THE STUDY OF NORMAL GRADE FORM OF RELATIONAL DATABASE THROUGH BASED ON ROUGH SETS THEORY

GayathriVoggu, NAGIREDDY RAJENDER REDDY, K MAHESWARAN
Assistant Professor, Department of Mathematics, Ellenki college of Engineering and Technonlogy, patelguda (vi)' near BHEL ameenpur (m), Sangareddy Dist. Telangana 502319.

Abstract

In this paper a new method to judge the grade of normal form for relational database is proposed based on rough sets theory. First, some concepts about 1NF, 2NF, 3NF and BCNF are given and the principles of rough set theory are discussed. Second, the method to judge the grade of normal forms for a given relation is analyzed using rough sets theory, and some properties of a relation satisfying some grade of normal form are obtained. The study in this paper is a new application of rough sets theory.

1. Introduction

It's well known that the normalization of relational database is an important research field in database theory. Previous studies show that Boyce-Codd normal form is the highest grade normalization if functional dependency is only considered.

Normal form and functional dependency (FD) are two key concepts in the relational database model, which are the kernel of the relational normalization theory. The relational normalization theory is the foundation of relational database logic design, and the handle ability about relational normalization will directly affect the design quality of database system. It has been shown that redundancies and various updating anomalies (threatening the integrity of database) can be avoided by designing relation schemes which conform to certain normal forms [1][2].

The normalization theory was proposed by E. F. Codd in 1970's, and the rough set theory was introduced by Pawlak in 1982[3]. Although the theory research of relational normalization has been complete up to now, it is necessary to be developed and perfected. The reason for that are the verdict of functional dependency is mainly depend on the semantics of attribute in relation theory and it is difficult to be handled in practical applications. Moreover, the rough set theory has accelerated the development of relational database theory. In this paper, we will use rough set theory to judge the functional dependencies in a relation. In addition, we can analyze its normal form grade using rough set theory for an arbitrary relation.

2.Normal form and rough set theory

Definition 1 A relation schema consists of 1) the name of the relation. Relation names must be unique across the database. 2) The names of the attributes in the relation along with their associated domain names.

3) The integrity constraints. Integrity constraints are restrictions on the relation instances of this schema [4]. Definition 2 A functional dependency, denoted by X

□ Y, between two sets of attributes X and Y (X and Y are subsets of R) specifies a constraint on the possible tuples that can form a relation instance r of R: for any two tuples t1 and t2 in r such that t1[X]=t2[X], we must have t1[Y]= t2[Y].

A functional dependency is a property of the meaning or semantics of the attributes, i.e., a property of the relation schema. They must hold on all relation states (extensions) of R. Relation extensions r(R) that satisfy the FD is called

ISSN (PRINT): 2393-8374, (ONLINE): 2394-0697, VOLUME-4, ISSUE-4, 2017
DOI: 10.21276/ijcesr.2017.4.4.1

legal extensions.
Definition 3 A FD X □ Y is a full functional dependency if removal of any attribute from X means that the dependency does not hold any more; otherwise, it is a partial functional dependency.

Definition 4 A relation R is in second normal form if every non-prime attribute A in R is not partially dependent on any key of R.
In other words, R is in 2NF if every non-prime attribute A in R is fully dependent on every key of R.
Definition 5 A relation R is in Boyce-Codd normal form if for every FD X □ A that holds on R, X is a superkey of R.Normalization is a procedure that allows the non-normalizedschemastobetransformedintonewschemasforwhichthesatisfactionofanormalformis guaranteed
Rough set theory is based on equivalence relations describing partitions made of classes of indiscernible objects, and it is ground on the premise that lowering the degree of precision in the data makes the data pattern more visible, whereas the central premise of the rough set philosophy is that the knowledge consists in the ability of classification. In other words, the rough set approach can be considered as a formal framework for discovering facts from imperfect data [5].
Definition 6 An information system I is a system
<U, A>, where U={u1,u2,…,u|U|} is a finite non-empty set, called a universe or an object space, elements of U are called objects; A={a1,a2,…a|A|} is also a finite non- empty set; elements of A are called attributes; for every
a □ A there is a mapping a from U into some space, i.e.
a:U □ Da(U) and a(U) = {a(u)|u□U} is called
the
domain of attribute a.

3. The relationships between equivalence relation and functional dependencies

In section 2, the definition of functional dependency is given. According to the fundamental principle of rough set theory, partition, equivalence relation and functional dependency have any relationships. We analyze their relationships as follows.
We define a partial ordering among partitions of U called refinement. Let P1 and P2 be partitions of U. We say P1 is a refinement of P2, written P1 □P2, if S2 □P2[6].
Lemma 1 Let □□□ be a over R, and let r(R) be a table. Then r satisfies □□□ if and only if P □□ P□
Lemma 1 allows statements about functional dependency to be translated into equivalent statements about partitions, in other words, the judgement of functional dependency for all relations can finish by partition or equivalence relation. Moreover, we can determine the grade of normal forms that the relation satisfies.

4. The judgement principles of normal forms based on rough set theory

In this section, we will study the judgement principles of normal forms for a relation based on rough set theory.In the relation normalization theory, database schema is the set of relation schema. For a database relation, the following theorems can be obtained.

Theorem 1 Let □□□ be dependency on R <U, F>, r(R) be a relation, then r(R) satisfies

□□□ if and only if Card (P□) =m and Card (P □) □m h tuples, Card () is the cardinal number of partition, P □ and P□are partiti

Theorem 2 Let R<U,F> be a database schema, r □R<U,F> only if t[A]|=1, where | t[A]| is the cardinal number of attribute elements included in t[A].

The relation schema R = {E#, JC, D#, M#, CT}, where E#: employee number ; JC: job code ; D#: department number; M#: employee number of manager; CT: contract type. The relation r □R is sho amending some attribute values in paper [7].

Theorem 3 Let R<U,F> be a relational database schema, and r □R<U,F> attribute set of r be Z= {key1, key2,…,keyn}, the non- prime attribute set of r be Z □. If r □2NF, the P keyi □Pattr holds while Pk

ISSN (PRINT): 2393-8374, (ONLINE): 2394-0697, VOLUME-4, ISSUE-4, 2017
DOI: 10.21276/ijcesr.2017.4.4.1

□Pattr does not always hold, where k is a proper subset of keyi and P is a partition defined in r.

TABLE I : Employee data

E#	JC	D#	M#	CT
1	A	x	11	g
2	C	x	11	g
3	A	y	12	n
4	B	x	11	g
5	B	y	12	n
6	C	y	12	n
7	A	z	13	n
8	C	z	13	n

Proof: Let R<U,F> be a relational database schema, and r □R<U,F> be a relation, the prime attribute set of r be Z= {key1,key2,…,keyn}, the non-prime attribute set of r be Z □. Now given r □2NF, then according to the definition of functional dependency, every non-prime attribute in r is not partially dependent on any key of r.

So for an attribute attr □Z□ keyi□attr and k↛attr hold, where k is a proper subset of keyi, according to the conclusion of lemma 1, P keyi □Pattr holds while

Pk □Pattr does not always hold.

For example in TABLE I, the relation schema satisfies 2NF, and its prime attribute set is {E#}, non- prime attribute is {JC, D#, M#, CT}. According to theorem 3, U/E# □U/JC, U/E# □U/D#, U/E#□U/M# and U/E#□U/CT hold (E# hasn't proper subset), and U/E#={{1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}},

U/JC={{1,3,7}, {2,6,8}, {4,5}}, obviously, U/E# □

U/JC holds, similarly, U/E#□U/D#, U/E#□ U/M# and

U/E# □U/CT hold.

On the contrary, a judgement theorem which a relation satisfies 2NF or not can be obtained.

Theorem 4 Let R<U,F> be a relational database schema, and r □R<U,F> be a relation, r□1NF, let the prime attribute set of r be Z={key1, key2, …, keyn}, and the non-prime attribute set of r be Z □. If an keyi □Pattr and relational schema r □2NF, wh subset of keyi, and P is a partition defined in r. On the contrary, for all attributes attr □Z□in a relation schema r □1NF, if while Pk □Pattr does not

Proof: Let R<U,F> be a relational database schema, and r □R<U,F> the prime attribute set of r is Z={key1, key2, …, keyn}, and the non-prime attribute set of r is Z □. If an keyi □Pattr and to lemma 1, keyi □attr and according to the definition of 2NF, r □2NF. On the contrary, for all attributes attr □Z□ in a relation schema r □1NF, if while Pk □Pattr does not hold, then keyi ↛attr and k hold. According to the definition of 2NF, the relation schema satisfies 2NF.

For example in TABLE I, the prime attribute set is

{E#}, and the non-prime attribute set is Z □={JC,D# {4}, {5}, {6}, {7},

{8}}, U/JC={{1, 3, 7}, {2, 6, 8}, {4, 5}}, obviously

U/JC, U/E# □ U/JC ho U/E# 6}, {7,

8}}. So U/E# □U/D# ho □U/M# and U/E#□U/CT hold. In addition, E# hasn't proper subset, so this relation schema satisfies 2NF. Theorem 5 Let R<U,F> be a relational database schema, and r □R<U,F> relation, r □1NF, let be Z={key1, key2,…, keyn}, and the non-prime attribute set of r be Z □. If r□3 keyi □Pattr hol not exist an attribute k and proper subset kk of keyi such that P keyi □ Pk, Pk□ Pattr hold, where P is a partition defined in r.

Above theorem 5 gives the properties of a relation that non-prime attributes satisfy when this relation schema satisfies 3NF.

Theorem 6 Let R<U,F> be a relational database schema, and r
let the prime attribute set of r be Z={key1, key2,…, keyn}, and the non-prime attribute set of r be Z
attribute k makes P keyi
exists, then relation schema r
a proper subset of keyi and P is a partition defined in r. On the contrary, for all attributes attr in Z□a relation schema r
keyi
such that Pk □Pattr and an attribute set kk in r

such that P keyi
exist, thenr
prime attribute set is{E#}, and the non-prime attribute set is Z
obtain U/E#={{1}, {2},{3}, {4},{5}, {6},{7},{8}}
{7,8}}, whereas U/M#={{1,2,4},{3,5,6},{7,8}}, so U/D# □
U/M#, indicating transitive dependencies exist in this relation, so r □3NF.

Theorem 7 Let R<U,F> be a relational database schema, and r
let the attribute set of r be U={attr1,attr2,…,attrn}, if r
following conclusions can be obtained:
1) P key j
P key j and P attri are partitions formed by keyj and attri respectively; 2) both an attribute set k and proper
subsetkk of keyi such that P keyi
Pk
where P is a partition of r.
The theorem above gives some properties that all attributes of a relation satisfy BCNF.

Theorem 8 Let R<U,F> be a relational database schema, and r
let the attribute set of r be U={attr1,attr2…attrn}, if r
□P attri holds for all attribute attri□U, and an attribute set k and proper subset kk of keyi cause P keyi
not exist in r, where P is a partition of r, so r
attri are partitions formed by keyj and attri respectively.
For example in TABLE I, its prime attribute set is
{E#}, and its non-prime attribute set is Z
although U/E#

U/M# and U/E# □U/CT ho
□R<U,F> be a relation, r□3NF, in this relation,
Functional dependencies exist in this relation, such as E# □D# and
does not satisfy BCNF.
□5.For all attributes attr□Z□, if an
□Pk and Pk□Pattr
□3NF, where k is

Conclusions

In this paper we discuss the judegement principle of functional dependency and normal form grade in a given relation using rough set theory. This study extends the application areas of rough set theory. Furthermore, we will study the judgement arithmetic and related partition dependency and its depth applications in database theory [8].

□1NF, if P
□Pattr holds, and proper subset k of keyi
□P kk and Pkk□Pattr do not
□3NF.For example in TABLE I, the
□={JC,D#, M#,CT}. We can
□U/D#= {{1, 2,4}, {3,5,6},
□R<U,F> be a relation, r□1NF,
□BCNF, then
□P attri holds for all attri□U, where
□Pk,
□Pattr and Pkk□Pattr do not exist in r,
□R<U,F> be a relation, r□1NF,
□1NF, then P key j
□Pk , Pk□Pattr and Pkk□Pattr do
□BCNF can be obtained, where P key j and P

References

[1] E. F. CODD, “Recent investigations in relational database systems.” Information Processing 74, North-Holland Pub. Co., Amsterdam, 1974, pp. 1017-1021.

[2] J.D. Ullman, Principles of Database Systems, Computer Science Press, Rockville, MD, 1982.

[3] Z. Pawlak, Rough Sets. Theoretical Aspects of Reasoning about Data, Kluwer Academic Publisher, Dordrecht, Netherlands, 1991.

[4] Philip M.Lewis, Arthur Bernstein, Michael Kifer. Databases and transaction processing:An Application-Oriented Approach. Higher Education Press Pearson Education,2002, ch. 4.

[5] B. Walczak, D.L. Massart, “Rough sets theory.” Chemometrics and Intelligent Laboratory systems 47, 1999 , pp.1–16.

[6] M.J.Fischer, “Notes on Functional Dependencies and partitions(CPSC 437b: Introduction to Databases).” February 28, 2003, Handout #10.

[7] Philip A. Bernstein, Nathan Goodman, “What Does BOYCE- CODD Normal Form Do.” Sixth International Conference on Very Large Data Bases, October 1-3, 1980, Montreal, Quebec, Canada, Proceedings. IEEE-CS, 1980, IEEE Catalog Number 80 ,pp 245-259.

[8] WojciechZiarko, “Partition Dependencies in hierarchies of probabilistic Decision Tables.” Rough Sets And Knowledge Technology, 2006, 7,pp.42-49.

□={JC, D#, M#,CT}. According to theorem 8,
□ U/JC,U/E#□U/D#, U/E#□