

С. С. ТАНЯНСКИЙ, д-р техн. наук, проф. ХНУРЭ, Харьков

МЕТОД ПОСТРОЕНИЯ СЕМАНТИЧЕСКИ ЭКВИВАЛЕНТНЫХ СОСТОЯНИЙ НЕОДНОРОДНЫХ БАЗ ДАННЫХ

Пропонується метод побудови множини можливих станів бази даних при модифікації структури даних. Визначено умови припустимих операцій і введено оператор переходу станів бази даних. Запропоновано відповідні алгоритми та доведена їхня коректність, адекватність й ефективність.

Ключові слова: база даних, семантика даних, стан, модифікація структури, цілісність даних, обмеження, логічне правило.

Предлагается метод построения множества возможных состояний базы данных при модификации структуры данных. Определены условия допустимых операций и введен оператор перехода состояний базы данных. Предложены соответствующие алгоритмы и доказана их корректность, адекватность и эффективность.

Ключевые слова: база данных, семантика данных, состояние, модификация структуры, целостность данных, ограничение, логическое правило.

Propose a method of constructing a set of possible states of a database by modifying the structure of the data. The conditions of the permitted operations and introduce the operators of state transition of the database. Suggest appropriate algorithms and prove their correctness, adequacy and effectiveness.

Keywords: database, data semantics, state, modifying the structure, data integrity, constraint logic rule.

Введение. Разработка и внедрение независимых систем автоматизации деятельности организации приводят к несогласованности и семантической неоднородности данных в различных подсистемах. Для эффективного управления современной организацией требуется интегрированная информационная система (ИС), позволяющая работать со всем объемом накопленной в организации информации.

Выбор того или иного сценария интеграции, помимо функциональных требований прикладной системы, зависит от таких факторов, как качество и актуальность данных, доступность исходного кода, интенсивность работы с подсистемой, особенности сетевого доступа и другие технические характеристики. Кроме этого оценка семантических свойств ИС играет важную роль при построении интегрированных баз данных (БД).

Моделирование семантики БД – одна из наиболее трудных задач в проектировании и обработке данных. Для этой цели в различных моделях используются ограничения целостности, которые выражают свойства, накладываемые предметной областью, определяют связи между компонентами и описывают поведение БД.

Обобщенную модель БД можно выразить набором положений, определяющих структурные свойства хранящихся в БД объектов и их поведение. При аксиоматическом подходе модель задается описанием своей

структуры, где семантика описывается аксиомами или ограничениями целостности, определяющие допустимые состояния БД.

Анализ основных достижений и литературы. За время своего существования системы баз данных прошли путь от громоздких БД, организованных как системы коллективного пользования, к гибким распределенным информационным системам, которые функционируют без непосредственного участия конечных пользователей.

Практические задачи определяют необходимость проведения научных исследований по широкому кругу вопросов проектирования и организации функционирования распределенных систем баз данных. В частности, задачи, которые рассматриваются в статье, основываются на результатах исследований теории реляционных БД. Среди них – построение формального аппарата описания ограничений целостности в форме правил. В работах Э. Тененбаума [1], Дж. Фридла [2], рассматриваются вопросы построения эффективных логических схем БД, в работах Д. Мейера [3] и О. Горчинской [4] – организация функционирования БД с неполной информацией, решения проблем, связанных с выполнением запросов в среде интегрированных БД представлены в работах Л. Калиниченко [5] и С. Кузнецов [6].

В процессе работы с ИС, а также при работе с уже построенной БД возникает необходимость уточнения или расширения модели предметной области, расширения информации об объектах, которые определяются предметной областью и отражаются в соответствующей модели. То есть, возникает необходимость манипулирования структурой данных и выполнения различных операций над элементами БД.

Цель исследования, постановка задачи. Проводимые в статье исследования направлены на разработку метода и алгоритмов таких модификации БД, которые допустимы только в том случае, если они не нарушают заданных ограничений целостности. При построении интегрированной ИС ограничения целостности данных локальных БД могут не совпадать, при этом выполнение запросов не должно нарушать эти ограничения. Для решения этого вопроса необходимо иметь возможность восстановить ограничения при их нарушении или так изменять ограничения, чтобы они не влияли на обобщенные требования ИС.

Будем рассматривать данные как некоторый информационный объект I , который нужно добавить или удалить из базы данных DB , при этом необходимо сохранить состояние БД $\tau_i \in T = \{\tau_0, \tau_1, \dots, \tau_n\}$ адекватное заданным ограничением $I = \{I_j \downarrow I_i\}$.

Поддерживаемая в современных системах управления базами данных (СУБД) работа транзакций, для которых любое нарушение целостности приводит к восстановлению начального состояния БД τ_0 , недостаточно гибкая и значительно ограничивает возможности модификации данных.

С целью построения системы управления интегрированной ИС постановка решаемой в статье задачи может быть сформулирована как

разработка метода и алгоритмов обеспечения целостности структуры БД при изменении семантики данных, и исследования их свойств для поддержки актуальности информации в интегрированной среде, а также разработка метода корректных обновлений семантики, удовлетворяющих заданным ограничениям.

Оператор перехода состояний базы данных. В работах [7, 8, 9] рассматривалась проблема обновлений структуры БД, при которых требовалось обеспечить целостность данных и сохранить согласованность структур. Были определены меры измерения изменений семантики БД и сформулированы условия эквивалентности семантик модифицированной БД.

Исследуя вопросы изменения непосредственно данных, определим условия корректности состояний БД при выполнении запросов на модификацию данных. Будем основываться на принципах сохранения информации, то есть при выполнении изменений в БД необходимо максимально сохранить существующие данные, а затем добавить новые.

Определение 1. Пусть τ_0, τ_1, τ_2 , – некоторые состояния БД. Состояние τ_1 “ближе” (в смысле минимальности изменений элементов множества τ_1) к состоянию τ_0 , чем τ_2 , если выполняются условия:

- 1) $\tau_0 \cap \tau_2 \subset \tau_0 \cap \tau_1$ или
- 2) $(\tau_0 \cap \tau_2 = \tau_0 \cap \tau_1)$ и $(\tau_1 - \tau_0) \subset (\tau_2 - \tau_0)$.

Два состояния τ_0 и τ_1 будем называть смежными, если нет состояния τ_2 , которое находится к τ_0 ближе, чем τ_1 .

Определим действия, которые по исходному состоянию БД τ_0 и запросу на обновление ставят в соответствие новое состояние τ_1 , удовлетворяющее ограничениям целостности $I = \{l_j \dashv l_i\}$, где l_i – информационный объект (атрибут) БД, и являющееся ближайшим к исходному состоянию τ_0 .

Определение 2. Пусть обновления Λ и ограничения I совместны (то есть не содержат никакого объекта одновременно с его отрицанием). Оператор $\mathfrak{G}^{(I, \Lambda)}$ на множестве состояний T называется оператором модификации, если для каждого состояния $\tau_i \in T (i = \overline{1, n})$ выполняются условия:

- 1) $\mathfrak{G}(I, \Lambda) \in \dot{\tau}(I, \Lambda)$, где $\dot{\tau}$ – некоторое новое состояние, определяемое как множество $\dot{\tau}(I, \Lambda) = \{\tau_i \mid (\tau_0 \rightarrow \tau_i) \in \delta^{\tau}(I, \Lambda)\}$ и $\delta^{\tau}(I, \Lambda)$ – множество всех отображений $\delta: \tau_0 \rightarrow \tau_1$, на которых выполнена модификация Λ ;
- 2) после применения $\mathfrak{G}(I, \Lambda)$ состояние τ_1 и начальное состояние τ_0 являются смежными по отношению $\dot{\tau}(I, \Lambda)$.

Содержательно первое условие определяет тот факт, что запрос должен быть выполнен и полученное состояние должно удовлетворять ограничениям целостности. Второе условие фиксирует непосредственно модификацию, то есть выполняются только необходимые (минимальные) изменения в начальном состоянии, чтобы смежное состояние удовлетворяло первому условию.

Основываясь на определении 2, определим свойства полноты оператора модификаций для неоднородных БД.

Теорема 1. Пусть $\mathfrak{G}^{(I,\Lambda)}$ – оператор модификации и пусть τ_0 и τ_1 – состояния БД, такие, что $(\tau_0, \tau_1) \mid \Rightarrow I$. Тогда существует обновление Λ' , совместное с I , такое, что $\tau_1 = \mathfrak{G}^{(I,\Lambda')}(\tau_0)$.

Доказательство. Предположим, что Λ' выполняет обновление элементов: добавить элементы $(\tau_1 - \tau_0)$ и удалить элементы $(\tau_0 - \tau_1)$. Очевидно, что такие обновления удовлетворяют отображению $\delta: \tau_0 \rightarrow \tau_1$, таким образом, $(\tau_0 \rightarrow \tau_1) \in \delta^\tau(I, \Lambda')$. Также легко проверить, что для любого состояния $\tau_2 \in \dot{\tau}(I, \Lambda')$ выполняются следующие условия: $(\tau_0 - \tau_1) = \Lambda'^{del}$ (обновление удаление), $(\tau_1 - \tau_0) = \Lambda'^{ins}$ (обновление добавление) и $\Lambda'^{del} \subseteq \tau_0 - \tau_2$, $\Lambda'^{ins} \subseteq \tau_2 - \tau_0$. Следовательно, состояния τ_0 и τ_1 являются смежными, и не существует никакого другого состояния $\tau_2 \in \dot{\tau}(I, \Lambda')$, одновременно смежного с τ_0 и τ_1 . При этом τ_1 является единственно возможным результатом применения оператора $\mathfrak{G}^{(I,\Lambda')}(\tau_0)$.

Доказательство закончено.

Из данного утверждения вытекает, что для заданного множества ограничений I в некотором состоянии τ_0 область значений оператора модификации $\mathfrak{G}^{(I,\Lambda)}$ при отображении $\delta: \tau_0 \rightarrow \tau_1$ совпадает с множеством информационных объектов состояния τ_1 .

Рассмотрим множество всех возможных состояний $\tau_1 \in \dot{\tau}(I, \Lambda')$, смежных с τ_0 относительно $\dot{\tau}(I, \Lambda')$, в обозначении $T^{I,\Lambda'}(\tau_0)$. Согласно теореме 1, мощность $|T^{I,\Lambda'}(\tau_0)| = 1$, хотя в общем случае мощность может быть больше 1. Рассмотрим пример, в котором размер множества возможных состояний $T^{I,\Lambda'}(\tau_0)$ может экспоненциально зависеть от размера его параметров.

Пусть $I = \{l_c \dashv l_{a_i}, l_{b_i}\}$, где $i = \overline{1, n}$ и, соответственно, $\tau_0 = \{l_{a_1}, \dots, l_{a_n}, l_{b_1}, \dots, l_{b_n}, l_c\}$. Допустим, что необходимо удалить элемент l_c , то есть необходимо выполнить действие Λ' , тогда очевидно, что $T^{I,\Lambda'}(\tau_0)$ включает все состояния БД вида $\{l_{x_1}, \dots, l_{x_m} \mid l_{x_i} \in \{l_{a_i}, l_{b_i}\}, i = \overline{1, n}\}$, следовательно, $|T^{I,\Lambda'}(\tau_0)| = 2^n$.

Таким образом, для обобщения полученных результатов, необходимо решить задачу нахождения элемента из $T^{I,\Lambda'}(\tau_0)$ по заданным ограничениям целостности I , обновляемым информационным объектам Λ и текущему состоянию БД τ_0 , а также сформулировать условия нахождения всех возможных состояний БД из $T^{I,\Lambda'}(\tau_0)$.

Построение множества допустимых состояний базы данных. Для построения алгоритмов поиска элементов из $T^{I,\Lambda'}(\tau_0)$ и построения всех

возможных состояний БД при модификации данных определим некоторые свойства выполнимости ограничений целостности.

Лемма 1. Для неоднородных БД и для любых ограничений целостности выполняются следующие условия:

- 1) если для некоторого отображения $\hat{\delta}: \hat{\tau}_0 \rightarrow \hat{\tau}_1$ существует замыкание (то есть множество правил, которые можно вывести из заданного множества) η^{+} для состояний $(\hat{\tau}_0 \rightarrow \hat{\tau}_1)$, то $\eta^{+}(\hat{\tau}_0 \rightarrow \hat{\tau}_1) \models I$;
- 2) если существует отображение $\delta: \tau_0 \rightarrow \tau_1$ и $(\tau_0 \rightarrow \tau_1) \models I$, то для информационных объектов перехода состояний $(\hat{\tau}_0 \rightarrow \hat{\tau}_1) \subseteq (\tau_0 \rightarrow \tau_1)$ справедливо $\eta^{+}(\hat{\tau}_0 \rightarrow \hat{\tau}_1) \subseteq (\tau_0 \rightarrow \tau_1)$ и $\eta^{+}(\hat{\tau}_0 \rightarrow \hat{\tau}_1) \models I$;
- 3) если модифицированные информационные объекты Λ совместны с ограничениями I для некоторого состояния БД τ_0 и $\eta^{(\Lambda)}(\tau_0 \rightarrow \emptyset) = (\tau_0 \rightarrow \tau_1)$, то τ_1 является состоянием с наименьшим количеством изменений среди $\dot{\tau}(I, \Lambda)$;
- 4) если состояние $\tau' \in T^{(\Lambda)}(\tau_0)$, то $(\tau_0 \rightarrow \tau') = \eta^{(\Lambda)}(\tau_0 \rightarrow \tau_0 \cap \tau')$.

Доказательство. Доказательство условий 1-го и 2-го вытекает непосредственно из определения замыкания [10] и того факта, что замыкание вычисляется за конечное число шагов. Доказательство 3-го условия основывается на том, что из совместности ограничений I и модификаций Λ следует $\dot{\tau}(I, \Lambda) \neq \emptyset$. Тогда, если τ_2 – некоторое произвольное состояние БД из $\dot{\tau}(I, \Lambda)$, то $(\tau_0 \rightarrow \emptyset) \subseteq (\tau_0 \rightarrow \tau_2)$ и согласно условию 2 данной леммы $\eta^{+}(\tau_0 \rightarrow \emptyset) = (\tau_0 \rightarrow \tau_1) \subseteq (\tau_0 \rightarrow \tau_2)$ и $\eta^{+}(\tau_0 \rightarrow \emptyset) \models I$. Таким образом, $\tau_1 \subseteq \tau_2$, то есть τ_1 - состояние с наименьшим количеством изменений среди $\dot{\tau}(I, \Lambda)$. Для доказательства 4 условия также используем условие 2 этой леммы. Так как $(\tau_0 \rightarrow \tau_0 \cap \tau') \subseteq (\tau_0 \rightarrow \tau')$, то $\eta^{(\Lambda)}(\tau_0 \rightarrow \tau_0 \cap \tau') = (\tau_0 \rightarrow \tau_1) \subseteq (\tau_0 \rightarrow \tau')$ и $(\tau_0 \rightarrow \tau_1) \models I$. Из того, что оператор замыкания монотонный, следует, что $\tau_0 \cap \tau' \subseteq \tau_0 \cap \tau_1$, а так как τ_0 и τ' смежные, то $\tau_0 \cap \tau' = \tau_0 \cap \tau_1$ и $\tau' - \tau_0 = \tau_1 - \tau_0$, то есть $\tau_0 = \tau_1$.

Доказательство закончено.

Используя рассмотренные условия, построим алгоритм для нахождения состояния БД из $T^{(\Lambda)}(\tau_0)$.

Алгоритм 1. Построение корректного перехода состояний БД.

Вход. Состояние БД τ_0 , модификации $\Lambda = \{\Lambda^{ins}, \Lambda^{del}, \Lambda^{upd}\}$, совместные с I для τ_0 .

Выход. Состояние БД τ_1 .

Метод.

Шаг 1. Фиксируем множество добавляемых информационных объектов как $\lambda^{ins} = \Lambda^{ins} \cup \{l_{i_1} \mid \exists l_{i_2} ((l_{i_1}, l_{i_2}) \in \Lambda^{upd})\}$.

Шаг 2. Фиксируем множество удаляемых информационных объектов как $\lambda^{del} = \Lambda^{del} \cup \{l_{i_2} \mid \exists l_{i_1} (l_{i_1} \in \lambda^{ins} \text{ и } (l_{i_1}, l_{i_2}) \in \Lambda^{upd})\}$.

Шаг 3. Определяем некоторое состояние вида $\tau^\lambda = \tau_0 - (\lambda^{ins} \cup \lambda^{del})$.

Шаг 4. Фиксируем некоторое упорядоченное множество информационных объектов $\{l_{o_1}, \dots, l_{o_m}\} \subseteq \tau^\lambda$.

Шаг 5. Определяем состояние БД τ_2 , соответствующее объединению $\tau_0 \cap \lambda^{ins}$.

Шаг 6. Последовательно, от 1 до m , формируем элементы отображения $\delta': \tau'_0 \rightarrow \tau'_1$, таких, что $(\tau'_0 \rightarrow \tau'_1)$ соответствует $\eta^{(I, \Lambda)}(\tau_0 \rightarrow \tau_2 \cup \{l_{o_i}\})$. Если все элементы проанализированы, то шаг 8.

Шаг 7. Если $\tau'_0 = \tau_0$ и существует отображение $\delta': \tau'_0 \rightarrow \tau'_1$, а также $\tau'_1 \cap \lambda^{ins} = \emptyset$, то $\tau_2 = \tau_2 \cup \{l_{o_i}\}$ и шаг 6, в противном случае состояние τ_2 остается без изменения и шаг 6.

Шаг 8. Формируем элементы отображения $\delta: \tau_0 \rightarrow \tau_1$ такие, что $(\tau_0 \rightarrow \tau_1)$ соответствует $\eta^{(I, \Lambda)}(\tau_0 \rightarrow \tau_2)$.

Шаг 9. Фиксируем сформированное состояние τ_1 и шаг 10.

Шаг 10. Алгоритм закончен.

Покажем, что данный алгоритм может выполняться за полиномиальное время.

Теорема 2. Пусть задано начальное состояние БД τ_0 , множество ограничений I и множество модифицированных информационных объектов Λ , совместных с I для τ_0 , тогда алгоритм 1 вычисляет состояние $\tau_1 \in T^{I, \Lambda}(\tau_0)$ за полиномиальное время.

Доказательство. Из того, что по определению обновление Λ совместно с I для τ_0 , то множество $\dot{\tau}(I, \Lambda)$ не пусто. Тогда по определению модификации получаем, что для каждого состояния $\tau_1 \in \dot{\tau}(I, \Lambda)$ и для множеств λ^{ins} и λ^{del} имеем $\lambda^{ins} \subseteq \tau_1$ и $\lambda^{del} \cap \tau_1 = \emptyset$. Исходя из данных выкладок, получаем, что для любого информационного объекта $l_i \in \{\tau_0 \cap \tau_1\}$ справедливо условие принадлежности: $l_i \in \lambda^{ins}$ или $l_i \in \tau^\lambda$. Далее подтвердим однозначность условий, выполняемых в теле цикла на шагах 6-7.

Рассмотрим два условия:

1) существует $\tau'_1 \in \dot{\tau}(I, \Lambda)$ такое, что $\tau_2 \subseteq \tau'_1 \cap \tau_0$;

2) пусть $l_j \notin \tau_2$ ($j \leq i$), тогда не существует такого состояния $\tau_1 \in \dot{\tau}(I, \Lambda)$, для которого $(\tau_2 \cup \{l_j\}) \subseteq \tau_1$.

Вначале (при $i = 0$) условия а) и б) выполняются согласно условию 2 леммы 1, так как $\tau_0 \cap \lambda^{ins} \subseteq \tau_1$ для любого $\tau_1 \in \dot{\tau}(I, \Lambda)$. Дальнейшее рассуждение проведем по индукции. Предположим, что на i -ой итерации

условия также выполняются и покажем, что они выполняются на $i+1$ итерации. Если $\tau_2 = \tau_2 \cup \{l_{o_1}\}$ (шаг 7), то на следующей итерации τ_2 соответствует такому значению, что и определенное на шаге 6 $\tau'_1 \in \dot{\tau}(I, \Lambda)$ (согласно условию 2 леммы 1). Если на шаге 7 состояние τ_2 остается без изменения, то, следовательно, по индукции условие а) выполняется. Условие б) также выполняется, так как если некоторое состояние $\tau_j \notin \tau_0 \cap \tau_2$, то $j \leq i-1$ при $l_i \in \tau_2$. Предположим, что при $j \leq i$ информационный объект $l_i \notin \tau_2$. Рассмотрим два случая: если $j < i$, то условие б) выполняется в силу предположения индукции; если $j = i$ и состояние $\tau_1 \in \dot{\tau}(I, \Lambda)$ содержит $\tau_2 \cup \{l_i\}$, то согласно условию 2 леммы 1 состояние $\tau'_1 \in \dot{\tau}(I, \Lambda)$ и условия шага 7 для отображения $\delta': \tau'_0 \rightarrow \tau'_1$ выполняются положительно, следовательно, условие б) также выполняется.

Отметим, что конечное состояние τ_1 формируется после m -ой итерации, при этом $(\tau_0 \rightarrow \tau_1)$ соответствует $\eta^{I, \Lambda}(\tau_0 \rightarrow \tau_2)$. Тогда из выполнения условия а) следует, что $\tau_1 \in \dot{\tau}(I, \Lambda)$, а из выполнения условия б) следует, что $\tau_0 \cap \tau_1 = \tau_2$. Используя условия 4 леммы 1, можно сделать вывод, что $\tau_1 \in T^{I, \Lambda}(\tau_0)$.

Теперь рассчитаем сложность алгоритма. Пусть n – длина входных данных алгоритма. Отметим, что количество итераций для формирования состояния τ_2 выполняется не более чем n раз, а поиск непосредственно перехода $\tau'_0 \rightarrow \tau'_1$ вычисляется за линейное от n время. Другие операции существенно не влияют на вычислительную сложность. Таким образом, общее время работы алгоритма не превышает $O(n^2)$.

Доказательство закончено.

Далее покажем, что с помощью алгоритма 1 для любого состояния τ_0 , представленного упорядоченным множеством информационных объектов $\{l_{o_1}, \dots, l_{o_m}\} \subseteq \tau^\lambda$ (множество τ^λ получено на шаге 3 алгоритма 1), можно построить состояние $\tau_1 \in T^{I, \Lambda}(\tau_0)$.

Теорема 3. Для любого состояния из $T^{I, \Lambda}(\tau_0)$ существует такой порядок информационных объектов $\{l_{o_1}, \dots, l_{o_m}\} \subseteq \tau^\lambda$, для которого алгоритм 1 вычисляет состояние τ_1 .

Доказательство. Пусть τ_0 – начальное состояние БД и пусть пересечение множеств $\tau_0 \cap \tau^\lambda$ содержит k информационных объектов ($k \leq m$). Упорядочим множество $\{l_{o_1}, \dots, l_{o_m}\}$ таким образом, чтобы элементы пересечения $\tau_0 \cap \tau^\lambda$ были вначале, то есть $\tau_0 \cap \tau^\lambda = \{l_{o_1}, \dots, l_{o_k}\}$. Очевидно, что информационные объекты $(l_{o_1}, \dots, l_{o_k})$ после k итераций цикла (шаг 6) сформируют состояние τ_2 . При этом на следующих итерациях никакой другой информационный

объект l_{o_i} ($i > k$) не будет добавлен в τ_2 , так как пересечение $\tau_0 \cap \tau_1$ будет максимальным (подпоследовательность наибольшей длины). Следовательно, в конце работы алгоритма 1 выполняется равенство $\tau_2 = \tau_0 \cap \tau_1$ и согласно условию 4 леммы 1 результирующим состоянием будет τ_1 .

Доказательство закончено.

Для решения задачи построения полного множества состояний из $T^{I,\Lambda}(\tau_0)$ преобразуем алгоритм 1. Введем дополнительную функцию Φ_m^O , которая вычисляет все перестановки информационных объектов множества $\{l_{o_1}, \dots, l_{o_m}\} \subseteq O$.

Алгоритм 2. Построение множества состояний БД.

Вход. Состояние БД τ_0 , модификации $\Lambda = \{\Lambda^{ins}, \Lambda^{del}, \Lambda^{upd}\}$, совместные с I .

Выход. Множество состояний БД T .

Метод.

Шаг 1. Фиксируем множество добавляемых информационных объектов как $\lambda^{ins} = \Lambda^{ins} \cup \{l_{i_1} \mid \exists l_{i_2} ((l_{i_1}, l_{i_2}) \in \Lambda^{upd})\}$.

Шаг 2. Фиксируем множество удаляемых информационных объектов как $\lambda^{del} = \Lambda^{del} \cup \{l_{i_2} \mid \exists l_{i_1} (l_{i_1} \in \lambda^{ins} \text{ и } (l_{i_1}, l_{i_2}) \in \Lambda^{upd})\}$.

Шаг 3. Определяем некоторое состояние вида $\tau^\lambda = \tau_0 - (\lambda^{ins} \cup \lambda^{del})$.

Шаг 4. Фиксируем некоторое упорядоченное множество информационных объектов $\{l_{o_1}, \dots, l_{o_k}\} \subseteq \tau^\lambda$.

Шаг 5. Фиксируем T как пустое множество состояний.

Шаг 6. Количество элементов информационных объектов фиксируем как $m = |\tau^\lambda|$.

Шаг 7. Последовательно, от 1 до $m!$, выполняем перебор элементов множества $\{l_{o_1}, \dots, l_{o_k}\} = \Phi_m^{\tau^\lambda}$.

Шаг 8. Определяем состояние БД τ_2 , соответствующее объединению $\tau_0 \cap \lambda^{ins}$.

Шаг 9. Последовательно, от 1 до m , выполняем формирование элементов отображения $\delta': \tau'_0 \rightarrow \tau'_1$, таких, что $(\tau'_0 \rightarrow \tau'_1)$ соответствует $\eta^{I,\Lambda}(\tau_0 \rightarrow \tau_2 \cup \{l_{o_i}\})$. Если все элементы проанализированы, то шаг 11.

Шаг 10. Если $\tau'_0 = \tau_0$ и существует отображение $\delta': \tau'_0 \rightarrow \tau'_1$, а также $\tau'_1 \cap \lambda^{ins} = \emptyset$, то $\tau_2 = \tau_2 \cup \{l_{o_i}\}$ (то есть в τ_2 заносим информационный объект l_{o_i}) и шаг 6, в противном случае состояние τ_2 остается без изменения и шаг 9.

Шаг 11. Формируем элементы отображения $\delta: \tau_0 \rightarrow \tau_1$ такие, что $(\tau_0 \rightarrow \tau_1)$ соответствует $\eta^{I,\Lambda}(\tau_0 \rightarrow \tau_2)$.

Шаг 12. Фиксируем сформированного состояния τ_1 .

Шаг 13. Если $\tau_1 \notin T$, то $T = T \cup \{\tau_1\}$ (то есть в T заносим состояние τ_1).

Шаг 14. Если число итераций меньше $m!$, то шаг 7, в противном случае шаг 15.

Шаг 15. Фиксируем сформированное множество состояний T и шаг 16.

Шаг 16. Алгоритм закончен.

Содержательно алгоритм 2 можно разделить на следующие этапы. На шаге 5 инициализируется список состояний T . Шаг 7 определяет внешний цикл, в котором перебираются все перестановки информационных объектов множества τ^λ и для каждой из них во внутреннем цикле (шаг 9) ищется состояние τ'_1 , после чего на шаге 13 инициализируется результирующий список T . Согласно теореме 3 получаем, что в множество T попадут все состояния из $T^{I,\Lambda}(\tau_0)$.

Исходя из оценки времени алгоритма 1, можно оценить время работы алгоритма 2. Цикл на шаге 7 выполняется $m!$ раз, а его тело, согласно теореме 2, требует $O(n^2)$ шагов. Таким образом, общее время работы алгоритма 2 соответствует $O(m!n^2)$.

Обобщая полученные результаты, отметим, что для некоторого начального состояния БД τ_0 , множества ограничений целостности I и множества обновляемых информационных объектов Λ , совместного с I для τ_0 , алгоритм 2 корректно вычисляет все состояния БД из $T^{I,\Lambda}(\tau_0)$ за время $O(m!n^2)$, где $n = |I| + |\Lambda| + |\tau_0|$, а $m = |\tau_0 - (\lambda^{ins} \cup \lambda^{del})|$.

Выводы. Анализ отечественных и зарубежных источников показал, что исследования не рассматривают формальную сторону проблемы, позволяющую выстроить единую концепцию совместного использования БД при локальной независимости информационных систем. При этом существующие решения не позволяют осуществить эффективную интеграцию гетерогенных ИС.

Средства, определяющие корректность модификации данных при интеграции БД выражаются в виде ограничений, накладываемых на значения, допустимые в некотором состоянии БД. При модификации структуры данных предлагается сохранить начальные ограничения или так их изменить, чтобы не противоречить семантике данных.

Таким образом, в статье исследованы и решены следующие задачи. Введено понятие близости состояний БД. Определен оператор, который по исходному состоянию БД и запросу на обновление вычисляет новое состояние, удовлетворяющее ограничениям целостности и являющееся ближайшим к исходному состоянию. Сформулирована и доказана теорема, подтверждающая совместность выполняемых модификаций с заданными ограничениями целостности. То есть, определено, что для заданного множества ограничений в некотором состоянии БД область значений оператора перехода совпадает с множеством информационных объектов нового состояния.

Для построения алгоритма поиска множества допустимых состояний сформулированы условия существования неподвижной точки для модифицированных информационных объектов, совместных с ограничениями целостности. Доказана корректность построения перехода состояний и показано, что алгоритм выполняется за полиномиальное время. Построен алгоритм нахождения полного множества возможных состояний БД при выполнении операций модификации данных.

Полученные результаты подтверждают адекватность предлагаемых методов и алгоритмов и могут быть использованы при усовершенствовании технологий дедуктивных БД для организации доступа к реляционным БД, а также реализации методов представления интенционала БД для процедур вывода результатов запроса средствами реляционных системы управления данными.

Дальнейшие исследования в рассматриваемой области необходимо сконцентрировать на вопросах организации запросов в системах БД с неоднородной структурой, выбора формализованных средств описания и манипулирования данными, а также разработке методов построения и поддержки эквивалентных запросов для гетерогенных информационных систем и распределенных неоднородных баз данных.

Список литературы: 1. *Таненбаум Э. С.* Распределенные системы. Принципы и парадигмы / Э. С. Таненбаум, М. Ван Стеен – СПб. : Питер, 2003. – 877 с. 2. *Фридл Дж.* Регулярные выражения / Дж. Фридл – СПб. : Питер, 2003. – 464 с. 3. *Мейер Д.* Теория реляционных баз данных / Д. Мейер. – М. : Мир, 1987. – 608 с. 4. *Горчинская О. Ю.* Разложение отношений и логическое проектирование реляционных баз данных / О. Ю. Горчинская, С. В. Петров // Автоматика и телемеханика. – 1983. – № 2. – С. 159–166. 5. *Калиниченко Л. А.* Методы и средства интеграции неоднородных баз данных / Л. А. Калиниченко. – М. : Наука, 1983. – 424 с. 6. *Кузнецов С. Д.* Методы оптимизации выполнения запросов в реляционных СУБД / С. Д. Кузнецов // Итоги науки и техники. Вычислительные науки. – 1989. – Т. 1. – С. 76–153. 7. *Танянский С. С.* Модель поддержки целостности в интегрированных базах данных неоднородной структуры / С. С. Танянский // Вестник Херсонского государственного технического университета. – 2004. – № 1(19). – С. 230–235. 8. *Танянский С. С.* Модификация ограничений на ведение данных для обеспечения целостности крупномасштабных информационных систем / С. С. Танянский, В. В. Тулунов, Д. А. Руденко // Вестник Национального технического университета «ХПИ». – 2006. – № 23. – С. 137–144. 9. *Танянский С. С.* Поддержка логической целостности в задачах преобразовании структуры баз данных / С. С. Танянский // Вестник Херсонского Национального технического университета. – 2010. – № 2(38). – С. 227–231. 10. *Танянский С. С.* Характеристические свойства объектов информационных систем / С. С. Танянский // Искусственный интеллект. – Научный журнал. – Институт проблем искусственного интеллекта. – Донецк – 2007. – № 1. – С. 78–89.

Надійшла до редколегії 05.12.2012