Chapter 7

Fuzzy Queries from Databases: Applications

Database is an organized structure designed with the help of computer science to store, relate, and retrieve data. Standard databases contain crisp data which can be retrieved by formulating crisp queries. The concept of standard database has been generalized by the means of fuzzy sets and fuzzy logic in order to include and handle vague, incomplete, and contradictory data. In this chapter we concentrate on formulating queries of fuzzy nature to the database for instance "which funds have a big asset increase and high return." These types of fuzzy queries can be used as a decision aid in various business, finance, and management activities. Applications involve small companies, stocks, and mutual funds.

7.1 Standard Relational Databases

There are many types of standard databases with crisp data called also classical databases. We review briefly only *relational databases*[1]; they provide the foundation for the *fuzzy databases*.[2]

A standard relational database consists of a group of relations expressed as tables made of columns and rows. The names of the columns are called *attributes*. The cells in a column form the *domain* of the

attribute. The rows called *tuples* contain records or entries each occupying a cell. Several tables having common domains connected together represent a *relational database*.

Example 7.1

Typical inventory records contain whatever data are relevant such as part number, part name, standard cost, quantity, specification, size, color, weight, supplier, etc. Table 7.1 formed by three connected tables represent a simplified inventory relational database of a small aircraft component manufacturing company.

Table 7.1. Inventory relational database of a small aircraft component manufacturing company.

PART

P#	P NAME	SPECIFICATION	SIZE	CITY
P1	Solid rod	QA 225/6	144 in	Pico Rivera (CA)
P2	Plate	MS 516-02	6912 si	Los Angeles (CA)
P3	Sheet	QA 250/5	45 sf	Los Angeles (CA)
P4	Rubber	MS 2221	96 in	Tukwilla (WA)

SUPPLIER

S#	S NAME	CITY
S1	Aero-Space Metals	Pico Rivera
S2	Ruber and Metal	Tukwilla
S3	Metal Products	Los Angeles

SHIPPING

S#	P#	QUANTITY
S1	P1	30
S2	P1	20
S2	P4	120
S3	P3	15
S3	P4	55

This relational database above is made of three related tables: PART, SUPPLIER, and SHIPPING. For instance in the table labeled PART the first row or tuple starting with $P1$ is usually represented as

$< P_1$, Solid rod, QA225/6, 144in, Pico Rivera (CA) $>$. The attributes in PART are P#, P NAME, SPECIFICATION, SIZE, CITY; the domain of the attribute P NAME consists of solid rod, plate, sheet, rubber. The framework of the database can be written as

PART (P#, P NAME, SPECIFICATION, SIZE, CITY),
SUPPLIER (S#, S NAME, CITY),
SHIPPING (S#, P#, QUANTITY).

□

Searching and finding data of interest out of a database is a process called *retrieval of data*. For the retrieval of data from a standard database a query language call SEQUEL (Structured English Query Language) was design (see Chamberlin and Boyce (1974)).

Access to the data is made by the SELECT command followed by clarifications FROM and WHERE (or WITH). SELECT command means to select attributes FROM one or more specified tables. WHERE means to select in the query process rows from a table that meet certain specified condition. The attributes are considered to be crisp objects; the query is called *standard query*.

Example 7.2

Consider the standard query from the relational database in Table 7.1 (Example 7.1):

SELECT NAME
FROM PART
WHERE QUANTITY < 100

The outcome of the query is given in Table 7.2.

Table 7.2. Parts whose quantity is smaller than 100.

S#	P#	QUANTITY
$S1$	$P1$	30
$S2$	$P2$	20
$S3$	$P3$	15

□

7.2 Fuzzy Queries

The query language SEQUEL has been used also to retrieve data when the query is of fuzzy nature (Tahani (1977)). By this we mean that the attributes of the database are considered to be linguistic variables.

The difference between standard and fuzzy query is outlined in the following case study.

Case Study 25 (Part 1) *Retrieval from a Small Company Employee Database*

Consider an employee database of a small company shown in Table 7.3. The employees are labeled by $E_i, i = 1, \ldots, 16$.

Table 7.3. Employee database of a small company.

NAME	AGE	SALARY
E_1	30	28,000
E_2	25	24,000
E_3	30	35,000
E_4	34	38,000
E_5	20	24,000
E_6	55	76,000
E_7	25	30,000
E_8	40	80,000
E_9	36	42,000
E_{10}	54	65,000
E_{11}	38	40,000
E_{12}	28	34,000
E_{13}	46	50,000
E_{14}	50	110,000
E_{15}	63	40,000
E_{16}	42	72,000

1. Standard retrieval of data

A simple standard query from the database in Table 7.3 involving only two attributes, name and age, can be presented in the form

SELECT NAME
FROM EMPLOYEE
WHERE $35 \leq$ AGE ≤ 45

The intent of the query is to select middle age employees where middle is defined by the interval $[35, 45]$ on a scale measured in years. Table 7.4 shows the result of the query.

Table 7.4. Standard query where age is between 35 and 45.

NAME	AGE
E_8	40
E_9	36
E_{11}	38
E_{16}	42

Employee E_8, whose age is 40—in the middle of the interval [35, 45]—fits best the intent of the query. Then follow employees E_{11} and E_{16}, and employee E_9 who, although close to the lower boarder 35, is still inside the interval.

From Talbel 7.3 we see that employee E_4 (age 34) lacks one year to be considered as middle age and employee E_{13} (age 46) is one year older than the upper boarder 45; they do not qualify for inclusion in Table 7.4. However, they could be included with a note that they are close to the boundaries (cut-off points) of the interval [35, 45]. Another option is to change the boundaries of the interval describing middle age. Assume the new interval is [30, 50]. Then five more employees, E_1, E_3, E_4, E_{13}, and E_{14} are to be added to Table 7.4. But then employees E_1 (age 30), E_3 (age 30), and E_{14} (age 50) who are borderline cases qualify equally to be on the list middle age as employee E_8 (age 40). In other words, there is no graduation concerning age between the employees.

A further extension of the interval to [25, 55] will include employees E_2 (age 25), E_7 (age 25), and E_{10} (age 54) into Table 7.4. But who will accept a person of 25 years to be characterized as being middle age.

We encounter similar difficulty with a query from the database on Table 7.3 when dealing with the attributes name and salary:

SELECT NAME
FROM EMPLOYEE

WHERE SALARY $\geq$ 80,000

The intent of the query is to select employees with high salary defined as 80,000 or greater. The search produces Table 7.5 with only two employees.

Table 7.5. Standard query where salary $\geq$ 80,000.

NAME	SALARY
E_8	80,000
E_{14}	110,000

Employee E_6 (salary 76,000) does not qualify to be in the table. Moving the boundary down, from 80,000 to 75,000 will include E_6, but not E_{16} (salary 72,000). Also there is no gradation between 80,000 and 110,000.

From the standard queries considered here arise the questions: does the definitions of middle age and high salary lacking any gradation reflect the intention of the query? If we start changing the boundaries of the defining intervals, where we have to stop?

The problem is rooted in the words *middle age* and *high salary*. They are linguistic values and can be defined better by recognizing their fuzzy nature.

2. Fuzzy retrieval of data

The attribute name on Table 7.3 is crisp but the attributes *age* and *salary* are fuzzy. They are linguistic variables (see Section 2.4). For instance in Example 2.4 (Section 2.4) *age* is described by five terms while in Case Study 20 (Section 6.1) it is described by three terms. That depends on the context in which *age* is seen, say by a medical doctor, financial expert, or a personnel officer.

Suppose that for the present study the financial experts find it relevant to partition *age* and *salary* into the following terms (linguistic values):

$$\begin{aligned} Age &= \{young, middle, old\}, \\ Salary &= \{low, medium, high\} \end{aligned}$$

shown in Fig. 7.1 and Fig. 7.2.

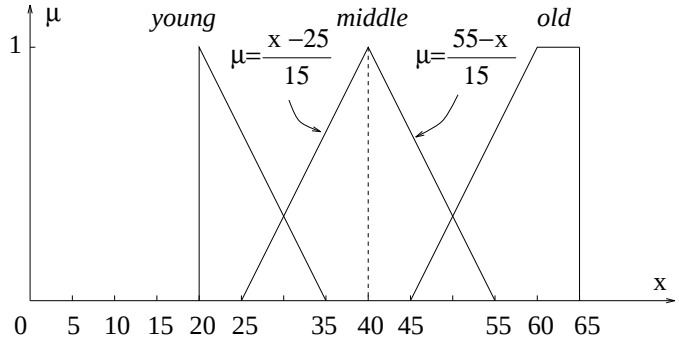

Fig. 7.1. Terms of the linguistic variable *age* in a Small Company Employee Database.

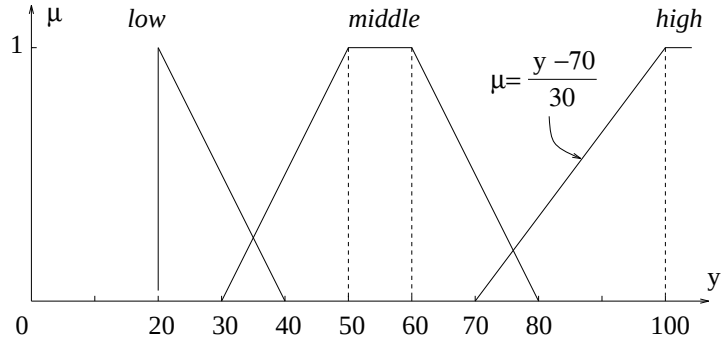

Fig. 7.2. Terms of the linguistic variable *salary* in a Small Company Employee Database.

The base variables x and y represent age in years and salary in thousands of dollars, correspondingly.

The membership functions of the terms in Fig. 7.1 and Fig. 7.2 overlap partially on the universal sets years and dollars. In Fig. 7.1 there is no overlapping on the intervals [15, 25], [35, 45], and [55, 65]; in Fig. 7.2 there is no overlapping on the intervals [20, 30], [40, 70], and [80, 100]. In most cases the terms are design to overlap entirely on the universal set, but this is not a mandatory requirement. It depends on the opinion of the experts dealing with a particular situation. Note that the terms of *age* in Fig. 7.1 have different supporting intervals from those of *age* in Case Study 20.

Now we make two simple fuzzy queries involving only one fuzzy attribute.

Query 1. Of employee database of a small company (Table 7.3) select employees who are *middle age*:

SELECT NAME
FROM EMPLOYEE
WHERE AGE IS MIDDLE

We have to match (Section 5.4) each entry in the second column (attribute AGE) (Table 7.3) with the term *middle* (Fig. 7.1) meaning to calculate the corresponding degree of membership. The term *middle* is represented by a triangular number on the supporting interval [25, 55]. The entries in the domain of AGE which fall in this interval substituted for x in $\mu = \frac{x-25}{15}$ for $25 < x < 40$ and $\mu = \frac{55-x}{15}$ for $40 < x < 55$ produce the ranked data in Table 7.6.

Table 7.6. Fuzzy query from a Small Company Employee Database: employee whose *age* is *middle*.

NAME	AGE MIDDLE	MEMBERSHIP DEGREE
E_8	40	1.00
E_{11}	38	0.87
E_{18}	42	0.87
E_9	36	0.73
E_4	34	0.60
E_{13}	46	0.60
E_1	30	0.33
E_3	30	0.33
E_{14}	50	0.33
E_{12}	28	0.20
E_{10}	54	0.07

Employee E_{10} has a very small membership grade 0.07, i.e. belongs little to the term *middle age*. The experts may decide to exclude E_{10} from the table if they establish a threshold value (see Section 1.3, pp. 14–15) for the membership grades, say 0.1. Then any grade below 0.1 is practically reduced to zero. Usually the threshold value is specified at the beginning of the query.

Employee E_8 is full member of the fuzzy set (term) *middle age* (membership degree 1), E_{11} and E_{16} are almost full members (degree 0.87), E_9 is close to full member (degree 0.73). In contrast, when classical query was used (Table 7.4), those employees had equal status as being of middle age. In the case of extended interval [30, 50] (classical query), employees E_3 and E_{14} who had the same status as E_8, now when the query is fuzzy belong to middle age only to degree 0.33.

Query 2. Of all employee in Table 7.3 select those with high salaries, i.e.

SELECT NAME
FROM EMPLOYEE
WHERE SALARY IS HIGH

The term *high salary* has a zero degree membership value below (including) 70,000 (see Fig. 7.2). Salaries above 70,000 qualify as high to various degrees. The entries 76,000, 80,000, 110,000, and 72,000 into the attribute salary in Table 7.3 have to be substituted for y in $\mu = \frac{y-70}{30}$ for $70 \leq y < 100$; for $y \geq 100$ the degree is one. The query produces the ranked Table 7.7.

Table 7.7. Fuzzy query from a Small Company Employee Database: employee with *high salary*.

NAME	SALARY HIGH	MEMBERSHIP DEGREE
E_{14}	110,000	1.00
E_8	80,000	0.33
E_6	76,000	0.20
E_{16}	72,000	0.07

Now let us compare Table 7.7 to Table 7.5 (classical query). Employee E_{14} (Table 7.7) is full member of the term *high salary*, E_8 has degree of membership 0.33, i.e. has a salary that is a little high. According to the classical query, both, E_{14} adn E_8 have high salary, i.e. have equal membership in the classical set salary $\geq$ 80,000. Employees E_6 and E_{16} are included in Table 7.7 but not in Table 7.5. Actually E_{16} whose membership degree is very low, only 0.07—below a threshold value 0.1, may be excluded from the list. While the standard query

has to specify a rigid salary (80,000) as a lower boundary below which salaries do not qualify as high, the fuzzy query using grades of the term *high* (Fig. 7.2) can include for consideration salaries close to 80,000 from below.

□

7.3 Fuzzy Complex Queries

Queries based on logical connectives

Most often a fuzzy SEQUEL query involves two or more fuzzy attributes in the WHERE predicate. They are joined by the logical connectives *conjunction* (*and*) and *disjunction* (*or*) defined by min and max in Section 2.1 formulas (2.2) and (2.3), correspondingly. The truth values of p and q in (2.2) and (2.3) are expressed by membership grades.

The asking of fuzzy complex queries is illustrated in a case study (continuation of Case Study 25 (Part 1)).

Case Study 25 (Part 2) *Fuzzy Complex Query from a Small Company Employee Database by Logical Connectives*

Query 3. Of all employee in Table 7.3 select those whose *age* is *middle* and *salary* is *high*:

```
SELECT NAME
FROM EMPLOYEE
WHERE AGE IS MIDDLE
        AND SALARY IS HIGH
```

In this query there are three attributes; name is a crisp one, *age* and *salary* are fuzzy (connected by *and*).

To facilitate the complex query we combine Table 7.3 with Table 7.6 and 7.7 into one containing the degree of membership of *high salary* and *middle age* (first five columns in Table 7.8).

The following abrievations are introduced in Table 7.8: A=AGE, N=NAME, DM=DEGREE MIDDLE, SAL=SALARY, DH=DEGREE HIGH, AVE=AVERAGE.

The task is to establish a list of employees who satisfy to various degrees the query.

Table 7.8. Fuzzy complex queries from a Small Company Employee Database.

N	A	DM	SAL	DH	AND	OR	AVE
E_1	30	0.33	28,000	0	0	0.33	0.17
E_2	25	0	24,000	0	0	0	0
E_3	30	0.33	35,000	0	0	0.33	0.17
E_4	34	0.60	38,000	0	0	0.6	0.3
E_5	20	0	24,000	0	0	0	0
E_6	55	0	76,000	0.2	0	0.20	0.10
E_7	25	0	30,000	0	0	0	0
E_8	40	1.00	80,000	0.33	0.33	1.0	0.67
E_9	36	0.73	42,000	0	0	0.73	0.37
E_{10}	54	0.07	65,000	0	0	0.07	0.04
E_{11}	38	0.87	40,000	0	0	0.87	0.44
E_{12}	28	0.20	34,000	0	0	0.20	0.10
E_{13}	46	0.60	50,000	0	0	0.60	0.30
E_{14}	50	0.33	110,000	1.00	0.33	1.00	0.67
E_{15}	63	0	40,000	0	0	0	0
E_{16}	42	0.87	72,000	0.07	0.07	0.87	0.44

For instance, for the first tuple in Table 7.3, $< E_1, 30, 28,000 >, E_1$ has the membership values $\mu_{middle}(30) = 0.33$ and $\mu_{high}(28) = 0$ in the terms *middle age* and *high salary* (see Table 7.8). The degree to which employee E_1 satisfies the query according to (2.2) is $\min(0.33, 0) = 0$. Hence E_1 is not included in the list. This is true for the employees who have at least one membership value equal to zero. Only the employees in the 8th,14th, and 16th tuples qualify to be in the list. For $E_8, \min(1.00, 0.33) = 0.33$; for $E_{14}, \min(0.33, 1.00) = 0.33$, and for $E_{16}, \min(0.87, 0.07) = 0.07$ (below threshold value 0.1). These results are registered in Table 7.8 in the 6th column labeled AND. We can say that they reflect the *degree of membership* of each employee *in the conclusion* in the query.

The fact that the degree of membership in the conclusion cannot be stronger (greater) than the weakest (smallest) individual grade is a conservative requirement. In some cases it can be a severe restriction on the query. For instance if a grade in one term is zero no matter what is

the value of the grade in the other terms, the degree of membership in the conclusion is also zero. That is why in Table 7.8, column AND, only three grades are different from zero. An alternative approach based on averaging is discussed at the end of this section.

Query 4. Of all employee in Table 7.3 select those whose *age* is *middle or salary* is *high*:

SELECT NAME
FROM EMPLOYEE
WHERE AGE IS MIDDLE
OR SALARY IS HIGH

In this query the two fuzzy attributes *age* and *salary* are connected by *or* (max), hence formula (2.3) applies. The employees who are either in Table 7.6 or in Table 7.7, or in both, qualify to be in the list. For instance, for employee $E_1, \max(0.33, 0) = 0.33$, for $E_2, \max(0, 0) = 0$, for $E_3, \max(0.33, 0) = 0.33$, for $E_4, \max(0.60, 0) = 0.60, \ldots$, for $E_{16}, \max(0.87, 0.07) = 0.87$. The results are presented in Table 7.8, 7th column labeled OR.

In conclusion, the numbers in the AND and OR columns indicate to what degree an employee satisfies the corresponding query. The degree is also interpreted as truth value for the query concerning each employee. □

Queries based on averaging

The joining of attributes in the WHERE predicate by the logical connective *and* can be replaced by the average (see (3.1), Section 3.1) of the individual degrees of membership. This technique ensures that each individual membership grade contributes to the degree of membership in the conclusion.

Case Study 25 (Part 3) *Fuzzy Complex Query from a Small Company Employee Database by using Averaging*

Consider again *Query 3* but instead of the connective *and* (min) let us use the average. From 3th and 5th columns of Table 7.8 we calculate: for $E_1, \frac{0.33+0}{2} = 0.17, \ldots$, for $E_6, \frac{0+0.20}{2} = 0.10, \ldots$, for $E_8, \frac{1+0.33}{2} = 0.67$, etc. The results are presented in Table 7.8 in the last column labeled

AVE. There are 12 employees in the list produced by the query while there were only three when then the connective *and* (min) was used.

□

7.4 Fuzzy Queries for Small Manufacturing Companies

Cox (1995) used a database consisting of small companies to show the advantage fuzzy queries have against standard queries. Here we present a case study which is typical of small manufacturing companies. The database is a modification of that considered by Cox. Also we model the attributes by triangular and trapezoidal numbers while in Cox they are described by bell-shaped fuzzy numbers.

Case Study 26 *Fuzzy Complex Queries of Database of Small Manufacturing Companies*

The database consists of 12 small companies labeled $C_i, i = 1, \ldots, 12$, listed in Table 7.9, ranked in 1996 according to their age measured in years.

Table 7.9. Database of small manufacturing companies in 1996.

CN	AGE	AR	PC	EC	PR	EPS
C_1	44	52	2	81	0.8	0.5
C_2	42	38	2	30	1.0	1.6
C_3	34	105	12	120	3.2	3.0
C_4	26	34	1	18	-0.3	0.3
C_5	24	47	6	64	1.4	2.5
C_6	23	92	8	70	2.6	2.2
C_7	17	68	5	48	0	0.2
C_8	16	65	6	44	2.0	5.0
C_9	12	90	4	50	1.0	2.4
C_{10}	8	70	3	109	-0.8	0
C_{11}	3	59	7	72	1.7	1.7
C_{12}	2	84	9	91	2.1	3.2

In this table only the first attribute—company—is crisp. The other six are considered to be fuzzy attributes (linguistic variables).

In Table 7.9 we use the notations: CN=COMPANY NAME, AR=ANNUAL REVENUE (in millions), PC=PRODUCT COUNT, EC=EMPLOYEE COUNT, PR=PROFIT (in millions), EPS = EARNING PER SHARE (in dollars).

To be able to make fuzzy queries we model the attributes by fuzzy sets (terms) shown below. The equations of the segments to be used later are given in the figures.

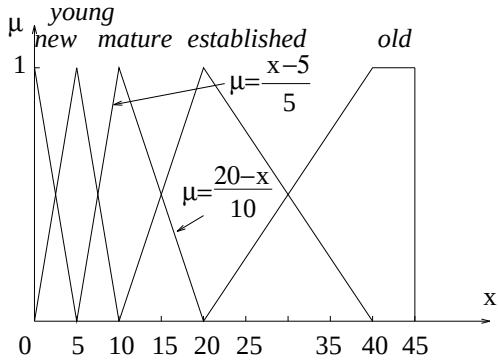

Fig. 7.3. Terms of company *age.*

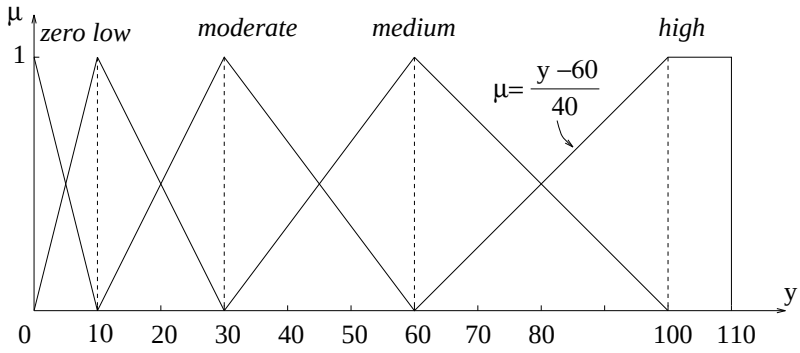

Fig. 7.4. Terms of *annual revenues.*

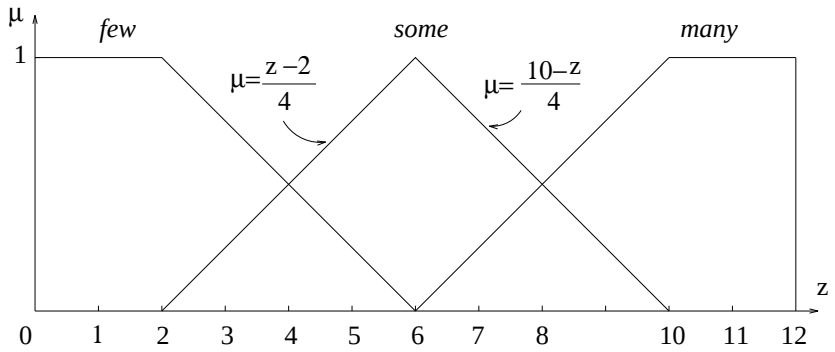

Fig. 7.5. Terms of *product count.*

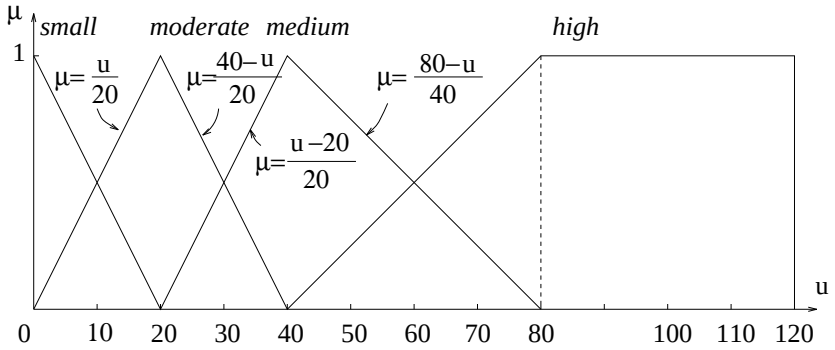

Fig. 7.6. Terms of *employee count.*

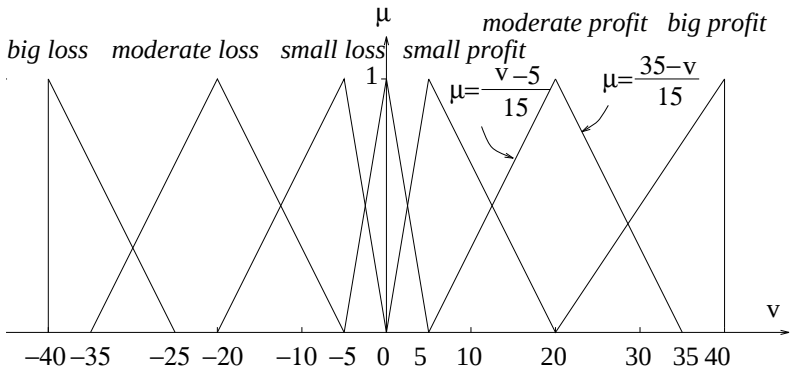

Fig. 7.7. Terms of *profit*; negative profit is *loss.*

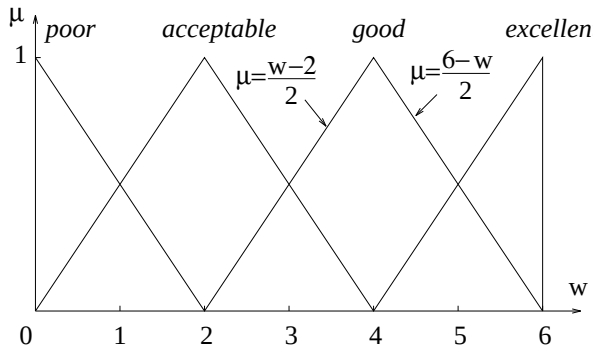

Fig. 7.8. Terms of *earnings per share.*

The base variables defined on the universal sets are measured as follows: x in years, y and v in millions of dollars, w in dollars, z and u are integer numbers.

We will use the database in Table 7.9 to make four complex queries.

Query 1 Consider the companies in Table 7.9.

SELECT NAME
FROM COMPANY
WHERE AGE IS MATURE
AND ANNUAL REVENUE IS HIGH
AND PRODUCT COUNT IS SOME
AND EMPLOYEE COUNT IS MODERATE
AND PROFIT IS MODERATE
AND EARNING PER SHARE IS GOOD

In this query all six attributes are involved. We have to repeat six times the matching procedure used in Case Study 25 (Part 1), Query 1. This will give the degree of membership of each entry in every term in the query which belongs to an appropriate attribute.

For instance the term *mature* in the attribute *age* (Fig. 7.3) is described by a triangular number on the supporting interval [5, 20] as follows: $\mu = \frac{x-5}{5}$ for $5 \leq x \leq 10$ and $\mu = \frac{20-x}{10}$ for $10 \leq x \leq 20$. The values (entries) 8, 12, 16, 20 of the domain of *age* which belong to [5, 20] have to be matched against the term *mature*. Substituting 8 (row C_{10}) into the first equation, 12 (row C_9), 16 (row C_8), and 17 (row C_7) into

the second equation gives μ the values 0.6, 0.8, 0.4, 0.3 correspondingly. The other entries of the domain of *age* are not in [5, 20]; they have zero degree of membership in the term *mature*. These results are recorded in Table 7.10, the second column—*age* is *mature*.

The same procedure is applied to the other five terms, *high*, *some*, *moderate*, *moderate*, *good* shown in Figs. 7.4–7.8, correspondingly. The membership degrees obtained are recorded in Table 7.10, third to seventh columns. The following short notations are used in Table 7.10: CN=COMPANY NAME, DMA=DEGREE MATURE, H=HIGH, S=SOME, DMOE=DEGREE MODERATE (concerning employee count), DMOP = DEGREE MODERATE PROFIT, DG = DEGREE GOOD.

The attributes in the query are connected by *and* (min). Most of the companies (excluding C_8 and C_9) have at least one entry 0, hence the outcome of the min operation is also 0 (column AND in Table 7.10). For instance, for company C_3, $\min(0, 1, 0, 0, 0.2, 0.5) = 0$; for C_8 we calculate $\min(0.4, 0.125, 1, 0.9, 1, 0.5) = 0.125$ and for C_9, $\min(0.8, 0.75, 0.5, 0.75, 0.33, 0.2) = 0.2$.

Table 7.10. Fuzzy complex Querie 1 from the database of small manufacturing companies.

CN	DMA	H	S	DMOE	DMOP	DG	AND	AVE
C_1	0	0	0	0	0.2	0	0	0.03
C_2	0	0	0	0.5	0.33	0	0	0.14
C_3	0	1	0	0	0.2	0.5	0	0.28
C_4	0	0	0	0	0	0	0	0
C_5	0	0	1	0.4	0.6	0.25	0	0.38
C_6	0	0.8	0.5	0.25	0.6	0.1	0	0.38
C_7	0.3	0.2	0.75	0.8	0	0	0	0.34
C_8	0.4	0.125	1	0.9	1	0.5	0.125	0.65
C_9	0.8	0.75	0.5	0.75	0.33	0.2	0.2	0.56
C_{10}	0.6	0.25	0.25	0	0	0	0	0.18
C_{11}	0	0	0.75	0.2	0.8	0	0	0.29
C_{12}	0	0.6	0.25	0	0.93	0.6	0	0.40

One can observe that as the number of *and* connections in the WHERE predicate increases the likelihood is that the membership grade

in the conclusion (AND) decreases. The contrary is true when the connection is *or* (see Query 2 which follows).

Let us use averaging instead of *and* (min) to connect the attributes (see Queries based on averaging, in Section 7.3). The results are recorded in the last column AVE in Table 7.10. For instance, for company C_3 we get the membership degree in the conclusion by adding the six entries in the same row and dividing the sum by 6, i.e. $\frac{0+1+0+0+0.2+0.5}{6} = 0.28$. Similarly for company C_8 we calculate $\frac{0.4+0.125+1+0.9+1+0.5}{6} = 0.65$.

Query 2.

```
SELECT NAME
FROM COMPANY
WHERE AGE IS MATURE
        OR ANNUAL REVENUES ARE HIGH
        OR PRODUCT COUNT IS SOME
        OR EMPLOYEE COUNT IS MODERATE
        OR PROFIT IS MODERATE
        OR EARNING PERSHARE IS GOOD
```

This query formally can be obtained from Query 1 by changing AND by OR. Hence now the attributes are connected by *or* (max). For company C_3 (Table 7.10) for instance we get $\max(0, 1, 0, 0, 0.2, 0.5) = 0.5$; for C_8, $\max(0.4, 0.125, 1, 0.9, 1, 0.5) = 1$. The results for all companies are given in the second column OR in Table 7.11.

Query 3.

```
SELECT NAME
FROM COMPANY
WHERE AGE IS MATURE
        AND ANNUAL REVENUES ARE HIGH
        AND EARNING PER SHARE IS GOOD
```

This query does not involve all attributes in the database. We use from Table 7.10 only the columns labeled DMA, H, and DG to find the membership degree in the conclusion AND (see Table 7.11).

Table 7.11. Fuzzy complex Queries 2, 3, 4 from the database of small manufacturing companies.

CN	Query 2 OR	Query 3 AND	Query 4 AND/OR
C_1	0.2	0	0
C_2	0.5	0	0
C_3	0.5	0	0
C_4	0	0	0
C_5	1	0	0.25
C_6	0.8	0	0.1
C_7	0.8	0	0.2
C_8	1	0.125	0.5
C_9	0.8	0.2	0.75
C_{10}	0.6	0	0.25
C_{11}	0.8	0	0
C_{12}	0.93	0	0

Query 4

SELECT NAME
FROM COMPANY
WHERE AGE IS MATURE
AND ANNUAL REVENUES ARE HIGH
OR EMPLOYEE COUNT IS MODERATE
AND EARNING PER SHARE IS GOOD

Four attributes take part in the WHERE predicate. They are joined by both connectives *and* and *or*. The membership grades for each tuple can be calculated from the schematically presented formula

[MATURE *and* HIGH] *or* [MODERATE *and* GOOD]

which can be written as

$$\max[\min(\text{MATURE, HIGH}), \min(\text{MODERATE, GOOD})], \qquad (7.1)$$

where the terms are substituted by the appropriate entries in the tuples.

We use the entries forming the domains of DMA, H, DMOE, and DG in Table 7.10. For instance for company C_8 formula (7.1) gives

$$[\max[\min(0.4, 0.125), \min(1, 0.5)]] = \max[0.125, 0.5] = 0.5$$

Similarly the rest of the membership grades are calculated and presented in the column AND/OR in Table 7.11.

□

7.5 Fuzzy Queries for Stocks and Funds Databases

Common stocks represent one of the most complex and varied fields of investment. The stock market is an arena in which success measured in profit depends not only on combination of skills, information, and knowledge, but also on unforeseen events of political and social character, drastic changes in nature, and on the subjectivity of investors expectations and confidence. There are thousand of stocks in the world that are traded in hundreds of stock exchanges. For a common investor to play on the stock market is both risky and time consuming. Stock markets go up and down generally along an increasing saw-line curve but also on rare occasions catastrophes called crashes happened. For instance the largest decline in one day in the history of the stock market, "Black Monday," occured on Monday, October 19, 1987. Then the Dow Jones Industrial Average in U.S.A. declined by 23 %; other countries also had a fast and large decline in their stock market. The worst stock market crash occured on 29 October, 1929. The consequences for millions of people were devastating.

Mutual funds are financial vehicles that offer portfolio diversification and professional management. One advantage is a great deal of time saved for the investor, but funds, in general less risky than stocks, are not risk-free. There are thousands of funds managed by financial corporations, companies, banks, and trusts. They are in fierce competition trying to perform better and attract more costumers. Fund managers are presenting their investment strategy and recommendations in various reports and letters. Buy and sell decisions usually reflect the consensus of several managers in charge of funds in a group.

Since the 1960s the stock markets have experienced fast changes. One major factor for that has been the advances in computer technology.

Computer selected stocks

Of particular interest is using computers to select stocks or funds in order to outperform the market. While there are activities in this area not much can be found in the literature.[4]

One such case was reported on a single page by Mandelman (1979). All U.S.A. stocks were screened with a computer. Aim: to select those that met five requirements:

"Low debt in the underlying company's capital structure.
A high return on equity.
A high dividend yield on the stock.
A very low PE ratio.
A low stock price."

Here PE means price–earnings ratio; it is a tool for comparing the relative merit of different stocks. For instance if a company A produces a product that has estimated year-end earnings of \$2 per share and the trading at the moment is \$12 per share, the PE ratio is $\frac{12}{2} = 6$. Another company B produces similar product with the same earnings of \$2 per share but the trading is \$16 per share, hence the PE ratio is $\frac{16}{2} = 8$. Then normally one could expect that company A is more attractive.

It is not explained how the border lines for "low debt," "high return," "high dividend," "very low PE ratio," and "low stock price" were determined. This might be a difficult task since the words "low debt" and "low stock price" require analysis and clarifications; "high dividend" is easier to define, say above \$4.50. Only nine stocks were selected and bought on March 12, 1979. On Oct. 16, after seven months, the gain was 15.7% (28.4% if annualized). This is considered in the report as a good gain under the specific circumstances at that time: "New York market was drifting sideways for much of the summer, and that we've taken the prices of the stocks on October 16—well after the big slump that began October 8." The author concludes "Our experiment confirms our belief that a computer can be a worthwhile tool in selecting stocks."

Essentially this is a standard retrieval from a large database—all stocks in U.S.A.

Fuzzy logic approach

The fuzzy logic methodology can produce better results. Each requirement stated by Mandelman (1979) has to be characterized by the linguistic variables: *debt, return, dividend yield, PE ratio*, and *stock price*. *Low, very low*, and *high* are terms of appropriate linguistic variables. The financial experts should be able to describe the above variables (see Chapter 5, Section 5.2) and initiate a fuzzy complex query using computers:

```
SELECT NAME
FROM STOCKS
WHERE DEBT IS LOW
        AND RETURN IS HIGH
        AND DIVIDEND YIELDS IS HIGH
        AND PE RATIO IS VERY LOW
        AND STOCK PRICE IS LOW
```

There are financial institutions in various countries using fuzzy logic for portfalio management, but it is very difficult to obtain information about their activities.[3] In a short note, Schwartz (1990) reports: "Fuzzy information processing takes place every day at Yamaichi Securities, the first securities-trading company to offer a fund with purchases based on fuzzy-system decisions. Currently, the system monitors over 1100 stocks, but makes only a few trades each day. Employing fuzzy reasoning, expert system technology, and conventional number crunching, the system is tuned daily by Yamaichi trading experts. The fund has been operating for approximately nine months and claims to be sporting a 40-percent annual return for investors.[4]"

We illustrate the fuzzy logic approach on a small database containing funds.

Case Study 27 *Fuzzy Query from the 20 Biggest Mutual Funds in Canada*

Consider the database presented in Table 7.12.

Table 7.12. The 20 biggest mutual funds in Canada ranked by total assets at 31 Dec. 1995; in billions of dollors.

FN	TOTAL ASSET		CH %	RETURN %		
	31/12/95	31/03/94		1 Y	3 Y	5 Y
F_1	4.08	2.31	76.6	14.1	17.3	19.4
F_2	3.19	1.57	103.2	14.2	18.2	21.7
F_3	3.03	3.59	−15.6	11.5	6.6	8.0
F_4	2.61	1.86	40.3	18.8	9.8	10.8
F_5	2.45	2.58	−5.3	10.3	8.3	9.1
F_6	2.44	1.81	34.8	9.9	14.3	13.6
F_7	2.36	2.43	−3.0	6.3	5.2	6.4
F_8	2.13	0.64	232.8	11.7	14.6	n/a
F_9	2.10	1.31	60.3	10.6	13.3	12.2
F_{10}	2.04	2.79	−26.9	12.9	7.8	9.8
F_{11}	2.00	1.70	17.6	14.8	19.6	24.6
F_{12}	1.98	1.60	23.8	11.9	12.9	9.6
F_{13}	1.94	2.03	−4.4	6.1	4.9	n/a
F_{14}	1.92	2.22	−13.5	14.3	11.0	11.3
F_{15}	1.88	1.46	28.8	15.3	18.2	17.6
F_{16}	1.81	1.16	56.0	16.7	20.8	23.9
F_{17}	1.79	0.97	84.5	15.0	14.1	13.4
F_{18}	1.64	1.72	−4.7	19.3	9.2	10.8
F_{19}	1.59	1.68	−5.4	19.9	23.0	n/a
F_{20}	1.44	1.20	20.0	10.7	15.9	15.8

We use the abrivations: FN=FUND NAME, CH=CHANGE, 1 Y=1 YEAR, 3 Y=3 YEAR, and 5 Y=5 YEAR. Table 7.12 is taken from "The Mutual Fund Advisory" written and edited by C. Tidd (February 1996). We do not give the real names of the funds; here they are labeled $F_i, i = 1, \ldots, 20$.

The author reminds "that the single purpose of this particular exercise is to determine shifts into (and out of) the country's 20 largest Mutual Funds" and also makes a short analysis based on the data covering 21 months (31 March 1994 to 31 December 1995).

Our aim is to use the real data in Table 7.12 for making fuzzy queries.

We consider *change* and *return* as linguistic variables. They are partitioned into terms (linguistic values) presented in Fig. 7.9 (*change*) and Fig. 7.10 (one-, two-, and three-year *return*).

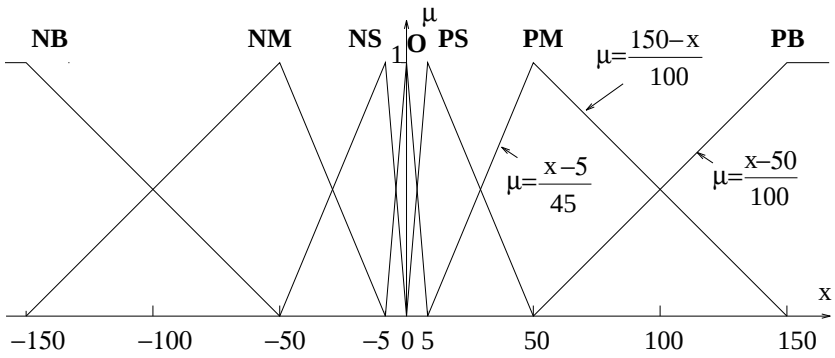

Fig. 7.9. Terms of *change* for the 20 biggest mutual funds in Canada.

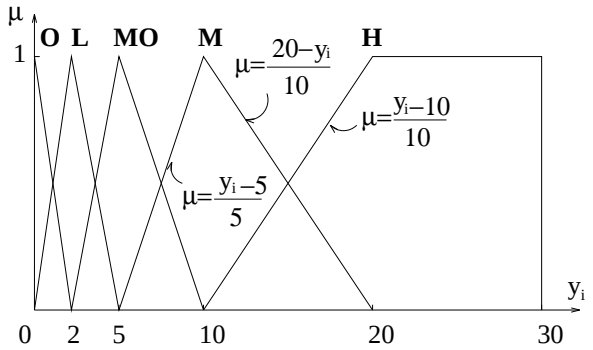

Fig. 7.10. Terms of *one-,three-,five-year return* for the 20 biggest mutual funds in Canada; $y_i = 1, 3, 5$.

The terms of *change* are defined as follows: **NB** $\stackrel{\triangle}{=}$ *negative big*, **NM** $\stackrel{\triangle}{=}$ *negative medium*, **NS** $\stackrel{\triangle}{=}$ *negative small*, **O** $\stackrel{\triangle}{=}$ *zero*, **PS** $\stackrel{\triangle}{=}$ *positive small*, **PM** $\stackrel{\triangle}{=}$ *positive medium*, **PB** $\stackrel{\triangle}{=}$ *positive big*. The base variable x is measured in percentage.

The terms of *return* (1, 3, and 5 year) are defined by **O** $\stackrel{\triangle}{=}$ *zero*, **L** $\stackrel{\triangle}{=}$ *low*, **MO** $\stackrel{\triangle}{=}$ *moderate*, **M** $\stackrel{\triangle}{=}$ *medium*, **H** $\stackrel{\triangle}{=}$ *high*. The base

variable $y_i, i = 1, 3, 5$, is expressed in percentage; y_i is positive since the return for all funds (Table 7.12) is gain. In situations with negative return (loss) Fig. 7.10 has to be extended to the left symmetrically about the μ-axis.

Now we consider three queries.

Query 1

SELECT FUND
FROM TABLE 7.12
WHERE CHANGE IS POSSITIVE BIG
AND 1 YEAR RETRUN IS HIGH
AND 3 YEAR RETRUN IS HIGH
AND 5 YEAR RETRUN IS HIGH

The aim of this query is to identify funds picking up huge amount of money (meaning more business) while producing consistently high returns.

Following the procedure for calculating the membership values in this chapter we obtain the results in Table 7.13. (second to fifth columns), where CHPB= CHANGE POSITIVE BIG and 1,3,5 YH = 1, 3, 5 YEAR HIGH. We present the calculations only for fund F_1. Substituting 76.6 from Table 7.12 for x into equation $\mu = \frac{x-50}{100}$ (see Fig. 7.9) gives 0.27. Substituting 14.1 for y_1, 17.3 for y_3, and 19.4 for y_5 from the same table correspondingly into equation $\mu = \frac{y_i-10}{10}$, $i = 1, 3, 5$, gives 0.41, 0.73, and 0.94.

The aggregation by *and* is given in the sixth column labeled AND and that by *averaging* in the seventh column labeled AVE. For the fund F_1 aggregation by *and* gives $\min(0.27, 0.41, 0.73, 0.94) = 0.27$ and aggregation by averaging produces $\frac{0.27+0.41+0.73+0.94}{4} = 0.59$. For the fund F_8 5 year return is not available (n/a); the fund is younger than 5 years. The aggragation for F_8 is based on the presented data, i.e. for operation *and*, $\min(1, 0.17, 0.46) = 0.17$, for average, $\frac{1+0.17+0.43}{3} = 0.54$.

We can use the membership values in the conclusions AND and AVE in Table 7.13 to rank the funds which satisfy the query. Also we can use a threshold value $\alpha = 0.2$, which means that the funds with membership values below 0.2 are to be dropped. The results are presented in Table 7.14.

Table 7.13. Membership grades for Query 1 from 20 biggest mutual funds in Canada (31 March 1994 to 31 December 1995).

FN	CHPB	1 YH	3 YH	5 YH	AND	AVE
F_1	0.27	0.41	0.73	0.94	0.27	0.59
F_2	0.53	0..42	0.82	1.00	0.42	0.69
F_3	0	0.15	0	0	0	0.04
F_4	0	0.88	0	0.08	0	0.24
F_5	0	0.03	0	0	0	0.01
F_6	0	0	0.43	0.36	0	0.20
F_7	0	0	0	0	0	0
F_8	1	0.17	0.46	n/a	0.17	0.54
F_9	0.10	0.06	0.33	0.22	0.06	0.18
F_{10}	0	0.29	0	0	0	0.07
F_{11}	0	0.48	0.96	1.00	0	0.61
F_{12}	0	0.19	0.29	0	0	0.12
F_{13}	0	0	0	n/a	0	0
F_{14}	0	0.43	0.10	0.13	0	0.17
F_{15}	0	0.53	0.82	0.76	0	0.53
F_{16}	0.04	0.67	1.00	1.00	0.04	0.68
F_{17}	0.35	0.50	0.41	0.34	0.34	0.40
F_{18}	0	0.93	0	0.08	0	0.25
F_{19}	0	0.99	1.00	n/a	0	0.66
F_{20}	0	0.07	0.59	0.58	0	0.31

If a threshold value $\alpha = 0.1$ is addopted, then more funds have to be included in the ranked tables (Table 7.14) as follows. The fund F_8 goes to the first table (AND) and the funds F_9 and F_{14} join the second table (AVE).

Both aggragation procedures, *and* and *average*, rank fund F_2 at first place but after that there is considerable difference. It was already indicated that *and* procedure is quite conservative (Section 7.3). In this case it emphasizes too much the linguistic variable *change*: namely funds whose *positive change* is below 50% do not qualify. On the other hand side, fund F_8 with the biggest increase of 232.8% is not included for ranking since one-year return of 11.7% has a low membership value 0.17. The fund managers may decide to tune the model representation

of the linguistic variables *change* and *return* (see Section 5.8) shifting to the left the lower boundaries 50 of **PB** and 10 of **H**. Actually for Query 1 only the terms **PB** (Fig. 7.9) and **H** (Fig. 7.10) are needed. Having the other terms allows the making of various queries.

Table 7.14. Ranking the biggest mutual funds in Canada produced by Query 1.

RANK	FN	AND
1	F_2	0.42
2	F_{17}	0.34
3	F_1	0.27

RANK	FN	AVE
1	F_2	0.69
2	F_{16}	0.68
3	F_{19}	0.66
4	F_{11}	0.61
5	F_1	0.59
6	F_8	0.54
7	F_{15}	0.53
8	F_{17}	0.40
9	F_{20}	0.31
10	F_{18}	0.25
11	F_3	0.24
12	F_6	0.20

Query 2

SELECT FUND
FROM TABLE 7.13
WHERE CHANGE IS POSITIVE MEDIUM
AND 1 YEAR RETURN HIGH
AND 3 YEAR RETURN IS HIGH
AND 5 YEAR RETURN IS MEDIUM

This query is focused on funds which are expanding their business and producing high returns in the last three years thus improving their performance.

The final results are presented in Table 7.15 where CHPM=CHANGE POSITIVE MEDIUM and 5YM=5 YEAR MEDIUM. The attributes 1 YH and 3 YH have the same domain as those in Table 7.13.

Table 7.15. Membership grades for Query 2 from 20 biggest mutual funds in Canada (31 March 1994 to 31 December 1995).

FN	CH PM	1 YH	3 YH	5 YM	AND	AVE
F1	0.73	0.41	0.73	0.06	0.06	0.48
F2	0.47	0.42	0.82	0	0	0.43
F3	0	0.15	0	0.60	0	0.19
F4	0.64	0.88	0	0.92	0	0.61
F5	0	0.33	0	0.82	0	0.29
F6	0.54	0	0.43	0.57	0	0.39
F7	0	0	0	0.14	0	0.04
F8	0	0.17	0.46	n/a	0	0.21
F9	0.90	0.06	0.33	0.78	0.06	0.52
F10	0	0.29	0	0.96	0	0.31
F11	0.23	0.48	0.96	0	0	0.42
F12	0.34	0.19	0.29	0.92	0.19	0.44
F13	0	0	0	n/a	0	0
F14	0	0.43	0.10	0.87	0	0.35
F15	0.43	0.53	0.82	0.24	0.24	0.51
F16	0.94	0.67	1.00	0	0	0.65
F17	0.66	0.50	0.41	0.66	0.41	0.56
F18	0	0.93	0	0.92	0	0.46
F19	0	0.99	1.00	n/a	0	0.66
F20	0.27	0.07	0.59	0.42	0.07	0.34

Query 3

SELECT FUND
FROM TABLE 7.13
WHERE CHANGE IS NEGATIVE SMALL
AND 1 YEAR RETURN IS MODERATE
AND 3 YEAR RETRUN IS MODERATE
OR LOW

The query wants to depict funds that are lossing business (the worst case is -26.9%) and also having an unimpressive return during the last three years in comparison to their competitors. In the one-year performance there is no fund with low return while in the three-year there is

one such fund. This explains the introduction of *or* connective into the WHERE predicate concerning the attribute 3 YEAR in Table 7.12.

The calculations are similar to those in the previous queries discussed in this chapter. We have to construct a table similar to Table 7.13 and 7.15 having top row

FN	CNNS	1YMO	3YMO	3YL	AND/OR

where CNNS=CHANGE NEGATIVE SMALL, 1YMO=1 YEAR MODERATE, 3YMO=3 YEAR MODERATE, and 3YL=3 YEAR LOW.

The membership grades for each tuple can be calculated according to the formula

$$\text{CNNS } \textit{and } \text{1YMO } \textit{and } (\text{3YMO } \textit{or } \text{3YL})$$

which can be expressed by min and max in the form

$$\min(\text{CNNS, 1YMO}, \max(\text{3YMO, 3YL})).$$

Here CNNS, 1YMO, 3YMO, and 3YL have to be substituted by the appropriate entries in the tuples. Note that here the connective *or* (max) appears in a different place than *or* (max) in Case Study 26, Query 4.

□

7.6 Notes

1. Research on database began with a paper on a relational data model by Codd (1960), a researcher at the IBM Santa Terresa in San Jose, California.

2. According to Terano, Asai, and Sugeno (1987), the term fuzzy database was first used by Kunii (1976). Fuzzy databases are briefly considered by Klir and Folger (1988).

3. Graham and Jones (1988) made the comment "One major difficulty in surveying financial applications is the secrecy and even paranoia which surrounds successful ones. Because one of their

chief benefits is the competitive edge they provide this is hardly surprising, but as with the defence sector a certain amount of knowledge is in the public domain. Although this is manifest it is also possible that some of the secrecy could have arisen from the vested interests of the developers, who are concerned not to expose their infant and struggling applications to the glare of publicity until they are proved to be robust."

4. Management Intelligenter Technologien GmbH, Promenade 9, 52076 Aachen, Germany, advertises a software tool based on fuzzy logic and neural networks for analyzing complex tasks that was successfully used for the forecasting of the Standard & Poor's 500 Index.