УДК 004.652
С.С.Забара, И.В.Изварин
Открытый международный университет развития человека «Украина»

СЕМАНТИЧЕСКИЕ БАЗЫ ДАННЫХ ДОКУМЕНТОВ С РАЗЛИЧНЫМ РЕКВИЗИТНЫМ СОДЕРЖАНИЕМ

В статье рассматриваются проблемы связанные с хранением и поиском документов с различным реквизитным наполнением и предлагаются пути их решения. Детально показаны возникающие технические проблемы при добавлении в базу данных новых типов документов и проанализировано время, затрачиваемое пользователем на составление условий поиска в реальном приложении.

Ключевые слова*: База данных, документ, реквизиты документа, семантическая составляющая, поиск информации, условия поиска, поиск в базе данных.*

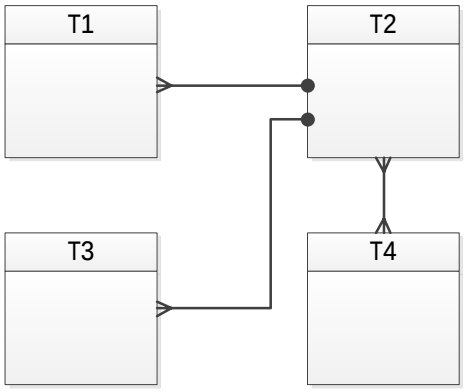
Определение: Докуме́нт (от documentum — образец, свидетельство, доказательство, любой материальный объект) — облеченный в письменную форму акт, удостоверяющий наличие фактов определенного значения.

Определение (толковый словарь Д.Н. Ушакова): РЕКВИЗИ́Т, реквизита, ·муж. (от ·лат. requisitum - потребность). — Совокупность формальных элементов в составе сделки или документа, отсутствие которых лишает сделку или документ юридической силы (офиц. юр.). Вексельный реквизит [2].

Современная производственная деятельность человека во многом связана с обработкой информации. Существует информация в виде таблиц, списков, то есть такая, которую можно просто систематизировать и с которой можно удобно работать. Однако, есть и такая информация, которая не поддается простому структурированию и систематизации. Примером такой информации могут служит: разрешительные государственные документы (лицензии), сложные договора, инвестиционные договора, юридические документы, договора аренды и прочее.

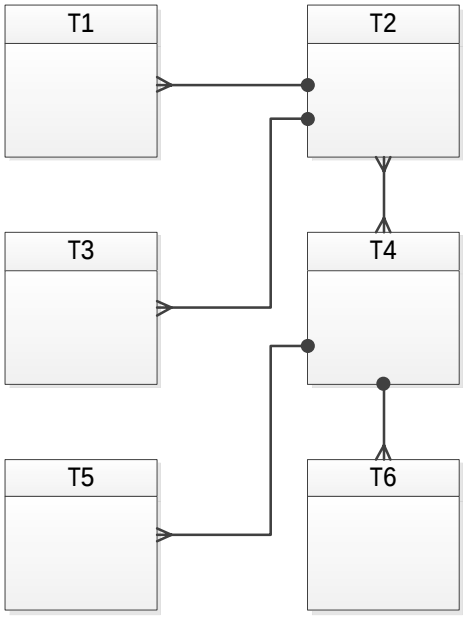
Даже в таких документах можно идентифицировать информационные объекты, которые связаны между собой по смыслу. Эти объекты называются реквизитами документа. Такого же рода информацию можно выявить и среди документов различных типов. Например, лицензия на ввоз определенной группы товаров, выданная юридическому лицу и договор аренды скадских помещений для того же юридического лица, которые коррелируют (или не коррелируют) между собой по срокам действия лицензии и договора.

Рассмотрим поиск информации в базе данных, построенной по классическому реляционному методу. Предположим, у нас есть группа таблиц для хранения информации по какому-либо одному типу документов (договор аренды, лицензия на вывоз труб). Для выполнения операции поиска необходимо реализовать оператор SELECT языка SQL [1,4] с соответствующим условием (или условиями) поиска WHERE поле1 ОП значение 1 И/ИЛИ ... полеN ОП значениеN. Такой оператор вернет группу записей, удовлетворяющих условиям поиска (смотрите рисунок 1). Введение в базу данных нового типа документа (договор лизинга, лицензия на вывоз текстиля) приведет к изменению структуры базы данных. В простейшем случае будет добавлена новая группа таблиц. В более сложном случае к существующей структуре будет добавлены новые таблицы, а сама существующая структура будет изменена. В любом случае после такого изменения структуры базы данных нужно будет радикально переделать оператор SELECT для выполнения поиска (смотриет рисунок 2).




```
SELECT
    T1.*
    T2.*
    T3.F...
FROM
    T1 JOIN T2
    AND T3 JOIN T2
WHERE
    T1.F1 = Value1
    AND T1.F2 > Value2
    AND T2.F1 < Value3
    ...
```

Рисунок 1. Структура базы данных для простого документа одного типа и соответствующий ей оператор SELECT для поиска требуемой информации.




```
SELECT
    T1.*
    T2.*
    T3.F...
    T4.*
    T5.*
    T6.*
FROM
    T1 JOIN T2
    AND T3 JOIN T2
    AND T2 JOIN T4
    AND T4 JOIN T5
    AND T4 JOIN T6
WHERE
    T1.F1 = Value1
    AND T1.F2 > Value2
    AND T2.F1 < Value3
    AND T5.F1 = Value4
    OR T6.F1 BETWEEN Value5 AND
Value6
    ...
```

Рисунок 2. Структура измененной базы данных для документов двух типов и соответствующий ей оператор SELECT для поиска требуемой информации.

Если для нового типа документа в структуру базы данных будут добавлены новые таблицы, то потребуется несложное изменение оператора SELECT: в группе ... будут перечислены новые поля, а в условие поиска будут добавлены новые условия, которые работают с новыми таблицами.

Если же для нового типа документа будет изменена существующая структура базы данных и добавлены новые таблицы, то изменение оператора SELECT будет более сложным, поскольку он должен будет учитывать логику, содержащуюся в новой структуре базы данных. Если для первого случая достаточно работы опытного программиста, то во втором случае может понадобиться вовлечение архитектора базы данных и бизнес аналитик. То есть реализация задачи поиска усложняется.

При увеличении числа типов документов до нескольких десятков комплексность структуры базы данных и реализация операций поиска становятся сложными в реализации, сопровождении и внесении изменений.

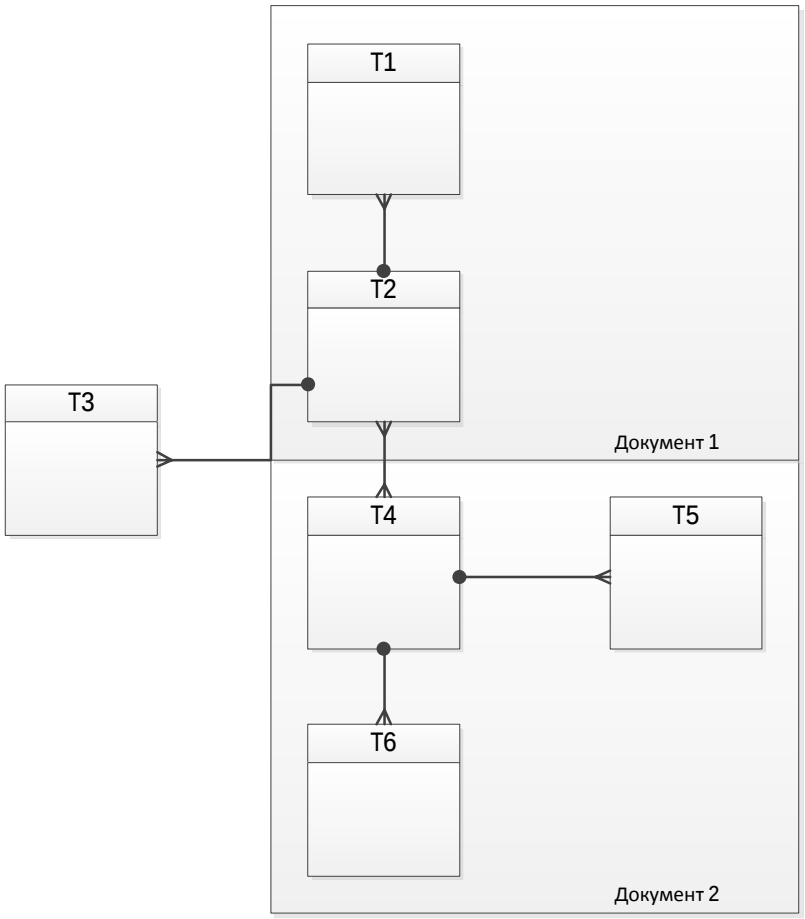


Рисунок 3. Структура базы данных для документов двух типов с отдельным хранением общей для документов информации.

Рассмотрим более детально процедуру поиска.

Пусть имеется множество документов (одного типа). Необходимо найти подмножество данного множества документов, которое бы отвечало условиям поиска. Для этого формируется булево выражения условия, в котором принимают участия реквизиты, их значения и булевы операнды. После того как сформировано условие поиска, необходимо обратиться к хранилищу данных и при помощи сформулированного выражения выполнить выборку документов, отвечающих (удовлетворяющих) поставленным условиям. Сложность (или простота) данной операции зависит от способа хранения документов (конкретно информации о значениях всех реквизитов, входящих в информационную составляющую документов).

Ведь в общем случае процедуру поиска можно сформулировать следующим образом: пользователю необходимо получить некоторую информацию о содержании документов, в которых значение определенного реквизита (или нескольких реквизитов) отвечают определенным образом сформулированным условиям.

Формализуем данное положение:

1. Выделить из множества реквизитов подмножество, по которым будут задаваться условия поиска требуемых документов. Выделение осуществляется с помощью множества семантических параметров реквизитов. Данное действие выполняется исключительно пользователем. Пользователь получает перечень семантических значений реквизитов и, в зависимости от характера стоящей перед ним задачи, выбирает реквизиты, условия к значениям которых помогут выбрать нужное подмножество реальных документов, хранящихся в системе. Поэтому говорить о скорости выполнения данной операции можно, на наш взгляд, только с точки зрения трудоемкости. Причем основным параметром будет количество элементов множества семантических параметров реквизитов, которые анализируются пользователем с целью формирования условий поиска;

2. Задать условия для значений выделенного подмножества реквизитов.
3. Сформировать булево выражение общего критерия поиска требуемого подмножества документов;
4. Выделить из имеющегося множества реальных документов (данного типа) подмножества, отвечающие условиям подмножества реквизитов, сформированном на втором этапе;
5. Выполнить булево выражение, используя в качестве аргументов полученные подмножества документов. В результате получим искомое (исходя из общего критерия поиска) подмножество документов. Как правило, данный результат не является конечной целью задачи поиска. Необходимо, на основе значений реквизитов полученного подмножества реальных документов, сформировать отчетный документ. В простейшем случае он может представлять собой таблицу, в которой столбцы являются реквизитами документа, а строки – найденными документами. Их пересечение – значения выбранных пользователем для отчетного документа реквизитами полученного подмножества хранимых документов;
6. Сформировать множество реквизитов выделенных документов. В случае, когда рассматривается один тип документов, то данный пункт выполнять не нужно, так как множество (перечень) реквизитов выделенных документов совпадает со всем множеством реквизитов документа типа j;
7. Определить подмножество реквизитов, значение которых требуется выводить в отчетный документ.
8. Сформировать отчетный документ.

Программный инструмент, с помощью которого пользователь может производить поиск требуемой информации, имеет две стороны: одна сторона обращена к хранилищу информации, а другая сторона обращена непосредственно к пользователю. Для того, чтобы подобный программный инструмент мог производить эффективный поиск в различных типах документов он должен быть основан на гибкой и бстрой технической реализации хранилища реквизитной информации и семантической разбивки реквизитов, а также предоставлять удобный интерфейс пользователя для построения условий поиска. Интерфейс также должен быть основан на семантической составляющей реквизитного состава документа.

В настоящее время сложность программных инструментов, которыми пользуется человек, очень велика и многие такие инструменты пересекают ту грань, когда им можно эффективно и просто пользоваться. Для использования подобных инструментов зачастую необходимо пройти специализированное обучение, которое затрагивает только аспекты самого инструмента и не затрагивает аспекты прикладной области использования – считается, что обучаемый в этой области уже эксперт. Если представить себе задачу составления запроса для поиска документов пяти типов, каждый из которых содержит до 10 параметров, то составление запроса из комбинации 50-ти различных параметров соединенных различными логическими условиями предаставляет собой сложную задачу для человека.

Рассмотрим сценарии работы пользователя по поиску документов удовлетворяющих определенным критериям [5].

Первый сценарий: поиск докуметов в базе данных документов одного типа. Для того, чтобы пользователь мог составить условия поиска создается специальная форма, которая соедржит все параметры поиска (реквизиты документа), с возможностью задания условия отбора и граничного значения. Например, реквизит «Дата подписи» с возможностью задания условий, «до», «после», «между» и граничным значение (или двумя граничными значениями). Такая форма уже сама по себе достаточно сложна, поскольку содержит, по крайней мере, по два управляющих элемента интерфейса на каждый реквизит документа. То есть в случае документа с 10 реквизитами форма задания условия поиска содержит не менее 20-ти элементов управления. Кроме этого, нужно учесть, что отдельные условия поиска соединяются между собой логическими операторами «И» и «ИЛИ» для формирования окончательного булевого выражения.

Такая форма представляет собой исключительно техническое решения для составления булевых выражений в терминах базы данных и приведенное к терминологии пользователя (непосредственное отображение интерфейсного термина «Дата подписи» в поле таблицы базы данных f_dateOfSignature). Такое техническое решение существует в техническом и технологическом пространстве и не существует в смысловом пространстве пользователя. Налицо явный разрыв: предлагаемое решение является техническим решением, продиктованным

современным развитием информационнх технологий, для нужд пользователя, а не прикладным решением в терминах и пространстве пользователя с использованием информационных технологий в качестве инструмента для создания таких решений.

Второй сценарий: поиск документов в базе данных документов разных типов. В данном случае у нас может быть уже несколько форм, по одной на каждый тип документа, и в некоторых из них присутствовать реквизит «Дата подписи», а может быть и так, что будет созана одна специальная форма с общими реквизитами (на которой и будет присутствовать реквизит «Дата подписи») и отдельные упрощенные формы на каждый тип документа. Построение булевого выражения условия поиска в этом случае становится еще сложнее. Фактически пользователь теперь превращается в технического специалиста для построения выражений и полностью отключается от своей первоочередной прикладной задачи поскольку ему теперь приходится мыслить в терминах выражений: «больше», меньше», «между», «И», «ИЛИ», «НЕ». Однако нужно помнить, что пользователю нужно «Отобрать все документы подписанные Ивановым И.И. до 1 мая 2010 года» или «Выбрать список всех документов клиентом в которых значится компания 'АБВ' и она ввозила товары группы '1123' по ТВЭД». Здесь тоже виден явный разрыв между предлагаемым техническим решением и смыслом тех операций, которые хочет выполнить пользователь.

Вывод: внесение изменений в существующее классическое решение в виде добавления нового типа документа приводит к созданию еще более технического решения основанного на современных информационных технологиях и вносит еще больший разрыв между предлагаемым решением и тем, что ожидает пользователь.

Оценим время, которое необходимо пользователю для построения условия поиска в случае классического реляционного подхода [3]. Время, необходимое для построения условия посика можно представить в виде следующей формулы:

$$T = \sum_{i=1}^{N_r} \left(t_{lookup} + t_{accept}\right) + \sum_{i=1}^{N_a} \left(t_{condition} + t_{value} + t_{relation}\right) - t_{relation},$$

где,

T — общее оценочное время, необходимое для составления условия поиска;

N_r — общее число реквизитов документа;

N_a — число реквизитов документа, принятое к дальнейшему использованию в построении йсловия поиска;

t_{lookup} — время, необходимое для визуального поиска і-го реквизита на интерфейсе;

t_{accept} — время, необходимое для принятия решения о том, будет ли включен данный реквизит в условие поиска или нет;

$t_{condition}$ — время, необходимое для построения условия (больше, меньше, равно, не равно, больше или равно, меньше или равно, между, и др.) для данного реквизита;

t_{value} — время, необходимое для задания значения используемого в условии отбора;

$t_{relation}$ — время, необходимое для построения булева условия объединения (И, ИЛИ) соседних условий отбора.

Хотя точная численная оценка требует проведения эксперимента по замеру времени на каждую операцию, мы можем дать предварительную численную оценку основываясь на замерах, произведенных соавтором статьи при составлении условия поиска должников банковских кредитов в одной из прикладных программ поддержки коллекторской деятельности:

$N_r = 75$ — общее число реквизитов в системе;

$N_a = 5 ё 10$ — число реквизитов, которые участвуют в условии поиска;

$t_{lookup} = 2 ё 5s$ — среднее время, затрачиваемое на поиск и восприятие реквизита;

$t_{accept} < 1s$ — на решение о принятии данного реквизита в условие поиска составляет менее одной секунды;

$t_{condition} = 1 ë 2s$ — условие отбора для данного реквизита;

$t_{value} = 5 ë 10s$ — это время зависит от сложности значения (дата, строка, число) и количества значений (в случае условия «между» необходимо ввести два значения);

$t_{relation} = 1$ — в простом случае.

Тогда общее время будет иметь следующую численную оценку:

$$T = 75 \cdot \left(\frac{2}{5} ë 5 + 1\right) + \frac{5}{10} ë 10 \cdot \left(\frac{1}{2} ë 2 + \frac{5}{10} ë 10 + 1\right) =$$
$$= \frac{2}{5} 25 ë 450 + \frac{3}{5} 35 ë 120 =$$
$$= 337 + \frac{3}{5} 35 ë 120 =$$
$$= \frac{3}{5} 372 ë 457$$

Отсюда видно, что основное время затрачивается на просмотр реквизитов с целью отбора тех, которые участвуют в формировании условия поиска, и даже в среднестатистическом случае это значение достаточно велико. (Только эксперт, или специалист, достаточно долго проработавший с прикладной программой, знает расположение требуемых реквизитов на интерфейсе и может снизить это время.)

Для реализации описанных выше требований необходимо:

Во-первых, необходимо разработать логическую модель базы данных для хранения документов с различным реквизитным наполнением, которая бы поддерживала возможность расширения новыми документами, и возможность эффективного семантического поиска, а также доказать, что разработанная модель позволяет разрешить описанные выше проблемы.

Во-вторых, разработать физическую модель сбазы данных оответствующую логической модели на основе реляционной, объектной или нереляционной СУБД;

В-третьих, разработать пакет низкоуровневых методов (функций) для работы с данной схемой;

В-четвертых, разработать механизмы составления запросов для поиска информации;

В-пятых, разработать интерфейс пользователя, позволяющий минимизировать сложности, связанные с формированием поискового запроса.

1. Дейт К. Дж. Введение в системы баз данных.: Пер с. Англ. – 6-е изд. – К.: Диалектика, 1998. – 784 с.: ил.
2. Толковый словарь русского языка: В 4 т. / Под ред. Д. Н. Ушакова. Т. 1. М., 1935; Т. 2. М., 1938; Т. 3. М., 1939; Т. 4, М., 1940. (Переиздавался в 1947-1948 гг.); Репринтное издание: М., 1995; М., 2000.
3. Christof Ebert, Reiner Dumke. Software Measurement. Springer-Verlag Berlin Heidelberg 2007.
4. Foundations and Trends in Databases. Volume 1 Issue 2, 2007. now Publishers Inc. PO Box 1024 Hanover, MA 02339, USA.
5. User experience re-mastered: your guide to getting the right design/edited by Chauncey Wilson. Morgan Kaufmann Publishers, USA, 2010.