are denoted  $\mathcal{D}, \mathcal{D}', \mathcal{D}_1, \dots$
- RDT on  $R$  over  $\{\langle D_y, \approx_y \rangle \mid y \in R\} =$  fuzzy relation between  $D_y$
- map  $\mathcal{D}: \text{Tupl}(R) \rightarrow L$
- degree  $\mathcal{D}(r)$  is called a **rank of  $r$  in  $\mathcal{D}$**

# Notes on Generalization of Codd's Model of Data

**Classic (Codd's) Model** results by:

- taking two-valued Boolean algebra for  $\mathbf{L}$  (complete residuated lattice);
- considering each  $\approx_y$  to be identity relation on  $D_y$ .

**Consequence:** all ranks become 1 (match) and 0 (no match)

## Nonranked RDT

- all ranks are from  $\{0, 1\} \subseteq L$ ,  $\mathbf{L}$  is arbitrary;
- stored data prior to querying;

Important feature of our model: stored data = results of queries

RDTs represent both

- **stored data**, and
- **results of queries**.

# Notes on Domain Similarities and Ranks

## Where do similarities come from?

- can be assigned by an expert:
  - finite  $\mathbf{L}$  or a finite subset of infinite linear  $\mathbf{L}$ ;
  - Likert scale  $L = \{1, \dots, 5\}$  of degrees of satisfaction (Miller's  $7 \pm 2$  phenomenon);
- can be determined based on “distance”:
  - $\mathbf{L}$  on  $[0, 1]$  with  $\otimes$  being continuous Archimedean t-norm;
  - (pseudo)metric  $\iff \otimes$ -transitive similarity  $\mathbf{L}$ -relation relation;
- similarities are *purpose dependent*;
- implementation remark: can be stored (as data) / computed on demand.

## Where do ranks come from?

- appear from nonranked data after performing similarity-based queries,
- can be assigned by experts,
- important aspect: *comparative meaning of truth degrees*.

# Operations with RDTs

## Relational Operations

- ① counterparts of Boolean operations (basic / derived operations)
- ② new operations that arose in fuzzy logic
- ③ similarity-based selection and join (and projection)
- ④ further operations:  $top_k$ , extensional hull

## Notation:

- $\mathbf{L}$  ... complete residuated lattice
- $\mathcal{D}, \mathcal{D}', \mathcal{D}_1, \mathcal{D}_2, \dots$  ... ranked data tables over domain with similarities
- $y, y', y_1, y_1$  ... attributes
- $t, t', t_1, t_2, \dots, s, s', s_1, s_2, \dots$  ... tuples
- $t(y)$  ... denotes a value from  $D_y$  of tuple  $t$  on attribute  $y$
- $name, type$  ... (concrete) attribute names

## Counterparts to Boolean Operations

**Union**  $\mathcal{D}_1 \cup \mathcal{D}_2$ :

$$(\mathcal{D}_1 \cup \mathcal{D}_2)(t) = \mathcal{D}_1(t) \vee \mathcal{D}_2(t), \quad \text{for each tuple } t.$$

Interpretation: “a degree to which  $t$  matches  $Q_1$  or  $t$  matches  $Q_2$ ”.

In general:

$$(\mathcal{D}_1 \text{ symbol } \mathcal{D}_2)(t) = \mathcal{D}_1(t) \text{ operation } \mathcal{D}_2(t) .$$

Note: for  $\cup$ ,  $\cap$ , and  $\otimes$ : result is a data table (finite).

**Further Operations** (e.g., residuation and negation):

$$(\mathcal{D}_1 \rightarrow \mathcal{D}_2)(t) = \mathcal{D}_1(t) \rightarrow \mathcal{D}_2(t),$$

$$(\neg \mathcal{D})(t) = \mathcal{D}(t) \rightarrow 0.$$

Restricted to *active domains* (as in Codd's model).

## Operations Specific to Our Extension

- nontrivial *unary operations*
- interesting meaning in natural language

**$a$ -shifts:**  $(a \rightarrow \mathcal{D})(t) = a \rightarrow \mathcal{D}(t)$

**truth-stressing hedges:**  $(\mathcal{D}^*)(t) = \mathcal{D}(t)^*$

Example (0.6-shift: tuples that match  $Q$  at least to degree 0.6)

	$a$	$b$	$c$
0.9	769	325000	3
0.6	635	240000	3
0.6	659	200000	2
0.5	628	250000	3
0.3	567	200000	1
0.2	535	300000	4

$\Rightarrow$

	$a$	$b$	$c$
1.0	635	240000	3
1.0	659	200000	2
1.0	769	325000	3
0.9	628	250000	3
0.7	567	200000	1
0.6	535	300000	4

## Derived Relational Operations

Example (Derived Operation “ $a$ -cut”):

$$({}^a\mathcal{D})(t) = \begin{cases} 1, & \text{if } \mathcal{D}(t) \geq a, \\ 0, & \text{otherwise.} \end{cases}$$

can be defined:  ${}^a\mathcal{D} = (a \rightarrow \mathcal{D})^*$ , where  $*$  is *globalization*.

Example (Derived Operation “Above  $a$ ”):

$$(\text{above}(\mathcal{D}, a))(t) = \begin{cases} \mathcal{D}(t), & \text{if } \mathcal{D}(t) \geq a, \\ 0, & \text{otherwise.} \end{cases}$$

can be defined:  $\text{above}(\mathcal{D}, a) = \mathcal{D} \cap {}^a\mathcal{D} = \mathcal{D} \cap (a \rightarrow \mathcal{D})^*$ .

# Projection

**Projection**  $\pi_A(\mathcal{D})$  of  $\mathcal{D}$  onto  $R \subseteq T$ :

$$(\pi_R(\mathcal{D}))(r) = \bigvee_{s \in \text{Tupl}(T \setminus R)} \mathcal{D}(rs),$$

**Meaning:**

- $rs$  is (usual) concatenation of tuples over disjoint relation schemes;
- $(\pi_R(\mathcal{D}))(r) =$  “degree to which there is  $s \in \text{Tupl}(T \setminus R)$  such that  $rs$  is in  $\mathcal{D}$ .”

Example (Projection  $\pi_{\{b\}}(\mathcal{D})$ )

	<i>a</i>	<i>b</i>
0.9	567	1
0.9	628	3
0.8	659	2
0.8	535	4
0.4	769	3



	<i>b</i>
0.9	1
0.9	3
0.8	2
0.8	4

# Similarity-Based Selection

- makes use of similarities on domains
- “ $y = c$ ” means (the value of)  $y$  is equal to (constant) “ $c$ ”

**Selection Formulas** (for our purpose):

- ① “ $p = q$ ” where  $p, q$  are variables or (constants for) values
- ② each (constant for) a truth degree  $a \in L$  is a selection formula
- ③ if  $\varphi$  and  $\psi$  are selection formulas then  $(\varphi \otimes \psi)$ ,  $(\varphi \vee \psi)$ ,  $(\varphi \wedge \psi)$ ,  $(\varphi \Rightarrow \psi)$ , and  $\Delta\varphi$  are selection formulas.

**Selection**  $\sigma_\varphi(\mathcal{D})$  of tuples in  $\mathcal{D}$  matching  $\varphi$ :

$$\sigma_\varphi(\mathcal{D})(t) = \mathcal{D}(t) \otimes \|\varphi\|_t .$$

Meaning: *If  $\mathcal{D}$  is a result of query  $Q$ , the rank of  $t$  in  $\sigma_\varphi(\mathcal{D})$  is a “degree to which  $t$  matches  $Q$  and in addition it matches the selection formula  $\varphi$ ”.*

## Example

### Example (Similarity-Based Selection)

**Input Data  $\mathcal{D}$**  (all ranks = 1):

<i>name</i>	<i>type</i>	<i>bedrooms</i>	<i>price</i>
Kelly	Log Cabin	1	85000
Lee	Single Family	4	370000
Miller	Penthouse	2	290000
Nelson	Ranch	4	340000
Ortiz	Single Family	3	335000

**Result** of  $\sigma_{bedrooms=4 \otimes price=360000}(\mathcal{D})$ :

	<i>name</i>	<i>type</i>	<i>bedrooms</i>	<i>price</i>
0.9	Lee	Single Family	4	370000
0.9	Nelson	Ranch	4	340000
0.7	Ortiz	Single Family	3	335000
0.3	Miller	Penthouse	2	290000

# Cartesian Products and Similarity-Based Joins

**Cartesian product**  $\mathcal{D}_1 \times \mathcal{D}_2$ :

$$(\mathcal{D}_1 \times \mathcal{D}_2)(st) = \mathcal{D}_1(s) \otimes \mathcal{D}_2(t), \quad \text{where } \mathcal{D}_1(s) > 0 \text{ and } \mathcal{D}_2(t) > 0.$$

**Join**  $\mathcal{D}_1 \bowtie_{\varphi} \mathcal{D}_2$  of  $\mathcal{D}_1$  and  $\mathcal{D}_2$  by selection formula  $\varphi$ :

$$\mathcal{D}_1 \bowtie_{\varphi} \mathcal{D}_2 = \sigma_{\varphi}(\mathcal{D}_1 \times \mathcal{D}_2) .$$

**Notes:**

- generalizes (inner)  $\theta$ -joins;
- special case:  $\mathcal{D}_1 \bowtie_{y_1=y_2} \mathcal{D}_2$  (equi-join / join over similar values);
- natural join (issues with duplicate column);
- other types of joins:
  - joins with thresholds (combination with  $\alpha$ -cuts);
  - semijoins, ...

## Example (Similarity-Based Join: Input Data)

### Buyers ( $\mathcal{D}_1$ ) and Sellers ( $\mathcal{D}_2$ )

<i>name1</i>	<i>type1</i>	<i>bedrooms1</i>	<i>price1</i>	<i>credit-score</i>
Adams	Single Family	3	250000	628
Black	Single Family	3	325000	769
Chang	Residential	4	300000	535
Davis	Condominium	1	200000	567
Enke	Ranch	3	240000	635
Flores	Penthouse	2	200000	659

<i>name2</i>	<i>type2</i>	<i>bedrooms2</i>	<i>year</i>	<i>price2</i>	<i>tax</i>
Kelly	Log Cabin	1	1956	85000	1250
Lee	Single Family	4	1975	370000	8350
Miller	Penthouse	2	1994	250000	2100
Nelson	Ranch	4	1969	320000	6500
Ortiz	Single Family	3	1982	250000	4350

## Example (Similarity-Based Join: Results)

$\pi_{\{name1, price1, price2, type1, type2\}}(top_5(D_1 \bowtie_{type1=type2} \otimes_{price1=price2} D_2))$

	<i>name1</i>	<i>price1</i>	<i>price2</i>	<i>type1</i>	<i>type2</i>
1.0	Adams	250000	250000	Single Family	Single Family
0.8	Black	325000	370000	Single Family	Single Family
0.8	Flores	200000	250000	Penthouse	Penthouse
0.7	Black	325000	320000	Single Family	Ranch
0.7	Enke	240000	250000	Ranch	Single Family

## Selected Laws

$$\begin{aligned}\mathcal{D}_1 \otimes (\mathcal{D}_2 \cup \mathcal{D}_3) &= (\mathcal{D}_1 \cup \mathcal{D}_2) \otimes (\mathcal{D}_1 \cup \mathcal{D}_3), \\ \mathcal{D}_1 \rightarrow (\mathcal{D}_2 \cap \mathcal{D}_3) &= (\mathcal{D}_1 \rightarrow \mathcal{D}_2) \cap (\mathcal{D}_1 \rightarrow \mathcal{D}_3),\end{aligned}$$

$$\pi_A(\mathcal{D}_1 \cup \mathcal{D}_2) = \pi_A(\mathcal{D}_1) \cup \pi_A(\mathcal{D}_2),$$

$$\pi_A(\mathcal{D}^*) \subseteq \pi_A(\mathcal{D}),$$

$$\pi_A(\mathcal{D}_1 \cap \mathcal{D}_2) \subseteq \pi_A(\mathcal{D}_1) \cap \pi_A(\mathcal{D}_2),$$

$$\pi_A(a \rightarrow \mathcal{D}) \subseteq a \rightarrow \pi_A(\mathcal{D}),$$

$$\pi_A(\mathcal{D}_1 \otimes \mathcal{D}_2) \subseteq \pi_A(\mathcal{D}_1) \otimes \pi_A(\mathcal{D}_2),$$

$$\pi_A(a \otimes \mathcal{D}) = a \otimes \pi_A(\mathcal{D}),$$

$$\sigma_\varphi(\mathcal{D}_1 \cup \mathcal{D}_2) = \sigma_\varphi(\mathcal{D}_1) \cup \sigma_\varphi(\mathcal{D}_2),$$

$$\sigma_\varphi(\mathcal{D}_1 \cap \mathcal{D}_2) \subseteq \sigma_\varphi(\mathcal{D}_1) \cap \mathcal{D}_2,$$

$$\sigma_\varphi(\mathcal{D}_1 \otimes \mathcal{D}_2) = \mathcal{D}_1 \otimes \sigma_\varphi(\mathcal{D}_2),$$

$$\sigma_\varphi(\mathcal{D}_1 \otimes \mathcal{D}_2) = \sigma_\varphi(\mathcal{D}_1) \otimes \mathcal{D}_2.$$

## Relational Operation $top_k$

**Tuples with  $k$  best ranks:**  $top_k(\mathcal{D})$

can be defined in our model by

$$(top_k(\mathcal{D}))(t) = \mathcal{D}(t) \otimes (Q_{<k}t')(\neg(\mathcal{D}(t') \rightarrow \mathcal{D}(t))^* \otimes (\mathcal{D}(t) \rightarrow \mathcal{D}(t'))^*),$$

where

- $*$  is globalization, and
- $(Q_{<n}x)\varphi$  is a quantifier “there are at most  $n - 1$  objects  $x$  having  $\varphi$ ”  
**generalized quantifiers** (Hájek: *Metamathematics of Fuzzy Logic*)

**Explanation:**  $\neg(\mathcal{D}(t') \rightarrow \mathcal{D}(t))^* \otimes (\mathcal{D}(t) \rightarrow \mathcal{D}(t'))^* = 1$  iff

- $\mathcal{D}(t) \leq \mathcal{D}(t')$  and
- $(\mathcal{D}(t') \rightarrow \mathcal{D}(t))^* = 0$  which is iff  $\mathcal{D}(t') \rightarrow \mathcal{D}(t) \neq 1$  iff  $\mathcal{D}(t') \not\leq \mathcal{D}(t)$

Altogether:  $\mathcal{D}(t) < \mathcal{D}(t')$ .

**Conclusion:**  $top_k$  is relational operation (!!!)

# Further Results Available

## Data Dependencies

- (approximate) functional dependencies (axiomatization, bases, ...)
- similarity issues
- (domain) relational calculus

## Relational Querying

- similarity issues
- (domain) relational calculus

## Future Research

- alternative axiomatizations, query languages, (approximate) constraints,
- further data dependencies (multivalued FFDs, cyclic dependencies, join dependencies),
- prototypes, implementation (FCALISP already available).