

# Інфокомунікаційні системи

УДК 004.652

В.И. Есин

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

## СХЕМА БАЗЫ ДАННЫХ С УНИВЕРСАЛЬНОЙ МОДЕЛЮ ДАННЫХ

Описываются факторы успешного проектирования, на основе которых разрабатывалась схема базы данных с универсальной моделью данных, сам процесс разработки, а также отличительные особенности полученной стандартной (универсальной) схемы базы данных (БД) для реляционной системы управления БД (СУБД), которую можно использовать для работы с данными любой предметной области.

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

### Введение

Разработка схемы любой базы данных достаточно сложный процесс, требующий тщательной проработки многих вопросов на всех ее стадиях. Ее итогом должно быть создание в разумные сроки и при разумных затратах приемлемой базы данных. Под приемлемой понимается такая база данных, которая соответствует требованиям пользователей (эффективности, адаптивности, независимости, защищенности, целостности и т.д.) и удовлетворяет системным ограничениям (требования к аппаратному и программному обеспечению). Схема БД с универсальной моделью данных (УМД) [1], в этом смысле не является исключением.

Поэтому прежде чем приступать собственно к разработке физической схемы базы данных с УМД, была выполнена определенная работа по анализу существующих методов проектирования БД [2 – 4].

На основе проведенного анализа были сформулированы некоторые рекомендации (в виде основных факторов успешного проектирования), следование которым позволяет спроектировать успешную физическую схему базы данных с универсальной моделью данных, удобно сопровождаемую и переносимую на различные платформы. А именно:

- четкое формулирование цели разработки;
- строгое следование методологии универсальной модели данных;
- учет особенностей требуемой обработки информации (объем хранимых данных; поддержка структурной целостности и согласованности; требования к защите информации и т. д.);
- учет характеристик пользователей системы (важность, приоритеты; число пользователей; распределение функций между пользователями; возможность работы в распределенной среде);
- учет возможностей технических и программных средств, на которых планируется реали-

зация схемы базы данных (объем памяти; быстродействие; особенности программного обеспечения, языков манипулирования данными и т. д.);

- возможность возврата к уже выполненным ранее этапам, если это требуется для достижения оптимальных результатов;
- наличие средств автоматизации проектирования.

**Постановка задачи.** Руководствуясь факторами (показателями качества) успешного проектирования, требуется разработать такую стандартную (универсальную) схему базы данных для реляционной СУБД, которую можно было бы использовать для работы с данными любой предметной области (ПРО).

### Разработка схемы базы данных с универсальной моделью данных

Зафиксируем конечное множество отношений универсальной модели данных  $\mathfrak{R} = \{R_1, R_2, \dots, R_{17}\} = \{M, R, C, T, O, H, Z_{01}, Z_{02}, D, I, E, B, X, Y, P, J, K\}$  и определим пространство состояний базы данных, как конечное семейство конечных отношений различной арности.  $M, R, C, T, O, H, Z_{01}, Z_{02}, D, F, I, E, B, X, Y, P, J, K$  – соответствующие отношения: предметных областей, разделов, классов типов, экземпляров, характеристик объектов, значений фактических и паспортных характеристик объектов, документов, папок документов, единиц измерений, классов, экземпляров, характеристик, значений характеристик событий, классов параметров, характеристик параметров и значений характеристик параметров объектов УМД [5]

Тип каждого отношения не зависит от состояния и задается множеством имен атрибутов  $at(R_i) = \{A_{i1}, A_{i2}, \dots, A_{ik_i}\}$ , называемым *схемой отношения с именем  $R_i$* , или просто *схемой отношения  $R_i$* . Для явного указания конкретного типа отношения используется следующая форма записи:  $R_i(A_{i1}, A_{i2}, \dots, A_{ik_i})$ .

Тогда множество схем отношений УМД будет иметь следующий вид:

$$\begin{aligned} R_1(A_{11}, A_{12}, \dots, A_{1k_1}) &= M(m_1, m_2), \\ R_2(A_{21}, A_{22}, \dots, A_{2k_2}) &= R(r_1, r_2, m_1), \\ &\dots \\ R_{17}(A_{17_1}, A_{17_2}, \dots, A_{17_{k_{17}}}) &= K(k_1, k_2, j_1, o_1, m_1). \end{aligned} \quad (1)$$

Множество схем отношений (1) является инвариантом БД с УМД или схемой БД с УМД.

Со схемой БД с УМД связан конечный (базисный) набор атрибутов:

$$\text{at}(\text{БД}_{\text{УМД}}) = \bigcup_{i=1}^n \text{at}(R_i), \text{ где } n=17, \quad (2)$$

и конечное множество доменов:  $D = \{D_1, D_2, \dots, D_m\}$ . Для каждого  $D_\alpha$  найдется имя атрибута  $A_{ik_j} \in \text{at}(\text{БД}_{\text{УМД}})$  такое, что  $D_\alpha = \text{dom } A_{ik_j}$ , где  $j=1..n$ .

Далее осуществим переход от имеющихся отношений УМД к таблицам схемы БД. Не забывая при этом, что такой переход допускает возможную декомпозицию одного отношения на несколько таблиц, с учетом важных свойств: свойства соединения без потерь; свойства сохранения зависимостей.

Опираясь на определенные в работе [1] математические отношения и соответствующие им реляционные отношения (1), определим основные таблицы схемы БД с УМД, их назначение и состав (атрибуты, базисный набор которых определяется выражением (2)).

Таблица *разделов* (соответствует отношению разделов R) – содержит разделы проектов или организационных структур рассматриваемых ПрО. На псевдокоде (используя зарезервированные слова языка SQL) данную таблицу можно представить следующим образом:

```
LPU_ID NUMERIC(12) not null PRIMARY KEY,
OWNER_ID NUMERIC(12),
LPU_NAME VARCHAR(255) not null,
ID_PRO NUMERIC(12) not null.
```

Таблица *классов объектов* (соответствует отношению классов объектов C) – содержит классы объектов рассматриваемых ПрО с возможным ограничением числа экземпляров объектов, входящих в определенный класс. На псевдокоде ее можно представить следующим образом:

```
OBJ_CLASS_ID NUMERIC(12) not null PRIMARY KEY,
OBJ_CLASS_NAME VARCHAR(255) not null,
CLASS_OWNER_ID NUMERIC(12),
MAX_CHILDREN NUMERIC(5),
ID_PRO NUMERIC(12) not null.
```

Таблица *типов объектов* (соответствует отношению типов объектов T) – содержит типы объектов рассматриваемых ПрО. На псевдокоде ее можно представить следующим образом:

```
OBJ_TYPE_ID NUMERIC(12) not null PRIMARY KEY
OBJ_TYPE_NAME VARCHAR(255) not null,
OBJ_CLASS_ID NUMERIC(12) not null,
ID_PRO NUMERIC(12) not null.
```

Таблица *экземпляров объектов* (соответствует отношению экземпляров объектов O) – содержит экземпляры объектов рассматриваемых ПрО. На псевдокоде ее можно представить следующим образом:

```
OBJECT_ID NUMERIC(12) not null PRIMARY KEY
OBJ_SNAME VARCHAR(255) not null,
OBJ_OWNER_ID NUMERIC(12),
LPU_ID NUMERIC(12),
OBJ_CLASS_ID NUMERIC(12) not null,
OBJ_TYPE_ID NUMERIC(12) not null,
ID_PRO NUMERIC(12) not null.
```

Таблица *классов событий* (соответствует отношению классов событий E) – содержит классы событий рассматриваемых ПрО и состоит из следующих атрибутов:

```
EVENT_TYPE_ID NUMERIC(12) not null PRIMARY KEY,
EVENT_NAME VARCHAR(255) not null,
EV_TYPE_PARENT_ID NUMERIC(12),
CLASS_ID NUMERIC(12),
ID_PRO NUMERIC(12) not null.
```

Таблица *экземпляров событий* (соответствует отношению экземпляров событий B) – содержит экземпляры событий. На псевдокоде ее можно представить следующим образом:

```
EVENT_ID NUMERIC(12) not null PRIMARY KEY,
EVENT_TYPE_ID NUMERIC(12) not null,
EVENT_TIME DATE not null,
EVENT_END_TIME DATE,
EV_OWNER_ID NUMERIC(12),
NOTION_ID NUMERIC(12) not null,
ID_PRO NUMERIC(12) not null.
```

Подобным образом можно представить все остальные таблицы схемы БД с УМД.

Физическое проектирование этих таблиц в среде целевой СУБД может осуществляться как с помощью различных CASE-средств, так и «вручную», например, с помощью языка определения данных DDL языка SQL-запросов. Что определяется квалификацией проектировщика, знанием им конкретных CASE-средств и просто привычкой. Например, физическое проектирование таблиц для конкретной реализации схемы БД с УМД в СУБД Oracle осуществлялось с помощью: CASE-средства компании *Logic Works – ERwin*; языка определения данных DDL (использовались и единичные операторы языка DDL и специально написанные скрипты); интегрированных средств программного приложения *PL/SQL Developer*.

Чтобы учесть особенности требуемой обработки информации (в том числе обеспечение целостности, доступности, защищенности данных), характеристики будущих пользователей системы, возможности технических и программных средств, на которых планируется реализация (в соответствии с определенными выше факторами успеха), в схеме базы данных с УМД были разработаны и некоторые другие объекты. А именно, кроме базовых таблиц в схему БД с УМД были включены такие объекты как: представления, триггеры, серверные процедуры и функции, синонимы, последовательности, роли, специальная таблица-журнал измененных данных и т.д. Рассмотрим их подробнее.

*Представление* – это динамический результат, одной или нескольких реляционных операций над базовыми отношениями с целью создания некоторого иного отношения. К схеме БД с УМД относятся только те представления, которые необходимы для поддержки функциональности ее объектов (серверных процедур, функций). Представления же для нужд конкретных пользователей не относятся к схеме БД с УМД (ее ядру). Они могут разрабатываться дополнительно и отдельно, в том числе и самим пользователем.

*Триггеры* схемы БД с УМД определяют действия, которые должны быть предприняты базой данных при возникновении в приложении некоторого события. Они используются для осуществления определенных ограничений ссылочной целостности, комплексных ограничений предметной области базы данных, контроля изменений в данных, контроля доступа и т.д. Например, с помощью триггеров схемы БД с УМД обеспечиваются: генерация значений кодов первичных ключей основных таблиц схемы БД с УМД; ограничения на возможность редактирования таблиц; проверка соответствия типов значений характеристик объектов, событий, параметров; проверка единиц измерения характеристик объектов, событий, параметров и т.д.

С помощью специальных конструкций языка определения данных (DDL), используемых при создании базовых таблиц схемы БД с УМД, а также системы *триггеров и серверных процедур* в БД с УМД обеспечивается целостность:

- иерархий разделов, классов объектов, классов событий, классов параметров объектов, папок документов;
- иерархий экземпляров объектов и событий;
- имен классов объектов, событий, параметров объектов;
- имен характеристик объектов, событий, параметров объектов;
- типов объектов;
- допустимых значений и значений по умолчанию и т.д.

Также *системой триггеров и серверных процедур* схемы БД с УМД реализуются свойственные ей бизнес-правила, а именно: в один и тот же момент времени с одним экземпляром объекта не может происходить больше одного события; у одного события, происходящего с одним экземпляром объекта, может быть несколько подчиненных событий с разными экземплярами объектов, происходящими в один момент времени; но у конкретного события (его экземпляра), которое происходит с экземпляром объекта определенного класса, событие-владелец может быть только одно; ограничение на количество экземпляров объектов, входящих в определенный класс объектов.

При этом схема БД с УМД не имеет ограничений по количеству иерархий, разделов, классов объектов, событий, параметров и типов объектов, типов

характеристик объектов и событий, значений характеристик объектов, событий и параметров. Эти значения соответствуют реальным потребностям, возникающим при описании большинства предметных областей. Ограничения для создаваемой конкретной БД с УМД определяются лицензионными и техническими ограничениями используемой СУБД и ресурсами средств.

Кроме того *серверные процедуры* используются для корректного (с соответствующими верификационными сообщениями в случае неверных действий) добавления, удаления и редактирования записей в таблицах БД с УМД. Например, серверные процедуры позволяют добавлять, удалять, редактировать записи в таблице документов (включая BLOB объекты), в таблице классов объектов, событий и т.д. *Серверные функции* схемы БД с УМД применяются для осуществления различных преобразований данных. Например, имеются: функция преобразующая дату в формате *varchar* в формат *date*; функция, возвращающая по соответствующему номеру (*ID* коду) имена разделов, классов, типов, экземпляров объектов, классов событий и т.д. Кроме того, с помощью *пакетов, серверных процедур и функций* можно манипулировать данными и осуществлять распределение прав доступа, вплоть до конкретного элемента данных.

*Журнал измененных данных* – это специальная таблица схемы БД с УМД, в которую при необходимости могут помещаться некоторые данные, способствующие восстановлению неправильно измененных или утраченных данных, а также помогающие «отследить» кто, когда и какие изменения осуществлял, используя для этого строки метаописания ЯМД. Функции этого журнала несколько сходны с некоторыми функциями журналов повторного выполнения, которые, как правило, являются неотъемлемой частью всех СУБД. Однако последние используются только для восстановления при сбое или при поддержке резервной базы данных (в том числе и восстановления таблицы журнала измененных данных). Журнал же измененных данных, храня информацию: о пользователе (системное имя, IP-адрес компьютера, активизировавший сеанс работы), который производил изменения или удаления, с указанием его конкретных действий (вставка, удаление, изменение данных) и времени; об имени базы данных; о специальном операторе языка модели данных, может использоваться как при восстановлении экземпляра той БД, где он (журнал) определен, так и при репликациях в распределенных системах.

*Последовательности* – объекты базы данных, которые используются для генерации целочисленных значений, следующих с определенным шагом (в некоторых БД их называют генераторами). В схеме БД с УМД последовательности используются в триггерах для автоматического формирования первичных ключей (очень удобно, и снимает с разработчика необходимость реализовывать алгоритм для создания

значений первичного ключа). Их число определяется числом основных таблиц схемы БД с УМД. *Синонимы, роли, политики, профили* – объекты базы данных, которые используются в БД с УМД для организации прав доступа. *Синоним* – это объект базы данных, необходимый для указания альтернативного имени какого-либо объекта базы данных. *Роль* – это объект базы данных, который используется, для упрощения управления привилегиями системного и объектного уровня (другими словами это именованная группа привилегий). *Политика* – набор правил или комплекс алгоритмов, в соответствии с которыми осуществляется управление вычислительными системами, сетями, базами данных и т.п. *Профиль* – объект базы данных, ограничивающий использование пользователем системных ресурсов.

Таким образом, в состав схемы БД с УМД (рис. 1) вошли: фиксированный набор взаимосвязанных таблиц (часть из которых представлена на

рис. 2); представления; триггеры; пакеты, серверные процедуры и функции; специальная таблица-журнал измененных данных; иные объекты: синонимы, последовательности, роли и т.д.

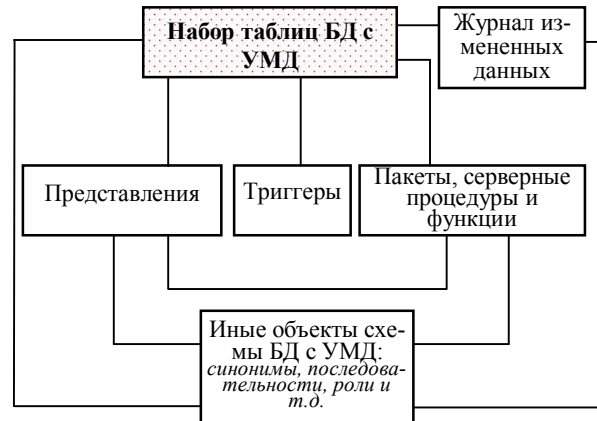


Рис. 1. Структура схемы БД с УМД

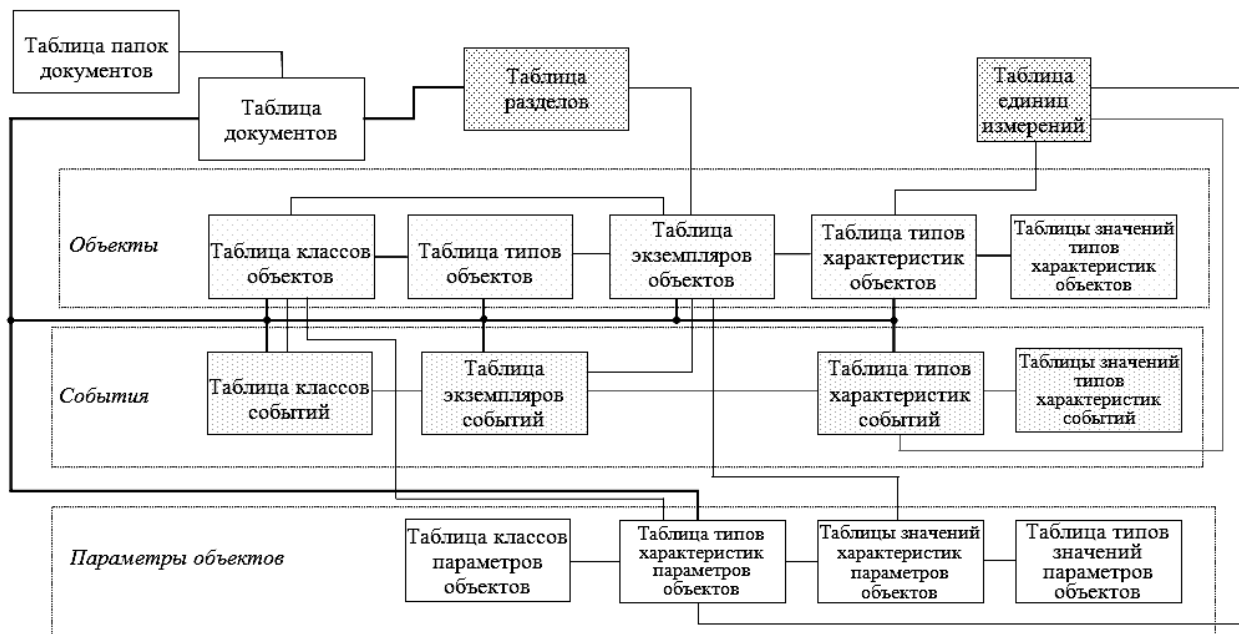


Рис. 2. Основные таблицы схемы БД с УМД и их связи

Для облегчения работы администраторов базы данных разработаны (и могут разрабатываться пользователями самостоятельно) специальные сценарии (скрипты) формирующие эти объекты схемы БД при инсталляции и в процессе эксплуатации.

Таким образом, *схема базы данных с УМД* представляет собой неизменяемую (стандартную) для разных предметных областей физическую схему БД в реляционной СУБД. Ее структура не зависит от набора данных.

Поэтому, когда осуществляется моделирование любой предметной области, то заранее неограниченное многообразие элементов этой предметной области распределяется по ограниченному набору таблиц, соответствующих метаонтологиям модели «объект-событие» [6]. Каждый элемент предметной области в БД с УМД имеет не менее двух записей:

«запись-описание» (метаданные) и «запись-значение» (данные), что обеспечивает совместное хранение метаданных и данных. Это позволяет при любом расширении набора объектов, событий, характеристик объектов и событий, параметров объектов и т.д. не добавлять в структуру данных строку, таблицу, поле или ключ, а просто добавлять в существующую таблицу схемы БД новую запись.

При этом в БД с УМД имеется возможность путем задания соответствующей записи в соответствующей таблице определять абстрактные типы данных для различных характеристик объектов, событий, параметров. Например, характеризуя каждого водителя, в таблице характеристик объектов (и в таблице-справочнике значений характеристик объектов) можно определить для него такие абстрактные типы как: пол (мужской, женский), должность (водитель,

водитель-инструктор), класс (1, 2, 3, мастер) и т.д. Хотя конкретные атрибуты каждой из таблиц схемы БД с УМД принимают определенные допустимые типы значений конкретной СУБД (например: *char*, *number*, *date*, *varchar* и т.п., смотри описание таблиц на псевдокоде). Эти данные, а точнее метаданные таблиц схемы БД с УМД, хранятся в специальных системных каталогах (словарях) данных СУБД.

### Выводы

1. Разработанная в рамках реляционной СУБД 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. Последние предполагается проектировать каждый раз по-новому для конкретной предметной области. Что объясняется в первую очередь принципиальным отличием в назначении базовых таблиц схем. В схеме БД с УМД заранее фиксированный набор таблиц используется для описания любых объектов реального мира независимо от рассматриваемой предметной области, а в традиционных схемах БД каждая таблица связывается с конкретной сущностью рассматриваемой ПрО.

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

2. В базах данных, построенных на основе схемы БД с УМД, связи (как они понимаются при традиционном проектировании, когда между таблицами сущностей для определения значений соответствующих атрибутов специально создается определенный тип связи) между моделируемыми объектами реального мира физически не прописываются. Они виртуальны и отражают естественное представление элементов реального мира в терминах модели «объект-событие». Поэтому они и легко реализуются между любыми классами, экземплярами и другими метаонтологиями модели «объект-событие» (в отличие от возможного наличия физически прописанных связей только между экземплярами сущностей при традиционном проектировании) без изменения имеющейся схемы связи.

3. В базах данных, реализованных на основе схемы БД с УМД, расширение любого набора данных добавляет только новые записи в существующие базовые таблицы и не требует изменения схемы, как это предполагается при традиционном построении БД. Так как каждый элемент предметной области в БД с УМД имеет не менее двух записей: «запись-описание» (метаданные) и «запись-значение» (данные). Причем описание объектов, их атрибутов, иерархий и связей хранятся совместно с этими сущностями, их атрибутами, иерархиями и связями, в отличие от раздельного хранения описания данных (метаданных) в специальных служебных словарях СУБД и самих данных собственно в базе данных при традиционном построении БД. Что позволяет, в том числе, использовать в БД с УМД абстрактные типы данных.

### Список литературы

1. Есин В.И. Универсальная модель данных и ее математические основы / В.И. Есин // Системы обработки информации. – Х.: Харківський університет Повітряних Сил, 2011. – № 2(92) – С. 21-24.
2. Райордан Р. Основы реляционных баз данных / пер. с англ. Р. Райордан – М.: Издательско-торговый дом «Русская Редакция», 2001. – 384 с.
3. 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.
4. Диго С.М. Базы данных. Проектирование и создание: Учебно-методический комплекс / С.М. Диго. – М.: Изд. центр ЕАОИ. 2008. – 171 с.
5. Есин В.И. Универсальная модель данных и ее отличительные особенности / В.И. Есин // Вісник Харківського національного університету. – Х.: Харьковский национальный университет им. В.Н. Каразина, 2011. – № 960. – С. 141-147 – (Серія: Математичне моделювання. Інформаційні технології. Автоматизовані системи управління"; вип. 16).
6. 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. – (Серія: Математичне моделювання. Інформаційні технології. Автоматизовані системи управління"; вип. 14).

Поступила в редколлегию 18.11.2011

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

### СХЕМА БАЗИ ДАНИХ З УНІВЕРСАЛЬНОЮ МОДЕЛЛЮ ДАНИХ

В.І. Єсін

*Описуються чинники успішного проектування, на основі яких розроблялася схема бази даних з універсальною моделлю даних, сам процес розробки, а також характерні особливості отриманої стандартної (універсальної) схеми бази даних для реляційної СУБД, яку можна використовувати для роботи з даними будь-якої наочної області.*

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

### DATABASE SCHEME WITH THE UNIVERSAL DATA MODEL

V.I. Yesin

*The factors of the successful designing of database scheme with the universal data model are described. The development process of database scheme with the universal data model which was designed on the basis of factors of the successful designing is described. The distinctive features of the got standard (universal) database scheme for relational DBMS which can be used for work with data of any subject domain is described also.*

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