

УДК 004.652

В.И. Есин

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

## ВОЗМОЖНОСТИ РАЗЛИЧНЫХ СХЕМ БАЗ ДАННЫХ ПО ПРЕДСТАВЛЕНИЮ И РАБОТЕ С ДАННЫМИ

Описываются возможности схем баз данных, построенных в реляционной СУБД по традиционной технологии и с использованием универсальной модели, по представлению данных различных предметных областей и работе с ними. Раскрываются преимущества схемы базы данных с универсальной моделью данных.

**Ключевые слова:** универсальная модель данных, база данных, схема базы данных, база данных с универсальной моделью данных.

### Введение

Рассмотренная в работах [1, 2] универсальная модель данных (УМД) является инструментом моделирования логической модели предметной области (или логической схемы базы данных). В ней объектом моделирования является база данных (БД), точнее, ее логическая схема. Но так как УМД разрабатывалась не просто как инструмент моделирования логической модели конкретной предметной области (ПрО), а именно любой ПрО, то ее физическая реализация, точнее, реализация ее физической схемы, также представляет интерес, с точки зрения возможности ее применения для моделирования различных предметных областей.

Поэтому далее рассмотрим возможности физической схемы БД с УМД как инструмента моделирования любой ПрО на физическом уровне (в смысле построения физической схемы базы данных для любой предметной области) и сравним их с возможностями схем БД реляционных СУБД, построенных по традиционной технологии.

### Возможности различных схем баз данных

Если, используя уровни архитектуры *ANSI/SPARC* [3], попытаться представить данные различных предметных областей через абстрактное понимание содержимого баз данных, построенных по традиционной технологии и с использованием универсальной модели, то можно обнаружить принципиальные различия в таком представлении. А именно, учитывая то, что на *внешнем уровне* (представление базы данных с точки зрения пользователей) описывается та ее часть, которая относится к каждому пользователю, на *концептуальном уровне* (обобщающее представление базы данных) описывается то, какие данные хранятся в базе данных, а также связи, существующие между ними, и на *внутреннем уровне* (физическое представление базы данных в компьютере)

описывается, как информация хранится в базе данных, схематичное представление содержания уровней архитектуры *ANSI/SPARC* при традиционном построении БД для различных ПрО можно отобразить следующим образом (рис. 1).

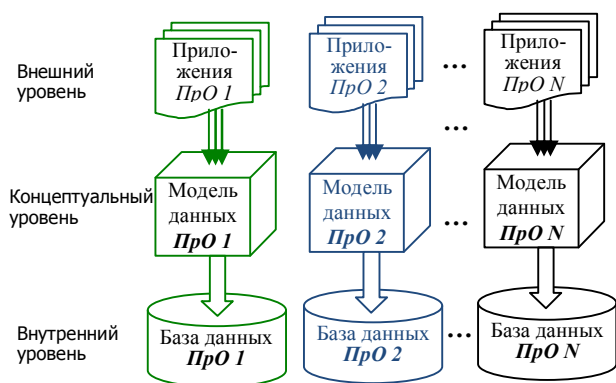


Рис. 1. Представления данных при традиционном построении схем БД

База же данных с УМД позволяет хранить данные различных предметных областей, что отражено в структуре таблиц ее схемы, которые были созданы на основе математических отношений, и в которых присутствует элементы множества условных идентификаторов предметных областей [2]. Поэтому схематичное представление данных различных предметных областей на трех уровнях архитектуры *ANSI/SPARC* в случае использования схемы БД с УМД будет выглядеть несколько иначе (рис. 2).

Как видно из этих рисунков, на концептуальном уровне архитектуры *ANSI/SPARC* при традиционном проектировании для описания  $N$  даталогических предметных областей требуется  $N$  различных моделей данных (для каждой ПрО), а в случае использования схемы БД с УМД, только одна – универсальная модель данных.

Аналогично и для внутреннего уровня. В первом случае требуется физическая реализация  $N$  схем

баз данных, а во втором – одна – схема БД с УМД. Для обеспечения такой возможности при реализации схемы БД с УМД был использован механизм, так называемых, виртуальных частных баз (VPD) [4]. В настоящее время схема БД с УМД реализована в СУБД Oracle и PostgreSQL.

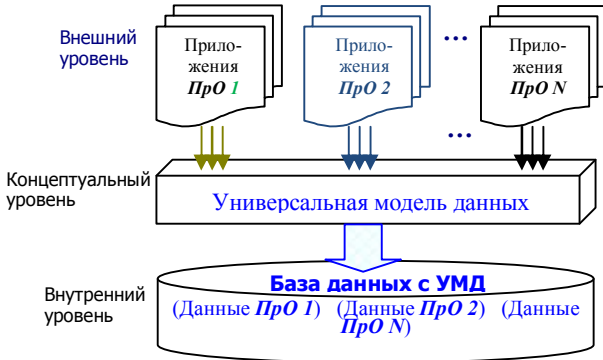


Рис. 2. Представления данных при использовании схемы БД с УМД

Для лучшего понимания структуры схемы БД с УМД и ее возможностей по представлению данных рассмотрим примеры экстензионалов (состояний) базы данных автотранспортного предприятия, реализованной в реляционной СУБД по традиционным технологиям и с использованием УМД. При этом конкретные связи между таблицами не приводятся, а отдельные атрибуты или опускаются, или видоизменяются (например, столбцы с так называемыми ID-номерами заменяются столбцами названий, которым они соответствуют – \*).

Пример экстензионала базы данных, построенной по традиционной технологии

Таблица «Техническое средство»

ID_T	Гаражный номер	Марка техники	Гос. номер	Год выпуска	Номер шасси	Цвет
00003	20001	Volvo	MA2299BV	2005	AC324516FD	зеленый
00005	20018	ГАЗ-31029	MA2200BV	2006	DS435678GF	белый

Таблица «Водитель»

ID_V	Водитель	Дата рождения	Табельный номер
00101	Иванов Иван Иванович	19.02.1955	10100
00104	Кравченко Сергей Владимирович	10.05.1972	10101

Таблица «Двигатель»

ID_D	Номер двигателя	Марка двигателя	Год выпуска
100001	GA123FC	VOL15i	2005
100002	HJ567KU	ЗМЗ-402.10	2006

Пример экстензионала базы данных, построенной на основе использования схемы БД с УМД

Таблица «Классов объектов»

OBJ CLASS ID	Класс объектов
0000201	Техническое средство
0000205	Водитель
0000209	Двигатель

Таблица «Тип объектов»

OBJ_TYPE_ID	Тип объекта	Класс объектов*
369081	Volvo	Техническое средство
369072	ГАЗ-31029	Техническое средство
370184	VOL15i	Двигатель
370180	ЗМЗ-402.10	Двигатель
369624	-	Водитель

Таблица «Экземпляров объектов»

OBJECT_ID	Тип объекта*	Класс объектов*	Экземпляр объекта
1329150	Volvo	Техническое средство	20001
1337718	ГАЗ-31029	Техническое средство	20018
1341332	-	Водитель	Иванов Иван Иванович
1337742	-	Водитель	Кравченко Сергей Владимирович
1337409	VOL15i	Двигатель	GA123FC
337411	ЗМЗ-402.10	Двигатель	HJ567KU

Таблица «Типов характеристик объектов»

PROP_T TYPE ID	Класс объектов*	Тип характеристики объекта	Тип значения
28907	Техническое средство	Год выпуска	дата
28977	Водитель	Дата рождения	дата
28970	Водитель	Табельный номер	строковый

Таблица «Значений типов характеристик объектов»

ID	Значение характеристики	Экземпляр объекта*	Тип характеристики объекта*
193033	2005	20001	Год выпуска
193072	2006	20018	Год выпуска
193077	2005	GA123FC	Год выпуска
193081	2006	HJ567KU	Год выпуска
193328	19.02.1955	Иванов Иван Иванович	Дата рождения
193385	10.05.1972	Кравченко Сергей Владимирович	Дата рождения
193244	10100	Иванов Иван Иванович	Табельный номер
193223	10101	Кравченко Сергей Владимирович	Табельный номер

Таблица «Классов событий»

EVENT_TYPE ID	Класс событий	Класс объектов*
10811	Регистрация	Техническое средство
10818	Установка на техническое средство	Двигатель

Таблица «Экземпляров событий»

EVENT_ID	Экземпляр объекта*	Класс события*	ВремяНС
561736	20001	Регистрация	20.01.2005 10:00:00
554850	20018	Регистрация	10.07.2006 09:00:00
552944	10818	Установка на техническое средство	25.06.2006 09:30:00

Таблица «Типов характеристик событий»

PROP_T YPE_ID	Класс события*	Тип характе- ристики события	Тип значения
7166	Регистрация	Гос. номер техники	строковый
7165	Регистрация	Номер шасси	строковый
7163	Регистрация	Цвет	строковый

Таблица «Значений типов характеристик событий»

ID	Значение ха- рактеристики	Тип характе- ристики события*	Класс собы- тия*	ВремяНС
579156	MA2299VB	Гос. номер	Регистрация	20.01.2005 10:00:00
579424	AC324516FD	Номер шасси	Регистрация	20.01.2005 10:00:00
577648	Зеленый	Цвет	Регистрация	20.01.2005 10:00:00
579139	MA2200VB	Гос. номер	Регистрация	10.07.2006 09:00:00
579577	DS435678GF	Номер шасси	Регистрация	10.07.2006 09:00:00
577640	Белый	Цвет	Регистрация	10.07.2006 09:00:00

На первый взгляд может показаться, что реализация БД с УМД сложнее, чем традиционная схема реляционной БД для конкретной ПрО. Но это только кажущаяся сложность, если рассматривать малое количество таблиц базы данных. Если количество таблиц большое, а это как раз обычный случай для баз данных корпоративных информационных систем, то в этом случае такое построение оправдано (в схеме БД с УМД количество таблиц всегда фиксированное и не такое большое – порядка 40), даже если не рассматривать связи между таблицами.

С другой стороны, представьте себе, что необходимо добавить какое-то новое отношение (таблицу), характеризующую новую сущность или в какую-то таблицу ввести дополнительные столбцы (атрибуты), а связи уже прописаны, приложения написаны, и база функционирует.

Что делать?

В лучшем случае можно создать новые таблицы с дополнительными столбцами (например, при вводе дополнительного атрибута «регистрация автомобиля») и перегрузить в них данные из старых таблиц. На это понадобится какое-то время, иногда значительное. Например, такая простая операция, как добавление к большой таблице нового столбца со значением по умолчанию, может занимать часы, поскольку данные организованы построчно (по крайней мере, в самых известных СУБД) и добавление данных к каждой строке требует переноса огромного объема данных, что подобно стихийному бедствию [5]. Но, в конечном счете, задача будет решена.

А что делать, если требуется учесть сложные зависимости, происходящие с объектом во времени?

Например, на транспортное средство может устанавливаться в различное время различных тип и экземпляр двигателя.

Или необходимо ввести значительное количество дополнительных таблиц, соответствующих различным сущностям, для моделируемой предметной области, которые требуют внесения изменений в существующие связи схемы БД. Простой перезагрузкой здесь не обойтись. Необходимо перепроектировать всю БД. О последствиях можно не говорить!

При использовании же БД с УМД со структурой схемы делать ничего не надо, какова бы ни была сложной и подвергающейся постоянным изменениям рассматриваемая предметная область.

Так для введения того же атрибута «регистрация автомобиля» достаточно просто ввести новую дату для события «Регистрация» (следует обратить внимание на то, что в традиционных моделях нет понятия «событие» как факта или действия, которое происходит с некоторым объектом в определенный момент или интервал времени, вообще). Более того, так как в реальности достаточно часто происходит процедура перерегистрации транспортного средства, и чтобы отследить данный процесс во времени и сохранить его в БД, построенной на традиционной основе, следует немало потрудиться. При использовании же БД с УМД, нет проблем, достаточно ввести новую дату события «Регистрация». А чтобы предусмотреть возможность замены двигателя на автомобиле, достаточно просто ввести новый класс события «Установка двигателя» и «связать» его с классом события «Установка на техническое средство» и соответствующим экземпляром технического средства и двигателя. При этом понятие «связать» означает не прописывание связи в схеме базы данных (структура схемы не меняется), а добавление недостающей записи данных в имеющиеся уже таблицы. То есть в БД с УМД достаточно просто включать любые новые объекты, события, которые с этими объектами происходят и т.д., и дальше продолжать работать. Нарращивание состава БД с УМД можно проводить в темпе их возникновения. При этом состав данных, включаемых в БД, не требует предварительного рассмотрения и определяется по мере развития и реализации функций информационной системы. А изменения, связанные с пониманием ПрО, обеспечением новой функциональности и т.д., которые очень часто и закономерно ставятся заказчиками перед разработчиками БД информационных систем, намного проще и быстрее внести в схему базы данных, построенную на основе УМД, чем в схему, построенную по традиционной технологии. Это объясняется тем, что для формирования базы данных конкретной ПрО используется терминология семантической модели данных «объект-событие», а не реляционной модели данных (отношение – таблица, атрибут – столбец, кортеж – строка и т.п.). Процесс проектирования осуществляется на концептуальном уровне, что позволяет разработ-

чикам, больше сосредоточиться на ПрО, а, следовательно, лучше продумать состав необходимых данных в соответствии с реализуемыми бизнес-процессами.

К тому же при изменении традиционных схем баз данных часто возникают трудности не только требующие внесения изменений в структуру базы данных, но и в программы, осуществляющие доступ к измененным структурам. Даже с использованием современных технологий этот процесс выполняется очень медленно и с большими трудозатратами. В результате большинство организаций вынуждено использовать существующие структуры баз данных.

В случае же использования схемы БД с УМД существующие приложения как работали, так дальше и будут работать, их переписывать нет необходимости.

Поэтому для удовлетворения требования повышенной гибкости целесообразно использовать именно схему БД с УМД, которая не препятствует естественному развитию системы в целом.

Кроме того, многие сегодня сталкиваются с проблемой сопровождения большого количества баз данных, которые создаются для каждой предметной области, что опять же ведет к ненужной сложности их администрирования и соответственно увеличению ее стоимости (стоимость администрирования различных баз данных доходит до десятков тысяч долларов в год [6]).

А имея одну БД с УМД, реализованную для нескольких предметных областей (рис. 2), можно существенно минимизировать затраты не только на проектирование баз данных для разных ПрО, но также и на системное сопровождение только одной СУБД, а не нескольких, на которых они могли бы быть реализованы.

## Выводы

1. Схема базы данных с универсальной моделью может использоваться для работы с данными любой предметной области, в отличие от схем баз данных, построенных по традиционной технологии

(которые проектируются каждый раз по-новому для конкретной предметной области).

2. Схема БД с УМД – это инструмент моделирования базы данных любой ПрО в реляционной СУБД, который призван облегчить труд разработчиков. При использовании схемы БД с УМД процесс проектирования базы данных любой ИС начинается не с разработки каждый раз новых объектов базы данных, а с инсталляции стандартной схемы БД со структурой УМД и дальнейшим ее формированием.

3. Инвариантность структуры схемы БД с УМД по отношению к различным предметным областям повышает ее открытость, упрощает процесс проектирования и формирования данных.

## Список литературы

1. Есин В.И. Универсальная модель данных и ее математические основы / В.И. Есин // Системи обробки інформації. – Х.: Харківський університет Повітряних Сил, 2011. – Вип. 2(92). – С. 21-24.
2. Есин В.И. Универсальная модель данных и ее отличительные особенности / В.И. Есин // Вісник Харківського національного університету ім. В.Н. Каразіна, 2011. – № 960. – С. 141-147 – (Серія: Математичне моделювання. Інформаційні технології. Автоматизовані системи управління"; вип. 16).
3. American National Standards Institute (1975). AN-SI/X3/SPARC Study Group on Data Base Management Systems. Interim Report, FDT. ACM SIGMOD Bulletin, 7(2).
4. Кайт Т. Oracle для профессионалов: пер. с англ. / Т. Кайт. – СПб.: ООО «ДиаСофтЮП», 2003. – 672 с.
5. Фаро С. Рефакторинг SQL-приложений: пер. с англ. / С. Фаро, Л. Паскаль. – СПб.: Символ-Плюс, 2009. – 336 с.
6. Dontcheff J. On the Total Cost of Database Administration [Electronic resource] / Julian Dontcheff. – Access mode : <http://juliandontcheff.wordpress.com/2011/03/25/on-the-total-cost-of-database-administration/>.

Поступила в редколлегию 8.02.2012

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

## МОЖЛИВОСТІ РІЗНИХ СХЕМ БАЗ ДАНИХ З ПРЕДСТАВЛЕННЯ ТА РОБОТИ З ДАНИМИ

В.І. Єсін

Описуються можливості схем баз даних, побудованих в реляційній СУБД за традиційною технологією і з використанням універсальної моделі, з уявлення даних різних наочних областей та роботи з ними. Розкриваються переваги схеми бази даних з універсальною моделлю даних.

**Ключові слова:** універсальна модель даних, база даних, схема бази даних, база даних з універсальною моделлю даних.

## POSSIBILITIES OF VARIOUS SCHEMES OF DATABASES ON PRESENTATION AND WORK WITH DATA

V.I. Yesin

Possibilities of databases schemes, built in relational DBMS on traditional technology and with the use of universal model, on presentation of data of various subject domains and work with them are described. Advantages of database scheme with the universal data model are described.

**Keywords:** universal data model, database, database scheme, database with the universal data model.