

©Зыкин В. С., Зыкин С. В., 2016

DOI: 10.18255/1818-1015-2017-2-155-167

УДК 004.652.4

Анализ типизированных зависимостей включения с неопределенными значениями

Зыкин В. С., Зыкин С. В.

получена 24 августа 2016

Аннотация. Неопределенные значения стали актуальной проблемой с момента создания реляционной модели данных. Влияние неопределенностей сказывается на всех видах зависимостей, используемых при проектировании и эксплуатации базы данных. В полной мере это относится и к зависимостям включения, которые являются теоретической основой ссылочной целостности на данные. Попытки решения указанной проблемы содержат неточности как в постановке задачи, так и в самом ее решении. К постановочным ошибкам можно отнести использование в определении нетипизированных зависимостей включения, что приводит к перестановкам атрибутов, хотя в технологиях баз данных атрибуты идентифицируются по имени, а не по их позиции. Кроме того, связывание зависимостью включения разнородных, пусть даже однотипных, атрибутов является признаком потерянной функциональной зависимости и приводит к взаимодействию нетривиальных зависимостей включения и функциональных зависимостей. Зависимости включения должны определять количественное соотношение объектов друг с другом, а не значений атрибутов. Неточности в решении указанной проблемы содержатся в формулировках аксиом и доказательстве их свойств, в том числе полноты. В этой статье предлагается оригинальное решение этой проблемы только для типизированных зависимостей включения при наличии неопределенных значений: предложена система аксиом, доказана ее полнота и непротиворечивость. На основе правил вывода разработан алгоритм построения не избыточного множества типизированных зависимостей включения. Доказана корректность этого алгоритма.

Ключевые слова: база данных, зависимости включения, аксиоматика, неопределенные значения

Для цитирования: Зыкин В. С., Зыкин С. В., "Анализ типизированных зависимостей включения с неопределенными значениями", *Моделирование и анализ информационных систем*, **24:2** (2017), 155–167.

Об авторах:

Зыкин Владимир Сергеевич, orcid.org/0000-0002-6492-2464, аспирант,
Омский государственный технический университет,
просп. Мира, 11, г. Омск, 644050 Россия, e-mail: vszykin@mail.ru

Зыкин Сергей Владимирович, orcid.org/0000-0002-0576-2149, д-р техн. наук, профессор,
Институт математики им. С.Л. Соболева СО РАН,
ул. Певцова, 13, г. Омск, 644043 Россия, e-mail: szykin@mail.ru

Введение

Целостность базы данных (database integrity) – соответствие имеющейся в базе данных (БД) информации логике соответствующей прикладной области. Ссылочные ограничения целостности на данные (referential integrity) являются одним из основных видов ограничений, которые позволяют сохранить структурную целостность

БД. В большинстве существующих систем управления базами данных (СУБД) поддерживается такой вид ограничений, и задаются эти ограничения в виде связей (relationship) на схеме БД.

Теоретической основой ссылочных ограничений являются зависимости включения и их взаимодействие с другими видами зависимостей. Проблеме исследования зависимостей включения до сих пор уделяется внимание со стороны исследователей, поскольку, с одной стороны, остаются нерешенными некоторые теоретические проблемы. С другой стороны, практика использования БД формулирует новые требования к ссылочным ограничениям целостности.

Для методического сопровождения последующих формальных построений рассмотрим два примера схем БД.

Пример 1. Пусть задано некоторое множество отношений – фрагмент схемы БД учебного заведения, где подчеркнуты ключевые атрибуты отношений:

$R_1 = \text{Студенты}$ (№ студента, № группы, ФИО студента);

$R_2 = \text{Список групп}$ (№ группы, Код группы, № специальности, № курса);

$R_3 = \text{Предметы}$ (№ предмета, Предмет);

$R_4 = \text{Экзамен}$ (№ студента, № группы, № предмета, Оценка);

$R_5 = \text{Аттестация}$ (№ студента, № группы, № предмета, Вид аттестации, Балл).

В примере 1 на схеме должны быть установлены следующие ссылочные ограничения целостности: нельзя поставить оценку за экзамен либо балл по аттестации по предмету, которого нет в отношении R_3 . Кроме того, нельзя поставить оценку за экзамен либо балл по аттестации студенту, информацию о котором не внесли в БД. С другой стороны, между отношениями R_4 и R_5 могла быть установлена связь по атрибутам (№ студента, № группы, № предмета), которая содержательно задает следующее ограничение целостности: аттестация студентов может осуществляться только по тем предметам, по которым проставлена оценка. Однако это противоречит прикладной области, и такого ограничения быть не может.

Пример 2. Рассмотрим фрагмент схемы БД кинотеатра:

$R_1 = \text{Фильмы}$ (№ фильма, Наименование фильма, Жанр фильма);

$R_2 = \text{Расписание сеансов}$ (Дата сеанса, Время сеанса, № зала, № фильма);

$R_3 = \text{Билеты}$ (№ билета, Дата сеанса, Время сеанса, № зала, № ряда, № места).

В примере 2 на схеме должны быть установлены следующие ссылочные ограничения целостности: нельзя в расписании назначить сеанс, если неизвестен фильм. Однако возможно продать билет со свободной (неопределенной) датой посещения, и/или свободным временем, и/или неопределенным номером зала и т.д. Это означает, что определенным значениям в отношении R_2 могут быть поставлены в соответствие неопределенные значения (Null) в R_3 . И если появится необходимость заменить неопределенные значения в R_3 на определенные, то выбор может быть сделан только из соответствующих определенных значений в R_2 .

В данной работе предлагается исследование формальной теории для типизированных зависимостей включения при наличии неопределенных значений. Разработанная теория используется для построения не избыточного множества ссылочных ограничений целостности.

1. Обзор результатов

Формирование структуры отношений (таблиц) на схеме БД осуществляется с использованием функциональных зависимостей, многозначных зависимостей и зависимостей соединения [1, 2]. Однако этих зависимостей недостаточно для установления ссылочных ограничений целостности между сформированными отношениями. Для этого перечисленные зависимости дополняются зависимостями включения. Формальное определение зависимостей включения приведено в работе [3]: пусть $U = \{A_1, A_2, \dots, A_n\}$ – множество атрибутов, определенных в БД, $[R_i]$ – множество атрибутов, на которых определено отношение R_i , $[R_i] \subseteq U$, $1 \leq i \leq k$, $\mathfrak{R} = (R_1, R_2, \dots, R_k)$ – БД, $S = \{[R_1], [R_2], \dots, [R_k]\}$ – схема БД.

Определение 1. Пусть $[R_i]$ и $[R_j]$ – схемы отношений (не обязательно различные), $V \subseteq [R_i]$ и $W \subseteq [R_j]$, $|V| = |W|$, тогда соотношение $R_i[V] \subseteq R_j[W]$ называется зависимостью включения.

В определении 1 $|V|$ – мощность множества V , $R_i[V] = \pi_V(R_i)$ – проекция отношения R_i по атрибутам V . Зависимость включения считается типизированной, если $V = W$, в противном случае – нетипизированной.

В этой же работе [3] представлена система аксиом зависимостей включения:

- **IND1**) (рефлексивность): $R_i[X] \subseteq R_i[X]$, если X – последовательность отдельных атрибутов R_i .
- **IND2**) (проецирование и перестановка): если $R_i[A_1, \dots, A_m] \subseteq R_j[B_1, \dots, B_m]$, тогда $R_i[A_{i_1}, \dots, A_{i_q}] \subseteq R_j[B_{i_1}, \dots, B_{i_q}]$ для каждой последовательности i_1, \dots, i_q различных целочисленных значений из множества $\{1, \dots, m\}$.
- **IND3**) (транзитивность): если $R_i[X] \subseteq R_j[Y]$ и $R_j[Y] \subseteq R_l[Z]$, тогда выполнено $R_i[X] \subseteq R_l[Z]$.

Относительно системы **IND1–IND3** в [3] представлено доказательство полноты. Ввиду очевидности доказательство непротиворечивости (надежности) опущено. Действительно, при условии отсутствия неопределенных значений доказательство надежности аксиом сводится к сопоставлению связанных кортежей в отношениях. Вопросы возникают при анализе доказательства полноты системы аксиом **IND1–IND3**. При доказательстве условия $\Sigma \models \sigma \Rightarrow \Sigma \vdash \sigma$: если зависимость σ выполнима (логически следует из Σ), то σ выводима из Σ с использованием аксиом **IND1–IND3**, рассматривается правило (*Rule*) формирования представления БД, которое удовлетворяет зависимостям Σ . Затем показано, что если в БД выполнена зависимость σ , то она выводима из Σ . При такой схеме доказательства необходимо показать выводимость σ для любого состояния БД, а не только единственного состояния, построенного по правилу. Несомненно, можно показать выводимость σ для любого состояния БД, но это в работе [3] не сделано. Во всех остальных работах, посвященных зависимостям включения, доказательство полноты системы аксиом сводится к ссылке на работу [3]. Это была одна из причин написания данной статьи.

Разносторонний анализ нетипизированных зависимостей включения показал, что полная аксиоматизация существует по отдельности для зависимостей включения и функциональных зависимостей, тогда как совместно для этих зависимостей

полная аксиоматизация отсутствует [3–5]. В частных случаях, в том числе для одно-местных зависимостей включения и произвольных функциональных зависимостей, существует полная аксиоматизация [6, 7].

Продолжением проблемы взаимодействия зависимостей является построение соответствующих нормальных форм (IDNF), основанных на нормальной форме Бойса–Кодда и с ограничением в виде зависимостей включения для ациклических схем БД [7, 8]. Впервые условия ациклическости схем БД были исследованы в [9]. Структурная интерпретация зависимостей включения в виде графа представлена в [10]. Хотя исследуются отличные от [8] условия ациклическости, в работе [11] показана связь между ними.

Отсутствие полной аксиоматизации говорит о том, что в общем случае нельзя определить выводимость той или иной зависимости. Как следствие, процедура построения нормальных форм в общем случае не имеет решения. На наш взгляд, причиной тому является использование нетипизированных зависимостей включения. Действительно, что означает ссылочная целостность между двумя (и более) неоднородными атрибутами? Формально это нетривиальная функциональная зависимость, которая могла быть реализована при проектировании в структуре логических записей на схеме БД, а не в виде ссылочной целостности. Необходимость использования нетипизированных зависимостей включения в этом случае отпадает. С другой стороны, типизированным зависимостям включения соответствуют тривиальные функциональные зависимости, которые при формировании структуры логических записей не используются. Следовательно, типизированные зависимости включения с нетривиальными функциональными зависимостями не взаимодействуют и могут рассматриваться отдельно друг от друга.

Значительная часть работ посвящена разработке и исследованию алгоритмов поиска зависимостей включения. В работе [10] представлен полиномиальный по времени алгоритм для поиска избыточных зависимостей на основе графического теоретического подхода. Представленный алгоритм является полным аналогом алгоритма поиска избыточных функциональных зависимостей. В работах [11–14] представлены различные алгоритмы поиска зависимостей включения, основанные на использовании свойств схемы БД, анализе данных и обработке входящих запросов. Все эти алгоритмы не гарантируют корректность и полноту обнаружения зависимостей включения, но позволяют частично автоматизировать этот процесс.

Полная автоматизация построения зависимостей включения в общем виде нереализуема, поскольку они отражают специфику бизнес-правил в конкретной прикладной области. Однако способствовать их построению позволяют алгоритмы, которые определяют полноту условий за счет определения пороговых значений качества [16, 17].

В настоящее время активно развивается направление анализа зависимостей включения. Для улучшения и автоматизации проверки и диагностики соблюдения правил на основе описания семантики данных (бизнес-данных) предлагаются структуры и алгоритмы для обнаружения возможных нарушений ограничений бизнес-данных, в том числе при модификации схемы БД [17–20]. Это подчеркивает актуальность проводимых исследований в настоящее время и в будущем.

В данной работе рассматриваются основы формальной теории для типизированных зависимостей включения. Хотя такие зависимости считаются частным случа-

ем нетипизированных зависимостей включения, система аксиом получилась иной: во второй аксиоме отсутствует необходимость использования перестановок. Сами по себе перестановки в системе **IND1-IND3** являются искусственными. Действительно, в технологиях БД атрибуты идентифицируются по имени, а не по расположению. Тогда как перестановки фиксируют расположение атрибута. Вместо перестановок можно использовать переименование атрибутов и т.п., что лишней раз подтверждает искусственную природу нетипизированных зависимостей включения.

Наличие неопределенных значений в БД является неизбежным. Поэтому формальная теория должна учитывать их при проектировании. В работах [21, 22] рассматривается проблема выводимости совместно для функциональных зависимостей и зависимостей включения, допускающих наличие неопределенных значений. Как и в случае отсутствия неопределенных значений, удается построить полную и надежную аксиоматику только в частных случаях. Основным препятствием для получения более общих результатов является взаимодействие функциональных зависимостей и нетипизированных зависимостей включения. В работе [23] рассматриваются простые и частичные нетипизированные зависимости включения с неопределенными значениями, имеющие место в стандарте языка SQL. Представлены две системы аксиом, содержащие аксиомы с перестановками атрибутов. Для обеих систем аксиом утверждается наличие полноты со ссылкой на работу [21], однако в работе [21] при утверждении полноты системы аксиом ссылка идет уже на упомянутую работу [3]. В большинстве рассмотренных работ предполагается, что зависимости включения являются нециклическими.

2. Основы формальной теории

В определении 1 представлена формулировка для типизированных зависимостей включения без учета неопределенных значений. Рассмотрим расширение этого понятия. Предварительно определим соответствующие друг другу кортежи при наличии неопределенных значений.

Определение 2. *Кортеж $t_i[X]$ соответствует кортежу $t_j[X]$ по атрибутам X ($t_j[X] \preceq t_i[X]$), если $t_i[A_l] \neq \text{Null}$, тогда $t_j[A_l] = t_i[A_l]$ или $t_j[A_l] = \text{Null}$; если $t_i[A_l] = \text{Null}$, тогда $t_j[A_l] = \text{Null}$ для любого атрибута $A_l \in X$.*

Очевидно, что заданное в определении 2 отношение $t_j[X] \preceq t_i[X]$ является транзитивным. То есть справедливо утверждение: если $t_j[X] \preceq t_i[X]$ и $t_i[X] \preceq t_m[X]$, тогда $t_j[X] \preceq t_m[X]$.

Определение 3. *Зависимость включения $\sigma = R_j[X] \subsetneq R_i[X]$ от главной таблицы $R_i[X]$ к подчиненной таблице $R_j[X]$ по атрибутам X существует, если для любого кортежа $t_j[X] \in R_j[X]$ имеется соответствующий кортеж $t_i[X]$ в отношении $R_i[X]$. Такую зависимость будем называть типизированной с допущением неопределенных значений.*

Замечание. Кортеж $t_j[X] \in R_j[X]$ может иметь множество соответствующих кортежей в отношении $R_i[X]$. Для замены неопределенных значений в кортеже $t_j[X]$ могут быть выбраны только значения одноименных атрибутов одного из соответствующих кортежей отношения $R_i[X]$, как это было показано в примере 2.

Обозначим множество зависимостей включения, определенных на схеме БД через Σ , а σ пусть будет произвольная зависимость, возможно, σ является элементом множества Σ .

Определение 4. Зависимость σ является логическим следствием множества зависимостей Σ ($\Sigma \models \sigma$), если данные в БД удовлетворяют всем зависимостям в Σ , тогда данные удовлетворяют зависимости σ . В этом случае зависимость σ будем называть **выполнимой**.

Заметим, что все зависимости Σ по определению 4 являются выполнимыми.

Представим систему аксиом, для зависимостей включения с возможными неопределенными значениями:

- **INN1)** (рефлексивность): если $X \subseteq [R_i]$, тогда $R_i[X] \subseteq R_i[X]$;
- **INN2)** (проекция): если $R_j[Y] \subseteq R_i[Y]$ и $X \subseteq Y$, тогда $R_j[X] \subseteq R_i[X]$;
- **INN3)** (транзитивность): если $R_j[X] \subseteq R_i[X]$ и $R_i[X] \subseteq R_l[X]$, тогда выполнено $R_j[X] \subseteq R_l[X]$.

Отличие системы аксиом **INN1–INN3** от системы **IND1–IND3**, кроме допущения неопределенных значений, в отсутствии перестановок в аксиоме **IND2**. В типизированных зависимостях включения могут быть сопоставлены друг другу только одноименные атрибуты, а на каком они находятся месте – не важно. Это соответствует существующим технологиям БД: на логическом уровне атрибуты идентифицируются своим именем, а не своей позицией в наборе значений.

Заметим, что аксиомы **INN1–INN3** задают правила вывода.

Определение 5. Зависимость σ **выводима** из Σ за счет системы аксиом ($\Sigma \vdash \sigma$), если при применении аксиом к зависимостям Σ за конечное число шагов будет получена зависимость σ .

Для любой системы аксиом, прежде всего, необходимо показать ее непротиворечивость (надежность). Для этого покажем, что если зависимость σ выводима из множества Σ с использованием системы аксиом, то она является логическим следствием Σ : $\Sigma \vdash \sigma \Rightarrow \Sigma \models \sigma$.

Теорема 1 (Надежность). Система аксиом **INN1–INN3** надежна.

Доказательство. Последовательно докажем надежность каждой из аксиом.

Рефлексия. Рассмотрим произвольный кортеж $t \in R_i$. В соответствии с определениями 2 и 3 кортеж t всегда будет соответствовать сам себе для любого $X \subseteq [R_i]$, что доказывает надежность аксиомы **INN1**.

Проекция. Предположим, что зависимость $R_j[X] \subseteq R_i[X]$ не выполнена. Тогда допустимы реализации R_j и R_i такие, что существует кортеж $t_j \in R_j$, для которого нет соответствующего кортежа в отношении R_i по атрибутам X . Поскольку выполнена зависимость $R_j[Y] \subseteq R_i[Y]$, для кортежа t_j существует соответствующий кортеж $t_i \in R_i$ по атрибутам Y , то есть $t_j[Y] \preceq t_i[Y]$. Поскольку $X \subseteq Y$, то $t_j[X] \preceq t_i[X]$, что противоречит предположению об отсутствии соответствующего кортежа для t_j по атрибутам X . Это доказывает надежность аксиомы **INN2**.

Транзитивность. Рассмотрим произвольный кортеж $t_j \in R_j$. Поскольку выполнена зависимость $R_j[X] \subsetneq R_i[X]$, то существует кортеж $t_i \in R_i$: $t_j[X] \preceq t_i[X]$. Так как имеет место зависимость $R_i[X] \subsetneq R_l[X]$, то существует кортеж $t_l \in R_l$ такой, что $t_i[X] \preceq t_l[X]$. В силу транзитивности операции \preceq выполнено условие $t_j[X] \preceq t_l[X]$, что доказывает надежность аксиомы **INN3**. Теорема доказана.

Замечание. Для отношений, удовлетворяющих условию аксиомы транзитивности, выполнены следующие соотношения:

$$\begin{aligned} R_j[X] \cap R_i[X] &\subsetneq R_i[X] \cap R_l[X], \\ R_j[X] \cap R_i[X] &\subsetneq R_j[X] \cap R_l[X], \\ R_j[X] \cap R_l[X] &\subsetneq R_i[X] \cap R_l[X], \end{aligned}$$

где \cap – реляционный оператор пересечения.

В [1] после доказательства надежности системы аксиом функциональных зависимостей рассматриваются правила (теоремы), которые позволяют сократить вывод других правил и получать полезные свойства схемы БД. Наиболее полезными являются правила декомпозиции и объединения функциональных зависимостей. Аналогом правила декомпозиции является аксиома проекции **INN2**, а аналога правила объединения для зависимостей включения не существует: из зависимости $R_j[X \cup Y] \subsetneq R_i[X \cup Y]$ выводимы зависимости $R_j[X] \subsetneq R_i[X]$ и $R_j[Y] \subsetneq R_i[Y]$. Обратное утверждение, к сожалению, не верно. Это является принципиальным отличием функциональных зависимостей от зависимостей включения. С одной стороны, это упрощает доказательство полноты системы аксиом **INN1–INN3**, с другой стороны – отсутствует возможность удаления атрибутов в зависимостях при построении их минимального покрытия (рассмотрено далее).

Рассмотрим оригинальное доказательство полноты системы аксиом **INN1–INN3**, в котором учтены ранее высказанные замечания.

Теорема 2 (Полнота). *Система аксиом **INN1–INN3** полна.*

Доказательство. Необходимо показать, что если зависимость $\sigma = R_j[X] \subsetneq R_i[X]$ выполнима: $\Sigma \models \sigma$, то она выводима: $\Sigma \vdash \sigma$.

Для того, чтобы зависимость σ была выводима, достаточным условием является существование цепочки выполнимых зависимостей:

$$\left\{ \begin{array}{l} R_j[Y_1] \subsetneq R'_1[Y_1] \\ R'_1[Y_2] \subsetneq R'_2[Y_2] \\ \vdots \\ R'_k[Y_{k+1}] \subsetneq R_i[Y_{k+1}], \end{array} \right. \quad (1)$$

где $X \subseteq Y_l$, $l = \overline{1, k+1}$. Действительно, по аксиомам проекции (**INN2**) и транзитивности (**INN3**), а в случае $i = j$ и по аксиоме рефлексии (**INN1**), получаем выводимость зависимости σ .

Предположим, что зависимость σ не выводима. Тогда любая последовательность (1) содержит, по крайней мере, одну не выполнимую зависимость, пусть это будет $\sigma' = R'_{m-1}[Y_m] \subsetneq R'_m[Y_m]$, где $Y_m = X$, и/или зависимость σ' выполнима, но $X \not\subseteq Y_m$. В этом случае в R'_{m-1} , а следовательно, в R_j , может существовать кортеж t , которому нет соответствующего кортежа в R'_m , а следовательно, в R_i , по атрибутам X .

Получили, что зависимость σ не выполнима. Допустим, что в Σ есть зависимости, которые препятствуют появлению кортежа t в R'_{m-1} по атрибутам X . Поскольку это должно выполняться для любого состояния БД, удовлетворяющего Σ , то зависимость $R'_{m-1}[X] \subsetneq R_i[X]$ является выполнимой и ее можно использовать вместо невыполнимого участка последовательности (1). Полученные противоречия доказывают теорему.

Заметим, что предложенную схему доказательства достаточно просто распространить на нетипизированные зависимости включения. Однако по выше названным причинам данный вид зависимостей в нашей статье не рассматривается.

Аксиомы **INN1–INN3** задают правила вывода для зависимостей включения. Следовательно, они могут быть использованы для поиска выводимых (избыточных) зависимостей в Σ .

3. Минимальное покрытие множества зависимостей

На практике ссылочная целостность, теоретической основой которой являются зависимости включения, реализуется СУБД в виде индексных файлов. Эти файлы надо хранить и модифицировать в процессе работы СУБД, что требует дополнительной памяти и времени. Поэтому целесообразно избавиться от избыточных зависимостей в Σ .

В предыдущем разделе доказано, что выводимая зависимость является выполнимой. Следовательно, ее можно удалить без всяких нежелательных последствий для БД: множество допустимых состояний БД останется без изменений. Поиск выводимых зависимостей напрямую является экспоненциальной задачей, поэтому воспользуемся известным аппаратом построения замыканий [1] для функциональных зависимостей. Аналогичный аппарат без доказательства корректности использован в работе [10]. Адаптируем эти результаты для типизированных зависимостей включения.

Определение 6. Замыканием отношения R_i на множестве зависимостей Σ относительно атрибутов X будем называть множество отношений $R_i^+[X]$, где $R_j \in R_i^+[X]$, если зависимость $\sigma = R_j[X] \subsetneq R_i[X]$ выводима из Σ за счет аксиом **INN1–INN3**, то есть $\Sigma \vdash \sigma$.

Рассмотрим алгоритм построения замыкания. Текущее замыкание обозначим $R_i^*[X]$. Будем считать, что используемые в алгоритме множества имеют глобальные имена и их не надо передавать в процедуру через параметры.

```

PROCEDURE CLOSURE( $R_i^*[X]$ );
 $R_i^*[X] = \emptyset$ ;
IF  $X - [R_i] \neq \emptyset$  THEN EXIT PROC;
 $R_i^*[X] = R_i$ ;
 $substitution = TRUE$ ;
WHILE  $substitution$ ;
     $substitution = FALSE$ ;
    FOR EACH  $R_l[Y] \subsetneq R_m[Y]$  FROM  $\Sigma$ ;

```

```

    IF  $R_m \in R_i^*[X]$  AND  $R_l \notin R_i^*[X]$  AND  $X \subseteq Y$  THEN;
         $R_i^*[X] = R_i^*[X] \cup R_l$ ; substitution = TRUE;
    END IF;
END FOR;
END WHILE;
END PROC;
```

Внешний цикл WHILE не имеет явного ограничения. Однако для выполнения следующего цикла необходимо дополнение хотя бы одного отношения к замыканию во внутреннем цикле FOR. Следовательно, максимальное количество итераций в алгоритме равно nk , где n – количество зависимостей в Σ и k – количество отношений в БД.

Теорема 3 (Замыкание). *Алгоритм CLOSURE корректно формирует множество $R_i^+[X]$.*

Доказательство. Пусть R_j – произвольное отношение и X произвольное множество атрибутов. Необходимо показать, что $R_j \in R_i^+[X]$ тогда и только тогда, когда $R_j \in R_i^*[X]$, или $R_i^+[X] = R_i^*[X]$.

1. (Необходимость) Пусть $R_j \in R_i^+[X]$. Множество $R_i^*[X]$ формируется в трех операторах:

а) $R_i^*[X] = \emptyset$. Множество $R_i^*[X]$ остается пустым, если X содержит атрибуты, которых нет в отношении R_i . По аксиомам **INN1–INN3** в этом случае также ничего не выводимо: $R_i^+[X] = \emptyset$.

б) $R_i^*[X] = R_i$. При выполнении условия $X \subseteq [R_i]$ замыкание $R_i^+[X]$ также будет содержать R_i по аксиоме рефлексии **INN1**.

в) $R_i^*[X] = R_i^*[X] \cup R_l$, если выполнены условия $R_l[Y] \subsetneq R_m[Y] \in \Sigma$, $R_m \in R_i^*[X]$, $R_l \notin R_i^*[X]$ и $X \subseteq Y$. По индукции покажем, что $R_l \in R_i^+[X]$. Базис индукции соответствует варианту (б). Предположим, что все отношения в $R_i^*[X]$ до появления зависимости $R_l[Y] \subsetneq R_m[Y]$ соответствуют выводимым зависимостям, то есть зависимость $R_m[X] \subsetneq R_i[X]$ выводима. Тогда по аксиоме проекции **INN2** имеем $R_l[X] \subsetneq R_m[X]$, так как $X \subseteq Y$, и по аксиоме транзитивности **INN3** имеем $R_l[X] \subsetneq R_i[X]$. Это верно для любого l , где $R_l \in R_i^*[X]$, в том числе для $l = j$. Следовательно, $R_i^*[X] \subseteq R_i^+[X]$.

2. (Достаточность) Пусть $R_j \in R_i^+[X]$. Тогда существует k строк вывода, где последней строкой является зависимость $R_j[X] \subsetneq R_i[X]$. При $k = 1$ имеем $j = i$, либо $R_j[X] \subsetneq R_i[X] \in \Sigma$. В обоих случаях по алгоритму отношение R_j будет присоединено к $R_i^*[X]$. Пусть условие выполнимости имеет место для всех зависимостей, вывод которых содержит не более $k-1$ строк. Кроме вариантов, рассмотренных для случая $k = 1$, зависимость $R_j[X] \subsetneq R_i[X]$ может быть получена за счет аксиомы проекции **INN2** из зависимости $R_j[Y] \subsetneq R_i[Y]$, где $X \subseteq Y$. Поскольку R_i уже содержится в $R_i^*[X]$, и, если R_j еще нет в $R_i^*[X]$, то по алгоритму R_j будет присоединено к $R_i^*[X]$, поскольку выполнены все три условия оператора IF.

Кроме того, зависимость $R_j[X] \subsetneq R_i[X]$ может быть получена за счет аксиомы транзитивности **INN3** и R_j пока нет в $R_i^*[X]$. Тогда должно существовать отношение R_l , что зависимость $R_l[Y] \subsetneq R_i[Y]$, где $X \subseteq Y$, выводима, и зависимость $R_j[Z] \subsetneq R_l[Z]$, где $X \subseteq Z$, принадлежит Σ . По предположению $j \neq l$. Тогда

$R_l \in R_i^*[X]$, так как цепочка вывода зависимости $R_l[Y] \subsetneq R_i[Y]$ короче, чем k . Поскольку вновь выполнены все условия оператора IF, то R_j будет принадлежать $R_i^*[X]$. Следовательно, $R_i^+[X] \subseteq R_i^*[X]$. Теорема доказана.

Имея в распоряжении полиномиальный алгоритм поиска избыточных зависимостей, осталось воспользоваться им для построения не избыточного множества типизированных зависимостей включения. Такое множество в [1] называется минимальным покрытием.

```

PROCEDURE MIN-COVER( $\Sigma$ );
FOR EACH  $\{R_j[X] \subsetneq R_i[X]\}$  FROM  $\Sigma$ ;
  IF  $R_j \in \text{CLOSURE}(R_i^*[X])$  THEN;
     $\Sigma = \Sigma - \{R_j[X] \subsetneq R_i[X]\}$ ;
  END IF;
END FOR;
END PROC;
```

С учетом количества итераций в алгоритме CLOSURE, результирующее количество итераций в алгоритме MIN-COVER будет равно n^2k .

После рассмотрения алгоритма MIN-COVER закономерным является вопрос об эквивалентности зависимостей Σ на входе и на выходе алгоритма. Однако уже доказанная связь между замыканием и выводимостью и то, что все выводимые зависимости являются выполнимыми, гарантирует одни те же ограничения на допустимые состояния БД со стороны зависимостей включения до и после работы алгоритма MIN-COVER.

4. Заключение

В работе рассмотрены типизированные зависимости включения, которые, по мнению авторов, наиболее приемлемы при классическом подходе к проектированию БД [1, 2]. Количественное соотношение значений атрибутов друг с другом, прежде всего, определяется функциональными зависимостями. Тогда как задача зависимостей включения – определять количественное соотношение объектов друг с другом. Использование нетипизированных зависимостей включения приводит к смешению этих двух базовых видов зависимостей и, как следствие, к проблемам при проектировании схемы БД. Причем эти проблемы сказываются не только в теории, но и на практике, когда объект БД подменяется связью на схеме.

Для подтверждения сказанных слов рассмотрим пример схемы БД на сайте разработчиков MySQL: <https://dev.mysql.com/doc/sakila/en/>. Очевидно, что при формировании этой схемы использовались в основном эвристики, а зависимости на данных имели второстепенное значение или игнорировались вовсе. Поэтому схема содержит множество неточностей. Рассмотрим одну из них, касающуюся содержания нашей статьи. Отношение (таблица) “film” имеет в своем составе атрибуты: “language_id” и “original_language_id”. Для этих атрибутов установлена ссылочная целостность с отношением “language” по атрибуту “language_id”. В результате имеем

нетипизированную зависимость включения:

$$\text{film}[\text{original_language_id}] \not\subseteq \text{language}[\text{language_id}].$$

Все фильмы имеют значение атрибута “язык оригинала”, однако только некоторые из них имеют перевод. Могут быть еще атрибуты: “автор перевода”, “дата выпуска перевода” и т.д. Не будем вдаваться в детали корректного проектирования схемы БД на основе зависимостей, это займет много места. Отметим только, что отношение БД было склеено с отношением “film”. В схеме должно существовать отдельное отношение “translate” с типизированными зависимостями включения:

$$\text{translate} [\text{language_id}] \subseteq \text{language}[\text{language_id}]$$

и

$$\text{translate}[\text{film_id}] \subseteq \text{film}[\text{film_id}],$$

а нетипизированная зависимость должна быть удалена вместе с соответствующим атрибутом.

Отметим, что при рассмотрении формальной теории зависимостей включения в нашей работе не потребовалось ограничиваться ациклическими зависимостями и ограничивать арность зависимостей, что делает предложенный аппарат достаточно универсальным. На практике циклические зависимости могут существенно снизить функциональные возможности БД, например при дополнении новой информацией. Однако это проблема проектировщика БД, который формирует множество Σ , и эта проблема решается на семантическом уровне. В качестве технологической поддержки предложенный аппарат может быть расширен правилами и алгоритмами поиска циклических зависимостей.

Список литературы / References

- [1] Ullman J., *Principles of database systems*, Computer Science Press, Stanford University, 1980, 379 pp.
- [2] Maier D., *The theory of relational databases*, Computer Science Press, Rockville, 1983, 637 pp.
- [3] Casanova M., Fagin R., Papadimitriou C., “Inclusion Dependencies and Their Interaction with Functional Dependencies”, *Journal of Computer and System Sciences*, **28**:1 (1984), 29–59.
- [4] Chandra A. K., Vardi M. Y., “The Implication Problem for Functional and Inclusion Dependencies is Undecidable”, *SIAM Journal on Computing*, **14**:3 (1985), 671–677.
- [5] Fagin R., Vardi M. Y., “Armstrong databases for functional and inclusion dependencies”, *Information Processing Letters*, **16**:1 (1983), 13–19.
- [6] Kanellakis P. C., Cosmadakis S. S., Vardi M. Y., “Unary inclusion dependencies have polynomial time inference problems”, Proceedings of the fifteenth annual ACM symposium on Theory of computing (STOC '83) (New York, USA, 1983), 1983, 264–277.
- [7] Cosmadakis S. S., Kanellakis P. C., Vardi M. Y., “Polynomial-time implication problems for unary inclusion dependencies”, *J. ACM*, **37**:1 (1990), 15–46.
- [8] Levene M., Vincent M. W., “Justification for Inclusion Dependency Normal Form”, *IEEE Transactions on Knowledge and Data Engineering*, **12**:2 (2000), 281–291.

- [9] Beeri C., Fagin R., Maier D., Yannakakis M., “On the Desirability of Acyclic Database Schemes”, *ACM*, **38**:3 (1983), 479–513.
- [10] Missaoui R., Godin R., “The Implication Problem for Inclusion Dependencies: A Graph Approach”, *SIGMOD Record*, **19**:1 (1990), 36–40.
- [11] Зыкин В. С., “Ссылочная целостность данных в корпоративных информационных системах”, *Информатика и ее применения*, **9**:3 (2015), 119–127; [Zykin V. S., “Ssylochnaya tselostnost dannykh v korporativnykh informatsionnykh sistemakh”, *Informatika i ee primeneniya*, **9**:3 (2015), 119–127, (in Russian).]
- [12] Biskup J., Dublish P., “Objects in Relational Database Schemes with Functional, Inclusion and Exclusion Dependencies”, *Theoretical Informatics and Applications*, **27** (1993), 183–219.
- [13] Johnson D. S., Klug A., “Testing Containment of Conjunctive Queries under Functional and Inclusion Dependencies”, *Computer and System Sciences*, **28** (1984), 167–189.
- [14] Marchi F. D., Lopes S., Petit J. M., “Efficient Algorithms for Mining Inclusion Dependencies”, *Advances in Database Technology - EDBT 2002 (Prague, Czech Republic, 2002)*, 2002, 199–214.
- [15] Ma S, Fan W, Bravo L., “Extending inclusion dependencies with conditions”, *Theoretical Computer Science*, **515** (2014), 64–95.
- [16] Bauckmann J., Abedjan Z., Leser U., Müller H., Naumann F., “Discovering conditional inclusion dependencies”, *21st ACM international conference on Information and knowledge management (CIKM '12) (ACM, New York, NY, USA)*, 2012, 2094–2098.
- [17] Gómez-López F. T., Gasca R. M., Pérez-Álvarez J. M., “Compliance validation and diagnosis of business data constraints in business processes”, *Information Systems*, **48** (2015), 26–43.
- [18] Visser J., “Coupled Transformation of Schemas, Documents, Queries, and Constraints”, *Electronic Notes in Theoretical Computer Science*, **200**:3 (2008), 3–23.
- [19] Garmany J., Walker J., Clark T., *Logical Database Design Principles*, CRC Press, Auerbach Publications, New York, NY, USA, 2005, 69 pp.
- [20] Lopes S., Petit J. M., Toumani F., “Discovering interesting inclusion dependencies: application to logical database tuning”, *Information Systems*, **27**:1 (2002), 1–19.
- [21] Levene M., Loizou G., “Null Inclusion Dependencies in Relational Databases”, *Information and Computation*, **136**:2 (1997), 67–108.
- [22] Levene M., Loizou G., “The additivity problem for data dependencies in incomplete relational databases”, *Semantics in Databases*, *Lecture Notes in Computer Science.*, **189**, Springer-Verlag Berlin Heidelberg, 1998, 136–169.
- [23] Köhler H., Link S., “Inclusion Dependencies Reloaded”, *The 24th ACM International on Conference on Information and Knowledge Management (CIKM '15) (ACM, New York, NY, USA)*, 2015, 1361–1370.

Zykin V. S., Zykin S. V., "Analysis of Typed Inclusion Dependencies with Null Values", *Modeling and Analysis of Information Systems*, **24**:2 (2017), 155–167.

DOI: 10.18255/1818-1015-2017-2-155-167

Abstract. Null values have become an urgent problem since the creation of the relational data model. The impact of the uncertainty affects all types of dependencies used in the design and operation of the database. This fully applies to the inclusion dependencies, which are the theoretical basis for referential integrity on the data. Attempts to solve this problem contain inaccuracy in the statement of the problem and its solution. The errors in formulation of the problem can be associated with the use in the definition of untyped inclusion dependencies, which leads to permutations of the attributes, although, the attributes in database technology are identified by name and not by their place. In

addition, linking with the use of the inclusion dependencies of heterogeneous attributes, even of the same type, is a sign of lost functional dependencies and leads to interaction of inclusion dependencies and non-trivial functional dependencies. Inaccuracies in the solution of the problem are contained in the statements of axioms and the proof of their properties, including completeness. In this paper we propose an original solution of this problem only for typed inclusion dependencies in the presence of Null values: a new axiom system is proposed, its completeness and soundness are proved. On the basis of inference rules we developed an algorithm for the construction of a not surplus set of typed inclusion dependencies. The correctness of the algorithm is proved.

Keywords: database, inclusion dependences, axiomatic, NULL values

About the authors:

Vladimir S. Zykin, orcid.org/0000-0002-6492-2464, graduate student,
Omsk State Technical University,
11 Mira av., Omsk 644050, Russia, e-mail: vszykin@mail.ru

Sergey V. Zykin, orcid.org/0000-0002-0576-2149, doctor of sciences in technic, professor,
Sobolev Institute of Mathematics SB RAS,
13 Pevtsova str., Omsk 644043, Russia, e-mail: szykin@mail.ru