

Анализ проблем разработки методов в рамках методики проектирования реляционных баз данных с использованием существующей информации табличного вида

© А.В. Брешенков

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

К числу основных компонентов любой методики относятся взаимосвязанные методы, ориентированные на успешное выполнение какой-либо деятельности. В статье рассмотрены проблемы разработки взаимосвязанных методов, ориентированных на решение основных задач проектирования реляционных баз данных, представленных в табличном виде. К этим проблемам относятся: приведение заполненных таблиц к реляционному виду; нормализация заполненных таблиц; назначение ключевых полей в заполненных таблицах; формирование связей между заполненными таблицами; объединение и разбиение заполненных таблиц; анализ применимости современных теоретических и практических разработок.

Ключевые слова: реляционные базы данных, нормализация таблиц, назначение ключевых полей, формирование связей между таблицами.

Проблема приведения заполненных таблиц к реляционному виду. Для того чтобы обеспечить возможность преобразования информации табличного вида (ИТВ) в таблицы реляционных баз данных (БД), прежде всего необходимо привести их к реляционному представлению данных. В соответствии с [1–3] реляционная таблица представляет собой двумерный массив и обладает следующими свойствами.

• *Каждый элемент таблицы — один элемент данных.*

Отношение

$$R = (A_1, \dots, A_i, \dots, A_k), \quad i = \overline{1, k},$$
$$A_i = \{e_{i1}, \dots, e_{ij}, \dots, e_{in}\}, \quad j = \overline{1, n},$$

где A_i — множество атрибутов отношения; e_{ij} — j -й элемент атрибута A_i ; k — степень отношения; n — мощность отношения.

Необходимо обеспечить выполнение условия

$$\left(|e_{ij}| = 0\right) \vee \left(|e_{ij}| = 1\right).$$

• *Все столбцы таблицы однородные, т. е. все элементы в столбце имеют одинаковый тип.*

Для всех атрибутов необходимо обеспечить выполнение условия

$$T(e_{i1}) = \dots = T(e_{ij}) = \dots = T(e_{in}), \quad j = \overline{1, n},$$

где $T(e_{ij})$ — тип j -го элемента атрибута A_i .

- *Каждый столбец имеет уникальное имя:*

$$A_1 \neq \dots \neq A_i \neq \dots \neq A_k, \quad i = \overline{1, k}.$$

- *Одинаковые строки в таблице отсутствуют:*

$$S_1 \neq \dots \neq S_i \neq \dots \neq S_n, \quad i = \overline{1, n},$$

$$S_i = (e_{i1}, \dots, e_{ij}, \dots, e_{ik}), \quad j = \overline{1, k}.$$

- *Порядок следования столбцов и строк может быть произвольным.*

В общем случае ИТВ этими свойствами не обладает. Более того, заголовки столбцов исходных таблиц и элементы данных могут располагаться на нескольких строках, в заголовках столбцов могут присутствовать запрещенные с точки зрения БД символы («», «!» и др.). Все это недопустимо для реляционных таблиц. В связи с этим первым шагом преобразования в файлы РБД является *генерация таблиц*, обладающих перечисленными свойствами, на основе ИТВ.

Для решения этой проблемы сначала нужно разработать формальную модель ИТВ и формальную модель реляционных таблиц, а затем на основе их использования разработать метод, обеспечивающий выполнение сформулированных условий и преобразование ИТВ к реляционному виду.

Проблема нормализации заполненных таблиц. После того как информация приведена к реляционному виду, нет никакой гарантии, что получены самые удачные таблицы с точки зрения их представления и использования в БД. Сформулируем постановку задач приведения заполненных таблиц к нормальным формам.

Постановка задачи приведения отношений к первой нормальной форме. Дано отношение

$$R = (A_1, \dots, A_i, \dots, A_k), \quad i = \overline{1, k},$$

причем

$$A_i = (A_{i1}, \dots, A_{ij}, \dots, A_{in}), \quad j = \overline{1, n},$$

$$A_{ij} = (A_{ij1}, \dots, A_{ijt}, \dots, A_{ijp}), \quad t = \overline{1, p},$$

где A_i — множество атрибутов отношения; A_{ij} — j -й подзаголовок первого уровня i -го атрибута; A_{ijt} — t -й подзаголовок второго уровня j -го подзаголовка первого уровня i -го атрибута.

Для приведения отношений к первой нормальной форме необходимо обеспечить выполнение требования

$$|A_i| = 1 \text{ и } |A_{ij}| = 1.$$

Постановка задачи приведения отношений ко второй нормальной форме. Дано отношение

$$R = (K_1, \dots, K_m, \dots, K_n, A_1, \dots, A_i, \dots, A_j, \dots, A_k), \quad m = \overline{1, n}, \quad i = \overline{1, k},$$

$$S_1 = (K_{11}, \dots, K_{m_1}, \dots, K_{n_1}, A_{11}, \dots, A_{i_1}, \dots, A_{j_1}, \dots, A_{k_1}),$$

$$S_t = (K_{1t}, \dots, K_{m_t}, \dots, K_{n_t}, A_{1t}, \dots, A_{it}, \dots, A_{jt}, \dots, A_{kt}),$$

$$S_p = (K_{1p}, \dots, K_{mp}, \dots, K_{np}, A_{1p}, \dots, A_{ip}, \dots, A_{jp}, \dots, A_{kp}),$$

где $(K_1, \dots, K_m, \dots, K_n)$ — атрибуты, составляющие сложный ключ;

S_1, S_t, S_p — соответственно 1-я, t -я и p -я запись отношения R .

Для приведения отношений ко второй нормальной форме необходимо обеспечить истинность выражения

$$\neg((K_{m_1}, A_{i_1}, \dots, A_{j_1}) = \dots = (K_{m_t}, A_{i_t}, \dots, A_{j_t}) = \dots = (K_{m_p}, A_{i_p}, \dots, A_{j_p})),$$

$$m = \overline{1, n}, \quad i = \overline{1, k}, \quad j = \overline{1, k}, \quad i \neq j.$$

Постановка задачи приведения отношений к третьей нормальной форме. Дано отношение

$$R = (K_1, \dots, K_m, \dots, K_n, A_1, \dots, A_i, \dots, A_j, \dots, A_k), \quad m = \overline{1, n}, \quad i = \overline{1, k},$$

при этом

$$S_1 = (K_{11}, \dots, K_{m_1}, \dots, K_{n_1}, A_{11}, \dots, A_{i_1}, \dots, A_{j_1}, \dots, A_{k_1}),$$

$$S_t = (K_{1t}, \dots, K_{m_t}, \dots, K_{n_t}, A_{1t}, \dots, A_{it}, \dots, A_{jt}, \dots, A_{kt}),$$

$$S_p = (K_{1p}, \dots, K_{mp}, \dots, K_{np}, A_{1p}, \dots, A_{ip}, \dots, A_{jp}, \dots, A_{kp}),$$

где $(K_1, \dots, K_m, \dots, K_n)$ — атрибуты, составляющие сложный ключ;

S_1, S_t, S_p — соответственно 1-я, t -я и p -я запись отношения R .

Для приведения отношений к третьей нормальной форме необходимо обеспечить истинность выражения

$$\neg\left((A_{i_1}, \dots, A_{j_1}) = \dots = (A_{i_t}, \dots, A_{j_t}) = \dots = (A_{i_p}, \dots, A_{j_p})\right),$$

$$i = \overline{1, k}, \quad j = \overline{1, k}, \quad i \neq j.$$

Постановка задачи приведения отношений к четвертой нормальной форме. Дано отношение

$$R = (A_1, \dots, A_{i,\dots}, A_j, \dots, A_k, \dots, A_m, \dots, A_p), \quad i = \overline{1, p},$$

при этом

$$S_1 = a_{1_1}, \dots, a_{i_1}, \dots, a_{j_1}, \dots, a_{k_1}, \dots, a_{m_1}, \dots, a_{p_1},$$

$$S_f = a_{1_f}, \dots, a_{i_f}, \dots, a_{j_f}, \dots, a_{k_f}, \dots, a_{m_f}, \dots, a_{p_f},$$

$$S_z = a_{1_z}, \dots, a_{i_z}, \dots, a_{j_z}, \dots, a_{k_z}, \dots, a_{m_z}, \dots, a_{p_z},$$

$$S_q = a_{1_q}, \dots, a_{i_q}, \dots, a_{j_q}, \dots, a_{k_q}, \dots, a_{m_q}, \dots, a_{p_q},$$

$$M1 = \left\{ (a_{i_1}, \dots, a_{j_1}), \dots, (a_{i_f}, \dots, a_{j_f}), \dots, (a_{i_z}, \dots, a_{j_z}), \dots, (a_{i_q}, \dots, a_{j_q}) \right\},$$

$$M2 = \left\{ (a_{k_1}, \dots, a_{m_1}), \dots, (a_{k_f}, \dots, a_{m_f}), \dots, (a_{k_z}, \dots, a_{m_z}), \dots, (a_{k_q}, \dots, a_{m_q}) \right\},$$

$$M1 \neq M2,$$

$$M1' = \left\{ (a_{i_f}, \dots, a_{j_f}), \dots, (a_{i_z}, \dots, a_{j_z}) \right\}, \quad M1' \subseteq M1,$$

$$M2' = \left\{ (a_{k_f}, \dots, a_{m_f}), \dots, (a_{k_z}, \dots, a_{m_z}) \right\}, \quad M2' \subseteq M2,$$

где S_1 , S_f , S_z и S_q — кортеж значений соответственно 1-й, f -й, z -й и q -й записи; $M1$ и $M2$ — два разных множества сочетаний значений одноименных кортежей.

Для приведения отношений к четвертой нормальной форме необходимо обеспечить истинность выражения

$$\neg\left(\left((a_{i_f}, \dots, a_{j_f}) = \dots = (a_{k_f}, \dots, a_{m_f})\right) \wedge \right. \\ \left. \wedge \left((a_{i_z}, \dots, a_{j_z}) = \dots = (a_{k_z}, \dots, a_{m_z})\right)\right).$$

Если данные из заполненных реляционных таблиц добавляются к таблицам существующей, нормализованной БД, а структура добавляемых данных совпадает со структурой таблиц БД, то проблемы нормализации не существует, так как предполагается, что БД нормализована и изменениям не подлежит.

Если данные из заполненных реляционных таблиц добавляются к таблицам существующей, нормализованной БД, а структура добавляе-

мых данных не совпадает со структурой таблиц БД, то возникает проблема нормализации и преобразования структур добавляемых данных.

Если создается новая БД на основе использования ИТВ или к существующей БД добавляются новые заполненные реляционные таблицы, то проблема нормализации актуальна.

В связи с этим вторым этапом преобразования ИТВ в файлы РБД должен быть *этап нормализации реляционных таблиц*.

В соответствии с [3] в РБД проблема избыточности и противоречивости данных сводится к минимуму в том случае, если таблицы БД удовлетворяют нормальным формам.

Пусть дано отношение

$$R = (K_1, \dots, K_m, \dots, K_n, A_1, \dots, A_i, \dots, A_j, \dots, A_k), \quad m = \overline{1, n}, \quad i = \overline{1, k},$$

где K_m — атрибут, входящий в сложный ключ; A_i — неключевой атрибут.

Необходимо обеспечить истинность следующих высказываний:

1. $(|A_{i_i}| = \emptyset) \wedge (|A_{i_j}| = \emptyset)$.
2. $\neg((K_m \rightarrow A_i) \vee \dots \vee (K_m \rightarrow A_j))$.
3. $\neg((A_i \rightarrow A_j) \vee \dots \vee (A_n \rightarrow A_m))$.
4. $\neg((A_i \rightarrow A_j) \wedge \dots \wedge (A_n \rightarrow A_m))$.

Проблема назначения ключевых полей в заполненных таблицах. В работах, посвященных теоретическим основам построения БД, в частности в [1–3], дается определение ключевых полей, обосновывается их необходимость, формулируются требования к первичным ключам и определяются свойства внешних ключей. Эти требования сравнительно легко реализуются на этапах инфологического и датологического проектирования БД до момента их заполнения данными. В рассматриваемом же случае данные уже существуют сами по себе, представляют ценность и должны быть наиболее эффективным образом задействованы в РБД [3]. В связи с этим возникает проблема назначения ключевых полей на основе анализа данных, представленных в ИТВ. Она распадается на две подзадачи — назначение первичных ключей и назначение внешних ключей.

Полностью автоматизировать процесс назначения первичного ключа невозможно. При анализе ИТВ на предмет выбора атрибута или совокупности атрибутов, удовлетворяющих требованиям, предъявляемым к первичным ключам, необходимо учитывать не только эти требования, но и семантику данных. В этом случае не обойтись без вмешательства разработчика БД, которому необходимо предоставить соответствующие методы и средства.

При назначении ключевых полей в реляционных таблицах в процессе преобразования ИТВ в РБД следует учитывать, что в отдельных таблицах после выполнения этапа нормализации ключевые поля уже назначены.

Таким образом, постановка задачи имеет следующий вид. Дано отношение

$$R = (A_1, \dots, A_i, \dots, A_k), \quad i = \overline{1, k},$$

причем

$$A_i = \{e_{i1}, \dots, e_{ij}, \dots, e_{in}\}, \quad j = \overline{1, n},$$

где A_i — множество атрибутов отношения; e_{ij} — j -й элемент атрибута A_i ; k — степень отношения; n — мощность отношения.

Необходимо найти такой атрибут A_i , чтобы обеспечилась истинность выражения

$$e_{i1} \neq \dots \neq e_{ij} \neq \dots \neq e_{in}.$$

При этом следует учитывать, что таких атрибутов может быть несколько или ни одного. Кроме того, допустимо сочетание нескольких атрибутов, составляющих ключ.

Для решения проблемы назначения ключевых полей в ИТВ необходимо разработать метод, учитывающий сформулированные особенности.

Проблема формирования связей между заполненными таблицами. Одним из начальных этапов традиционного процесса проектирования РБД является построение инфологической модели предметной области. При этом разрабатываются модели сущность-связь, моделируются локальные представления, выполняются объединения локальных представлений [2, 3]. Таким образом, уже на ранних этапах традиционного проектирования РБД формируются связи между объектами проектируемой БД. На следующем этапе проектирования БД — датологическом проектировании — в качестве модели данных выступают сами данные. При выполнении этого этапа средствами инструментальной системы управления базами данных (СУБД) объектам БД ставятся в соответствие реальные таблицы инструментальной СУБД, а связям между объектами — связи между таблицами.

На обоих этапах проектирования осуществляются мероприятия по нормализации, которые также приводят к формированию связей между таблицами. Эти связи в итоге отображаются в инструментальной СУБД на схеме данных. После выполнения этапов инфологического и датологического проектирования новые связи между таблицами стро-

ятся редко, необходимость их формирования обычно связана с концептуальными ошибками в проекте БД или с ее расширением и модификацией. Таким образом, при проектировании БД с нуля в процессе ее эксплуатации данные заносятся в таблицы, связанные друг с другом, т. е. решаются проблемы целостности сущностей, целостности согласования, существенно упрощается процедура формирования запросов на выборку данных, размещенных в нескольких таблицах.

В случае если РБД создается на основе ИТВ, инфологический и датологический этапы остаются за рамками процесса проектирования. Данные табличного вида уже сформированы, и о связях между ними не может быть и речи. Несмотря на это, состав ИТВ нередко позволяет сделать вывод о наличии между ними логических связей. Выявление этих связей и отражение их на схеме данных позволит существенно улучшить качественные характеристики проектируемой БД.

Даны отношения A и B . Необходимым условием наличия между ними связи является условие

$$\forall b_{pj} \in Z(B_j) \exists a_{ri} \in Z(A_i),$$

такая, что

$$b_{pj} = a_{ri}, \quad p = \overline{1, q}, \quad j = \overline{1, k}, \quad r = \overline{1, m}, \quad i = \overline{1, n},$$

где q и k — степень и мощность отношения B ; m и n — степень и мощность отношения A ; $Z(B_j)$ — значения атрибута B_j ; $Z(A_i)$ — значения атрибута A_i .

Для решения проблемы выявления связей между ИТВ необходимо разработать метод, позволяющий не только выявлять связи, но и идентифицировать их типы.

Проблема объединения и разбиения заполненных таблиц. Если удастся преобразовать ИТВ в реляционные таблицы, это не всегда означает, что их можно непосредственно преобразовывать в формат БД. Может потребоваться объединение нескольких таблиц в одну. Это обусловлено различными причинами, в частности:

- информация, которую нужно включить в БД, поступает от нескольких контрагентов;
- в существующей БД, в которую вводятся новые данные, используется таблица, объединяющая информацию из нескольких включаемых таблиц.

С другой стороны, может возникнуть обратная проблема, которая связана с преобразованием одной таблицы в несколько таблиц. Это может произойти в следующих случаях:

- данные из таблицы, включаемые в существующую БД, уже представлены в ней несколькими таблицами;
- таблица в БД не удовлетворяет условиям нормализации.

Одним из самых простых является случай, когда несколько одинаковых по сути таблиц требуется объединить в одну. Решение задачи может свестись к выбору исходной таблицы и добавлению в нее записей с данными из других таблиц. Однако, если столбцы исходных таблиц не расположены в одинаковой последовательности, исходные таблицы необходимо предварительно преобразовать к нужному виду. Кроме того, в соответствии с требованиями к таблицам БД их записи должны быть уникальны, т. е. в них должны отличаться значения хотя бы одного поля. Это требование далеко не всегда выполняется, например, когда в исходных таблицах записи отличаются только порядковыми номерами, а эти номера в различных таблицах могут совпадать. В связи с этим может возникнуть необходимость добавления в таблицу специального ключевого поля, значение которого должно быть уникальным для каждой записи.

Более сложным является случай, когда требуется объединить несколько таблиц, дополняющих по данным друг друга. При этом как количество, так и смысловая нагрузка столбцов исходных таблиц могут не совпадать. Тогда кроме проблем объединения возникают новые проблемы. Чтобы решить задачу объединения такого характера, необходимо выявить поля в исходных таблицах, по которым можно провести объединение. Эти поля, называемые ключевыми, должны присутствовать во всех таблицах и иметь одинаковую смысловую нагрузку. Затем необходимо формировать записи целевой таблицы из записей исходных таблиц. При этом критериями формирования новых записей являются одинаковые значения ключевых полей в исходных таблицах. Кроме того, в процессе формирования записи исключаются заголовки и данные одинаковых столбцов.

В случае необходимости разбиения одной таблицы на несколько следует руководствоваться критериями эффективности построения новой БД на основе использования исходных таблиц. В частности, следует руководствоваться требованиями нормализации таблиц. Если БД уже существует, то руководствуются составом таблиц БД, их структурой и связями между таблицами. При разбиении таблиц необходимо иметь в виду, что пользователю БД может потребоваться обработка информации из нескольких таблиц одновременно. Поэтому между результирующими таблицами должны быть обеспечены связи, например, через механизм первичных и внешних ключей.

Необходимо обеспечить объединение записей отношений:

$$S = SA \cup SB,$$

где $SA = \{SA_1, \dots, SA_m, \dots, SA_n\}$, $m = \overline{1, n}$ (n — число записей отношения A ; SA_m — m -я запись A); $SB = \{SB_1, \dots, SB_p, \dots, SB_q\}$, $p = \overline{1, q}$ (q — число записей отношения B ; SB_p — p -я запись B).

При этом следует учитывать, что существуют совместимые и частично совместимые по объединению отношения.

Для качественного, бездефектного решения задач объединения и разбиения реляционных таблиц необходимо разработать формальные модели исходных и целевых таблиц, а также формальную методику преобразования таблиц.

Для решения проблемы объединения и разбиения ИТВ требуется разработать метод, позволяющий учитывать сформулированные особенности процесса преобразования информации.

Анализ применимости современных теоретических разработок. Сложившаяся на данный момент методология проектирования реляционных БД освещена практически во всех источниках по проектированию РБД, в частности в [1–3]. Следуя ей, разработчик может выполнить этапы проектирования от постановки задачи, инфологического и датологического проектирования до физической реализации и создать высокоэффективную БД, удовлетворяющую современным требованиям. Кроме того, он может построить реляционные таблицы, удовлетворяющие всем нормальным формам. Рекомендации по проектированию БД помогают разработчику назначить ключевые и индексные поля, сформировать связи между таблицами, обеспечить безопасность данных и осуществить много полезных мероприятий, ориентированных на разработку высококачественного программного продукта. Другими словами, сегодня разработчик вооружен мощной теорией построения БД и множеством инструментальных средств, ориентированных на создание БД различного назначения в рамках этой теории. В связи с этим возникает вопрос: насколько применимы существующая теория и инструментальные средства для проектирования РБД на основе ИТВ с заполненными таблицами?

Вся проблема в том, что в рассматриваемом случае данные уже есть, таблицы сформированы и, конечно, при их создании никто не руководствовался теоретическими аспектами построения БД. Чаще всего эти аспекты были неактуальны, разработчик решал другие задачи. Таким образом, при формировании ИТВ выпали все этапы проектирования БД. Выполнить же условия, характерные для этапов построения БД в процессе преобразования ИТВ в РБД, о которых идет речь, не представляется возможным. Если их выполнять, необходимо спроектировать новую РБД, а затем вручную перенести ИТВ в таб-

лицы БД. Это, может быть, и оправданно, если данных немного. Здесь же речь идет о ситуации, при которой в табличном виде накоплены значительные объемы информации. Таким образом, теоретические положения поэтапной разработки БД в рассматриваемом случае неприемлемы, приходится исходить из таких представлений ИТВ, которые уже сформированы.

Могут ли быть полезными другие положения теории построения реляционных БД для разработки математических и программных средств, ориентированных на преобразование ИТВ в таблицы РБД? Вероятнее всего, да.

Требования к реляционным таблицам сформулированы, и можно описать целевые функции, позволяющие осуществить автоматизированное преобразование существующей ИТВ в реляционные таблицы.

Условия нормализации таблиц сформулированы и, в отличие от традиционного подхода к проектированию, представляется возможным формализовать процесс нормализации заполненных реляционных таблиц.

Рекомендации к ключевым и индексным полям существуют, и их можно использовать для автоматизированного назначения соответствующих полей в заполненных таблицах, что невозможно при традиционном подходе.

Следует обратить внимание, что традиционная теория позволяет сформулировать проектные задачи, но не позволяет формализовать проектные процедуры, ориентированные на решение этих задач.

Проанализировав доступные источники в области теории и практики проектирования РБД, можно сделать вывод о том, что комплексной теории проектирования РБД на основе ИТВ не существует. Известны лишь отдельные практические разработки, позволяющие решать только частные задачи преобразования ИТВ с заполненными таблицами в РБД.

Анализ применимости современных практических разработок. В системе Oracle предусмотрено специализированное средство для перемещения данных — SQL*Loader [4]. В нем в процессе перемещения данных формируются файлы отвергнутых и некорректных записей, т. е. записей, которые по каким-либо причинам не могут быть включены в БД. С одной стороны, это свидетельствует о том, что проблема преобразования ИТВ в файлы БД актуальна (коль скоро ею занимается самая крупная в мире корпорация в области БД), а с другой стороны, эта проблема при использовании данной утилиты не всегда имеет корректное решение.

Средства Microsoft SQL Server [5, 6] обеспечивают импорт данных, представленных в различных форматах. Однако эти данные должны быть представлены в виде реляционных таблиц, проблемы

нормализации заполненных таблиц и формирования связей между ними не снимаются.

Автоматизированные средства нормализации таблиц на основе анализа содержащихся в них данных уже существуют, в частности, в СУБД Microsoft Access [7, 8]. Проблема в том, что далеко не всегда для разработчика БД очевидны группировки полей, которые осуществляют автоматизированные средства. В связи с этим возможны ситуации, когда использование средств анализа не улучшит, а ухудшит положение вещей. Кроме того, в средствах анализа системы Access обеспечивается, и то не всегда, приведение таблиц только ко второй и третьей нормальным формам.

Средства экспорта-импорта большинства инструментальных СУБД и другие механизмы обмена данными (например, ODBC) до определенной степени реализуют преобразование ИТВ в файлы РБД. Однако эти средства не снимают проблем преобразования ИТВ к реляционному виду, нормализации заполненных таблиц, назначения ключевых полей в ИТВ, формирования связей между таблицами.

Обобщая сказанное, можно сделать следующие выводы:

1. Отдельные положения современной теории построения реляционных БД могут быть использованы при разработке методов и средств преобразования ИТВ в таблицы РБД. Однако необходимо разработать новые методы и методики их использования, учитывающие преимущества наличия реальных данных, а также их специфику.

2. Существующие практические разработки могут быть использованы для решения отдельных задач преобразования небольшой по объему ИТВ в РБД. Однако необходимы программные средства, позволяющие решать задачи проектирования РБД на основе использования ИТВ любых объемов.

ЛИТЕРАТУРА

- [1] Дейт К.Дж. *Введение в системы баз данных*. Москва, Наука, 1980, 464 с.
- [2] Дейт К.Дж. *Введение в системы баз данных*. 6-е изд. Киев, Диалектика, 1998, 784 с.
- [3] Дейт К.Дж. *Введение в системы баз данных*. 7-е изд. Москва, Вильямс, 2001, 1072 с.
- [4] *Энциклопедия пользователя. Oracle 8*. Киев, ДиаСофт, 1999, 864 с.
- [5] Тихомиров Ю.В. *Microsoft SQL Server 7.0*. Санкт-Петербург, БХВ-Петербург, 1999, 720 с.
- [6] Тихомиров Ю.В. *Microsoft SQL Server 7.0: разработка приложений*. Санкт-Петербург, БХВ-Петербург, 1999, 352 с.
- [7] Литвин П., Гетц К., Гунделой М. *Разработка настольных приложений в Access 2002. Для профессионалов*. Санкт-Петербург, Питер, 2002, 1008 с.
- [8] Литвин П., Гетц К., Гилберт М. *Access 2000. Руководство разработчика. Корпоративные приложения*. Т. 2. Киев, BHV, 2001, 912 с.

Ссылку на эту статью просим оформлять следующим образом:

Брешенков А.В. Анализ проблем разработки методов в рамках методики проектирования реляционных баз данных с использованием существующей информации табличного вида. *Инженерный журнал: наука и инновации*, 2013, вып. 11. URL: <http://engjournal.ru/catalog/it/hidden/1012.html>

Брешенков Александр Владимирович родился в 1955 г., окончил МВТУ им. Н.Э. Баумана в 1982 г. Д-р техн. наук, профессор кафедры «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана. Автор 130 научных работ в области САПР ЭВМ и баз данных. e-mail: breshenkov@rambler.ru