

**ЧИСЛЕННЫЙ АНАЛИЗ
КАРКАСНОЙ СХЕМЫ
РЕЛЯЦИОННОЙ БАЗЫ
ДАнных**

Введение. В работе [1] предложен алгоритм синтеза новой модели данных – реляционного каркаса – на основе многозначных зависимостей ключевых атрибутов. Предлагается использовать N -арные таблицы для моделирования связей степенью $H:G$. Решается проблема модифицируемости структуры реляционного хранилища данных.

Показано, что реляционный каркас позволяет разработать обобщенный универсально-гибкий алгоритм синтеза схем БД, моделирующий предметные области (ПрО), используя единую процедуру автоматизации синтеза. Такая процедура предоставляет схеме БД полную модифицируемость [2], т. е. минимизирует число операций по модификации. И позволяет вносить изменения в динамическом режиме – непосредственно в процессе эксплуатации приложения. А также оптимизировать объединение различных приложений, построенных в соответствии с данным алгоритмом, в единую информационную систему.

Как показано в [3], описываемый алгоритм может быть использован, прежде всего, для генерации схем БД, удовлетворяющих условиям доменно-ключевой нормальной формы (ДКНФ). Однако его можно также использовать для проектирования устройств распознавания речи, устройств-переводчиков, экспертных систем, систем автоматизированного аудита корректности работы введенных в эксплуатацию комплексов приложений. А также систем автоматизированного проектирования дейтацентров. При этом понимается,

Неформально описан алгоритм каркасной сепарации сущностей-объектов для проектирования схем базы данных (БД) в соответствии с новой моделью данных – реляционного каркаса. Предлагается новая классификация сущностей-объектов, а также новый взгляд на моделирование семантики данных. Приводятся результаты численного эксперимента доступа к данным.

© Панченко Б.Е., 2012

что ПрО имеет произвольные
структуру и объем.

Близким к предложенному является алгоритм предварительного формального описания ПрО, применяющийся в некоторых онтологиях [4]. Тем не менее, этот подход также имеет существенный недостаток – в нем отсутствует единый фактор, позволяющий систематизировать семантику сущностей-объектов, т. е. существительных из описания ПрО. А также отсутствует подход, который доказательно минимизирует количество базовых категорий, позволяющих вести автоматизированное отнесение сущностей-объектов к той или иной категории – *separации* сущностей-объектов – из значительного числа синонимов и терминов из начального потока описания ПрО.

Классификация и регрессия. Принципиально важным для описываемого алгоритма является совпадение схем всех каркасных отношений со схемой известной задачи классификаций и регрессии в Data Mining [5].

Требуется определить значение зависимой переменной сущности-объекта на основании значений других переменных, характеризующих ее атрибуты. Формально задачу классификации и регрессии можно описать следующим образом. Имеется множество объектов:

$$I = \{i_1, i_2, i_3, \dots, i_l, \dots, i_n\},$$

где i_l – исследуемый объект. Примером таких сущностей-объектов может быть информация о проведении вступительных экзаменов в разных ВУЗах Киева (табл. 1).

ТАБЛИЦА 1

ВУЗ	Предмет	Преподаватель	Абитуриент	Оценка	Длительность
КГУ	Математика	Иванов	Сергеев	Удовл	30 мин
КПИ	Математика	Петров	Сергеев	Хор	20 мин
НПУ	Иностранный	Александров	Разин	Отл	60 мин
НАУ	Иностранный	Александров	Разин	Хор	40 мин

Каждая сущность-объект характеризуется набором переменных:

$$I_i = \{x_1, x_2, x_3, \dots, x_h, \dots, x_m\},$$

где x_h – независимые переменные, с известными значениями. На их основании определяется значение зависимой переменной y . В данном примере независимыми переменными являются: *ВУЗ, предмет, преподаватель и абитуриент*. Зависимой переменной является *оценка и длительность*.

В Data Mining часто набор независимых переменных обозначают в виде вектора: $X = \{x_1, \dots, x_m\}$. Каждая переменная x_1 может принимать значения из некоторого множества (домена): $C_h = \{c_{h1}, c_{h2}, \dots\}$.

Если значениями переменной являются элементы конечного множества, то говорят, что она имеет категориальный тип. Например, переменная *предметы* принимает значения на множестве значений {*математика, иностранный язык, история*}.

Если множество значений $C = \{c_1, c_2, \dots, c_r, \dots, c_k\}$ переменной y – конечное, то задача называется задачей классификации. Если переменная y принимает значение на множестве действительных чисел R , то задача называется задачей регрессии.

Несмотря на то, что способ определения значения зависимой переменной назван функцией классификации или регрессии, он необязательно может быть выражен математической функцией. Существуют основные виды представления таких способов: классификационные правила, деревья решений и математические функции. Для алгоритма синтеза схем БД первые два представления особенно важны.

Классификационные правила состоят из двух частей: условия и заключения: *если* (условие) *то* (заключение).

Условие – проверка одной или нескольких независимых переменных. Проверки нескольких переменных могут быть объединены с помощью операций "*и*", "*или*" и "*не*". Заключением является значение зависимой переменной или распределение ее вероятности по классам.

Основное достоинство правил – легкость их восприятия и запись на естественном языке. Другое преимущество – их взаимная независимость. Данные свойства полностью совпадают со свойством каркасных отношений, большинство которых моделируют высказывания [3, 6]. Тем самым реляционный каркас естественным способом готовит БД к завершающим этапам эксплуатации хранилищ данных – OLAP и Data Mining [5].

Однако самое важное сходство – принцип модифицируемости. Поскольку совокупность правил строится также на фрагменте булеана, добавление нового правила осуществляется без изменения уже существующих. Важным свойством совокупности правил при этом является их взаимная независимость. Аналогичный принцип заложен и в каркасную совокупность отношений, и в совокупность кортежей каждого отношения, построенную на шунтированной многозначной [7] и декартовой зависимости [3].

Каркасные сущности-объекты ПрО. В соответствии с [8] в описываемом алгоритме все сущности-объекты распределяются на пять категорий.

Атомарные сущности-объекты – сущности-объекты, значения атрибутов которых в моделируемой ПрО практически не изменяются во времени. И потому, так или иначе, определяют структуру ПрО. Они также не имеют зависимости существования [9]. В некоторых моделях данных их называют стержневыми, базовыми и т. п. [10].

Примером атомарных сущностей-объектов может быть сущность-объект *ЧЕЛОВЕК*, *ВСЕЛЕННАЯ*, *СОБАКА*, *КОШКА* и т. п. Причем принадлежность данных сущностей-объектов к определенным дальнейшим категориям – так называемая классификация атомарных сущностей-объектов – искусственная семантическая надстройка пользователя, маскирующая содержание сущности-объекта.

Слабые сущности-объекты – сущности-объекты, которые функционально зависят от атомарных [9]. В иных моделях имеют аналогичное название. Причем такая зависимость может быть как лишь на уровне идентификации слабых атрибутов, так и на уровне всего существования зависимых слабых сущностей-объектов. В начальном приближении модели ПрО слабые сущности-объекты условно отнесены к атомарным, так как их атрибуты также не зависят от времени.

Примером слабых, но, тем не менее, условно относимых к категории атомарных, могут быть сущности-объекты *ПОДРАЗДЕЛЕНИЕ, ОТДЕЛ, ЛАБОРАТОРИЯ, КВАРТИРА* – каждая из этих сущностей-объектов не является самодостаточной. И в ПрО функционально зависит от «предковых» сущностей-объектов. Однако зависимостью ее атрибутов от времени в ПрО можно пренебречь.

Составные («постсвязные») сущности-объекты – в моделях данных имеют еще и название многосторонних связей [9, 11] – связь атомарных сущностей-объектов.

Пример составных сущностей-объектов – событийные сущности-объекты – *ЭКЗАМЕН, КОНЦЕРТ, ВЫСТАВКА, СОГЛАШЕНИЕ, МИТИНГ* и т. п. Их содержание представляет собой «продукт» равноправного взаимодействия нескольких атомарных сущностей-объектов.

В работе [12] также описаны сущности-объекты, полученные агрегацией данных. Там *ЭКЗАМЕН* также отнесен к категории агрегированных сущностей-объектов, т. е., составных. «Например, связь между сущностями *СТУДЕНТ, ДИСЦИПЛИНА, ПРЕПОДАВАТЕЛЬ, ОЦЕНКА* имеет смысловое описание: «студент по фамилии __ получил на экзамене по дисциплине __ у преподавателя по фамилии __ оценку __» и может быть представлена агрегированным элементом: сущностью *ЭКЗАМЕН* с атрибутами *ФИО_СТУДЕНТА, НАЗВАНИЕ_ДИСЦИПЛИНЫ, ФИО_ПРЕПОДАВАТЕЛЯ, КОД_ОЦЕНКИ*» [12].

Каркасная модель ПрО предоставляет проектировщику совокупность отношений, каждое из которых моделирует либо статическое состояние сущности-объекта (тривиальную связь), либо одну из связей совокупности сущностей-объектов.

Описанные типы сущностей-объектов, так или иначе, встречались в других концептуальных моделях. Однако, как показывает практика проектирования приложений, их не достаточно. Для более точного моделирования ПрО необходимо использование еще двух типов – артефакты и неопределенные сущности-объекты.

Артефакты, т. е. сущности-копии, данные о которых будут условно размещаться в БД по решению пользователя.

К артефактам, например, можно отнести любой документ, который пользователи ПрО создают именно ради того, чтобы скопировать те или иные атрибуты тех или иных сущностей-объектов. Причем не просто скопировать атрибуты одной конкретной сущности-объекта, а еще и объединить в этой новой искусственно созданной сущности-объекте несколько атрибутов от разных сущностей-объектов.

Артефакты – это, как правило «пост-следственные» сущности-объекты. Поэтому регистрируя их в приложении, пользователь сталкивается со значительным дублированием данных. А это, в свою очередь, приводит к потребности дополнительного отслеживания целостности еще и избыточных данных. Исключение составляют совокупность искусственных сущностей-объектов, каждая из которых объединяет лишь определенную *часть* атрибутов другой, более общей не искусственной сущности-объекта.

Объединение совокупностей атрибутов каждой искусственной сущности-объекта строго тождественно совокупности всех атрибутов общей, не искусственной сущности-объекта. То есть, ни одна из искусственных сущностей-объектов не имеет ни одного атрибута, который является общим хотя бы для двух искусственных сущностей-объектов. А также не существует ни одного атрибута у общей не искусственной сущности-объекта, для которого не существовало бы копии среди совокупности искусственных сущностей-объектов.

Таким образом, эту совокупность искусственных сущностей-объектов каркасный алгоритм также отнесет к категории «артефакты». Тем не менее, отслеживание целостности таких дублированных данных упрощается. Заранее отметим, что именно такие артефакты на втором этапе каркасного алгоритма используются как *маски* сущностей-объектов.

Примером артефактов могут быть *НАКЛАДНАЯ, СЧЕТ* (к уплате в ресторане, за другие услуги и т. п.), *АКТ* и т. д.

Неопределенные сущности-объекты, семантика которых подлежит дальнейшему уточнению. Возникают в описании ПрО на этапе незавершенного исследования функционирования ПрО.

Этимология смысла сущности-объекта. Для реализации алгоритма автоматизированной сепарации сущностей-объектов, для отношений, моделирующих каждую из них, формируются *структурированные идентификаторы*, структура каждого из которых не произвольная, не задана пользователем и не получена каким-то иным отличным способом, а строго исчисляется.

Для автоматизированного исчисления структуры идентификатора используются математические критерии, построенные в соответствии с закономерностями, выявленными в ПрО. В основе этих критериев – единый обобщенный фактор – *происхождение* смысла сущности-объекта, т. е. *этимология* ее смысла (в дальнейшем – просто «этимология»).

В описываемом алгоритме используется **гипотеза**: все факторы, характеризующие семантику любой сущности-объекта в ПрО, являются функционально зависимыми от этимологии смысла сущности-объекта.

Под *этимологией смысла* сущности-объекта понимается строковый структурированный идентификатор со следующей общей схемой:

$$X_1^{m_1} + X_2^{m_2} + X_3^{m_3} + \dots + X_{k_i}^{m_k},$$

где каждое звено $X_{k_i}^{m_k}$ – отделенный идентификатор факта происхождения i -й сущности-объекта, k_i – номер звена идентификатора i -й сущности-объекта (подстрочный индекс), m_k – номер соответствующей порождающей сущности-объекта из базовой совокупности сущностей-объектов – объединенной группы атомарных и слабых сущностей-объектов (надстрочный индекс), причем каждое m_k может получить какое либо значение только из множества $\{1, 2, \dots, N_0, \dots, N\}$, где N_0 – общее количество атомарных сущностей-объектов, N – суммарное количество атомарных и слабых сущностей, i – номер произвольной сущности-объекта в ПрО.

Причем в случае полной совокупности связей $i = \{1, 2, \dots, N_0, \dots, \dots, N, (N+1), \dots, (2^N - 1)\}$. Знак «плюс» в общем виде схемы этимологии означает строковое объединение (конкатенацию). Для атомарных сущностей этимологией есть лишь одно звено X^i , в котором $m = i$. То есть атомарная сущность-объект порождает сама себя.

В каркасном алгоритме атомарные сущности-объекты получают в общей совокупности первые номера, т. е. для них $i = 1, N_0$. Для слабых сущностей-объектов этимологией является вышеупомянутая строковая сумма звеньев, где каждому номеру k_i звено $X_{k_i}^{m_k}$ соответствует строко. То есть последовательность звеньев строго соответствует последовательности зависимостей каждого следующего звена от предыдущего, что в свою очередь соответствует последовательности синтеза каждой предыдущей слабой сущностью-объектом, вплоть до старшей атомарной, следующей слабой сущности-объекта.

Для составных сущностей-объектов этимология – вышеупомянутая строковая сумма звеньев, где место каждого звена $X_{k_i}^{m_k}$ не строгое, т. е. последовательность звеньев не имеет значения. Тем не менее, суммарная совокупность звеньев строго отвечает совокупности формирующих сущностей-объектов. Таким образом, в общем случае для какой либо сущности-объекта весь структурированный идентификатор ячейки представляет собой суммарную строку букв или цифр, каждое звено которой имеет минимально достаточный строковый размер. Такой идентификатор, например, в реляционной модели данных может использоваться в качестве минимально достаточного суррогатного ключа отношения, объединяющего все свойства конкретной сущности-объекта.

Каждое звено этимологии сущности-объекта означает связь с иными атомарными сущностями-объектами, которые принимали участие в происхождении данной сущности-объекта, если последняя представляет собой или слабую, или составную, т. е. постсвязную сущность-объект.

Таким образом, каждое звено $X_{k_i}^{m_k}$ идентификатора ячейки строится в строгом соответствии с происхождением сущностей-объектов из описания ПрО. Каждая сущность-объект в ПрО может быть атомарной, а значит иметь *унарный* X^i , или *составной* идентификатор $\sum X_{k_i}^{m_k}$, где суммирование ведется по k_i , $k_i = 1, K_i$. Причем общее число звеньев K_i представляет собой арность отношения, моделирующего такую сущность-объект в реляционной модели данных (РМД). В общем случае может равняться 2, 3, ..., 10 и т. д. А в случае атомарной сущности-объекта равняется исключительно единице.

Результаты экспериментальных исследований. В работе проведен специализированный эксперимент, основной характеристикой которого является время доступа к любому атрибуту любого отношения исследуемой БД. Как известно, самой ресурсоемкой операцией реляционной алгебры являются соединения отношений [9]. Данная операция применяется при выполнении запросов к БД более низкой «нормальности» – не выше 3НФ. Отличительной же особенностью именно каркасной схемы БД является то, что все отношения уже соединены. И целостность БД гарантирует наличие любых ключевых данных для любых

соединений. Поэтому отсутствие операции соединения приводит к значительной экономии времени доступа к БД. Если, конечно же, иное не навязано пользователю реализацией конкретной СУБД.

Для экспериментального исследования каркасного метода проектирования схемы БД и проведения указанного численного эксперимента выбрана ПрО под условным названием ОАО «Горгаз» – предприятие, продающее потребителю в одном из областных центров Украины энергоресурсы (рис. 1). Несмотря на запрет открыто публиковать наименование базового предприятия, указанный проект является реальным. А результаты – достоверными.

На основании алгоритмов нормализации [10] и каркасного синтеза ДКНФ [3, 13] исследованы.

1. Схема существующей ЗНФ БД и схема уточненной каркасной ДКНФ БД.
2. Семантика ПрО.
3. Характеристики доступа к большим объемам данных.

В качестве примера рассмотрим фрагмент вышеуказанной ПрО – несколько составных в смысле [8] сущностей-объектов.

Для формирования унифицированного запроса к ЗНФ БД, возвращающего группу данных для документа *НАРЯД НА ОБХОД СТРОЕНИЙ НА УЧАСТКЕ*, применяется комплексная операция *выборка из соединения*. Но для каркасной схемы БД никаких дополнительных операций, кроме индексного поиска по сумме ключевых идентификаторов каждого из указанных признаков в многоарном отношении, не требуется. Результирующее отношение будет октарным (8-арным).

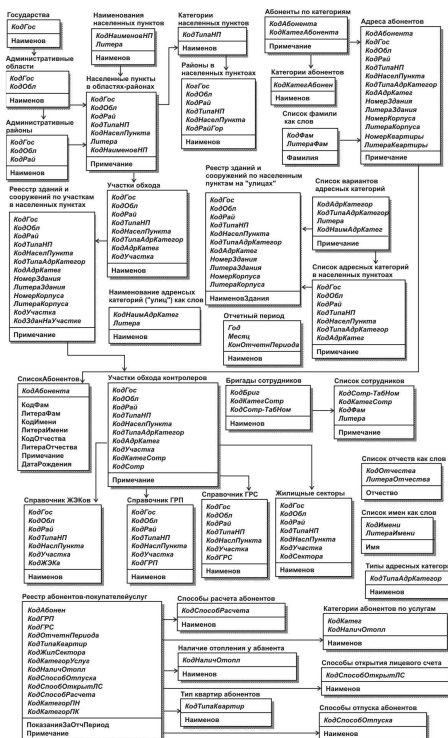


РИС. 1

В эксплуатируемой БД исследуемой ПрО были заменены 29 ЗНФ-отношений на каркасные. При этом общее число отношений в БД возросло лишь до 37. Но скорость доступа к данным по типовым запросам повысилась на несколько порядков.

На рис. 2, а показаны графики, иллюстрирующие рост времени доступа к данным при получении одной записи из увеличивающихся пачек записей при запросе на соединение 8-ми ЗНФ-отношений (кривая 2). И слабый прирост времени на индексную выборку этой же записи из одного октарного отношения в ДКНФ (кривая 1) при таком же увеличении числа записей. На рис. 2, б показано еще меньший прирост времени доступа к 14-ти арному отношению в ДКНФ (кривая 1) с увеличением числа обрабатываемых записей. И значительный рост времени выполнения этого же запроса при соединении 14 отношений в ЗНФ (кривая 2). При этом результирующее отношение имело в среднем от 100 до 200 кортежей.

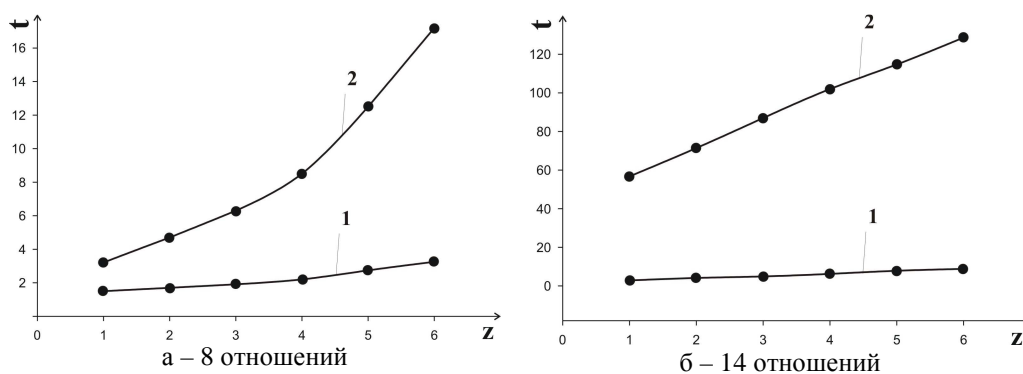


РИС. 2

В табл. 2 и 3 приведены результаты замеров времени формирования результирующего отношения на типовой запрос пользователя в зависимости от числа отношений и операций соединений, а также числа записей исходных отношений (в миллионах записей). Оказалось, что для обращения к ДКНФ-схеме число соединений примерно в 2 раза меньше. И соответствует числу обращений к справочникам, что дает дополнительное преимущество.

ТАБЛИЦА 2. Время выполнения запросов к классической схеме БД в ЗНФ

Отношений	Соединений	1 млн.	2 млн.	3 млн.	4 млн.	5 млн.	6 млн.
4	9	1.25	1.79	2.04	2.35	2.94	3.43
8	17	3.21	5.13	7.11	8.50	12.80	17.19
12	25	16.91	19.42	25.84	31.96	38.48	44.14
14	29	56.74	71.43	86.98	94.22	114.74	128.65

ТАБЛИЦА 3. Время выполнения запросов к каркасной схеме БД

Отно- шений	Соеди- нений	1 млн.	2 млн.	3 млн.	4 млн.	5 млн.	6 млн.
4	4	0.23	0.29	0.30	0.35	0.42	0.58
8	8	1.11	1.70	1.91	2.20	2.74	3.25
12	12	2.04	2.44	2.94	3.41	4.27	5.93
14	14	2.84	4.12	4.91	6.24	7.85	8.84

Из показанной на рис. 1 схемы БД очевидно, что чем сложнее ПрО, тем большее число многоарных связей присутствует в схеме. И тем более интенсифицированы отношения напоминают высказывания – обычные словесные предложения. То есть, каркасные отношения обладают одновременно и реляционной безаномальностью, и семантической. Фрагмент схемы БД показывает, что подавляющее большинство каркасных отношений моделирует связи, что полностью соответствует концепции OLAP и Data Mining [5].

Отметим также, что на рис. 1 показана не традиционная ER-диаграмма Чена [11]. Здесь дуги играют роль не связей между сущностями-объектами, а указателей на основные взаимообусловленности доменов и ключей. А значит и маршруты отслеживания целостности для СУБД. Очевидно, что соответствие имен ключевых атрибутов и означает их взаимность. Впервые аналогичные диаграммы предложены в [14, 15].

Закключение. Все многообразие ПрО моделируется совокупностями N -арных сочетаний ключевых атрибутов, каждый из которых отвечает за уникальную сущность. Очевидно, что особенность приведенных отношений заключена именно в том, что они образуют *полный* [3] каркас всех схем реляционных отношений произвольной ПрО. Как следует из теоремы о ДКНФ [3], все отношения данной совокупности при определенных условиях, налагаемых проектировщиком на ограничения ПрО, имеют безаномальную схему. [3, 13].

В работе исследован алгоритм, позволяющий проектировать многократные связи каждой сущности-объекта с произвольной группой сущностей-объектов. Отсутствие ограничений на количество *видов связей* на любом уровне арности в каждой конкретной группе сущностей, свойственное такой сущности, как, например, *ЛЮДИ* и приводящее к *рекурсивным* связям, в изложенном алгоритме учитывается с помощью *копий* сущностей, дополнительных таблиц, смысл которых – *маски для ролей* в связях сущностей.

Предлагается универсальная технология размещения данных в хранилище, которая не зависит от особенностей определенной ПрО и позволяет в динамике и без переработки эксплуатирующей системы минимально достаточными операциями выполнять любые семантически целесообразные модификации схемы БД. Сформировать набор единых процедур обработки данных – групповых функций. И таким образом *унифицировать* технологию генерации и эксплуатации хранилищ данных.

Б.С. Панченко

ЧИСЕЛЬНИЙ АНАЛІЗ КАРКАСНОЇ СХЕМИ РЕЛЯЦІЙНОЇ БАЗИ ДАНИХ

Дано неформальний опис алгоритму каркасної сепарації сутностей-об'єктів для проектування схем БД у відповідності до нової моделі даних – реляційного каркасу. Пропонується нова класифікація сутностей-об'єктів, а також новий підхід до моделювання семантики даних. Наводяться результати чисельного експерименту доступу до даних.

V.E. Panchenko

NUMERIC ANALYS FRAMEWORK SCHEME RELATIONAL DATABASE

The algorithm of framework separation of entity-objects for the Database scheme design in accordance with the new model of data - relational framework - is designed. A new classification of entity-objects and a new approach to modelling the data semantics is offered. The results of the numerical experiment of data access are given.

1. *Панченко Б.Е.* Об алгоритме синтеза реляционного каркаса. Постановка задачи и формализация // Компьютерная математика. – 2012. – № 1. – С. 84 – 93.
2. *Карпуша В.Д., Панченко Б.Е.* Моделирование и проектирование реляционных баз данных. Учебное пособие. – Суми: Изд. СумДУ, 2010. – 385 с.
3. *Панченко Б.Е.* Каркасное проектирование доменно-ключевой схемы реляционной базы данных // Кибернетика и системный анализ. – 2012. – № 3. – С. 123 – 128.
4. *Соловьев В.Д., Добров Б.В., Иванов В.В., Лукашевич Н.В.* Онтологии и тезаурусы; учебное пособие. – М.: МГУ, 2006. – 157 с.
5. *Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И.* Методы и модели анализа данных: OLAP и Data Mining, БХВ-Петербург, 2004. – 336 с.
6. *Яловец А.Л.* Представление и обработка знаний с точки зрения математического моделирования. Проблемы и решения. – Киев: Наук. думка, 2011. – 360 с.
7. *Fagin R.A.* Normal Form for Relational Databases That Is Based on Domains and Keys // ACM Transactions on Database Systems. – 1981. – Vol. 6, N 3. – P. 387 – 415.
8. *Пат. 99921.* Способ предварительной каркасной сепарации данных перед их модифицируемым размещением в хранилище или процессом дальнейшей обработки / Панченко Б.Е. / Промислова власність. – 2012. – № 20. – С. 3 – 30.
9. *Ульман Д.Д., Уидом Д.* Основы реляционных баз данных. – М.: 2006. – 374 с.

10. *Codd E.F.* Extending the Database Relational Model to Capture More Meaning. // ACM Trans. on Database Systems – 1979. – Vol. 4. – P. 397 – 434.
11. *Chen P.P.* The Entity-Relationship Model: toward a unified view of data // ACM Trans. on Data base systems. – 1976. – Vol. 1. – P. 9 – 36.
12. *Григорьев Ю.А., Ревунков Г.И.* Базы данных. Учебное пособие. – М.: МГТУ, 2002. – 319 с.
13. *Панченко Б.Е., Писанко И.Н.* Свойства реляционного каркаса на множестве семантически атомарных предикатов // Кибернетика и системный анализ. – 2009. – № 6. – С. 177 –182.
14. *Панченко Б.Е., Гайдабрус В.Н., Церковицкий С.Л.* Создание сетевых информационных комплексов // Компьютеры плюс программы. – Киев. – 1993. – № 5 (6). – С. 46 – 50.
15. *Панченко Б.Е., Гайдабрус В.Н., Церковицкий С.Л.* Сетевые вычислительные комплексы // Компьютеры плюс программы. – Киев. – 1994. – Спец. вып. – С. 30 – 37.

Получено 19.12.2011

Об авторе:

Панченко Борис Евгеньевич,
кандидат физико-математических наук, старший научный сотрудник
Института кибернетики имени В.М. Глушкова НАН Украины.