

## О ПОЛНОТЕ И ЕДИНСТВЕННОСТИ УНИВЕРСАЛЬНОГО КАРКАСА В РЕЛЯЦИОННОЙ МОДЕЛИ ДАННЫХ

Б.Е. ПАНЧЕНКО, И.Н. ПИСАНКО

Введено представление о путях нормализации в универсальном каркасе реляционных баз данных и о топологии этих путей. Сформулирована и доказана теорема о полноте и единственности реляционного каркаса.

### ВВЕДЕНИЕ

Методики конструирования схемы реляционных баз данных ограничиваются классической парадигмой Кодда [1, 2] со всеми ее расширениями, уточнениями, модификациями и обобщениями, время от времени появляющимися вплоть до настоящего момента [3]. Попытка Дейта и Дарвена [4] создать формальную «надстройку» над реляционной моделью, отвечающую современным реалиям и требованиям, остается абстрактным решением, не выходящим в область практического применения. Уже традиционным стало построение либо модели «сущность-связь» (ER-модель) [5] либо т.н. семантической объектной модели (SOM) и последующий «перевод» получаемых орграфов-схем или соответственно семантических структур в реляционные схемы [6, 7]. Практика такой «трансляции» считается эффективной не только в методическом плане, но и по затратам времени и усилий на построение логической структуры баз данных. Вместе с тем эту практику отличает известная локальность построений, отсутствие универсальности, приводящее к сложностям при модификации структуры базы данных, вплоть до необходимости тотального редизайна структуры [8, 9]. Локальность ER/SOM-построений заключается, прежде всего, в работе с фиксированным графом-схемой либо с заданными множествами семантически определенных (или «четких») объектов. Это сказывается на гибкости и модифицируемости таких построений.

Дополнительные затруднения концептуального характера связаны с нечеткостью ключевых определений «атрибут», «сущность» и «объект», а также с отсутствием общепринятой аксиоматики. Строгие алгебраические определения понятий по типу [10, 11], которые для ER/SOM-построений являются ключевыми, фактически отсутствуют. Это указывает на то, что ER/SOM-построения не приспособлены для создания универсальных структурных схем, которые были бы формально полны и непротиворечивы. И ER-модель и SOM-модель дают не столько удобный и эффективный, сколько привычный и безальтернативный инструментарий для конструирования логической структуры баз данных.

От гипотетических универсальных построений можно ожидать большей эффективности как минимум в части модифицируемости логической структуры баз данных, а само существование базовой универсальной схемы

(назовем ее  $\alpha$ -схемой) является не только интуитивно допустимым, но и необходимым. В данной работе мы отмечаем лишь некоторые особенности  $\alpha$ -схемы, а также выявляем и анализируем несколько важных соответствий между свойствами  $\alpha$ -схемы и традиционных реляционных схем.

## ПОСТАНОВКА ЗАДАЧИ

Пусть  $\{a_i\}$  — конечное множество атомарных сущностей (объектов) в смысле [9]. Там под атомарностью понимается унарное формирование множества атрибутов этой сущности (объекта), то есть одноместный предикат, объединяющий это множество. Тогда  $n$ -местным  $j$ -предикатом  $P_j^n(a_i)$ , принимающим истинное значение,  $P_j^n(a_i) = 1$  (многоместным предикатом), очевидно, моделируется формирование составных (пост-связных) сущностей. Комбинирование атрибутов  $a_i$  порождает полную совокупность  $\{r_l\}$  реляционных схем, представляющих собой именованные отношения. В свою очередь, комбинирование реляционных схем  $r_l$  порождает полную совокупность схем  $\{D_j\}$  реляционных баз данных:

$$\{a_i\} \Rightarrow \{r_l\} \Rightarrow \{D_j\}. \quad (1)$$

Выражение (1) представляет собой ядро  $\alpha$ -схемы. Сразу следует отметить ключевое отличие между  $\alpha$ -схемой и ER/SOM-построениями: для заданного множества атрибутов в ER/SOM-моделях никогда не задействованы *полные* совокупности отношений  $\{r_l\}$  и схем  $\{D_j\}$ , а только их подмножества. В этом и состоит локальность ER/SOM-моделей. Такая локальность объясняется тем, что выбор атрибутов и конструирование «начальных» отношений (и их совокупностей) при «трансляции» из ER/SOM-моделей определяются семантикой соответствующей предметной области. Другими словами, ER/SOM-конструкции являются решениями частных задач.

Выбранная начальная схема  $D^{(0)}$ , как правило, требует нормализации в силу наличия функциональных и/или многозначных зависимостей между отдельными атрибутами и/или их совокупностями. По сути, нормализация является преобразованием от начальной схемы  $D^{(0)}$  базы данных (как совокупности реляционных схем) к некоторой конечной схеме  $D^{(k)}$ , каждое из отношений которой удовлетворяет известным критериям нормализации. В большинстве случаев (для нетривиальных множеств атрибутов и нетривиальных зависимостей между ними) нормализация представляет собой многошаговую процедуру, т.е. вначале нормализуется одно отношение, а затем другое; при этом *вся* схема базы данных, естественно, изменяется:

$$D^{(0)} \rightarrow D^{(1)} \rightarrow \dots \rightarrow D^{(k)}. \quad (2)$$

Основной акцент при проектировании логической структуры баз данных ставится именно на нормализацию, т.е. на получение конечной схемы  $D^{(k)}$ , все отношения которой находятся в требуемой нормальной форме.

Поскольку для  $\alpha$ -схемы совокупность  $\{D_j\}$  является полной (т.е. представляет собой множество всех возможных комбинаций всех возможных реляционных схем  $r_l$ , построенных комбинированием атрибутов  $a_i$ ), она уже содержит необходимую конечную схему  $D^{(k)}$ . В силу этого полную совокупность  $\{D_j\}$  можно считать универсальным «реляционным каркасом» (relational framework) для *любых* схем баз данных, которые могут быть построены на фиксированном множестве атрибутов  $\{a_i\}$  при *любых* зависимостях между атрибутами. Обсуждение понятия «каркас схем баз данных» (в дальнейшем — просто *каркас*) и логические предпосылки к его введению имеются в работах [9, 12]

Объект нашего исследования — каркас  $\{D_j\}$  и его свойства.

Классический способ нормализации — это декомпозиция отношений. Суть декомпозиции состоит в замене каждой реляционной схемы  $r^*$ , не удовлетворяющей критериям требуемой нормальной формы, на другие реляционные схемы  $\{r_l\} \sim r^*$ , в совокупности эквивалентные исходной схеме  $r^*$ , но по отдельности соответствующие критериям нормальной формы. Всю совокупность схем баз данных в каркасе  $\{D_j\}$  удобно проиндексировать. Любую нормализацию (2), а значит и соответствующую декомпозицию отношений, идентифицируем последовательностью  $Q(j_0, j_1, \dots, j_k)$  индексов схем баз данных, которую будем далее называть *путем нормализации*. Для изучения свойств каркаса  $\{D_j\}$  целесообразно проанализировать «топологию» путей  $Q$ . Метод данного исследования — анализ топологии путей нормализации в каркасе  $\{D_j\}$ .

Каждый путь нормализации определяется двумя факторами: начальной схемой  $D^{(0)}$  базы данных, а также зависимостями в отношениях начальной схемы (и, возможно, последующих схем). Эти функциональные и/или многозначные зависимости, как правило, тесно связаны с ключами (элементарными либо композитными), которые можно выделить в экземплярах отношений. Поскольку всякое нетривиальное отношение может содержать несколько зависимостей (равно как и несколько ключей), для каждой начальной схемы можно предположить наличие *нескольких* путей нормализации. Далее, не для всяких начальной  $D^{(0)}$  и конечной  $D^{(k)}$  схем может существовать путь нормализации. Все это указывает на то, что для нетривиальных зависимостей на множестве атрибутов  $\{a_i\}$  топология путей нормализации в каркасе  $\{D_j\}$  может оказаться достаточно сложной.

**Цель работы** — исследовать топологию путей нормализации реляционного каркаса. Эта топология относится к структуре, обладающей свойствами полноты и единственности, а именно к каркасу  $\{D_j\}$ . И хотя полнота и единственность каркаса интуитивно очевидны, тем не менее, мы приведем строгое доказательство этих характеристик.

Нормализация посредством декомпозиции реляционных схем является широко распространенным методом устранения аномалий в реляционных

базах. Декомпозиция каждого конкретного отношения строится на основе данных о зависимостях между атрибутами экземпляра *этого* отношения, т.е. может быть применена только локально. Предположим, что зная топологию путей нормализации (например, особенности влияния мощностей отношений в каркасе на принадлежность отношений заданным нормальным формам, связь между множеством зависимостей в некотором отношении и областью в каркасе, куда приведет декомпозиция этого отношения и т.п.) удастся существенно упростить процедуры построения и модификации конечных логических структур реляционных баз данных. Во-первых, за счет использования новых унифицированных методик дизайна этих структур. И, во-вторых, за счет эксплуатации тех (топологических) особенностей унифицированной структуры, которые ранее были совершенно незаметны при решении частных задач. Например, введение адекватной меры  $\|Q\|$  для путей нормализации позволяет говорить о выборе оптимальных путей нормализации схем баз данных.

### ПРЕОБРАЗОВАНИЯ

Пусть в экземплярах  $R_i$  любого из отношений  $r_i$  каждый атрибут  $a_i$  принимает значения  $[a_i]_p$  из некоторого непустого множества (домена)  $A_i$ ,  $\forall p : [a_i]_p \in A_i$ . Здесь индекс  $p$  идентифицирует любое из возможных значений атрибута в домене  $A_i$ , причем  $\max p(A_i)$  характеризует алгебраическую мощность домена  $A_i$ , т.е. является его кардинальным числом.

Рассмотрим преобразование  $\{a_i\} \Rightarrow \{r_i\}$ . В отличие от традиционного синтеза реляционных схем  $r_i$  как декартовых произведений *отдельных* доменов, преобразование  $\{a_i\} \Rightarrow \{r_i\}$  приводит к построению полной совокупности реляционных схем.

Вначале для заданной предметной области формируется множество атрибутов —  $\{a_i\}$  и множество соответствующих доменов —  $\{A_i\}$ . Как правило, атрибуты просто перечисляются, а соответствующие им домены соотносятся с определенными типами, формами, определяющими свойствами. Тем не менее, каждый новый элемент в домене мы индексируем. Атрибуты и их домены индексируются ( $i$ ) в произвольном порядке. При этом:

$$\forall A_i \exists p^* \neq p : [a_i]_{p^*} = [a_i]_p, \quad (3)$$

т.е. в каждом домене все значения атрибутов единственны.

Ясно, что в процессе дизайна и сопровождения реляционной базы состав множества атрибутов  $\{a_i\}$ , равно как и состав отдельных доменов  $A_i$ , может изменяться. Подобная модификация может быть обусловлена как выделением в предметной области дополнительных атрибутов и новых объектов, так и исключением атрибутов и объектов, ставших неактуальными. Поэтому удобно говорить о *состоянии* реляционной базы — совокупности  $\langle \{a_i\} \{A_i\} \rangle$ , актуальной в некоторый момент времени. Заметим, что схема

реляционной базы «эволюционирует» от состояния к состоянию за счет изменения  $\{a_i\}$ , но не  $\{A_i\}$ . Модификация самой базы (а не ее схемы) — добавление новых кортежей, изменение или удаление существующих — может привести к изменению кардинальных чисел отдельных доменов  $A_i$ , а также к изменению нормальной формы, в которой находится то или иное отношение. Понятие зависимости между атрибутами (а значит и понятие о соответствии нормальным формам) относится только к *экземплярам* реляционных схем, но не к самим схемам. Заметим, что на этой стадии построения каркаса  $\{D_j\}$  никакие сведения о зависимостях между атрибутами *не учитываются*, что существенно упрощает процедуру синтеза.

Далее конструируется полная совокупность  $\{r_l\}$  реляционных схем. Традиционно отношения интерпретируются как подмножества декартовых произведений доменов [6]. Указанная интерпретация отношений, несомненно, формально верна. Вероятно, истоки этой интерпретации следует искать в представлениях об «экземплярах» отношений, в пределах которых и вводятся функциональные либо многозначные зависимости. Действительно, в подавляющем большинстве реальных ситуаций элементы доменов  $A_k$  являются не перечисляемыми, а типизируемыми. Поэтому построение (как перечисление) всех кортежей декартовых произведений доменов, как правило, практически невозможно. Здесь и далее мы будем использовать построения, основанные на комбинировании атрибутов  $a_i$ , а не их значений в экземплярах отношений.

Итак, пусть имеются некоторые множества  $A$  и  $B$ , имеющие кардинальные числа  $p_A$  и  $p_B$  соответственно. Введем оператор  $L$ , действующий следующим образом:

$$\{C_k\} = L_A(B), C_k = B \cup a, a \in A \setminus B. \quad (4)$$

Оператор  $L$  продуцирует индексированную совокупность  $\{C_k\}$  множеств  $C_k$ , содержащих все элементы множества  $B$  и один из элементов множества  $A$ , не принадлежащий  $B$ . Здесь и далее множество  $A$  будем называть источником, множество  $B$  — базовым, а оператор  $L$  — оператором роста. Заметим, что базовое множество может представлять собой *совокупность* множеств (в этом случае оператор роста действует на каждое из множеств базовой совокупности).

**Пример 1.** Для источника  $A = \{1, 2, 3, 4, 5, 6, 7\}$  и базового множества  $B = \{1, 3, 5, 6, 8\}$  получаем

$$\begin{aligned} L_A(B) &= L_A\{1, 3, 5, 6, 8\} = \{\{1, 2, 3, 5, 6, 8\}, \{1, 3, 4, 5, 6, 8\}, \{1, 3, 5, 6, 7, 8\}\}, \\ L_A^2(B) &= L_A(L_A(B)) = L_A\{\{1, 2, 3, 5, 6, 8\}, \{1, 3, 4, 5, 6, 8\}, \{1, 3, 5, 6, 7, 8\}\} = \\ &= \{\{1, 2, 3, 4, 5, 6, 8\}, \{1, 2, 3, 5, 6, 7, 8\}, \{1, 3, 4, 5, 6, 7, 8\}\} \\ L_A^3(B) &= L_A(L_A^2(B)) = \\ &= L_A\{\{1, 2, 3, 4, 5, 6, 8\}, \{1, 2, 3, 5, 6, 7, 8\}, \{1, 3, 4, 5, 6, 7, 8\}\} = \{1, 2, 3, 4, 5, 6, 7, 8\}, \end{aligned}$$

$$L_A^4(B) = L_A(L_A^3(B)) = L_A(\{1, 2, 3, 4, 5, 6, 7, 8\}) = \{1, 2, 3, 4, 5, 6, 7, 8\} = L_A^3(B). \quad \square$$

Рассмотрим основные свойства оператора  $L$ .

Прежде всего, оператор роста никогда не уменьшает мощность базового множества. Если  $A \setminus B \neq \emptyset$ , то мощность базового множества возрастает на единицу, что дает мощность любого из множеств  $C_k$  совокупности  $\{C_k\}$ , т.е.  $\forall k: p(C_k) = p_B + 1$ . Базовое множество является подмножеством любого из порожденных  $C_k$ . Фактически оператор роста обеспечивает «перекачку» элементов от источника к базовому множеству. В силу того, что реальный источник предполагается конечным (т.е.  $p_A$  является конечным числом),  $m$ -кратное действие  $L_A^m(B)$  оператора роста *всегда* приводит к «насыщению» базового множества. Этот эффект выражается в том, что  $C_k \rightarrow B$  при  $A \setminus B = \emptyset$ , т.е., когда базовое множество уже содержит источник ( $A \subseteq B$ ; источник «исчерпан»). Пороговым является значение  $M = p(A \setminus B)$ ; при  $A \setminus B = \emptyset$  получаем  $M = p(\emptyset) = 0$ , что указывает на «насыщение» базового множества для всех  $m \geq M$ .

В конструктивном отношении важны две особенности оператора  $L$ : во-первых, оператор роста позволяет получить все возможные комбинации атрибутов  $a_i$ , и, во-вторых, полная совокупность этих комбинаций оказывается *структурированной*. Действительно, каждое  $m$ -кратное действие  $L_A^m(B)$  порождает совокупности  $\{C_i\}_m$  множеств  $C_i$ , имеющих равные кардинальные числа, но взаимно отличающихся (если множеств  $C_i$  оказывается более одного) не более чем  $m - 1$  элементами. Любую такую совокупность  $\{C_i\}_m$  будем называть  $m$ -уровнем. Ясно, что множества  $C_i$  каждого  $m$ -уровня могут быть идентифицированы  $m$  числами, например, индексами элементов  $a_i$  источника  $A$ . Если же структура получаемой совокупности  $\{C_i\}_m$  множеств несущественна, то можно представить ее в виде  $\{C_k\}$  простым перечислением порожденных множеств.

**Пример 2.** Пусть элементы рассмотренного выше источника  $A = \{1, 2, 3, 4, 5, 6, 7\}$  проиндексированы следующим образом:  $a_1 = 2$ ,  $a_2 = 5$ ,  $a_3 = 1$ ,  $a_4 = 3$ ,  $a_5 = 6$ ,  $a_6 = 4$  и  $a_7 = 7$  (здесь контент элементов совершенно не имеет значения). Для базового множества  $B = \{1, 3, 5, 6, 8\}$  получаем трехуровневую совокупность множеств, а именно:

1)  $L_A(B)$  порождает тройку

$$C_1 \equiv C(1) = \{1, 2, 3, 5, 6, 8\} = B \cup a_1;$$

$$C_2 \equiv C(6) = \{1, 3, 4, 5, 6, 8\} = B \cup a_6;$$

$$C_3 \equiv C(7) = \{1, 3, 5, 6, 7, 8\} = B \cup a_7;$$

2)  $L_A^2(B)$  порождает тройку

$$C_4 \equiv C(1, 6) = \{1, 2, 3, 4, 5, 6, 8\} = B \cup \{a_1, a_6\} = C(1) \cup a_6 = C(6) \cup a_1;$$

$$C_5 \equiv C(1, 7) = \{1, 2, 3, 5, 6, 7, 8\} = B \cup \{a_1, a_7\} = C(1) \cup a_7 = C(7) \cup a_1;$$

$$C_6 \equiv C(6, 7) = \{1, 3, 4, 5, 6, 7, 8\} = B \cup \{a_6, a_7\} = C(6) \cup a_7 = C(7) \cup a_6;$$

3)  $L_A^3(B)$  порождает единственное множество

$$\begin{aligned} C_7 &\equiv C(1, 6, 7) = \{1, 2, 3, 4, 5, 6, 7, 8\} = B \cup \{a_1, a_6, a_7\} = \\ &= C(6, 7) \cup a_1 = C(1, 7) \cup a_6 = C(1, 6) \cup a_7 = B \cup (A \setminus B) = A \cup B, \end{aligned}$$

после чего наступит состояние «насыщения» базового множества.  $\square$

Ясно, что каждое «дочернее» множество может иметь несколько «родительских» множеств. Мощность  $A \setminus B$ , как правило, велика (т.е. велико число атрибутов, выделяемых в предметной области), поэтому в большинстве случаев граф, отражающий эту взаимосвязь между множествами, не будет планарным. Весь формализм теории графов (связность, диаметр, маршруты и т.п.), естественно, применяются для конструируемой совокупности множеств.

Рассмотрим, как можно использовать оператор роста  $L$  при построении каркаса схем реляционных баз данных и оперировании с этим каркасом.

Пусть базовое множество  $B$  является пустым ( $B = \emptyset$  и  $p_B = 0$ ), а источник  $A$  представляет собой индексированную совокупность  $\{a_i\}$  атомарных атрибутов ( $p_A = N$ , где  $N$  — количество объектов-сущностей, выделяемых в некоторой предметной области). Имеем  $A \setminus B = A$  и  $\{C_i\} = L_A(\emptyset) \equiv L_A^1$ ,  $C_i = a_i \in A$ . Таким образом, в случае однократного применения оператора роста ( $L^1$ ) пустое базовое множество можно не учитывать, рассматривая лишь совокупность атрибутов  $A$ . Необходимо отметить, что при многократном действии оператора роста ( $L^{m>1}$ ) аргументом будет совокупность множеств, полученных в результате предыдущих действий оператора. Следовательно, сам оператор роста определяется индуктивно.

В данной работе базовое множество вводится для того, чтобы эффективно объединить логические структуры нескольких баз данных, построенных для взаимно пересекающихся предметных областей. Другую важную область применения базового множества — модификацию логической структуры базы данных при изменении совокупности атрибутов предметной области — также целесообразно рассмотреть в дальнейших работах.

Итак, действие  $L_A^1$  для пустого базового множества порождает  $S_1 = N$  унарных отношений  $\{a_i\}$ , однозначно соотносимых с атрибутами  $a_i$  из источника  $A$ . Действие  $L_A^2$  оперирует с совокупностью  $\{a_i\}$  как с аргументом, порождая элементы 2-уровня, т.е. все возможные бинарные отношения. При этом  $B = L_A^1$  и  $S_2 = \frac{1}{2}(N-1)N$ . Далее, совокупность бинарных отношений является аргументом для действия  $L_A^3$ , которое порождает элементы 3-уровня, т.е. все возможные тернарные отношения. Аналогично,

$B = L_A^2$  и  $S_3 = \frac{1}{6}(N-2)(N-2)N$ . Вообще,  $m$ -уровень ( $m \leq N$ ), который синтезируется действием  $L_A^m$  с базовой совокупностью  $B = L_A^{m-1}$  в качестве аргумента, содержит  $S_m = \frac{N!}{m!(N-m)!}$   $m$ -арных отношений. При  $m = N$  получаем  $S_N = 1$ , что указывает на «исчерпывание» совокупности атрибутов  $A$ .

Общее число отношений в синтезируемом каркасе составляет

$$S(N) = \sum_{m=1}^N S_m = \sum_{m=1}^N \frac{N!}{m!(N-m)!} = 2^N - 1. \quad (5)$$

Например,  $S(5) = 31$ ,  $S(10) = 1023$ ,  $S(20) = 1048575$ ,  $S(30) = 1073741823$ . Очевидно, что в практически значимых случаях, когда в предметной области выделяются десятки и сотни атрибутов объектов, каркас не может быть реализован *целиком*. Каркас, как естественным образом структурированная совокупность множеств атрибутов, является аналогом множества «состояний» физической системы, в пределах которого разворачивается весь процесс синтеза логической структуры реляционных баз данных. Для логической структуры, состоящей из  $R \leq S(N)$  реляционных схем  $\{r_i\}$ , удобно ввести понятие *заполнения*  $\xi_R$  каркаса

$$\xi_R = \frac{R}{S(N)} = \frac{R}{2^N - 1}. \quad (6)$$

Как правило,  $\xi_R \ll 1$ , причем вопрос о распределении  $\xi_R(N)$  для реальных реляционных баз данных пока что остается открытым.

## ПОЛНОТА И ЕДИНСТВЕННОСТЬ РЕЛЯЦИОННОГО КАРКАСА

Имеет место следующая *теорема полноты и единственности*.

**Теорема.** Каркас, порождаемый оператором роста на конечном множестве атомарных атрибутов, единственен и полон.

**Доказательство.** Докажем вначале полноту каркаса.

Пусть заданы источник  $A$  и базовое множество  $B$ . Пусть существует некоторое множество  $C = \{c_i\}$  такое, что  $C \supseteq B$  и  $\exists c \in A \setminus B$ , но  $\forall m: C \neq L_A^m(B)$ . Тогда, из определения оператора роста,  $C \neq B \cup C_m^*$ , где  $\forall m: C_m^* \subseteq A \setminus B$ , а значит  $\bar{\exists} c \in C_m^*$ . Но это противоречит допущению  $\exists c \in A \setminus B$ . Следовательно, действие  $L_A^m(B)$ , порождающее множество  $C$ , действительно существует. Поэтому для каркаса имеем  $R = S(N)$ , т.е. его заполнение  $\xi_R = 1$ .

Доказательство единственности каркаса основано на его полноте. Пусть имеется пара каркасов  $\{C_k\} = L_A^m(B)$  и  $\{C_k^*\} = L_A^m(B)$ . В силу полноты каждого из них имеем  $\{C_k\} \setminus \{C_k^*\} = \emptyset$  и  $\{C_k^*\} \setminus \{C_k\} = \emptyset$ , следовательно,

$\{C_k\} = \{C_k^*\}$ , т.е. каркасы идентичны или, другими словами, на конечном множестве атрибутов сущностей оператор роста порождает единственный каркас.  $\square$

Единственность и полнота совокупности отношений, порождаемых оператором  $L_A$  из множества  $A$  атрибутов сущностей, выделяемых в предметной области, указывает на исключительное положение каркаса в процессе разработки логической структуры баз данных. Действительно, любое отношение на заданном множестве атрибутов принадлежит каркасу. Более того, результат операций над элементами каркаса также будет элементом каркаса. Это свойство замкнутости непосредственно следует из свойства полноты. Поэтому каркас является вполне естественным (хотя и формальным) компонентом при построении *унифицированной* схемы синтеза логических структур баз данных. Важно отметить, что сам каркас как структурированная совокупность всех возможных множеств атрибутов (в алгебраическом смысле каркас является *булеаном* источника  $A$ ) предельно избыточен ( $\xi = 1$ ), поэтому для его эффективного использования необходим развитый инструментарий «навигации по каркасу», преобразований, редукций, локализаций и оперирования в целом, который будет рассмотрен в дальнейшем.

Итак, преобразование  $\{a_i\} \Rightarrow \{r_i\}$  в  $\alpha$ -схеме (1) заключается в том, что заданное множество атрибутов порождает структурированную совокупность отношений. Следующее преобразование  $\alpha$ -схемы, а именно  $\{r_i\} \Rightarrow \{D_j\}$ , состоит в синтезе всех возможных схем реляционных баз данных. Ясно, что для этого можно использовать оператор роста  $L$  (4) с *полной* совокупностью отношений в качестве аргумента.

Пусть базовое множество  $B$  вновь пусто ( $B = \emptyset$  и  $p_B = 0$ ), а источником  $A$  является индексированная совокупность  $\{r_i\}$  отношений ( $p_A = S(N) = 2^N - 1$  — полное число синтезированных отношений). В этом случае действие  $L_A^1$  порождает совокупность схем баз данных, образованных единичными отношениями (фактически это и есть каркас отношений, рассмотренный выше), действие  $L_A^2$  — всеми возможными парами отношений, действие  $L_A^m$  — всеми возможными  $m$ -компонентными совокупностями отношений ( $m \leq p_A$ ). Общее число схем реляционных баз данных составит  $S_D = 2^{S(N)} - 1 = 2^{2^N - 1} - 1$ . Естественно, что всякая реляционная база данных имеет в основе *единственную* схему, с необходимостью являющуюся элементом каркаса схем. Это объясняется тем, что согласно теореме о полноте и единственности реляционного каркаса совокупность схем баз данных, порождаемая оператором роста, также обладает свойствами единственности и полноты.

**Пример 3.** Пусть в предметной области выделены 3 атомарных атрибута  $a, b, c$ . Совокупность этих атрибутов образует источник  $A$ . Каркас отношений состоит из  $2^3 - 1 = 7$  отношений, а именно:  $a, b, c$  ( $L_A^1$  — унар-

ные отношения),  $ab$ ,  $bc$ ,  $ac$  ( $L_A^2$  — бинарные отношения),  $abc$  ( $L_A^3$  — единственное тернарное отношение). Каркас схем баз данных состоит из  $2^7 - 1 = 127$  элементов, синтезируемых из 7-элементного каркаса отношений как аргумента оператора роста  $L$ , например,  $a$ ,  $abc$  (для действия  $L_A^1$ ),  $a + ac$ ,  $abc + ac$ ,  $ac + ab$  (для действия  $L_A^2$ ),  $abc + a + bc$ ,  $a + b + c$ ,  $ab + ac + b$  (для действия  $L_A^3$ ).

## ВЫВОДЫ

Между каркасом отношений и каркасом схем баз данных имеется существенное отличие. Элементом каркаса отношений является отношение как некоторое множество атрибутов предметной области. Элементом каркаса схем баз данных является множество отношений как реляционных схем, т.е. некоторая схема базы данных.

Очевидно, что структура каркаса схем баз данных гораздо богаче структуры каркаса отношений. Действительно, в каркасе отношений мы различаем  $m$ -уровни, группируя отношения с равными кардинальными числами; в силу свойства «насыщения» оператора роста можно выделить только  $N$  таких уровней. Элементами каркаса схем баз данных могут быть отношения разных уровней, причем само количество элементов схемы может быть разным.

Каркас схем баз данных также занимает исключительное положение при построении унифицированной  $\alpha$ -схемы. Действительно, *любая* схема реляционной базы данных, синтезируемой для заданного множества атрибутов, является элементом этого каркаса. Поэтому решение *любой* задачи синтеза или задачи модификации уже содержится в каркасе схем баз данных. В [9, 12] на основе *реляционного ключевого каркаса* получена *универсальная логическая модель данных* как особое подмножество всех реляционных моделей.

Следует отметить, что до сих пор мы не учитывали зависимости между атрибутами. Так как эти зависимости целиком определяются семантикой предметной области, для которой создается (или под которую модифицируется) схема реляционной базы данных, то каркасы отношений и схем совершенно *не зависят* от какой бы то ни было специфики предметной области, являясь универсальными конструкциями. Для нахождения этого решения предполагается использовать концепцию «путей нормализации».

## ЛИТЕРАТУРА

1. Codd E.F. A Relational Model of Data for Large Shared Data Banks // Communications of the ACM. — 1970. — **13**, № 6. — P. 377–387.
2. Codd E.F. Extending the database relational model to capture more meaning // ACM Transactions on Database Systems. — 1979. — **4**, № 4. — P. 397–434.
3. Codd E.F. The Relational Model For Database Management. Version 2. — NY.: Addison-Wesley Publishing Co, 1990. — 538 p.

4. *Darwen H., Date C.J.* The third manifesto // ACM SIGMOD Records. — 1995. — **24**, № 1. — P. 39–49.
5. *Chen P.P.* The Entity-Relationship Model: toward a unified view of data // ACM Trans on Data base systems. — 1976. — **1**, № 1. — P. 9–36.
6. *Silberschatz A., Korth H.F., Sudarshan S.* Database system concepts // PRC edition, McGraw-Hill, Boston, 5<sup>th</sup> edit., 2005. — 1024 p.
7. *Simsion G.C., Witt G.C.* Data modeling essentials // Morgan Kaufmann Publishers, 3<sup>th</sup> edit. — 2005. — 532 p.
8. Патент № 63036. Способ расположения данных в компьютерном хранилище, обеспечивающий модифицируемость ею структуры / Панченко Б.Е. // Промышленная собственность. — 2004. — № 1. — С.3–134.
9. *Панченко Б.Е.* О шунтировании многозначительных зависимостей в реляционной модели данных // Проблемы программирования. — 2010. — № 2–3. — С. 428–433.
10. *Курош А.Г.* Общая алгебра. — М.: Физматлит, 1979. — 150 с.
11. *Плоткин Б.И.* Универсальная алгебра, алгебраическая логика и базы данных. — М.: Наука, 1991. — 448 с.
12. *Панченко Б.Е., Писанко И.Н.* Свойства реляционного каркаса на множестве семантически атомарных предикатов // Кибернетика и системный анализ. — 2009. — № 6. — С. 120–129.

*Поступила 22.04.2009*