

УДК 004.652

Ю.И. Рогозов, А.С. Свиридов, С.А. Кучеров, Ю.А. Жибулис

**ПОДХОД К РЕАЛИЗАЦИИ БД СО СТАТИЧЕСКОЙ СТРУКТУРОЙ
НА ОСНОВЕ МОДЕЛИ ДАННЫХ EAV**

Предлагается подход к реализации структуры базы даны (БД) на основе модели данных Entity-Attribute-Value (EAV). Подход позволяет организовать такую схему БД, которая является неизменной (статической) независимо от состава и структуры хранимых данных, а также устранить недостатки существующих реализаций EAV.

Entity-attribute-value модель; статическая структура БД; FV-структура.

U.I. Rogozov, A.S. Sviridov, S.A. Kucherov, U.A. Zhibulis

**THE APPROACH TO REALIZATION OF STATIC STRUCTURE DB ON THE
BASIS OF EAV DATA MODEL**

Approach to realization of static structure DB on the basis of EAV data model is offered. The approach allows organizing such DB scheme, which is invariable (static) irrespective of stored data structure and obviating lacks of existing EAV realizations.

Entity-attribute-value model; static DB structure; FV- structure.

Введение. На сегодняшний момент существует множество моделей данных, каждая из которых подходит для выполнения определенного круга задач, ограниченного возможностями данной модели по параметрам, определяющим эффективность ее использования. Целью данной статьи является рассмотрение особенностей модели данных EAV с точки зрения случаев, когда структура хранимых данных динамически изменяется в процессе использования БД, и определение подхода к ее реализации.

Наиболее широко используемая на данный момент реляционная модель данных является неэффективной при использовании среды «быстрой» разработки приложений либо создании БД для хранения данных с динамически изменяющейся структурой. В обоих случаях данные имеют переходящие состояния: набор сущностей и их атрибутов заранее неизвестен и потенциально велик, но фактическое их количество, используемое в базе данных, сравнительно мало.

В реляционной модели данных отдельная таблица представляет собой отдельную сущность, поэтому при изменении количества сущностей изменяется и количество таблиц. Учитывая то, что логическая структура представления данных идентична физической структуре, любое изменение структуры хранимых данных влечет за собой переработку как логической, так и физической структуры БД. Это говорит о том, что БД имеет динамическую структуру. Использование динамической структуры требует временных и финансовых затрат на доработку и оптимизацию новой БД и информационной системы (ИС) в целом.

Модель данных EAV. С целью устранения подобной проблемы целесообразно применять модель данных, позволяющую отделить физическую структуру от логической, и на основании данной модели разработать статическую структуру БД, не требующую реорганизации при изменении структуры хранимых данных. Подобной моделью является модель данных EAV (Entity-Attribute-Value), в основе которой лежит идея создания фиксированного набора таблиц в БД, не изменяемого в процессе эксплуатации и эволюции информационной системы. Применение EAV приводит к отличиям в процессе создания БД (рис. 1).

Отличие в процессе создания БД заключается в следующем:

1. Реляционная модель (традиционный подход). На первом этапе разработки создается E-R-модель данных, отражающая логическую структуру будущей БД. После процесса нормализации производится переход к физической структуре, которая фактически идентична логической. Последним этапом является непосредственная реализация в виде реляционной БД.
2. EAV-модель. Отличия логического уровня реализации от физического позволяют для конкретной предметной области получить статическую структуру БД, не изменяющуюся в процессе своего использования.

Достоинства модели данных EAV, являющиеся также отличительными особенностями данной модели, рассматриваются ниже. Важность достоинств этой модели постоянно подталкивает разработчиков искать подходы к устранению существующих у нее недостатков. В данной статье предлагается один из таких подходов.

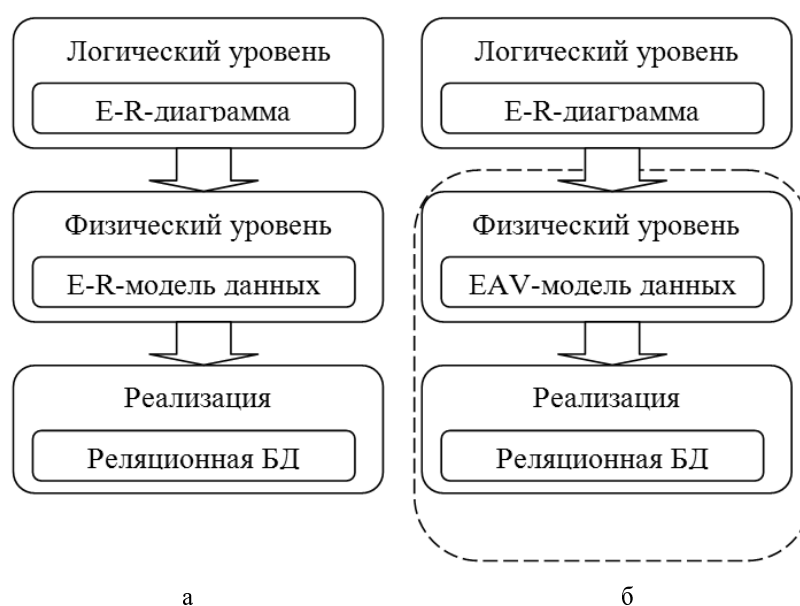


Рис. 1. Процесс создания БД

Первой особенностью EAV является способ хранения данных в виде таблицы, содержащей три столбца [1]:

- ◆ сущность – внешний ключ в таблицу описания сущности;
- ◆ атрибут – внешний ключ в таблицу описания атрибутов;
- ◆ значение – значение атрибута для конкретной связки сущность-атрибут.

Из этого видно, что в отличие от реляционных баз данных, где для каждого атрибута существует отдельная колонка, ряды в которой хранят множество характеристик сущности, в модели данных EAV каждая пара атрибут-значение – это характеристика, описывающая отдельную сущность, и ряд в таблице EAV хранит единственную характеристику.

Второй особенностью EAV являются метаданные. Посредством метаданных пользователь взаимодействует с системой в терминах логической схемы, а не физической, которые, как было сказано ранее, радикально отличаются в данной модели. Фактически метаданные используются для настройки поведения системы.

Структура метаданных в EAV достаточно сложна. Они содержатся в собственной подсхеме в пределах базы данных: внешние ключи в таблицах данных

ссылаются на таблицы в пределах этой подсхемы. Поскольку бизнес-логика находится в метаданных, а не явно в схеме базы данных, то она менее очевидна для тех, кто незнаком с системой.

Из особенностей модели данных EAV вытекают следующие достоинства [1]:

- ◆ гибкость. Нет никаких ограничений для числа атрибутов у сущности. Число атрибутов может расти, поскольку база данных развивается без модернизации схемы;
- ◆ пространственно-эффективное хранение для разреженных данных: Нет необходимости единожды резервировать место для атрибутов, значения которых являются пустыми;
- ◆ формат физических данных с частичным самоописанием данных;
- ◆ эффективность поиска. Возможность организации поиска через всю систему для поиска информации о любом интересующем объекте без предварительного указания категории, к которой он относится.

Недостатки модели данных EAV обусловлены способом представления данных и применяемыми подходами к ее реализации. В отличие от реляционной модели в EAV используется не колоночный, а строковый способ представления данных, который требует дополнительных операций преобразования. На рис. 2 показан способ реализации EAV на примере ИС Trial DB [2].

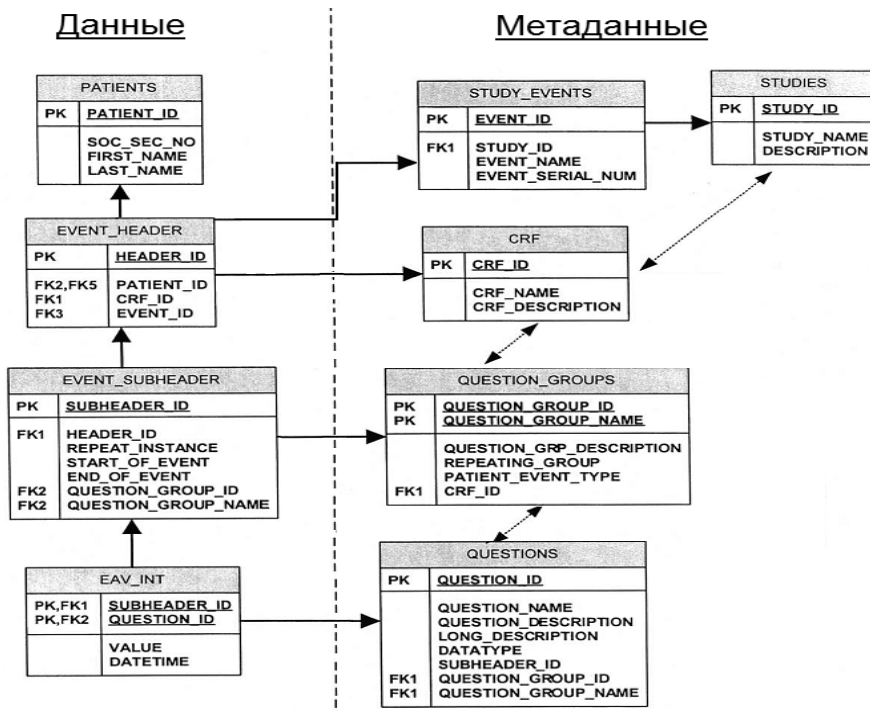


Рис. 2. Упрощенная схема Trial DB

В результате анализа приведенной схемы можно выделить следующие недостатки EAV [3]:

- ◆ структурная сложность запросов. В отличие от реляционных БД, при выборке данных из БД, основанной на EAV-модели, для получения данных по одной сущности необходимо работать с тремя и более таблицами: таблицей описания сущности, таблицей описания атрибута, таблицей

данных и таблицами метаданных. Структурная сложность запросов также повышает трудоемкость процесса их построения;

- ◆ быстроедействие операций с данными. Во-первых, для представления данных в привычной для восприятия пользователем колоночной форме необходимо выполнить операцию «поворота» – преобразования данных из строк, в виде которых они хранятся в EAV, в столбцы и склеивание данных столбцов; во-вторых, каждый новый запрос к данным необходимо компилировать. Оба этих процесса требуют дополнительных временных затрат, которые сказываются на производительности.

С точки зрения организации статической структуры БД модель данных EAV обладает еще одним существенным недостатком: она обеспечивает статичность только на уровне атрибутов сущностей. На изменение структуры БД не влияет только изменение набора атрибутов, а набор сущностей является фиксированным, заранее определяемая предметной областью, и его изменение влечет изменение структуры данных.

Задачей подхода к реализации, описываемого ниже, является создание такой структуры БД, которая бы позволила добиться статичности не только на уровне атрибутов, но и на уровне сущностей, а также устранила имеющиеся недостатки модели данных EAV.

Подход к реализации статической структуры на основе EAV-модели данных. Подход предусматривает хранение, представление данных в виде совокупности сущностей и их атрибутов в виде фактов (Fact), которые характеризуются значениями (Value). Структура БД в таком случае носит имя Fact-Value (FV).

Пример реализации FV-структуры, статичный на уровне фактов, изображен на рис. 3.

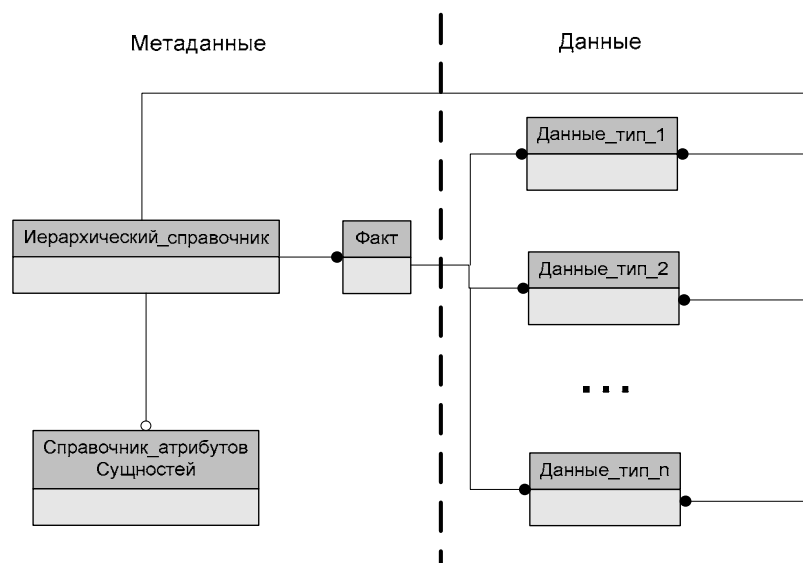


Рис. 3. FV-структура БД

FV-структуру можно условно разделить на две части: данные и метаданные. В области данных хранятся непосредственно значения фактов, причем количество таблиц соответствует количеству используемых типов данных. Метаданные, описывающие данные и определяющие логику их хранения, представлены в виде трех таблиц:

1. Иерархический справочник. Содержит информацию о хранимых данных (имена атрибутов и сущностей, краткие описания и т.д.). Так же посредством этой таблицы определяются типы данных для конкретного атрибута, осуществляется иерархия вида «страна – город – улица – дом».
2. Справочник атрибутов сущностей. Сохранение связи между сущностью и ее атрибутами.
3. Факт. Организация связи Факт–Значение.

FV-структура позволяет строго ограничить количество таблиц в БД на уровне трех таблиц метаданных и нескольких таблиц данных (5-7 таблиц), что позволяет независимо от количества хранимых сущностей и атрибутов ограничить структурную сложность запроса, повышая тем самым производительность системы, – чем меньше количество таблиц, с которыми работает запрос на выборку данных, тем меньше время его работы. Наличие хранимых процедур, осуществляющих выполнение типовых операций (просмотр, удаление, добавление данных), которое обеспечивает FV-структура, также повышает производительность работы БД за счет отсутствия необходимости компилировать каждый запрос. Вместо этого для выборки данных из БД используется передача параметров хранимым процедурам.

Обобщим отличия реализации БД на основе EAV-модели данных и FV-структуры (табл. 1).

Таблица 1

Отличия FV-структуры от EAV

Особенность	EAV	FV-структура
Структура данных	При изменении набора атрибутов структура БД не изменяется. При изменении набора сущностей структуры БД меняется	При изменении набора атрибутов структура БД не изменяется. При изменении набора сущностей структуры БД не меняется
Скорость работы	Компиляция каждого запроса в процессе выборки данных	Однократная компиляция хранимых процедур
Структура запроса	Запрос с объединением нескольких таблиц для выборки данных по одной сущности	Вызов хранимых процедур и передача им параметров, по которым необходима выборка

Таким образом, применяя на физическом уровне (см. рис. 1,б) FV-структуру, можно не только устранить недостатки существующих реализаций EAV, но и добиться статичности структуры БД в процессе эксплуатации и эволюции ИС.

Об эффективности FV-структуры можно судить на примере реализации программного средства «Примиус» [4,5]. По экспертным оценкам, в результате процесса эксплуатации FV-структура позволила добиться ускорения процесса разработки ИС в 2–4 раза, а также повышения производительности БД в 3-5 раз по сравнению с реализациями EAV.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Prakash Nadkarni*. «An Introduction to Entity-Attribute-Value Design for Generic Clinical Study Data Management Systems» // Center for Medical Informatics, Yale University Medical School, New Haven, CT.
2. *Cynthia Brandt, Prakash Nadkarni*. «Temporal query of attribute-value patient data: utilizing the constraints of clinical studies Aniruddha M. Deshpande» // International Journal of Medical Informatics. – 2003. – № 70.
3. *Jacob Anhoj*. «Generic Design of Web-Based Clinical Databases» // Journal of Medical Internet Research, 2004.

4. *Рогозов Ю.И., Бутенков С.А., Свиридов А.С. и др.* Метод создания инструментальных средств разработки автоматизированных информационно-управляющих систем // Мехатроника, автоматизация, управление. – 2008. – № 1 (82).
5. *Рогозов Ю.И., Бутенков С.А., Свиридов А.С. и др.* Метод создания инструментальных средств разработки информационных систем // Информационно-измерительные и управляющие системы. – 2008. – Т. 6. – № 3.

Рогозов Юрий Иванович

Технологический институт Федерального государственного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.

E-mail: rogozov@tsure.ru.

347928, г. Таганрог, пер. Некрасовский, 44.

Тел.: 88634371787.

Свиридов Александр Славьевич

E-mail: sviridov@tsure.ru.

Тел.: 88634371787.

Кучеров Сергей Александрович

E-mail: sergey.kutcherov@gmail.com.

Тел.: 89281922577.

Жибулис Юрий Алексеевич

E-mail: sharkru@mail.ru.

Тел.: 88634371787.

Rogozov Jury Ivanovich

Taganrog Institute of Technology – Federal State-Owned Educational Establishment of Higher Vocational Education “Southern Federal University”.

E-mail: rogozov@tsure.ru.

44, Nekrasovskiy, Taganrog, 347928, Russia.

Phone: 88634371787.

Sviridov Alexander Slavevich

E-mail: sviridov@tsure.ru.

Phone: 88634371787.

Kuchеров Sergey Alexandrovich

E-mail: sergey.kutcherov@gmail.com.

Phone: 89281922577.

Gibulis Jury Alekseevich

E-mail: sharkru@mail.ru.

Phone: 88634371787.

УДК 681.322

Д.А. Белоглазов, И.С. Коберси, В.И. Финаев

СИНТЕЗ НЕЧЕТКИХ РЕГУЛЯТОРОВ

Рассматриваются особенности построения регуляторов для априори неопределенных объектов управления, показаны алгоритмы нечеткого вывода с примерами их реализации в среде Matlab.

Неопределенность параметров; нечеткая логика; регулятор; алгоритмы.