

Метод построения схем реляционных баз данных, использующий семантическую информацию

© И.П. Убалехт
ОмГТУ,
Омск
ivan@ubaleht.com

Аннотация

В данной работе предлагается метод построения избыточных и непротиворечивых схем реляционных баз данных, использующий семантическую информацию. Предлагаемый метод сочетает в себе элементы проектирования схем баз данных методом синтеза (проектирование от атрибутов и функциональных зависимостей к отношениям) и элементы проектирования от сущностей предметной области. В статье предлагается на инфологическом уровне использовать семантически расширенные аналоги функциональных зависимостей, предлагается способ формализации процесса получения функциональных зависимостей, предлагается алгоритм автоматизированного построения схем реляционных баз данных и прототип графической нотации.

1 Введение

В настоящее время существует достаточно много работ, посвящённых исследованию принципов и методов формирования схем реляционных баз данных (БД). Несмотря на это процесс формирования схем реляционных БД остаётся недостаточно формализованным. Формализация процесса формирования схем реляционных баз данных остаётся важной задачей теории реляционных баз данных. Помимо этого актуальной является задача разработки методов формирования схем реляционных БД, имеющих такие характеристики как: высокая автоматизация процесса построения схем; развитые средства, обеспечивающие пользователю (проектировщику схем БД) наглядность и удобство управления процессом построения схем БД.

В данной работе предлагается метод построения избыточных и непротиворечивых схем БД. Цель метода – получение схемы реляционной БД,

находящейся, как минимум в третьей нормальной форме (3НФ).

Ключевые свойства метода, предлагаемого в данной работе:

- высокая формализация процесса получения схем БД;
- наличие выразительных средств, обеспечивающих наглядность и удобство построения схем БД для конечного пользователя;
- высокий уровень автоматизации построения схем БД.

2 Обзор работ и публикаций

Кратко рассмотрим существующие в настоящее время подходы к проектированию схем реляционных БД. Рассмотрим новые публикации в этой области, модели, близкие к модели, на которой основан предложенный в статье метод.

Можно выделить четыре основных подхода к проектированию схем реляционных БД:

1. Построение схем через декомпозицию отношений. Данный подход известен со времён создания реляционной модели данных и описан во многих источниках. Сущность подхода заключается в последовательной декомпозиции (нормализации) первоначально заданных отношений через применение к ним ряда правил [13, 16, 17, 23]. Метод применяется на даталогическом уровне.

2. Построение схем реляционных баз данных посредством синтеза схемы из множества функциональных зависимостей (ФЗ). В дальнейшем будем называть этот подход методом синтеза. Данный подход также давно известен [2, 16, 17, 23] и также используется на даталогическом уровне – на уровне реляционной базы данных (рис. 1, б).

3. Подходы, ведущие проектирование от семантики предметной области (ПрО). Данная группа подходов включает в себя большое количество различных моделей [7,8,13].

Общее для всех подходов и моделей данной группы то, что они используются на инфологическом уровне и с их помощью получается концептуальная схема ПрО. На инфологическом

уровне может быть формализована только часть элементов и взаимосвязей ПрО, например, при использовании классической ER-модели (рис. 1, а), но может быть формализовано и большинство элементов и взаимосвязей, например, при использовании UML модели с языком OCL, ORM модели [7, 8] или расширенной ER модели, предложенной в статьях [3, 10] (рис. 1, в). При использовании семантического метода, предложенного в настоящей работе, также большинство элементов и взаимосвязей ПрО формируются на инфологическом уровне (рис. 1, в).

4. Уточнение и нормализация схемы на основе информации, получаемой из уже готовой и заполненной (либо заполненной частично) базы данных. Будем называть этот подход построением схемы базы данных обратным методом. Суть подхода в том, что в данных уже заполненной базы данных либо в промежуточном прототипе базы данных ищутся закономерности, в том числе и те, которые можно расценивать как ФЗ. На основе этих найденных закономерностей оптимизируется схема базы данных. Этот процесс может быть итерационным и совмещённым с подходом № 2 (методом синтеза), как описано в статье [21].

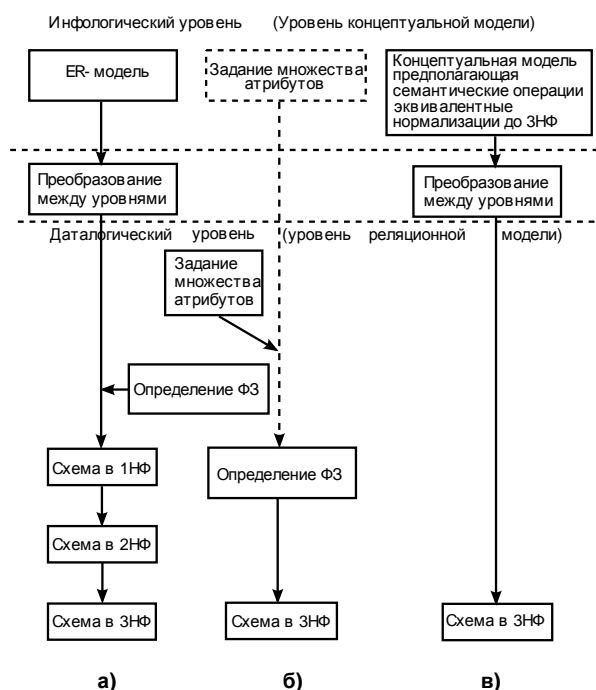


Рис. 1. Процесс получения схем реляционных баз данных: а) при использовании ER-модели, б) при использовании метода синтеза, в) при использовании моделей, позволяющих определять большинство элементов и взаимосвязей на инфологическом уровне

Метод, предлагаемый в данной работе, можно в основном отнести к подходу № 3, но он содержит элементы и других описанных выше подходов.

В настоящее время активно создаются новые методы, модели и подходы позволяющие получать избыточные и непротиворечивые схемы реляционных БД. Большинство из них можно отнести к подходам, ведущим проектирование от семантики ПрО в представленной выше классификации. Кратко рассмотрим некоторые современные публикации в этой области, имеющие параллели с методом, предлагаемым в данной работе.

Модель «Объект-Роль» (Object-Role Model) предоставляет очень развитые выразительные средства для проектирования БД на инфологическом уровне и для вербализации данных. Данная модель существует довольно давно, однако постоянно выходят новые публикации, посвящённые исследованию этой модели [7, 8].

Семантическая модель «Сущность – Связь – Отображение» является развитием идей ER-модели. Так же, как предыдущая модель, имеет хорошие выразительные средства для детального описания на инфологическом уровне закономерностей исследуемых ПрО [12]. В предлагаемом в данной работе методе так же, как и в двух вышеупомянутых моделях, большое значение играет инфологический уровень, семантическая информация и высокий уровень формализации взаимосвязей между элементами концептуальной схемы.

В статьях [3, 10] предлагается современный вариант расширения ER-диаграмм – Articulated Entity Relationship (AER) diagram, который помогает в автоматизированной нормализации схем БД. В методе AER-диаграмм и в методе, предложенном в данной работе есть параллели, это формализация ФЗ на инфологическом уровне и расширение ER-нотации.

В ряде статей [18–20] Панченко Б.Е. вводится понятие реляционного каркаса, с помощью которого можно автоматизировано синтезировать высоко-нормализованные и безаномальные схемы БД. Важную роль в этой модели играют многозначные зависимости.

В работе [22] рассматривается собственная специальная реляционная модель, использующая «отношения-сущности» и «отношения-связывания» для получения схем БД в доменно-ключевой нормальной форме.

Статьи [14, 15] посвящены формализации процесса нормализации схем реляционных БД. Для решения задачи получения оптимальных схем БД в этих статьях предлагается адаптировать известные алгоритмы, например, алгоритм Балаша. В настоящей работе для получения оптимальных схем БД также предлагается алгоритм, который в будущем предлагается описать, например, как алгоритм информированного поиска путей на имплицитном графе.

Таблица. Множество состояний бинарной связи RS

Тип связи	Количественное отношение	Описание
Тип 1		Один-к-одному. Для каждого значения из A_{first} имеется строго одно значение из A_{second} , для каждого значения из A_{second} имеется строго одно значение из A_{first} .
Тип 2		Многие-к-одному. Для каждого значения из A_{first} имеется строго одно значение из A_{second} , для каждого значения из A_{second} имеется не менее одного значения из A_{first} .
Тип 3		Один-к-многим. Для каждого значения из A_{first} имеется не менее одного значения из A_{second} , для каждого A_{second} имеется строго одно значение из A_{first} .
Тип 4		Многие-ко-многим. Для каждого значения из A_{first} имеется не менее одного значения из A_{second} , для каждого значения из A_{second} имеется не менее одного значения из A_{first} .

3 Модель построения схем баз данных с учётом семантической информации

Основные характеристики предлагаемого метода:

– сочетание элементов проектирования схем БД через синтез отношений (проектирование от атрибутов и ФЗ к отношениям) и элементов проектирования через декомпозицию (проектирование от сущностей предметной области);

– использование двухстрочных отношений для определения семантики связей между атрибутами (выявление элементов связанности). При переходе на даталогический уровень элементы связанности преобразуются в ФЗ. Таким образом, на инфологическом уровне предоставляется механизм для формализации ФЗ;

– алгоритм получения безаномальных схем БД, используемый в данном методе можно охарактеризовать как алгоритм локальной декомпозиции с глобальным синтезом схем БД;

– графическая нотация с помощью которой удобно выявлять семантические аналоги ФЗ на инфологическом уровне. Нотация ориентированна на встраивание элементов данного подхода в

наиболее распространённые на практике ER-диаграммы в нотации Баркера.

Формализуем основные понятия модели.

Определение 1. Элементом области связанности или просто элементом связанности будем называть тройку $EAI \langle A_{first}, A_{second}, RS \rangle$, где $A_{first}, A_{second} \in A, A \subseteq U; RS \in \mathbf{RS}$. A – множество атрибутов входящих в область связанности; U – множество всех атрибутов заданной ПрО; \mathbf{RS} – множество состояний бинарной связи между атрибутами $A_i, A_j \in A$, где i, j любые целые числа от 1 до n, n – мощность множества A .

В таблице представлено \mathbf{RS} – множество всех вариантов состояний бинарной связи RS . Связь RS присутствует между каждыми двумя атрибутами в области связанности. Графическое отображение связей RS между атрибутами представлено рис. 2 б. В данной работе не рассматриваются варианты связи RS учитывающие возможность неопределённых значений (null) атрибутов.

Определение 2. Пусть A – произвольное множество атрибутов $\in U; U$ – множество всех атрибутов заданной ПрО; $A \subseteq U; A_{first}, A_{second} \in A$. Областью связанности AI будем называть такое

множество троек $EAI < A_{first}, A_{second}, RS >$ (множество элементов связанности), что: A_{first}, A_{second} каждой тройки EAI является элементом декартова произведения $A \times A$. Количество троек $EAI \in AI$ равно количеству элементов множества $A \times A$.

Рассмотрим по шагам схему предлагаемого метода получения избыточных и непротиворечивых схем БД. На шаге 1 производится определение первоначальной схемы областей связанности $AI(S_{start})$. На шаге 2 производится установка связей между атрибутами внутри элементов связанности (выявление семантических аналогов ФЗ). На шаге 3 – запуск алгоритма распределения элементов связанности по областям связанности (см. алгоритм). Шаг 2 и 3 выполняются итеративно до тех пор, пока не сработают условия, определяющие, что исходная схема эквивалентна состоянию как минимум ЗНФ. На шаге 4 осуществляется перевод конечной нормализованной схемы областей связанности $AI(S_{finish})$ с инфологического уровня в схему реляционных отношений на даталогическом уровне. На даталогическом уровне схема отношений будет как минимум в ЗНФ.

Шаг 1. На данном шаге имеется множество областей связанности, построенных исходя из неформального понимания проектировщиком предметной области. Основой некоторой формализации на данном этапе может служить проектная документация. Проектировщик определяет области связанности – это наборы атрибутов, между которыми предполагается существование некоторых связей.

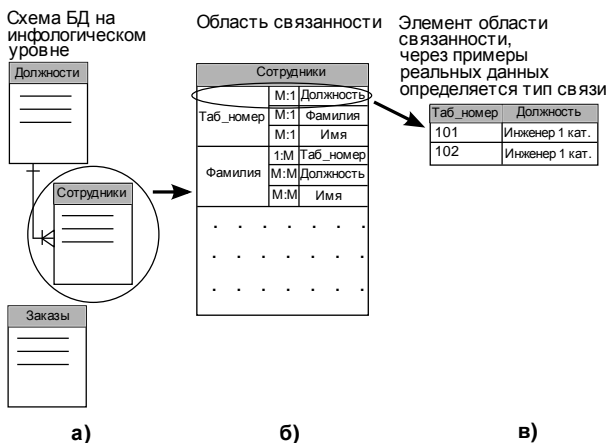


Рис. 2. Определение связи между атрибутами *Tab_Номер* и *Должность* через использование двухстрочного отношения: а) схема БД на инфологическом уровне; б) детализирована область связанности *Сотрудники*; в) детализирован элемент связанности *Tab_Номер – Должность* области связанности *Сотрудники*, связь между атрибутами которого определяется через использование двухстрочного отношения, заполненного реальными данными

Шаг 2. В определённых на шаге 1 областях связанности в соответствии с таблицей устанавливаем типы связей между атрибутами внутри всех элементов связанности, то есть определяем все элементы связанности.

Для определения типов связей между атрибутами внутри элементов связанности используется формальная модель. Эта модель включает: интерактивный графический язык, близкий по возможностям языку Query-By-Example (QBE), логику двухстрочных отношений [17] (рис. 2) и реляционное исчисление.

Рассмотрим данную модель подробнее, модель должна осуществлять следующее: на инфологическом уровне используя семантическую информацию позволять пользователю задавать элементы связанности внутри областей связанности; поддерживать удобную графическую нотацию; строго формализовать переход от элементов связанности на инфологическом уровне к ФЗ на даталогическом уровне.

Покажем корректность перехода от элементов связанности на инфологическом уровне к ФЗ на даталогическом уровне. Каждый элемент связанности можно представить как двухстрочное отношение. Опираясь на [17] определим понятие двухстрочного логического отношения.

Определение 3. Пусть r – отношение со схемой R , A – атрибут в R , отношение r будем называть двухстрочным логическим отношением, если оно содержит в точности два кортежа t_1 и t_2 и если с отношением r ассоциировано Ψr – присваивание истинностных значений атрибутам из r . Ψr – является функцией из R в {истина, ложь}, такой что

$$\Psi r(A) = \begin{cases} \text{истина, если } t_1(A) = t_2(A), \\ \text{ложь, если } t_1(A) \neq t_2(A). \end{cases}$$

Таким образом, элемент связанности можно представить как двухстрочное логическое отношение. В качестве значений в строках этого отношения могут быть данные, соответствующие семантике ПрО. Далее нужно формализовать переход от двухстрочного логического отношения к ФЗ в произвольных конечных отношениях.

В соответствии с [17] приведём теорему об эквивалентности ФЗ в отношениях с произвольным (конечным) множеством кортежей, ФЗ в двухстрочных логических отношениях и импликацией, доказательство см. в [17].

Теорема. Пусть F – множество ФЗ над схемой R и $A \rightarrow B$ есть зависимость над R . Тогда следующие утверждения эквивалентны: из F следует $A \rightarrow B$ для произвольных (конечных) отношений; из F следует $A \rightarrow B$ для двухстрочных логических отношений; из F следует $A \rightarrow B$ как логическая формула (импликация).

Рассмотрим на примере описанные выше преобразования этого шага. На рис. 2 представлен пример предметной области. Заданы области

связанности, которые можно соотнести с сущностями: *Сотрудники*, *Должности*, *Заказы* (рис. 2, а). Пользователь просматривает область связанности – *Сотрудники* (на рис. 2, б выделена область связанности *Сотрудники*, показан момент анализа пользователем этой области связанности). Пользователю достаточно задать только часть связей между атрибутами в области связанности, остальная часть связей может быть автоматически достроена через применение правил вывода. Далее на рис. 2 в показан момент анализа пользователем отношения (связи) между атрибутами *Таб_Номер* и *Должность*. При анализе пользователь заполняет две строки примерами реальных данных, соответствующих семантике и ограничениям атрибутов *Таб_Номер* и *Должность* в данной ПрО. Из рис. 2 видно, что в соответствии с введёнными реальными данными между атрибутами *Таб_Номер* и *Должность* у пользователя получилась связь типа 2 – многие-к-одному (см. таблицу), что может быть представлено следующим логическим выражением (термом):

$$\{t^{(2)} | (\exists u)(\text{Сотрудники}(t) \wedge \text{Сотрудники}(u) \wedge (t[\text{Таб_номер}] \neq u[\text{Таб_номер}] \vee t[\text{Должность}] = u[\text{Должность}]))\}$$

Получившаяся связь многие-к-одному и представленное логическое выражение эквивалентны тому, что в соответствии с приведенной выше теоремой между атрибутами *Таб_Номер* и *Должность* в произвольном конечном отношении на даталогическом уровне будет ФЗ.

Как видно из примера выше, связи, семантически выраженные в двухсторонних отношениях, можно рассматривать как логические выражения, которые в соответствии с приведённой теоремой эквивалентны ФЗ в произвольных конечных отношениях. Таким образом, области связанности с установленными связями моделируют будущие отношения с ФЗ, которые появятся при переходе на даталогический уровень.

Шаг 3. На предыдущих шагах задана начальная конфигурация областей связанности. Внутри всех областей связанности и для всех элементов связанности определены типы связей. На данном шаге элементы связанности должны стать условиями формирования (преобразования) областей связанности. Под влиянием элементов связанности, заданных на предыдущем шаге, с областями связанности могут автоматически производиться следующие действия: декомпозиция областей связанности; отсоединение не связанных с данной областью связанности атрибутов; поиск недостающих атрибутов и присоединение их к областям связанности.

Эти действия можно алгоритмизировать, алгоритм по преобразованию областей связанности назовём *SemanticNormalization*.

Применение алгоритма *SemanticNormalization* можно отразить следующим выражением:

$$(AI(S_{start}), Rules, Heuristic()) \Rightarrow AI(S_{finish}),$$

где $AI(S_{start})$ – множество областей связанности заданных на шаге 1, образующих схему S_{start} ;

Rules – множество правил, которые определяют какие из операций (рис. 3) нужно применить к текущей области связанности AI , состав правил в *Rules* в данной статье не уточняется, правила в *Rules* могут образовывать формальную логическую систему;

Heuristic() – эвристическая функция;

$AI(S_{finish})$ – множество областей связанности, образующее схему S_{finish} после работы алгоритма *SemanticNormalization*.

Цель алгоритма – за конечное число операций по преобразованию областей связности перейти от начальной ненормализованной схемы $AI(S_{start})$ к конечной схеме областей связанности $AI(S_{finish})$ такой, что при переводе её на даталогический уровень она образует схему БД как минимум в 3НФ.

Ниже приведен алгоритм *SemanticNormalization*.
Вход: $AI(S)$ и *Rules* *SemanticNormalization*($AI(S_{start})$, *Rules*)

```

begin
  while (BufferOfAttributes is Change) or
  (BufferOfAttributes != Empty) or ( $\exists AI$  is Change)
  begin
    //перебор всех областей связанности
    foreach(AI in AI)
      begin
        //применение операций и правил к областям
        //связанности и получение множества
        //областей связанности с новой схемой
        //AI(S_new)
        AI(S_new) = AppliedOperationsToCurrentAI(
          Operations = { DeleteAttributeOperation(),
            DecompositionOperation() }, Rules);
      end foreach
      //проверка связанности атрибутов из буфера на
      //присоединённых атрибутов BufferOfAttributes
      //с областями связанности из AI(S_new)
      if(Heuristic is possible)
        AI(S_newest) = Heuristic(AI(S_new), Buffer);
      else
        //если эвристика не возможна, то полный
        //перебор
        foreach(Element in BufferOfAttributes)
          begin
            foreach(AI in AI)
              AddAttributeOperation(AI, Element);
            end foreach
          end foreach
        end if
      end while
    end
  
```

Алгоритм `SemanticNormalization` работает, используя множество строгих правил вывода `Rules` и эвристику. Множество правил `Rules` определяет, какую из перечисленных ниже операций нужно применить к текущей области связанности `AI`. Далее опишем операции, которые можно производить над областями связанности (рис. 3): операция отсоединения лишнего атрибута от области связанности – `DeleteAttributeOperation` (рис. 3, а); операция добавления атрибута из буфера несвязанных атрибутов `BufferOfAttributes` к текущей области связанности – `AddAttributeOperation` (рис. 3, б); операция разделения области связанности – `DecompositionOperation` (рис. 3, в).

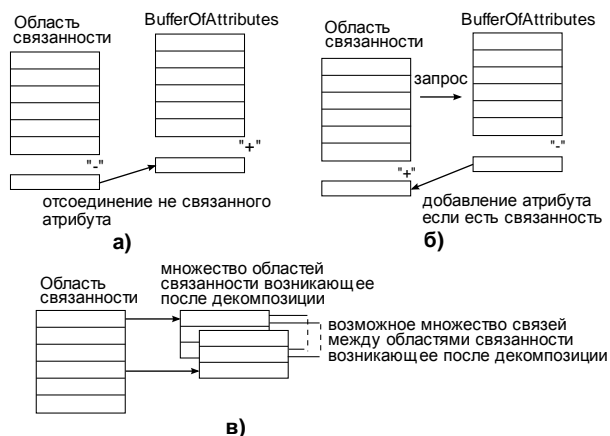


Рис. 3 Операции, производимые над областями связанности: а) `DeleteAttributeOperation`, б) `AddAttributeOperation`, в) `DecompositionOperation`

Алгоритм прекращает свою работу тогда когда:

- в буфере `BufferOfAttributes` нет элементов;
- после очередной итерации к буферу атрибутов `BufferOfAttributes` не добавлено ни одного элемента;
- применяя к каждой области связанности множество правил `Rules` невозможно произвести ни одной операции из операций, представленных на рис. 3.

После завершения работы алгоритма `SemanticNormalization` формируется множество областей связанности со схемой S . При переводе множества областей связанности образующих конечную схему S , во множество реляционных отношений на даталогическом уровне, это множество отношений образует схему БД находящуюся как минимум в 3НФ.

Шаг 4. Множество областей связанности со схемой S_{finish} с инфологического уровня переводится во множество отношений на даталогическом уровне. Этот процесс хорошо исследован см., например, источники [13, 16] и подробно в этой работе не описывается.

4 Заключение

В статье рассмотрен метод построения избыточных и непротиворечивых схем

реляционных БД, использующий семантическую информацию. Новизна представленного метода заключается в следующем:

- метод сочетает элементы проектирования схем БД через синтез отношений (проектирование от атрибутов и ФЗ к отношениям), это придаёт методу строгость и элементы проектирования через декомпозицию (проектирование от сущностей предметной области), это предаёт методу наглядность;

- представленный метод можно отчасти рассматривать как семантическое расширение метода синтеза. В проанализированных современных источниках [1, 4–6, 9, 11] при практическом использовании метода синтеза совершенно не формализовывался процесс получения исходного множества ФЗ. Представленный метод строго формализует процесс получения исходного множества ФЗ через применение двухстрочных отношений с примерами реальных данных и логических формул на инфологическом уровне. Такой способ формализации определения ФЗ представляется вполне хорошо реализуемым в программном продукте;

- представлены элементы графической нотации, которые могут использоваться в диаграммах на инфологическом уровне для работы пользователя с семантической информацией. Представленные элементы графической нотации хорошо сочетаются с ER-диаграммами в нотации Баркера (данная нотация ER-диаграмм является в настоящее время наиболее используемой) и могут хорошо дополнять ER-диаграммы при практической реализации данного метода в программном продукте. Предлагаемая нотация может успешно конкурировать с другими современными вариантами расширения ER-диаграмм, например, с нотацией представленной в [3, 10], которая является более громоздкой;

- разработан алгоритм с локальной декомпозицией и глобальным синтезом областей связанности, использующий семантическую информацию и работающий на инфологическом уровне, нормализующий схему будущей БД до состояния эквивалентного 3НФ.

Литература

- [1] Bahmani A., Naghibzadeh M., Bahmani B. Automatic database normalization and primary key generation // 21th Canadian Conference on Electrical and Computer Engineering. – Niagara Falls, Canada, 2008.
- [2] Bernstein P.A. Synthesizing Third Normal Form Relations from Functional Dependencies // ACM Transactions on Database Systems (TODS), 1976. Vol. 1, Iss. 4. P. 277–298.
- [3] Dhabe P.S., Patwardhan M.S., Deshpande A.A., Dhore M.L., Barbadekar B.V., Abhyankar H.K. Articulated entity relationship (AER) diagram for

- complete automation of relational database normalization // *International Journal of Database Management Systems (IJDMS)*, 2010. Vol. 2, No. 2. P. 84–100.
- [4] Dongare Y.V., Dhabe P.S., Deshmukh S.V. RDBNorma: – A semi-automated tool for relational database schema normalization up to third normal form // *International Journal of Database Management Systems (IJDMS)*, 2011. Vol. 3, No. 1. P. 133–154.
- [5] Du H., Wery L. Micro: A normalization tool for relational database designers // *Journal of Network and Computer Application*, 1999. Vol. 22, No. 4. P. 215–232.
- [6] Georgiev N. A web based environment for learning normalization of relational database schemata. Masters thesis. – Umea, Umea university, 2008.
- [7] Halpin T. *Conceptual Schema and Relation Database Design*. – 2th ed. – Sydney: Prentice-Hall of Australia Pty., Ltd, 1995.
- [8] Halpin T., Morgan T. *Information Modeling and Relational Databases*. – 2th ed. Kaufmann Publishers, 2008. 943 p.
- [9] Kung H., Tung H. A web-based tool to enhance teaching/learning database normalization. 9th Annual Conference of the Southern Association for Information Systems (SAIS). – Jacksonville, USA, 2006.
- [10] Patwardhan M.S., Dhabe P.S., Deshpande A.A., Londhe S.G., Dhore M.L., Abhyankar H.K. Diagrammatic approach for complete automation of relational database normalization at conceptual level // *International Journal of Database Management Systems (IJDMS)*, 2010. Vol. 2, No. 4. P. 132–151.
- [11] Yazici A., Ziya K. (2007), JMathNorm: A database normalization tool using mathematica // *International Conference on Computational Science 2007 (ICCS 2007)*. – Beijing, China, 2007.
- [12] Бабанов А.М. Семантическая модель «Сущность – Связь – Отображение» // *Вестник Томского государственного университета. Управление, вычислительная техника и информатика*, 2007. №1. С. 77–91.
- [13] Дейт К. Дж. *Введение в системы баз данных* : пер. с англ. – 7-е изд. – М.: Издат. дом «Вильямс», 2001. 1072 с.
- [14] Клименко И.В. Метод построения семейства максимальных транзитивно независимых множеств атрибутов // *Вопросы современной науки и практики. Университет им. В.И. Вернадского*, 2011. Т. 35, вып. 4. С. 70–78.
- [15] Клименко И.В. Модификация алгоритма Балаша для решения задачи нормализации реляционных баз данных // *Вопросы современной науки и практики. Университет им. В.И.Вернадского*, 2012. Т. 37, вып. 1. С. 43–49.
- [16] Кузнецов С.Д. *Основы баз данных: учебное пособие*. – 2-е изд. – М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. 484с.
- [17] Мейер Д. *Теория реляционных баз данных: пер. с англ.* – М.: Мир, 1987. 608 с.
- [18] Панченко Б.Е. Об алгоритме синтеза реляционного каркаса. Постановка задачи и формализация // *Компьютерная математика*. 2012. № 1. С. 84–93.
- [19] Панченко Б.Е. Каркасное проектирование доменно-ключевой схемы реляционной базы данных // *Кибернетика и системный анализ*. 2012. № 3. С. 174–187.
- [20] Панченко Б.Е. Алгоритм синтеза реляционного каркаса - неформальное описание // *Проблемы управления и информатики*. 2013. № 1. С. 83–103.
- [21] Радченко В.А., Мальков Ю.А., Балюк С.А., Горпиненко Ю.С. Синтез логической схемы реляционной базы данных на основе выявления множества функциональных зависимостей // *Системы обработки информации*. 2011. Т. 95, вып. 5. С. 218–224.
- [22] Тукеев У.А., Алтайбек А.А. Концептуальная, логическая модели и алгоритм проектирования баз данных в доменно-ключевой нормальной форме // *Труды 13-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL'2011)*. – Воронеж, 2011. С. 119–125.
- [23] Ульман Дж. *Основы систем баз данных: пер. с англ.* – М.: Финансы и статистика, 1983. 334 с.

Method of Creation of Schemes of Relational Databases Using Semantic Information

Ivan P. Ubaleht

This paper proposes a method for creating non-redundant and consistent schemes of relational databases using semantic information. The proposed method combines elements of the database schema design method of its synthesis (design from attributes and functional dependencies to relations) and design proceeding from the domain entities. This paper proposes to use semantically extended analogs of functional dependencies on the Infological level; a method for definition of functional dependencies; algorithm of automatic creation of schemes of relational databases and a prototype of graphical notation.