UDC 004.9:004.75

Yu.M. LISETSKYI[*]

METHODS TO BUILD OPTIMAL DATABASE MODEL

[*]S&T Ukraine, Kiev, Ukraine

Анотація. *У статті розглядається пошук оптимальних моделей баз даних та ефективних способів їх побудови. Розглядаються традиційні моделі даних, а також виявляються їх обмеження та недоліки при впровадженні в сучасні інформаційні системи. Вивчаються основні моменти теорії нормалізації реляційних баз даних, а також визначаються основні вимоги до оптимальної моделі бази даних.*

Ключові слова: *модель, структура даних, база даних, система управління базами даних, відносини, нормалізація, аномалії.*

Аннотация. *В статье рассматривается поиск оптимальных моделей баз данных и эффективных способов их построения. Рассматриваются традиционные модели данных, а также выявляются их ограничения и недостатки при внедрении в современные информационные системы. Изучаются основные моменты теории нормализации реляционных баз данных, а также определяются основные требования к оптимальной модели базы данных.*

Ключевые слова: *модель, структура данных, база данных, система управления базами данных, отношения, нормализация, аномалии.*

Abstract. *The paper considers the search for optimal database models and effective ways of their building. Traditional data models are considered and their limitations and shortcomings are revealed during implementation in modern information systems. The main points of relational database normalization theory are studied as well as essential requirements to optimal database model are defined.*

Keywords: *model, data structure, database, database management system, relations, normalization, anomalies.*

1. Introduction

It has already been for quite a long time that experts are discussing the crisis of traditional databases which include not only relational but also the object-oriented model [1–3]. In particular, they note that it is the underlying primitive data structure which imposes significant limitations on the relational model. This structure is not effective enough for implementation in the modern information systems working with heterogeneous data and dynamically changed data structures. Other significant limitations of relational databases are their limited capacity in representing application semantics and insufficient connection between conceptual and physical layers of data representation which leads to abrupt leaps between different phases of software development process. In this connection, it appears reasonable to search for optimal database models and most effective ways of their building.

2. Main Part

The optimal logical database model is the model free from anomalies caused by database (DB) modification being the issues connected with data updates, insertions or deletions.

To build such a database model the relational database normalization model is applied, irrespective of what database management system (DBMS) is used: hierarchical, network or relational.

Normalization of relationships of the informational model of the subject matter is the mechanism to build the relational database logical model. From the mathematical point of view the task of building both informational model of the subject matter and relational database logical model is addressed by resolution of the following combinatory tasks:

- attributes grouping in relation to subject matter;
- attributes distribution as to database relations.

Normalization of relations is the backward iterative process of initial relation decomposition into several simpler and smaller relations.

The reversibility suggests that composition of relations resulted from decomposition should produce the initial relation. As the result of normalization the database relations attributes have to meet the following criteria:

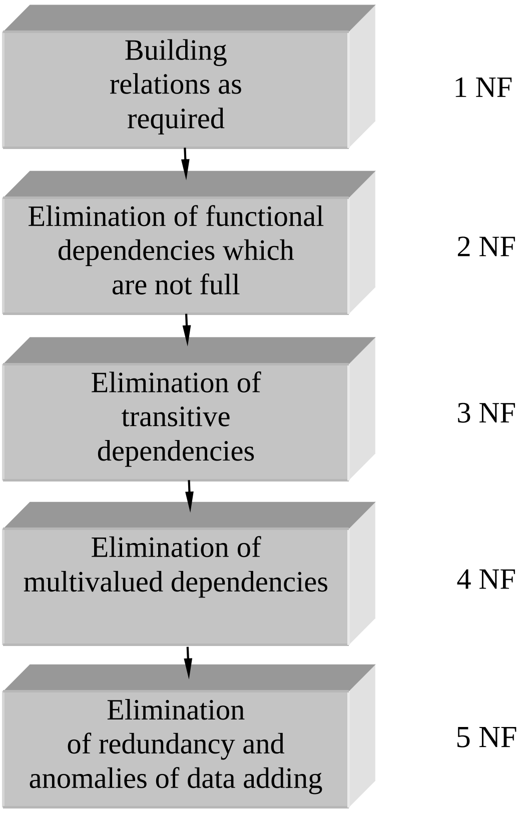
Building relations as required
1 NF
Elimination of functional dependencies which are not full
2 NF
Elimination of transitive dependencies
3 NF
Elimination of multivalued dependencies
4 NF
Elimination of redundancy and anomalies of data adding
5 NF

Fig. 1. Steps of Relation Normalization

– undesirable functional dependencies between attributes should be eliminated;

– attributes grouping should be free from unnecessary data duplication;

– attributes procession and restoration should be free from complications.

Normalization apparatus had been developed by E.F. Codd [4]. Every normal form limits the type of permitted dependencies between attributes. E. Codd described three normal forms (abbreviations: 1NF, 2NF, 3NF). Today the 4NF and 5NF are known and described as well. Relations normalization is completed in several steps (fig. 1).

1st step is the transformation of relations to First Normal Form (1NF). The 1NF relation should meet the following requirements:

– all relation attributes should be atomic;

– all table rows should have the same structure or have the same number of attributes with identical names;

– column names should be different while values should be homogeneous (have the same format);

– the sequence of rows is insignificant.

Each database relation contains both structural and semantic information. Structural information is set by relation schema while semantic information expresses functional connections of attributes.

The relation keys are derived during the 2nd step of normalization as well as corresponding dependencies are analyzed to eliminate functional dependencies which are not full.

Definition 1. Attribute B depends on A in relation R when every moment of time not more than one value of B corresponds to the same value of A. Functional dependency corresponds to the 1:1 relationship of attributes.

Definition 2. The attribute is in full functional dependency if it depends on the whole key and does not depend on its components.

If relation has functional dependencies which are not full then it is decomposed into two or more other relations without functional dependencies which are not full and composition of thereof will produce initial relation.

2NF benefits: convenience of modifications. It is significantly easier to modify 2NF database compared to non-normalized database.

3rd step of normalization provides for elimination of transitive dependencies. 3NF relations should be analyzed for presence of transitive dependencies.

Transitive dependency is the dependency of non-key attributes.

For instance, in relation R(A*,B,C,D) where attribute D is not immediately dependent on key but is dependent on non-key attribute C which is dependent on A, the D is said to be transitively dependent on A.

Transitive dependencies are eliminated by relation decomposition into two or more relations with no transitive dependencies and composition of which will produce initial relation.

The 4th step of normalization which is also called 4NF or Boyce-Codd Normal Form provides for analysis for presence of independent multivalued dependencies in relation. If there are then the relation is decomposed.

Multivalued dependency is the type of functional dependency. It is corresponded by 1:B relationship of attributes.

Attribute B has multivalued dependency on attribute A in relation R(A,B,C) if B depends only on A in any of its combinations with other relation attributes.

If the relation has A®B and A®C then it should be decomposed into two other relations R(A,B) and R(A,C). The notion of multivalued dependency is more complicated than the notion of functional dependency. Its revelation requires a significantly deeper semantic analysis of attributes. There exist trivial and non-trivial multivalued dependencies.

Dependency of X®Y and Y®X type is trivial while X®Y and Y®X dependency is non-trivial. Presence of non-trivial multivalued dependencies in relation schema and independency of their right sides defines the combinatorics of the right sides of relation.

Definition 3. Relation R is in 4NF when the structure of multivalued dependency defined on the multitude of attributes contains only trivial or non-trivial multivalued dependencies with left side of each of them being the key.

Decomposition of initial relation into several other should guarantee its reversibility, or provide for producing initial relation through composition of relation found through decomposition. However, decomposition does not always guarantee reversibility. Relations with more than three multivalued dependencies require special measures to guarantee decomposition reversibility. For this purpose there exists 5NF. The 4NF decomposition produces projections containing at least one possible key and at least one non-key attribute of initial relation.

The 5th step of normalization eliminates redundancy and anomalies of updating the database. Anomaly is a scientific term for issues which might possibly arise out of work with non-normalized tables. This is why the whole hierarchy of normalized forms is built in the manner where every next form limits the list of possible anomalies of previous form. This process corresponds to the process of decreasing database entropy or presence of redundant information.

The pointed dependency relationships between three attributes are a very rare occasion. Dependency relationships between four, five, and more attributes are practically impossible to identify. Certain DBMS have special mechanisms eliminating possibility of retrieval of unreliable information. However, there should be followed the general recommendation that database structure be built in such a manner that 4NF and 5NF become unnecessary.

We have considered five normal forms; however, they do not exhaust the list. In 1981 R. Fagin published the paper [5] introducing the notion of Domain/Key Normal Form (DKNF). He demonstrated that DKNF relation has no modification anomalies. So whatever the changes are there are no losses in DKNF if all the key and domain constraints are observed.

In fact, the definition is quite general but its essence is that if all rules are followed then whatever the actions with the table can be its consistency and all necessary information is preserved.

DKNF. The relation variable is in DKNF if and only its every constraint is a logical consequence of domain and key constraints for relation variable. Domain constraint is the constraint requiring usage of only the values from set domain for certain attribute. The constraint in itself is only the list (or logical equivalent of the list) of permitted values of type and declaration that the attribute has this type.

The key constraint is the constraint declaring that certain attribute or combination of attributes is the potential key. Any relation variable in DKNF is necessarily in 5NF. However, not every relation variable can be reduced to DKNF.

Sometimes DKNF is called 6th Normal Form [6]. The relation variable is in 6NF when and only when it meets all non-trivial constraints of relationship. The definition suggests that variable is in 6NF when and only when it is irreducible or cannot be further decomposed without losses. Every relation variable which is in 6NF is also in 5NF.

The idea of final decomposition arouses before the studies on chronological data but found no support. However, maximal possible decomposition of chronological database enables fighting redundancy and simplifies maintaining database consistency.

To summarize it we would like to note that relations normalization eliminates the following dependencies of attributes: non-full functional, transitive, non-trivial (independent) multivalued. By eliminating these dependencies we avoid data duplication as well as anomalies during data updates, replacements and deletions.

Here are the essential requirements to optimal database:

1. Adequate representation of subject matter logics in respective data model.

2. Reasonable data redundancy. The database should be the single aggregate of integrated data.

In the systems which do not use databases every application will have its files. For instance, the application for human resource management and application for personnel training can both have their own files with information on personnel. It leads to redundancy in data storage. The inconsistency may arise out of data redundancy when, for instance, two records for the same employee differ.

3. Availability of effective database management tools (creation, addition, modification, deletion and search).

Data creation tools are designed to upload data from external user-oriented representation into a system one.

4. Data consistency (meeting the requirement of uniqueness of all database records and their consistency during users' operation as well as simultaneous modifications management).

Consistency requires correctness and accuracy of database data. The conflict of two records representing the same fact is an example of insufficient consistency. Majority of existing databases are characterized by lack of sufficient consistency support control.

5. Data security. It is the protection from unauthorized data access and database destruction (willful or occasional).

Centralized nature of database system requires existence of a security system. The data access is only permitted for authorized users.

6. Database restructuring. There should be tools for data restructuring required when database queries are changed.

7. Availability of language tools for data definition and manipulation which are concise, convenient and easy to learn. These are data definition and data manipulation languages. Autonomous data language or the language not included into universal language is also called the query language.

8. Availability of documentation.

9. Simplicity of learning.

10. Mutual independence of programs and data.

3. Summary

Thus, the optimal database should preserve operability in case and in the course of software and hardware development. Changes to physical data organization or storage device parameters should have no impact on a user or rather on application. Changes to user representation should require no expenses on reorganization and modification of mechanism of access to physical data files. Data independency enables system functioning during changes from both sides (user and physical data) which is the most important feature and main database goal. Also, it impacts other features including data redundancy, availability of security, consistency etc. Data independency can be defined as the application immunity to changes in data storage structures and data access methods.

REFERENCES

1. Codd E.F. Recent Investigations in Relational Data Base Systems / Codd E.F. – San Jose: IBM Corporation, 1974. – 42 p.

2. Когаловский М.Р. Энциклопедия технологий баз данных / Когаловский М.Р. – М.: Финансы и статистика, 2002. – 800 с.

3. Крёнке Д. Теория и практика построения баз данных / Крёнке Д. – [8-е изд.]. – Питер, 2003. – 800 с.

4. Codd E.F. Further Normalization of the Data Base Relational Model / E.F. Codd // In Data base systems, Englewood Cliffs. – N.J. Prentice-Hall, 1972. – P. 33 – 64.

5. Fagin R. A Normal Form for Relational Databases That Is Based on Domians and Keys / R. Fagin // TODS. – 1981. – Vol. 6, N 3. – P. 387 – 415.

6. Дейт К.Дж. Введение в системы баз данных / Дейт К.Дж.; пер. с англ. М.Л. Степановой. – [8-е изд.]. – М.: Вильямс, 2005. – 1328 с.

Стаття надійшла до редакції 29.01.2018