Reengineering relational database on analysis functional dependent attribute

Valentin Filatov, Vjatcheslav Radchenko

Abstract - **The task of re-engineering information system which is based uses a relational database. An approach to the definition of functionally dependent attributes of the database in step reengineering and modified synthesis algorithm logic of a relational database.**

Keywords - **reengineering, information systems, relational database, relation, the database schema, functional dependencies, normalization.**

I. INTRODUCTION

Reengineering of information systems, which are based on a database (DB), has recently attracted increasing attention of specialists in the sphere of modern information technologies. It is due to several reasons. During its operation, the database will inevitably undergo changes due to the variability of the sphere. A key component, which largely determines the characteristics of a relational database (RDB), is a logical scheme, which is a description of the data tables and links between them.

To date, the most common are relational databases that provide the best combination of reliability, ease of use and performance for different tasks. Therefore, it's impossible to do maintenance tasks without changing the logical database schema. The qualification of database design determines the efficiency of the functioning of an information system in general and the ability to adapt to changing requirements.

II. THE PURPOSE OF THE CONDUCTED RESEARCH

The conducted analysis of publications on issues of re-engineering of databases has shown the following. The main areas of research are: reengineering logic via an intermediate representation, such as ER-model, and application of a set of special rules for translation of objects of model to the RDB structures [1,2]; reengineering outdated database [3]; extracting the structure of outdated and relational databases and presenting it as a conceptual data model, in particular, ER-model [4,5].

Valentin Filatov - Harkiv National University of Radioelectronics, Lenin av., 14, Harkiv, 61166, UKRAINE, E-mail: Filatov_val@ukr.net
Vjatcheslav Radchenko - Harkiv National University of Radioelectronics, Lenin av., 14, Harkiv, 61166, UKRAINE, E-mail: cayrad@gmail.com

Also, there are approaches to reengineering RDB without using an intermediate models [6].

As the result of the analysis we can identify classes of RDB reengineering problems:

1. Migration - converting outdated database into a modern analogue (eg., Relational or object); transfer the database to another platform of the same type (eg., MySQL → Oracle);
2. Refactoring - improving the performance without changing the functionality;
3. Adaptation - changes in accordance with the requirements of the sphere.

In general, the process of reengineering involves two stages: the formation of the structure of RDB which in compliance with the new requirements, and data migration into this structure.

The aim of the research is the development and research of methods of re-engineering of relational databases, based on the search, analysis and classification of the set of functional dependencies in the current data set. The set of functional dependencies to be used as input to the method of synthesis of a relational database schema in third normal form.

III. MAIN PART

In the method of the data scheme reengineering is proposed to use an approach based on analysis of the set of stringent functional dependencies. This approach allows us to go from the initial relations to the set of relations in third normal form by applying a sequence of decomposition rules.

Imagine starting RDB as a set $DB = \{\rho_1 ... \rho_n\}$, where ρ_i - the relation of the database, $i = \overline{1,n}$ (n - the number of relations of the database). Express ρ_i in terms of a set $< \sigma_i, p_i >$ where σ_i - logical schema of relation, and p_i - an exemplar of the relation (set of corteges).

The logical schema of relation is considered as $\sigma_i =< R_i, F_i >$, where R_i - the set of attributes and R_i - a lot FD running on R_i.

On the other hand, $DB =< \Sigma, P >$, where $\Sigma = \{\sigma_1 ... \sigma_n\}$ - logical scheme, and $P = \{p_1 ... p_n\}$ - an exemplar of the data of RDB. Logical schema Σ can be expressed $\Sigma = \{R, F\}$ as a set, where $R = \bigcup R_i$ - a

common set of attributes, and $F = \bigcup F_i$ - a common set of FD.

In the process of normalization in the design of the original structure of RDB there is occurs the decomposition universal relation U according to a set of the FD. The purpose of normalization is removing data redundancy and abnormalities updating/deleting. Usually a normal form Boyce-Codd (BCNF) is considered to be a normal form. In this form there are no anomalies.

But due to the fact that not all schemas can be reduced to NBFC, the basic form ifs considered to be third normal form (3NF) [7]. We denote Σ' as the current logical scheme of RDB, which is the result of a number of changes in accordance with changes in the requirements for IP. The current set of attributes denoted by R'. In operation of RDB also happens the transition from P to P' where - P' the current data exemplar.

We can not guarantee the fact that Σ' is in 3NF: if $R' \neq R$, then it follows, that $F' \neq F$, where F' - the current set of FD. Because of $\Sigma' =< R', F' >$, you need to check is the current logical scheme Σ' for compliance with the 3NF, as well as bringing to it otherwise.

To solve this problem is proposed to construct 3NF for the current logical scheme, using the proposed method of synthesis of F. Bernstein [8]. It takes as input the set of FD and generates many possible implementations of schemes $S = \{\hat{\Sigma}_1 ... \hat{\Sigma}_m\}$ that are in 3NF. It should be noted that this problem is not trivial.

The set F', which is the input data for the synthesis method is not fully determined due to the fact that when the set of attributes changes and a set of FD, that are complied with these attributes. Therefore, the first step is to find the set F' of solutions, which is valid for P'.

Let F_S - the set of FD that it is possible to obtain by analyzing the constraints of a relational database, such as primary and foreign keys.

We denote by F_H the set FD, which is implicit dependence of the existence of which was not known at the time of the original design. They exist in the form of patterns in data, established during the operation, and will be used in the synthesis of the target schema. Thus, the desired set FD $F' = F_S \cup F_H$; finding is not considered in this paper.

For the definition of the set F_H there are a number of methods of identifying dependencies of exemplars of RDB. Basically, they are used in tasks of data mining is about and allow approximate FD (AFD). Unlike classical "stringent" FD it is that allowed the existence of the FD, even if in RDB there are lines, that violating correctness FD. This assumption is based on the assumption that in the operation of RDB may be inclusion "wrong" lines. By "wrong" line should be understood a cortege that is not contrary to the integrity of the existing restrictions, but violates observe up to that point implicit FD.

One such method is the method of [9] and its modifications, which served as the basis for other similar solutions. As input data you must provide an exemplar P'; the result of this method is the sought-for set F_H. However, the correct result can not be guaranteed for the data containing null values (NULL).

The initial data for the solution of finding the set of subtasks are: logical scheme of relational database $\Sigma = \{\sigma_i\}, i = \overline{1,n}$, where σ_i - the diagram of relations included in the database, n - the number of relations; Schemes $\sigma_i =< R_i, F_i >$ where R_i - carrier of relation (set of attributes), and F_i - a set of functional dependencies (FD), satisfying this relation. $P = \{\rho_i, i = \overline{1,n}\}$ - A set of relations of the database.

The main stages of the method of synthesis:

1. Eliminating unnecessary attributes. Let F - initial set FD. After eliminating the unnecessary attributes from the left sides of each FD in F it will be received as a result the set F'. Attribute is redundant if its removal does not affect the closure of the FD.

2. Finding the cover. It is necessary to find a set H for F' that is possible to derive any FD from F' using the dependencies from H.

3. Split. Split H into groups, such that all the FD in each group will have the same left-hand sides.

4. Combine the equivalent keys. Let J = ∅. For each pair of groups Hi and Hj with left parts X and Y, respectively, must combine H1 and H2, if there exists a bijection X↔Y in H +. For each of these bijections add X→ Y and Y ← X to J. There is conducted a check for every attribute $A \in Y$ if X→ A is in H, you should remove it from the H. The same is done for each X→ B to H with $B \in X$.

5. Eliminate transitive dependencies. To do this, you should find $H' \subseteq H$ such that $(H' + J)^+ = (H + J)^+$ and no subset that belongs to H' should not have this feature. Then you need to add each FD belonging to J, to the appropriate group H'.

6. Build relations. For each group construct relation that consists of all the attributes that are in this group. Each set of attributes, which is in the left side of any FD in the group, is a key of relation. (Step 1 ensures that a set will not contain extra attributes). All keys have been found this way be called synthesized.

A set of constructed relations is a schema for a given set of FD.

Minimality of this method ensures that all non-redundant covers results in the same number of relations, because the number of equivalence classes synthesized key is the same for all non-redundant covers of some set FD.

In case $\Sigma^{'} \notin S$, the current schema is not in 3NF so to receive schema in 3NF most obvious option is to choose one implementation $\hat{\Sigma} \in S$ as a target. In the case where there are some implementations $\hat{\Sigma}$, the selection criterion is proposed the use of expert review, as they all are correct from the standpoint of conditions 3NF.

The proposed approach to the task of reengineering of relational database allows the synthesis of these schemes, as well as to obtain estimates for compliance with the logical scheme of the third normal form. The final structure will take into account not only the explicit relationship between the data expressed by a relational integrity constraints, but also the hidden dependence of sphere, which have been established historically in the operation of information systems.

IV. CONCLUSION

The article describes the approach to the task of reengineering of information systems based on relational databases. Highlighted the problems of reengineering of RDB, investigated the step of forming the logical database schema, which is common for problems of adaptation and refactoring.

We reviewed the sub-task checking logical schema RDB for the compliance with the 3NF, it is shown that its decision involves a number of difficulties, in particular, the need to determine the set of functional dependencies.

The proposed in the Article functional dependencies search method makes it possible to include in the process of reengineering those relationships of sphere, which were not included in the initial design stage. This ensures optimum of final logical schema obtained by the proposed method.

REFERENCES

[1] Nick Rossiter. Re-engineering relational databases:the way forward: ISWSA '11, ACM New York, NY, USA, 2011. – 17 c.

[2] J. Lemaitre , J.-L. Hainaut. Transformation-based Framework for the Evaluation and Improvement of Database Schemas / Int. Conf. on Advanced Information Systems Engineering (CAiSE), 2010.

[3] M. Talla, and R. Valverde. Data oriented and Process oriented Strategies for Legacy Information Systems Reengineering / ACEEE Int. J. on Information Technology, Vol. 02, No. 01, 2012.

[4] D. Yeh, Y. Li. Extracting entity relationship diagram from a table-based legacy database / In Proc. European Conference on Software Maintenance and Reengineering, 2005.

[5] M. Andersson. Extracting an entity-relationship schema from a relational database through reverse engineering / In proceedings of ER'94, LNCS, 1994. – C. 403-419.

[6] Nummenmaa, J., Seppi A., Thanisch P. Automating Support for Refactoring SQL Databases. Proceedings of the 16th International Conference on Information and Software Technologies. – 2010. C. 343-349.

[7] Bernstein P.A. Synthesizing Third Normal Form Relations from Functional Dependencies / ACM Transactions on Database Systems (TODS) Volume 1 Issue 4. – 1976. – C. 277 – 298.

[8] Nick Rossiter. Re-engineering relational databases: the way forward: ISWSA '11, ACM New York, NY, USA, 2011. – 17 c.

[9] Henrard, J. Data dependency elicitation in database reverse engineering: Software Maintenance and Reengineering Conference, 2001. – C. 11-19.