УДК 004.652.5
DOI 10.25205/1818-7900-2019-17-3-123-134

Обзор объектно-ориентированной парадигмы в приложении к разработке баз данных

Е. А. Яценко

Сургутский государственный университет
Сургут, Россия

Аннотация

В статье приводится обзор проектов, технологий, программных продуктов, разрабатываемых для реализации идей объектно-ориентированного подхода к проектированию баз данных. В 1980-е гг. велось множество проектов, посвященных идее ООБД, многие специалисты ожидали, что в ближайшее время реляционные базы данных будут вытеснены объектно-ориентированными. Несмотря на впечатляющее количество проектов, проводимых как коллективами ученых, так и коммерческими компаниями, ориентированными на практическое внедрение, не появилось четкой формулировки объектно-ориентированной модели данных, каждый коллектив представлял свое видение применения принципов ООП к проектированию баз данных. Отсутствие универсальной модели данных с проработанным математическим аппаратом (как в случае реляционных баз данных) и сегодня является основной проблемой распространения ООСУБД. Однако применение реляционных СУБД порождает ряд проблем, которые наиболее остро ощущаются в таких областях, как системы автоматизированного проектирования, автоматизированное производство, системы, основанные на знаниях, и др. ООБД позволяют объединить программный код и данные, избежать различий между представлениями информации в базе и прикладной программе, вследствие чего интерес к ним проявляют современные разработчики. Современный рынок программного инструментария представлен рядом ООСУБД, но ни одна из них не может конкурировать с крупнейшими поставщиками СУБД.

Ключевые слова

объектно-ориентированный подход, объектно-ориентированная база данных, объектно-ориентированная СУБД, модель данных

Для цитирования

Яценко Е. А. Обзор объектно-ориентированной парадигмы в приложении к разработке баз данных // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17, № 3. С. 123–134. DOI 10.25205/1818-7900-2019-17-3-123-134

Overview of Object-Oriented Paradigm in an Appendix to the Development of Databases

E. A. Yatsenko

Surgut State University
Surgut, Russian Federation

Abstract

The article provides an overview of projects, technologies, software products developed to implement the ideas of an object-oriented approach to database design. In the 80s of the 20th century, there were many projects devoted to the idea of OODB, many experts expected that in the near future relational databases would be crowded out with object-oriented ones. Despite the impressive number of projects conducted by both teams of scientists and commercial companies focused on practical implementation, there was no clear formulation of an object-oriented data model, each team presented its own vision of applying object-oriented concepts to database design. The absence of a universal data

© Е. А. Яценко, 2019

model, with a well-developed mathematical apparatus (as in the case of relational databases), is still the main problem in the distribution of an OODBMS. However, the use of relational DBMS raises a lot of problems that are most acutely felt in areas such as computer-aided design, computer-aided production, knowledge-based systems, and others. OODB allow to combine the program code and data, to avoid differences between the representations of information in the database and the application program, as a result of which modern developers show interest in them. There are a lot of OODBMS, but they cannot compete with the largest storage organization systems.

Keywords

object-oriented approach, object-oriented database, object-oriented database, data model

For citation

Yatsenko E. A. Overview of Object-Oriented Paradigm in an Appendix to the Development of Databases. *Vestnik NSU. Series: Information Technologies*, 2019, vol. 17, no. 3, p. 123–134. (in Russ.) DOI 10.25205/1818-7900-2019-17-3-123-134

Объектно-ориентированная база данных (ООБД) – база данных, данные в которой структурированы в соответствии с моделью данных, основанной на понятиях и принципах объектно-ориентированного проектирования (ООП).

В 1980-е гг. велось множество проектов, посвященных идее ООБД, многие специалисты были уверены, что в ближайшие 2–3 года реляционные базы данных будут вытеснены объектно-ориентированными. Сергей Дмитриевич Кузнецов среди проектов, оказавших наибольшее влияние на развитие ООБД, называет Encore [1] (Брауновский университет, Провиденс, Род-Айленд, США); Cactis [2] (Университет Колорадо, США), Thor [3] (Массачусетский технологический институт, США); Exodus [4] (Висконсинский университет, США); Pisa [5] (университеты Глазго и Сент-Э́ндрюсм, Шотландия).

Большой вклад в развитие объектно-ориентированной парадигмы сделала группа исследовательского института OGI (Oregon Graduate Institute). OGI был уникальным, частным, последипломным исследовательским университетом в округе Вашингтон, штат Орегон, в западной части Портленда, с 1963 по 2001 г., с 1989 г. именовался Орегонским институтом высших учебных заведений. Результатом деятельности OGI в области объектно-ориентированных баз данных стала ООСУБД Gemstone [6; 7]. В качестве базового языка GemStone использовался Smalltalk [8; 9]. Данный язык разработан в 1970-х гг. на основе уже существовавших в то время концепций, в нем была введена новая терминология, являющаяся наиболее характерной для современных языков программирования. Smalltalk основан на идее посылки сообщений, реализует возможность программирования с динамической типизацией, оказал большое влияние на развитие таких объектно-ориентированных языков, как Java, Ruby, Objective-C. Многие революционные идеи появились в сообществе Smalltalk. К ним можно отнести рефакторинг, шаблоны проектирования (применительно к ПО), карты «класс – обязанности – взаимодействие» и экстремальное программирование в целом. GemStone была одной из первых коммерчески доступных ООСУБД.

Проект корпорации MCC (Microelectronics and Computer Technology Corporation) ознаменован появлением таких СУБД, как Itasca [10] и UniSQL [11; 12]. Во французском исследовательском центре INRIA [1] осуществлялся проект Altair, в рамках которого была создана O2. Компанией «Hewlett Packard» была разработана IRIS, DEC выпустили систему Trellis. В виде завершенных коммерческих систем появились G-Base, упомянутая ранее, Gemstone, Statice, Vbase.

К первым СУБД, реализующим принципы объектно-ориентированного подхода, Сергей Дмитриевич Кузнецов сформулировал два основных замечания. Первое сводится к невозможности практического применения указанного спектра систем ввиду таких характеристик, как ненадежность, отсутствие поддержки безопасности данных, низкая производительность и др. Второе замечание касается отсутствия единого языка, коим, по мнению Сергея Дмит-

[1] Inria, a major actor in the European and international research arena. INRIA: официальный сайт. URL: https://www.inria.fr/

риевича, должен был быть C++. Необходимость изучать новый язык, а у каждой такой СУБД он был свой, понижала коммерческий интерес [5].

Дальнейшее развитие методологии проектирования баз данных находило отражение в так называемых «Манифестах». Первый манифест был издан в 1989 г. Его авторами были преимущественно ученые [13], это была первая попытка собрать имеющиеся разноплановые наработки в области применения ОО-подхода к проектированию БД в единое определение ООСУБД. Был приведен перечень основных характеристик, которыми должна обладать СУБД, чтобы быть классифицированной как ООСУБД. Характеристики подразделялись на три группы: обязательные, необязательные и открытые. Обязательными характеристиками должна обладать любая СУБД, относящаяся к классу объектно-ориентированных, необязательные характеристики могут быть как присущи данным системам, так и нет. Открытые характеристики реализуются разработчиком в соответствии с его собственными соображениями, унификация на этом уровне не предполагается. К обязательным, например, были отнесены такие характеристики, как идентифицируемость объектов, инкапсуляция, классы; множественное наследование – это уже пример необязательной характеристики, к открытым относится, например, парадигма программирования. Будущее развитие СУБД авторы манифеста видели в ООСУБД, поддерживающей традиционные возможности.

Второй манифест появился в 1990 г. [14]. В противовес первому авторы второго манифеста инженеры – разработчики баз данных, ориентированных на применение языка SQL. В данном документе дается классификация СУБД по поколениям: ранее созданные реляционные, иерархические и сетевые СУБД – первое поколение, используемые на текущий момент реляционные СУБД – второе, СУБД третьего поколения – те, что соответствуют предложенным в документе требованиям. Требования включают три принципа и 13 предложений. В соответствии с первым принципом СУБД третьего поколения должны предоставлять средства хранения и манипулирования объектами, структура которых может включать более широкий спектр типов данных, таких как тексты, например, а также средства создания правил для элементов данных, как ссылочная целостность в реляционных БД и более сложные. Второй принцип состоит в необходимости поддержки возможностей СУБД второго поколения. Третий принцип гласит об открытости СУБД, т. е. о поддержке интеграции с прикладными программами, разработанными на различных платформах.

Летом 1991 г. была задумана и впоследствии образована Object Database Management Group (ODMG) [13; 15], в 1998 г. была переименована в Object Data Management Group, чтобы отобразить распространение своей деятельности не только на объектно-ориентированные базы данных, но и технологии ORM (Object-Relational Mapping) [2]. ORM – объектно-реляционное отображение, или преобразование, – технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных».

Основной целью ODMG было разработать комплекс спецификаций, которые позволят разрабатывать портируемые приложения для объектной базы данных и ORM-приложений. В период с 1993 по 2001 г. ODMG опубликовал пять изменений в своей спецификации. Последней была ODMG версии 3.0, после чего группа распалась.

Предлагаемая ODMG архитектура показана на рис. 1. В этой архитектуре определяются способ хранения данных и разные виды пользовательского доступа к этому «хранилищу данных». Единое хранилище данных доступно из языка определения данных, языка запросов и ряда языков манипулирования данными. На рис. 1 ODL означает Object Definition Language (язык определения объектов), OQL – Object Query Language (язык объектных запросов) [16] и OML – Object Manipulation Language (язык манипулирования объектами).

[2] Stack Exchange Inc. Is ObjectDB production ready? URL: https://stackoverflow.com/questions/5291950/is-objectdb-production-ready

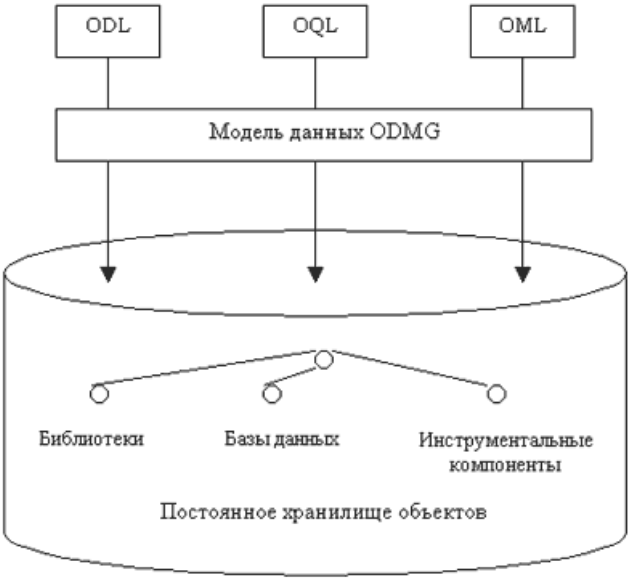

Рис. 1. Архитектура ODMG
Fig. 1. ODMG architecture

После публикации второго манифеста на рынок вышел целый новый класс СУБД, именуемый объектно-реляционными СУБД, среди них Informix Universal Server [17], Oracle 8, DB 2 Universal Database.

В марте 1995 г. была впервые официально опубликована статья Хью Дарвена и Кристофера Дейта, названная авторами «Третьим манифестом» [18]. Авторы статьи ищут прочные основы для будущего управления данными. При этом Хью Дарвен и Кристофер Дейт считают, во-первых, что такие основы не способен обеспечить язык баз данных SQL, во-вторых, любые такие основы должны твердо корениться в реляционной модели данных, впервые представленной миру Э. Ф. Коддом в 1969 г. При этом авторы сознают ценность поддержки горячо обсуждаемых возможностей объектно-ориентированного подхода, называя их «ортогональными реляционной модели». И в силу этого реляционная модель не нуждается в каких-либо расширениях, коррекции или категоризации, а самое главное, в каких-либо искажениях, для того чтобы внедрить указанные возможности в какой-либо язык баз данных, способный представлять те основы, которые мы ищем. Основу Манифеста представляет описание такого языка: «Допустим, что такой язык имеется и называется D». Авторы приводят предписания и запреты языка D. Некоторые из этих предписаний проистекают из реляционной модели данных и названы РМ-предписаниями. Предписания, которые не происходят из реляционной модели, названы остальными ортогональными предписаниями – ОО-предписаниями. Подобным же образом категоризованы и запреты для языка D.

Несмотря на впечатляющее количество проектов, проводимых как коллективами ученых, так и коммерческими компаниями, ориентированными на практическое внедрение, не появилось четкой формулировки объектно-ориентированной модели данных, каждый коллектив представлял свое видение применения принципов ООП к проектированию баз данных.

Отсутствие универсальной модели данных, с проработанным математическим аппаратом (как в случае реляционных баз данных) и сегодня является основной проблемой распространения ООСУБД.

А. М. Эльдарханов предлагает рассматривать проблематику разработки ООСУБД с позиции разделения по трем уровням представления данных. Так же, как и в классической модели структуры базы данных ANSI / SPARC [19; 20], представлены три уровня описания данных.

В обеих названных иерархиях уровни располагаются в порядке убывания степени абстракции. Самый нижний уровень, наиболее приближенный к физическим процессам, реализующим задачи хранения и управления данными, в обеих моделях является физический уровень. Над физическим уровнем А. М. Эльдарханов разместил уровень модели данных, далее следует уровень формальной математической модели [21].

Для создания формальной математической модели существуют два подхода. Первый подход предполагает последовательное исполнение ряда действий с целью получения заданного результата аналогично функциональному программированию и называется функциональным. В литературе представлены два метода данного подхода: алгебра select-project-join (SPJ) и многосортное исчисление предикатов высшего порядка [22]. Язык SQL реализует алгебру SPJ [23].

Второй подход создания формальной математической модели – дедуктивный. Дедуктивный подход основан на использовании логического вывода, запросы к базам данных формулируются в виде аксиом и правил. Кристофер Дейт в своем фундаментальном труде приводил следующий пример (листинг 1, 2)

Листинг 1

GOOD_SUPPLIER (s, st, sc) ⇐ S (s, sn, st, sc) AND st > 15

На листинге 1 представлено определение предиката «GOOD_SUPPLIER» на языке Datalog. На листинге 2 приведены примеры запросов к предикату «GOOD_SUPPLIER».

Листинг 2

1. Определить всех добросовестных поставщиков. ?
   ⇐GOOD_SUPPLIER (s, st, sc)
2. Определить всех добросовестных поставщиков из Парижа. ?
   ⇐ GOOD_SUPPLIER (s, st, Paris)
3. Является ли поставщик si добросовестным? ?
   ⇐ GOOD_SUPPLIER (S1, st, sc)

Средний уровень представления данных – уровень модели данных, представлен двумя аспектами: поведенческим и структурным. Структурный аспект представляет два уровня описания: уровень данных (рис. 2) и уровень схемы (рис. 3).

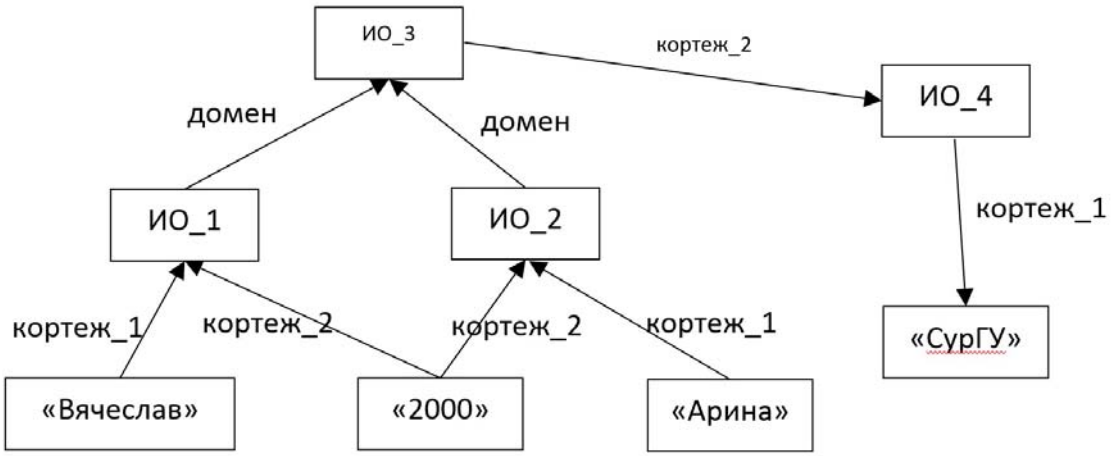

Рис. 2. Описание модели данных на уровне данных. Предметная область: студенты вуза

Fig. 2. Description of the data model at the data level. Subject area: university students

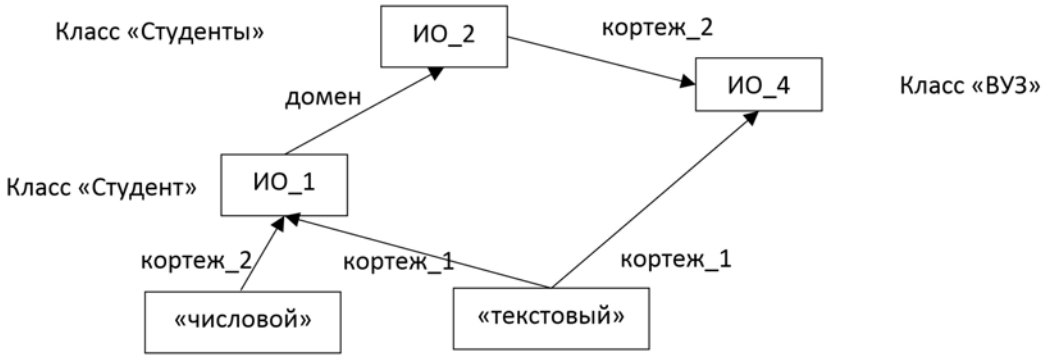

Рис. 3. Описание модели данных на уровне схемы. Предметная область: студенты вуза
Fig. 3. Description of the data model at the schema level. Subject area:university students

Важнейшим понятием объектной модели данных, характерным для любой ООСУБД, является идентификатор объекта. В нашем примере на рис. 1. ИО_1 и ИО_2 – идентификаторы двух объектов, моделирующих 2-х студентов; ИО_4 – идентификатор объекта, моделирующего вуз.

Над объектами в объектной модели данных определяются составные операции. На нашей схеме составные операции именованы русскоязычными аналогами обозначений функций в англоязычной литературе и языках программирования. Операция «домен» (set) создает множество однотипных объектов, как в нашем примере, результатом операции «домен» является новый объект с идентификатором ИО_3, являющийся набором двух студентов с идентификаторами ИО_1 и ИО_2.

Операция «кортеж» (tuple) создает кортеж в общем случае разнородных объектов, номер в обозначениях на схеме, следующий за словом «кортеж» означает, какой это по счету операнд операции составления кортежа.

База данных на рис. 1 представлена графом, вершины которого – объекты, а ребра – отношения вида «быть i-м операндом операции с данным результатом».

На рис. 3. Представлен граф уровня схемы для тех же объектов, что и на рис. 2. В ходе анализа текущего уровня развития ООБД, Эльдарханов приходит к выводам: «Ограничения на корректность объектов теории являются… внешними средствами по отношению к теории, в основном выраженными в форме указания конкретных классов некорректных объектов с обозначением пути разрешения проблемы… важной является также проблема реализации в данной модели сочетания функционального и логического программирования… Полной формализации включения функций и отношений в объектную модель пока не создано» [21].

Несмотря на наличие проблем применения объектно-ориентированного подхода к проектированию баз данных, объектно-ориентированные СУБД являются альтернативными средствами при решении такого класса задач, где применение реляционных СУБД обнаруживает ограниченность модели.

В англоязычной литературе встречается термин «The object-relational impedance mismatch», данный термин применяется для обозначения концептуальных и технических трудностей, возникающих при взаимодействии прикладных программ или комплексов прикладных программ, разработанных на объектно-ориентированных языках, с реляционными базами данных.

К одному из ряда подобных осложнений относится отсутствие возможности в реляционных СУБД работы с агрегатом значений атрибутов сущности как с единым целым экземпля-

ром сущности, моделирующим объект предметной области. Например, информацию об экземпляре сущности «Книга» можно получить из базы данных, схема которой представлена на рис. 4, не обратившись к целостному объекту, а собрав необходимые сведения из соответствующих таблиц посредством конструкции запросов на языке SQL с применением оператора «JOIN».

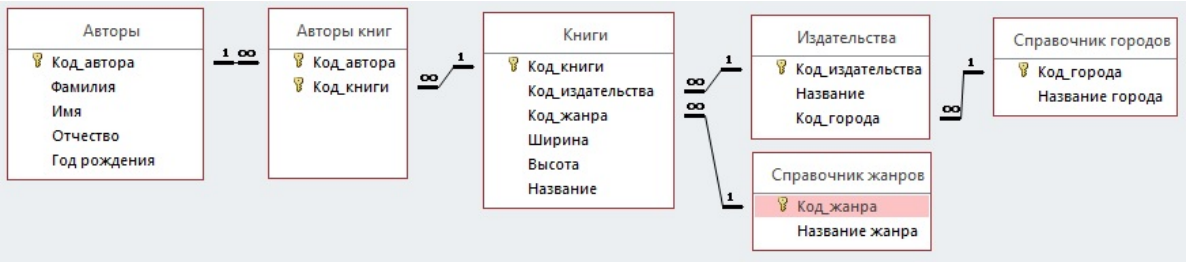

Рис. 4. Схема данных базы «Книги» в реляционной СУБД MS Access
Fig. 4. Scheme of the database “Books”, relational database management system MS Access

Конструкция запроса, которая может быть применена, представлена на листинге 3.

Листинг 3

```
SELECT Авторы.Фамилия, Авторы.Имя, Авторы.Отчество, Книги.Название, [Справочник
жанров].[Название жанра], Издательства.Название, [Справочник городов].[Название го-
рода] FROM [Справочник жанров] INNER JOIN ((([Справочник городов] INNER JOIN Из-
дательства ON [Справочник городов].Код_города = Издательства.Код_города) INNER
JOIN Книги ON Издательства.Код_издательства = Книги.Код_издательства) INNER JOIN
(Авторы INNER JOIN [Авторы книг] ON Авторы.Код_автора = [Авторы
книг].Код_автора) ON Книги.Код_книги = [Авторы книг].Код_книги) ON [Справочник
жанров].Код_жанра = Книги.Код_жанра;
```

Проблемы применения реляционных СУБД наиболее остро ощущаются в таких областях, как системы автоматизированного проектирования, автоматизированное производство, системы автоматизированного управления предприятием, системы, основанные на знаниях, мультимедийные системы.

ООБД позволяют объединить программный код и данные, избежать различий между представлениями информации в базе и прикладной программе, вследствие чего интерес к ним проявляют современные разработчики.

Среди современных объектно-ориентированных СУБД в первую очередь следует упомянуть о Caché [3], продукте компании InterSystems, существует и успешно развивается на протяжении 21-го года. ООСУБД Cache используется при проектировании автоматизированных систем управления здравоохранением, банковских и финансовых услуг, государственного управления и других секторов.

Вызывает интерес ConceptBase – дедуктивная и объектно-ориентированная система управления базами данных, разработанная в Университете Аахена и Университете Скуве. ConceptBase используется для концептуального моделирования и метамоделирования в области разработки программного обеспечения [24].

[3] Inter Systems: официальный сайт. URL: https://www.intersystems.com/products/cache/

В связи с активным развитием глобальных сетей и как следствие технологий разработки кросс-платформенных приложений возрос интерес к системе GemStone, в частности языку Smalltalk как инструменту создания JavaScript, реализующих задачи коммерческих приложений и веб-страниц. GemStone/S на сегодняшний день является одним из лучших примеров ООСУБД, которая хорошо подходит для масштабируемых, высокопроизводительных, многоуровневых распределенных систем [25].

Примером современной встраиваемой ООСУБД является ObjectDatabase++ (ODBPP), разработчик – Ekky Software [4]. В данной системе для хранения информации об экземпляре книги будет использоваться одна запись в противопоставление рассмотренному выше примеру структурирования данной информации в реляционной СУБД. Такой способ представления данных позволяет оперировать целым объектом и уменьшает необходимость объединения данных из разных таблиц; сокращает количество требуемых операций блокировки, чтения и записи, что способствует эффективной работоспособности системы при больших объемах данных.

ObjectDB – это объектная СУБД для Java. Она может использоваться в режиме клиент-сервер и во встроенном (в процессе) режиме. В то время как основные СУБД, представленные на рынке, предоставляют API (application program interface). Например, в состав MySQL входит библиотека, написанная на языке C, позволяющая обращаться к данным из любой программы, разработанной на этом языке [26]. ObjectDB не предоставляет собственный, для работы с этой СУБД используется один из двух стандартных API Java: Java Persistence API (JPA) или Java Data Objects (JDO). Спецификация JPA предоставляет возможность сохранять в удобном виде Java-объекты в базе данных [5]. Объекты данных Java (JDO) [27] представляют собой классы языка программирования Java; для них нет необходимости реализовывать определенные интерфейсы или расширяться из специальных классов. Оба API-интерфейса встроены в ObjectDB [6], поэтому промежуточное программное обеспечение ORM не требуется [7] [28].

ObjectDB является кросс-платформенной и может использоваться в различных операционных системах с Java SE 5 или выше. Он может быть интегрирован в веб-приложения Java EE и Spring и развернут в контейнерах сервлетов (Tomcat, Jetty), а также на серверах приложений Java EE (GlassFish, JBoss) [29]. Она была протестирована на различных JVM, включая HotSpot, JRockit и IBM J9 [8].

Максимальный размер базы данных составляет 128 ТБ (131 072 ГБ). Количество объектов в базе данных неограниченно (кроме размера базы данных).

Все устойчивые типы JPA и JDO поддерживаются ObjectDB, включая определяемые пользователем классы сущностей, определяемые пользователем встраиваемые классы, стандартные коллекции Java, базовые типы данных (примитивные значения, значения обертки, String, Date, Time, Timestamp) и любые другие сериализуемые классы.

Каждый объект в базе данных имеет уникальный идентификатор. ObjectDB поддерживает как традиционные идентификаторы базы данных объектов, так и механизмы реляционных баз данных, такие как первичные ключи, включая составные первичные ключи и создание и присвоение автоматического значения как часть поддержки JPA, которая в основном представляет собой API для РСУБД.

Поддерживаются два языка запросов. Язык запросов JDO (JDOQL), основанный на синтаксисе Java, и язык запросов JPA (JPQL), основанный на синтаксисе SQL. Также поддерживаются запросы по критериям JPA 2.

---

[4] Eccy Software. URL: http://www.ekkysoftware.com/

[5] Java Persistence API FAQ. Oracle Corporation. URL: https://www.oracle.com/technetwork/java/javaee/persistence-jsp-136066.html

[6] ObjectDB Software. URL: http://www.objectdb.com/

[7] JPA Performance Benchmark. URL: http://www.jpab.org

[8] ObjectDB. Object Database Features. URL: http://www.objectdb.com/object/db/database/features

Автоматическая эволюция схемы ObjectDB обрабатывает большинство изменений классов прозрачно, включая добавление и удаление постоянных полей, изменение типов постоянных полей и изменение иерархии классов. Также поддерживается переименование устойчивых классов и постоянных полей.

Современный рынок программного инструментария представлен рядом ООСУБД, но, если посмотреть на рейтинги используемых СУБД, как, например, результаты ежегодного опроса, проводимого сайтом «Stack Overflow» [9], где в 2018 г. приняли участие свыше 100 000 разработчиков программного обеспечения (см. таблицу), ни одну ООСУБД мы не встретим.

Самые популярные СУБД, по версии опроса «Stack Overflow»
The most popular DBMSs, according to the Stack Overflow survey

№ п/п	Название системы организации хранения данных	Количество опрошенных, применяющих данный инструментарий, %
1	MySQL	58,7
2	SQL Server	41,2
3	PostgreSQL	32,9
4	MongoDB	25,9
5	SQLite	19,7
6	Redis	18,0
7	Elasticsearch	14,1
8	MariaDB	13,4
9	Oracle	11,1
10	Microsoft Azure (Tables, CosmosDB, SQL, etc)	7,9
11	Google Cloud Storage	5,5
12	Memcached	5,5

В качестве итога можно привести слова Ричарда Энга: «Хотя ООСУБД и не получили широкого распространения, всё же объектные базы данных имеют свои нишевые рынки» [25].

Список литературы

1. **Mitchell G., Zdonik S. B.** Umeshwar Dayal. Optimization of Object-Oriented Queries: Problems and Approachers. *Advances in Object-Oriented Database Systems. Nato ASI Subseries F*, 2013, vol. 130, p. 119–146.
2. **Hudson S. E., King R.** The Efficient Support of Functionally-Defined Data in Cactis. In: Dittrich K. R., Dayal U., Buchmann A. P. On Object-Oriented Database Systems. Springer Science & Business Media, 2012, p. 341–355.
3. **Liskov B., Day M., Shrira L.** Distributed Object Management in Thor. In: Ozsuaet T. et al. (eds.). Distributed Object Management. San Mateo, CA, Morgan Kaufmann, 1994, p. 79–91.
4. **Thearle R. W.** A survey of object oriented database systems. In: Tagg R., Mabon J. (eds.). Object Management. Routledge, 2019.
5. **Кузнецов С. Д.** Три манифеста баз данных: ретроспектива и перспективы // Базы данных и информационные технологии XXI века: Докл. Междунар. науч. конф., 2003. URL: http://citforum.ru/database/articles/manifests/

[9] Developer Survey Results. 2018. URL:https://insights.stackoverflow.com/survey/2018/

6. **Stein J., Maier D.** Associative Support in GemStone. In: Dittrich K. R., Dayal U., Buchmann A. P. (eds.). On Object-Oriented Database Systems. Springer Science & Business Media, 2012, p. 323–338.
7. **Prabhu C. S. R.** Object-Oriented Database Systems: Approaches and Architectures. PHI Learning Pvt. Ltd., 2005, 232 p.
8. **Meuter W. de.** (ed.). Advances in Smalltalk: 14th International Smaltalk Conference, ISC 2006. Prague, Springer Science & Business Media, 2007, 158 p.
9. **Liu Ch.** Smalltalk, Objects, and Design. iUniverse, 2000, 312 p.
10. **Salmons J., Babitsky T.** International OOP Directory. COOT, Incorporated, 1992.
11. **D'Andrea A., Janus P.** UniSQL's next-generation object-relational database management system. *ACM SIGMOD Record*, 1996, vol. 25, no. 3 (September).
12. **Кузнецов С. Д.** Объектно-реляционные базы данных: прошедший этап или недооцененные возможности? URL: http://citforum.ru/database/articles/ordbms10/4.shtml
13. **Thearle R. W.** A survey of object oriented database systems. In: Tagg R., Mabon J. (eds.). Object Management. Routledge, 2019.
14. **Hugh D., Date C. J.** The Third Manifesto. *ACM SIGMOD Record*, 1995, vol. 24, no. 1 (March).
15. **Barry D. K., Duhl J.** Object Storage Fact Books: Object DBMSs and Object-Relational Mapping. Barry & Associates, Inc., 2001.
16. **Surhone L. M., Tennoe M. T., Henssonow S. F.** Object Query Language. URL: https://www.books.ru/books/object-query-language-1467950/ © Books.ru
17. **Галатенко В., Гвоздев А.** Типы и структуры данных в INFORMIX-Universal Server. Изд-во «Открытые системы», 1997. URL: https://www.osp.ru/news/articles/1997/0602/13031546
18. **Дейт К., Дарвен Х.** Основы будущих систем баз данных. Третий манифест. М.: Янус-К, 2004.
19. ANSI/X3/SPARC Study Group on Data Base Management Systems: Interim Report. FDT. *ACM SIGMOD bulletin*, 1975, vol. 7, no. 2.
20. **Jardine D. A.** The ANSI/SPARC DBMS Model. North-Holland Pub. Co., 1977.
21. **Эльдарханов А. М.** Обзор моделей данных объектно-ориентированных СУБД // Тр. ИСП РАН. 2011. Т. 21. С. 205–226.
22. **Beeri C.** A formal approach to object-oriented databases. *Data & Knowledge Engineering*, 1990, vol. 5, p. 353–382.
23. **Дейт К. Дж.** Введение в системы баз данных: Пер. с англ. 8-е изд. М.: Вильямс, 2005. 1328 с.: ил.
24. **Jeusfeld M.** ConceptBase.cc – A System for Metamodeling and Method Engineering. URL: http://conceptbase.sourceforge.net/
25. **Eng R. K.** How learning Smalltalk can improve your skills as a programmer. 2016. URL: https://medium.com/smalltalk-talk/how-learning-smalltalk-can-improve-your-skills-as-a-programmer-fc5b2f56685
26. **Дюбуа П.** MySQL. 3-е изд. М.: Вильямс, 2006. 344 с.
27. **Jordan D., Russell C**. Java Data Objects. 1st ed. O'Reilly Media, 2003, April 22, 384 p.
28. **Srinivasan K.** Create applications using ObjectDb and JPA in NetBeans. URL: http://www.javabeat.net/2011/02/create-applications-using-objectdb-and-jpa-in-netbeans
29. **Anghel L.** Integrate ObjectDB into Your JPA-based Java Web App. URL: http://www. developer.com/java/web/integrate-objectdb-into-your-jpa-based-java-web-app.html

References

1. **Mitchell G., Zdonik S. B.** Umeshwar Dayal. Optimization of Object-Oriented Queries: Problems and Approachers. *Advances in Object-Oriented Database Systems. Nato ASI Subseries F*, 2013, vol. 130, p. 119–146.

2. **Hudson S. E., King R.** The Efficient Support of Functionally-Defined Data in Cactis. In: Dittrich K. R., Dayal U., Buchmann A. P. On Object-Oriented Database Systems. Springer Science & Business Media, 2012, p. 341–355.
3. **Liskov B., Day M., Shrira L.** Distributed Object Management in Thor. In: Ozsuaet T. et al. (eds.). Distributed Object Management. San Mateo, CA, Morgan Kaufmann, 1994, p. 79–91.
4. **Thearle R. W.** A survey of object oriented database systems. In: Tagg R., Mabon J. (eds.). Object Management. Routledge, 2019.
5. **Kuznetsov S. D.** Three database manifest: retrospective and perspective. In: Databases and information technologies XXI century: report of the international scientific conference, 2003. URL: http://citforum.ru/database/articles/manifests/
6. **Stein J., Maier D.** Associative Support in GemStone. In: Dittrich K. R., Dayal U., Buchmann A. P. (eds.). On Object-Oriented Database Systems. Springer Science & Business Media, 2012, p. 323–338.
7. **Prabhu C. S. R.** Object-Oriented Database Systems: Approaches and Architectures. PHI Learning Pvt. Ltd., 2005, 232 p.
8. **Meuter W. de.** (ed.). Advances in Smalltalk: 14th International Smaltalk Conference, ISC 2006. Prague, Springer Science & Business Media, 2007, 158 p.
9. **Liu Ch.** Smalltalk, Objects, and Design. iUniverse, 2000, 312 p.
10. **Salmons J., Babitsky T.** International OOP Directory. COOT, Incorporated, 1992.
11. **D'Andrea A., Janus P.** UniSQL's next-generation object-relational database management system. *ACM SIGMOD Record*, 1996, vol. 25, no. 3 (September).
12. **Kuznetsov S. D.**Object-relational databases: past stage or underestimated opportunities? URL: http://citforum.ru/database/articles/ordbms10/4.shtml
13. **Thearle R. W.** A survey of object oriented database systems. In: Tagg R., Mabon J. (eds.). Object Management. Routledge, 2019.
14. **Hugh D., Date C. J.** The Third Manifesto. *ACM SIGMOD Record*, 1995, vol. 24, no. 1 (March).
15. **Barry D. K., Duhl J.** Object Storage Fact Books: Object DBMSs and Object-Relational Mapping. Barry & Associates, Inc., 2001.
16. **Surhone L. M., Tennoe M. T., Henssonow S. F.** Object Query Language. URL: https://www.books.ru/books/object-query-language-1467950
17. **Galatenko V., Gvozdev A.** Types and data structures in INFORMIX-Universal server.Open System Publishing House, 1997. URL: https://www.osp.ru/news/articles/1997/0602/ 13031546
18. **Date C. J., Hugh Darwen**. Databases, Types, and the Relational Model. The Third Manifesto. Addison Wesley; 3 edition (February 24, 2006).
19. ANSI/X3/SPARC Study Group on Data Base Management Systems: Interim Report. FDT. *ACM SIGMOD bulletin*, 1975, vol. 7, no. 2.
20. **Jardine D. A.** The ANSI/SPARC DBMS Model. North-Holland Pub. Co., 1977.
21. **Eldarkhanov A. M.** An Overview of Data Models of Object-Oriented DBMS. In: Proceedings of the Institute for System Programming RAS. 2011, vol. 21, p. 205–226.
22. **Beeri C.** A formal approach to object-oriented databases. *Data & Knowledge Engineering*, 1990, vol. 5, p. 353–382.
23. **Date C. J.** An Introduction to Database Systems. Publisher: Pearson; 8 ed. (August 1, 2003).
24. **Jeusfeld M.** ConceptBase.cc – A System for Metamodeling and Method Engineering. URL
25. **Jeusfeld M.** ConceptBase.cc – A System for Metamodeling and Method Engineering. URL: http://conceptbase.sourceforge.net/
26. **Eng R. K.** How learning Smalltalk can improve your skills as a programmer. 2016. URL: https://medium.com/smalltalk-talk/how-learning-smalltalk-can-improve-your-skills-as-a-programmer-fc5b2f56685
27. **Jordan D., Russell C**. Java Data Objects. 1st ed. O'Reilly Media, 2003, April 22, 384 p.
28. **Jordan D., Russell C**. Java Data Objects. 1st ed. O'Reilly Media, 2003, April 22, 384 p.

29. **Srinivasan K.** Create applications using ObjectDb and JPA in NetBeans. URL: http://www.javabeat.net/2011/02/create-applications-using-objectdb-and-jpa-in-netbeans
30. **Anghel L.** Integrate ObjectDB into Your JPA-based Java Web App. URL: http://www. developer.com/java/web/integrate-objectdb-into-your-jpa-based-java-web-app.html

Материал поступил в редколлегию
Received
21.05.2019

Сведения об авторе / Information about the Author

Яценко Елена Александровна, кандидат технических наук, доцент кафедры автоматизированных систем обработки информации и управления, Политехнический институт, Сургутский государственный университет (ул. Энергетиков, 22, Сургут, Россия)

Elena A. Yatsenko, Candidate of Science (Technology), Associate Professor of Automated information processing and management systems, Polytechnic Institute, Surgut State University (22 Energetikov Str., Surgut, Russian Federation)

elisia@yandex.ru
ORCID iD 0000-0002-3239-4409