УДК 004.932.2:681.3.06

В.А. Радченко

ОБ ОДНОМ ПОДХОДЕ К РЕИНЖИНИРИНГУ ЛОГИЧЕСКОЙ СХЕМЫ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

Введение. Стремительное развитие информационных технологий привело к серьезным изменениям во всех этапах жизненного цикла информационных систем (ИС), в частности, этапа поддержки и сопровождения. Постоянно изменяющиеся требования в условии сжатых сроков разработки негативным образом сказываются как на качестве продукта, так и дальнейших возможностей по его развитию и обслуживанию. Нередко возникают ситуации, когда дальнейшая поддержка отдельных компонентов либо всей системы в целом становится крайне затруднительной из-за значительного усложнения или запутанности внутренней структуры, а проектирование новой системы, способной удовлетворить текущие требования, является нецелесообразным по ряду причин (экономическим, временным и т.д.). Одним из путей решения таких проблем является проведение реинжиниринга, целью которого является улучшение характеристик исходной ИС на основе предварительного анализа текущего состояния ИС и ее отдельных компонентов.

Реинжиниринг информационных систем в целом и, в частности, баз данных стали объектом пристального внимания и активной исследовательской работы. Для того чтобы расширить функциональность и улучшить производительность существующих систем, процесс реинжиниринга требует определения и понимания всех компонентов таких систем. База данных, входящая в состав, является наиболее важной составляющей ИС. На сегодняшний день реляционные базы данных (РБД) занимают доминирующее положение и используются в подавляющем большинстве существующих программных приложений и сервисов. Этим и обусловлен выбор РБД в качестве объекта исследования.

Был проведен анализ существующих публикаций с исследованиями по данной тематике. Основными направлениями развития являются: реинжиниринг логической схемы с помощью промежуточного представления, такого, как ER-модель либо собственные метамодели, и применение набора предопределенных правил для трансляции объектов модели в конструкции РБД [1,2]; реинжиниринг устаревших БД [3]; извлечение структуры как устаревших, так и реляционных БД и представление ее в виде концептуальной модели данных, в частности, ER-модели[4,5]. Также имеют место подходы к реинжинирингу РБД без использования промежуточных моделей [6].

В результате проведенного анализа стало возможным выделить классы задач реинжиниринга РБД:

1. Миграция – преобразование устаревшей БД (напр., сетевой либо иерархической) в современный аналог (напр., реляционную или объектную); перенос БД на другую платформу того же типа (напр., MySQL → Oracle);
2. Рефакторинг – улучшение характеристик без изменения функциональности;
3. Адаптация – внесение изменений согласно требованиям предметной области.

В общем случае процедура реинжиниринга включает в себя два этапа: формирование структуры РБД, отвечающей новым требованиям, и перенос данных в эту структуру.

Рассмотренные методы имеют ряд недостатков; в частности, методы, использующие промежуточное представление в виде концептуальной модели, формируют конечную структуру РБД согласно набору четко заданных правил преобразования сущностей концептуальной модели в сущности целевой РБД. То есть, качество конечной структуры зависит от количества и точности набора правил; с другой стороны, промежуточный этап в виде построения концептуальной модели не гарантирует отсутствие погрешностей, связанных с переходом на концептуальный уровень и обратно. Подход к реинжинирингу без использования промежуточного уровня представления, в частности, [6], опирается лишь на информацию об ограничениях целостности используемой РСУБД, таких как первичные и внешние ключи, и не принимает во внимание дополнительные взаимосвязи данных, обусловленные предметной областью, которые имеют место при расширении первоначальной структуры РБД.

Следовательно, постановка задачи в данном случае представляет собой следующее: разработать подход к реинжинирингу РБД, учитывающий неявные взаимосвязи данных, которые могут повлиять на результирующую структуру, а также не требующий использования промежуточного представления модели данных.

Подход к реинжинирингу РБД. Представим исходную РБД как множество $DB = \{\rho_1 ... \rho_n\}$, где ρ_i – отношение БД, $i = \overline{1,n}$ (n – количество отношений БД). Выразим ρ_i через набор $< \sigma_i, p_i >$, где σ_i – логическая схема отношения, а p_i – экземпляр отношения (множество кортежей). Логической схемой отношения будем считать $\sigma_i =< R_i, F_i >$, где R_i – носитель отношения (множество атрибутов), а F_i – множество функциональных зависимостей (ФЗ), выполняющихся на R_i.

С другой стороны, $DB =< \Sigma, P >$ где $\Sigma = \{\sigma_1 ... \sigma_n\}$ – логическая схема, а $P = \{p_1 ... p_n\}$ – экземпляр данных РБД. Также логическую схему Σ можно выразить как набор $\Sigma = \{R, F\}$, где $R = \bigcup R_i$ – общее множество атрибутов, а $F = \bigcup F_i$ – общее множество ФЗ.

В процессе нормализации при проектировании исходной структуры РБД происходит декомпозиция универсального отношения U согласно множеству ФЗ, которые выполняются на U. Целью нормализации является устранение избыточности данных, а также аномалий обновления/удаления. Как правило, достаточной считается нормальная форма Бойса-Кодда (НФБК), в которой отсутствуют какие-либо аномалии. Но ввиду того, что не все схемы можно привести к НБФК, основной принято полагать третью нормальную форму (3НФ) [7]. Обозначим через Σ' текущую логическую схему РБД, которая является результатом внесения ряда изменений согласно требованиям предметной области. Это выражается в формировании измененного множества атрибутов R' относительно исходного множества R (были добавлены новые или удалены существующие атрибуты). В процессе функционирования ИС также происходит переход от P к P', где P' – текущий экземпляр данных РБД.

В общем случае нельзя гарантировать факт нахождения Σ' в 3НФ, поскольку из того, что $R' \neq R$, следует $F' \neq F$, где F' – текущее множество ФЗ. Обоснование: пусть $R' = R \cup X$, где X – множество добавленных атрибутов, Y – множество удаленных атрибутов, F – множество ФЗ, которые соблюдаются на R. Следовательно, должно существовать множество F_X, которое состоит из ФЗ, включающих атрибуты из X в свои левые или правые части. Аналогично, множество F_Y представляет собой те ФЗ, которые перестают существовать после удаления Y из R.Таким образом, $F' = (F - F_Y) \cup F_X \neq F$.

Поскольку имеет место $\Sigma' =< R', F' >$, то одной из задач реинжиниринга является проверка текущей логической схемы Σ' на соответствие 3НФ, а также приведение к ней в случае отрицательного результата проверки.

В качестве одного из подходов для решения поставленной задачи предлагается построить 3НФ для текущей логической схемы, используя предложенный Ф.Бернштейном метод синтеза [8]. Этот метод принимает в качестве входного параметра множество ФЗ и формирует множество возможных реализаций схем $G = \{\hat{\Sigma}_1 ... \hat{\Sigma}_m\}$ в 3НФ на выходе. Проверка заключается в сопоставлении текущей схемы Σ' и каждой из полученных реализаций $\hat{\Sigma}$. Если существует $\hat{\Sigma}$ такая, что соблюдается $\Sigma' \equiv \hat{\Sigma}$, то нормализация Σ' не требуется, поскольку текущая декомпозиция удовлетворяет требованиям 3НФ. В ином случае будем считать, что Σ' частично денормализирована.

Следует отметить, что данная задача не является тривиальной. Множество F', которое является исходными данными для метода синтеза, не полностью определено ввиду того, что $F' = (F - F_Y) \cup F_X$, – при изменении множества атрибутов изменяется и множество ФЗ, соблюдающихся в РБД. Поэтому первым этапом должно стать нахождение F', справедливых для текущего состояния данных – экземпляра P'.

Пусть F_S – множество тех ФЗ, которые представляется возможным получить путем анализа ограничений целостности реляционной СУБД, таких, как первичные и внешние ключи. Обозначим с помощью F_H то множество ФЗ, которое представляет собой неявные, скрытые зависимости, о существовании которых не было известно во время исходного проектирования. Они выражаются в качестве закономерностей в данных, установившихся в процессе функционирования ИС, и могут содержать ценные знания о взаимосвязях предметной области, которые затем будут использованы в процессе синтеза целевой схемы. Таким образом, искомое множество ФЗ $F' = F_S \cup F_H \cup F_X$; нахождение F_S не рассматривается в данной работе.

Для определения множества F_H существует ряд методов выявления зависимостей из экземпляров РБД. В основном, они используются в задачах интеллектуального анализа данных (Data Mining) и позволяют находить приблизительные ФЗ (ПФЗ). Отличие от классических «строгих» ФЗ состоит в том, что допускается существование ФЗ, даже если в РБД присутствуют строки, нарушающие корректность ФЗ. Это допущение опирается на предположение о том, что в процессе функционирования РБД возможно внесение «ошибочных» строк. Под «ошибочной» строкой следует понимать такой кортеж, который не противоречит существующим ограничениям целостности, но нарушает соблюдавшуюся до этого момента неявную ФЗ.

Одним из таких методов является Tane [9] и его модификации, послуживший основой для других подобных решений. Была предложена собственная модификация данного метода, позволяющая выявлять только строгие ФЗ [10]. В качестве входных данных необходимо предоставить экземпляр P'; результатом работы метода является искомое множество F_H.Тем не менее, корректность результата не гарантируется для данных, содержащих пустые значения (NULL). Поддержка пустых значений в задаче выявления ФЗ не

рассматривается в данной работе.

Промежуточный результат F_H нуждается в дополнительной обработке, поскольку F_H содержит все ФЗ, справедливые для P' на момент проведения анализа. Существует вероятность того, что для текущего состояния P' будут выявлены случайные ФЗ – корректные с точки зрения данных, но бессмысленные по отношению к рассматриваемой предметной области. Эта задача более подробно рассмотрена в [11].

Полученное множество ФЗ, представляющее собой наиболее полную информацию о взаимосвязях внутри экземпляра данных рассматриваемой РБД, является входными данными для метода синтеза для получения множества возможных реализаций схем G. В случае, если $\Sigma' \notin G$, текущая схема не находится в 3НФ, и для получения схемы в 3НФ наиболее очевидным вариантом является выбор одной из реализаций $\hat{\Sigma} \in G$ в качестве целевой. В случае, когда реализаций $\hat{\Sigma}$ несколько, критерием выбора предлагается использование экспертной оценки, поскольку они все являются корректными с позиции условий 3НФ.

Использование подхода к реинжинирингу, предложенного в работе, позволяет осуществить проверку на соблюдение 3НФ в некоторой РБД и получить целевую логическую схему в 3НФ. Такая схема учитывает не только явные взаимосвязи между данными, выраженные с помощью ограничений целостности реляционной СУБД, но и скрытые, неявные зависимости предметной области, которые установились в процессе функционирования ИС.

Выводы. В работе рассмотрено направление реинжиниринга информационных систем и реляционных баз данных в частности, актуальность этого направления. Выделены классы задач реинжиниринга РБД; исследован этап формирования целевой логической схемы, которая является общей для задач адаптации и рефакторинга. Рассмотрена подзадача проверки соответствия логической схемы РБД 3НФ в рамках данного этапа с помощью метода синтеза; показано, что ее решение сопряжено с рядом трудностей, в частности, необходимостью нахождения множества функциональных зависимостей, выполняющихся на текущем экземпляре данных некоторой РБД. Предложен подход для нахождения такого множества ФЗ применением метода выявления ФЗ из экземпляра данных РБД. В качестве направления для дальнейших исследований можно выделить реализацию поддержки пустых значений на этапе выявления ФЗ, а также вопросы переноса данных без потерь из исходной структуры РБД в целевую, полученную в результате применения методов реинжиниринга.

ЛИТЕРАТУРА:

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. Гарсиа-Молина Г., Ульман Д., Уидом Д. Системы баз данных. Полный курс. : Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – С. 135.
8. Bernstein P.A. Synthesizing Third Normal Form Relations from Functional Dependencies / ACM Transactions on Database Systems (TODS) Volume 1 Issue 4. – 1976. – С. 277 – 298.
9. Y. Huhtala, J. Kärkkäinen, P. Porkka. Tane: An Efficient Algorithm For Discovering Functional and Approximate Dependencies / The Computer Journal 42 (2). – 1999. – С. 100-111.
10. В.А. Радченко. Модификация метода выявления функциональных зависимостей в реляционных базах данных / Інформаційні технології в навігації і управлінні: стан та перспективи розвитку. Матеріали другої міжнародної науково-технічної конференції. – К.: ДП «ЦНДІ НіУ», 2011. – 52 с.
11. В.А. Радченко, С.С. Танянский. Выявление скрытых зависимостей между данными в задачах реинжиниринга информационных систем / Системи обробки інформації. Випуск 3(101) – Х.: ХУПС, 2012. – 268 с.

РАДЧЕНКО Вячеслав Алексеевич – аспирант кафедры электронных вычислительных машин.

Научные интересы: проектирование и поддержка распределенных информационных систем и баз данных