Эффективные методы работы с вертикальной моделью данных

Д.В. Гмарь, Ю.А. Игнатова, Э.В. Цуранов, К.И. Шахгельдян

Аннотация. Адаптируемая модель данных является основой корпоративных информационных систем, таких как системы электронного документооборота, системы управления контентом и др. В статье рассматриваются проблемы, связанные с использованием адаптируемой вертикальной модели данных entity-attribute-value (EAV), и предлагаются методы работы с этой моделью, позволяющие значительно повысить ее производительность.

Ключевые слова: модель entity-attribute-value, методы работы с данными, адаптируемые системы.

Введение

Современные информационные системы, имеющие длительный жизненный цикл (к таким системам, например, относятся корпоративные информационные системы), в течение всего периода эксплуатации требуют реализации многочисленных изменений в структуре понятий, которыми они оперируют. Необходимость возникает как в создании новых понятий, так и в модификации существующих, например, расширении атрибутов понятий, а также определении новых связей между понятиями информационной системы. При этом информационная система должна обеспечивать манипуляции с такими понятиями без изменения программного кода или с его минимальной модификацией, что обеспечивает высокий уровень адаптируемости системы к изменяющимся требованиям.

Подход, который может быть использован для решения этой задачи, предполагает наличие обобщенного репозитория понятий, а также репозитория экземпляров понятий (объектов). Обобщенный репозиторий понятий должен содержать мета-описания (метаданные) объектов корпоративной информационной системы (КИС). Для физического хранения обобщенного репозитория используется источник данных, в котором хранятся атрибуты понятий не по столбцам, а по строкам. Такой подход известен как вертикальная модель данных или модель Entity-Attribute-Value (EAV) [1].

Модель EAV позволяет представить сущность в качестве абстрактного понятия, которое может содержать неограниченный набор атрибутов для задания простых или составных значений различных типов данных. Это позволяет создавать информационные системы, в которых требования по количеству и составу понятий меняется в течение всего жизненного цикла. Единый способ хранения для всех понятий составляет серьезное преимущество перед использованием традиционной реляционной модели базы данных, когда для каждого понятия создается отдельная таблица. В рамках модели EAV добавление новых понятий не влечет за собой изменение состава и структуры таблиц базы данных, создавать новые понятия, так же как определять и изменять состав их атрибутов, можно автоматизировано, и, главное, на любом этапе жизненного цикла КИС.

Достоинством такого подхода является весьма существенная гибкость решения и высокий уровень универсальности, что позволяет значительно уменьшить затраты на внедрение информационной системы в разных организациях и на ее дальнейшее сопровождение. Однако, любая универсальная система менее эффективна, чем специализированная, и обладает недостатками.

Основных недостатков у модели EAV три. Прежде всего, это сложность соблюдения целостности данных, в результате чего ухудшается их качество. Во-вторых, модели EAV присуща сложность запросов для проведения операций над данными. В-третьих, достаточно часто имеет место низкая производительность выполнения запросов для модели EAV.

Последнее обстоятельство особенно важно для КИС, где чаще всего возникает необходимость в гибкости работы с данными. И, если проблема обеспечения целостности данных довольно просто решается на уровне приложений (все манипуляции с данными осуществляются только через специализированные процедуры, которые отслеживают вопросы целостности), то, как показывает анализ, вопрос производительности становится ключевым, когда объем данных превышает несколько миллионов записей, и в большинстве случаев становится серьезным препятствием к использованию модели в больших приложениях КИС.

В статье рассмотрены методы повышения производительности работы с вертикальной моделью данных. В разделе 2 приведены основные публикации по модели EAV, ее модифицированной версии и работы, в которых анализируется основной недостаток модели EAV – низкая производительность. В разделе 3 рассмотрены модель EAV, методы, позволяющие повысить производительность информационных систем, работающих на основе модели EAV. В первую очередь выполняется сравнение двух вариантов модели EAV. Затем предлагается модифицировать запросы к модели EAV для повышения их производительности. Выполняется сравнение производительности модифицированного подхода EAV с реляционной моделью. Для дальнейшего увеличения производительности модели EAV предлагается использовать метод выравнивания статистики. Повысить общую производительность работы информационных систем, использующих модель EAV, предлагается с помощью алгоритма с раздельной выборкой атрибутов и использование вертикального представления данных. Рассмотрено также преимущество использования индексов в модели EAV. В заключении суммируются все рассмотренные подходы, обеспечивающие приемлемую производительность модели EAV для информационных систем КИС.

1. Обзор публикаций

Первые работы по модели EAV относятся к хранению биомедицинских данных, где приходится иметь дело с разнородной информацией [2, 3]. Авторами отмечается снижение производительности при использовании модели EAV в 3-5 раз по сравнению с традиционной реляционной моделью хранения данных [2].

Вариацией модели EAV является модель данных, предложенная в работе [4]. Отличие этой модели от модели EAV, обсуждаемой в работах [2, 3], состоит в том, что рассматривается создание отдельной таблицы на каждый тип данных (строка, число, дата/время и т.д.). Распределение данных по нескольким таблицам обеспечивает их более эффективное хранение, упрощает манипулирование данными и повышает масштабируемость. В работе [5] выполнен сравнительный анализ по скорости разработки, гибкости программной системы, времени выполнения операций над хранимыми объектами для различных моделей. В целом можно сделать вывод о снижении производительности запросов для модели EAV в 2-5 раз по сравнению с реляционной моделью данных.

Еще одно сравнение представлено в работе [6], где анализируются выборки по отдельным атрибутам и показано, что производительность для выборки по одному атрибуту для модели EAV снижется в 15 раз.

Проблема неэффективности выборки данных из модели EAV обсуждалась и в работе [7]. Авторами предложена оптимизированная для чтения модель EAV, предназначенная для организации хранилищ данных. Оптимизационный подход предназначен для случаев, когда запись вставляется один раз и в дальнейшем только читается, что подходит не для всех задач.

Технологические решения, использующие модель EAV, – это, прежде всего, решения класса Enterprise Recourse Planning (ERP), Customer Recourse Management (CRM), Enterprise Content Management (ECM), Content Management System (CMS). Примером последних можно считать платформу электронной коммерции Magento, реализованную на основе модели EAV [8], и которую мы использовали при тестировании, считая ее классическим примером модели EAV.

2. Модель EAV и методы повышения производительности запросов

2.1. Модель EAV

Модель EAV представляет собой универсальную структуру данных, в которой данные и метаданные хранятся в одном линейном списке. В модели EAV понятие описывается набором атрибутов. Атрибуты, которые позволяют определять связи между понятиями, называются свойствами понятий. Модель EAV, поддерживающая свойства, принято называть EAV/CR [9, 10]. Каждый атрибут понятия описывается набором характеристик.

Для организации структуры хранения данных модели EAV на физическом уровне в реляционной базе данных достаточно иметь всего три таблицы – таблицу, описывающую понятия, таблицу, описывающую атрибуты, и таблицу, содержащую непосредственно значения атрибутов. При этом таблица атрибутов содержит ссылку на идентификатор понятия из таблицы понятий, а таблица значений содержит ссылки на идентификатор атрибута, позволяя, таким образом, определить к какому понятию относятся те или иные значения (Рис. 1) [1].

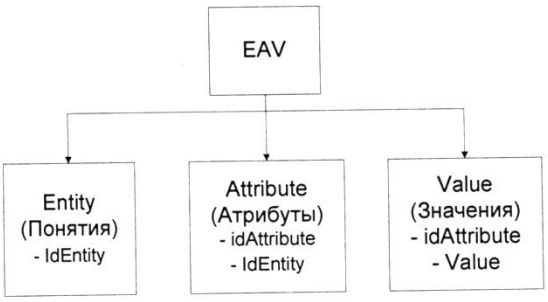

Рис. 1. Модель EAV

Первые две таблицы (таблицы понятий и атрибутов) представляют собой метаданные, таблица значений атрибутов – данные.

Для пояснения модели EAV в простом виде представим понятие в реляционной базе данных и в модели EAV. Реляционная модель описывает одно понятие как:

Понятие 1	Значение атрибута1	...	Значение атрибутаN

Модель EAV описывает одно понятие как:

Понятие 1	Атрибут11	Значение атрибута11
...	...	...
Понятие1	Атрибут1N	Значение атрибута1N

Упрощенная логическая схема базы данных модели EAV представлена на Рис. 2. Таблица

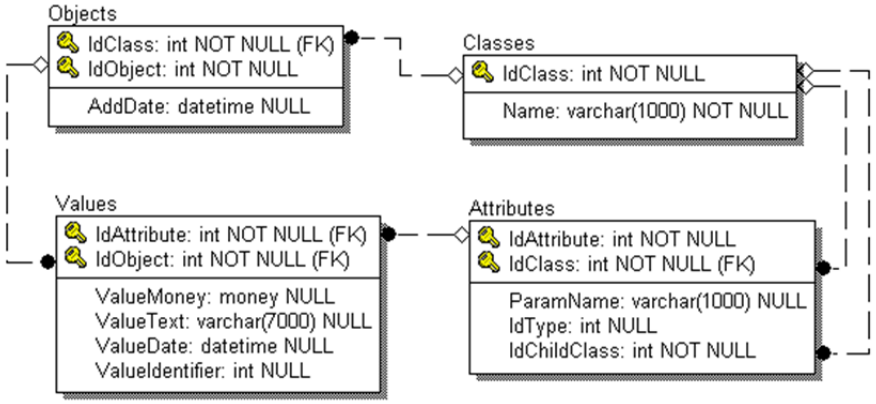

Рис. 2. Упрощенная модель EAV с хранением всех типов данных в одной таблице

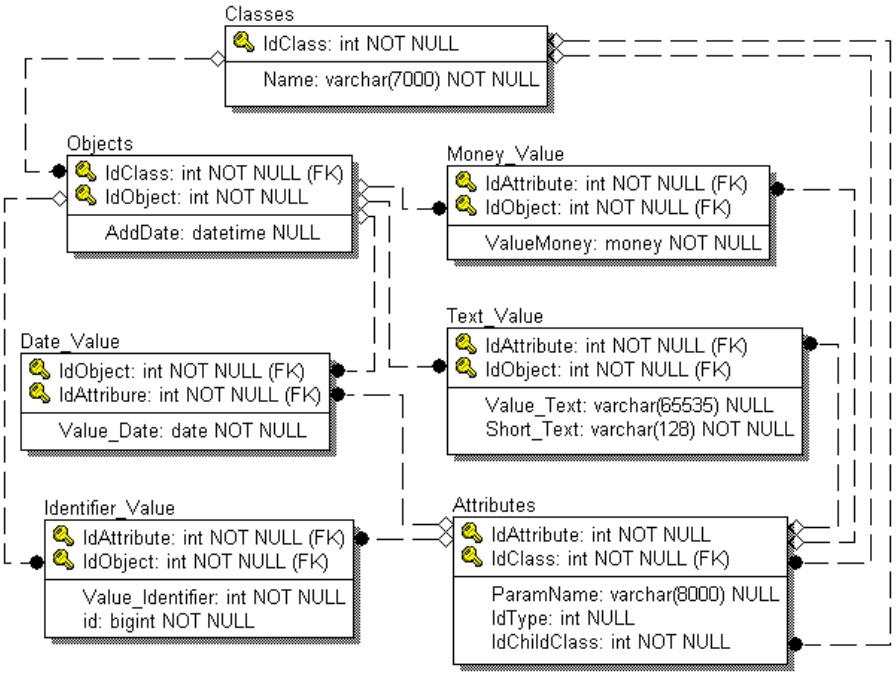

Рис. 3. Упрощенная логическая модель EAV с хранением разных типов данных в разных таблицах

Classes включает список всех понятий, экземплярами которых являются объекты КИС. Таблица Objects содержит список всех объектов КИС. Таблица Attributes содержит описание всех атрибутов понятий с указанием их типов данных. Таблица Values хранит значения всех атрибутов – каждое поле отвечает за свой тип данных атрибута.

Таблицы Classes и Attributes относятся к метаданным, таблицы Objects и Values – к данным. Использование типа money в данном случае не принципиально, объясняется достаточной для задач КИС точностью (4 знака после запятой). Для большей точности можно использовать иные типы данных, например, decimal или numeric.

Альтернативная модель EAV приведена на Рис. 3. Таблицы с постфиксом _Value хранят значения атрибутов соответствующих им типов данных. Таблицы Money_Value, Text_Value, Date_Value, Identifier_Value представляют собой таблицы значений, включающие данные – число, текст, дата/время, идентификатор соответственно.

2.1. Условия проведения экспериментов

Основным недостатком модели EAV является ее низкая производительность. Авторами этой работы предложены методы, обеспечивающие приемлемую производительность приложений КИС, которые используют модель EAV. Для оценки производительности предлагаемых методов проведены эксперименты с использованием «Классификатора адресов России» (КЛАДР, http://ru.wikipedia.org/wiki/КЛАДР), который находится в свободном доступе. Количество адресов на момент проведения экспериментов составляло 900 тысяч. База данных КЛАДР не обеспечивает моделирование всех возможных ситуаций, но для беспристрастности эксперимента использовались данные, которые наилучшим образом соответствуют реляционной модели.

Наиболее распространенная операция с данными — это выборка значений атрибутов объектов, удовлетворяющих определенному условию (фильтру по значениям атрибутов). Дополнительно к этой задаче можно добавить сортировку и представление данных в табличном виде. Поэтому для экспериментов сформированы следующие варианты запросов:

Запрос 1. Выборка всех атрибутов понятия с фильтрацией по одному атрибуту (по названию улицы) без сортировки.

Запрос 2. Выборка всех атрибутов с сортировкой и фильтром по одному атри-

буту (фильтр по названию улицы и сортировка – область, район, город, населенный пункт, улица).

Запрос 3. Выборка всех атрибутов с сортировкой и фильтром по двум атрибутам: улица и населенный пункт.

Эти три запроса являются наиболее распространенными среди запросов к данным в КИС: простая выборка с фильтрацией по одному атрибуту, выборка с сортировкой и фильтрацией по одному атрибуту и тоже самое по нескольким атрибутам.

Дополнительно в отдельных экспериментах мы используем запрос с фильтрацией по трем атрибутам:

Запрос 4. Выборка всех атрибутов с сортировкой и фильтром по трем атрибутам: улица, населенный пункт, область.

Для каждого теста выполнялись 1000 запросов с разными значениями атрибутов для фильтрации. В результатах представлено суммарное время выполнения 1000 запросов в миллисекундах.

В процедурах сравнения разных методов используются три источника данных. Первый источник данных представляет собой реляционную модель, содержащую 5 таблиц, приведенных к нормализованному виду. Второй источник данных предполагает хранение всех атрибутов в одной таблице (в этом случае атрибуты разных типов данных хранятся в одной таблице, что соответствует модели EAV на Рис. 2). Эту модель далее будем называть EAV1. Третий источник данных предполагает, что для каждого типа данных атрибута выделяется отдельная таблица. Модель в этом случае будем называть EAV2 (соответствует представлению на Рис. 3).

Для повышения производительности моделей EAV1 и EAV2 построены индексы для таблиц, где хранятся значения атрибутов:

1. по наборам полей - объект, атрибут, значение
   id_entity, id_attribute, XXX _value
2. по наборам полей - атрибут, значение, объект
   id_attribute, XXX_value, id_entity

Все эксперименты выполнялись на сервере с Intel Xeon X5660 2.80Ghz, 6 ядер, 12 потоков оперативной памяти: 288 Гб, дисковая подсистема: 11 дисков (из них 2 SSD) под управлением Windows Server 2008 Enterprise Версия 6.0 (64 разрядная). Система управления базами данных (СУБД): Microsoft SQL Server Standard Edition (64-bit) версия 10.0.6000.29.

2.2. Сравнение EAV1 и EAV2

Цель первого эксперимента - сравнить производительность запросов к данным для моделей EAV1 и EAV2.

Результат сравнения выполнения трех вышеприведенных запросов приведен в Табл. 1. Так как модель EAV2 использует больше таблиц (по одной таблице на каждый тип данных), чем модель EAV1, а запросы к полям с пустыми значениями для модели EAV1 имеют минимальные накладные расходы, то запросы к модели EAV2 имеют больше операций ввода-вывода и, соответственно, более низкую производительность. При этом разница в производительности EAV2 и EAV1 в среднем составляет 3%.

Табл. 1. Результаты тестов запросов EAV1 и EAV2

Запрос	Модель EAV1 (мс)	Модель EAV2 (мс)
Запрос 1	7977	8275
Запрос 2	12129	12490
Запрос 3	4620	4785

Заметим, что результаты этого эксперимента противоречат данным, представленным в работе [4], где показано, что модель EAV1 медленнее, чем модель EAV2. Следует отметить, что мы получали в отдельных тестах такое же соотношение, как авторы [4], но в целом, по результатам наших экспериментов модель EAV1 обеспечила небольшое опережение по производительности в сравнении с моделью EAV2. В тоже время результаты нашего эксперимента соответствует данным работы [11].

Наибольшее влияние на производительность выполнения запросов оказывает количество операций ввода-вывода [12, 13]. Поэтому при разработке методов мы стремились уменьшить это количество и тем самым повысить производительность запросов.

2.3. Модифицированный подход к построению запросов к EAV

Рассмотрим общепринятый подход к формированию запросов для модели EAV1 на при-

мере платформы Magento, который состоит в том, что количество таблиц в запросе соответствует количеству атрибутов, используемых для выборки и фильтрации.

Общий вид запроса для выборки с фильтрацией для общепринятого подхода модели EAV1 выглядит следующим образом:

```
select t1.ValueXXX, t2.ValueXXX, ...,
tN.ValueXXX from Values tn1
join Values tn2 on
tn2.IdObject=tn1.IdObject and
tn2.IdAttribute=n2 join Values tnN on
tnN.IdObject=tn1.IdObject and
tnN.IdAttribute=nN join Values tm1 on
tm1.IdObject=tn1.IdObject and
tm1.IdAttribute=m1 join Values tmM on
tmM.IdObject= tn1.IdObject and
tmM.IdAttribute=mM where
tn1.IdAttribute=n1 and [условие
фильтрации]
```

где N – количество атрибутов в выборке, M – количество атрибутов в фильтрации, XXX – Money, Text, Date или Identifier (или иной тип данных) в зависимости от типа данных атрибута.

Так выглядят запросы в различных системах, использующих модель EAV. В этом запросе многократно используется одна и та же таблица в соответствие с количеством атрибутов в выборке (N+M).

Для того чтобы повысить производительность запросов к EAV1, нужно уменьшить число операций ввода-вывода. Для уменьшения числа операций ввода-вывода можно сократить количество обращений к одной и той же таблице в вышеприведенном запросе для атрибутов выборки (по N). Для этого атрибуты для выборки данных предлагается выводить «вертикально» из одной таблицы. Этот подход далее будем называть «модифицированный».

Для модифицированного подхода модели EAV1 запрос для выборки с фильтрацией выглядит так:

```
select t.IdAttribute, t.ValueMoney,
t.ValueText, t.ValueDate,
t.ValueIdentifier
from Values t where exists(select * from
Values tm1
join Values tm2 on
tm2.IdObject=t.IdObject and
tm2.IdAttribute=m2
join Values tmM on
tmM.IdObject=t.IdObject and
tmM.IdAttribute=mM
  where tm1.IdAttribute=m1 and
tm1.IdObject=t.IdObject and [условие
фильтрации])
```

Таким образом, для общепринятого подхода запрос содержит M+N таблиц, а для модифицированного M+1.

Результаты эксперимента по сравнению общепринятых запросов и модифицированных запросов приведены в Табл. 2. В ней видно, что производительность модифицированного подхода на 7-25% выше, чем общепринятого.

Табл. 2. Результаты тестов общепринятого и модифицированного запроса для модели EAV1

Запрос	Классическая модель EAV1 (мс)	Модель EAV1 с модифицированным подходом (мс)
Запрос 1	10629	7977
Запрос 2	13422	12129
Запрос 3	4975	4620

В дальнейших экспериментах без специального указания запросы к EAV модели основываются на модифицированном подходе как более производительном.

2.4. Сравнение модели EAV и реляционной модели

Модель EAV, как было отмечено в публикациях [2-6], имеет серьезные проблемы в части производительности в сравнении с реляционной моделью. В этом разделе предлагаем сравнить классическую модель EAV, модель EAV с модифицированным подходом и реляционную модель. Результаты экспериментов приведены в Табл. 3.

Табл. 3. Сравнение различных моделей

Запрос	Классическая EAV1 (мс)	Модифицированная EAV1 (мс)	Реляционная модель (мс)
Запрос 1	10629	7977	3469
Запрос 2	13422	12129	5557
Запрос 3	4975	4620	3069

Для большей наглядности в Табл.4 произведено сравнение производительности подходов. Реляционная модель быстрее классической

модели в 2,17-3,17 раза. Классическая модель EAV проигрывает модели EAV с модифицированным подходом в 1,2 – 1,45 раза, т.к. индексы для этой системы не оптимальны и запросы не модифицированы, что приводит к дополнительным расходам на операции ввода-вывода. Модель EAV с модифицированным подходом отстает по производительности от реляционной в 1,5-2,3 раза.

Табл. 4. Сравнение результатов экспериментов для модели EAV с модифицированным подходом, классической модели EAV и реляционной модели

Запрос	EAV с модифицированным подходом vs Реляционной модели	Классическая EAV vs EAV с модифицированным подходом	Классическая EAV vs Реляционной модели
Запрос 1	2,3	1,2	2,76
Запрос 2	2,18	1,45	3,17
Запрос 3	1,5	1,44	2,17

Таким образом, чем сложнее запрос (добавление фильтрации – запрос 2, и увеличение атрибутов при фильтрации – запрос 3), тем меньше становится разница между производительностью модифицированного EAV и реляционной модели. Сложные запросы могут быть значительно улучшены за счет метода выравнивания статистики, предложенного в следующем разделе.

2.5. Выравнивание статистики

Модель EAV имеет большие преимущества в части гибкости и возможности поддержки информационных систем без привлечения программистов. Последнее означает, что необходимо уменьшить зависимость производительности запросов от профессионального уровня разработчиков этих запросов. Для этого рассмотрим оптимизационные мероприятия, которые выполняет СУБД.

Перед выполнением запроса оптимизатор строит план запроса на основе статистики по полям таблиц и индексов. Статистика имеет ограничения, например, в MS SQL Server хранится информация по не более, чем 200 диапазонам идентификаторов с соответствующей им гистограммой, и достаточно часто такая статистика бывает неравномерной. Это может приводить к ошибкам в оценке количества возвращаемых фильтрами значений, выполняемой оптимизатором, и, как следствие, к выбору неоптимального плана запроса и значительной потере производительности, что справедливо как для модели EAV, так и для реляционной модели.

Для примера рассмотрим модель EAV, используемую для системы управления контентом веб-сайта www.vvsu.ru. Наиболее частой операцией при выборке данных является фильтрация объектов с использованием пары: идентификатор атрибута и его значение. В таблице Indentifier_Value находится около 4.6 миллиона уникальных пар «атрибут-значение» на 15.7 миллионов записей со статистическим распределением, приведенным на Рис. 4.

По вертикали (логарифмическая шкала) - количество объектов для пары, по горизонтали – порядковый номер (скрыт). Подобное неравномерное распределение имеет большинство EAV источников данных. В этом случае оптимизатор строит неоптимальные планы запросов, в том смысле, что не используется нужный индекс (опираясь на неравномерное распределение, он пренебрегает нужным индексом).

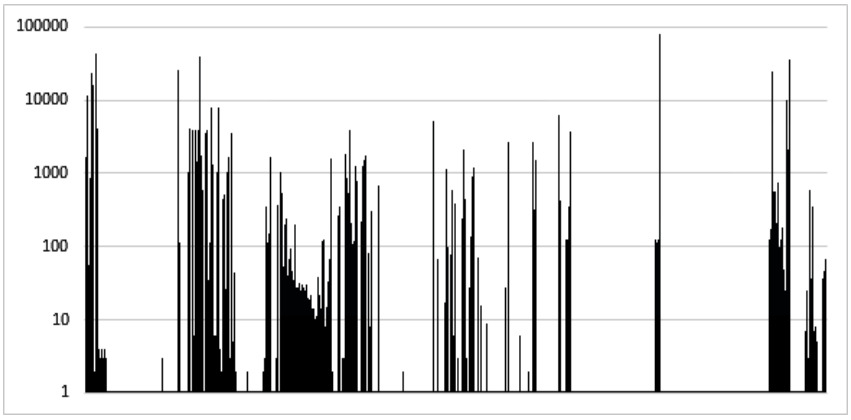

Рис. 4. Гистограмма распределения пары «атрибут-значение»

Проблему можно решить несколькими способами. Во-первых, это можно сделать с помощью подсказок оптимизатору в запросах (hints) на основе знаний о распределении статистики в разных ситуациях. Но использование подсказок не обеспечивает гибкости, так как данные могут измениться, и подсказки станут неэффективными, а для их оптимизации требуется привлечение профессиональных программистов.

Во-вторых, можно выровнять статистику для соответствующих полей таблицы модели EAV. Авторами предлагается преобразовать таблицу Identifier_Value(Id_Entity, Id_Attribute, Value_Identifier) в две таблицы: Identifier_Value (Id_Entity, Id_AV) и IV_Attribute_Value(Id_AV, Id_Attribute, Value_Identity). Id_AV – идентификатор, представляющий собой порядковый номер, сформированный таким образом, чтобы большему номеру соответствовала пара «атрибут-значение», которая задана для большего числа объектов.

В результате получается следующее распределение для Id_AV (Рис. 5).

Так как подавляющее большинство идентификаторов соответствует только одному объекту, то большинство объектов попадет в первый диапазон статистики, что позволяет оптимизатору строить эффективные запросы. Это связано с тем, что так как большинство объектов ассоциируется с первым (одним) диапазоном, то для большинства запросов попадание идентификаторов объектов в этот диапазон будет наиболее вероятным. Кроме того, для немногих, не вошедших в первый диапазон статистики идентификаторов, сервер может построить более детальную статистику, что позволит оптимизатору строить более эффективный запрос.

Выравнивание статистики уменьшает зависимость от разработчика при использовании модели EAV, так как в конечном итоге автоматизирует построение эффективных запросов к модели EAV. Кроме того, такой подход может уменьшить объем хранимых данных за счет хранения уникальных пар «атрибут-значение».

В Табл. 5 приведены результаты сравнительных экспериментов для выравненной и не выравненной статистики. Как видно, выравнивание статистики позволяет повысить производительность запросов в 1,25- 1,49 раз.

Табл. 5. Сравнение производительности обработки запросов с выравненной и не выравненной статистикой

№	Запрос	Без выравнивания статистики (мс)	С выравниванием статистики (мс)
1	Запрос 3	17363	13887
2	Запрос 4	14317	9602

2.6. Алгоритм с раздельной выборкой атрибутов

Конечные пользователи КИС не рассматривают отдельно производительность модели данных (поиск и извлечение) и производительность обработки и представления. Для конечного пользователя наиболее важным является общая производительность информационной системы от момента посылки запроса до получения необходимого пользователю результата, представленного на одной странице.

Исходя из этого понимания, авторами предлагается следующая последовательность операций по обработке запросов пользователей от момента поиска до представления результатов, где используется общий репозиторий на основе модели EAV. В общем случае запросы представляют собой фильтрацию по некоторому

Рис. 5. Гистограмма распределения идентификаторов после перекодирования

подмножеству атрибутов и сортировку объектов по некоторому (возможно другому) подмножеству атрибутов.

Пусть множество $X = \{x_i, i = \overline{1, N_X}\}$ – это множество всех объектов некоторого понятия, для которого необходимо по запросу пользователя извлечь и представить объекты $X' \subseteq X$, которые удовлетворяют фильтру $X' = F(X)$. Пусть $X^{"} = S(X')$ - упорядоченное множество объектов, полученное путем сортировки S из множества X'. Объекты множества X описываются множеством свои атрибутов: $x_i = \{a_j^i, j = \overline{1, M}\}$.

Определим множество атрибутов, участвующих в фильтре F как $f = \{a_l\ , 1 \le l \le M\}$, для которых в заданном фильтре определены допустимые значения. При этом объекты должны быть отсортированы (упорядочены) по нескольким атрибутам $s = \{a_k\ , 1 \le k \le M\}$. Определим множество атрибутов, которые необходимы для представления пользователю, как $r = \{a_m, 1 \le m \le M\}$

Предлагается использовать следующую последовательность операций, отличающуюся от традиционной последовательности раздельной выборкой атрибутов.

1. Определим объединенное множество $t = f \cup s$, как множество атрибутов, которые должны быть выбраны в первой операции чтения.

2. Выполним в одной операции выборку по фильтру и сортировку $X^{"} = S\big(F(X)\big)$ только с теми атрибутами, которые определены в множестве t.

3. Выполним выбору необходимых пользователю атрибутов r для объектов из множества $X^{"}$.

4. Возвратим ограниченный набор данных для представления на одной странице.

Предлагаемый алгоритм выборки позволяет повысить производительность, как показано в Табл. 6, в 1,2 раза. В эксперименте использовались запросы вида Запрос 4.

2.7. Вертикальный вид данных

Еще одним подходом, позволяющим повысить производительность работы систем, на основе модели EAV, является подход, который подразумевает извлечение данных из модели EAV в вертикальном виде с последующей работой без транспонирования в горизонтальный вид. Для получения горизонтального представления, когда объекты имеют атрибуты, принимающие множественное значение, обычно используется декартово произведение значений множественных атрибутов, что увеличивает количество строк в выборке. Для получения же вертикального представления не требуется выполнения дополнительных операций и не увеличивается размер возвращаемых данных, что позволяет не потерять в производительности запросов. Представление данных пользователю в удобном виде предполагается в этих случаях выполнять на стороне приложения.

Табл. 6. Сравнение производительности запросов к EAV модели

№	Запрос	EAV с традиционной выборкой атрибутов (мс)	EAV с раздельным чтением атрибутов (мс)
1	Без совпадения атрибутов	24530	20317
2	С частичным совпадением атрибутов	21293	18458

Результаты сравнения приведены в Табл. 7. В ней модель EAV в случае возвращения данных в вертикальном виде имеет более высокую производительность, чем реляционная модель. Эксперименты проводились с запросами вида Запрос 4.

Табл. 7. Сравнение производительности при вертикальном и горизонтальном представлении выборки

Реляционная модель, табличный вывод данных (мс)	EAV– горизонтальный вывод данных (мс)	EAV– вертикальный вывод данных (мс)
19967	24122	19453

2.8. Построение индексов

Большое число атрибутов понятия приводит к проблеме построения индексов без привлечения программистов в реляционной модели данных.

Индексы – это объекты базы данных, используемые для повышения производительности поисковых запросов. В реляционной модели индекс состоит из столбцов таблицы и указателей на соответствующие строки таблицы. Индексы содержат один и более столбцов,

Табл. 8. Сравнение реляционной модели с ограниченным набором индексов и модели EAV в миллисекундах (среднее значение для одного запроса)

Запрос	Реляционная модель (мс)	Модифицированная модель EAV(мс)
Запрос 1 с индексом по атрибуту фильтрации (для реляционной модели)	1.1	1.9
Запрос 1 без индекса (для реляционной модели)	1070	1.9
Запрос 2 без индекса (для реляционной модели)	980	1.8
Запрос 2 с индексом по одному из двух атрибутов фильтрации (для реляционной модели)	12.4	1.8

но число столбцов в индексах ограничено. Индексы строятся под часто используемые запросы, и в большинстве ситуаций невозможно иметь все необходимые индексы в силу того, что большое число индексов приводит к потере производительности в запросах, которые модифицируют данные.

Индексы строятся администраторами баз данных и требуют высокой квалификации, при этом очевидно, что в процессе жизненного цикла КИС, количество объектов и их атрибутов меняется и, это приводит к необходимости расширять индексы, что не всегда возможно (и в силу отсутствия квалификации, и в силу ограниченности числа индексов).

Если в КИС используются механизмы, обеспечивающие масштабируемость данных, отличные от модели EAV, то результаты могут значительно проигрывать по производительности модели EAV из-за отсутствия необходимых индексов. В качестве примера мы использовали данные из системы «Флагман» («http://ru.wikipedia.org/wiki/Флагман_(ERP-система)»), в которой есть реляционная таблица DOC_ITEM, имеющая 240 полей и всего 8 индексов. Нами проведено тестирование на данных в объеме 2 млн. записей в этой таблице. Результаты представлены в Табл. 8. Для экспериментов используются запросы 1 и 2, в которых для реляционной модели добавляются индексы. Эти же самые данные использовались в модели EAV по тем индексам, которые настроены первоначально и больше не меняются, так как физическая структура таблиц не претерпевает изменений в процессе жизненного цикла.

Как видно из Табл. 8, при наличии необходимых индексов для атрибутов, используемых в фильтрации, производительность реляционной модели лучше, чем модели EAV в 1.7 раз, что соответствует выше рассмотренным экспериментам. При неполном покрытии атрибутов индексами модель EAV начинает опережать реляционную модель по производительности. При полном отсутствии индексов по атрибутам производительность реляционной модели хуже, чем модели EAV более чем в 500 раз.

В модели EAV всегда есть индексы и они не меняются в процессе жизненного цикла. Поэтому в ситуациях, когда структура данных меняется в процессе жизненного цикла, а объекты имеют большое число атрибутов, модель EAV будет обладать значительным преимуществом перед реляционной моделью.

Заключение

В работе рассмотрена модель EAV, позволяющая разрабатывать адаптируемые интегрируемые информационные системы с длительным жизненным циклом. Показано, что при использовании специализированных подходов модель EAV позволяет обеспечивать производительность, сопоставимую с производительностью реляционной модели, а в некоторых случаях значительно ее превосходящую.

Для получения максимальной производительности помимо стандартных оптимизационных мероприятий можно рекомендовать:

1. использование модели EAV1;
2. создание таблиц EAV с двумя индексами по понятию, атрибуту, значению и по атрибуту, значению, понятию;
3. использование модифицированного подхода к построению запросов, уменьшающего количество операций ввода-вывода;
4. выравнивание статистики с целью минимизировать ручную оптимизацию и зависимость от разработчика;

5. возвращение данных в виде вертикальной таблицы;
6. использование алгоритма чтения данных с раздельной выборкой атрибутов.

Модель EAV используется в обобщенном репозитории КИС Владивостокского государственного университета экономики и сервиса (ВГУЭС) с 2003 г. В настоящее время в полной мере на этой модели базируются следующие системы и сервисы КИС ВГУЭС: хранилище полнотекстовых научных, видео и учебно-методических материалов, система управления электронным документооборотом, система управления контентом. В настоящий момент общий объем данных составляет около 25 млн. записей.

Модель EAV используют также многие другие системы КИС ВГУЭС в качестве промежуточного контейнера, обеспечивающего масштабируемость системы и независимость от изменяющихся источников первичных данных. Например, система рейтинговой оценки деятельности кафедр и преподавателей использует обобщенный репозиторий понятий для описания произвольного набора показателей деятельности кафедр и преподавателей, что позволяет постоянно менять систему показателей (и соответствующих показателям источников данных), не меняя кода программы [14, 15].

Литература

1. Nadkarni P. The EAV/CR Data Model. http://ycmi.med.yale.edu/nadkarni/eav_CR_contents.htm
2. Nadkarni P., Marenco L., Chen R., Skoufos E.,Shepherd G., Miller P. Organization of heterogeneous scientific data using EAV/CR Representation//Journal of the American medical information Association.-1999. V.6, N 6 -.p. 478-493.
3. Chen R., Nadkarni P., Marenco L., Levin F., Erdos J., Miller P. Exploring performance issues for a clinical database organized using an entity-attribute-value representation//Journal of the American medical information Association.-2000.- #7 -.p. 475-487.
4. Тенцер А. База данных - хранилище объектов// КомпьютерПресс. – 2001.- №8.
5. Змеев О., Моисеев А. Сравнительный анализ некоторых методов O-R-преобразования// Вестник Томского государственного университета. -2003. –N 280. -с. 263-271.
6. EAV vs Row Modeling. Тест производительности на PostgreSQL. http://valv.ru/eav-vs-row- url:modeling-test-proizvoditelnosti-na-postgresql.html
7. Razan P., Abu Sayed Latiful Hoque. Optimized entity attribute value model: a search efficient representation of high dimensional and sparse data//Open Access, Open Review Journal.-July 6, 2011.
8. Варшевский В. Структура базы данных Magento: знакомство с EAV. http://gurumagento.com/struktura-bazy-dannyx-magento-znakomstvo-s-eav.htm
9. The EAV/CR Model of Data Representation http://ycmi.med.yale.edu/nadkarni/eav_cr_frame.htm
10. Shaker H. El-Sappagh, Samir El-Masri, A. M. Riad, Mohammed Elmogy. Electronic Health Record Data Model Optimized for Knowledge Discovery// International Journal of Computer Science Issues, Vol. 9, Issue 5, No 1, September 2012. pp. 329-338.
11. Prakash M., Nadkarni P., Brandt C. Data extraction and ad hoc query of an entity-attribute-value database//Journal of the American Medical infomatics association.- Vol.5. – Num. 6, 1998. – pp. 511-527.
12. Морис Льюис. Настройка SQL Server 6.5 на обработку запросов с высокой производительностью// SQL Server Magazine ONLINE, No 1, 2000 http://www.osp.ru/data/www2/win2000/sql/2000/01/007.htm
13. Рогов Е. SQL или PL/SQL? — взгляд с точки зрения производительности. http://egorius.hardsign.com/tmp/sql_or_plsql.pdf
14. Архипова Е.Н., Кононова О.В., Крюков В.В., Шахгельдян К.И. Автоматизация рейтинговой оценки деятельности преподавателей//Университетское управление. - 2010.-№5. – с.51-62
15. Архипова Е.Н., Крюков В.В., Шахгельдян К.И. Автоматизация рейтинговой оценки деятельности учебного подразделения вуза // Университетское управление: практика и анализ, №1, 2012. – с. 80-90.

Гмарь Дмитрий Викторович. Руководитель Центра информационно-технического обеспечения Владивостокского государственного университета экономики и сервиса (ВГУЭС). Окончил ВГУЭС в 2002 году. Автор 19 печатных работ. Область научных интересов: разработка информационных систем, базы данных, облачная инфраструктура. E-mail: dimer@vvsu.ru

Игнатова Юлия Александровна. Ведущий программист Центра информационно-технического обеспечения ВГУЭС. Окончила Дальневосточный государственный технический университет (ДВГТУ им. Куйбышева) в 2006 году. Автор 5 печатных работ. Область научных интересов: разработка информационных систем. E-mail: Yuliya.Ignatova@vvsu.ru

Цуранов Эдуард Владимирович. Ведущий программист Центра информационно-технического обеспечения ВГУЭС. Окончил ДВГУ в 1996 году. Область научных интересов: базы данных. E-mail: Eduard.Tsuranov@vvsu.ru

Шахгельдян Карина Иосифовна. Директор института Информационных технологий ВГУЭС. Окончила ДВГУ в 1989 году. Доктор технических наук, доцент. Автор 125 печатных работ. Область научных интересов: разработка информационных систем, интеграция данных и приложений, корпоративная информационная среда предприятия.
E-mail: carinash@vvsu.ru