

# **Relational Database Module**

---

Relational Strict and Fuzzy Database User's Manual  
Version 0.1  
30 July 2014

**by Mohammad Deni Akbar, Kyushu University.**

Ver. 1.0 30 July 2014



# 1 Introduction

This is a Mathematica Module for database theory especially for relational database. Using database theory, we also develop theory for using fuzzy database. Implement database operation and also fuzzy relation using relational calculus to make some formula in Mathematica.

To use this package "RelationDatabase", users should set a directory where the modules is stored.

[Example]

```
SetDirectory[
  FileNameJoin[$HomeDirectory, "Dropbox/Ymken2013 (1)/2013_Deni"];
<< RelationDatabase';
```

## 2 Relational Database

### 2.1 Basic Notations

#### 2.1.1 AllPairs

AllPairs[*alldata*]

:: Create all pairs in a list *alldata*.

*alldata* List of data

*return* List of pairs

[Example]

```
DB[Name] = {"Deni", "Herpin", "Fatih"};
```

```
AllPairs[DB[Name]]
```

```
{{"Deni", "Deni"}, {"Deni", "Herpin"}, {"Deni", "Fatih"}, {"Herpin",
  "Deni"}, {"Herpin", "Herpin"}, {"Herpin", "Fatih"}, {"Fatih",
  "Deni"}, {"Fatih", "Herpin"}, {"Fatih", "Fatih"}}
```

#### 2.1.2 not

not[A, Y] :: Difference set between A and Y.

A, Y set

*return* set

[Example]

```
A = {Name, Subject, Room}; y = {Name, Subject};
```

```
not[A,y]
```

```
{Room}
```

#### 2.1.3 RelComp

RelComp[x, y]

:: Composition relation of x and y.

x, y relations

*return* relation

[Example]

```
x = {"Deni", "Mathematics"}, {"Herpin", "Algorithm"}, {"Fatih",
  "Mathematics"};
```

```
y = {"Mathematics", 103}, {"Algorithm", 205};
```

```
RelComp[x,y]
```

```
{{Deni,103},{Fatih,103},{Herpin,205}}
```

```
{1,1}
```

#### 2.1.4 RelInv

RelInv[a]

:: Invers of relation a, a is pair of element attribute.

```

a      relations(pair of element attribute)
return relations(pair of element attribute)
      [Example]
      x = {"Deni", "Mathematics"}, {"Herpin", "Algorithm"}, {"Fatih",
          "Mathematics"};
      RelInv[x]
      {"Mathematics", "Deni"}, {"Algorithm",
          "Herpin"}, {"Mathematics", "Fatih"}

```

### 2.1.5 Id

```

Id[a]   ::Identity relation element of set a
a       set
return  relation set(pair of elements set)
      [Example]
      y = {"Mathematics", 103}, {"Algorithm", 205};
      Id[y]
      {"Mathematics", 103}, {"Mathematics", 103}, {"Algorithm",
          205}, {"Algorithm", 205}}

```

### 2.1.6 r

```

r[a]    ::relation contains pair of set a
a       set
return  pair of element set
      [Example]
      Name = 1; Subject = 2; Class = {Name, Subject};
      DB[Name] = {"Deni", "Herpin", "Fatih"};
      DB[Subject] = {"Mathematics", "Algorithm"};

      r[DB[Name]]
      {"Deni", "Deni"}, {"Fatih", "Fatih"}, {"Herpin", "Herpin"}}

```

### 2.1.7 SubsetQ

```

SubsetQ[a,b]
          ::to confirm that every element of set a is subset of set b.
a,b      set
return   True or False
      [Example]
      DB[Name] = {"Deni", "Herpin", "Fatih"};

      SubsetQ[{"Deni", "Fatih"}, DB[Name]]
      True

```

## 2.1.8 DBProduct

### 2.1.8.1 Two Set

**DBProduct**[*a*,*b*]  
 ::Cartesian product between element of set attribute *a* and element of set attribute *b*.

*a, b* set

*return* Set

[Example]

```
DB[Name] = {"Deni", "Herpin", "Fatih"};
DB[Subject] = {"Mathematics", "Algorithm"};
```

```
DBProduct[DB[Name], DB[Subject]]
{"Deni", "Mathematics"}, {"Deni", "Algorithm"}, {"Herpin",
  "Mathematics"}, {"Herpin", "Algorithm"}, {"Fatih",
  "Mathematics"}, {"Fatih", "Algorithm"}}
```

### 2.1.8.2 Collection Sets

**DBProduct**[*l*]  
 ::Cartesian product of collection attribute *l*, element of *l* is attribute of database.

*l* collection of sets

*return* Set

[Example]

```
Name = 1; Subject = 2; Class = {Name, Subject};
DB[Name] = {"Deni", "Herpin", "Fatih"};
DB[Subject] = {"Mathematics", "Algorithm"};
```

```
DBProduct[Class]
{"Deni", "Mathematics"}, {"Deni", "Algorithm"}, {"Herpin",
  "Mathematics"}, {"Herpin", "Algorithm"}, {"Fatih",
  "Mathematics"}, {"Fatih", "Algorithm"}}
```

### 2.1.9 Idx

**Idx**[*x*,*a*] ::Position an attribute *a* in set of attribute *x*.

*a* An element of attribute set

*x* Set of attribute

*return* Position of *a* in *x*

[Example]

```
Name = 1; Subject = 2; Class = {Name, Subject};
DB[Name] = {"Deni", "Herpin", "Fatih"};
DB[Subject] = {"Mathematics", "Algorithm"};
```

```
Idx[Class, Name]
1
```

### 2.1.10 RhoXa

```
RhoXa[x, a]
    ::Pair of projection element of set attribute x to an attribute a, a is element of
    x
```

*a* An element of attribute set

*x* Set of attribute

*return* Set

[Example]

```
Name = 1; Subject = 2; Class = {Name, Subject};
DB[Name] = {"Deni", "Herpin", "Fatih"};
DB[Subject] = {"Mathematics", "Algorithm"};
```

```
RhoXa[Class, Name]
{{{"Deni", "Mathematics"}, "Deni"}, {"Deni", "Algorithm"},
  "Deni"}, {"Herpin", "Mathematics"},
  "Herpin"}, {"Herpin", "Algorithm"},
  "Herpin"}, {"Fatih", "Mathematics"},
  "Fatih"}, {"Fatih", "Algorithm"}, "Fatih"}}
```

### 2.1.11 Intersections

```
Intersections[l]
    ::Intersection of collection set.
```

*l* An Collection set

*return* Set

[Example]

```
Intersections[{{1, 2}, {1}}]
1
```

### 2.1.12 Rho

```
Rho[x, y] ::Pair of projection element of set attribute x to element of set attribute y, such
that set attribute y is subset of set attribute x
```

*x,y* Attribute set

*return* Relation Set between element of attribute set

[Example]

```
Name = 1; Subject = 2; Class = {Name, Subject};
DB[Name] = {"Deni", "Herpin", "Fatih"};
DB[Subject] = {"Mathematics", "Algorithm"};
```

```
Rho[Class, {Name}]
```

```

{{{ "Deni", "Algorithm"}, "Deni"}, {" "Deni", "Mathematics"},
  "Deni"}, {" "Fatih", "Algorithm"},
  "Fatih"}, {" "Fatih", "Mathematics"},
  "Fatih"}, {" "Herpin", "Algorithm"},
  "Herpin"}, {" "Herpin", "Mathematics"}, "Herpin"}}}

```

### 2.1.13 Theta

Theta[A,x]

::Pairs of equivalence relation set of attribute x from complete set of attribute A, with condition attribute set x is subset of attribute sets A

A,x      Attribute set

return    Sets relation between element of attribute set x and attribute set A

[Example]

```

Name = 1; Subject = 2; Class = {Name, Subject};
DB[Name] = {"Deni", "Herpin", "Fatih"};
DB[Subject] = {"Mathematics", "Algorithm"};

```

Theta[Class, {Name}]

```

{{{ "Deni", "Algorithm"}, {"Deni", "Algorithm"}, {" "Deni",
  "Algorithm"}, {"Deni", "Mathematics"}, {" "Deni",
  "Mathematics"}, {"Deni", "Algorithm"}, {" "Deni",
  "Mathematics"}, {"Deni", "Mathematics"}, {" "Fatih",
  "Algorithm"}, {"Fatih", "Algorithm"}, {" "Fatih",
  "Algorithm"}, {"Fatih", "Mathematics"}, {" "Fatih",
  "Mathematics"}, {"Fatih", "Algorithm"}, {" "Fatih",
  "Mathematics"}, {"Fatih", "Mathematics"}, {" "Herpin",
  "Algorithm"}, {"Herpin", "Algorithm"}, {" "Herpin",
  "Algorithm"}, {"Herpin", "Mathematics"}, {" "Herpin",
  "Mathematics"}, {"Herpin", "Algorithm"}, {" "Herpin",
  "Mathematics"}, {"Herpin", "Mathematics"}}}

```

## 2.2 Database Operations

### 2.2.1 Projection

DBProjection[A, rdb, y]

::Projection of relation database *rdb* with attributes A to attributes y, such that y is subset of A

A      Complete set of attribute with a relation

*rdb*    relations set of element every attribute

y      Set of attribute that we want to select, subset set of A

return    pair elements of attribute sets y

[Example]

```

Name = 1; Subject = 2; Class = {Name, Subject};

```



```

DB[Name] = {"Deni", "Herpin", "Fatih"};
DB[Subject] = {"Mathematics", "Algorithm"};
rClass = {"Deni", "Mathematics"}, {"Fatih", "Mathematics"};

DBProjection[Class, rClass, {Subject}]
{"Mathematics", "Mathematics"}

```

### 2.2.2 Natural Join

**NaturalJoin**[*x,y,r1,r2*]  
 ::Join relation between relation database *r1* with attributes *x* and relation database *r2* with attributes *y*

*x,y* Complete set of attribute of a relation

*r1, r2* relations set of element every attribute

*return* pair elements of union attribute sets *x* with elements *r1* and *y* with elements *r2*

[Example]

```

Name = 1; Subject = 2; Class = {Name, Subject};
DB[Name] = {"Deni", "Herpin", "Fatih"};
DB[Subject] = {"Mathematics", "Algorithm"};
rClass = {"Deni", "Mathematics"}, {"Fatih", "Mathematics"};

```

```

Room = 3;
DB[Room] = 103, 205, 304, 202;
Lecture = {Subject, Room};
rLecture = {"Mathematics", 103}, {"Algorithm", 205};

```

```

NaturalJoin[Class, Lecture, rClass, rLecture]
{"Deni", "Mathematics", 103}, {"Deni", "Mathematics",
103}, {"Fatih", "Mathematics", 103}, {"Fatih", "Mathematics",
103}}

```

### 2.2.3 Selection

**Selection**[*A,r,Condition*]  
 ::Selection of relation database *r* with attribute *A*, with some condition in attribute *a*, for example  $a < b$ ,  $a > b$ ,  $b < a < c$ , etc. *b* and *c* is value that we want

*A* Complete set of attribute of a relation

*r* relations set of element every attribute

*Condition* Condition that is selected, consider *a* is subset of *A*. So the condition is mean the condition of *a* that we want. For example, in relation Test has attribute set name and score,  $\text{Test} = \{\text{Name}, \text{Score}\}$ . From relation Test, we want to select only elements with condition:  $\text{Score} > 50$ .

*return* pair elements of relation sets.

[Example]

```

Name = 1; Subject = 2; Class = {Name, Subject};

```

```
DB[Name] = {"Deni", "Herpin", "Fatih"};
DB[Subject] = {"Mathematics", "Algorithm"};
rClass = {"Deni", "Mathematics"}, {"Fatih", "Mathematics"};

Score = 4; Result = {Name, Subject, Score};
DB[Score] = {0, 20, 50, 60, 70, 80, 90, 100};
rScore = {"Deni", "Mathematics", 90}, {"Deni", "Algorithm",
        80}, {"Herpin", "Mathematics", 60}, {"Herpin", "Algorithm",
        70}, {"Fatih", "Mathematics", 100}, {"Fatih", "Algorithm", 90}};

DBSelection[Result, rScore, Score > 70]
{{{"Deni", "Algorithm", 80}, {"Deni", "Algorithm", 80}}, {"Deni",
"Mathematics", 90}, {"Deni", "Mathematics", 90}}, {"Fatih",
"Algorithm", 90}, {"Fatih", "Algorithm", 90}}, {"Fatih",
"Mathematics", 100}, {"Fatih", "Mathematics", 100}}}
```

## 3 Fuzzy Relational Database

### 3.1 Basic Fuzzy Notations

#### 3.1.1 FuzzyRelComp

FuzzyRelComp[x, y]

:: Composition of fuzzy relation between element of fuzzy set attribute x and element of fuzzy set attribute y.

x, y        Fuzzy relations set.

return     Fuzzy relation set.

[Example]

```
rLikes = {"Deni", "Mathematics", 0.9}, {"Herpin", "Algorithm",
        0.5}, {"Fatih", "Mathematics", 0.7}};
rFill = {"Mathematics", "Coding", 0.6}, {"Algorithm", "Coding",
        0.8}, {"Mathematics", "Discrete Theory", 0.8}, {"Algorithm",
        "Discrete Theory", 0.6}};
```

FuzzyRelComp[rLikes, rFill]

```
{"Deni", "Coding", 0.6}, {"Deni", "Discrete Theory",
 0.8}, {"Herpin", "Coding", 0.5}, {"Herpin", "Discrete Theory",
 0.5}, {"Fatih", "Coding", 0.6}, {"Fatih", "Discrete Theory", 0.7}}}
```

#### 3.1.2 FuzzyId

FuzzyId[a]

:: Identity fuzzy relation element of fuzzy set a

a        set

return   relation fuzzy set(pair of elements set)

[Example]

```
rLikes = {"Deni", "Mathematics", 0.9}, {"Herpin", "Algorithm",
        0.5}, {"Fatih", "Mathematics", 0.7}};
```

FuzzyId[rLikes]

```
{"Deni", "Mathematics"}, {"Deni", "Mathematics"},
 0.9}, {"Herpin", "Algorithm"}, {"Herpin", "Algorithm"},
 0.5}, {"Fatih", "Mathematics"}, {"Fatih", "Mathematics"}, 0.7}}}
```

#### 3.1.3 ToFuzzy

ToFuzzy[r]

:: To give fuzzy value of set r

r        Relation sets

return   Relation fuzzy set with fuzzy value

```
[Example]
rExperience = {"Young", "Low"}, {"Old", "High"};
```

```
ToFuzzy[rExperience]
{ {"Young", "Low", 1. }, {"Old", "High", 1.} }
```

### 3.1.4 FuzzyRho

FuzzyRho[x,y]

::Pair of projection element of set attribute x to element of set attribute y with fuzzy value in every pair,such that y is subset of x

x,y Relation sets

return Relation fuzzy set with fuzzy value

```
[Example]
Name = 1; Subject = 2; Likes = {Name, Subject};
DB[Name] = {"Deni", "Herpin", "Fatih"};
DB[Subject] = {"Mathematics", "Algorithm"};
```

```
FuzzyRho[Likes, {Subject}]
{{"Deni", "Algorithm"}, "Algorithm",
 1.}, {"Deni", "Mathematics"}, "Mathematics",
 1.}, {"Fatih", "Algorithm"}, "Algorithm",
 1.}, {"Fatih", "Mathematics"}, "Mathematics",
 1.}, {"Herpin", "Algorithm"}, "Algorithm",
 1.}, {"Herpin", "Mathematics"}, "Mathematics", 1.}}
```

### 3.1.5 FuzzyRelInv

FuzzyRelInv[a]

::Invers of fuzzy relation a, a is pair of element attribute with fuzzy value.

a Fuzzy relations(pair of element attribute with fuzzy value)

return Fuzzy relations(pair of element attribute with fuzzy value)

```
[Example]
rFill = {"Mathematics", "Coding", 0.6}, {"Algorithm", "Coding",
 0.8}, {"Mathematics", "Discrete Theory", 0.8}, {"Algorithm",
 "Discrete Theory", 0.6}};
```

```
FuzzyRelInv[rFill]
{"Coding", "Mathematics", 0.6}, {"Coding", "Algorithm",
 0.8}, {"Discrete Theory", "Mathematics", 0.8}, {"Discrete Theory",
 "Algorithm", 0.6}}
```

### 3.1.6 FuzzyIntersection

#### 3.1.6.1 Two Sets

FuzzyIntersection[x,y]

::Intersection between fuzzy relation set x and fuzzy relation set y.

*x,y* Fuzzy relations set.

*return* Fuzzy relations set.

[Example]

```
rMathDept = {"Mathematics", "Coding", 0.6}, {"Algorithm", "Coding",
0.8}, {"Mathematics", "Discrete Theory", 0.8}, {"Algorithm",
"Discrete Theory", 0.6}};
rInfDept = {"Mathematics", "Coding", 0.9}, {"Algorithm", "Coding",
0.9}, {"Mathematics", "Discrete Theory", 0.8}, {"Algorithm",
"Discrete Theory", 0.6}};
```

```
FuzzyIntersection[rMathDept, rInfDept]
{"Mathematics", "Coding", 0.6}, {"Algorithm", "Coding",
0.8}, {"Mathematics", "Discrete Theory", 0.8}, {"Algorithm",
"Discrete Theory", 0.6}}
```

### 3.1.6.2 Collection set

FuzzyIntersection[d]

::Intersection between collection of fuzzy relation sets in one set *d*.

*d* Collection of fuzzy relations sets in one set.

*return* Fuzzy relations set.

[Example]

```
rAll= {"Mathematics", "Coding", 0.6}, {"Algorithm", "Coding",
0.8}, {"Mathematics", "Discrete Theory", 0.8}, {"Algorithm",
"Discrete Theory", 0.6}, {"Mathematics", "Coding", 0.9}, {"Algorithm", "Coding",
0.9}, {"Mathematics", "Discrete Theory", 0.8}, {"Algorithm",
"Discrete Theory", 0.6}};
```

```
FuzzyIntersection[rAll]
{"Algorithm", "Coding", 0.8}, {"Algorithm",
"Discrete Theory", 0.6}, {"Mathematics", "Coding",
0.6}, {"Mathematics", "Discrete Theory", 0.8}}
```

### 3.1.7 FuzzyUnion

#### 3.1.7.1 Two Sets

FuzzyUnion[x,y]

::Union between fuzzy relation set *x* and fuzzy relation set *y*.

*x,y* Fuzzy relation set.

*return* Fuzzy relation set.

[Example]

```
rMathDept = {"Mathematics", "Coding", 0.6}, {"Algorithm", "Coding",
0.8}, {"Mathematics", "Discrete Theory", 0.8}, {"Algorithm",
"Discrete Theory", 0.6}};
```

```
rInfDept = {"Mathematics", "Coding", 0.9}, {"Algorithm", "Coding",
0.9}, {"Mathematics", "Discrete Theory", 0.8}, {"Algorithm",
"Discrete Theory", 0.6}};
```

```
FuzzyUnion[rMathDept, rInfDept]
{"Mathematics", "Coding", 0.9}, {"Algorithm", "Coding",
0.9}, {"Mathematics", "Discrete Theory", 0.8}, {"Algorithm",
"Discrete Theory", 0.6}}
```

### 3.1.7.2 Collections set

FuzzyUnion[d]

::Union between collection of fuzzy relation sets in one set *d*.

*d* Collection of fuzzy relations sets in one set.

return Fuzzy relations set.

[Example]

```
rAll= {"Mathematics", "Coding", 0.6}, {"Algorithm", "Coding",
0.8}, {"Mathematics", "Discrete Theory", 0.8}, {"Algorithm",
"Discrete Theory", 0.6}, {"Mathematics", "Coding", 0.9}, {"Algorithm", "Coding",
0.9}, {"Mathematics", "Discrete Theory", 0.8}, {"Algorithm",
"Discrete Theory", 0.6}};
```

FuzzyUnion[rAll]

```
{"Algorithm", "Coding", 0.9}, {"Algorithm",
"Discrete Theory", 0.6}, {"Mathematics", "Coding",
0.9}, {"Mathematics", "Discrete Theory", 0.8}}
```

## 3.2 Fuzzy Operations

### 3.2.1 FuzzySelection

FuzzySelection[A,r,Condition]

::Selection of fuzzy relation set database *r* with attribute *A*, with some condition in attribute *a*, for example  $a < b$ ,  $a > b$ ,  $b < a < c$ , etc. *b* and *c* is value that we want

*A* Complete set of attribute of a fuzzy relation

*r* Fuzzy relations set of element every attribute

*Condition* Condition that is selected, consider *a* is subset of *A*. So the condition is mean the condition of *a* that we want. For example, in relation Test has attribute set name and score,  $\text{Test} = \{\text{Name}, \text{Score}\}$ . From relation Test, we want to select only elements with condition:  $\text{Score} > 50$ .

return pair elements of fuzzy relation sets.

[Example]

```
Name = 1; Subject = 2; Likes = {Name, Subject};
DB[Name] = {"Deni", "Herpin", "Fatih"};
DB[Subject] = {"Mathematics", "Algorithm"};
```

```

rLikes = {"Deni", "Mathematics", 0.9}, {"Herpin", "Algorithm",
        0.5}, {"Fatih", "Mathematics", 0.7}};

FuzzySelection[Likes, rLikes, Subject == "Mathematics"]
{{{ "Deni", "Mathematics"}, {"Deni", "Mathematics"},
  0.9}, {" "Fatih", "Mathematics"}, {"Fatih", "Mathematics"}, 0.7}}

```

### 3.2.2 FuzzyProjection

**FuzzyProjection**[*A*, *rdb*, *y*]  
 ::Projection of fuzzy relation set database *rdb* with attributes *A* to attributes *y*, such that *y* is subset of *A*

*A* Complete set of attribute with a fuzzy relation

*rdb* Fuzzy relations set of element every attribute

*y* Set of attribute that we want to select, subset set of *A*

*return* pair elements of attribute fuzzy sets *y*

[Example]

```

Name = 1; Subject = 2; Theory = 5;
DB[Name] = {"Deni", "Herpin", "Fatih"};
DB[Subject] = {"Mathematics", "Algorithm"};
DB[Theory] = {"Coding", "Discrete Theory"};
Complete = {Name, Subject, Theory};
rComplete = {"Deni", "Mathematics", "Coding", 0.8}, {"Deni",
  "Mathematics", "Discrete Theory", 0.9}, {"Herpin", "Algorithm",
  "Coding", 0.8}, {"Herpin", "Mathematics", "Discrete Theory",
  0.6}, {"Fatih", "Mathematics", "Coding", 0.6}, {"Fatih",
  "Algorithm", "Discrete Theory", 0.9}};

```

```

FuzzyProjection[Complete, rComplete, Subject, Theory]
{{{ "Algorithm", "Coding"}, {"Algorithm", "Coding"},
  0.8}, {" "Algorithm", "Discrete Theory"}, {"Algorithm",
  "Discrete Theory"},
  0.9}, {" "Mathematics", "Coding"}, {"Mathematics", "Coding"},
  0.8}, {" "Mathematics", "Discrete Theory"}, {"Mathematics",
  "Discrete Theory"}, 0.9}}

```

### 3.2.3 FuzzyNaturalJoin

**NaturalJoin**[*x*, *y*, *r1*, *r2*]  
 ::Join relation between fuzzy relation set database *r1* with attributes *x* and fuzzy relation set database *r2* with attributes *y*

*x*, *y* Complete set of attribute of a fuzzy relation

*r1*, *r2* Fuzzy relations set of element every attribute

*return* pair elements of union attribute fuzzy sets *x* with elements *r1* and *y* with elements *r2*

```

[Example]
Name = 1; Subject = 2; Likes = Name, Subject;
DB[Name] = {"Deni", "Herpin", "Fatih"};
DB[Subject] = {"Mathematics", "Algorithm"};
Theory = 5;
DB[Theory] = "Coding", "Discrete Theory"; Fill = Subject, Theory;
rLikes = {"Deni", "Mathematics", 0.9}, {"Herpin", "Algorithm",
          0.5}, {"Fatih", "Mathematics", 0.7}};
rFill = {"Mathematics", "Coding", 0.6}, {"Algorithm", "Coding",
         0.8}, {"Mathematics", "Discrete Theory", 0.8}, {"Algorithm",
         "Discrete Theory", 0.6}};

FuzzyNaturalJoin[Likes, Fill, rLikes, rFill]
{{{"Deni", "Mathematics", "Coding"}, {"Deni", "Mathematics",
   "Coding"},
  0.6}, {"Deni", "Mathematics", "Discrete Theory"}, {"Deni",
   "Mathematics", "Discrete Theory"},
  0.8}, {"Fatih", "Mathematics", "Coding"}, {"Fatih", "Mathematics",
   "Coding"},
  0.6}, {"Fatih", "Mathematics", "Discrete Theory"}, {"Fatih",
   "Mathematics", "Discrete Theory"},
  0.7}, {"Herpin", "Algorithm", "Coding"}, {"Herpin", "Algorithm",
   "Coding"},
  0.5}, {"Herpin", "Algorithm", "Discrete Theory"}, {"Herpin",
   "Algorithm", "Discrete Theory"}, 0.5}}

```



# Index

## A

AllPairs ..... 2

## C

Collection sets ..... 4

## F

FuzzyId ..... 9

FuzzyIntersection ..... 10

FuzzyIntersection[d] ..... 11

FuzzyIntersection[x,y] ..... 10

FuzzyNaturalJoin ..... 13

FuzzyProjection ..... 13

FuzzyRelComp ..... 9

FuzzyRelInv ..... 10

FuzzyRho ..... 10

FuzzyUnion[d] ..... 11, 12

FuzzyUnion[x,y] ..... 11

## I

Id ..... 3

Idx ..... 4

Intersections ..... 5

## N

Natural Join ..... 7

not ..... 2

## P

Projection ..... 6

## R

r ..... 3

RelComp ..... 2

RelInv ..... 2

Rho ..... 5

RhoXa ..... 5

## S

Selection ..... 7, 12

SubsetQ ..... 3

## T

Theta ..... 6

ToFuzzy ..... 9

Two sets ..... 4

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Relational Database</b>	<b>2</b>
2.1	Basic Notations	2
2.1.1	AllPairs	2
2.1.2	not	2
2.1.3	RelComp	2
2.1.4	RelInv	2
2.1.5	Id	3
2.1.6	r	3
2.1.7	SubsetQ	3
2.1.8	DBProduct	4
2.1.8.1	Two Set	4
2.1.8.2	Collection Sets	4
2.1.9	Idx	4
2.1.10	RhoXa	5
2.1.11	Intersections	5
2.1.12	Rho	5
2.1.13	Theta	6
2.2	Database Operations	6
2.2.1	Projection	6
2.2.2	Natural Join	7
2.2.3	Selection	7
<b>3</b>	<b>Fuzzy Relational Database</b>	<b>9</b>
3.1	Basic Fuzzy Notations	9
3.1.1	FuzzyRelComp	9
3.1.2	FuzzyId	9
3.1.3	ToFuzzy	9
3.1.4	FuzzyRho	10
3.1.5	FuzzyRelInv	10
3.1.6	FuzzyIntersection	10
3.1.6.1	Two Sets	10
3.1.6.2	Collection set	11
3.1.7	FuzzyUnion	11
3.1.7.1	Two Sets	11
3.1.7.2	Collections set	12
3.2	Fuzzy Operations	12
3.2.1	FuzzySelection	12
3.2.2	FuzzyProjection	13
3.2.3	FuzzyNaturalJoin	13
	<b>Index</b>	<b>15</b>