

14. РАЗНОЕ

14.1. Онтология и графовые базы данных

©Папуша С. И.

Финансовый университет при правительстве Российской Федерации, г. Москва, Российская Федерация

e-mail: sofia.papusha@gmail.com

Аннотация. Ставится задача: построить модель базы знаний, которая состоит из объектов и их свойств, тесно связанных друг с другом, с возможностью быстрого поиска конкретных объектов и добавления новых данных без потери скорости и без трудоёмкой и сложной работы над базой. В качестве решения были предложены реляционные, нереляционные, т.е. сетевая, иерархическая и графовая, базы данных, но, в ходе анализа, сделан вывод, что такие базы данных без каких-либо улучшений не обеспечивают должной скорости работы и простоты добавления новых данных. Все вышеперечисленные модели данных имеют общие недостатки: медленный поиск объектов при больших объёмах информации, и трудоёмкий процесс добавления новых данных. В случае перестроения базы данных, приходится заново прорабатывать всю систему, что является крайне нетривиальной задачей и требует, как минимум, затрат во времени. В итоге был проанализирован метод онтологий в совмещении с такими базами данных, который максимально подходит для решения данной задачи. Свойства объектов строятся как графовая база данных, сами же объекты и свойства записываются с помощью, к примеру, RDF как набор триплетов объект – отношение – объект. Таким образом, при переходе к одному из свойств, известна также информация обо всех связях этого свойства, его дочерних и родительских узлах, объектах, которые ему принадлежат. Поиск становится значительно проще и быстрее, так как одним запросом можно сократить круг до нескольких объектов. Введение новых данных также упрощается – теперь следует только создать новый объект или сферу и внести все связи.

Ключевые слова: онтология, графовые базы данных, базы знаний, моделирование.

Для цитирования: Папуша С. И. Онтология и графовые базы данных // *Проблемы экономики и юридической практики*. 2020. Т. 16. №3. С. 268-272.

Ontology and graph databases

©S. I. Papusha

Financial University under the Government of the Russian Federation, Moscow, Russian Federation

e-mail: sofia.papusha@gmail.com

Abstract. The task is: to build a model of the knowledge base, which consists of objects and their properties that are closely related to each other, with the ability to quickly search for specific objects and add new data without losing speed and without laborious and complex work on the database. As a solution, relational, non-relational, i.e. network, hierarchical and graph databases, but, during the analysis, it was concluded that such databases without any improvements do not provide the proper speed and ease of adding new data. All of the above data models have common disadvantages: a slow search for objects with large amounts of information, and the time-consuming process of adding new data. In the case of rebuilding the database, you have to re-design the entire system, which is an extremely non-trivial task and requires at least time-consuming. As a result, the ontology method was analyzed in combination with such databases, which is most suitable for solving this problem. The properties of objects are constructed as a graph database, while the objects and properties themselves are recorded using, for example, RDF as a set of triplets object – relation – object. Thus, when going to one of the properties, information is also known about all the relationships of this property, its child and parent nodes, objects that belong to it. Searching becomes much simpler and faster, as a single query can reduce the circle to several objects. The introduction of new data is also simplified – now you only need to create a new object or sphere and make all the connections.

Keywords: ontology, graph databases, knowledge bases, modeling.

For citation: S. I. Papusha Ontology and graph databases // *Economic problems and legal practice*. 2020. Vol. 16. №3. P. 268-272.

ВВЕДЕНИЕ

Изучение баз данных и их усовершенствование является одной из важнейших задач современного мира. Чем больше становится информации, тем больше способов её классифицировать и тем более удобными и корректными они должны быть. При этом основными критериями выбора той или иной

модели данных можно считать скорость поиска, удобство навигации и простоту введения новых данных.

Система баз данных — это компьютеризированная система хранения однотипных записей. Можем назвать базу данных некой электронной картотекой, пользователю которой доступны следующие функции:

— добавление незаполненных файлов;

- редактирования существующих файлов;
- получение данных из существующих файлов;
- удаление данных из существующих файлов;
- изменение данных в существующих файлах;
- удаление файлов из базы данных. [1, стр. 43]

Современный мир строится на различных моделях данных. Каждая из них выполняет различные функции и пригодна для зачастую абсолютно противоположных целей. История развития баз данных включает в себя множество периодов, конечным из которых на данный момент можно считать построение онтологий.

Имеем следующую задачу: есть база данных некоторых объектов, каждый из которых имеет некоторые свойства. Свойства делятся на подгруппы, при этом, в некоторых случаях, имея дочерние и/или родительские свойства. При этом будем считать, что некоторый объект, который обладает свойством N, обладает и свойством M, если свойство N является дочерним свойству M. При этом имеет неограниченное количество таких свойств и таких объектов.

Таким образом, сами свойства строятся как иерархическая модель знаний – свойство A имеет дочернее свойство B, свойство B имеет дочерние свойства C и D.

При этом имеем некоторый объект, имеющий свойство D. Тогда считаем, что этот объект также обладает свойствами A и B. Соответственно, при поиске объекта с любым из этих свойств, база знаний должна выдавать в том числе и все объекты, обладающие любым из родительских свойств данного свойства.

Основные требования данной системы – быстрый поиск объектов на основе их свойств. Проблема существующих баз знаний будет заключаться в том, что создание новых объектов будет требовать большого объема работы – придётся создавать новый объект в самом низу каких-либо узлов, что, при большом количестве объектов и при необходимости практически ежедневного их создания, естественно, будет тратить много усилий. Также возникает проблема при поиске – опять же, при большом количестве объектов, поиск каждого из них будет, фактически, переходом от свойства к свойству всё глубже и глубже. Это тратит время и ресурсы.

Приведём пример: имеем игру мафия, в которой есть четыре вида роли: мирный, мафия, дон и шериф. Таким образом, делим все роли на два типа: красный город и чёрный город. К красному относятся шериф и мирный, к чёрному – дон и мафия. Имеем свойства: может проверять, может стрелять, может голосовать, знает чёрных игроков.

Таким образом, если объект обладает свойством «знает чёрных игроков», он не может иметь роль «красный», а если обладает свойством «может проверять», то может быть только шерифом или доном. Поиск роли производится переходом по дереву всё ниже и ниже, причём объект обладает всеми свойствами, которые являются родительскими узлами по отношению к нему.

Естественно, созданная база данных должна обладать также всеми нижеописанными свойствами:

- компактность. Отпадает необходимость вести какие-либо бумажные картотеки данных;
- быстродействие. Компьютер работает значительно быстрее человека, а значит выбор и обновление данных будет тратить гораздо меньше времени;
- низкие трудозатраты. Нет человека, которому следует вручную актуализировать картотеку или искать что-либо в ней;
- актуальность. В любой момент актуальная информация находится в свободном доступе;

- защита. Данные лучше защищены от случайной потери и несанкционированного доступа. [1, стр. 58]

РАЗЛИЧНЫЕ БАЗЫ ДАННЫХ В УСЛОВИЯХ ПОСТАВЛЕННОЙ ЗАДАЧИ

Иерархическая модель данных – один из первых этапов построения баз знаний. Ключевая особенность – родитель имеет конкретный дочерний узел, дочерний узел имеет только одного родителя. В дальнейшем, была разработана сетевая модель данных. Отличие от иерархической в том, что дочерний узел может иметь несколько родителей. [7]

Следующим этапом было разделение на реляционные модели данных и нереляционные. Первые состоят, фактически, из таблиц. Основное преимущество таких баз данных состоит в том, что в них просто что-либо найти – достаточно обратиться к какой-либо таблице. Такие базы данных управляются с помощью языка SQL – он является инструментом, предназначенным для организации, управления, выборки и обработки информации из базы данных. SQL – это язык программирования, который применяется для организации взаимодействия пользователя с базой данных. [8] SQL работает с реляционными базами данных, которые представляют собой основной способ организации данных в широком диапазоне приложений. В вычислительной системе имеется база данных, в которой хранится важная информация. Если говорить о промышленности, то в базе данных может содержаться информация о капитале компании, принадлежащей ей недвижимости, выпуске продукции. Если же это персональный компьютер, то речь может идти о телефонных справочниках, заметках, списке покупок и т.д. Компьютерная программа, которая управляет базой данных, называется системой управления базой данных, или СУБД. [2, стр. 30]

Нереляционные же базы знаний состоят из разрозненных данных, могут содержать фото, видео, ссылки, без какой-либо явной структуры. Мы по-прежнему рассматриваем реляционные базы данных как один из вариантов хранения данных. [13] Эту точку зрения часто называют постоянством полиглота – использование разных хранилищ данных в разных обстоятельствах. Вместо того, чтобы просто выбирать реляционную базу данных, потому что все так делают, нам нужно понять природу данных, которые мы храним, и то, как мы хотим манипулировать ими. В результате большинство организаций будут использовать различные технологии хранения данных для разных обстоятельств. [3, стр. 25]

Нереляционные базы знаний имеют свои плюсы – информацию можно в любой момент добавить и изъять, не меняя и не редактируя структуру самой базы данных. Если рассмотреть данные модели данных на примере, то реляционная база данных – это таблица с названиями кустарников, информации об их цене, способах ухода, фотографиями. В такой базе данных просто найти какой-либо кустарник, но становится проблемой внести новую информацию, отличную от существующей – например, место их произрастания. Нереляционная база данных в таком случае – это файл, в котором в разрозненном порядке хранится вся данная информация, включая фотографии кустарников, ссылки на место их покупки в Интернете, статьи и прочее. Найти что-либо в таком файле тяжело, зато добавить в него что-либо не составит труда. [10]

Сетевая и иерархическая модели данных, по сути, являются нереляционными моделями данных. В чистом виде обе модели данных не подходят для поставленной задачи: иерархиче-

ская модель данных ограничивает возможности – у одного дочернего узла не может быть несколько родителей. Если такая ситуация возникает, приходится перестраивать базу данных, что может повлиять на время поиска данного объекта или логику построения базы знаний. [11] Например, если требуется, чтобы роль «шериф» имела одновременно два родителя, которые по каким-то причинам находятся на одном уровне: «мирный город» и «может проверять», то придётся перестраивать базу знаний. Этого недостатка нет в сетевой модели данных и такая модель данных подходит для визуального представления данных. Тем не менее, есть существенный недостаток: для поиска любого объекта следует перейти по всем узлам к самому низу. При большом количестве узлов, при ошибках поиска со стороны пользователя, при больших объёмах информации это приведёт к большим затратам во времени. Также при добавлении новых свойств и ролей в систему, в большинстве случаев, требуется изменение всей модели данных. Предположим, вводится роль «доктор». Для однозначного её определения следует ввести свойство «может лечить», которое встраивается в граф. Так будет происходить при добавлении абсолютно любого нового объекта.

Не стоит говорить, почему другие нереляционные модели данных не подходят: невозможно производить быстрый поиск объектов. [14] Это общий недостаток таких моделей данных, а в условиях, когда одна из задач – быстрый поиск, в чистом виде эти модели не могут использоваться. Как не странно, именно такая модель данных может стать решением данной задачи, но об этом позднее.

Реляционные базы данных решают одну из задач: добавление новых объектов и свойств будет являться довольно простой задачей. Любую базу данных можно представить, как таблицы свойств, объектов, связей при условии присвоения каждому объекту уникального кода. Как бы ни была построена система ранее, есть возможность добавить новую таблицу, или редактировать уже существующую.

Остаётся только проблема быстрого поиска. В такой модели данных для поиска каждого объекта следует переходить от таблицы к таблице по уникальным ключам свойств вплоть до объекта поиска, или задавать определённые условия через SQL – язык, который позволяет отправлять запросы к реляционным базам данных. [15] При большом объёме информации или неправильной архитектуре базы данных, такой поиск будет занимать много времени.

Таким образом, различные модели данных могут быть использованы для всевозможных целей. Тем не менее, вышеуказанные модели не могут быть использованы для описания таких систем как Всемирная паутина. Для такого рода данных используются графы, которые относятся к нереляционным базам знаний.

Графовые базы данных специализируются на сборе информации, но в гораздо большем масштабе, чем может прочитать удобочитаемая диаграмма. Это идеально подходит для сбора любых данных, состоящих из сложных отношений, таких как социальные сети, предпочтения продукта или правила отбора. Основная модель данных графовой базы данных очень проста: узлы, соединённые ребрами (также называется дуги). Помимо этой существенной характеристики, в моделях данных есть много вариаций, в частности, какие механизмы вы должны хранить в своих узлах и ребрах. [3, стр 37]

Одним из примеров таких графов является Вики-граф – база данных, основанная на информации из Википедии. Данный граф был выбран для исследования, так как масштабы Вики-

педии сравнительно малы, но при этом вся информация в ней структурирована одинаково, а также каждая статья имеет ссылки на другие статьи. Таким образом, информацию можно представить в виде графа, узлами которого являются статьи, а рёбрами – множественные связи между ними. Есть и другие графовые базы данных, но данный пример наиболее близок к поставленной задаче – большие объёмы информации, объекты связаны друг с другом, требуется осуществлять поиск объектов. [5]

ВИКИ-ГРАФ

Одним из решений данной задачи может являться Вики-граф. Суть данного графа в следующем: если обозначить веб-граф как $G = (V, E)$, где множество V вершин графа G состоит из всех веб-страниц сети Интернет, то можно выделить из множества V подмножество $V' \subset V$, которое является статьями Википедии, а из множества E ребер графа G – подмножество $E' \subset E$, в котором конечные и начальные вершины ребер принадлежат множеству V' . Таким образом, графом Википедии, или Вики-графом, мы будем называть граф $G' = (V', E')$, являющийся подграфом веб-графа G . При этом рёбра могут быть как однонаправленными, то есть ссылка на статью A находится в статье B , но не наоборот, так и двунаправленными, то есть ссылка на статью A находится в статье B и наоборот.

Важным аспектом при этом является то, что статьи Википедии создаются по единому шаблону, строгим правилам, что во многом облегчает анализ статей и исследование Вики-графа.

Основным преимуществом использования графа является то, что с помощью такой модели данных появляется возможность строить модели предметной области абсолютно любой сложности. При этом под узлом понимается вершина графа, а под связью – ребро графа. У определённого узла также могут быть некоторые свойства. Таким образом, граф ничуть не уступает реляционной базе данных, позволяя при этом хранить связи между узлами, не затрачивая лишние ресурсы на их хранение или обработку при поиске каких-либо объектов. При сравнении с реляционными базами знаний оказывается, что в данных базах знаний приходится создавать и хранить так называемые уникальные ключи для обозначения связи двух таблиц. При этом каждый раз, когда пользователю данной базы данных требуется извлечь связи, приходится проводить операции, которые, при большом объёме данных, могут затрачивать как большое количество времени, так и большое количество ресурсов. При работе с графовыми базами данных, данные операции имеют фиксированную вычислительную сложность и не зависят от количества узлов в графе. [5]

Также важным преимуществом является то, что в графовых базах данных пользователь способен в любой момент времени внести изменения в структуру графа и свойства его узлов. Таким образом, при изменении предметной области, имеется возможность сравнительно быстро поменять и саму базу данных о ней.

Использование графовых баз данных особенно необходимо при построении различных моделей системы. Тем не менее, использование таких баз данных в чистом виде для решения поставленной задачи не подходит по некоторому ряду причин:

- Ссылочный способ всё ещё является достаточно трудоёмким методом связи объектов и свойств;
- Поиск объектов осложняется потенциальным наличием «петель», то есть ссылок объектов друг на друга.

ОНТОЛОГИЯ

В современном мире любое предприятие, отдел, государство или иная организованная система могут быть смоделированы с использованием онтологического моделирования. Онтология — это всеобъемлющая и подробная формализация некоторой области знаний с помощью концептуальной схемы. Цель онтологии — представить некоторую систему таким образом, чтобы в результате её формализации, была возможность понять, какие изменения приводят к определённым результатам, каким способом можно улучшить работу системы, как система работает.

Моделирование любой системы является нетривиальной задачей как минимум потому, что каждый человек воспринимает одну и ту же информацию по-разному. При использовании простых понятий, проблем может не возникать, тем не менее, в момент, когда затронуты какие-либо сложные парадигмы, задачей аналитика становится рассмотреть первичные концепты этих парадигм, чтобы сконструировать модель, которая будет восприниматься всеми одинаково.

Важнейшей частью моделирования является представление о пространстве и времени в модели. Различные пространственно-временные концепты позволяют точно описать, где и как воспринимается и работает определённая часть системы. Пространственные концепты говорят о том, как мы видим данную часть — считаем ли мы её отдельным объектом, воспринимается ли она индивидуально, важно ли как она работает, или же следует воспринимать её в контексте модели «чёрного ящика». Временные концепты позволяют описать, когда и как происходят некоторые события — отдельно друг от друга, или одновременно, считаем ли мы некоторые процессы одним целым, следует ли разделять определённые события. При моделировании следует также соотносить различные пространственно-временные объёмы друг с другом: при моделировании работы предприятия следует помнить, что изготовление одной единицы продукции занимает, к примеру, несколько минут в рамках конкретного станка, а изготовление партии продукции будет занимать несколько часов в рамках целого цеха.

Также важной проблемой построения любой онтологии является классификация объектов. В зависимости от того, считается ли сущность объекта неизменной, по каким признакам проводится классификация, следует ли проводить классификацию по пространственному или временному признаку, возникает огромное количество вариантов классификации объектов даже в рамках одной системы. Выбор конкретной классификации зависит от того, какую модель требуется построить на основе данной системы. [6]

Онтология относится скорее к моделированию систем, чем к базам данных. Тем не менее, в итоге построения каких-либо онтологий, мы получаем модель системы, которую следует перевести в цифровой вид для упрощения работы с расчётами и самой моделью. Именно для этой цели используются графовые базы данных. Рассмотрим основные принципы работы таких баз данных на примере стандартов моделирования RDF/OWL.

RDF (Resource Description Framework) — это модель для представления данных, которая представляет факты и свойства некоторых объектов в виде, пригодном для машинной обработки.

OWL (Web Ontology Language) — язык для описания онтологий.

Фактически, данные стандарты формализуют описание взаимосвязанных сущностей. Данные записаны в виде триплетов: объект — отношение — объект. Данная запись позволяет указать все возможные связи между объектами системы, а также их свойства в рамках данной модели. Такая запись позволяет получать быстрый доступ к любым объектам, их свойствам, связям и объектам, с которыми они связаны.

Для наглядности, проведём моделирование определённой семьи. Создаём объект семья, который включает три объекта типа «человек»: мама, папа и сын. При этом следует создать тип «человек», который имеет некоторые свойства: есть ребёнок, есть родитель, есть партнёр, пол и возраст.

Далее переходим к созданию каждого члена семьи. При создании следует указывать не только параметры, необходимые для создания объекта типа «человек», но и все связи данного объекта с другими объектами модели.

Объект: Папа, тип: «человек»;

Свойство: пол: «мужской»;

Свойство: возраст «40»;

Свойство: имеет сына «Сын»;

Свойство: имеет жену «Жена»;

Перед созданием каждого объекта следует описать, что означает каждое свойство:

Свойство: пол, диапазон: «мужской», «женский».

При этом, предположим, при создании свойства «имеет сына», первоначально создаётся свойство «имеет ребёнка», а свойство же «имеет сына» — это конкретный случай данного свойства при условии, что пол ребёнка — мужской. Та же ситуация происходит при создании свойства «имеет жену» — это частный случай свойства «имеет партнёра».

Подобное моделирование позволяет понять, что, если объект обладает свойством «имеет сына», то он также будет обладать свойством «имеет ребёнка», «является родителем», а ребёнок его имеет свойство «пол» со значением «мужской». Таким образом, при грамотном создании свойств, взглянув на одно свойство одного объекта, мы имеем возможность не только получить дополнительную информацию об этом объекте, но и об объекте, с ним связанным.

Если применить данную модель к примеру, про игру «Мафия», можем создать следующие объекты и свойства:

1. Свойство «красный город». Связано с ролью «мирный», ролью «шериф», свойством «может голосовать».
2. Свойство «чёрный город». Связано с ролью «мафия», ролью «дон», свойствами «знает чёрных игроков», «может стрелять», «может голосовать».
3. Свойство «может проверять». Связано с ролями «дон» и «шериф», свойством «может голосовать».
4. Свойство «может стрелять». Связано с ролями «дон», «мафия», свойством «может голосовать».
5. Свойство «знает чёрных игроков». Связано с ролями «дон» и «мафия», со свойствами «может стрелять», «может голосовать».
6. Свойство «может голосовать». Связано с ролями «дон», «шериф», «мирный», «мафия»
7. Роль «Дон». Имеет свойства 2, 3, 4, 5, 6.
8. Роль «Мирный». Имеет свойства 1, 6.
9. Роль «Шериф». Имеет свойства 1, 3, 6.
10. Роль «Мафия». Имеет свойства 2, 4, 5, 6.

Таким образом, если мы узнаем, что какой-либо объект имеет свойство «может проверять», автоматически становится известно, что этот объект может голосовать и является либо доном, либо шерифом. Так, узнав только об одном свойстве

объекта, ограничивается круг поиска и становятся известными часть свойств объекта.

Задача онтологического моделирования – построить такую модель связи между объектами, чтобы доступ от одного объекта к другому и связям между ними полностью описывал данный объект и всю систему, на основе которой строится данная модель. Графовые базы данных в данном случае – это лучший способ описания таких систем, которые позволяет достичь важного результата: построенная модель данных читаема для машины. С помощью вышеуказанных стандартов, мы не только передаём модель на электронный носитель, но и позволяем компьютеру понять, как работает указанная нами система. Это и есть основная цель – дать возможность машине не просто считывать данные, но и обрабатывать их, максимально «понимая» их смысл.

ВЫВОД

Таким образом, подобная систематизация данных, при использовании онтологий, позволяет удовлетворить все требования: любая связь прописывается через свойства и объекты

Список литературы:

1. Введение в системы баз данных//К. Дж. Дейт. 2005. 1316 с. ISBN 5-8459-0788-8
2. SQL. Полное руководство//Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель. 2015. 960 с. ISBN 978-5-8459-1654-9
3. NoSQL Distilled//Martin Fowler. 234 с. 2013.
4. Парадигмы цифровой экономики//под редакцией М.А. Эскиндарова, В.И. Соловьева. Финансовый университет при Правительстве Российской Федерации. 2019. 326 с. ISBN 978-5-89353-550-1
5. Онтологическое моделирование предприятий: методы и технологии//Издательство Уральского университета. 2019. 237 с. ISBN 978-5-7996-2580-1
6. Основные технологии баз данных//Б.А. Новиков, Е.А. Горшкова, Н.Г. Графеева.
7. Encyclopedia of Database Systems// Ling Liu, M. Tamer Ozsu. Springer Science, 2009. 3818 p. ISBN 9780387355443.
8. Trees and Hierarchies in SQL for Smarties// Joe Celko. 2nd Edition. Morgan Kaufmann. 2012. 296 p. ISBN 0123877334.
9. Graph Databases// Robinson, I. and Webber, J. and Eifrem, E. O'Reilly Media, 2013. 178 p. ISBN 978-1449356262.
10. Professional NoSQL//Shashank Tiwari. John Wiley & Sons, 2011. ISBN 978-1-118-16780-9.
11. Introduction to Database Management System// Satinder Bal Gupta, Aditya Mittal. Laxmi Publications, 2009. 288 p. ISBN 813180724X, 9788131807248.
12. Хранилища данных. От концепции до внедрения//С. Архипенков, Д. Голубев, О. Максименко. М.: Диалог-Мифи. 2002. 528 p.
13. Базы данных//С.В. Глушаков, Д.В. Ломотко. М.: Харьков. Фолио. 2000. 504 p.
14. Базы данных: разработка и управление//Хансен Генри; , Джеймс. М.: Бином. 2015. 704 p.
15. Базы данных и информационные системы//В.Н. Редько, И.А. Басараб. М.: Знание. 2017. 341 p.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Папуша Софья Ильинична, Департамент анализа данных и машинного обучения, Финансовый университет при правительстве Российской Федерации, г. Москва, Российская Федерация, <https://orcid.org/0000-0002-5871-1230>, e-mail: sofia.papusha@gmail.com

постепенно – каждое свойство ссылается на другое, имея лишь одно, есть возможность при этом обладать и многими другими; создание новой предметной области или объекта не вызывает трудностей, так как следует прописать только самую «глубокую» часть конкретного свойства.

ЗАКЛЮЧЕНИЕ

В результате проведённого анализа различных баз знаний и методов их построения, можно прийти к выводу, что метод графовых баз данных при совмещении с онтологией и стандартами моделирования RDF/OWL может обеспечить решение всех поставленных задач. Сложности могут возникнуть на стадии построения онтологии, так как данная задача требует анализа всего комплекса предприятия от и до, и на стадии описания данной онтологии с помощью, например, RDFS следует изучить данный стандарт моделирования. Тем не менее, в данный период времени, это лучшее решение данной задачи.

Статья проверена программой «Антиплагиат». Оригинальность 83,71%.

Reference list:

1. Introduction to database systems // K. J. Date. 2005.1316 p. ISBN 5-8459-0788-8
2. SQL. The Complete Guide // James R. Groff, Paul N. Weinberg, Andrew J. Oppel. 2015.960 s. ISBN 978-5-8459-1654-9
3. NoSQL Distilled // Martin Fowler. 234 p. 2013.
4. Paradigms of the digital economy // edited by MA Eskindarova, V.I. Solovoyov. Financial University under the Government of the Russian Federation. 2019.326 s. ISBN 978-5-89353-550-1
5. Ontological modeling of enterprises: methods and technologies // Publishing House of the Ural University. 2019.23 s ISBN 978-5-7996-2580-1
6. Basic database technologies // B.A. Novikov, E.A. Gorshkova, N.G. Grafeeva.
7. Encyclopedia of Database Systems // Ling Liu, M. Tamer Ozsu. Springer Science, 2009.3818 p. ISBN 9780387355443.
8. Trees and Hierarchies in SQL for Smarties // Joe Celko. 2nd Edition. Morgan Kaufmann. 2012.296 p. ISBN 0123877334.
9. Graph Databases // Robinson, I. and Webber, J. and Eifrem, E. O'Reilly Media, 2013.178 p. ISBN 978-1449356262.
10. Professional NoSQL // Shashank Tiwari. John Wiley & Sons, 2011. ISBN 978-1-118-16780-9.
11. Introduction to Database Management System // Satinder Bal Gupta, Aditya Mittal. Laxmi Publications, 2009.288 p. ISBN 813180724X, 9788131807248.
12. Data Warehouses. From concept to implementation // C. Arkhipenkov, D. Golubev, O. Maksimenko. M.: Dialog-Mifi. 2002.528 p.
13. Databases // S.V. Glushakov, D.V. Lomotko. M.: Kharkov. Folio. 2000. 504 p.
14. Databases: development and management // Hansen Henry; James. M.: Binom. 2015.704 p.
15. Databases and information systems // V.N. Redko, I.A. Basarab. M.: Knowledge. 2017.341 p.

INFORMATION ABOUT THE AUTHORS

Sofya I. Papusha, Department of Data Analysis and Machine Learning, Financial University under the Government of the Russian Federation, Moscow, Russian Federation, <https://orcid.org/0000-0002-5871-1230>, e-mail: sofia.papusha@gmail.com