

# Графическая нотация моделирования документных баз данных

**Цели и задачи.** Графические модели зарекомендовали себя как надежный, понятный и удобный инструмент создания эскизных моделей баз данных. Большинство существующих нотаций разработаны для реляционной модели данных, доминирующей последние тридцать лет моделью данных. Однако, развитие информационных технологий привело к росту популярности нереляционных моделей данных, в первую очередь документной модели. Одной из проблем её применения на практике является отсутствие подходящих инструментов, позволяющих выполнять графическое моделирование базы данных, с учетом особенностей документной модели, на этапе логического проектирования. Разработка соответствующих инструментов является важной и актуальной задачей, так как их применение в практических исследованиях позволяет выявить, классифицировать и разобрать типовые ошибки моделирования, что позволяет проектировщику снизить риск их появления в дальнейшем.

Цель данной статьи заключается в разработке графической нотации, с одной стороны обеспечивающей удобство работы для проектировщика, а с другой стороны учитывала особенности построения и функционирования *noSQL* документной модели хранения данных.

**Материалы и методы.** Материалами для исследования послужили многочисленные публикации, посвященные разработке графических нотаций в задачах и их применению проектирования баз данных для различных информационных систем. Отобранные материалы были проанализированы и выявлены основные графические нотации, применяемые для описания реляционной модели данных. Из них было отобрано три нотации, набор графических

стереотипов, которых наиболее отличался друг от друга, анализ которых позволил выделить основные паттерны изображения составляющих реляционной модели.

Полученные паттерны были применены к основным элементам документной базы данных, которые были получены путём анализа документации популярной СУБД *MongoDB*.

**Результаты.** Результатом проведенного исследования стало создание нового инструмента для моделирования документных баз данных на логическом уровне, который, состоит из набора графических стереотипов и правил их применения. Разработка с одной стороны хорошо знакома практикам, ранее работавшим с реляционными моделями данных, так как при его разработке учитывался многолетний опыт применения графических моделей в области проектирования реляционных баз данных, а с другой стороны отражает особенности структуры документной модели.

**Заключение.** Практическое применение разработанной модели показало удобство ее использования как в процессе проектирования документных баз данных, так и в процессе обучения студентов в рамках данной предметной области. Применение графических моделей, построенных в предложенной графической нотации, позволит исследователям создавать и иллюстрировать типовые паттерны документных баз данных, что, несомненно, положительно скажется на динамике развития перспективных технологий хранения данных.

**Ключевые слова:** проектирование баз данных, логическая модель, документная модель данных, реляционная модель, графическая модель.

Mikhail V. Smirnov, Ruslan S. Tolmasov

<sup>1</sup> MIREA – Russian Technological University, Moscow, Russia

## Graphical Notation for Document Database Modeling

**Goals and objectives.** Graphical models have proven to be a reliable, clear and convenient tool for creating sketch models of databases. Most of the existing notations are designed for the relational data model, the dominant data model for the last thirty years. However, the development of information technologies has led to an increase in the popularity of non-relational data models, primarily the document model. One of the problems of its application in practice is the lack of suitable tools that allow performing graphical modeling of the database, taking into account the features of the document model, at the stage of logical design. The development of appropriate tools is an important and actual task, since their application in practical research makes it possible to identify, classify and analyze typical modeling errors that allow the designer to reduce the risk of their occurrence in the future. The purpose of this article is to develop a graphical notation that, on the one hand, providing convenience for the designer, and on the other hand, taking into account the peculiarities of creating and functioning of the *noSQL* document storage model.

**Materials and methods.** The materials for the study were numerous publications devoted to the development of graphical notations in problems and their application to database design for various information systems. The selected materials were analyzed and the main graphical notations used to describe the relational data model were identified. Three notations were selected from them, a set of graphic stereotypes, which were most different from each other, the

analysis of which allowed us to identify the main image patterns of the components of the relational model.

The resulting patterns were applied to the main elements of the document database, which were obtained by analyzing the documentation of the popular *MongoDB* DBMS.

**Results.** The result of the research was the creation of a new tool for modeling document databases at the logical level, which consists of a set of graphic stereotypes and rules for their application. On the one hand, the development is well known to practitioners who have previously worked with relational data models, since its development took into account many years of experience in using graphical models in the field of relational database design, and on the other hand, it reflects the features of the structure of the document model.

**Conclusion.** The practical application of the developed model has shown the convenience of its use both in the process of designing document databases and in the process of teaching students within this subject area. The use of graphical models constructed in the proposed graphical notation will allow researchers to create and illustrate typical patterns of document databases, which will undoubtedly have a positive impact on the dynamics of the development of promising data storage technologies.

**Keywords:** database design, logical model, document data model, relational model, graphical model.

## Введение

Графические модели изображения структур хранения данных используются с 60-х годов прошлого века, и зарекомендовали себя как надежный, понятный и удобный инструмент создания эскизных моделей баз данных. Графические модели последовательно создавались, испытывались и применялись для иерархических, сетевых, реляционных баз данных.

Значимость графических моделей данных, в первую очередь семантических, отмечают как зарубежные, так и отечественные исследователи в области теории баз данных. Кузнецов С.Д., внесший значительный вклад в развитие теории баз данных на территории России, в своём учебнике так описывал необходимость построения моделей: «... построение мощной и наглядной концептуальной схемы БД позволяет более полно определить специфику моделируемой предметной области и избежать возможных ошибок на стадии проектирования схемы реляционной БД» [1]. С этой точкой зрения согласны Ф.А. Попов и А.В. Максимов [2], которые провели исследование подходов к проектированию баз данных, дополняя её утверждением, что ошибки допущенные при концептуальном моделировании наиболее трудно выявляемые и устраняемые в дальнейшем. Среди зарубежных исследователей решающий вклад в развитие доминирующей в последние 30 лет реляционной модели хранения данных внес американский математик Э.Ф. Кодд, который создал реляционную модель данных и сформировал основы реляционной алгебры [3]. Его принципы легли в основу всех используемых на проектной стадии графических моделей баз данных. Также следует отметить Питера Чена, предложившего графическую

модель «сущность-связь», успешно применяемой на этапах концептуального и логического проектирования [4].

Однако семантические (концептуальные) модели — это модели с высоким уровнем абстракции, они позволяют глубже понять предметную область, назначение конкретных сущностей и необходимость того или иного ограничения, но совершенно не учитывают модель данных, которая будет использована в дальнейшем. Эти особенности отражаются уже на уровне логического проектирования. Для реляционных моделей создано большое количество инструментов для проектирования логических моделей, однако в настоящее время наблюдается повышение интереса к конкурирующим с реляционными моделям данных.

С развитием специализированных информационных систем и технологий обработки данных возникли предметные области, где применение традиционной реляционной модели оказывалось существенно затруднено или даже невозможно. Так, при работе с Большими Данными (*Big Data*) использование статичной и строго определённой структуры хранения данных, подразумеваемой реляционной парадигмой, становилось невозможным, поскольку в большинстве случаев Большие Данные являются плохо структурированными.

Это привело к формированию концепции *noSQL* баз данных, которая в настоящее время активно набирает популярность, что, в свою очередь, подталкивает к изучению новых принципов *noSQL* моделей данных как профессионалов отрасли, так и людей, проходящих обучение специальностям, связанным с технологиями хранения данных. В обзорной статье *A B M Moniruzzaman* и *Syed Akhter Hossain* [5] опираясь на мировой опыт, авторы показывают в каких условиях

выбор *noSQL* модели оптимален, а также приводят общую характеристику и классификацию существующих на данный момент нереляционных моделей: хранилище «ключ-значение», документная модель, графовые модели и др.

Наиболее часто обсуждаемой в научном сообществе и применяемой в практике *noSQL* моделью хранения данных сейчас является документная модель хранения данных, являющаяся второй по популярности после реляционной модели по состоянию на 2021 год [6]. Документная модель широко применяется в сфере WEB-программирования, аналитике и других сферах, где хранятся и обрабатываются плохо структурированные данные [7–10].

Одной из проблем применения документной модели данных на практике является отсутствие на настоящей момент времени графической нотации, которая позволяла бы исследователю или проектировщику, с учетом особенностей выбранной модели хранения, осуществлять моделирование базы данных на этапе логического проектирования.

Решение этой проблемы является актуальной и важной задачей. Опыт практического применения графических моделей при проектировании реляционных баз данных на основании известных нотаций Питера Чена, Мартина, IDEF1x и других показывает, что они позволяют сформировать и повторно использовать типовые паттерны реляционных баз данных. При этом решается проблема демонстрации связей и элементов по-настоящему комплексных структур многопользовательских баз данных. Применение графических моделей в практических исследованиях позволяет выявить, классифицировать и разобрать типовые ошибки моделирования, что позволяет проектировщику

снизить риск их появления в дальнейшем.

Цель данной статьи заключается в разработке графической нотации, с одной стороны обеспечивающей удобство работы для проектировщика, а с другой стороны учитывала особенности построения и функционирования *noSQL* документной модели хранения данных.

Для этого необходимо последовательное решение следующих задач:

1. Проанализировать существующие графические нотации, применяемые для проектирования моделей баз данных на логическом уровне.

2. Выявить ключевые особенности и основные элементы документной модели данных, которые необходимо отразить графической нотацией.

3. Изучить существующие решения, которые могут быть применены для создания логической модели документной базы данных.

4. Разработать набор графических элементов и правил их применения, позволяющих корректно передать семантику проектируемой базы данных и логические взаимосвязи между её составляющими.

В результате был разработан новый инструмент для построения логических моделей документных баз данных, состоящий из набора графических элементов, для обозначения основных объектов документной модели (коллекции, документы и др.) и набор правил их применения, который позволит исследователям создавать и иллюстрировать типовые паттерны документных баз данных, что, несомненно, положительно скажется на динамике развития перспективных технологий хранения данных.

Апробация результатов была проведена на задаче перепроектирования существующей реляционной базы данных в документную для WEB-приложения MSUniversity [11], в

ходе которой была получена наглядная и простая для понимания логическая модель документной базы данных.

### Анализ графических нотаций для описания реляционных моделей хранения данных

При создании инструментария и правил изображения графической нотации документной модели баз данных, за основу был взят многолетний опыт использования графических нотаций, применяемых реляционных структур хранения данных [4, 12–18].

Для графического описания реляционной модели данных существуют несколько наиболее распространённых нота-

ций: нотация Питера Чена [4], IDEF1x [15], нотация Crow's foot [16], диаграмма классов UML [1, стр. 160–174]. Каждая из них содержит элементы необходимые для изображения атрибутов, сущностей и связей между ними, а также правила, регламентирующие их применения. Благодаря этим элементам становится возможным проведение моделирования реляционных баз данных практически для любой предметной области. Детальное сравнение существующих нотаций было дано в ряде работ зарубежных учёных, в частности в работе Най Д. С. [19], подробно описавшего сходство и различия применяемых графических нотаций.

Таблица

Графические стереотипы, применяемые в различных нотациях

Table

Graphic stereotypes used in various notations

	Нотация Питера Чена	IDEF1x	UML
Сущность			
Атрибут			
Первичный ключ			Часто не указывается. Может быть добавлено «(PK)» рядом с именем ключевого атрибута
Внешний ключ	Не указывается	Добавляется «(FK)» рядом с атрибутом внешнего ключа	Часто не указывается. Может быть добавлено «(FK)» рядом с именем ключевого атрибута
Связь			

Для анализа были выбраны три нотации: нотация Питера Чена и IDEF1x и диаграмма классов UML. Такой выбор обусловлен существенными различиями используемых графических обозначений для различных элементов реляционной модели базы данных. В ходе анализа были выделены основные элементы реляционной модели данных и для каждой из них были рассмотрены соответствующие графические стереотипы, применяемые в выбранных нотациях. Результаты сведены в Таблицу, где в строках указаны основные элементы модели, в столбцах — рассматриваемые нотации, в ячейках указаны применяемые для их обозначения графические стереотипы.

Анализ существующих нотаций для описания реляционных моделей баз данных позволил выделить основные графические стереотипы для изображения основных элементов модели. Использование этих стереотипов при разработке нотации для документной базы данных позволит создать инструмент, который с одной стороны хорошо знаком исследователям и проектировщикам реляционных моделей данных, но с другой — выполнять моделирование модели данных, имеющих существенно отличающейся от реляционной.

### Особенности документной модели данных

Рассмотрим документную модель данных подробнее с целью выявить ключевые особенности и элементы, которые необходимо отразить в разрабатываемой модели.

На рис. 1 приведена типовая структура документной базы данных.

Основные элементы модели рассматриваются на примере популярной документной СУБД MongoDB. Согласно рейтингу, опубликованному на странице db-engines.com,

MongoDB стабильно занимает 5 место по популярности среди разработчиков, что повлияло на выбор данной СУБД

В рассматриваемой модели данные хранятся в документах. Согласно официальной документации [20], документ — основная единица хранения данных. Каждый документ содержит набор атрибутов, которые характеризуют объект, описываемый документом. Каждый атрибут представляет собой пару «ключ: значение», где «ключ» представляет собой уникальное в пределах рассматриваемого набора имя, удовлетворяющее требованиям к именам, установленными СУБД. В «значение» указывается любое допустимое значение соответствующего типа. То есть, при описании структуры документа, «значение» может быть представлено типом данных, которые планируется в нем разместить.

Документы могут и должны быть объединены в коллекции. При этом необязательно, но рекомендуется объединять в коллекции документы максимально похожие наборы атрибутов. Для идентификации документов, в MongoDB создаётся специальный атрибут «*id*», который содержит уникальное в рамках БД значение и позво-

ляет однозначно определить документ и его принадлежность к той или иной коллекции. Стоит отметить, что сами коллекции не имеют собственного набора атрибутов, но они содержат все необходимые методы для работы с документами, помещенными в них.

Благодаря наличию «*\_id*», документы могут ссылаться друг на друга, как в пределах одной коллекции, так и на документы, расположенные в других коллекциях той же базы данных. Но сами связи и механизм их работы существенно отличается от связей в реляционной модели. В MongoDB связи можно разделить на два типа: связь «включение», когда документ включается в состав родительского документа, и связь «ссылка на документ» которая с помощью дополнительного атрибута может служить указателем на любой документ, содержащийся в базе, что позволяет реализовать псевдореляционную модель.

Также, существенным отличием документной модели данных от реляционной является то, что схема хранения данных в документной модели не может быть определена и зафиксирована на стадии проектирования, и может быть изменена динамически, в зависимости

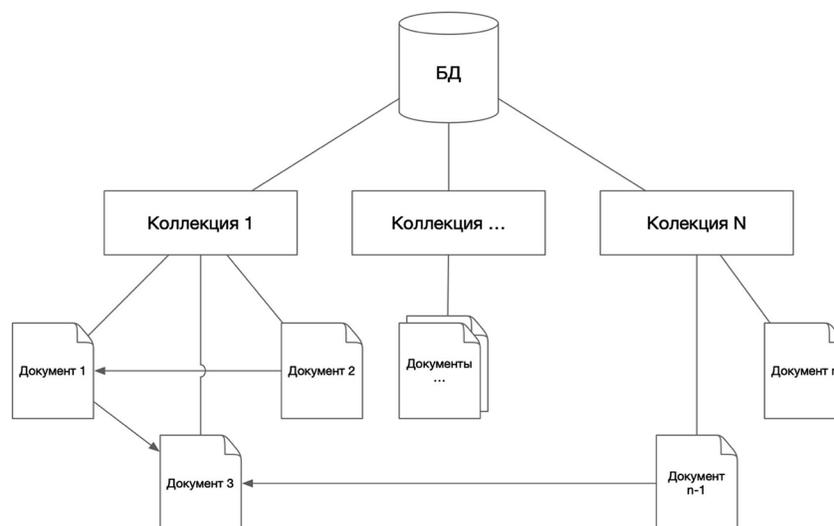


Рис. 1. Структура документной БД

Fig. 1. The structure of the document database

от потребностей пользователя в настоящий момент без существенных затруднений. Это отличие, не оказывает влияния на графическое изображение модели данных, но её необходимо учитывать при практическом применении модели.

Таким образом, были выявлены следующие ключевые особенности документной модели, которые необходимо учесть при разработке графической нотации:

- основной единицей моделирования является документ, который описывает объект предметной области или отдельный его аспект;
- документ должен быть включён в состав не более чем одной коллекции;
- коллекция представляет собой способ логического объединения документов;
- понятие «связь» и механизм её работы в документной модели существенно отличается от реляционной;
- выделяют два типа связей в документной модели «включение» и «ссылка на документ».

### Существующие решения для визуализации документных моделей

Для организации доступа к данным и работы с ними применяется специально адаптированная версия *JSON*, которая получила название *BSON*. Современные источники научно-исследовательской информации не дают ссылки на статьи и другие материалы, которые бы описывали визуализацию структуры документных моделей, обслуживаемых *JSON/BSON-запросами*. Тем не менее, для визуализации структуры *JSON-документа*, существует ряд решений, одно из которых *JSONDesigner* [21], доступное в виде *WEB-приложения* и на мобильных устройствах под управлением *Android* и *iOS*.

Для оценки возможности применения *JSONDesigner* для

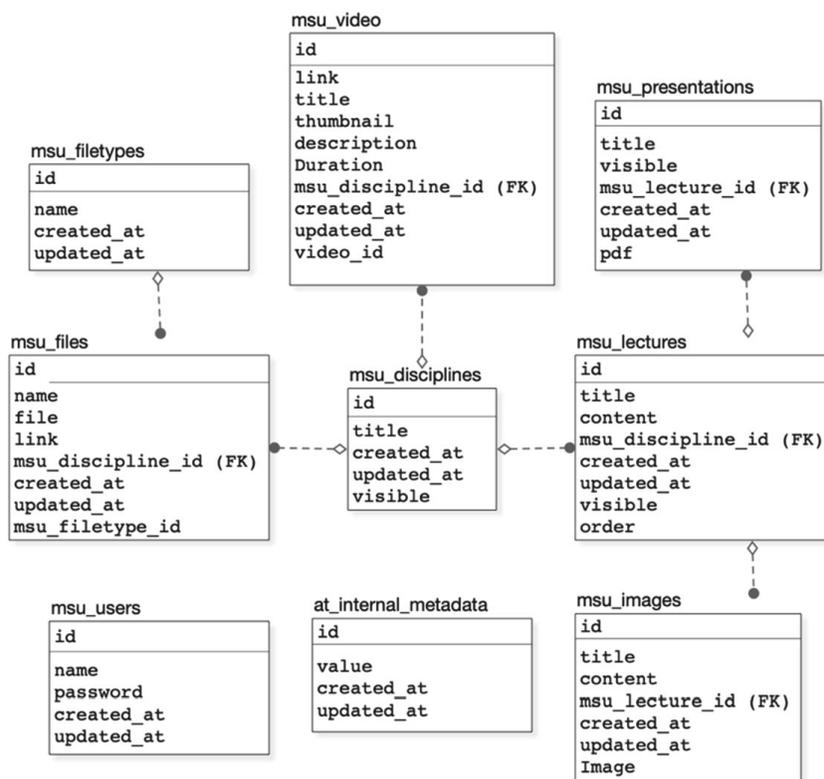


Рис. 2. Логическая модель базы данных MSUniversity (нотация IDEF1x)  
 Fig. 2. Logical model of the MSUniversity database (notation IDEF1x)

визуализации документной модели данных подготовим пример опираясь на реально существующую реляционную базу данных *WEB-приложения MSUniversity* [11], структура которой приведена на рис. 2.

Для этого структура базы данных была описана в формате *JSON*. Так как для проверки возможностей приложения *JSONDesigner* в части визуализации структуры типы данных полей не играют существенной роли при описании структуры использовались псевдотипы. Описание структуры в формате *JSON* приведено в листинге 1.

### Листинг 1. Структура БД описанная с помощью JSON

#### Listing 1. Database structure described using JSON

```

{
  "title": "some_title",
  "created_at": "cr_date",
  "updated_at": "up_date",
  "visible": "y/n",
  "msu_files": {
    "name": "name",
    "file": "file",
    "link": "link", "created_at": "cr_date",
  }
}
    
```

```

"updated_at": "up_date",
"msu_filetypes": {
  "name": "name",
  "created_at": "cr_date",
  "updated_at": "up_date"
},
"msu_videos": {
  "link": "link",
  "title": "title",
  "thumbnail": "thumbnail",
  "description": "description",
  "duration": "duration",
  "created_at": "cr_date",
  "updated_at": "up_date",
  "video_id": "video_id"
},
"msu_lectures": {
  "title": "title",
  "content": "content",
  "created_at": "cr_date",
  "updated_at": "up_date",
  "visible": "y/n",
  "order": ["1", "2"]
},
"msu_images": {
  "title": "title",
  "created_at": "cr_date",
  "updated_at": "up_date",
  "image": "image"
},
"msu_presentations": {
  "title": "title",
  "visible": "y/n",
  "created_at": "cr_date",
  "updated_at": "up_date",
  "pdf": "doc"
}
    
```

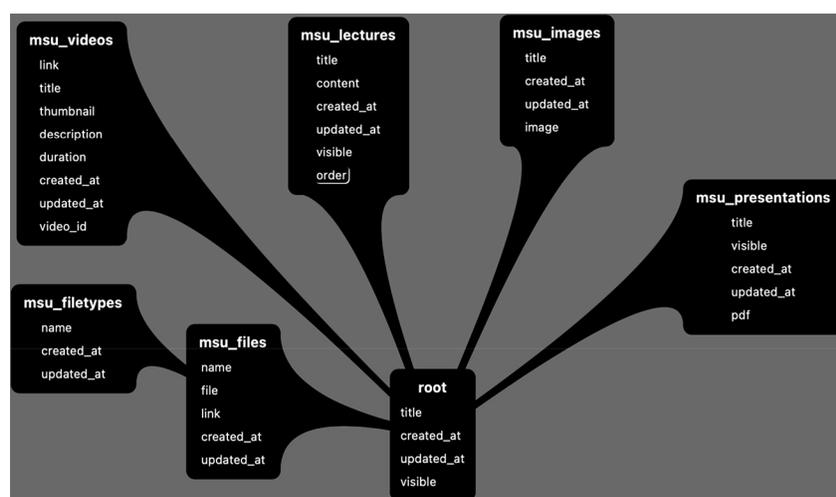


Рис. 3. Результат визуализации JSON-файла в программном средстве JsonDesigner

Fig. 3. Result of JSON file visualization in the software JsonDesigner

Код, приведённый на листинге 1 был загружен в приложение JsonDesigner, полученный результат приведён на рис. 2.

Как видно из рис. 3, графическая нотация JsonDesigner позволяет четко и наглядно передать структуру JSON-документа, что, однако не является достаточным для изображения документной модели хранения данных. Выделим несколько основных проблем использования рассмотренной графической нотации для представления документной модели:

1. Данная нотация не предоставляет возможности отображать более одной коллекции, что является критическим фактором для моделирования комплексных документных баз данных.

2. По упомянутой в п. 1 причине, в приведенной нотации невозможно показать отношение документов к коллекциям в модели.

3. Отсутствует визуализация связей типа «включение», типа «ассоциация» и типа «ссылка на документ».

4. Отсутствует возможность визуализации в схеме типов данных для атрибутов документа.

### Описание разработанной графической нотации документной модели данных

Рассмотрев аспекты существующих графических нотаций, особенностей документных моделей данных и имеющихся на текущий момент программных средств, был разработан набор графических элементов и правил их применения для построения документной модели данных.

При выборе элементов для построения графической модели предпочтения отдавались тем графическим элементам, которые хорошо известны и понятны практикам, поскольку в похожем виде, они используются в графических нотациях реляционной модели данных.

### Описание элементов

Ниже приводится описание предлагаемых графических обозначений.

#### 1) Коллекция:

Коллекции представляют собой объекты-контейнеры. Они не содержат собственных атрибутов, предназначенных для хранения данных и служат для объединения документов в единый набор. База данных может содержать одну и более коллекций.

База данных может содержать множество коллекций, которые в свою очередь могут содержать различные виды документов. Поэтому, в зависимости от конкретной ситуации предлагается использовать один из двух вариантов.

В случае использования первого варианта коллекция изображается в виде прямоугольника, внутри которого отображается её название (рис. 4 (а)). Принадлежность документов при этом показывается с помощью связи специального типа «Ассоциация». Данный способ может быть использован при небольшом количестве коллекций и принадлежащих им документов.

Недостаток данного варианта состоит в необходимости использования дополнительного типа связи, которая ухудшит восприятие схемы с ростом числа документов.

Второй вариант — изображается в виде прямоугольника, в левом верхнем углу которого указывается название коллекции. Название коллекции отделяется от содержания горизонтальной чертой и/или цветом. Размер прямоугольника должен быть достаточным для того, чтобы вместить все документы принадлежащие коллекции (рис. 4 (б)).

Данный способ также имеет недостаток, связанный с тем, что с увеличением количества атрибутов в документе увеличивается и площадь прямоугольника, что также может привести к ухудшению восприятия конечной модели.

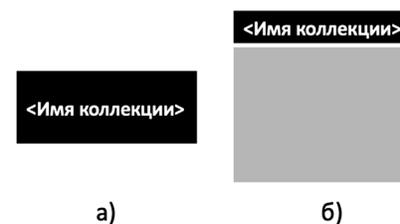


Рис. 4. Варианты отображения коллекций. а) Как отдельный элемент; б) Как контейнер  
Fig. 4. Options for displaying collections. a) As a separate element; b) As a container

Примеры применения предложенных вариантов будут приведены далее.

**2) Документ:**

Основная единица хранения данных в модели. Содержит в себе необходимое количество атрибутов, представленных парами «ключ: значение», где ключ определяется именем поля, а значение — типом данных, которое оно может принимать.

Документ изображается в виде таблицы, состоящей минимум из двух строк и двух столбцов, ячейки первой строки таблицы объединены и служат для ввода имени документа, последующие строки должны содержать пары «ключ (имя поля) — допустимый тип значения» (рис. 5).

<Имя документа>	
<Имя поля>	<тип>
<Имя поля>	[тип]

**Рис. 5. Графическое изображение документа**  
**Fig. 5. Graphic representation of the document**

Следует отметить, что атрибут может содержать более одного элемента (массив), в этом случае необходимо заключить имя типа в квадратные скобки.

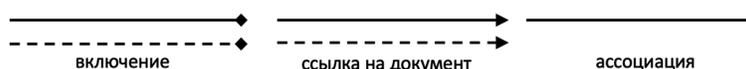
**3) Связи:**

Как уже упоминалось, в документной модели возможно три типа связей:

- связь «включение», когда один документ является частью другого документа;
- связь «ссылка на другой документ», когда один или несколько документов содержат ссылку на другой документ этой же или другой коллекции;
- ассоциация (или принадлежность), для обозначения принадлежности документа к коллекции.

Графические элементы для отображения связей приведены на рис. 6.

Для построения логических моделей документных баз данных, используя разработанные графические элементы, и обеспечить однозначное вос-

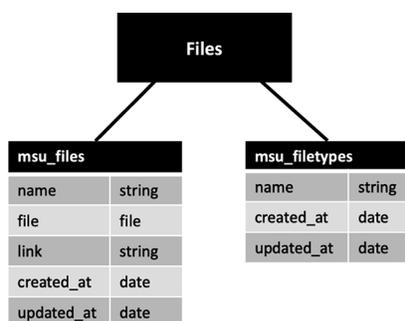


**Рис. 6. Графическое изображение связей**  
**Fig. 6. Graphic representation of links**

приятие готовой модели был сформирован набор правил применения элементов, указывающий, когда следует использовать то или иное обозначение. Предложенные правила согласованы с базовыми правилами документной модели данных.

**Правила применения элементов графической нотации документной модели данных**

1. Модель должна содержать хотя бы одну коллекцию.
2. Каждый документ должен принадлежать к не более чем к одной коллекции.
3. Каждый документ должен содержать не менее одного атрибута.
4. Демонстрация принадлежности документа к коллекции возможна одним из двух описанных ниже способов:
  - 1) Документ принадлежит к коллекции, если между элементом «Коллекция» (рис. 4 (а)) и документом (рис. 5) установлена связь «Ассоциация» (рис. 6). Данный способ проиллюстрирован на рис. 7.



**Рис. 7. Указание принадлежности документа к коллекции с помощью ассоциации**

**Fig. 7. Indication of a document belonging to a collection using an association**

2) Документ принадлежит к коллекции, если он изображен на свободном месте внутри элемента «Коллекция» (рис. 4 (б))

под чертой (или выделенной цветом областью), отделяющей название коллекции (рис. 8). Размеры элемента коллекция должны позволять вместить все входящие в неё документы.

Files			
msu_files		msu_filetypes	
name	string	name	string
file	file	created_at	date
link	string	updated_at	date
created_at	date		
updated_at	date		

**Рис. 8. Способ изображения документов включённых в коллекцию**

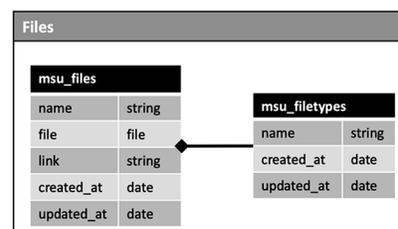
**Fig. 8. Method of displaying documents included in the collection**

5. Между документами могут быть установлены следующие типы связей:

1) Связь «Включение». Применяется если один документ входит в состав другого документа.

Для обозначения включения между двумя документами необходимо применить связь «Включение» в направлении от включаемого к родительскому документу.

Связь «Включение» изображается сплошной линией в случае, когда документ должен быть обязательно включён в состав родительского документа. Пример использования связи «Включение» приведён на рисунке ниже (рис. 9):



**Рис. 9. Пример использования связи «Включение» (обязательное)**

**Fig. 9. An example of using the connection “Enable” (mandatory)**

Приведённый пример (рис. 9) может быть описан следующей фразой: «Документ *msu\_files* обязательно включает в себя не менее одного экземпляра документа *msu\_filetypes*»

В случае если вложение документа необязательно, то связь «Включение» изображается в виде пунктирной линии. Пример связи «Включение» для случая необязательного вложения документов описывается следующим утверждением: «Документ *msu\_lecture* может включать в себя экземпляры документа *msu\_images*». Данная связь изображается на рис. 10.

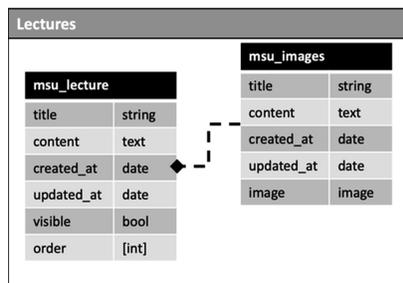


Рис. 10. Пример использования связи «Включение» (необязательное)

Fig. 10. An example of using the connection «Enable» (optional)

2) Связь «Ссылка на документ». Применяется в случае, если один документ ссылается на другой документ той же или другой коллекции базы данных. Обозначается в виде стрелки, направленной от ссылающегося документа.

В случае, если наличие такой ссылки обязательно, то линия изображается сплошной линией (рис. 11):

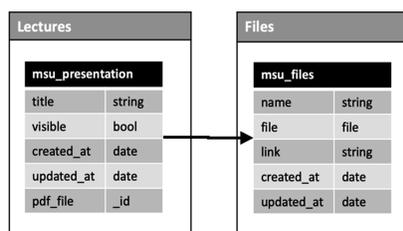


Рис. 11. Пример использования «Ссылка на документ» (обязательное наличие ссылки)

Fig. 11. Example of using «Link to document» (mandatory link)

На рис. 11 описывается следующая ситуация: «Документ *msu\_presentation*, входящий в коллекцию *Lectures*, обязательно содержит ссылку на документ *msu\_files* коллекции *Files*»

Если ссылка на документ может отсутствовать, то линия изображается пунктиром (рис. 12):

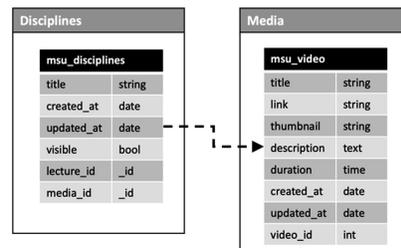


Рис. 12. Пример использования «Ссылка на документ» (необязательное наличие ссылки)

Fig. 12. Example of using «Link to document» (optional link)

На рис. 12 приведена ситуация необязательной ссылки на документ: «Документ *msu\_disciplines*, коллекции *Disciplines* может содержать ссылку на документ *msu\_video* коллекции *Media*»

Описанные графические элементы и правила в совокупности представляют собой законченный инструмент пригодный для построения логической модели документной базы данных.

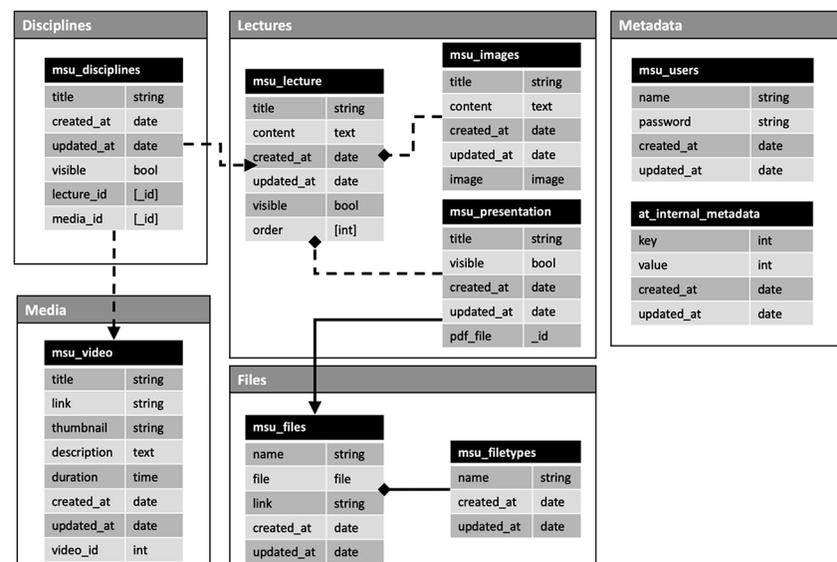


Рис. 13. Графическая модель документной модели данных

Fig. 13. Graphical model of the document data model

### Пример практического применения модели

Руководствуясь определенными графическими элементами нотации, было произведено построение логической модели документной базы данных WEB-приложения, для которого в настоящее время существует реляционная модель приведённая на рис. 2. Результат приведен на рис. 13.

Документы, хранящие информацию об имеющемся контенте отсортированы по тематике и разнесены по соответствующим коллекциям. Модель содержит 9 видов документов, собранных в 5 коллекций:

– *Disciplines* — документы данной коллекции предназначены для хранения информации о дисциплинах, доступных для изучения с помощью приложения. Документ коллекции *msu\_disciplines* может ссылаться на медиаматериалы расположенные на сторонних платформах (например, YouTube) с помощью необязательной ссылки на документ *msu\_video*, коллекции *Media*, и на тематические материалы к лекционным и практическим занятиям с помощью необя-

зательной ссылки на документ *msu\_lectures*, коллекции *Lectures*;

– *Lectures* — коллекция содержит документы, отвечающие за хранение материалов (текстовых лекций, наглядных пособий и т.д.). Коллекция содержит три вида документов: *msu\_lectures*, содержащий основную информацию о занятии, *msu\_images*, хранящий информацию о прикрепленных изображениях, и *msu\_presentation*, в котором содержатся файлы, прикрепленные к какому-либо занятию. Документы *msu\_images* и *msu\_presentation* могут быть включены в состав документа *msu\_lectures* при необходимости;

– *Files* — коллекция, содержащая файлы различных типов, используемых приложением. В первую очередь это файлы, прикрепляемые к занятиям. В рассматриваемом примере это реализуется с помощью связи «ссылка на документ», которой связаны *msu\_presentation*, коллекции *Lectures* и *msu\_files* коллекции *Files*, которая означает что если существует экземпляр документа *msu\_presentation*, то он обязательно должен ссылаться на экземпляр документа *msu\_files*. Документ *msu\_filetypes* — описывает тип (формат) файла, описываемого документом *msu\_files*. Экземпляр этого документа должен быть обязательно включен в состав экземпляра документа *msu\_files*. В данном случае не используется связь «ссылка на документ», так как проведение нормали-

зации документных баз данных не требуется, и официальная документация *MongoDB* рекомендует отдавать предпочтение связи типа «Включение» при проектировании;

– *Metadata* — содержит документы, описывающие пользователей приложения и данные, необходимые для корректной работы приложения.

Как уже упоминалось, документная модель данных не имеет заранее определенной схемы, в связи с чем у СУБД нет возможности определить соответствие добавляемых пользователем данных схеме, заданной проектировщиком. Используя графическую модель, представленную на рис. 13, пользователь получает полное представление о логическом устройстве документной базы данных и структуре входящих в нее документов, что позволяет снизить различные риски на этапе эксплуатации.

### Заключение

В ходе работы был произведен анализ существующих графических нотаций, применяемых при проектировании реляционных баз данных и выявлены основные графические стереотипы, рассмотрены основные особенности документной модели данных, произведен поиск и анализ существующих решений для визуализации документных моделей данных и близких к ним структур. Основным результатом проведенного ис-

следования стало создание графической нотации моделирования документных баз данных. Был предложен новый графический инструментарий, который, с одной стороны хорошо знаком практикам, ранее работавшим с реляционными моделями данных, а с другой стороны отражает особенности структуры документной модели. Практическое применение разработанной модели показало удобство ее использования как в процессе проектирования документных баз данных, так и в процессе обучения студентов в рамках данной предметной области.

Одним из направлений дальнейшего развития нотации является разработка и проверка в практических условиях механизма визуализации многократного вложения однотипных документов, что позволит более точно отразить особенности некоторых предметных областей. Данный механизм для документных БД позволит пользователю понять на сколько экземпляров документа он должен сослаться (или может сослаться) или включить в родительский документ.

Применение графических моделей, построенных в предложенной графической нотации, позволит исследователям создавать и иллюстрировать типовые паттерны документных баз данных, что, несомненно, положительно скажется на динамике развития перспективных технологий хранения данных.

### Литература

1. Кузнецов С.Д. Базы данных: языки и модели. М.: ООО «Бином-Пресс», 2008. 720 с.
2. Попов Ф.А., Максимов А.В. Подходы к проектированию баз данных для автоматизированных систем [Электрон. ресурс] // Известия АлтГУ. 2003. № 1. Режим доступа: <https://cyberleninka.ru/article/n/podhody-k-proektirovaniyu-baz-dannyh-dlya-avtomatizirovannyh-sistem>. (Дата обращения: 20.07.2021).
3. Codd E. F. A relational model of data for large shared data banks // Comm. ACM. 1970. № 113(6).

4. Chen P. The Entity-Relationship Model Toward a Unified View of Data // ACM Transactions on Database Systems. 1976. Т. 1. № 11.

5. Moniruzzaman A. B. M., Hossain S. A. Nosql database: New era of databases for big data analytics-classification, characteristics and comparison // arXiv preprint arXiv:1307.0191. 2013.

6. DBMS popularity broken down by database model [Электрон. ресурс]. Режим доступа: [https://db-engines.com/en/ranking\\_categories](https://db-engines.com/en/ranking_categories). (Дата обращения: 01.08.2021).

7. Миронов В. В., Гусаренко А. С., Юсупова Н. И. Ситуационно-ориентированные базы

данных как виртуальный интеграционный слой в веб-приложениях // Information Technologies for Intelligent Decision Making Support (ITIDS'2016). 2016. С. 123–128.

8. Зимовец А. И., Хомоненко А. Д. Обоснование выбора модели хранения данных для системы мониторинга космического пространства // Автоматика на транспорте. 2019. Т. 5. № 2.

9. Блинков Ю. А., Панкратов И. А. Документ-ориентированное хранение и обработка научных публикаций // Математическое моделирование, компьютерный и натурный эксперимент в естественных науках. 2018. № 4. С. 28–36.

10. Бедердинова О. И. Результаты проектирования базы данных комплекса оборудования процессов лесопиления с использованием case-технологии IDEF1X // Информационные технологии в проектировании и производстве. 2007. № 3. С. 33–35.

11. Главная страница MSUniversity [Электрон. ресурс] // MSUniverdity.ru. Режим доступа: <http://msuniversity.ru>. (Дата обращения: 20.05.2021).

12. Алексеев В. А., Телегина М. В., Янников И. М. Создание базы данных биомониторинга потенциально опасных объектов // Вестник Ижевского государственного технического университета. 2008. № 4. С. 138–143.

13. Азовцев А. И. и др. Разработка инфологической модели базы данных предварительного информирования таможенных органов для судоходной компании // Морские интеллектуальные технологии. 2016. № 3(1). С. 327–332.

14. Горшков Е. А., Калинина А. В., Куликова С. А. Применение основ моделирования для автоматизации офисной деятельности санаторно-курортных учреждений // ББК 1 Н 34. 2019. С. 1683.

15. Kusiak A., Letsche T., Zakarian A. Data modelling with IDEF1x // International Journal of Computer Integrated Manufacturing. 1997. Т. 10. № 6. С. 470–486. DOI: 10.1080/095119297131039.

16. Everest G. C. Basic data structure models explained with a common example // Proc. Fifth Texas Conference on Computing Systems. 1976. С. 18–19.

17. Юсупова Д. Ж. Возможности семантической модели ERD на этапе инфологического проектирования базы данных // Современные материалы, техника и технология. 2019. С. 423–425.

18. Горшков Е. А., Калинина А. В., Куликова С. А. Применение основ моделирования для автоматизации офисной деятельности санаторно-курортных учреждений // ББК 1 Н 34. 2019. С. 1683.

19. Hay D. C. A comparison of data modeling techniques // Essential Strategies. 1999. Т. 41. С. 43.

20. Документация MongoDB [Электрон. ресурс]. Режим доступа: <https://docs.mongodb.com> (Дата обращения: 25.05.2021).

21. Страница Веб-приложения JSONDesigner [Электрон. ресурс]. Режим доступа: <https://jsondesigner.com/#/>. (Дата обращения: 19.05.2021).

## References

1. Kuznetsov S.D. Bazy dannykh: yazyki i modeli = Databases: languages and models. Moscow: OOO Binom-Press; 2008. 720 p. (In Russ.)

2. Popov F. A., Maksimov A. V. Approaches to designing databases for automated systems [Internet]. Izvestiya AltGU = News of Altai State University. 2003; 1. Available from: <https://cyberleninka.ru/article/n/podhody-k-proektirovaniyu-baz-dannyh-dlya-avtomatizirovannyh-sistem>. (cited 20.07.2021). (In Russ.)

3. Codd E. F. A relational model of data for large shared data banks. Comm. ACM. 1970; 113(6).

4. Chen P. The Entity-Relationship Model: Toward a Unified View of Data. ACM Transactions on Database Systems. 1976; 1: 11.

5. Moniruzzaman A. B. M., Hossain S. A. Nosql database: New era of databases for big data analytics-classification, characteristics and comparison. arXiv preprint arXiv:1307.0191. 2013.

6. DBMS popularity broken down by database model [Internet]. Available from: [https://db-engines.com/en/ranking\\_categories](https://db-engines.com/en/ranking_categories). (cited 01.08.2021).

7. Mironov V.V., Gusarenko A.S., Yusupova N.I. Situation-oriented databases as a virtual integration layer in web applications. Information Technologies for Intelligent Decision Making Support (ITIDS'2016). 2016: 123–128. (In Russ.)

8. Zimovets A.I., Khomonenko A.D. Substantiation of the choice of data storage model for the space monitoring system. Avtomatika na transporte = Automation on transport. 2019; 5: 2. (In Russ.)

9. Blinkov Yu.A., Pankratov I.A. Document-oriented storage and processing of scientific publications. Matematicheskoye modelirovaniye, komp'yuternyy i naturnyy eksperiment v yestestvennykh naukakh = Mathematical modeling, computer and natural experiment in natural sciences. 2018; 4: 28–36. (In Russ.)

10. Bederdinova O.I. Results of designing a database of a complex of equipment for sawmilling processes using case-technology IDEF1X. Informatsionnyye tekhnologii v proyektirovanii i proizvodstve = Information technologies in design and production. 2007; 3: 33–35. (In Russ.)

11. Glavnaya stranitsa MSUniversity = Main page of MSUniversity [Internet]. MSUniverdity.ru. Available from: <http://msuniversity.ru>. (cited 20.05.2021). (In Russ.)

12. Alekseyev V.A., Telegina M.V., Yannikov I.M. Creation of a database of biomonitoring of potentially dangerous objects. Vestnik Izhevskogo gosudarstvennogo tekhnicheskogo universiteta = Bulletin of the Izhevsk State Technical University. 2008; 4: 138–143. (In Russ.)

13. Azovtsev A.I. et al. Development of an infological model of a database of preliminary

informing of customs authorities for a shipping company. *Morskiye intellektual'nyye tekhnologii = Marine intellectual technologies*. 2016; 3(1): 327-332. (In Russ.)

14. Gorshkov Ye.A., Kalinina A.V., Kulikova S.A. Application of modeling principles for automation of office activities of sanatorium-resort institutions. *BBK 1 N 34 = LBC 1 N 34.* 2019: 1683. (In Russ.)

15. Kusiak A., Letsche T., Zakarian A. Data modelling with IDEF1x. *International Journal of Computer Integrated Manufacturing*. 1997; 10; 6: 470-486. DOI: 10.1080/095119297131039.

16. Everest G. C. Basic data structure models explained with a common example. *Proc. Fifth Texas Conference on Computing Systems*. 1976: 18-19.

17. Yusupova D.Zh. Possibilities of the semantic model of ERD at the stage of infological database design. *Sovremennyye materialy, tekhnika i tekhnologiya = Modern materials, equipment and technology*. 2019: 423-425. (In Russ.)

18. Gorshkov Ye.A., Kalinina A.V., Kulikova S.A. Application of modeling principles for automation of office activities of sanatorium-resort institutions. *BBK 1 N 34 = LBC 1 N 34.* 2019: 1683. (In Russ.)

19. Hay D.C. A comparison of data modeling techniques. *Essential Strategies*. 1999: 41: 43.

20. Dokumentatsiya MongoDB [Internet]. Available from: <https://docs.mongodb.com> (cited 25.05.2021).

21. Stranitsa Veb-prilozheniya JSONDesigner [Internet]. Available from: <https://jsondesigner.com/#/>. (cited 19.05.2021).

#### **Сведения об авторах**

**Михаил Вячеславович Смирнов**

*К.э.н., доцент кафедры КБ-9*

*МИРЭА – Российский технологический университет,*

*Москва, Россия*

*Эл. почта: mikhailsmirnov@gmail.com*

**Руслан Сергеевич Толмасов**

*Преподаватель кафедры КБ-9*

*МИРЭА – Российский технологический университет,*

*Москва, Россия*

*Эл. почта: tol\_rus@mail.ru*

#### **Information about the authors**

**Mikhail V Smirnov**

*Cand. Sci. (Associate Professor at the Department of “Subject-oriented information systems”*

*MIREA - Russian Technological University,*

*Moscow, Russia*

*E-mail: mikhailsmirnov@gmail.com*

**Ruslan S Tolmasov**

*Instructor at the Department of “Subject-oriented information systems”*

*MIREA - Russian Technological University,*

*Moscow, Russia*

*E-mail: tol\_rus@mail.ru*