ТРУДЫ ИНСТИТУТА МАТЕМАТИКИ И МЕХАНИКИ УрО РАН

Том 22 № 3 2016

УДК 004.652.6

ОБЛАСТИ ОПРЕДЕЛЕНИЯ ФУНКЦИОНАЛЬНЫХ ЗАВИСИМОСТЕЙ В БАЗАХ ДАННЫХ

С. В. Зыкин

В статье рассматривается один из вариантов решения проблемы проектирования схемы базы данных, в которой допускается наличие неопределенных значений. Эта актуальная проблема до сих пор не имеет удовлетворительного решения. Новая формальная теория должна быть обобщением классической теории: удаление из рассмотрения неопределенных значений должно обобщенную теорию сводить к классической теории. Существующие теории не удовлетворяют этому принципу: меняется система аксиом, вводятся новые операторы реляционной алгебры, не реализуемые средствами SQL, и т. д. В этой статье предлагается оригинальное решение этой проблемы, основанное на областях определения зависимостей (доменах).

Ключевые слова: база данных, функциональные зависимости, аксиоматика, область определения.

S. V. Zykin. Domains of functional dependences in databases.

We consider a solution to the problem of designing a database schema allowing for the presence of null values. This urgent problem still has no satisfactory solution. A new formal theory should generalize the classical theory: the removal of null values should reduce the generalized theory to the classical theory. The existing theories do not satisfy this principle: the system of axioms is changed, new operators of relational algebra that cannot be implemented in the SQL are introduced, etc. We propose an original solution to this problem based on the dependence domain.

Keywords: database, functional dependences, axiomatics, domain.

MSC: 68P15

DOI: 10.21538/0134-4889-2016-22-3-117-129

Введение

Основой проектирования структуры реляционной базы данных (БД) являются зависимости, которым подчиняются все данные в прикладной области. В настоящее время исследованы и используются в проектировании схемы БД функциональные зависимости, многозначные зависимости, зависимости соединения и зависимости включения. Исторически первыми были исследованы функциональные зависимости (ФЗ). Они впоследствии явились основой формальной теории проектирования схем БД [1; 2], эту теорию в данной статье мы будем называть классической. Согласно этой теории проектирование реляционной схемы БД начинается с универсального реляционного отношения R, заданного на множестве атрибутов $U = \{A_1, A_2, \ldots, A_n\}$. Предполагается, что отношение R содержит только те кортежи t, значения атрибутов в которых удовлетворяют свойствам реальных данных. Основные свойства данных задаются функциональными зависимостями.

О п р е д е л е н и е 1. Пусть X и Y — некоторые подмножества из множества атрибутов U. Будем говорить, что X функционально определяет Y, и записывать $X \to Y$, если в любой реализации отношения R не могут присутствовать два кортежа $t, u \in R$, такие что $t[X] = u[X]$ и $t[Y] \neq u[Y]$.

В определении 1 $t[X]$ — значения атрибутов множества X в кортеже t, равенство кортежей $t[X] = u[X]$ означает совпадение значений одноименных атрибутов, а неравенство $t[Y] \neq u[Y]$ означает неравенство значений хотя бы для одного атрибута.

Функциональные зависимости используются для формирования отношений и определения первичных ключей [1; 2]. Первичные ключи позволяют предотвратить дублирование информации в БД.

О п р е д е л е н и е 2. Множество атрибутов X называется первичным ключом отношения R, если существует функциональная зависимость $X \to A_1, A_2, \ldots, A_n$ и не существует множества атрибутов Y, где $Y \subset X$ и $Y \to A_1, A_2, \ldots, A_n$.

При проектировании БД должно быть получено множество отношений $\{R_1, R_2, \ldots, R_k\}$, называемое декомпозицией R. Каждое из отношений должно удовлетворять ряду свойств, которые обсуждаются в данной работе. В частности, в отношениях R_i задаются собственные первичные ключи, которые удовлетворяют определению 2 на множестве атрибутов R_i. В данной работе первичные ключи выделяются подчеркиванием соответствующего ключу множества атрибутов.

Для манипулирования отношениями была разработана реляционная алгебра [1; 2], послужившая основой для непроцедурного языка запросов SQL (structured query language). Таким образом, сложилась технология создания и эксплуатации реляционных БД, которая занимает доминирующее положение среди других информационных технологий. Однако с самого начала практического использования реляционных БД появилась проблема неопределенных значений, имеющих метку $Null$ в языке SQL. В настоящее время эта метка присваивается любым неопределенным значениям. При выполнении логических операций с неопределенными значениями в командах SQL результатом является $unknown$. Далее будет показано, что это не всегда правильный результат.

В статье [3] делается вывод, что интерпретация неопределенностей сводится к двум видам: 1) значение, возможно, существует, но оно не известно; 2) значение не существует. При этом, значение $Null$ не может быть отождествлено с каким-либо другим значением, в том числе само с собой. В [3] предложено обобщение реляционной алгебры на случай наличия значения ni, обозначающее отсутствие значения. Однако это предложение не реализовано в стандартах языков запросов к БД. Кроме того, в статье [3] рассматривается пример, на котором можно обнаружить специфическую проблему значения $Null$: задано отношение R (№ сотрудника, ФИО сотрудника, № руководителя). В отношении подчеркнут первичный ключ. Вывод можно сделать такой, что для всех сотрудников, кроме одного, № руководителя будет определен либо неизвестен, а для одного сотрудника (директора) это значение будет несуществующим. Рассуждая формально строго, нет зависимостей, на основании которых сформировано отношение R. В нем присутствуют аномалии дополнения и модификации: при смене начальника структурного подразделения необходимо менять несколько записей. Кроме того, атрибуты первый и третий в отношении R являются синонимами. То есть пример является надуманным; при устранении синонимов и дополнениями атрибутов “Должность сотрудника” и “Место работы” будет получена корректная схема БД на основании ФЗ: R^* (№ сотрудника, ФИО сотрудника, Должность сотрудника, Место работы); отношение R становится вычисляемым. Проблема смешения неопределенностей не возникает. Следовательно, решение семантических проблем на начальном этапе проектирования позволит избавиться от смешения неопределенностей.

Серьезный вклад в исследования неопределенностей был сделан в работах [4; 5]. В развитие этого направления в работе [6] сформулирована система аксиом совместно для сильных и слабых ФЗ, для которой рассматривается полнота и надежность. Но полученная система аксиом сильно отличается от аксиом классической теории, следовательно, полученная теория не является обобщением классической: если удалить из рассмотрения значение $Null$, то не получим аксиомы классической теории. Авторы [7] существенно продвинулись в исследовании свойств функциональных и многозначных зависимостей в присутствии неопределенностей. Получена система из восьми аксиом, также как и в классической теории. Однако аксиомами стали правила (теоремы классической теории), что не позволяет сказать о полученном обобщении. Основная причина — допущение наличия неопределенности в левой части ФЗ и допущение отождествления двух неопределенных значений и, как следствие, отсутствие транзитивности.

С точки зрения приложений левая часть ФЗ становится ключом отношения и идентификатором объектов. Наличие неопределенности в неключевых атрибутах — это неопределенность характеристики объекта, тогда как наличие неопределенности в ключевых атрибутах — это неопределенность объекта, что отвергается в существующих технологиях БД. Эта принципиальная разница проигнорирована в указанных работах.

Интересное с теоретической точки зрения обобщение неопределенностей предложено в работе [8]. Вводится определение нечеткой ФЗ: зависимость выполнена, если равны левые части ФЗ и доля кортежей (мера) с определенными значениями для правой части зависимости в отношении не меньше β ($0 \le \beta \le 1$). Кроме того, рассмотрены операции реляционной алгебры для отношений, сформированных по таким зависимостям. Использование нечетких зависимостей при проектировании схемы БД имеет ограниченное прикладное значение, поскольку на момент формирования схемы БД отсутствуют собственно данные. Это не позволяет определить меру неопределенности зависимости. Кроме того, в процессе эксплуатации БД мера может кардинально измениться. Если после этого потребуется замена схемы БД, то, как следствие, потребуется замена уже разработанного программного обеспечения. То есть нечеткие ФЗ не удовлетворяют принципу независимости данных.

Для пояснения дальнейших формальных построений рассмотрим два примера формирования схемы БД.

П р и м е р 1. Пусть задано множество атрибутов (схема отношения R): A_1 — № пациента, A_2 — ФИО пациента, A_3 — № показателя, A_4 — Показатель, A_5 — Значение показателя, A_6 — № дня получения показателя, A_7 — Группа пациентов. На предложенном множестве атрибутов существуют следующие зависимости: $F = \{A_1 \to A_2A_7, A_3 \to A_4, A_1A_3A_6 \to A_5\}$. По правилам построения нормальных форм будет получена следующая схема БД (вертикальная декомпозиция или разбиение одного общего отношения R по столбцам): Пациенты $= R_1(\underline{A}_1, A_2, A_7)$, Перечень анализов $= R_2(\underline{A}_3, A_4)$, Результаты анализов $= R_3(\underline{A}_1, \underline{A}_3, \underline{A}_6, A_5)$.

В примере 1 в явном виде отсутствуют атрибуты, которые могут принимать неопределенные значения. Допустим, что при эксплуатации БД появилась необходимость хранить адрес пациента. Поскольку схема БД была сформирована на основе ФЗ, то изменения будут выполнены без разрушения существующей схемы и необходимости переписывания прикладных программ (принцип независимости данных). Для этого достаточно дополнить атрибут A_8 — “Адрес пациента” в отношение R_1. На начальном этапе после модернизации схемы атрибут будет принимать значение $Null$ почти во всех кортежах отношения R_1. Однако в процессе эксплуатации БД постепенно значения $Null$ будут заменены на реальные значения.

Рассмотрим другой пример.

П р и м е р 2. Пусть задано множество атрибутов: A_1 — № сотрудника, A_2 — ФИО сотрудника, A_3 — должность сотрудника, A_4 — дата увольнения сотрудника. На предложенном множестве атрибутов существуют следующие зависимости: $F = \{A_1 \to A_2,\ A_1 \to A_3,\ A_1 \to A_4\}$. По правилам построения нормальных форм будет получена следующая схема БД: Сотрудники $= R(\underline{A}_1, A_2, A_3, A_4)$.

В примере 2 атрибут A_4 будет иметь неопределенное значение $Null$ для всех не уволенных сотрудников. Выполнение операции сравнения в запросе на языке SQL для атрибута A_8 в примере 1 и для атрибута A_4 в примере 2 выдаст значение $unknown$ в тех кортежах, где есть значение $Null$. Это соответствует действительности для примера 1, тогда как для примера 2 это ложный результат: известно, что данная информация отсутствует для не уволенных сотрудников.

Более правильная декомпозиция с практической точки зрения в примере 2 должна иметь вид: Сотрудники $= R_1(\underline{A}_1, A_2, A_3)$, Увольнение $= R_2(\underline{A}_1, A_4)$ (в R_2 есть кортежи только для уволенных сотрудников). Такой способ преобразования схемы состоит из вертикальной и горизонтальной декомпозиции, и значения для отсутствующей информации не появляются в БД.

В данной статье мы рассмотрим формальные основания для построения такой декомпози-

ции, которые не зависят от способа и места использования и хранения данных и не зависят от текущего состояния БД. Кроме того, будет предложено обобщение классической теории, которое естественным образом встраивается в существующие технологии проектирования и функционирования БД. Вновь вводимое понятие области определения зависимостей позволяет применять его на практике в самом начале проектирования, когда данных еще нет, поскольку является неотъемлемым свойством зависимости. Так, в примере 2 область определения зависимости $A_1 \to A_4$ легко определить как "Уволенные сотрудники" без использования собственно данных. Кроме того, эти области не зависят от текущего состояния БД, места и способа хранения данных, места и способа использования данных в приложениях. Следовательно, подход, основанный на областях определения зависимостей, удовлетворяет принципу независимости данных в БД.

Чтобы показать, что предлагаемый подход является обобщением классической теории, будем излагать материал в последовательности, как это сделано в [1].

1. Основные определения и свойства зависимостей

По аналогии с классической теорией откажемся от использования неопределенного значения $Null$ (значение возможно существует, но оно не известно) при проектировании схемы БД. Действительно, для получения устойчивой схемы БД не корректно использовать неопределенную информацию, поскольку неопределенность будет заложена в схему БД. А значения, про которые известно, что они не существуют, являются основой областей определения зависимостей и должны принять участие в проектировании, чтобы затем отсутствовать среди значений БД. То есть в проектировании участвует только определенная информация, а значение $Null$ может затем использоваться при заполнении БД для всех атрибутов, где это не противоречит целостности данных. Результат $unknown$ при обращении к таким данным будет согласован с семантикой запроса.

Идея использования только определенных значений атрибутов для построения системы аксиом функциональных зависимостей с областями их определения предложена в работе [9]. В данной работе рассматривается развитие этого подхода с использованием формальной теории в практике проектирования схем БД. Пусть nov (no value) — метка значения, про которое известно, что оно не существует. Относительно значения nov сделаем только одно допущение: его не должно быть ни в одном отношении БД после проектирования. Текущая реализация отношения R (универсальное реляционное отношение), где $[R] = U$, является отправным объектом проектирования схемы БД и содержит только определенные значения атрибутов и значения nov.

О п р е д е л е н и е 3. Пусть задана произвольная реализация отношения R на множестве атрибутов U, X — произвольное подмножество U. Областью определения $dom(X)$ атрибутов X будем называть кортежи $t \in R$, для которых выполнено: $t[A_j] \neq nov$ для всех атрибутов $A_j \in X$.

Пусть $X_1, X_2, \ldots X_m$ — произвольные множества атрибутов, $X = X_1 \cup X_2 \cup \ldots \cup X_m$. Из определения 3 следует равенство:

$$dom(X) = dom(X_1) \cap dom(X_2) \cap \ldots \cap dom(X_m). \tag{1.1}$$

Пусть X и Y — множества атрибутов, для которых $Y \subseteq X$. Из определения 3 следует еще одно очевидное соотношение:

$$dom(X) \subseteq dom(Y). \tag{1.2}$$

О п р е д е л е н и е 4. Пусть задана произвольная реализация отношения R на множестве атрибутов U. Областью определения $dom(X \to Y)$ зависимости $X \to Y$ будем называть кортежи $t \in R$, для которых выполнено:

а) $t[A_j] \neq nov$ для всех атрибутов $A_j \in X \cup Y$;

б) для любой пары кортежей $t_1, t_2 \in R$, если $t_1[X] = t_2[X]$ то $t_1[Y] = t_2[Y]$.

Пусть $F = \{F_1, F_2, \ldots, F_k\}$ — множество ФЗ. Из определения 4 следует равенство:

$$dom(F) = dom(F_1) \cap dom(F_2) \cap \ldots \cap dom(F_k).$$

Отношение R будем считать *принадлежащим* $dom(X \to Y)$, если для любого кортежа $t \in R$ выполнено $t \in dom(X \to Y)$.

О п р е д е л е н и е 5. Зависимость $X \to Y$ будем считать выполненной (логически следует) в области определения $dom(F)$ зависимостей F, если она выполнена для любого отношения $R \in dom(F)$.

Из определений 4 и 5 следует, если $D_i = dom(X \to Y)$, тогда зависимость $X \to Y$ выполнена в области D_i. Далее для краткости будем говорить, что зависимость выполнена, подразумевая, что она выполнена в области $dom(F)$.

Следующая лемма используется при доказательстве надежности системы аксиом, правил вывода и доказательстве эквивалентности множеств зависимостей.

Лемма 1. *Если зависимость $X \to Y$ выполнена, то $dom(X \to Y) = dom(X) \cap dom(Y)$.*

Д о к а з а т е л ь с т в о. Включение $dom(X \to Y) \subseteq dom(X) \cap dom(Y)$ следует непосредственно из определений 3 и 4. Покажем включение в обратную сторону. Пусть R — произвольное отношение, удовлетворяющее зависимостям F. Предположим, что существует кортеж $t_1 \in dom(X) \cap dom(Y)$ и $t_1 \notin dom(X \to Y)$. Из предположения следует, что в соответствии с определением 4 существует кортеж $t_2 \in dom(X) \cap dom(Y)$, $t_1[X] = t_2[X]$ и $t_1[Y] \neq t_2[Y]$. Тогда зависимость $X \to Y$ в соответствии с определением 5 не является выполненной. Полученное противоречие доказывает лемму.

Заметим, что полученный результат согласуется с семантикой приложений. Так, в примере 2 $dom(A_1 \to A_4) = dom(A_1) \cap dom(A_4)$. Действительно, зависимость $dom(A_1 \to A_4)$ может иметь место только на кортежах, в которых определенное значение имеют и № сотрудника и дата увольнения. С другой стороны, в реализации R не могут появиться два кортежа с совпадающим № сотрудника и различными датами увольнения, поскольку это противоречит исходному множеству ФЗ.

Рассмотрим еще одно важное свойство областей определения зависимостей. Для этого заметим, что в примере 1 имеем: $dom(A_1A_3A_6 \to A_5) \subseteq dom(A_3 \to A_4)$. Действительно, любое отношение $R \in dom(A_3 \to A_4)$, определенное на всем множестве атрибутов, может содержать кортежи, в которых A_1, A_5 и A_6 содержат значения *nov*. Например, анализы по какому-либо показателю запрещены для определенной группы пациентов. С другой стороны, если $R \in dom(A_1A_3A_6 \to A_5)$, то в R любой кортеж должен содержать определенные значения атрибутов A_3 и A_4, и любая пара кортежей, содержащая совпадающие значения атрибута A_3, должна содержать совпадающие значения атрибута A_4. Это обусловлено семантикой приложения. Аналогичные соображения в примере 2 были использованы для декомпозиции. Опираясь на данные рассуждения, сформулируем следующее определение.

О п р е д е л е н и е 6. Будем полагать, что $dom(X \to Y) \subseteq dom(V \to Z)$, если для любой реализации отношения $R \in dom(X \to Y)$ выполнено $R \in dom(V \to Z)$.

Из определения 6 непосредственно следует еще одно свойство логического следствия, которое доказано в следующей лемме и используется при построении замыкания множеств атрибутов.

Лемма 2. *Пусть зависимость $X \to Y$ выполнена в области D_j, тогда она выполнена в области D_i, где $D_i \subseteq D_j$.*

Д о к а з а т е л ь с т в о. Пусть R произвольное отношение в области D_i. По определению 6 $R \in D_j$. В D_j зависимость $X \to Y$ выполнена. Следовательно, любая пара кортежей отношения R удовлетворяет условиям определения 4, а значит, зависимость $X \to Y$ выполнена в D_i. Лемма доказана.

2. Система аксиом зависимостей

Впервые система аксиом для ФЗ была предложена Армстронгом [10]. Позднее была получена эквивалентная система с меньшим количеством аксиом [1]. Система имеет следующий вид:

A1. (Рефлексивность) Если $Y \subseteq X \subseteq U$, то $X \to Y$.

A2. (Пополнение) Если $X \to Y$ и $Z \subseteq U$, то $XZ \to YZ$ (где $XZ = X \cup Z$).

A3. (Транзитивность) Если $X \to Y$ и $Y \to Z$, то $X \to Z$.

Заметим, что аксиомы одновременно являются правилами вывода. В них подразумевается, что полученная зависимость имеет максимальную область определения $dom(X \to Y) = All$, совпадающую со всем множеством кортежей отношения R, поскольку не допускается наличие неопределенных и несуществующих значений.

Перепишем систему аксиом Армстронга в соответствии с определением 4 и леммой 1:

A1*. (Рефлексивность) Если $Y \subseteq X \subseteq U$, то $X \to Y$ и $dom(X \to Y) = dom(X)$.

A2*. (Пополнение) Если $X \to Y$ и $Z \subseteq U$, то $XZ \to YZ$ и $dom(XZ \to YZ) = dom(X) \cap dom(Y) \cap dom(Z)$.

A3*. (Транзитивность) Если $X \to Y$ и $Y \to Z$, то $X \to Z$ и $dom(X \to Z) = dom(X) \cap dom(Z)$.

Очевидно, что замена области определения зависимостей значением All приведет к исходной системе аксиом. При доказательстве полноты системы аксиом будет показано, что именно аксиомы **A1***–**A3*** соответствуют ФЗ с областями определения.

Рассмотрим надежность (непротиворечивость) системы аксиом.

Теорема 1. *Система аксиом* **A1***–**A3*** *надежна.*

Доказательство. Надо показать, что если зависимость выводима за счет аксиом, то она выполнима (логически следует). Формально это записывается следующим образом: $F \vdash \{X \to Y\} \Rightarrow F \models \{X \to Y\}$.

Для обоснования *надежности аксиомы* **A1*** рассмотрим произвольное отношение R, определенное на множестве атрибутов U. Любая пара кортежей, имеющая определенные значения для всех атрибутов множества X и совпадающая по этим атрибутам, будет совпадать по атрибутам Y, т. е. зависимость $X \to Y$ всегда выполнена. По лемме 1 имеем: $dom(X \to Y) = dom(X) \cap dom(Y)$. Применив соотношение 1.2, окончательно получим: $dom(X \to Y) = dom(X)$. Это доказывает надежность **A1***.

Надежность аксиомы **A2***. Задано отношение R на множестве атрибутов U. Пусть существуют два кортежа t_1 и t_2 в R, имеющие определенные значения на атрибутах X, Y и Z, причем $t_1[XZ] = t_2[XZ]$ и $t_1[YZ] \neq t_2[YZ]$. Поскольку $X \to Y$, то $t_1[Y] = t_2[Y]$ и $t_1[Z] \neq t_2[Z]$, что противоречит условию $t_1[XZ] = t_2[XZ]$. Следовательно, $t_1[YZ] = t_2[YZ]$ и зависимость $XZ \to YZ$ выполнена. По лемме 1 имеем: $dom(XZ \to YZ) = dom(XZ) \cap dom(YZ)$. Применив равенство 1.1, окончательно получим: $dom(XZ \to YZ) = dom(X) \cap dom(Y) \cap dom(Z)$. Это доказывает надежность **A2***.

Надежность **A3***. Рассмотрим произвольное отношение R, в котором выполнены зависимости $X \to Y$ и $Y \to Z$. То, что в R выполнена зависимость $X \to Z$ доказывается по аналогии с доказательством надежности **A2***: предполагаем, что не выполнена зависимость $X \to Z$, получаем противоречие зависимости $X \to Y$ или $Y \to Z$. Поскольку зависимость $X \to Z$ выполнена, то по лемме 1 $dom(X \to Z) = dom(X) \cap dom(Z)$. Надежность **A3*** доказана. Теорема доказана.

Убедившись в надежности аксиом **A1***–**A3***, можно получить некоторые полезные правила (теоремы). Наибольшее значение при проектировании схемы БД при построении нормальных форм имеют два правила: объединение (при синтезе отношений) и декомпозиция (при декомпозиции отношений) [1, с. 158]. В следующей лемме доказываются оба правила.

Лемма 3. *Из аксиом* **A1*–A3*** *выводимы следующие правила*:

Ru-1. (*Объединение*) *Если* $X \to Y$ *и* $X \to Z$, *то* $X \to YZ$ *и* $dom(X \to YZ) = dom(X) \cap dom(Y) \cap dom(Z)$.

Ru-2. (*Декомпозиция*) *Если* $X \to Y$ *и* $Z \subseteq Y$, *то* $X \to Z$ *и* $dom(X \to Z) = dom(X) \cap dom(Z)$.

Доказательство. Выводимость обоих правил доказана в лемме 5.2 [1]. Для обоснования областей определения зависимостей достаточно показать их выполнимость, что делается по аналогии с доказательством надежности системы аксиом. Далее, используя лемму 1, получаем соотношения для областей определения зависимостей. Лемма доказана.

В [1] доказательство полноты следует сразу после определения замыкания множества атрибутов и основывается на этом замыкании. При поверхностном изучении материала возникает иллюзия, что наличие замыкания является необходимым и достаточным условием полноты системы аксиом. Это заблуждение можно встретить в некоторых Интернет-изданиях, где двустрочное доказательство полноты по смыслу сводится к фразе: “потому что, потому” без упоминания самих аксиом. Однако если при доказательстве полноты какая-либо аксиома не используется, то закономерным является вывод о ее избыточности. При этом не обязательна выводимость этой аксиомы из остальных аксиом. Например, можно дополнить аксиому:

A4*. (Значность) Если $X \to Y$ и $Y \neq C$, то $X \neq \varnothing$, где C — константа.

Эта аксиома не выводима из остальных, но она описывает ненужное для выводимости свойство объекта, следовательно, является избыточной.

В [1] эта ситуация исправляется последующим исследованием свойств замыкания и доказательством эквивалентности выводимости замыканию. Из сказанного следует, что сначала надо исследовать замыкания множества атрибутов с учетом областей определения зависимостей, а затем доказать полноту системы аксиом.

Наличие областей определения позволяет рассматривать различные замыкания. Однако основной целью далее является выяснение выводимости зависимости в заданной области. Следовательно, нужно вычислять замыкание в определенной области, а не область определения уже полученного замыкания.

Замыканием X_D^+ *множества атрибутов* X *в области определения* D будем называть атрибуты $Y \subseteq U$ такие, что зависимость $X \to Y$ выводима из F за счет аксиом **A1*–A3*** и $D \subseteq dom(X \to Y)$.

Рассмотрим следующий алгоритм. Текущее замыкание обозначим через X^+.

Алгоритм 1: Построение замыкания множества атрибутов X в области D.

$X^+ = \varnothing$
$IF\ D \not\subseteq dom(X)\ THEN\ STOP$
$X^+ = X$
$substitution = TRUE$
$WHILE\ substitution$
  $substitution = FALSE$
  $FOR\ EACH\ X_i \to Y_i\ FROM\ F$
    $IF\ D \subseteq dom(X_i \to Y_i)\ THEN$
      $IF X_i \subseteq X^+ AND\ Y_i - X^+ \neq \varnothing\ THEN$
        $X^+ = X^+ \cup Y_i\ ;\ substitution = TRUE$
      $END\ IF$
    $END\ IF$
  $END\ FOR$
$END\ WHILE$
$STOP$

Внешний цикл $WHILE$ не имеет явного ограничения. Но для выполнения следующего цикла необходимо дополнение хотя бы одного атрибута к замыканию во внутренних циклах.

Следовательно, максимальное количество итераций в алгоритме равно nk, где n — количество атрибутов во множестве U и k — количество зависимостей во множестве F.

З а м е ч а н и е. Фактически в построении замыкания X^+ в соответствии с леммой 2 участвуют не все зависимости F, а только такие $F_i \in F$, для которых $D \subseteq dom(F_i)$. Кроме того, замыкание множества атрибутов, независимо от использования областей определения, есть некоторый индикатор возможности вывода зависимости, но не сам вывод. Так, при построении замыкания явно не использовалась аксиома **A2***, но это не значит, что она лишняя: правило объединения (лемма 3) без нее вывести невозможно. Тем не менее обе правые части зависимостей этого правила присутствуют в замыкании левой части, что свидетельствует об их выводимости.

Доказательство корректности построения замыкания по структуре похоже на доказательство теоремы 5.2 [1]. Все же учитывая специфику алгоритма построения замыкания и его важность для доказательства полноты системы аксиом, рассмотрим полностью доказательство следующей теоремы.

Теорема 2. *Алгоритм построения замыкания корректно формирует множество* X_D^+.

Д о к а з а т е л ь с т в о. Пусть Y — произвольное множество атрибутов. Необходимо показать, что $Y \subseteq X_D^+$ тогда и только тогда, когда $Y \subseteq X^+$ или $X_D^+ = X^+$.

Необходимость. Пусть $Y \subseteq X^+$. Пополнение множества X^+ выполняется в двух операторах:

а) $X^+ = X$ — все атрибуты множества X выводимы из X по аксиоме рефлексивности. Если $D \not\subseteq dom(X)$, то по определениям 3 и 4 при любом выводе зависимости $X \to Y$ выполнено: $dom(X \to Y) \subseteq dom(X)$, следовательно $X \to Y$ не выводима в области D. Более того, при $D \not\subseteq dom(X)$ из X ничего не выводимо в области D, даже само множество X, что соответствует пустому замыканию.

б) $X^+ = X^+ \cup Y_i$, где $X_i \to Y_i \in F$, $D \subseteq dom(X_i \to Y_i)$ и $X_i \subseteq X^+$. По аксиоме рефлексивности $X^+ \to X_i$ и по аксиоме транзитивности $X^+ \to Y_i$ в области D, следовательно, все атрибуты Y_i принадлежат X_D^+ (до или после подстановки), и зависимость $X \to Y$ выводима в области D.

Достаточность. Пусть $Y \subseteq X_D^+$. Тогда существует k строк вывода зависимости $X \to Y$. Область определения каждой строки вывода содержит область D. При $k = 0$ имеем $Y \subseteq X$ и вывод получен по аксиоме рефлексивности, либо $X \to Y \in F$. В обоих случаях $Y \subseteq X^+$ по алгоритму.

Пусть условие теоремы выполнено для вывода из $k-1$ строк, и зависимость $X \to Y$ — это k-я строка вывода при $k > 0$. Кроме вариантов вывода при $k = 0$ может иметь место вывод за счет аксиомы транзитивности из двух строк вывода: $X \to Z$ и $Z \to Y$. Поскольку вывод $X \to Z$ выполняется за количество строк менее k, то $Z \subseteq X^+$. Следовательно, Y будет принадлежать X^+ на следующей итерации алгоритма.

Пусть последняя строка вывода $X \to Y$ получена за счет аксиомы пополнения: атрибуты Z дополняются к атрибутам зависимости $V \to W$. Тогда $X = VZ$ и $Y = WZ$. Поскольку $Z \subseteq X$, то $Z \subseteq X^+$ и $W \subseteq X^+$. Так как цепочка вывода зависимости $V \to W$ содержит меньшее k строк, то атрибуты Y также будут принадлежать X^+. Теорема доказана.

Полученное доказательство эквивалентности выводимости и принадлежности замыканию существенно упрощает доказательство полноты системы аксиом. Достаточно показать для произвольной зависимости $X \to Y$, выполненной в области D, что $Y \subseteq X_D^+$.

Теорема 3. *Система аксиом* **A1***–**A3*** *полна.*

Д о к а з а т е л ь с т в о. Пусть D — произвольная область определения зависимостей. Необходимо показать, что если зависимость $X \to Y$ выполнима в области D: $F \models \{X \to Y\}$, то она выводима $F \vdash \{X \to Y\}$ в этой области: $Y \subseteq X_D^+$.

По аналогии с доказательством полноты в [1] рассмотрим отношение R, содержащее всего два кортежа: t_1 и t_2: $t_1[A_j] = t_2[A_j]$, если $A_j \in X_D^+$, и $t_1[A_j] \neq t_2[A_j]$ — в противном случае. Все зависимости $V \to W \in F$, для которых $D \subseteq dom(V \to W)$, выполнены в R. Действительно, если $V \in X_D^+$, то $W \in X_D^+$ по алгоритму.

Предположим, что зависимость $X \to Y$ не выводима. Тогда существует атрибут A^*: $A^* \in Y$ и $A^* \notin X_D^+$. Следовательно, R является реализацией отношения, в котором зависимость $X \to Y$ не выполнена. Полученное противоречие доказывает теорему.

Полученные свойства выводимости зависимостей в заданных областях определения далее будем использовать для построения нормальных форм отношений.

3. Нормальные формы

На основе множества ФЗ можно построить две финальные нормальные формы: а) третья нормальная форма (3НФ) и б) нормальная форма Бойса — Кодда (БКНФ), которая еще считается усиленной третьей нормальной формой. Алгоритм построения БКНФ основан на последовательном отщеплении выводимых атрибутов от главного отношения. С теоретической точки зрения БКНФ считается более совершенной, чем 3НФ, поскольку лучше решает проблему аномалий в отношениях и является основой для построения последующих нормальных форм.

С практической точки зрения БКНФ обладает существенными недостатками. Во-первых, экспоненциальный алгоритм поиска очередной выводимой ФЗ в реальных системах проектирования БД может просто не достигнуть результата из-за ограничений по времени. Причем эта задача предполагает поиск только глобального решения на всем множестве атрибутов. Во-вторых, БКНФ не сохраняет зависимости за счет удаления из дальнейшего рассмотрения выводимых атрибутов, что приводит к потере некоторых объектов БД. Это происходит, когда удаленные атрибуты участвуют в других зависимостях и с другими областями определения. Поэтому на практике чаще используется 3НФ с полиномиальным алгоритмом построения и без потери каких-либо ФЗ. Кроме того, если некоторое отношение в 3НФ имеет аномалию, то для решения проблемы достаточно провести локальную декомпозицию этого отношения на сравнительно небольшом количестве атрибутов.

Рассмотренный в [1] синтетический подход к формированию отношений, основанный на построении минимального покрытия множества ФЗ, достаточно просто обобщается на случай наличия областей определения у ФЗ.

В данной работе не вводилось определение замыкания множества ФЗ, поскольку при построении нормальных форм достаточно определить выводимость некоторой ФЗ. В соответствии с этим условием определим эквивалентность множеств ФЗ, которое далее используется для построения минимального покрытия множества ФЗ.

Два множества зависимостей F и E будем считать *эквивалентными* ($F \equiv E$), если каждая зависимость $X \to Y \in F$ с областью определения $D = dom(X \to Y)$ выводима на множестве E: $Y \subseteq X_D^+(E)$ и каждая зависимость $V \to W \in E$ с областью определения $C = dom(V \to W)$ выводима на множестве F: $W \subseteq V_C^+(F)$, где $X_D^+(E)$ — замыкание множества атрибутов X в области D на множестве зависимостей E.

Построим множество зависимостей E из множества F. Для каждой зависимости $X \to Y$ множества F во множество E дополняем зависимости $X \to A_i^{'}$, $i = \overline{1,m}$, где $Y = A_1^{'}, A_2^{'}, \ldots A_m^{'}$. По правилу декомпозиции (лемма 3) $dom(X \to A_i^{'}) = dom(X) \cap dom(A_i^{'})$.

Лемма 4. *Множества зависимостей F и E эквивалентны.*

Доказательство. По правилу декомпозиции (лемма 3) зависимости $X \to A_i^{'} \in E$ выводимы в F и $dom(X \to A_i^{'}) = dom(X) \cap dom(A_i^{'})$. Зависимость $X \to Y \in F$ выводима во множестве E по правилу объединения зависимостей (лемма 3) $X \to A_i^{'}$, $i = \overline{1,m}$. Используя

уравнение 1.1, получаем $dom(X \to Y) = dom(X) \cap dom(Y)$, что совпадает с исходной областью определения зависимости $X \to Y$. Лемма доказана.

Далее, используя полученное условие эквивалентности, построим множество зависимостей, которое в [1] называется минимальным покрытием. Строго говоря, использование слова "минимальный" не является корректным, поскольку отсутствует критерий минимальности. В данном случае строится не избыточное и непротиворечивое множество зависимостей, которое позволяет при синтетическом подходе получить не избыточную и непротиворечивую БД. В данной работе представим формальную версию такого алгоритма, где F — исходное множество, G — результат (он составлен с учетом анализа исходного алгоритма в работе [11]):

А л г о р и т м 2

$G = \varnothing$
$FOR\ EACH\ X_i \to Y_i\ FROM\ F$
  $FOR\ EACH\ A_j\ FROM\ Y_i$
    $G = G \cup \{X_i \to A_j\}$
    $dom(X_i \to A_j) = dom(X_i) \cap dom(A_j)$
  $ENDFOR$
$ENDFOR$
$modify = TRUE$
$WHILE\ modify$
  modify = FALSE
  $FOR\ EACH\ X \to A_j\ FROM\ G$
    $IF\ A_j \in X^{+}_{dom(X \to A_j)}(G - \{X \to A_j\})\ THEN$
      $G = G - \{X \to A_j\}$; $modify = TRUE$
    $END\ IF$
  $END\ FOR$
  $FOR\ EACH\ X \to A_j\ FROM\ G$
    $FOR\ EACH\ A_i\ FROM\ X$
      $Z = X - A_i$; $log1 = log2 = log3 = FALSE$
      $G^{'} = G - \{X \to A_j\} \cup \{Z \to A_j\}$
      $IF A_j \in Z^{+}_{dom(X \to A_j)}(G)\ THEN\ log1 = TRUE$
      $IF dom(X \to A_j) = dom(Z \to A_j)\ THEN\ log2 = TRUE$
      $IF A_j \in X^{+}_{dom(Z \to A_j)}(G^{'})\ THEN\ log3 = TRUE$
      $IF\ log1\ AND\ (log2\ OR\ log3)\ THEN$
        $G = G^{'}$; $modify = TRUE$
        $dom(Z \to A_j) = dom(Z) \cap dom(A_j)$
      $END\ IF$
    $END\ FOR$
  $END\ FOR$
$END\ WHILE$
$STOP$

За один проход в цикле $WHILE$ алгоритма удаляется одна зависимость или один атрибут, следовательно, максимальное количество проходов этого цикла равно $n+m$, где n — количество атрибутов во множестве U, m — количество зависимостей во множестве G при первоначальном формировании до цикла $WHILE$. Тогда с учетом затрат на построение замыканий можно определить максимальное количество итераций для всего алгоритма: $nk+(n+m)(m^2n+2m^2n^2)$, где k — количество зависимостей во множестве F. Очевидно, что реально алгоритм будет делать значительно меньше итераций. Главное в этой оценке то, что она полиномиальная. А это значит, что алгоритм применим для схемы БД любого размера.

Докажем корректность алгоритма удаления избыточных зависимостей и атрибутов во множестве F.

Теорема 4. *Множества зависимостей F и G эквивалентны.*

Доказательство. Эквивалентность множества G множеству F на предварительном этапе (до цикла WHILE) доказана в лемме 4. В цикле WHILE множество G подвергается изменению в двух операторах. Покажем, что всякий раз получаем эквивалентное множество зависимостей.

Если выполнено условие $A_j \in X^+_{dom(X\to A_j)}(G - \{X \to A_j\})$, то удаляется зависимость $X \to A_j$ из множества G. Исходное множество G и результирующее множество G отличаются только наличием зависимости $X \to A_j$ в исходном множестве. Поэтому проверять выводимость остальных зависимостей нет необходимости. Условие $A_j \in X^+_{dom(X\to A_j)}(G - \{X \to A_j\})$ говорит о выводимости зависимости $X \to A_j$ из остальных зависимостей множества G в области определения, включающей в себя $dom(X \to A_j)$. Следовательно, эта зависимость избыточна и ее удаление не повлияет на количество выводимых зависимостей.

При выполнении условия $log1\ AND\ (log2\ OR\ log3)$ атрибут A_i удаляется из левой части зависимости $X \to A_j$, где $A_i = X - Z$. Исходное множество G и результирующее множество G' отличаются двумя зависимостями: $X \to A_j$ есть только в исходном множестве, $Z \to A_j$ есть только в результирующем множестве. Остальные зависимости совпадают. Следовательно, надо проверить выводимость обеих зависимостей с собственными областями определения во множествах, которым эти зависимости не принадлежат. Из леммы 1, соотношений 1.1 и 1.2 следует, что $dom(X \to A_j) \subseteq dom(Z \to A_j)$, и если удаление атрибута A_i из множества X оставляет область определения X без изменений, то условие $A_j \in X^+_{dom(Z\to A_j)}(G')$ (условие $log3 = TRUE$) благодаря аксиоме **A1*** всегда выполнено. Следовательно, замена исходной зависимости на зависимость $Z \to A_j$ не повлияет на количество выводимых зависимостей при одновременном выполнении условий $log1$ и $log3$. Теорема доказана.

Конечной целью данной работы является построение схемы БД, учитывающей области определения функциональных зависимостей, и БД, построенная для этой схемы, не должна содержать значений *nov*. Исходными данными являются множества: $U = \{A_1, A_2, \ldots, A_n\}$ — атрибуты в прикладной области, $F = \{F_1, F_2, \ldots, F_m\}$ — зависимости $F_i = X_i \to Y_i$, для которых удалены избыточные зависимости и избыточные атрибуты в левых частях, и соответствующие им области определения зависимостей $\{D_1, D_2, \ldots, D_m\}$.

Алгоритм 3

```
FOR i = 1 TO m − 1
  IF Y_i ≠ ∅ THEN
    FOR j = i + 1 TO m
      IF X_i = X_j AND D_i = D_j AND Y_j ≠ ∅ THEN
        Y_i = Y_i ∪ Y_j ; Y_j = ∅
      END IF
    END FOR
  END IF
END FOR
k = 0
FOR i = 1 TO m
  IF Y_i ≠ ∅ THEN
    k = k + 1 ; R_k = TABLE(PRIMARY KEY(X_i), Y_i)
    dom(R_k) = dom(X_i → Y_i)
  END IF
END FOR
```

Команда $R_k = TABLE(PRIMARY\ KEY(X_i),\ Y_i)$ формирует отношение R_k со схемой $X_i \cup Y_i$ и первичным ключом X_i. Отношению R_k приписывается область определения, совпадающая с соответствующей ФЗ. Сложность алгоритма — $(m+1)m/2$ итераций.

Рассмотренный алгоритм формирует изначально пустую БД: $\{R_1, R_2, \ldots, R_k\}$, в отношения которой нельзя дополнить значения *nov*, поскольку это будет противоречить области

определения отношений. По аналогии с [1] несложно показать, что каждое R_i удовлетворяет требованиям третьей нормальной формы (3НФ).

Далее обычно определяются условия, при которых "декомпозиция" $\{R_1, R_2, \ldots, R_k\}$ (в нашем случае — синтез) [1] обладает свойством соединения без потерь информации (СБПИ). В классической теории это свойство рассматривается сразу для всего множества отношений БД. Ничего не изменилось и в более поздних публикациях, например [12], где это свойство исследуется для нечетких ФЗ. В обоих случаях это оправдано, поскольку выполнимость ФЗ предполагается для всей прикладной области. В данной работе это не так, и говорить о выполнении данного свойства для всех отношений некорректно: это свойство может быть формально выполнено, но в пустой области определения. По аналогии с построением замыкания надо говорить о выполнении свойства СБПИ в некоторой фиксированной области определения. В соответствии с леммой 2 в проверке свойства будут участвовать только те зависимости и отношения, область определения которых не меньше заданной.

Алгоритм проверки свойства СБПИ и доказательство его корректности аналогичны алгоритму и доказательству в [1], только если взять множество отношений $\{R_1^{'}, R_2^{'}, \ldots, R_m^{'}\}$, множество атрибутов X и подмножество зависимостей $F_i \in F$, для которых $D \subseteq dom(F_i)$.

4. Заключение

В работе получена формальная теория функциональных зависимостей с учетом областей определения, которая является обобщением классической теории и согласуется с практикой применения нормальных форм в базах данных. На основе полученных результатов разработаны алгоритмы формирования не избыточного множества зависимостей и третьей нормальной формы с учетом областей определения.

С учетом взаимосвязи функциональных зависимостей с другими видами зависимостей планируется распространить на них понятие области определения. Так, с практической точки зрения нет различий между встроенными и обычными многозначными зависимостями, тогда как для первого вида зависимостей формальная теория аксиоматически неполна, а для вторых — полна и непротиворечива. Планируется построить общую модель многозначных зависимостей. Аналогичные планы существуют относительно других видов зависимостей.

СПИСОК ЛИТЕРАТУРЫ

1. **Ullman J.** Principles of database systems / Stanford University. Stanford: : Computer Science Press, 1980. 379 p.
2. **Maier D.** The theory of relational databases / Oregon Graduate Center. Rockville: Computer Science Press, 1983. 637 p.
3. **Zaniolo C.** Database relations with null values // J. Comput. System Sci. 1984. No. 28. P. 142–166.
4. **Vassiliou Y.** Functional dependencies and incomplete information // Proc. of 6th Internat. Conf. on Very Large Data Bases (VLDB '80). Montreal, 1980. Vol. 6. P. 260–269.
5. **Atzeni P., Morfuni N.** Functional dependencies and constraints on null values in database relations // Information and Control. 1986. Vol. 70, no 1. P. 1–31.
6. **Levene M., Loizou G.** Axiomatisation of functional dependencies in incomplete relations // Theoretical Computer Science. 1998. Vol. 206, no 1-2. P. 283–300.
7. **Hartmann S., Link S.** The implication problem of data dependencies over SQL table definitions: axiomatic, algorithmic and logical characterizations // ACM Transactions on Database Systems. 2012. Vol. 37, no 2. P. 1–40.
8. Non-transitive fuzzy dependencies (I) / J.C. Cubero, J.M. Medina, O. Pons, M.A. Vila // Fuzzy Sets and Systems. 1999. Vol. 106, no 3. P. 401–431.
9. **Zykin S.V.** Domain of dependencies in database scheme // Omsk publishing house of OmSTU: Applied mathematics and fundamental informatics. 2014. P. 75–80.
10. **Armstrong W.W.** Dependency structures of data base relationships // Proc. IFIP Congress. Amsterdam, 1974. P. 580–583.

11. **Филиппович А.Ю.** Принципы взаимных функциональных зависимостей // Интеллектуальные технологии и системы: сб. ст. М.: Изд-во МГУП, 2002. No 4. С. 222–241.
12. **Liu J.Y.-C.** Lossless Join decomposition for extended possibility-based fuzzy relational databases // J. Appl. Math. 2014. Article ID 842680. P. 1–9.

Зыкин Сергей Владимирович
д-р техн. наук, профессор
зав. лабораторией
Институт математики им. С. Л. Соболева СО РАН
e-mail: szykin@mail.ru

Поступила 10.10.2015