IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4680

The Availability of Domain/Key Normal Form

Robert A. Schultz, Professor
Computer Information Systems, Woodbury University
7500 Glenoaks Blvd., Burbank, CA 91510
P: 818-767-0888, F: 818-504-9320, bob.schultz@woodbury.edu

INTRODUCTION

In my earlier paper, “Understanding Functional Dependency” (2002), I distinguished intensional and extensional characterizations of functional and other dependencies used in defining the Normal Forms for relational databases. In that paper, I left incomplete my discussion of how this distinction applies to Domain/Key Normal Form (DK/NF). In this paper, I will continue that discussion.

Extensional characteristics are those which remain the same through substitution of terms with the same reference, whereas intensional characteristics do not. In a database context, extensionality means that the only features of fields appealed to is the frequency of their appearance with other fields, with the same or different values. Meanings or connotations of field names or values, and connections between fields having to do with knowledge about the meanings or business rules or conventions connecting field values are intensional and therefore have no place in extensional database considerations. In my earlier paper (2002), I showed that the first three normal forms can be done on an extensional basis (and Boyce-Codd, fourth and fifth normal forms as well). Although many texts mention intensional elements in defining functional dependency, this is not necessary.

Normal Forms exist in order to produce well-behaved database designs that avoid the occurrence of anomalies, unexpected difficulties with deleting, adding, or modifying data. If a database is in DK/NF, it is provable that no anomalies can occur. Whereas, the other Normal Forms (First, Second, Third, Boyce-Codd, Fourth, and Fifth), were designed to avoid certain anomalies, and there is no guarantee that some further anomaly may pop up not prevented by these Normal Forms. Unfortunately, there is no effective procedure for putting a set of tables into DK/NF. As David Kroenke (2002, 134) puts it, “Finding, or designing, DK/NF relations is more of an art than a science.”

Ron Fagin’s original paper (1981) on DK/NF is done squarely within standard mathematical set theory, which is completely extensional in character. This implies that his characterization of possible anomalies and his proof that tables in DK/NF avoid them are also extensional. If they are reflected in patterns of repetition in the data, then one would want to know why they, or their absence, could not be detected by a simple (but possibly extended) search of the data for repetitions or their absence. This is in fact how “automated” procedures for doing first, second, and third normal form work. So offhand it seems as though DK/NF and its accompanying anomalies must not be reflected solely in patterns of repetition in the data.

In this paper, I will discuss:

1. The basis for the claim that all possible anomalies are prevented by DK/NF;
2. Why there is no effective procedure for producing DK/NF.

DK/NF AND THE ANOMALIES

Fagin defines *insertion anomaly* as follows:

(1) Relation schema R* has an *insertion anomaly* if there is valid instance R of R* and there is a tuple t compatible with R such that RÈ{t}, the relation obtained by inserting t into R, is not a valid instance of R* (i.e., violates a constraint of R*)(391).

If we replace ‘relation schema’ with ‘table structure’ (a relation schema is a set of attribute names plus some constraints on possible values for those attributes), ‘relation’ with ‘table’ (understood as with all its values), ‘tuple’ with ‘record,’ (1) reads as (1)’:

(1)’ A table structure R* has an *insertion anomaly* if a table with the structure R* can have a record added which by itself satisfies the constraints of the structure but the resulting table violates a constraint of the structure R*.

Fagin is almost always working at the level of generality of what I call table *structures* rather than tables themselves. Usually when anomalies are discussed in database textbooks, they are presented in terms of a few examples of tables without any attempt at a general formal characterization. So, how does Fagin’s formal definition square with intuitive examples of insertion anomalies?

Intuitive examples of insertion anomalies involve a nonkey field which has another functionally dependent field; thus it is not possible to add a record exhibiting a new nonkey dependency until a record with the key and nonkey fields is added. An example in Kroenke is a StudentActivity table with StudentID, Activity, and ActivityCost fields. Since StudentID functionally determines Activity and Activity functionally determines ActivityCost, there is no way to insert a record with a new Activity and ActivityCost until some student decides to engage in the activity (Kroenke, 126).

Fagin’s definition does not apply; and in fact his example of an insertion anomaly is different. The table structures are the same, but in terms of the above example the anomaly would be a new student with an existing activity who was assigned a different cost for that activity (Fagin, 392). On Fagin’s approach, the intuitive insertion anomaly is simply an attempt to insert an ineligible record. In fact, Fagin, in a discussion of Codd (1972), seems to view as illegitimate the attempt to restructure tables to capture information such as a student activity fee in the absence of a student (401).

Fagin’s definition of deletion anomaly is:

(2) Relation schema R* has a *deletion anomaly* if there is a valid instance R of R* and a tuple t in R such that the relation obtained by removing t from R is not a valid instance of R* (i.e., violates a constraint of R*)(395).

Replacing terms as above, (2) reads as (2)’:

(2)’ A table structure R* has a *deletion anomaly* if a table with the structure R* can have a record deleted which results in the table violating a constraint of the structure R*.

Intuitive examples of deletion anomalies involve nonkey fields with others functionally dependent on them; thus information about the dependency can be lost when the record is deleted. In the example StudentActivity(*StudentID*, Activity, ActivityCost), information about ActivityCost can be lost if all students are deleted who happen to be engaged in that activity (Kroenke, 126). Fagin’s examples of (DK) deletion anomalies are based on domain dependencies (396). In this case, Fagin’s example is similar to the intuitive one. Information about the relation of value of two fields is inadvertently lost because of the table structure. The difference is the nature of the dependency between the fields.

Also, Fagin’s definition of insertion and deletion anomaly supposes that the constraints mentioned in (1), (1)’, (2) and (2)’ are Key Dependencies and Domain Dependencies only, and not functional dependencies (FDs), multivalued dependencies (MVDs), and join dependencies (JDs). So Fagin’s insertion anomalies should more appropriately be called “domain key insertion anomalies,” and similarly his deletion anomalies should be called “domain key deletion anomalies.” So understood, the actual theorem stating that table schemata are in DK/NF if and only if they have no DK deletion or DK insertion anomalies seems perhaps less dramatic. Fagin himself notes that this theorem is "not deep" (398), presumably for similar reasons.

However, later in the paper he proves that DK/NF implies the traditional normal forms defined in terms of functional dependency multivalued dependency, and join dependency (403-409). The DK/NF theorem together with the implication of traditional normal forms does still show that DK/NF prevents whatever anomalies the traditional normal forms do, plus any DK anomalies.

FAGIN'S PROOF THAT DK/NF IS FREE OF ANOMALIES

If a table structure is in DK/NF, any table with that structure has no (DK) insertion or (DK) deletion anomalies. Fagin shows this by appealing to the definitions. A table derived from the original by deleting or inserting a tuple also satisfying the DK constraints, will satisfy any constraint the original does. So no anomalies.

If a (consistent) table structure is not in DK/NF, then it has anomalies. Fagin shows this by constructing an anomaly. If a table structure is not in DK/NF, there is a "bad" table for which a constraint fails when the set of DK constraints holds. Since the table structure is consistent, there is a "good" table which does satisfy the constraints. He constructs a sequence of tables starting with the "good" one, deleting one row at a time until all are gone. Then add one row from the "bad" table until we have the complete "bad" table. Somewhere in this sequence, the table goes from "good" to "bad". If it is in the first part, we have found a deletion anomaly. If it is in the second part, we have found an insertion anomaly.

AN EFFECTIVE PROCEDURE FOR DK/NF?

Fagin closes his discussion of how to put a database into DK/NF with a warning:

> *In the general case, it is not useful to think in terms of mechanical procedures for convesion to DK/NF, since we immediately run into undecidability results. For example, it is not even decidable as to whether a sentence of first-order logic is a tautology (this is Church's theorem).* (Fagin, 403)

To determine whether a table structure is in DK/NF, we have to determine that every constraint can be *inferred from* key dependencies and domain dependencies alone. The question is what rules of inference can be used. As long as we allow at least first-order predicate calculus (quantification theory), it is know that no decision procedure exists for what can be inferred from what. If an inference is valid, that can be proved. But for an arbitrary inference, it cannot be decided whether it is valid or not (Quine 1966, 212).

Fagin deliberately refuses to restrict constraints only to formulations in quantification theory (389). Quantification theory (or predicate logic) does seem to be a bare minimum because otherwise we cannot formulate constraints mentioning fields or attributes. Even though a restriction to quantification theory will not help, Fagin also raises the question of restricting domain and key dependencies to allow decidability for DK/NF (403). But such restrictions on domain and key dependencies may also invalidate the proof that all (DK) anomalies are prevented by DK/NF.

CONCLUSION

My original suspicion that the extensional/intensional distinction by itself might help understand the status of DK/NF turned out to be incorrect. Fagin makes only one explanatory statement about DK/NF using intensional considerations: "A 1NF relational schema is in DK/NF if every constraint can be inferred by simply ***knowing*** the DDs (domain dependencies) and the KDs (key dependencies)" (397, my italics). However, the rest of the paper is completely extensional in character. So help with my two puzzles about DK/NF lies elsewhere.

On the basis for the claim that all possible anomalies are prevented by DK/NF, there are two facts: DK/NF prevents all DK insertion anomalies and DK deletion anomalies; and DK/NF implies the traditional normal forms. Thus all DK insertion and deletion anomalies are prevented, and whatever anomalies prevented by the earlier normal forms are also prevented by DK/NF. However, some intuitive anomalies are not recognized as DK anomalies, even though they will also be prevented by DK/NF.

On why there is no effective procedure for producing DK/NF, I found that DK/NF is defined in terms of very general kinds of inference, including classes of inference known to be undecidable. My conclusion was that it was probably not workable to restrict allowable *inferences* to avoid undecidability, and that a restriction of allowable *dependencies* would have unpredictable effects on whether all anomalies can be prevented by DK/NF.

Taken together, both points suggest that the theoretical claims for DK/NF are probably unassailable. However, the practical difficulties in achieving DK/NF also can probably not be ameliorated.

REFERENCES

Codd, E. F. (1972) "Further Normalization of the Relational Database Model." in *Data Base Systems*. Englewood Cliffs, NJ: Prentice-Hall.

Fagin, Ronald (1981) "A Normal Form for Relational Databases that is Based on Domains and Keys," *ACM Transactions on Database Systems,* v.6 no.3, pp. 387-415.

Kroenke, David (2002) *Database Processing*. Upper Saddle River, NJ: Prentice-Hall.

Quine (1966) "Church's Theorem on the Decision Problem", *Selected Logic Papers.* New York: Random House.

Schultz, Robert (2002) "Understanding Functional Dependency," *2002 Proceedings of IRMA.* To appear in *Effective Databases for Text and Document Management,* ed. Becker.