

УДК 681.3.01: 004.652: 004.655: 004.432.4

В.И. Есин, М.В. Есина

Харьковский национальный университет им. В.Н. Каразина, Харьков

## ЯЗЫК ДЛЯ УНИВЕРСАЛЬНОЙ МОДЕЛИ ДАННЫХ

Описывается проблема создания формального языка, который позволял бы формировать данные и запросы с использованием естественного языка, и ее решение с помощью языка модели данных, который был разработан в рамках универсальной модели данных. Представлено описание основных операторов языка модели данных и примеры его использования.

**Ключевые слова:** модель данных, универсальная модель данных, база данных, язык модели данных.

### Введение

**Описание проблемы.** Проблема, связанная с созданием формального языка, который позволял бы формировать данные и запросы с использованием естественного языка, рассматривается в сообществе баз данных уже много лет [1, 2]: «Тридцать лет исследований в области языков запросов сводятся к тому, что "мы двигаемся от SQL к XQuery". В лучшем случае мы переходим от одного декларативного языка к другому, обладающему примерно тем же уровнем выразительности. Конечные пользователи не знают SQL, это язык профессиональных программистов. В родственных сообществах имеются идеи, которые могли бы повлиять на исследования в области интерфейсов к базам данных... наиболее интересные возможности связаны с исследованиями, ассоциируемыми с термином "semantic Web"... часть современных исследований сосредоточена на "онтологиях"... Работа над онтологиями может позволить пользователям баз данных задавать запросы в собственной терминологии с использованием естественного языка. Нужно учитывать эти возможности при разработке будущих СУБД». Но эта проблема до сих пор остается открытой. И простым заимствованием и подражанием другим системам в общем случае при поиске данных в БД не обойтись.

**Постановка задачи.** Прежде чем заимствовать некоторые методы поиска, присущие, например, некоторым информационно-поисковым системам, следует учитывать, что структура хранимой информации, например, в реляционных базах данных, существенно отличается от сплошного текста. И если в сплошных текстах бывает достаточно линейного просмотра в поисках ключевых слов, то организовать поиск в реляционной базе данных подобным образом невозможно. Доступ к реляционным базам предполагает знание пользователем модели базы данных и производится либо через заранее предопределенные формы, либо посредством языка запросов SQL. При этом необходимо сформировать синтаксически правильные и семантически осмысленные запросы (как

правило, на соответствующем SQL-диалекте) и проанализировать их релевантность [3]. В настоящее время данный процесс осуществляется программистами, так как конечному пользователю неизвестна модель данных, а ее изучение для составления соответствующего запроса требует значительного времени и квалификации, которые могут у него отсутствовать. Более того, квалификация программиста по работе с базами данных тоже должна быть достаточно высока, и он должен в совершенстве знать тонкости модели базы данных, ее физическую реализацию, возможности операционной системы и т.д.

Кроме того, сегодня существует еще и проблема, связанная с использованием программистами при разработке программных приложений для баз данных языков двух разных типов: языка программирования и языка запросов базы данных, так как процедурные языки и языки запросов баз данных основываются на разных семантике и стратегиях оптимизации [4]. Эта проблема получила название «потери соответствия» или несоответствия типов данных (*impedance mismatch*). Считается, что почти 30% трудозатрат программистов и объема кода расходуется на прямое и обратное преобразование данных из формата базы данных или формата файлов в формат, пригодный для внутреннего использования программой [5].

Вместе с тем, пользователь, являющийся специалистом в предметной области (ПрО), в состоянии сформулировать семантически правильный запрос с точностью до наименований элементов требуемой информации. Таким образом, возникает задача разработки методов автоматической трансформации семантически правильного запроса, составленного в терминах предметной области в синтаксически и терминологически корректный запрос к конкретной базе данных, которую можно разбить на две:

- согласования терминологических базисов исходного запроса и терминов, в которых описана структура реляционной базы данных;
- генерация исполняемого запроса к данным.

**Подход к решению проблемы.** Для решения этих задач необходимо иметь:

- универсальный механизм хранения структуры реляционной базы данных и терминов, в которых она описана;
- механизм преобразования пользовательского запроса в исполняемые запросы.

Если решение первой задачи можно осуществить благодаря схеме базы данных с универсальной моделью (УМД) [6, 7], которая позволяет унифицировать обращение к данным, то вторую задачу предстоит еще решить. Хотя та же БД с УМД способствует этому, так как она представляет собой набор стандартных (неизменных) отношений, которые используются для описания данных, связей между ними и ограничений, накладываемых на данные любой предметной области. Поэтому обращение к ее данным можно формализовать и унифицировать с помощью специального языка – языка модели данных (ЯМД).

### Язык модели данных

Язык модели данных – это специальный непроцедурный (декларативный) язык, основывающийся на метаонтологиях модели «объект-событие» [8], который позволяет определять данные и манипулировать ими в терминологии близкой к естественному языку. Он состоит из двух частей: языка определения (описания) данных и языка манипулирования данными, хранящимися в БД с УМД. Операторы языка определения данных используются для описания метаданных и данных предметной области. Как правило, результатом их выполнения, в случае физической реализации универсальной модели данных, является занесение в схему БД с УМД метаданных и данных соответствующих предметных областей. К метаданным ПрО (в соответствии с метаонтологиями модели «объект-событие»), относятся: разделы ПрО и их иерархии; классы объектов и их иерархии; классы событий и их иерархии; классы параметров объектов и их иерархии; типы объектов; значения паспортных характеристик типа объектов; типы характеристик экземпляров объектов, событий и параметров; типы значений параметров объектов; справочные (списочные) значения типов характеристик экземпляров объектов, событий и параметров; единицы измерения; папки документов. К данным ПрО относятся: экземпляры объектов, событий и их иерархии, значения характеристик экземпляров объектов, событий, конкретные документы и т.д. Операторы языка манипулирования данными позволяют: модифицировать, извлекать, удалять метаданные и данные предметной области, занесенные в БД с УМД, ограничивать права доступа и действий пользователей.

Язык модели данных, как любой другой формализованный язык, имеет свой синтаксис и семантику. Основными конструктивными элементами языка модели данных (языковыми конструкциями) являются его операторы. Оператор ЯМД представляет собой

служебное (ключевое, значимое) слово – как правило, в его качестве выступает одна из метаонтологий модели «объект-событие», заключенное в угловые скобки, после которого может следовать последовательность символов, заключенных в конструкцию [ ] ( , , ). После данной конструкции или сразу за угловыми скобками следует знак равенства. Знак равенства (=) может трактоваться и как оператор присвоения, который для указанной метаонтологии определяет ее значение, и как оператор равно (в операциях сравнения), в зависимости от смысла выполняемых действий. В конце оператора ЯМД ставится обязательный символ точка с запятой (;).

Основные служебные слова операторов языка описания данных и языка манипулирования данными приведены в табл. 1, 2 соответственно.

Таблица 1

Основные служебные слова операторов языка описания данных

Служебное слово*	Семантика
<Раздел>	Раздел предметной области.
<Папка>	Папка (директорий) документов.
<КлассО>, <КлассС>, <КлассПО>	Класс объектов, событий, параметров объектов соответственно.
<ЭкзО>	Экземпляр объекта определенного класса.
<ТипО>	Тип объекта, к которому относятся экземпляры объектов определенного класса.
<ВремяНС>, <ВремяКС>	Время начала и окончания события, которое произошло с конкретным экземпляром объекта.
<ВремяПО>	Время определения (ввода) значения параметра объекта.
<ТипЗнПО>	Тип значения параметра объекта.
<ТипХО> [ ] ( , , )	Тип характеристики экземпляра объекта: <ТипХО> [единица измерения – только для объекта с численной характеристикой] (тип данных: логическая; строковая; численная; дата, <u>признак нахождения данных в справочнике</u> : справочная (или списочная); любая, <u>тип характеристики</u> : паспортная; фактическая).
<ТипХС> [ ] ( , , )	Тип характеристики события: <ТипХС> [единица измерения – только для события с численной характеристикой] (тип данных: логическая; строковая; численная; дата, <u>признак нахождения данных в справочнике</u> : справочная; любая, <u>максимальное количество характеристик события с таким именем</u> : 1...N).
<ТипХПО> [ ] ( , , )	Тип характеристики параметра объекта: <ТипХПО> [единица измерения – только для параметров с численной характеристикой] (тип данных: логическая; строковая; численная; дата, <u>признак нахождения данных в справочнике</u> : справочная; любая, <u>интервал измерения значений параметра</u> : Н – час, D – день, М – месяц, Q – квартал, Y – год, нет интервала).

<ЗначХО>, <ЗначХС>, <ЗначХПО>	Значение характеристики экземпляра объекта, события, параметра объекта соответственно.
<Документ>(, .)	Документ, относящийся к объекту, событию, классу объекта или события, разделу и т. п., который может храниться в БД в виде ссылки или в виде BLOB объекта: <Документ> ( <u>признак ссылки: адрес (ссылка – URL), данные (BLOB объект), кому адресован: ОБЪЕКТ, СОБЫТИЕ</u> и т.д., <u>имя файла в файловой системе: любое имя, составленное из букв английского алфавита).</u>

Таблица 2

Основные служебные слова операторов языка манипулирования данными

Служебное слово*	Семантика
<Чтение>	Данные можно только считывать. Модификация, удаление, вставка данных запрещена.
<Интервал>	Признак того, что необходимо получить список событий, объектов, характеристик и т.д., связанных с интервалом времени от <ВремяНС> до <ВремяКС>.
<УдалитьР>, <УдалитьКлО>, <УдалитьО>, <УдалитьС>, ...	Оператор удаления: удалить раздел предметной области, удалить класс объектов, удалить экземпляр объекта определенного класса, удалить класс событий и т.д.
<ОбновитьКлО>, <ОбновитьЭкзО>, <ОбновитьКлС>, <ОбновитьДок>	Оператор обновления (изменения, модификации): обновить класс объекта, обновить экземпляр объекта, обновить класс события, обновить имя документа и т.д.

\* – имеется возможность определять служебные слова операторов языка в национальном алфавите (например: русский язык <КлассО> – английский язык <ClassО>; аналогично <УдалитьКлО> – <DeleteCLO> и т. д.).

Операторы языка объединяются в *строки метаописания*. Структура строки метаописания имеет следующий вид: {...} {...} ... {...}.

Внутри фигурных скобок {...} записываются операторы ЯМД. Конструкция, состоящая из одной фигурной скобки с заключенными в нее операторами называется *элементом строки метаописания*. Строка метаописания включает произвольное число элементов строки.

Операторы ЯМД, связанные с метаонтологиями модели «объект-событие»: *раздел, объект, событие, параметр объекта, документ* внутри элементов строки метаописания записываются между разделителями косая черта (слеш – /). Последовательность разделителей косая черта (/) между одноименными операторами языка, имеющими в своем составе служебные слова <Раздел>, <КлассО>, <КлассПО>, <ЭкзО>, определяет соответствующую иерархию разделов, классов объектов, классов параметров объектов, экземпляров объектов.

Примеры, демонстрирующие возможные формы записи иерархий в строках метаописания для метаданных и данных, достаточно подробно изложены в [9]. Ниже приведем пример использования строки метаописания ЯМД для формирования некоторых метаданных организационной структуры «Укртрансгаз»:

```
{<Раздел>=Укртрансгаз;/
 <КлассО>=УМГ;<ТунО>=-:/
 <КлассО>=ЛПУ;<ТунО>=-:/
 <КлассО>=КС;<ТунО>=-:/
 <КлассО>=Цех;<ТунО>=-:/
 <КлассО>=Агрегат;<ТунО>=ГПУ-10;
 <КлассО>=Агрегат;<ТунО>=ГТК-10I;
 <КлассО>=Агрегат;<ТунО>=ГПА-Ц-6,3С;}
```

Данная строка метаописания определяет организационную структуру – «Укртрансгаз»; классы объектов: «УМГ», «ЛПУ», «КС», «Цех», «Агрегат» и соответствующие для них типы: «-», «ГПУ-10», «ГТК-10I», «ГПА-Ц-6,3С»; иерархию подчиненности классов объектов (класс объектов «Агрегат» подчинен классу объектов «Цех», класс объектов «Цех» подчинен классу объектов «КС», классу объектов «КС» подчинен классу объектов «ЛПУ», который, в свою очередь, подчинен классу объектов самого высшего уровня (без владельцев) – «УМГ»).

Если проанализировать такое представление информации, то можно прийти к выводу, что строки метаописания ЯМД фактически являются формализованными предложениями естественного языка. И такое представление информации способны освоить и понимать не только программисты или специалисты информационных технологий (ИТ), но и специалисты предметных областей и простые пользователи. При этом главным является то, что им не надо знать модель базы данных (то есть ее схему).

Следуя всем правилам языка, можно составить строки метаописаний любой предметной области, которые позволяют записывать, считывать, модифицировать, удалять данные, хранящиеся в БД с УМД, местоположение (адрес) и значения которых задаются этой строкой метаописания.

Кроме того, в ЯМД имеется механизм, позволяющий избегать использования в приложениях SQL-запросов, не только при формировании адреса конкретного элемента ПрО для его чтения, изменения, записи, удаления, но и при получении требуемых списков элементов или последнего элемента списка. С этой целью используются специальные комбинации символов ЯМД, такие как: \*\*, \*\*\*, \*\*?, \*\*К, \*\*К?, \*\*К?. Например, используя следующую строку метаописания: {<КлассО>=\*\*.\*\*\*;}, можно получить список всех классов объектов, имеющих в БД с УМД, независимо от уровня их иерархии.

Рассмотрим пример получения списка водителей, которые работали в период с «08-окт-2007 08:15:00» по «11-окт-2007 18:15:00». Данный список можно достаточно просто получить, используя следующую строку метаописания ЯМД:

```
{<Раздел>=Хар_УГЭС; / <КлассО>=Водитель; <ТипО>=*.*; <ЭкзО>=*.?*; <КлассС>=Работа водителя; <ВремяНС>=08-окт-2007 08:15:00; <ВремяКС>=11-окт-2007 18:15:00; <Интервал>=;};
```

Если же простой пользователь попытается для этого примера составить SQL-запрос (используя оператор SELECT), то для него это будет не такая уже простая задача, так как данные находятся в различных таблицах, и для выполнения операции соединения необходимо обращаться к нескольким таблицам, для чего нужно хорошо знать структуру модели схемы БД с УМД, а она не такая уж простая. Конечно, можно воспользоваться заранее составленными наборами представлений. Но все равно, для этого, во-первых, потребуется квалифицированный программист, а не обычный пользователь, во-вторых, таких представлений может быть достаточно много (в зависимости от рассматриваемой ПрО), а, в-третьих, семантика таких запросов будет практически отсутствовать.

Операторы ЯМД могут также использоваться: при реализации механизма распределения прав доступа к данным вплоть до любого конкретного элемента данных; при конвертации данных из БД других информационных систем, построенных на различных платформах (таких как Oracle, PostgreSQL, Access, Lotus Notes), в БД с УМД; при создании отчетных документов, которые могут формироваться, как с помощью OLAP-технологии, так и с помощью иных средств; в качестве данных специального журнала измененных данных схемы БД с УМД и в некоторых других случаях.

При разработке языка модели данных большое внимание было обращено на его программную и документальную поддержку, так как без этого язык не смог бы существовать и развиваться.

Документальная и программная поддержка ЯМД представляет собой пакет соответствующих программ, библиотек, описаний и методик. А именно: подробное описание синтаксиса и семантики языка; интерпретатор языка для СУБД Oracle, реализованный на языке PL/SQL (планируется разработка интерпретатора для СУБД PostgreSQL и некоторых других) и описание его интерфейса; программные средства формирования и автономной отладки строк метаописания ЯМД; методика составления строк метаописания для записи новых данных или проведения конвертации данных из БД других информационных систем, построенных на различных платформах (таких как Oracle, PostgreSQL, Access, Lotus Notes), в БД с УМД; программа конвертации данных; библиотеки стандартных программ и компонент для разработки приложений с использованием ЯМД в средах Borland Delphi, Borland C.

Сложности согласования типов между языками программирования и базами данных, как отмечалось выше, традиционно считаются ключевой причиной

потери соответствия. Использование же интерфейса интерпретатора языка модели данных в программных приложениях, работающих с БД с УМД, позволяет решить эту проблему, так как типы всех параметров (входных и выходных) хранимых процедур, которые являются основой интерпретатора ЯМД, соответствуют типам параметров современных языков программирования и не требуют их дополнительного преобразования.

Кроме того, интерпретатор ЯМД потенциально обеспечивает лучшую производительность исполняемых запросов, так как он "пишется" и тестируется высококвалифицированными специалистами, хорошо понимающими структуру схемы БД с УМД, и лишен ошибок, свойственных прикладным программистам, не имеющим достаточного опыта в разработке приложений, особенно при работе в многопользовательском режиме с БД. К тому же код интерпретатора, представляющего собой набор серверных процедур, функций и пакетов, компилируется заранее, а это значит, что выполняться он будет также быстрее при правильном проведении так называемой статической оптимизации запросов, чем обычный неоптимизированный запрос.

## Выводы

1. Существующую сегодня проблему, связанную с созданием формального языка, позволяющего формировать данные и запросы к БД информационных систем с использованием естественного языка, можно считать частично решенной благодаря специальному языку модели данных, который был разработан в рамках УМД, основывающейся на модели данных «объект-событие».

2. Операторы языка манипулирования данными позволяют: модифицировать, извлекать, удалять метаданные и данные предметной области, занесенные в БД с УМД, а также ограничивать права доступа и действий пользователя.

3. Используя строки метаописания ЯМД, программисты могут больше внимания уделять продумыванию интерфейса и функциональности приложений, так как им не надо знать модель базы данных, тонкости и особенности стандартов языка SQL-запросов. Чтобы обратиться к БД с УМД программистам необходимо знать язык модели данных и структуру интерфейса доступа к серверным (хранимым) процедурам интерпретатора ЯМД. А специалисты предметной области, понимая операторы языка (так как последние достаточно просты, прозрачны и понятны), смогут более тесно взаимодействовать с программистами, ставя им конкретные задачи и согласовывая их действия на протяжении всего жизненного цикла эксплуатируемой информационной системы. При этом в определенной степени можно решить проблему нехватки высококвалифицированных программистов, разрабатывающих эффективные прило-

жения для работы с БД, так как многие сложные для программистов вопросы решаются благодаря возможностям, реализованным в интерпретаторе ЯМД.

### Список литературы

1. Кузнецов С.Д. Крупные проблемы и текущие задачи исследований в области баз данных [Электронный ресурс] / С.Д. Кузнецов. – Режим доступа к ресурсу: <http://www.citforum.ru/database/articles/problems/>.
2. Кузнецов С.Д. Предвестники новых манифестов управления данными [Электронный ресурс] / С.Д. Кузнецов. – Режим доступа к ресурсу: <http://www.citforum.ru/database/articles/premanifest/>.
3. Биряльцев Е.В. Онтологии реляционных баз данных. Лингвистический аспект [Электронный ресурс] / Е.В. Биряльцев, А.М. Гусенков. – Режим доступа к ресурсу: <http://www.dialog-21.ru/dialog2007/materials/html/08.htm>.
4. Кук В. Интеграция языков программирования с базами данных: в чем состоит проблема? [Электронный ресурс] / В. Кук, А. Ибрагим; Перевод: С. Кузнецов. – Режим доступа к ресурсу: [http://citforum.ru/database/articles/impedance\\_mismatch/](http://citforum.ru/database/articles/impedance_mismatch/).
5. Коннолли Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е издание: пер. с англ. / Т. Коннолли, К. Бегг. – М.: Издательский дом "Вильямс", 2003. – 1440 с.
6. Есин В.И. Технология проектирования модели предприятия на основе универсальной модели данных, 2008 [Электронный ресурс] / В.И. Есин, Ю.А. Пергаменцев. – Режим доступа к ресурсу: <http://www.citforum.ru/database/articles/udm/>.
7. Есин В.И. Универсальная модель данных и ее математические основы / В.И. Есин // Системы обработки информации. – Х.: Харьковський університет Повітряних Сил, 2011. – Вип. 2(92). – С. 21-24.
8. Есин В.И. Семантическая модель данных "объект-событие" / В.И. Есин // Вісник Харківського національного університету. – Х.: Харьковский национальный университет им. В.Н. Каразина, 2010. – № 925. – С. 65-73. – (Серия: Математичне моделювання. Інформаційні технології. Автоматизовані системи управління"; вип. 14).
9. Есин В.И. Язык описания и манипулирования данными, хранящимися в БД с УМД / В.И. Есин, М.В. Есина // Компьютерное моделирование в наукоемких технологиях (КМНТ-2010): междунар. науч.-техн. конф., 18-21 мая 2010 г. : тезисы докл. – Х.: Харьковский национальный университет им. В.Н. Каразина, 2010. – Часть 2. – С. 104-108.

Поступила в редколлегию 4.07.2011

Рецензент: д-р техн. наук, проф. Л.С. Сорока, Харьковский национальный университет им. В.Н. Каразина, Харьков.

### МОВА ДЛЯ УНІВЕРСАЛЬНОЇ МОДЕЛІ ДАНИХ

В.І. Єсін, М.В. Єсіна

Описується проблема створення формальної мови, яка дозволяла б формувати дані і запити з використанням природної мови, і її рішення за допомогою мови моделі даних, яка була розроблена в рамках універсальної моделі даних. Представлений опис основних операторів мови моделі даних і приклади його використання.

**Ключові слова:** модель даних, універсальна модель даних, база даних, мова моделі даних.

### LANGUAGE FOR UNIVERSAL DATA MODEL

V.I. Yesin, M.V. Yesina

The problem of formal language creation which would allow data forming and queries with the use of human language is described. Its decision is presented by means of data model language, which was developed within the framework of universal data model. Description of basic operators of data model language and examples of his use is presented.

**Keywords:** data model, universal data model, database, language of data model.