

УДК 004.82

## Подход к преобразованию модели базы знаний предметной области на основе мультиграфа в реляционную модель базы данных

Зайцев А. Ф.

**Постановка задачи:** проблема построения современных интеллектуальных и экспертных программных систем требует решения множества разнообразных задач. В настоящее время сложность построения подобных систем заключается в том, что известные «классические» модели и подходы к построению экспертных систем плохо согласуются с реляционной моделью представления данных, что делает невозможным эффективное применение современных промышленных систем управления базами данных для организации и представления знаний, а также их использования в различных информационных системах. **Целью работы** является преобразование предложенной модели базы знаний предметной области на основе мультиграфа в реляционную модель базы данных для повышения эффективности обработки информации с использованием современных автоматизированных систем управления. **Используемые методы:** Предложен подход, позволяющий преобразовывать и хранить знания предметных областей в табличном представлении и реляционных базах данных. В подходе отражены принципы и правила по выполнению декомпозиции модели базы знаний с мультиграфом, представлению мультиграфа знаний с использованием алгоритмических структур данных (хеши-таблиц), хранению мультиграфа знаний в табличном виде базы данных с использованием выбранного символьного формата (JSON), а также выполнению обратного преобразования и агрегации списков вершин мультиграфа из реляционного вида в нереляционный. При этом были использованы следующие общенаучные методы: анализ, декомпозиция, синтез, моделирование, формализация, структуризация, алгоритмизация, обобщение, сравнение, описание, вычислительный эксперимент, а также теория множеств и теория графов на базе которых была построена используемая модель базы знаний. В качестве материалов исследования использовались компьютеры и инструментальные средства программного обеспечения – системы управления базами данных, языки программирования и их интерпретаторы. **Новизна:** элементами новизны являются новая математическая модель базы знаний предметной области на основе мультиграфа, а также предложенный подход к преобразованию модели базы знаний на основе мультиграфа в реляционную модель базы данных. **Результат:** проведен сравнительный анализ, приведены основные определения, особенности и отличия между данными и знаниями. Описан процесс построения модели базы знаний на основе мультиграфа, предназначенной для решения задач из выбранной предметной области. Выявлены и описаны проблемы взаимодействия с мультиграфовой базой знаний. Разработан и применен подход к решению выявленных проблем, заключающийся в преобразовании модели базы знаний предметной области на основе мультиграфа в реляционную модель базы данных. Рассмотрены типы, формат и структура представления знаний с особенностями их применения в выбранной системе управления. **Практическая значимость:** Показана программная реализация базы данных и знаний на выбранном языке программирования. Приведены листинги с фрагментами кода на языках программирования для взаимодействия с базой, а также примеры представления и обработки знаний. Реализация предложенного подхода дает возможность организации и хранения знаний в виде баз данных, что позволяет более эффективно выполнять обработку и управление знаниями с применением реляционной модели. Кроме того, управление знаниями осуществляется с использованием существующих, проверенных и производительных систем управления базами данных промышленного уровня. Использо-

### Библиографическая ссылка на статью:

Зайцев А. Ф. Подход к преобразованию модели базы знаний предметной области на основе мультиграфа в реляционную модель базы данных // Системы управления, связи и безопасности. 2024. № 4. С. 158-178. DOI: 10.24412/2410-9916-2024-4-158-178

### Reference for citation:

Zaytsev A. F. An approach to transforming a model of multigraph knowledge base of the subject area into a relational database model. *Systems of Control, Communication and Security*, 2024, no. 4, pp. 158-178 (in Russian). DOI: 10.24412/2410-9916-2024-4-158-178

ние предложенного подхода открывает новые возможности для проектирования и реализации современных экспертных и интеллектуальных программных систем.

**Ключевые слова:** проблема представления знаний, базы знаний, базы данных, моделирование динамических систем, экспертные системы, мультиграф знаний, экспертная информация, интеллектуальные системы, программирование и автоматизация, разработка программных систем.

## Актуальность

Проблема построения современных интеллектуальных и экспертных программных систем требует решения множества разнообразных задач. В настоящее время проблема построения подобных систем заключается в том, что существующая «классическая» концепция экспертных и интеллектуальных систем, сложившаяся в 1970–1980-х годах, переживает некоторый кризис, связанный с её глубокой ориентацией на общепринятый в те годы текстовый человеко-машинный интерфейс, который сегодня во всех пользовательских приложениях почти полностью вытеснен графическим. Кроме того, известные «классические» модели и подходы к построению экспертных систем плохо согласуются с реляционной моделью представления данных, что делает невозможным эффективное применение современных промышленных систем управления базами данных (СУБД) для организации и управления знаниями, а также их использования в различных информационных системах. Разнообразные попытки объединить «классические» модели представления знаний [1] для построения экспертных и интеллектуальных систем с современными подходами, использующими реляционные базы данных и графический пользовательский интерфейс, пока не находят поддержки среди крупных компаний-производителей программного обеспечения и по этой причине остаются в экспериментальной стадии.

В связи с необходимостью решения существующей проблемы целью исследования является решение поставленной задачи по преобразованию предложенной модели базы знаний предметной области на основе мультиграфа в реляционную модель базы данных для повышения эффективности обработки и управления знаниями с использованием современных СУБД. Для достижения цели поставлены следующие подзадачи:

- 1) провести сравнительный анализ, выявить и описать особенности представления данных и знаний, а также их основные отличия;
- 2) описать формальную модель базы знаний предметной области на основе мультиграфа и её реализацию на выбранном языке программирования;
- 3) разработать, описать и реализовать подход по преобразованию модели базы знаний предметной области на основе мультиграфа в реляционную модель базы данных.

Решение поставленной задачи даст возможность организации и хранения знаний в базах данных, что позволит более эффективно выполнять обработку и управление знаниями с применением реляционной модели. Кроме того, управление знаниями будет осуществляться с использованием существующих, проверенных и производительных систем управления промышленного уровня.

Применение предложенного подхода откроет новые возможности для проектирования и реализации современных интеллектуальных и экспертных программных систем, использование которых особенно актуально в настоящее время.

### **Особенности представления данных и знаний в интеллектуальных информационных системах**

Под информацией принято понимать сведения о различных материальных и нематериальных сущностях окружающей среды (объектах или процессах) в рамках выбранной предметной области. У каждого объекта или процесса имеется собственное уникальное имя (название) и совокупность характеристик (признаков, свойств), отличающих его от других. Каждое отдельное свойство, характеризующее какую-либо величину, может принимать свои допустимые значения в пределах определённого диапазона. Количественные свойства характеризуются численными значениями, а качественные – символьными. В информатике и исследованиях по искусственному интеллекту вся информация подразделяется на две большие категории: данные и знания [2-4]. Разница между ними заключается в следующем:

1. Данные – формализованная информация в цифровом виде о различных сущностях, отражающая имеющиеся у них свойства, а также их количественные и качественные значения. Таким образом, данные представляют собой факты, отражающие простые утверждения об объектах и их характеристиках в статичной форме. Данные различных объектов могут быть взаимосвязаны в большом объеме (большие данные) и структурно организованы для длительного хранения в виде одного или нескольких файлов – базы данных (БД) [5]. Основной используемой математической моделью для построения баз данных является реляционная модель [6]. Кроме реляционной также существуют и некоторые нереляционные модели [7, 8]. Для выполнения действий по вводу, обработке, корректировке, поиску и выводу данных разрабатывают и используют системы управления базами данных (СУБД) [9]. В настоящее время существует огромное количество широко распространенных реляционных (табличные, колоночные, объектные) и нереляционных (ключ-значение, документные, графовые) СУБД, которые совместно используются различными информационными системами.

2. Знания – формализованная информация в цифровом виде о различных сущностях, отражающая имеющиеся у них свойства, а также множество законов отображения, которые отражают взаимосвязи и принципы получения новых количественных или качественных значений свойств. Таким образом, знания представляют собой совокупность исходных данных (фактов) и метаданных (правил вывода) по преобразованию исходных в динамической форме. Правила отражают закономерности предметной области и служат основой для определения и вывода результатов из взаимосвязанных комбинаций фактов. Как и в случае с данными, для организации и хранения знаний требуется построение баз знаний (БЗ). Для построения и организации баз знаний разрабатывают различные формальные модели [10, 11]. Одними из наиболее часто ис-

пользуемых математических моделей для представления знаний являются графы [12, 13]. Для выполнения действий по добавлению, обработке, корректировке, поиску и выводу знаний разрабатывают и используют системы управления базами знаний (СУБЗ). Они позволяют обеспечивать подключение и загрузку знаний, актуализацию и поддержание базы в достоверном состоянии, расширение и включение новых знаний, а также выполнять формирование и вывод решений задач из выбранных предметных областей. Базы знаний преимущественно используются в специализированных экспертных и интеллектуальных системах для решения разнообразных задач, связанных с логико-математическим моделированием искусственного интеллекта, а также с правдоподобными рассуждениями и выводом результатов на основе знаний [14-19].

Проанализировав приведенные выше сведения, стоит отметить, что между данными и знаниями есть некоторые сходства и отличия. Можно предположить, что базы знаний представляют собой особого рода базы данных. Но из-за необходимости хранения и использования правил вывода их структурная организация, в отличие от баз данных, является более сложной. Поэтому, для управления базами знаний часто требуется разработка собственных систем управления, а для управления базами данных существует множество предлагаемых к использованию СУБД.

### Модель базы знаний предметной области

Предложенная формальная модель базы знаний ( $KB$ ) предметной области представлена следующим образом:

$$KB = \{V_t, G(V, E), S, F\}, \quad (1)$$

где:

$G(V, E)$  – мультиграф знаний предметной области, в котором  $V = \{v_i\}, i = 1, 2, \dots, n$  – множество всех вершин, символически описывающих понятия (величины) выбранной предметной области,  $E = \{e_j\}, j = 1, 2, \dots, m$  – множество связей между вершинами и подмножествами других вершин;

$V_t = \{w_k\}, k = 1, 2, \dots, l$  – подмножество терминальных символов из множества  $G(V, E)$  обозначающих изначально известные величины из постановки задачи и используемые в процессе поиска решения;

$S$  – аксиома (нетерминальный символ из множества  $G(V, E)$  означающий то, что нужно найти в процессе решения задачи),  $S \in G(V, E)$ ;

$F = \{f_r\}, r = 1, 2, \dots, q$  – множество функций вида:  $f_r = f(x_1, x_2, \dots, x_n)$ , ассоциативно связанных с подмножествами различных вершин ( $f_r \rightarrow e_j$ ) графа  $G(V, E)$ , необходимых для вычисления сформированных математических выражений и получения численных результатов решений.

В качестве предметной области для базы знаний используется подобласть физики – «Механика», в рамках которой может быть выполнено решение множества различных задач. Постановка задачи задаётся на основе имеющихся знаний из базы, а решение задачи представляет собой абстрактную математическую модель с описанием выполнения процесса функционирования некоторой



линейной или нелинейной динамической системы [20]. Базу знаний в определенный момент можно пополнять, внося новые взаимосвязанные величины (переменные) и законы их отображения (функции). Предполагается, что чем больше величин и законов отображения будет внесено в базу знаний, тем больший круг задач можно будет решить в рамках выбранной предметной области.

Отличительной особенностью предложенной модели (1) является использование мультиграфа для организации и хранения знаний предметной области. Ключевой компонент модели базы знаний представлен в виде ориентированного, невзвешенного, циклического мультиграфа знаний  $G(V, E)$ . По сравнению с семантическими сетями [21, 22], мультиграфы могут иметь кратные дуги (рёбра) параллельно соединяющие множества вершин как в прямом, так и в обратном направлении. На рис. 1 представлен пример построенного мультиграфа для выбранной предметной области «Механика».

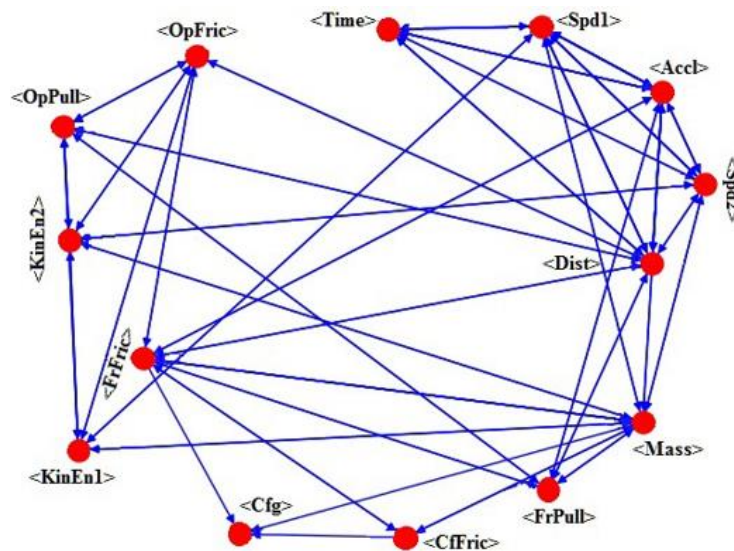


Рис. 1. Мультиграф знаний выбранной предметной области «Механика»

Символьные обозначения вершин мультиграфа означают следующие величины предметной области:

«*Dist*» – расстояние; «*Time*» – время; «*Spd1*» – начальная скорость; «*Spd2*» – конечная скорость; «*Accl*» – ускорение; «*Mass*» – масса; «*KinEn1*» – начальная кинетическая энергия; «*KinEn2*» – конечная кинетическая энергия; «*FrPull*» – сила тяги; «*FrFric*» – сила трения; «*OpPull*» – работа силы тяги; «*OpFric*» – работа силы трения; «*CfFric*» – коэффициент трения; «*Cfg*» – ускорение свободного падения.

Взаимосвязи между вершинами мультиграфа можно задавать в виде списка смежности. В таком случае, например, один выбранный элемент из списка вида: «*Spd2* → (*Spd1*, *Accl*, *Time*)» будет означать то, что из вершины «*Spd2*» исходят связанные с ней вершины: «*Spd1*», «*Accl*» и «*Time*».

Формально формирование и получение решения какой-либо задачи с использованием базы знаний будет представлять собой поиск на мультиграфе  $G(V, E)$  такого слабосвязанного подграфа, который должен содержать в себе



```
f14 = lambda x, y: ((2 * x) / (y * y))
f15 = lambda x, y, z: (x * y * z)
f16 = lambda x, y: (0.5 * x * y * y)
f17 = lambda x, y, z: ((x - y) + z)
f18 = lambda x, y: (x * y)
...
```

## Листинг 2. Заданные связанные функции

При использовании структур данных *Dict* и *OrderedDict* в языке программирования Питон (Python) поиск подмножеств смежных вершин происходит очень быстро за константное время. При этом добавление названий связываемых функций в подмножества смежных вершин ускоряет процесс их нахождения и запуска. После чего запускаемые в виде кода функции выполняют процедуры обработки необходимых найденных вершин из подмножеств, которые передаются им в качестве аргументов.

Представление баз знаний с мультиграфом подобным образом оказывается крайне удобным и реализуемым. Но есть несколько проблем с которыми приходится сталкиваться при такой организации баз знаний. Во-первых, базы знаний требуют постоянного пополнения с течением времени. Это означает, что файлы с кодом реализующем структуру мультиграфа будут постоянно увеличиваться в размере. Во-вторых, пополнять базы знаний, внося изменения прямо в файлы с кодом не совсем правильно. Ведь в какой-то момент времени это может привести к замедлению выполнения кода всей программной системы. В связи с этим возникает необходимость отделения структуры данных мультиграфа от основного кода и расположения её в отдельном файле. При этом возникает необходимость в организации создания, записи и чтения такого вида файлов. Кроме того, при последовательном чтении (построчном или поблочном) большого объема данных из подобных файлов [24], а также их передачи через сеть Интернет может возникнуть проблема долгого ожидания результата (тайм-аут). Подобные проблемы уже решены во многих современных СУБД промышленного уровня. Поэтому хорошим вариантом будет являться преобразование предложенной модели базы знаний предметной области на основе мультиграфа в реляционную модель БД, используя при этом существующие автоматизированные СУБД.

### Подход к преобразованию модели базы знаний в базу данных

Предложенный подход заключается в построении структуры базы данных на основе реляционной модели, представляющей собой описание совокупности определенных взаимосвязанных таблиц. Формальная модель такой базы данных может быть представлена как множество таблиц ( $T$ ) вида:

$$T = \{t_1, t_2, t_i, \dots, t_n\}, \quad (2)$$

в котором каждая таблица  $t_i$  состоит из множества доменов ( $D$ ) с определенными именами, заголовка таблицы ( $H$ ) с именами её столбцов и тела таблицы ( $B$ ) с заданными значениями хранимых ею данных:

$$t_i = \{D, H, B\},$$

где:

$D = \{ \langle d_1:s_1 \rangle, \langle d_2:s_2 \rangle, \langle d_j:s_j \rangle, \dots, \langle d_m:s_m \rangle \}$  – множество доменов с заданными именами, определенных в виде пар «имя домена : диапазон значений». Каждый домен задает имя  $d_j$  для типа хранимых данных (целых чисел, вещественных чисел, символов и др.) и определяет диапазон допустимых значений  $s_j$ . Например, задание диапазона для хранения информации в виде целых положительных чисел можно описать как:  $s_j = f(n)$  ( $n \geq 0$  и  $n \leq 2^b$ ), где  $n$  – элемент из множества всех натуральных чисел  $N$ ,  $b$  – количество бит информации, отведенное для хранения целых чисел в ячейках оперативной памяти вычислительной системы;

$H = \{ \langle c_1:d_1 \rangle, \langle c_2:d_2 \rangle, \langle c_k:d_j \rangle, \dots, \langle c_o:d_m \rangle \}$  – множество столбцов с именами, заданные в виде пар «имя столбца : имя домена» и позволяющие находить все связанные данные одного заданного типа  $d_j \in D$  по указанному имени столбца  $c_k$ ;

$B = \{ \langle c_1:r_1 \rangle, \langle c_2:r_2 \rangle, \langle c_k:r_i \rangle, \dots, \langle c_o:r_p \rangle \}$  – множество строк позволяющих хранить данные, заданные в виде пар «имя столбца : значения строки». Для любой такой пары строка  $r_i$  содержит множество значений  $\{x_1, x_2, \dots, x_q\}$  из определенного домена  $d_j \in D$ , который связан с именем столбца  $c_k \in H$ .

Для алгоритмической реализации, приведенной выше модели (2), можно использовать группу из ассоциативно связанных структур данных, представляющих собой хеш-таблицы, которые могут быть записаны на физический диск вместе с хранимой информацией в виде одного или нескольких файлов. Таким образом, можно построить собственный формат для хранения информации в виде файлов базы данных и собственную СУБД для управления ими. В связи с тем, что реализация собственного формата хранения данных и системы управления может оказаться очень долгим и трудоемким процессом, хорошим вариантом будет являться выбор и использование какой-либо из уже существующих СУБД: реляционных (PostgreSQL, MySQL, ClickHouse), колоночных (HBase, Cassandra, CosmosDB), документных (MongoDB, Firebase, CouchDB), ключ-значение (Redis, Memcached, KeyDB), объектных (ObjectDB, ObjectStore, db4o), графовых (Neo4j, Memgraph, ArangoDB) или других [25].

Хорошим вариантом кажется применение графовых СУБД [26]. Но различные модели графов (ориентированные, неориентированные, двудольные, мультиграфы, гиперграфы и другие) отражают разные структуры представления знаний, которые либо не в полной мере, либо совсем не реализованы в подобных СУБД. Например, большинство графовых СУБД не поддерживают хранение и обработку мультиграфов. По этой причине, а также в связи с тем, что самым популярным видом СУБД являются – реляционные, было решено организовать хранение мультиграфа знаний с использованием реляционной СУБД. В итоге для создания базы данных была выбрана существующая СУБД – PostgreSQL [27, 28] и её графическая оболочка – pgAdmin4.

Используя редактор кода из оболочки, взаимодействующей с выбранной СУБД, была создана база данных (*Mechanics*) и главная таблица (*Knowledges*). Главная таблица содержит знания из выбранной предметной области.



В листинге 3 приведен код на языке программирования SQL, используемый для создания базы данных и главной таблицы:

```
CREATE DATABASE "Mechanics" WITH
  OWNER = postgres
  ENCODING 'UTF8'
  LOCALE 'Russian_Russia.1251'
  TEMPLATE template0;

CREATE TABLE Knowledges (
  idv BIGSERIAL PRIMARY KEY,
  name VARCHAR(60) NOT NULL,
  symname VARCHAR(30) NOT NULL,
  interrelations JSON
);
```

### Листинг 3. Код создания базы данных и главной таблицы

Имена столбцов главной таблицы содержат метаинформацию, необходимую для определения знаний: величин (*name*, *symname*), законов их отображения (*interrelations*) и обозначений для их идентификации (*idv*). В строках таблицы хранятся заданные знания в виде символьных обозначений самих величин и правил их отображения с определенными взаимосвязанными функциями от  $f_1$  до  $f_n$ , которые представлены на рис. 2.

	name character varying (60)	symname character varying (30)	interrelations json
1	Сила трения	<FrFric>	[["f3", "<FrPull>", "<Mass>", "<Accl>"], ["f15", "<Cfg>", "<CfFric>", "<Mass>"]]
2	Работа силы трения	<OpFric>	[["f18", "<FrFric>", "<Dist>"], ["f17", "<OpPull>", "<KinEn2>", "<KinEn1>"]]
3	Работа силы тяги	<OpPull>	[["f18", "<FrPull>", "<Dist>"], ["f17", "<KinEn2>", "<KinEn1>", "<OpFric>"]]
4	Конечная скорость	<Spd2>	[["f1", "<Spd1>", "<Accl>", "<Time>"], ["f2", "<Accl>", "<Dist>", "<Spd1>"]]
5	Начальная скорость	<Spd1>	[["f3", "<Spd2>", "<Accl>", "<Time>"], ["f4", "<Dist>", "<Time>", "<Acc>"]]
6	Расстояние	<Dist>	[["f6", "<Spd1>", "<Time>", "<Accl>"], ["f7", "<Spd2>", "<Spd1>", "<Acc>"]]
7	Время	<Time>	[["f8", "<Spd2>", "<Spd1>", "<Accl>"], ["f9", "<Spd1>", "<Accl>", "<Dis>"]]
8	Ускорение	<Accl>	[["f8", "<Spd2>", "<Spd1>", "<Time>"], ["f10", "<Dist>", "<Spd1>", "<T>"]]
9	Масса	<Mass>	[["f8", "<FrPull>", "<FrFric>", "<Accl>"], ["f13", "<FrFric>", "<CfFric>", "<Mass>"]]
10	Сила тяги	<FrPull>	[["f1", "<FrFric>", "<Mass>", "<Accl>"], ["f12", "<OpPull>", "<Dist>"]]

Рис. 2. Представление и хранение знаний в виде таблицы

Стоит отметить то, что множество величин из всех строк по столбцу *symname* отражают – декларативные знания, а множество законов отображения и связанные с ними функции из всех строк по столбцу *interrelations* отражают – процедурные знания.

Наибольший интерес представляет преобразование вершин и связей мультиграфа из модели базы знаний (1). Это преобразование для базы данных выполнено следующим способом: поскольку структура мультиграфа знаний может быть представлена в виде списка смежности, то в таблице базы данных главный список будет представлен совокупностью всех строк по столбцу *symname*, а смежные списки совокупностью всех строк по столбцу *interrelations*. В итоге объединение выбранных строк с информацией по столбцам *symname* и *interrelations* приведёт к образованию структуры, подобной мультиграфу. При этом важно отметить то, что для упрощения такого преобра-

зования знания по столбцу *interrelations* представлены в символьном формате JavaScript Object Notation (JSON) который был предложен Дугласом Крокфордом [29]. В таком формате все связанные вершины в списках перечисляются через запятую и заключаются в квадратные скобки. Таким образом, все связанные списки являются содержимым одной большой символьной строки.

СУБД PostgreSQL позволяет обрабатывать информацию в символьном представлении, используя одноименный тип данных – json. Это даёт возможность параллельно выполнять извлечение и объединение множества символьных строк, что позволяет ускорить обработку знаний, хранимых в символьном виде. В настоящее время большинство современных языков программирования уже имеют все необходимые операции по преобразованию символьной информации из формата JSON в хорошо интерпретируемые алгоритмические структуры данных, такие как объекты и массивы. Таким образом, использование символьного формата JSON в СУБД позволяет упростить представление, хранение и обработку знаний.

В листинге 4 приведен фрагмент кода на языке программирования SQL, реализующий операцию добавления знаний в главную таблицу:

```
INSERT INTO Knowledges (name, symname, interrelations)
VALUES (
  'Сила трения',
  '<FrFric>',
  '[[["f3", "<FrPull>", "<Mass>", "<Accl>"], ["f15", "<Cfg>", "<CfFric>", "<Mass>"],
    ["f12", "<OpFric>", "<Dist>"]]']
);
INSERT INTO Knowledges (name, symname, interrelations)
VALUES (
  'Работа силы трения',
  '<OpFric>',
  '[[["f18", "<FrFric>", "<Dist>"], ["f17", "<OpPull>", "<KinEn2>", "<KinEn1>"]]']
);
INSERT INTO Knowledges (name, symname, interrelations)
VALUES (
  'Работа силы тяги',
  '<OpPull>',
  '[[["f18", "<FrPull>", "<Dist>"], ["f17", "<KinEn2>", "<KinEn1>", "<OpFric>"]]']
);
...
```

#### Листинг 4. Фрагмент кода для добавления знаний в таблицу

Кроме возможности использования известного языка SQL для создания, добавления, изменения и удаления данных, многие существующие СУБД имеют возможность работы через сеть Интернет и могут выполнять обработку множества входящих сообщений (в том числе параллельных) с запросами на выборку и передачу информации [30].

Отличительной особенностью использования баз данных для представления и управления знаниями является и возможность добавления новых дополнительных таблиц, которые могут быть взаимосвязаны друг с другом. Это позволяет добавлять новую информацию, необходимую для расширения возможностей и функционала программных систем, взаимодействующих с базой. Например, можно добавить дополнительную таблицу для хранения единиц

измерения величин (с целью более удобного и понятного отображения результатов), таблицу для хранения имён пользователей (с целью их регистрации и работы с системой через Интернет), таблицу с двоичным представлением кода всех вычислительных функций (с целью их отдельной загрузки и экономии памяти) [31] и многие другие.

В листинге 5 приведен код алгоритма взаимодействия с построенной базой данных:

```
def call_loadAllLinesFromKBFile(kbname = "Mechanics"):  
    from collections import OrderedDict  
    conn = connectToDB()  
    MG = OrderedDict()  
    try:  
        cur = conn.cursor()  
        cur.execute("SELECT name, symname, interrelations FROM Knowleges;")  
        rows = cur.fetchall()  
        for row in rows:  
            MG.update({row[1] : row[2]})  
    except BaseException as err:  
        print(f"Произошла ошибка: {err=}, {type(err)=}")  
        conn.close()  
        return False  
    else:  
        conn.close()  
        return MG
```

#### Листинг 5. Код алгоритма взаимодействия с базой данных

Блок с кодом алгоритма в терминологии языка Питон (Python) определён в виде одной функции. В начале выполняется вызов функции (*connectToDB*), в которой при помощи модуля *psycopg* осуществляется сетевое подключение к СУБД PostgreSQL. После подключения выполняется отправка сообщения в СУБД с SQL-запросом на выполнение команды «*SELECT...*» предназначенной для выборки знаний из всех строк таблицы *Knowleges* по столбцам *name*, *symname* и *interrelations*. Все знания считываются СУБД из файлов базы, передаются и записываются в переменную *rows*, образуя двумерный массив. Далее из полученных значений массива формируется множество пар (*{row[1]:row[2]}*), объединение которых приводит к образованию списка смежности и организации структуры данных *OrderedDict*. В конце сформированная структура, отражающая мультиграф знаний возвращается в качестве результата (*MG*) из функции и может быть передана на вход другим отдельным алгоритмам, входящим в состав более сложных программных комплексов или информационных систем.

### Результаты и их обсуждение

На основе предложенного подхода выполнено решение поставленной задачи по преобразованию существующей модели базы знаний (1) в реляционную модель базы данных. С использованием существующей СУБД (PostgreSQL) разработана база данных для хранения и управления знаниями. База данных предназначена для решения задач из выбранной предметной обла-

сти «Механики». На языках программирования Питон (Python) и SQL были разработаны алгоритмы для взаимодействия с базой, а также реализован прототип программной экспертной вычислительной системы для моделирования динамических систем. Были проведены вычислительные эксперименты, подтверждающие работоспособность прототипа программной системы и реализованных алгоритмов взаимодействия с базой.

В качестве примера рассмотрим вывод решения на основе знаний для следующей задачи:

*«За какое время автомобиль, двигаясь с ускорением  $0,4 \text{ м/с}^2$ , увеличит свою скорость с 12 до 20 м/с?»».*

Для запуска прототипа программной системы используется стандартный интерпретатор языка Питон (Python) – CPython. Это бесплатное программно-инструментальное средство, которое доступно для загрузки и изучения на официальном сайте [python.org](https://python.org), размещенном в сети Интернет. В качестве операционной системы используется Windows 11. Использование интерпретатора осуществляется путем его запуска через утилиту командной строки (cmd.exe) и передачи ему файла с кодом алгоритмов работы программной системы. После запуска программная система запрашивает ввод и ожидает поступления исходных знаний (аксиомы и известных величин) о решаемой задаче в необходимом формате (JSON). Листинг 6 отображает процесс запуска программной системы и ввода исходных знаний по решаемой задаче.

```
> python system.py

Введите аксиому:
<Time>

Введите знания в формате: {'величина' : значение}
{'<Acc1>' : 0.4, '<Spd1>' : 12, '<Spd2>' : 20}
```

#### Листинг 6. Ввод исходных знаний в программную систему

Так как в решаемой задаче требуется найти время, то в качестве аксиомы вводится величина – «*Time*». В качестве известных из постановки задачи величин вводятся: ускорение – «*Acc1*», начальная скорость – «*Spd1*» и конечная скорость – «*Spd2*». После ввода необходимых обозначений величин (терминальных вершин) и соответствующих им числовых значений программная система начинает процесс поиска слабосвязанного подграфа, представляющего собой символьный результат решения задачи. Если будет найден подграф, включающий в себя подмножество всех необходимых терминальных вершин, то будет выведено соответствующее сообщение. Далее все найденные вершины будут выведены системой на экран в виде линейной суперпозиции функций, в которой нетерминальные вершины будут заменены именами ассоциативно связанных с ними вычислительных функций, а все связанные с ними терминальные вершины будут обрамлены круглыми скобками. Таким образом, все терминальные вершины становятся аргументами функций, и при подстановке фактических числовых значений произойдет вычисление итогового численного результата.



Вывод результата решения задачи отображён в листинге 7.

```
Решение найдено!  
f8 (<Spd2>, <Spd1>, <Acc1>)
```

```
Ответ = 20.0
```

#### Листинг 7. Вывод результата решения задачи

Если будет введена недопустимая информация или по введенным знаниям не удастся осуществить поиск решения из-за неполноты базы знаний, то программная система отобразит сообщение об отсутствии решения.

Рассмотрим еще один пример формирования и вывода решения на основе знаний из базы для следующей задачи:

*«Самолет летит горизонтально на высоте  $H = 4000$  м. над поверхностью Земли со сверхзвуковой скоростью. Звук дошел до наблюдателя через время  $t = 10$  с. после того, как над ним пролетел самолет. Определить скорость самолета  $V$ , если скорость звука  $v_0 = 330$  м/с?».*

Для решения данной задачи вначале необходимо добавить в базу все знания о новых необходимых величинах и законы их отображения. После запуска программной системы введём исходные знания по решаемой задаче (аксиому и известные из постановки задачи величины) в требуемом формате. Листинг 8 отображает процесс запуска системы и ввода исходных знаний.

```
> python system.py
```

```
Введите аксиому:  
<Spd2>
```

```
Введите знания в формате: {'величина' : значение}  
{ '<Height>':4000, '<Time>':10, '<Spd1>':330}
```

#### Листинг 8. Запуск системы и ввод исходных знаний

Если решение будет найдено, алгоритмы системы построят и выведут суперпозицию функций, представляющую собой символьный (условно аналитический) результат решения задачи:

$$f20(\langle Height \rangle, \langle Time \rangle, \langle Spd1 \rangle).$$

Далее, за счет подстановки численных значений и выполнения сформированной в виде кода функции ( $f20$ ), которая представлена следующей формулой:

$$V = \frac{H \cdot v_0}{\sqrt{H^2 - (v_0^2 \cdot t^2)}},$$

будет получен численный результат решения задачи. Вывод результата решения показан в листинге 9.

Решение найдено!  
f20 (<Height>, <Time>, <Spd1>)

Ответ: 583,933

### Листинг 9. Вывод результата решения

Как можно видеть, результаты решения задач, полученные программной системой на основе знаний из базы, действительно соответствуют правильным ответам. Работоспособность прототипа реализованной программной системы с используемой базой данных была проверена при решении нескольких десятков задач из области механики.

Таким образом, используя разработанную базу данных, можно выполнять автоматизированную обработку и управление знаниями из выбранных предметных областей, а также осуществлять моделирование динамических систем с формированием и выводом решений для множества различных задач.

### Выводы

В результате исследования были решены следующие задачи:

1. Проведен сравнительный анализ, приведены определения, основные особенности и отличия *данных* и *знаний*.

2. Описан процесс построения формальной модели базы знаний на основе мультиграфа, предназначенной для моделирования динамических систем и решения задач из выбранной предметной области. Показана алгоритмическая реализация модели базы знаний на выбранном языке программирования. Выявлены проблемы по взаимодействию с предложенной моделью базы знаний, заключающиеся в необходимости её физической организации в виде отдельных файлов, постоянном увеличении размера файлов, чтении большого объёма данных, а также в передаче информации через распределённую сеть Интернет в следствие необходимости оперативного пополнения базы.

3. Разработан, описан и реализован подход по преобразованию предложенной модели базы знаний предметной области на основе мультиграфа в реляционную модель базы данных, для решения выявленных проблем. Рассмотрены модель БД, типы данных, символьный формат и алгоритмические структуры представления знаний с особенностями их применения в выбранной реляционной СУБД. Приведены листинги с фрагментами кода на языках программирования для взаимодействия с БД, а также примеры добавления, извлечения и обработки знаний с их интерпретацией.

Предложенный подход дает возможность организации и хранения знаний в виде базы данных, что позволяет более эффективно выполнять обработку и управление знаниями с применением реляционной модели. Кроме того, управление знаниями осуществляется с использованием существующей, проверенной и производительной системой управления промышленного уровня. В будущем предложенный подход планируется использовать для развития и совершенствования прототипа программной экспертной вычислительной системы моделирования линейных и нелинейных динамических систем.

## Литература

1. Зайцев А. Ф. Обзорный анализ и классификация моделей представления знаний в компьютерных интеллектуальных системах // Способы, модели и алгоритмы управления модернизационными процессами: сборник статей Международной научно-практической конференции (Пермь, 20 апреля 2023 г.). – Уфа: Аэтерна, 2023. – С. 31–43.
2. Russell S. J., Norvig P. Artificial intelligence: A modern approach, Fourth global edition. – London: Pearson Education Limited, 2021. – 1166 p.
3. Загорюлько Ю. А., Загорюлько Г. Б. Искусственный интеллект. Инженерия знаний. – М.: Издательство Юрайт, 2022. – 93 с.
4. Одинцов Б. Е. Модели и проблемы интеллектуальных систем: монография. – М.: ИНФРА-М, 2024. – 219 с.
5. Комаров В. И. Путеводитель по базам данных. – М.: ДМК-Пресс, 2024. – 520 с.
6. Куприянчик Е. М., Зудилова Т. В., Ананченко И. В. Сравнительные характеристики SQL и NOSQL СУБД, влияющие на разработку приложений баз данных // Современные наукоемкие технологии. 2022. № 7. С. 74–78. doi: 10.17513/snt.39236.
7. Еремеев А. П., Панявин Н. А. Унификация модели представления данных и преобразование форматов на основе нереляционной СУБД Neo4j // Программные продукты и системы. 2022. № 4. С. 549–556. doi 10.15827/0236-235X.140.549-556.
8. Briukhov D. O., Stupnikov S. A. Logical relational model of data structures for problem solving in land use management // Informatics and applications. 2022. vol. 16. no. 4. pp. 93–98. doi: 10.14357/19922264220414.
9. Sciore E. Database design and implementation. Second edition. – Cham: Springer international publishing, 2020. – 458 p.
10. Lipnitsky S. F. Model of knowledge representation in the system of information support for decision making // Informatics. 2022. vol. 19. no. 3. pp. 40–49. (in Russian). doi: 10.37661/1816-0301-2022-19-3-40-49.
11. Варламов О. О. Создание больших знаний и расширение областей применения миварных технологий логического искусственного интеллекта // Информационные и математические технологии в науке и управлении. 2023. № 4 (32). С. 30–41. doi: 10.25729/ESI.2023.32.4.003.
12. Рамазанова В. С., Самбетбаева М. А., Загорюлько Ю. А. Обзор и анализ представлений графов знаний // Труды университета. 2024. № 1 (94). С. 413–420. doi: 10.52209/1609-1825\_2024\_1\_413.
13. Никольский Д. Р. Анализ графовых систем управления базами данных // Вестник Воронежского государственного технического университета. 2023. Т. 19. № 6. С. 13–20. doi: 10.36622/VSTU.2023.19.6.002.
14. Андрейчиков А. В., Андрейчикова О. Н. Интеллектуальные информационные системы и методы искусственного интеллекта. – М.: ИНФРА-М, 2023. – 530 с. doi: 10.12737/1009595.

15. Demidovskij A. V., Babkin E. A. Integrated neurosymbolic decision support systems: problems and opportunities // *Business Informatics*. 2021. vol. 15. no. 3. pp. 7–23. doi: 10.17323/2587-814X.2021.3.7.23.

16. Efimova E. A. Implementation of a plausible reasoning in the Prolog programming language // *Automatic documentation and mathematical linguistics*. 2023. vol. 57. no. 3. pp. 166–171. doi: 10.36535/0548-0027-2022-09-2.

17. Bosov A. V., Zhukov D. V. Expert system for monitoring and forecasting of resource allocation processes // *Informatics and applications*. 2021. vol. 15. no. 3. pp. 29–40. doi: 10.14357/19922264210305.

18. Beniaminov E. M. Approaching the standard of knowledge representation in the Web by algebraic means // *Automatic documentation and mathematical linguistics*. 2022. vol. 56. no. 5. pp. 265–273. doi: 10.36535/0548-0027-2022-10-2.

19. Gavrilin D. N., Kustova I. A., Mantsivoda A. V. Object models as microservices: a query language // *The Bulletin of Irkutsk State University. Series Mathematics*. 2022. vol. 42. pp. 121–137. (in Russian). doi: 10.26516/1997-7670.2022.42.121.

20. Zaytsev A. F., Kravchenko V. A., Shirapov D. Sh. An approach to logical-mathematical computer modeling of linear and nonlinear dynamical systems // *E3S Web of Conferences*. vol. 583. 2024. pp. 06014. doi: 10.1051/e3sconf/202458306014.

21. Бурмистров А. А. Семантические сети как способ представления знаний // *Наукосфера*. 2023. № 3(2). С. 163–166. doi: 10.5281/zenodo.7788675.

22. Волчихин В. И. Представление и структурирование знаний в семантико-ориентированной вычислительной среде. Часть I. Интеграция концептуальных графов и логических сетей на основе формализации структурированных ситуаций // *Известия высших учебных заведений. Поволжский регион. Технические науки*. 2023. № 2(66). С. 24–51. doi: 10.21685/2072-3059-2023-2-3.

23. Зайцев А. Ф. Метод поиска и логического вывода экспертной информации в ориентированном циклическом мультиграфе знаний // *Информационные и математические технологии в науке и управлении*. 2022. № 4(28). С. 213–222. doi: 10.38028/ESI.2022.28.4.017.

24. Belov V. A., Illin D. Y., Nikulchev E. V. Comparative evaluation of the efficiency of data processing by storing the data in a relational database and column format files // *Computational technologies*. 2022. vol. 27. no. 3. pp. 46–65. (in Russian). doi: 10.25743/ICT.2022.27.3.005.

25. Бадмажапов А. Д. Сравнительный анализ производительности систем управления базами данных ArangoDB, MySQL и Neo4j // *Матрица научного познания*. 2024. № 5(2). С. 17–25.

26. Отрадных К. К., Алешкин А. С., Калинин В. Н. Преимущества использования графовых баз данных при разработке прикладных приложений // *Вестник Рязанского государственного радиотехнического университета*. 2023. № 86. С. 73–85. doi: 10.21667/1995-4565-2023-86-73-85.



27. Pirozzi E., Ferrari L. *Learn PostgreSQL: Use, manage and build secure and scalable databases with PostgreSQL 16*. Second Edition. – Birmingham: Packt Publishing, 2023. – 744 p.

28. Рогов Е. В. *PostgreSQL 16 изнутри*. – М.: ДМК Пресс, 2024. – 664 с.

29. Crockford D. RFC 4627. The application/json media type for JavaScript Object Notation (JSON). The Internet Society Publ., 2006. 10 p. doi: 10.17487/RFC4627.

30. Nasibullin A. R., Novikov B. A. Replication in distributed systems: models, methods, and protocols // *Programming and computer software*. 2020. vol. 46. no. 5. pp. 341–350. doi: 10.31857/S0132347420050064.

31. Pantilimonov M. V., Buchatskiy R. A., Zhuykov R. A. Machine code caching in PostgreSQL query JIT-compiler // *Proceedings of the Institute for System Programming of the RAS (Proceedings of ISP RAS)*. 2020. vol. 32. no. 1. pp. 205–220. (in Russian). doi: 10.15514/ISPRAS-2020-32(1)-11.

## References

1. Zaytsev A. F. Review analysis and classification of knowledge representation models in computer intelligent systems. *Sposoby, modeli i algoritmy upravleniya modernizatsionnymi protsessami. Sbornik statey Mezhdunarodnoy nauchno-prakticheskoy konferentsii* [Methods, models and algorithms of modernization processes management. Proceedings of the International scientific and practical conference]. Ufa, Aeterna, 2023, pp. 31–43 (in Russian).

2. Russell S. J., Norvig P. *Artificial intelligence: A modern approach, Fourth global edition*. London, Pearson Education Limited, 2021. 1166 p.

3. Zagorulko Yu. A., Zagorulko G. B. *Iskusstvennii intellekt. Inzheneriya znaniy* [Artificial Intelligence. Knowledge Engineering]. Moscow, Yurait publishing house, 2022. 93 p. (in Russian).

4. Odintsov B. Ye. *Modeli i problemi intellektualnikh sistem: monografiya* [Models and problems of intellectual systems: monograph]. Moscow, INFRA-M, 2024. 219 p. (in Russian).

5. Komarov V. I. *Putevoditel po bazam dannykh* [Guide to the databases]. Moscow, DMK-Press, 2024. 520 p. (in Russian).

6. Kupriyanchik E. M., Zudilova T. V., Ananchenko I. V. et al. Comparative characteristics of SQL and NOSQL DBMS affecting the development of database applications. *Modern high technologies*, 2022, no. 7, pp. 74–78 (in Russian).

7. Ereemeev A. P., Panyavin N. A. Data representation model unification and format transformation based on the non-relational DBMS Neo4j. *Programmnyye produkty i sistemy*, 2022, no. 4, pp. 549–556 (in Russian).

8. Briukhov D. O., Stupnikov S. A. Logical relational model of data structures for problem solving in land use management. *Informatics and applications*, 2022, vol. 16, no. 4, pp. 93–98. doi: 10.14357/19922264220414.

9. Sciore E. *Database design and implementation. Second edition*. Cham, Springer international publishing, 2020. 458 p.

10. Lipnitsky S. F. Model of knowledge representation in the system of information support for decision making. *Informatics*, 2022, vol. 19, no. 3, pp. 40–49. (in Russian). doi: 10.37661/1816-0301-2022-19-3-40-49.

11. Varlamov O. O. Sozдание bolshikh znanii i rasshirenie oblastei primeneniya mivarnikh tekhnologii logicheskogo iskusstvennogo intellekta [Creating big knowledge and expanding the applications of mivar technologies of logical artificial intelligence]. *Informational and mathematical technologies in science and management*, 2023, vol. 32, no. 4, pp. 30–41. (in Russian). doi: 10.25729/ESI.2023.32.4.003.

12. Ramazanova V. S., Sambetbaeva M. A., Zagorulko Yu. A. Obzor i analiz predstavlenii grafov znanii [Review and analysis of knowledge graph representations]. *Proceedings of the University*, 2024, vol. 94, no. 1, pp. 413–420. (in Russian). doi: 10.52209/1609-1825\_2024\_1\_413.

13. Nikolskii D. R. Analiz grafovikh sistem upravleniya bazami dannikh [Analysis of the graph systems of the database management]. *Bulletin of Voronezh State Technical University*, 2023, vol. 19, no. 6, pp. 13–20. (in Russian). doi: 10.36622/VSTU.2023.19.6.002.

14. Andreichikov A. V., Andreichikova O. N. Intellektualnie informatsionnie sistemi i metodi iskusstvennogo intellekta [Intellectual information systems and methods of the artificial intelligence]. Moscow, INFRA-M, 2023. 530 p. (in Russian). doi: 10.12737/1009595.

15. Demidovskij A. V., Babkin E. A. Integrated neurosymbolic decision support systems: problems and opportunities. *Business informatics*, 2021, vol. 15, no. 3, pp. 7–23. doi: 10.17323/2587-814X.2021.3.7.23.

16. Efimova E. A. Implementation of a plausible reasoning in the Prolog programming language. *Automatic documentation and mathematical linguistics*, 2023, vol. 57, no. 3, pp. 166–171. doi: 10.36535/0548-0027-2022-09-2.

17. Bosov A. V., Zhukov D. V. Expert system for monitoring and forecasting of resource allocation processes. *Informatics and applications*, 2021, vol. 15, no. 3, pp. 29–40. doi: 10.14357/19922264210305.

18. Beniaminov E. M. Approaching the standard of knowledge representation in the Web by algebraic means. *Automatic documentation and mathematical linguistics*, 2022, vol. 56, no. 5, pp. 265–273. doi: 10.36535/0548-0027-2022-10-2.

19. Gavrilin D. N., Kustova I. A., Mantsivoda A. V. Object models as microservices: a query language. *The Bulletin of Irkutsk State University. Series Mathematics*, 2022, vol. 42, pp. 121–137. (in Russian). doi: 10.26516/1997-7670.2022.42.121.

20. Zaytsev A. F., Kravchenko V. A., Shirapov D. Sh. An approach to logical-mathematical computer modeling of linear and nonlinear dynamical systems. *E3S Web of Conferences*, vol. 583, 2024, pp. 06014. doi: 10.1051/e3sconf/202458306014.

21. Burmistrov A. A. Semanticheskie seti kak sposob predstavleniya znanii [Semantic networks as a way of knowledge representation]. *Naukosfera*, 2023, vol. 2, no. 3, pp. 163–166. (in Russian). doi: 10.5281/zenodo.7788675.

22. Volchikhin V. I. Predstavlenie i strukturirovanie znanii v semantiko-orientirovannoi vichislitelnoi srede. Chast I. Integratsiya kontseptualnikh grafov i

logicheskikh setei na osnove formalizatsii strukturirovannikh situatsii [Knowledge representation and structuring in the semantic-oriented computing environment. Part I. Integration of the conceptual graphs and logical networks on the basis of the structured situations formalization]. *University proceedings. Volga region. Technical sciences*, 2023, vol. 66, no. 2, pp. 24–51. (in Russian). doi: 10.21685/2072-3059-2023-2-3.

23. Zaytsev A. F. Method of search and logical inference of expert information in a directed cyclic knowledge multigraph. *Informational and mathematical technologies in science and management*, 2022, vol. 28, no. 4, pp. 213–222. (in Russian). doi: 10.38028/ESI.2022.28.4.017.

24. Belov V. A., Illin D. Y., Nikulchev E. V. Comparative evaluation of the efficiency of data processing by storing the data in a relational database and column format files. *Computational technologies*, 2022, vol. 27, no. 3, pp. 46–65. (in Russian). doi: 10.25743/ICT.2022.27.3.005.

25. Badmazhapov A. D. Comparative analysis of the performance of ArangoDB, MySQL and Neo4j database management systems. *Matritsa nauchnogo poznaniya*, 2024, vol. 2, no. 5, pp. 17–25 (in Russian).

26. Otradnov K. K., Alyoshkin A. S., Kalinin V. N. Advantages of using graph databases in the development of applied applications. *Vestnik of Ryazan state radioengineering university*, 2023, no. 86, pp. 73–85 (in Russian).

27. Pirozzi E., Ferrari L. *Learn PostgreSQL: Use, manage and build secure and scalable databases with PostgreSQL 16. Second Edition*. Birmingham, Packt Publishing, 2023. 744 p.

28. Rogov Ye. V. *PostgreSQL 16 iznutri* [PostgreSQL 16 internals]. Moscow, DMK Press, 2024. 664 p. (in Russian).

29. Crockford D. RFC 4627. *The application/json media type for JavaScript Object Notation (JSON)*. Reston, The Internet Society, 2006. 10 p. doi: 10.17487/RFC4627.

30. Nasibullin A. R., Novikov B. A. Replication in distributed systems: models, methods, and protocols. *Programming and computer software*, 2020, vol. 46, no. 5, pp. 341–350. doi: 10.31857/S0132347420050064.

31. Pantilimonov M. V., Buchatskiy R. A., Zhuykov R. A. Machine code caching in PostgreSQL query JIT-compiler. *Proceedings of the Institute for System Programming of the RAS (Proceedings of ISP RAS)*, 2020, vol. 32, no. 1, pp. 205–220. (in Russian). doi: 10.15514/ISPRAS-2020-32(1)-11.

Статья поступила 18 сентября 2024 г.

### Информация об авторе

Зайцев Анатолий Федорович – старший преподаватель кафедры «Вычислительные и радиоэлектронные системы». Восточно-Сибирский государственный университет технологий и управления. Область научных интересов: системный анализ, моделирование и разработка программных систем, информационные технологии, языки программирования, структуры и алгоритмы обра-

ботки данных, системное программирование, операционные системы, базы данных, базы знаний, интеллектуальные системы, искусственный интеллект и машинное обучение. E-mail: lordsadler2010@mail.ru

Адрес: 670000, Россия, г. Улан-Удэ, ул. Смолина, д. 26.

---

## An approach to transforming a model of multigraph knowledge base of the subject area into a relational database model

A. F. Zaytsev

**Purpose.** The problem of building modern software-intelligent and expert systems requires solving a variety of different tasks. At present, the complexity of building such systems lies in the fact that the known 'classical' models and approaches to the construction of expert systems do not agree well with the relational model of data representation, which makes it impossible to effectively use modern industrial database management systems for the organization and representation of knowledge, as well as their use in various information systems. The purpose of the work is to transform the proposed multigraph knowledge base model of the subject area into a relational database model to improve the efficiency of information processing using modern automated control systems. **Methods.** The approach allowing the transformation and storage of knowledge of subject areas in tabular representation and relational databases are proposed. The approach reflects the principles and rules for performing the decomposition of the knowledge base model with a multigraph, representation of the knowledge multigraph using algorithmic data structures (hash-tables), storing the knowledge multigraph in the tabular form of the database using the selected symbolic format (JSON), as well as performing the reverse transformation and aggregation of the multigraph vertex lists from relational to non-relational form. The following generals' scientific methods were used: analysis, decomposition, synthesis, modeling, formalization, structurization, algorithmization, generalization, comparison, description, computational experiment, as well as set theory and graph theory, on the basis of which the used model of knowledge representation was built. Computers and software tools – database management systems, programming languages, and their interpreters – were used as research materials. **Novelty.** The elements of novelty are a new mathematical model of representation of knowledge of the subject area on the basis of the multigraph, as well as the proposed approach to the transformation of the knowledge base model on the basis of the multigraph into a relational database model. **Results.** A comparative analysis is carried out, and the main definitions, features, and differences between data and knowledge are given. The process of building the multigraph knowledge base model designed to solve computational problems in the selected subject area is described. Problems of interaction with the multigraph knowledge base are identified and described. An approach to solving the identified issues was developed and applied, which consists of transforming the multigraph knowledge base model of the subject area into a relational database model. Types, formats, and structures of knowledge representation and the peculiarities of their application in the selected management system are considered. **Practical relevance.** The software implementation of the knowledge base in the selected programming language is shown. Listings with code fragments in programming languages for interaction with the base, as well as examples of knowledge representation and processing, are given. The implementation of the proposed approach makes it possible to organize and store knowledge in the form of databases, which allows for more efficient knowledge processing and management using a relational model. In addition, knowledge management is performed using existing, proven, and productive industrial-level database management systems. The use of the proposed approach opens new opportunities for the design and implementation of modern software-intelligent and expert systems.

**Key words:** knowledge representation problem, knowledge bases, databases, dynamical systems modeling, expert systems, knowledge multigraph, expert information, intelligent systems, programming and automation, software systems development.



### Information about Author

*Anatoly Fedorovich Zaytsev* – Senior Lecturer at the Department of «Computational and Radioelectronical Systems». East Siberia State University of Technology and Management. Field of research: system analysis, modeling and software systems development, information technologies, programming languages, structures and data processing algorithms, system programming, operating systems, databases, knowledge bases, intelligent systems, artificial intelligence and machine learning.  
E-mail: lordsadler2010@mail.ru

Address: Russia, 670000, Ulan-Ude, Smolina St., 26.