CIENTIFIC
Publishing Limited

Digital Technologies Research and Applications
http://ojs.ukscip.com/index.php/dtra

Article

For Fuzzy Classification of Databases with Fuzzy Classification Query Language

Seyfali Mahini *

Islamic Azad university, khoy Branch,khoy, Iran
* Correspondence: my1341post@yahoo.com

Received: 27 February 2022; **Accepted**: 13 April 2022; **Published**: 29 April 2022

Abstract: Business information systems have extensive databases that are mainly managed in relational databases. what is often missing are automated procedures to analyze these inventories without major restructuring. Based on this, we develop the Fuzzy classification Query Language, FcQL, which enables fuzzy queries to the extended database schema using linguistic variables and converts them into sQL statements to the database. with this, we give the user a data mining tool so that he can start extended queries on his databases based on a pre-defined fuzzy classification and obtain an improved basis for decision making. As a result, the fuzzy classification query language enables marketers to improve customer value, launch useful programs, automate overall customization, and refine business campaigns.

Keywords: fuzzy classification; Information systems; relational database; query language; data mining; customer relationship management

1. Introduction

On the way to the information society, a lack of data has turned into an overabundance (information overload). Therefore, companies and organizations are interested in tools for data analysis in order to continue to have a basis for business decisions. of particular interest is the process of knowledge Discovery in Databases (kDD), which extracts valuable information from sometimes extensive databases. The primary goal of a kDD process is to reduce the complexity of the data or recognizing patterns in large databases. classic methods such as cluster analysis or regression analysis are mostly based on statistical methods.

They assume that the amounts of data or databases contain numerical values or contain sharp data values. As soon as the data itself or the classes that are defined in the data analysis and to which the data is then assigned to reduce complexity are no longer clearly defined, many conventional data analysis methods fail.

2. Motivation

In companies, databases contain non-numeric information in addition to numeric data. Database query languages require queries to be formulated with the same level of detail and precision as the data is stored in the database.

Relational query languages such as SQL do not allow for imprecisely formulated or fuzzy queries. Figure 1 illustrates a fuzzy query using the vague term “unacceptable” (see Table 1 for a table example, see Figure 6 for a definition of the term). This small illustrative example illustrates the need to be able to operate on large databases with vague and imprecise queries. There are also many practical examples from day-to-day business for fuzzy classification:

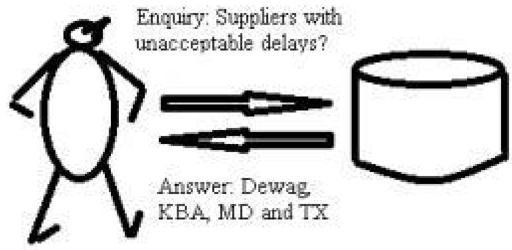

Figure 1. Fuzzy query of a relational database.

2.1. Customer Relationship Management

Recently, the management of customer relationships and processes has gained in importance. In addition, the customers are happy to be based on certain characteristics resp. of purchasing behavior divided into classes or customer segments. A common division is into A, B and c customers, depending on the customer,s purchasing pow- er. In most cases, this class affiliation is strictly defined, i.e., each customer belongs to exactly one class. If the development potential is to betaken into account in addition to the completed transactions, an individual customer can no longer be clearly assigned to a customer segment. Analyzing the possibly extensive customer base and creating fuzzy customer segments are a must. only in this way can marketing, sales and after-sales be appropriate.

2.2. Checking Creditworthiness and Risk

Insurance companies and banks divide their customers into various classes based on risk considerations. Age, purchasing power and other characteristics must be checked for a loan application. Instead of sharp creditworthiness or risk classes, fuzzy classes that work with the help of a membership function can be interesting. Based on research, it has been found that traditional credit checks and clearly different risks can result in the same overall rating for the customer. conversely, it is also possible that different overall ratings can arise if the customer,s characteristics are very similar.

2.3. Selection of Suppliers

Evaluations must be carried out for the selection of suppliers. conventional or sharp classifications of suppliers, e.g., according to delivery date and quality characteristics, are no guarantee that a promising and long-term business relationship can be maintained. The structure contained in the stored supplier data, but not visible or not yet visible, is made visible by fuzzy class formation. It allows a targeted and effective treatment of the individual supplier classes.

2.4. Analyzing Data

Business information systems generate a wealth of data. In day-to-day business, but above all to secure decisions, it is necessary to analyze these sometimes extensive data stocks or databases. without a proper KDD process with fuzzy data analysis, you risk missing out on valuable information- both risks and opportunities.

3. Databases and Fuzzy Logic

In databases, especially in relational database systems, the characteristic values are assumed to be unique and queries to the database produce clear results. Relational databases can be characterized as follows:

- The feature values in the databases are precise, i.e., they are unique. Already in the requirement of the first normal form we require that the characteristic values are atomic and come from a well-defined range of values. Imprecise characteristic values such as a supplier,s delay is “2 or 3 or 4 days” or vague characteristic values such as the delay is “approximately 3 days” are not permitted.
- The characteristic values stored in a relational database are secure, i.e., the individual values are either known or unknown. An exception is the zero values,i.e., characteristic values that are not or not yet known. In addition, the database systems do not support us in any way in modeling an existing stochastic uncertainty. In other words, probability distributions for feature values are excluded; it remains difficult to express whether a given feature value corresponds to the true value or not.
- Queries to the database are sharp. They always have a dichotomous character, i.e., a query value given in the query must either match or not match the characteristic values in the database. An evaluation of the database in which a query value “more or less” matches the stored characteristic values is not permitted.

For several years, findings from the field of fuzzy logic have been applied to data modeling and databases, see Bordogna and pasi [1], Bosc and Kacprzyk [2], chen [3], petry [4], pons et al. [5]. Most of this work is theoretical in nature; however, some research groups have attempted to demonstrate the usefulness of fuzzy database models and database systems with implementations.

With data modeling, a larger field of application can be opened up if incomplete, vague or imprecise facts are allowed. with the help of fuzzy logic, various model extensions were proposed, both for the entity-relationship model and for the relational model. For example, in his dissertation, Chen expanded the classical normal forms of database theory into fuzzy ones by allowing fuzziness in the functional dependencies [6,7]. Many different proposals for fuzzy data models for databases can be found [8].

Investigations have also been made for the extension of relational query languages with fuzzy logic. For example, Takahashi [9] proposes a fuzzy query language (FQL) based on the relational domain calculus [10]. The language FQUERY by kacprzyk and zadrozny [11] uses fuzzy terms and has been implemented as a prototype in the Microsoft product Access.

In our work on a fuzzy classification and a fuzzy classi- fication query language FCQL, we choose a slightly different research direction, originally indicated by Schindler [12]: we limit ourselves to an extension of the relational database schema by creating a context model for propose fuzzy classification of table contents.

Based on this, we develop the language FCQL, which allows fuzzy queries to the database schema using pre-defined linguistic variables and transmits them to the underlying (sharp) database in SQL calls. In this way, we avoid migrating the database to a fuzzy database at great expense [13] or confronting the user with fuzzy SQL [14]. Fuzzy predicates would lead to a variety of semantic effects and a user would have to make different interpretations. with FCQL, on the other hand, we give the user a data mining tool so that he can start extended queries and calculate improved decision-making bases based on a pre-defined fuzzy classification of his data stocks.

3. Context Model and Classification

Extensive databases are often confusing and therefore difficult to analyze and evaluate. In order to obtain meaningful information, the user must restructure and, if necessary, condense their stocks. To this end, various methods and concepts for building and operating a data warehouse have been developed. There are also data mining tools to gain new insights from the databases.

We choose a context model approach to be able to specify classes in the relational database schema. For the analysis and evaluation of many suppliers, for example, it makes sense to group suppliers that are as similar as possible into classes. You then get the set of "suppliers with quality problems" or the set of "suppliers with whom the business relationship should be expanded" as an example. Such a combination of suppliers into classes means a reduction in complexity. The user can thus maintain and analyze his supplier relationships more clearly, thanks to the reduced flood of data.

In addition, important characteristics of the suppliers are made visible through the classification. This additional knowledge allows the user to analyze entire classes in a targeted and holistic manner and to work out the relationships between different classes.

when classifying objects in a relational database, a distinction can be made between sharp and fuzzy methods. In the case of a strict classification, the database objects are assigned to the class in a dichotomous manner, i.e. the set membership function of the object to the class is 0 for not included or 1 for included. A classic procedure would therefore assign a supplier to the "suppliers with quality problems" class or the "suppliers with whom the business relationship should be expanded" class. A fuzzy procedure, on the other hand, allows values between 0 and 1 for the quantity membership function: For example, a supplier may belong to the "suppliers with quality problems" class with a value of 0.3 and at the sametime belong to the "suppliers with whom the business relationship was established" class with a value of 0.7. A fuzzy classification therefore enables a differentiated interpretation of the class affiliation; with database objects of a class, one can distinguish between peripheral and core objects, and database objects can also belong to two different classes at the sametime.

In the fuzzy-relational data model with contexts - in short, in the context model - a context is assigned to each attribute A_j, defined on a value range $D(A_j)$. A context $C(A_j)$ is apartition of $D(A_j)$ into equivalence classes. Are-lational database schema with contexts therefore consists of a set of attributes $A=(A_1,...,A_n)$ and a set of associated contexts $C=(C_1(A_1),...,Cn(A_n))$ [15].

Material-specific data on the delivered quality and delay should be recorded for a supplier evaluation. The corresponding database schema SE(A, C) for the supplier evaluation is in the attributes.

A = (supplier, material, quality, delay)

and the contexts

C = (C(supplier), C(material), C(quality), C(delay))

Specified. The material quality is described with the terms D(quality) = {high,medium, sufficient, low}. The delay is the delay that has occurred so far compared to the promised delivery date in days. The following partitions apply to the contexts:

C(supplier) = {{supplier names}}

C(material) = {{material identification numbers}}

C(quality) = {{high, medium},{sufficient, low}}

C(delay) = {{1, 2, 3, 4, 5},{6, 7, 8, 9, 10}}

The contexts C(supplier) and C(material) consist of the broadest equivalence classes that correspond to the respective value ranges. According to C(quality), it is assumed for the evaluation of a query, for example, that the quality levels "sufficient" and "low" are equivalent. In addition, C (delay) means that the distinction between a delay of e.g. one day and five days is irrelevant for queries. In Figure 2, we show the classification space for our simple example of supplier evaluation. By partitioning the value ranges of quality and delay, we get the four equivalence classes C1, C2, C3 and C4. The meaning of the content of the classes is made visible by semantic class names; e.g. for class C4, the name "Check relationship" is chosen. An orientation to the equivalence classes facilitates a meaningful interpretation.

It is one of the tasks of the database administrator to define suitable equivalence classes in cooperation with the relevant specialists.

In a classification, the selected features must be independent. To do this, it may be necessary to analyze the data and calculate the correlation coefficients between the characteristics. Another clue to dependent characteristics comes from uncovering transitive dependencies in database design.

D(delay)

<table>
<tr><td>10</td><td colspan="2" rowspan="5">C2
delay
admonish</td><td colspan="2" rowspan="5">C4
check
relationship</td></tr>
<tr><td>9</td></tr>
<tr><td>8</td></tr>
<tr><td>7</td></tr>
<tr><td>6</td></tr>
<tr><td>5</td><td colspan="2" rowspan="5">C1
expand
relationship</td><td colspan="2" rowspan="5">C3
discuss
quality</td></tr>
<tr><td>4</td></tr>
<tr><td>3</td></tr>
<tr><td>2</td></tr>
<tr><td>1</td></tr>
<tr><td></td><td>high</td><td>medium</td><td>sufficient</td><td>low</td></tr>
</table>

D(quality)

Figure 2. Classification space spanned by the characteristics of quality and delay.

5. Context Redundant Tuples

In the relational model, one speaks of redundant tuples or tables if there are multiple occurrences in the tuple components that could be deleted without loss of information. The theory of normal forms was developed for the preservation of redundancy-free relations [10,16]. The redundancy of two tuples in the context model is softer defined: Two tuplest and t, are called context-redundant if all tuple components ti and t,i belong to the same equivalence class. Context-redundancy-free relations are obtained by mixing operations, which are discussed in more detail below.

A database object belongs to a class if its feature vector points to the corresponding sub-domain. In the presented context model, the set-theoretic unification as a mixing operation is chosen as the classification function. This operation is performed when evaluating an expression of context-based relational algebra in order to obtain context redundancy free result relations [17,12].

Let,s look at an example: The suppliers shown in Figure 3 are to be evaluated for material 802.025. objects are the individual vendors identified by the vendor attribute of the composite primary key. The characteristics relevant for an evaluation are quality and delay. A classification of the suppliers for material 802.025 provides the following combination of context-based projection II and selection Σ. The operators of the relational context-based algebra are expressed using Greek capital letters, based on the work of schenoi [17].

II [supplier, Quality, Delay, C()] (Σ[Material$\sim_{C(Material)}$) 802.025](Table 1))

Supplier	Material	Quality	Schedule Delay
BAW	802.025	sufficient	8
DEWAG	802.025	medium	5
DEWAG	809.200	high	8
KBA	802.025	sufficient	7
KBA	809.025	sufficient	3
MD	840.024	low	9
MD	802.025	sufficient	8
MTX	809.200	medium	2
MTX	809.200	high	4
MAM	802.025	high	7
MAM	840.024	low	6
ZT	802.025	high	8
ZT	840.024	medium	2

Table 1. Supplier evaluation.

For the evaluation of the query, the context C(Material) is replaced by the precise context PC(Material)= {{802.025},{809.200},...}. The result of the query is the context-redundancy-free, imprecise relation Table 2, which results in a clear assignment of the objects supplier to the classes C1 and C4 in the column of the object-identifying characteristic supplier.

Supplier	Quality	Delay
MTX(Dewag)	{high, medium}	{2,5}
{KBA, MAM, BAW, MD}		{7,8}

Table 2. Clear allocation of suppliers to classes C1 and C4.

Class: C1: expand relationship , C4: check relationship

A material-independent evaluation of the suppliers shows that the evaluation of a query with the context model does not always produce results in the sense of a sharp classifier. For this we consider the following context-based projection of the relation Table 2:

II[supplier, Quality, Delay, C()](Table 2)

The result in Table 3 illustrates that only the suppliers BAW and MTX were sharply classified. On the other hand, for example, the vendor MD was assigned to class C1 and C4. This means that the merge operation for the supplier MD currently still delivers a contradictory recommendation, namely “expand relationship” (C1) and “check relationship” (C4). The evaluation of a query in the context model does not generally assign a database object to one and only one class.

The multiple assignment of database objects to classes can lead to conflicting recommendations for action. In order to reduce this uncertainty, the context model is further developed into a fuzzy classification using elements from the theory of fuzzy sets.

Supplier	Quality	Delay
{MTX, ZT, MD, Dewag}	{high, medium,}	{2, 4, 5}
{MAM, Dewag,ZT}	{high, medium}	{6, 8}
{KBA}	{sufficient}	{3}
{KBA, MAM, BAW, MD}	{sufficient, low}	{7, 8, 9}

Table 3. Fuzzy assignment of suppliers to classes (except BAW and MTX).

C1: expand relationship, C2: request delay

C3: discuss quality, C4: check relationship

basic variable is equal to the domain of values D(delay). The linguistic variable has the term set T(delay)={acceptable, unacceptable} with the verbal terms "acceptable" and "unacceptable" to describe the two equivalence classes {1, 2, 3, 4, 5} resp. {6, 7, 8, 9, 10}.

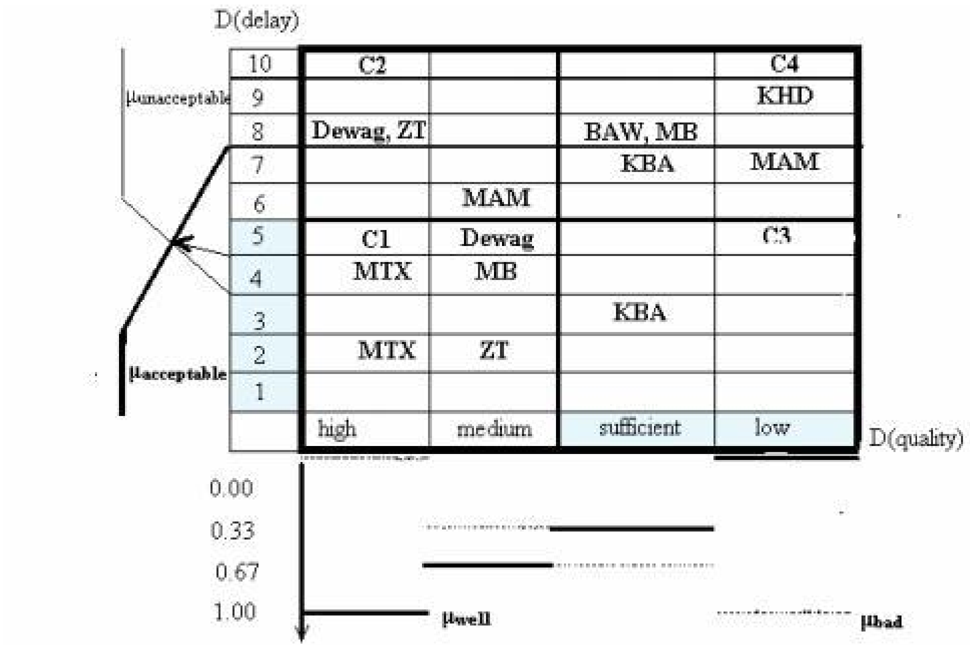

Figure 4. Fuzzy partitioning of the value ranges with membership functions.

6. Fuzzy Classification with Linguistic Variables

Classes represent different states for a specific database object. In the case of multiple assignments of an object to different classes, it is not possible to make a clear distinction between the different states. In order to be able to derive fuzzy classes from the sharp contexts considered so far, verbal terms are first assigned to the equivalence classes. An idea of the elements belonging to the equivalence class with a fuzzy assignment is then connected with these concepts. A formal description of the assignment of verbal terms to equivalence classes is possible with the concept of linguistic variables [18]. Linguistic variables do not have numbers or distributions as values, but rather linguistic constructs or terms (verbal terms). The content of these terms is defined by fuzzy sets on a so-called basis variable. As an example, Figure 3 shows the definition of the characteristic delay as a linguistic variable based on the basic variable "cumulative delay". The domain of definition of the

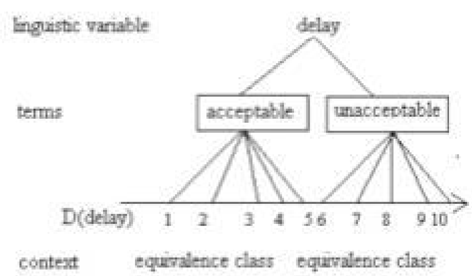

Figure 3. Assignment of verbal terms to equivalence classes.

The semantic description of fuzzy classes with linguistic variables makes it possible to accommodate human understanding. Each term is represented by a fuzzy set defined by membership functions on the domains of the attributes. The vague terms describing the equivalence classes are the labels of the fuzzy sets. In Figure 4, a lateness is simultaneously "acceptable" and "unacceptable" to the degree of 0.5, i.e. the belonging of

5 days to the sharp quantities $\mu_{acceptable}$ and $\mu_{unacceptable}$ is 0.5.

A schedule delay of 5 days is therefore not exclusively acceptable or unacceptable as with sharp classes.

The description of the vague terms "acceptable" and "unacceptable" with the membership functions $\mu_{acceptable}$ and $\mu_{unacceptable}$ causes the value range D(delay) to be partitioned indistinctly. similarly, the range of values D(quality) is divided by the terms "good" and "bad". This results in classes with continuous transitions,i.e. fuzzy classes, in the context model.

The class affiliation of an object results from the affiliation of the tuple components (characteristic values) to the fuzzy sets whose labels describe the class. The object affiliation of MTX, e.g. with the feature vector (4, high) related to the term "acceptable", of the feature delay cor- responds to the affiliation value.

M(MTX|acceptable) = $\mu_{acceptable}(4)$.

The affiliation M(oi|Ck) of an object oi to the class Ck results from the aggregation over all terms of the linguistic variables that define the class. Class C1, for example,is described by the terms "acceptable" and "good" . The aggregation must therefore correspond to a conjunction of the individual membership values. various operators have been developed for this in the theory of fuzzy sets [18]. A general class of operators are the triangular operators or t-norms with the minimum operator as an example. The membership of MTX(4,high) in the class C1 is with this operator.

M(MTX(4, high)|C1) = min$\{\mu_{acceptable}(4), |\mu_{well}(high)\}$.

The use of the minimum operator as an aggregation operator means that the characteristic with the lowest membership value is decisive for determining whether an object belongs to a class. When classifying suppliers, a supplier is acceptable in terms of its delay in delivery, but delivers poor quality. An aggregation with the minimum operator would only classify the supplier as a "poor" supplier because of its quality. If, on the other hand, the supplier evaluation takes place through human consideration, then one would make a certain compensation between poor quality and acceptable delay. In other words, in many cases, poor quality will not be the only criterion when ranking a supplier.

Human decision-making behavior is often characterized by compensatory considerations. For this purpose, special operators such as the compensatory and or the so-called Y-operator were developed in fuzzy logic [3,18]. The fuzzy result relation SE from Figure 8, in which the tuple components are fuzzy sets, shows a classification result with an aggregation via the Y-operator.

Supplier	Qualiy	Delay
{(MTX,1.00), (ZT, 0.45), (MD, 0.45), (Dewag, 0.35)}	well	acceptable
{(MAM, 0.40), (Dewag, 0.65), (ZT, 0.55)}	well	unacceptable
{(KBA, 0.31)}	bad	acceptable
{(KBA, 0.69), (MAM, 0.60), (BAW, 1.00), (MD, 0.55)}	bad	unacceptable

Table 4. Fuzzy classes by compensatory aggregation with the Y-operator.

C1: expand relationship, C2: to remind you of a day,

C3: discuss quality, C4: check relationship

The fuzzy quantities in the Supplier column show the fuzzy decomposition. For example, the supplier Dewag has an affiliation of 0.35 to the class "expand relationship" (C1) and 0.65 to the class "admonishment of delay" (C2). In the sharp classification of Table 3, on the other hand, Dewag was assigned to the classes "expand relationship" and "remind delay" with the affiliation 1.0. A comparison with the result from Table 4 illustrates that the supplier Dewag is more concerned with scheduling problems and not quality issues. The uncertainty regarding the maintenance of the relationship with the supplier is thus clarified.

7. Language FCQL for Fuzzy Classification

In our work, we refrain from developing fuzzy query languages on the basis of a context-based relational algebra or a fuzzy domain calculus, as variously proposed and partially implemented [11,15,19]. our language approach is limited to a classification language called FCQL, which can be used to define and query fuzzy classes. Specifically, we use SQL on the underlying sharp databases to produce a fuzzy result relation. To do this, it is necessary to use SQL statements to select and aggregate context-redundant tuples from the sharp database.

In Table 5, we discuss three different approaches to the contextual model query languages. The SQL language can be characterized as a sharp query language with precise contexts and single-element equivalence classes.

In their work, Finnerty and Shenoi [14] designed the query language MIQUEL in order to be able to direct fuzzy queries to a relational database. A MIQUEL query differs from an ordinary SQL query in the where clause select features ʃrom tables where selection condition with context designation:

According to Finnerty and Shenoi, the simplest form of a contextual condition is an expression of the form.

<attribute> <context> <reloperator> <expression>

E.g. in Figure 9 the condition "delay approximately = 4". In other words, with the MIQUEL language suggestion, the query condition always consists of a clear target value and optionally, a context. The MIQUEL language thus permits sharp and fuzzy queries at the same time.

Query Language	Example	Query Type in the Context Model
SQL	select supplier from SE where delay = 4	sharp query through precise contexts with single element equivalence class
MIQUEL	select supplier from SE where delay approx=4	fuzzy query through sharp contexts with multi-element equivalence class
FCQL	classify supplier from SE classify supplier from SE with class is appointment problem classify supplier from SE with delay is acceptable and the quality is good	fuzzy classification through vague contexts with fuzzy equivalence classes

Table 5. Overview of context-based language approaches for relational databases.

MIQUEEL

select features

ʃrom tables

where selection condition with context designation

In contrast to MIQUEL, classification queries with the language FCQL [20] operate on the linguistic level with vague contexts. This has the advantage that the user does not need to know a specific target value or context, just the column name of the object-identifying feature and the table or view that contains the feature values. For a targeted view of individual classes, the user can specify a class or specify features with a verbal description of their characteristics. Classification queries therefore work with verbal descriptions at the feature and class level:

classiʃy object

ʃrom table

with classification condition

The FCQL language is based on SQL, with the projection list in the select clause being replaced by the column name of the object to be classified. while the where clause in SQL contains a selection condition, we use the with clause to describe the classification condition.

As an example of an FCQL query, "classity supplier ʃrom SE" generates a classification of all suppliers in table SE (from Figure 3). with "classiʃy supplierʃrom SE with class is appointment problem" a specific class is queried. If one dispenses with the definition of the class, one can select a specific set of objects with the linguistic descriptions of the equivalence classes. As an example, consider the query "classiʃy supplierʃrom SE with late delivery is acceptable and quality is good". This query consists of the identifier of the object to be classified (supplier),the name of the basic table (SE), the critical characteristic names (delay or quality) and the names of the pre-defined equivalence classes (acceptable or good) .

8. Outlook

For code generation, we had to extend the database schema with three descriptive tables for contexts, classes and membership functions. we have tested our implementation on an industry database. The responsible marketing department has a mature data warehouse, but they want to further analyze and evaluate the customer and product inventories using methods of fuzzy logic. Thanks to the chosen approach with fuzzy classes, we didn't have to carry out a complex database migration to take over the data stock. During the field test, we observed that the modified classification language FCQL makes a useful addition to the toolset for data mining and knowledge discovery.

Conflicts of Interest

There is no conflict of interest.

References

1. Bordogna, G., Paso, G., 2019 (Eds.). Recent Issues on Fuzzy Databases. physical publishing house.
2. Bosc, p., Kacprzyk, J., 2018 (Eds.). Fuzzyness in Database Management systems. physical publishing house.
3. Chen, G., 2020. Fuzzy Logic in Data Modeling - semantics, constraints, and Database Design. kluwer Academic publishers.
4. Petry, F.E., 2019. Fuzzy Databases - principles and Applications. kluwer Academic publishers.
5. Pons, O., Vila, M.A., kacprzyk, J., 2020 (Eds.). knowledge Management in Fuzzy Databases. physical.
6. Chen, G., 2018. Design of Fuzzy Relational Data- bases Based on Fuzzy Functional Dependency. phD Dissertation No. 94, Leuven Belgium.
7. Shenoi, s., Melton, A., Fan, L.T., 2019. Functional Dependencies and Normal Forms in the Fuzzy Relational Database Model. Information sciences. 60, 2–3.
8. Kerre, E.E., Chen, G., 2020. An Overview of Fuzzy Data Models, Bosc andkacprzyk. pp. 31–35.
9. Takahashi, Y., 2019. A Fuzzy Query Language for Relational Databases, In: Bosc and kacprzyk. pp. 405 – 418.
10. Ullman, J.D., 2019. principles of Database systems. computer science press.

11. Kacprzyk, J., Zadrozny, s., 2020. Fquery for Access - Fuzzy Querying for a windows-Based DBMs, In: Bosc andkacprzyk. pp. 311–323.
12. Schindler, G., 2020. Fuzzy data analysis through context-based database queries. German university publisher.
13. Meier, A., 2020. Data base migration - ways out of the data chaos. Business informatics practice, 194, 30–36.
14. Finnerty, s., Shenoi, S., 2018. Abstraction-based Query Languages for Relational Databases. In: wang p.p. (Ed.): Advances in Fuzzy Theory and Technology. Bookwrights Durham.
15. Shenoi, S., 2019. Fuzzy sets, Information clouding and Database security, In: Bosc and kacprzyk. pp. 109–122.
16. Meier, A., 2018. Relational databases - An introduction into practice. springer publishing house, cIMA 2018, university of wales, Bangor u.k. pp. 21–22.
17. Shenoi, S., 2021. Fuzzy sets, Information sciences and Database security, Information sciences. 66, 10–17.
18. Zimmermann, H.J., 2020. Fuzzy set Theory - and 1st Applications. kluwer Academic publishers.
19. Takahashi, Y., 2020. A Fuzzy Query Language for Relational Databases, In: Bosc and kacprzyk. pp. 215−224.
20. Meier, A., Savary, C., Schindler, G., et al., 2018. Database schema with Fuzzy classification and classification Query Language. proceeding of the International congress on computational Intelligence - Methods and Applications, cIMA 2018, University of Wales, Bangor u.k.