

# КОНЦЕПЦИЯ РАЗРАБОТКИ МАТРИЧНОЙ УНИВЕРСАЛЬНОЙ БАЗЫ ДАННЫХ С ПОДДЕРЖКОЙ ДРЕВОВИДНОЙ СТРУКТУРЫ ЕДИНИЦЫ ИНФОРМАЦИИ И ЕЕ УНИВЕРСАЛЬНОГО ПРИЛОЖЕНИЯ

И. А. Микляев

*Филиал «Севмашвтуз» ГОУ ВПО «Санкт-Петербургский государственный  
морской технический университет» в г. Северодвинске*

Поступила в редакцию 14.10.2010 г.

**Аннотация:** Матричная универсальная база данных может применяться в проектировании информационных систем с целью унификации инструментария для структурирования и обработки информации в различных предметных областях, создания приложений, легко перенастраивающихся при изменениях в бизнес среде без модификации структуры базы данных и программного обеспечения.

В Универсально приложение реализован универсальный инструментарий для работы со всеми матричными универсальными базами данных.

Программа рассчитана на самый широкий спектр пользователей. В зависимости от уровня знаний пользователь может начать с самого простого: ввода и работы с табличной информацией в стандартном виде, не отвлекаясь на дополнительную информацию, а далее по мере развития углубляться до комплексного управления сразу множеством таблиц, характеристик и значений.

**Ключевые слова:** проектирование систем управления баз данных, унификация инструментария для структурирования информации, универсальный тип данных, универсальная база данных

**Annotation:** Matrix universal database is intended for the informational systems design based on the using structuring and processing data unified tools in the different application domains, for the engineering easy update software in the variable business environment without data base structure and software update.

In universal application is marketed universal toolbox for working with all matrix universal databases.

Program is calculated on the most broad spectrum of users. Depending on the level of knowledge's an user can begin with simplest: entering and work with tabular information in the standard type, not distracting to additional information, but hereinafter on the measure of development to deepen before complex control of immediately ensemble of tables, features and values.

Besides, exhibit is presented and in the manner of the compiled module, which program product developer, can comprise of its under development project.

**Keywords:** designing the control systems of databases, unification a toolbox for structuring information, universal type of data, universal database

## ВВЕДЕНИЕ

В настоящее время реляционные СУБД являются доминирующим типом систем управления базами данных. Рынок объектно-ориентированных СУБД (ОСУБД) составляет лишь 3% от всего рынка баз данных [1].

Многие компании-разработчики реляционных СУБД прекрасно осознают потенциальную угрозу и конкуренцию со стороны ОСУБД. Они признают, что их системы в настоящее время не

подходят для создания сложных специализированных приложений, и что требуются дополнительные функциональные средства [1].

Три ведущих компании в области разработки реляционных СУБД, а именно: Oracle, Informix и IBM, расширили свои системы до уровня объектно-реляционных СУБД (ОРСУБД) [1].

Современные концепции формирования ОРСУБД

Основная тенденция расширения реляционных СУБД состоит в использовании в описа-

нии полей объектного типа данных. Для реализации которых, образуется объектно-ориентированная среда описания классов.

Например, в СУБД Postgres [1] реляционная модель дополнена за счет включения следующих механизмов:

- абстрактные типы данных;
- данные типа “процедура”;
- правила.

Одним из главных недостатков и ООСУБД и ОРСУБД приводится отсутствие универсальности [4].

Концепция создания СУБД матричной универсальной базы данных

Концепция создания СУБД матричной универсальной базы данных основывается на формировании многомерной матрицы способной вмещать в себе всю структуру и содержание любой реляционной базы данных и на разработке системы реализации некоторого функционала объектно-ориентированных СУБД. Главным отличием от существующих ОРСУБД является то, что объектом является не тип, определённый в дополнительной среде формирования классов, и предназначенный для описания полей таблицы, а сама таблица (сущность). Таблица (сущность) по реляционным законам может быть описанием поля другой таблицы, выступая в роли справочника, или в понятии объектно-ориентированных систем – агрегация. Также реализовано наследование от таблицы (сущности) к таблице и другие объектно-ориентированные свойства.

### МАТРИЧНАЯ УНИВЕРСАЛЬНАЯ БАЗА ДАННЫХ (МУБД)

Наименование МУБД сформировано из следующих представлений.

Использование в наименовании понятия матрицы больше диктуется не строгим научным термином, а психологическим воздействием на пользователя понятия тензор, т.к. пользователями системы уже являются и люди очень далёкие от математических наук. Далее будет представлено, что всё информационное поле располагается в семимерном динамическом массиве, и если не углубляться далее единицы информации, то используется пятимерная матрица, или более правильно пятимерный тензор.

Доказательство универсальности достаточно сложно, но в основу теории МУБД было заложено

структура теории баз данных, которая и реализована в логической модели универсальной базы данных [2].

Логическая модель универсальной БД (УБД).

В первую очередь при формировании универсальной информационной модели необходимо сформировать универсальную логическую модель БД.

Для создания универсальной базы данных необходимо определить, что характерно для любой предметной области. Как правило, объектная декомпозиция бизнес среды предполагает получение ответов на три основных вопроса:

1. Что необходимо описать?
2. Что необходимо приписать?
3. Какие значения принимает то, что будет описано?

В стандартных реляционных СУБД эти вопросы устанавливаются при формировании кортежа (записи таблицы). Первый вопрос – это определение таблицы и строки таблицы, в которую вносится информация. Второй вопрос определяет поле, которому приписывается значение – вопрос третий.

Ответы на первые два вопроса обязательны, поскольку объекты предметной области (сущности) характеризуются свойствами (атрибутами). Третий вопрос не всегда актуален в связи с тем, что свойства могут иметь как количественные, так и качественные оценки, или не иметь их вовсе, но возможность установления значения для некоторого свойства должна быть предусмотрена. В стандартных реляционных СУБД вопрос наличия атрибута в кортеже (записи) сводится к значению «Null», что в МУБД определяется лишь отсутствием атрибута (характеристики) для экземпляра сущности (строки таблицы).

Следующий вопрос расширил элемент кортежа с двух составляющих (имя атрибута и его значения) до четырёх.

Для учета того, что все развивается во времени и некоторые свойства могут иметь место только в определенный момент времени, целесообразно поставить вопрос:

4. В какой период действует данная информация?

Ответ на последний вопрос может быть реализован в универсальной базе данных и с помощью предшествующих вопросов через две информационные единицы. Тогда какому-либо



цы», это позволяет формировать сложные многострочные единицы информации, сохраняя атомарность значения. Таким образом, устанавливается возможность создавать неограниченные многострочные предложения для изложения законченной информации.

Так же в атрибутах сущности «Характеристика экземпляра сущности» стоит отдельно отметить пятый атрибут «Номер родительского элемента», который позволяет установить древовидную структуру единицы информации.

### УНИВЕРСАЛЬНЫЙ ТИП ДАННЫХ

Основополагающим шагом в создании СУБД МУБД является разработка универсального типа данных. В БД типа XML таким типом является строчный, что легко читается и обрабатывается, но очень громоздкий. Для работы с большими объёмами данных необходим универсальный тип, не уступающий стандартным типам.

В данной работе универсальный тип данных разработан на основе понижения системы счисления символов в поле базы данных, основанного на статистической информации об их использовании.

При сохранении информации формируется ключ поля, состоящий из последовательности символов задействованных в поле. Таким образом, получается меньшая кратность для хранения информации, соответствующая размеру ключа плюс 1.

Универсальное приложение для работы с МУБД

Универсальное приложение для работы с МУБД представляет собой всего две формы: форма администрирования МУБД и рабочая форма.

Форма администрирования МУБД предназначена для управления метаданными. Главное отличие МУБД от всех ныне существующих типов БД в том, что метаданные находятся в первых трёх таблицах МУБД, т.е. сами представляют собой содержание этих таблиц.

Первая таблица содержит наименования таблиц, их поля (в МУБД они называются «Допустимые характеристики») и описание полей, связей и самой таблицы (сущности).

Вторая таблица содержит параметры служебных атрибутов, которые предустановлены в СУБД МУБД и обрабатываются ею в соответствии с установленными правилами.

Третья таблица содержит параметры пользовательских атрибутов и формирования доменов с дополнительным описанием. Эта таблица заполняется автоматически по мере формирования пользовательских таблиц и автоматического заполнения списка значений доменов. Она открыта пользователю и может быть им отредактирована.

Первая таблица редактируется в самой административной форме, вторая пользователю не доступна для редактирования.

Форма администрирования МУБД очень похожа на рабочую форму, поэтому далее представлено описание только последней. В принципе, можно было бы с таким же успехом обойтись и одной формой.

Рабочая форма универсального приложения МУБД

Информация таблиц отображается в трёх видах (рис. 3):

- табличное отображение (обычное реляционное представление),
- отображение информации экземпляра (строки) таблицы во вспомогательной таблице (объектно-ориентированное представление информации),
- отображение информации экземпляра (строки) таблицы в древовидной форме (объектно-ориентированное представление информации с поддержкой древовидной формы единицы информации)

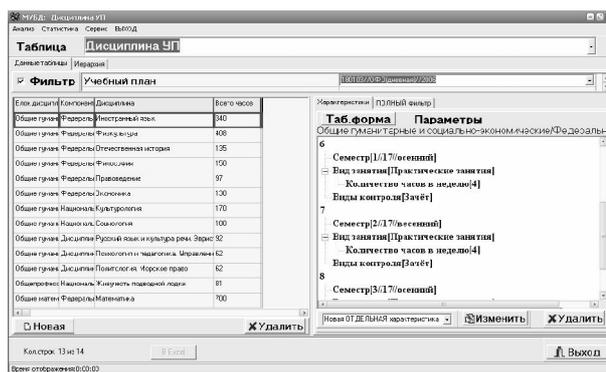


Рис. 3. Рабочая форма универсального приложения МУБД

На рис. 3 отображен пример таблицы «Дисциплина УП» из проекта информационной системы формирования основного учебного процесса в ВУЗе.

Левая таблица имеет стандартный реляционный вид. В правой таблице представлена в

древовидной форме полная информация о выбранной дисциплине учебного плана.

При реализации хранения информации в реляционной БД требуется 18 таблиц, в МУБД помимо трёх выше перечисленных таблиц с метаданными требуется 3 таблицы: «Учебный план», «Дисциплина УП» и «Семестр». Эти таблицы представляют собой реальные сущности, обладающие самостоятельным характером существования в реальности.

### НАСЛЕДОВАНИЕ

В МУБД реализован механизм наследования параметров сущностей, т.е. при описании таблицы, достаточно указать отдельной строкой параметр «Имеет характеристики сущности» и текущая таблица будет обладать всеми полями родительской сущности. После этого можно добавлять ей присущие поля (характеристики).

Так как метаданные это обычные данные первых трёх таблиц, то механизм наследования параметров полностью доступен и на уровне содержания таблиц пользователя, это достаточно эффективный инструмент.

Например, при регистрации параметров физических экспериментов, где каждый последующий отличается от предыдущего лишь на одно значение (обычная ситуация). Тогда в реляционных БД нет возможности сгруппировать с помощью дополнительной справочной таблицы, т.к. все эксперименты уникальны и повторяющихся групп атрибутов для всех экспериментов нет. В МУБД достаточно указать параметр «Имеет характеристики сущности» со значением – ссылкой на родительскую строку таблицы, и изменить тот параметр, который варьируется. В этом случае пользователь видит полный список параметров эксперимента, а в базе данных хранится только ссылка на родительскую строку и измененный параметр. Имея, например, 10 параметров у эксперимента, получаем пятикратную экономию.

### АГРЕГАЦИЯ

В МУБД реализован механизм агрегации сущностей. Если приписать параметр «Имеет характеристики сущности» дополнительно к полю (характеристике), то поле будет иметь тип записи, и само состоять из нескольких полей.

Например: Если в таблице «Координата» имеются поля «X», «Y» и «Z», а полю «Правая

верхняя контрольная точка» таблицы «Изображение эксперимента» указать дополнительную строку с параметром «Имеет характеристики сущности» со значением «Координата», то поле «Правая верхняя контрольная точка» будет иметь сложную структуру с полями «X», «Y» и «Z». Далее каждый элемент этой структуры отображается отдельно и имеет вид соответственно: «Правая верхняя контрольная точка ->X», «Правая верхняя контрольная точка ->Y» и «Правая верхняя контрольная точка ->Z».

### ИНТЕГРАЦИЯ ОДНОРОДНЫХ СУЩНОСТЕЙ С РАЗНОЙ СТРУКТУРОЙ

Объединение метаданных МУБД и самих данных открывает возможность устанавливать взаимосвязь между ними.

Например, в проекте документооборота учитывается большое количество типов документов, но даже документы, относящиеся к одной категории, могут значительно отличаться по составу полей, необходимых для их описания. Мы не будем углубляться в теорию документооборота, приведём пример на очевидных видах документов.

В проекте документооборота малого производственного предприятия все документы расположены в одной таблице, в их число входят: чертежи, приказы, письма и др. Понятно, что структура их описания разная. В МУБД после дополнительного описания у поля (например, «Категория») параметра «Управление табличным отображением», у остальных полей можно указать, для каких значений управляющего поля они доступны. Таким образом, для чертежей свой состав полей, а для приказов свой.

Состав управляющих полей не ограничен, и комбинация управляющих полей для выводимого поля также может быть составлена любая.

Имея древовидную структуру описания информации, появилась возможность адресного выделения и описания значений.

Например, характеристики «Доступ», «Цвет» и «Шрифт» могут быть приписаны, как к таблице в целом, так и к отдельной строке, отдельному полю (характеристике), и даже к отдельной ячейке, без какого-либо программного вмешательства. Т.е. можем, давая доступ к конкретной строке в целом, закрыть его к отдельной ячейке из неё.

## РЕКУРСИВНАЯ ОБРАБОТКА ИНФОРМАЦИИ

Особо стоит отметить открывшуюся возможность рекурсивной обработке информации. Рекурсия возможна в однородных структурах.

К одним из важных недостатков объектно-ориентированных баз данных относится отсутствие возможности рекурсивной обработки [4]. В реляционных базах данных это возможно внутри таблицы, но вариация таблиц и полей рекурсивно не возможна.

В МУБД однородность структуры расширена до всей базы данных, все таблицы и поля идентифицируются номерами, поэтому рекурсивные запросы на любую глубину иерархии таблиц достаточно просты и быстры.

В универсальном приложении для МУБД реализован комплексный ввод информации, т.е. при занесении адреса в одной древовидной структуре можно внести и улицу, и населённый пункт, и район, и страну, одновременно сразу в соответствующие нормализации данных таблицы для этой информации. Система комплексного ввода информации автоматически настраивается на иерархию всех таблиц связанных с текущей, и даёт возможность пользователю работать со всеми этими таблицами в одной древовидной структуре.

Рекурсивно реализована и фильтрация информации таблицы. Т.е. информация может быть отфильтрована по значению справочной таблицы, на которую у этой таблицы есть ссылка или есть ссылка у справочной таблицы, до любой глубины иерархии таблиц.

Например, могут быть отфильтрованы все адреса принадлежащие какому-либо району, минуя справочник улиц и населённых пунктов, без образования дополнительных полей.

## РАЗРАБОТКА ПРОГРАММНЫХ ПРОДУКТОВ НА ОСНОВЕ УНИВЕРСАЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ РАБОТЫ С МУБД

Главное направление последних разработок – это организация доступности к информации МУБД из других программных продуктов, как стандартными способами, так и более эффективными не стандартными методами.

Для реализации стандартных методов доступа к информации был реализован ретранслятор SQL запросов, который внедрён в класс объек-

тов, устанавливаемый в пользовательское приложение, и на сервер МУБД.

К нестандартным способам можно отнести возможность включения распространяемых модулей для работы с МУБД, которые просто включаются в проект.

Создание универсального приложения МУБД открывает более широкий спектр взаимодействий разработчика СУБД для МУБД, программиста и пользователя.

Разработчик СУБД берет на себя все проблемы, связанные с организацией унифицированного ввода, изменения и удаления информации справочников системы. Разработчику приложения остается лишь создать экранные формы для работы с исходящей информацией проекта по персональным требованиям заказчика, которые, впрочем, могут редактироваться и в унифицированной форме, и реализовать функции системы. Пользователь же получает возможность корректировки структуры информации и без участия разработчика приложения рис. 4.



Рис. 4. Распределение функций участников разработки и функционирования информационной системы с использованием МУБД.

Таким образом, процесс разработки приложения значительно упрощается с точки зрения программиста, следовательно, большее время может быть уделено изучению предметной области. Важно учитывать, что при использовании универсальной базы данных тестовая информация для отладки системы может быть внесена на ранних этапах проектирования, без опасений утратить ее в результате поэтапной модификации структуры базы данных. Пользователь же за счет управления справочниками получает возможность самому адаптировать информационную систему к изменяющейся предметной области.

Этапы разработки информационных систем с использованием МУБД

Важным достижением создания МУБД является изменение иерархии традиционных основных этапов разработки информационных систем. К ним относятся [1]:

1. Анализ требований, включающий изучение предметной области, построение моделей бизнес процессов.
2. Проектирование, в том числе определение сущностей, их атрибутов и связей, разработка логической модели базы данных.
3. Выбор системы управления базами данных и физическая реализация проекта.
4. Разработка интерфейса и программной логики приложения.
5. Тестирование и отладка.
6. Внедрение и сопровождение.

Прохождение этапов происходит строго последовательно, ни один из этапов нельзя пропустить и перейти к следующему. После каждого этапа происходит анализ соответствия полученных результатов требованиям заказчика. При значительных отклонениях разработчик вынужден возвращаться к предшествующим этапам и проходить последующие шаги заново (рис. 5.а).

Ввиду того, что этап «Изучение предметной области» является базовым, к нему приходится возвращаться наиболее часто, что приводит к значительному увеличению трудоемкости, затрат времени и ресурсов. Учитывая то, что все знать невозможно, а также то, что все со временем изменяется, получаем бесконечную задачу изучения предметной области, и, соответственно, весь последующий комплекс задач.

К сожалению, зачастую самые существенные коррективы разработчик вносит на последних этапах разработки интерфейса, тестирования и отладки приложения в реальных условиях. Это происходит, в основном, по двум причинам. Во-первых, разработчик программного обеспечения, как правило, не является участником исходного бизнес процесса, и, соответственно, познание нюансов предметной области происходит во время проектирования. Во-вторых, внедрение автоматизированных информационных систем предполагает изменение производственных процессов. Следовательно, на выходе мы имеем новую предметную область, с новыми участниками, обновленными функциями и задачами.

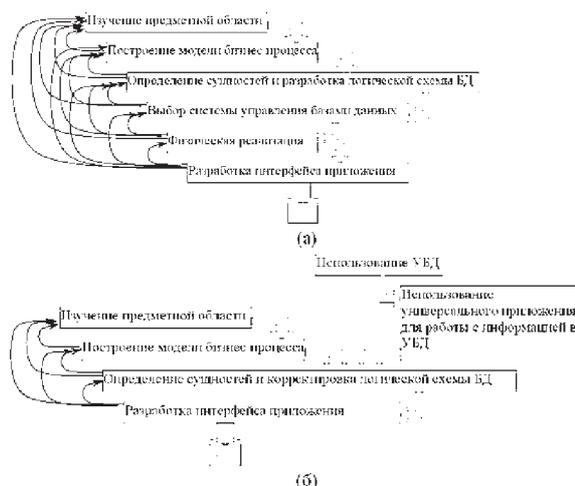


Рис. 5. Итерационная схема разработки ИС традиционная (а) и с использованием универсального приложения МУБД (б)

Одним из путей совершенствования технологии разработки информационных систем является исключение промежуточных этапов проектирования за счет создания универсальной базы данных (УБД), имеющей возможность без изменения структуры содержать в себе информацию любой предметной области.

Получим схему разработки информационных систем, состоящую из следующих этапов (рис. 5.б):

1. Анализ требований, включающий изучение предметной области, построение моделей бизнес процессов.
2. Проектирование, в том числе определение сущностей, их атрибутов и связей, корректировка логической модели базы данных.
3. Разработка интерфейса и программной логики приложения.
4. Тестирование и отладка.
5. Внедрение и сопровождение.

При создании информационного приложения обязательно необходимо пройти три этапа работы с информацией:

1. создание базы данных;
2. разработка форм для заполнения и отображения таблиц базы данных;
3. разработка форм отображения результата.

Основная цель любого проекта – это третий этап, на нём видно, что достигнуто, идёт согласование с заказчиком, на основании чего делаются выводы о корректировках в структуре информации, т.е. вносятся изменения на первом этапе. Это приводит к значительным исправле-

ниям и доработкам на втором тяжёлом рутинном этапе, что делает затраченные усилия и ресурсы на нём не эффективными.

При реализации системы с использованием универсального приложения МУБД, программист получил возможность создавать систему от результата. Т.е. он создаёт выходные формы, согласует результат с заказчиком, а затем наполняет начинку удобного для пользователя интерфейса для формирования базовой информации.

Таким образом, этапы получаем следующие:

1. создание базы данных;
2. заполнение базы данных через универсальное приложение;
3. разработка форм отображения результата;
4. разработка специализированных форм для внесения, изменения и отображения информации таблиц базы данных.

#### **АПРОБАЦИЯ УНИВЕРСАЛЬНОГО ПРИЛОЖЕНИЯ НА ОСНОВЕ МУБД**

Универсальное приложение реализовано в автоматизированной системе формирования основного учебного процесса высшего учебного заведения, в автоматизированной системе начисления заработной платы работников среднего и дошкольного образования под единым управлением, в системе управления малого производственного предприятия, в муници-

**Микляев И.А.** – зав. кафедрой, доцент Филиала СПбГМТУ «Севмашвтуз» в Северодвинске, кафедра «Прикладная математика и программирование». E-mail: ivanmia1@rambler.ru

пальном управлении и при поддержке крупных научных виброакустических исследований в кораблестроении.

#### **ЗАКЛЮЧЕНИЕ**

Переход от уже действующих баз данных на основу МУБД может быть осуществлён и автоматически. При этом первое и пятое измерение МУБД будут равны единице, а в шестом массиве будут использованы только нулевой и первый элементы, соответствующие наименованию атрибута и значению по этому атрибуту.

Верхнее измерение в массиве МУБД позволяет одновременную работу с несколькими базами данных и открывает возможности разработки распределённых баз данных.

Важным достоинством универсального приложения является лёгкость и простота, это может быть использовано и для бытового использования взамен программных продуктов Excel и Access при хранении структурированной информации.

#### **СПИСОК ЛИТЕРАТУРЫ**

1. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. Изд. 3-е: Пер. с англ. М.: Издательский дом «Вильямс», 2003. С.1440
2. Микляев И.А., Ундозерова А.Н., Кудяева М.В. Универсальная логическая модель базы данных. // Вестник Поморского Университета 1/2010. Сер. «Естественные науки» - С. 93-99.

**Miklyev I.A.** – assistant professor, managing pulpit Branch SPbSSU «Sevmashvtuz» in Severodvinsk, pulpit “Applied mathematics and programming”. E-mail: ivanmia1@rambler.ru