УДК 681.31

Модель проблемно-ориентированной запросной системы

П.А. Завьялов

В статье рассматривается модель предметно-ориентированной запросной системы к данным периодической отчетности. Производится классификация запросных систем. Описана формализованная модель данной системы. Приводится описание программной реализации предложенной модели.

i. Введение

Технологию работы в информационных системах принято разделять на две группы:

- Оперативная работа с данными,
- Аналитическая работа с данными.

Информационные системы, предназначенные для оперативной работы с данными, называют OLTP-системами. А системы, предназначенные для аналитической работы, – OLAP-системами.

В соответствии с решаемыми задачами различаются и принципы хранения данных в информационных системах. OLTP-системы требуют нормализованной структуры данных [1] в целях устранения хранения и обновления избыточной информации. А OLAP-системы используют ненормализованную структуру данных для хранения агрегированных значений. Четкую грань между указанными видами систем не всегда удается провести. Но существует несколько тестов, определяющих требования для OLAP-систем, такие как, тест FASMI [2] или 12 правил Кодда для OLAP-систем [1,2].

Способы взаимодействия пользователя с информационной системой представлены на рисунке 1.

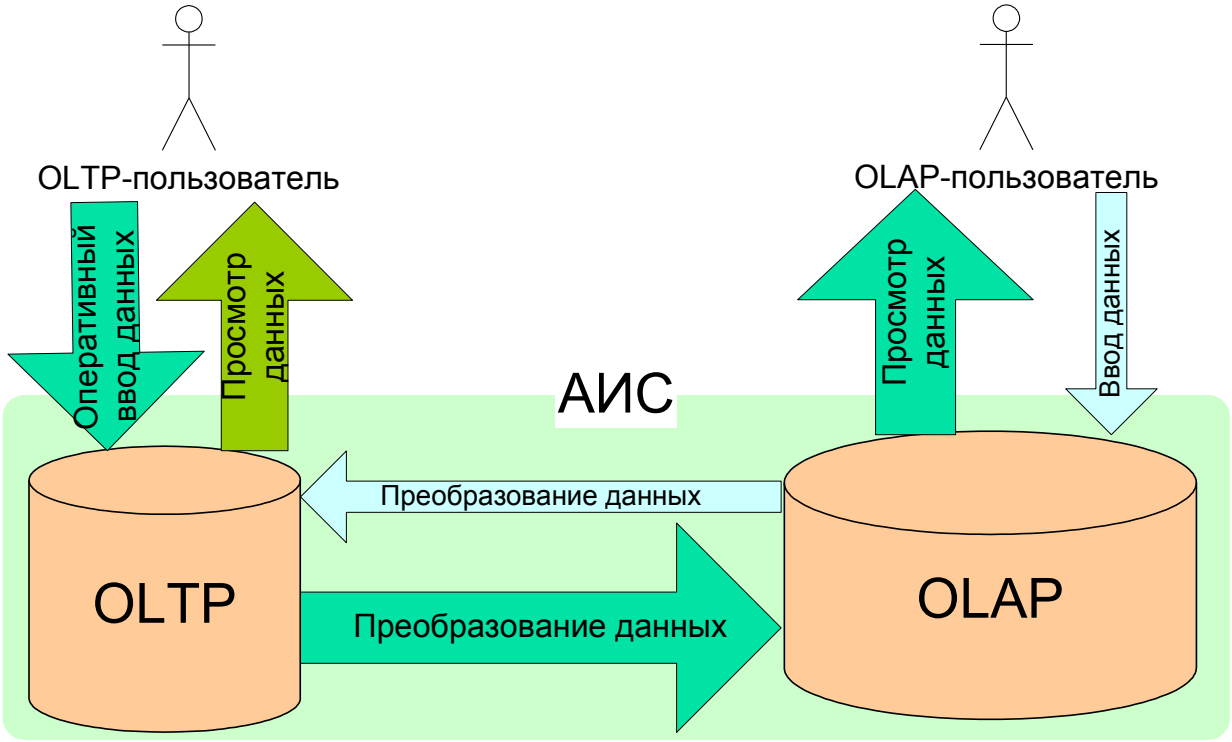


Рис. 1. Способы взаимодействия пользователя с информационной системой

Как показано на рисунке, в OLTP-системах осуществляется постоянный ввод относительно небольших порций данных, а также поиск данных и работа с каждой сущностью отдельно. В OLAP-системах ввод данных осуществляется преимущественно большими порциями за целый отчетный период. В процессе загрузки информации в базу данных осуществляется их преобразование в вид, удобный для проведения аналитической работы. Наиболее частыми операциями в OLAP-системах являются отбор и просмотр исходных и агрегированных данных.

Необходимость преобразования данных при их загрузке в базу данных OLAP-системы обусловлено различными способами их хранения.

Например, при копировании данных из учетной системы денежных поступлений крупного магазина в аналитическую базу данных, предназначенную для анализа спроса покупателей, копируются не сами первичные данные (накладные, проводки и т.д.), а уже агрегированные данные по дням, месяцам, годам и т.д.

Возможно и обратное преобразование данных из OLAP-системы в OLTP-систему. Например, бюджетирование организации может проводиться в головном офисе на основе аналитической системы. Затем, на основе заложенных алгоритмов, бюджет может передаваться уже непосредственно в учетные системы отельных подразделений.

Таким образом, существует несколько подходов к построению и обработке данных в зависимости от бизнес-требований к информационной системе. Очень часто, в силу

экономических причин или бизнес-задач, обработка данных происходит в OLTP-системах, без переброски в OLAP-хранилище. Так же обрабатывается не весь массив данных, а только часть ее. Поэтому встает вопрос об отборе информации. Отбор информации производится не напрямую средствами СУБД, а через бизнес-приложение пользователя. Вследствие этого, в каждом бизнес-приложении реализуется какая-то запросная система. А в крупных комплексах – несколько систем.

В общем случае любую программу или подсистему большой системы, которая осуществляет выборку данных из СУБД, можно назвать запросной системой. Такие подсистемы мы будем называть примитивными. Эти подсистемы или программы могут быть весьма сложными по своей сути и реализации, но с точки зрения возможности задания критерия или возможности отбора данных они являются примитивными.

Другой крайностью реализации запросной системы является использование языков запросов к базе данных, таких, например, как языки SQL или QBE. Такой подход позволяет выбирать любую информацию из базы данных, однако, он имеет ряд существенных недостатков. Пользователь должен

- иметь достаточно хорошие знания языка запросов,
- знать структуру базы данных и способ представления данных;
- понимать реляционную алгебру;
- уметь получить и задать информацию, которая находится вне реляционной структуры, в случае если не вся информация находится в реляционных таблицах;
- владеть навыками программирования;
- если не вся бизнес-информация лежит в базе данных в явном виде, то пользователь должен уметь вычислять всю необходимую бизнес-информацию сам.

Несмотря на указанные недостатки, данный метод тоже находит применение при реализации сложных бизнес-требований.

Еще одним подходом является создание проблемно-ориентированного запросного языка, который позволил бы пользователю гибко создавать запросы и при этом оперировать не наименованиями полей базы данных, а объектами предметной области в виде, удобном для пользователя, являющегося специалистом в своей предметной области. Этому подходу также присущи некоторые недостатки, перечисленные выше. Однако специализированный предметно-ориентированный язык заметно проще использовать специалисту в предметной области.

Таким образом, запросные системы можно разделить по способу задания критерия на группы:

1. Примитивные запросные системы.
2. Простые запросные системы. К данной группе относятся простые формы поиска. Например, форма поиска всех машин в базе данных ГИБДД, удовлетворяющих некоторым условиям: цвету, марки, году выпуска и т.д. Список полей строго закреплен в форме и ограничен десятком, реже, несколькими десятками. Выходная информация так же слабо поддается настройке.
3. Иерархические запросные системы. В качестве условий могут задаваться несколько десятков условий. Причем каждое поле условия имеет свою иерархию. Выходная информация может достаточно гибко настраиваться. К данному виду запросных систем относятся различные OLAP-системы.
4. Сложные поисковые формы. К данным системам относятся сложные формы поиска, в которых есть возможность гибко задавать запросы и структуру выходной информации. Такие подсистемы часто оперируют уже с сотнями полями. Часто предоставляется возможность задавать простейшие операции с полями. Например, можно задавать арифметические произведения полей с последующим наложением условия на результат. В качестве примера можно привести запросную систему в ГИБДД, которая позволяла бы оперировать не только со всеми характеристиками машины (цвет, масса, марка, мощность и т.д.), но и с показателями связанных с ней сущностей (владельцы, водители и их данные и др.). Эта запросная система должна иметь возможность накладывать условия на показатели связанных с машиной сущностей. Такая запросная система будет оперировать уже не десятками, а сотнями показателей. Так же данная запросная система должна иметь возможность выводить показатели всех связанных с машиной сущностей. Данная запросная система должна выполнить, например, такой запрос: отобрать все машины марки «ОКА» 2000 года выпуска, у которых владельцы зарегистрированы вне города Москвы, обслуживались инспектором Ивановым, и у которых полный пробег, деленный на количество лет использования меньше 20000. В качестве результирующего списка вывести: фамилию, имя, отчество владельца, регистрационный номер машины и дату обслуживания Ивановым данной машины. То есть, система должна предоставить возможность задать арифметическую операцию, должна позволить объединить несколько условий и вывести результат в виде списка с колонками, заданными пользователем.
5. Запросные системы на основе предметно-ориентированного запросного языка. К данным запросным системам относятся системы, в которых запрос задается в виде текста

на предметно-ориентированном языке. То есть в качестве элементов, над которыми совершаются действия, выступают атрибуты и сущности предметной области.

6. Запросные системы, ориентированные на использование внутреннего языка запросов СУБД. Данные запросные системы можно рассматривать как надстройки над СУБД, выполняющие трансляцию запросов пользователя на язык понятный СУБД.

На мой взгляд, для промышленных систем наибольший интерес при проведении аналитической работы с данными могут представлять запросные системы типа 5 и 6. Такие системы относительно просты для освоения конечным пользователем и обладают гибкостью, сравнимой с обычными запросами на языке SQL.

С точки зрения реализации запросная система на основе специализированного предметно-ориентированного языка, является более простой, по сравнению со сложными поисковыми формами. Такая система предоставляет больше возможностей для расширения и масштабируемости.

Технологию работы с этими двумя типами систем можно разделить на следующие 3 этапа:

- Подготовка критериев для отбора данных
- Выполнение запросов по подготовленным критериям
- Обработка отобранных данных для анализа

В данной статье рассмотрим предметно-ориентированную систему построения запросов для сбора информации по данным периодической отчетности с учетом изменчивости структуры отчетных табличных форм.

ii. Постановка задачи

Рассмотрим построение запросной системы по данным форм отчетности, которые представляют собой табличную структуру. Предложенная структура данных описывает периодическую отчетность нижестоящей организации перед вышестоящей организацией, периодическую финансовую отчетность, налоговую отчетность.

В качестве проблем анализа отчетности можно выделить следующие моменты:

- Изменчивость структуры форм отчетности. Формы отчетности могут видоизменяться с течением времени. Могут добавляться или удаляться ячейки, колонки или листы. Часть ячеек могут разбиваться на несколько или, наоборот, несколько ячеек могут объединяться в одну. Однако при работе с данными необходимо иметь возможность обрабатывать документы, не зависимо от периода действия формы.

- Неполнота или не корректность ввода информации. В базе данных может содержаться не вся информация. Это может происходить из-за того, что данные либо подготовлены не корректно или их не успели ввести. Например, может быть введено суммарное значение, но не введены ячейки, на основе которых оно рассчитывается или суммы могут не совпадать. Данное явление часто встречается в крупных системах с большим количеством отчетов.
- Неопределенность цели. Аналитик не всегда может точно знать, какие и в каком объеме он хочет получить данные или какие алгоритмы он хочет применить к данным. Например, обрабатывая огромный массив отчетов, аналитик хочет выбрать около десятка интуитивно понятных ему отчетов. Изначально, он может применить самое простое условие и получить огромное количество отчетов. Затем он может постепенно применять дополнительные условия для их сокращения. Или наоборот, изначально не получить результата, а затем, убирая дополнительные условия, он будет увеличивать количество отчетов в результирующем списке. В конечном итоге он получит примерно то количество отчетов, которых он хочет обработать. В процессе построения запроса, аналитик может выяснить, что те условия, которые он предполагал, не применимы вследствие неполноты или не корректности исходных данных. Он может частично изменить требования для получения адекватного ответа от системы.
- Различность в интерпретации данных разными аналитиками. Еще одной проблемой для универсальной реализации анализа данных, может стать субъективная интерпретация данных аналитиком. Например, условие отбора отчетов, может быть сформулировано следующим образом: отобрать из базы данных налогоплательщиков, у которых не сдана налоговая отчетность. Но аналитики из разных отделов налоговой инспекции могут по-своему понимать фразу «не сдана налоговая отчетность». Для аналитика из отдела, который занимается налогом на прибыль, это может быть декларация по прибыли и бухгалтерский баланс. Для аналитика из отдела ЕСН, это может быть декларация по ЕСН, декларация по ЕНВД и бухгалтерский баланс. Как видно из примера, различные аналитики будут иметь различные списки налогоплательщиков «не сдающих отчетность». Более того, при различном анализе один и тот же аналитик может вычислять значение одного и того же показателя различными способами.

Таким образом, написать аналитическую систему, которая позволяла бы производить разносторонний анализ данных, без возможности создания сложных алгоритмов обработки данных, не представляется возможным.

Исходя из выше перечисленных ограничений и сложностей при создании запросной системы, формализованную модель этой системы можно представить в следующем виде:

четверка множеств: $M_L = \{T, X, B, S\}$,

где

T - дискретное время (время подачи отчетности);

X - множество объектов отбора, где объект отбора $x \in X$ представляет собой объединение множеств $x = \{i\} \cup \{D_{1,1}, D_{1,2}, D,..., D_{1,N(1)},..., D_{t_k,1}, D_{t_k,2},..., D_{t_k,N(t_k)}\}$, где D_{l_1,l_2} - таблица размерности $m_{t_j} \times m_{t_j}$, состоящий из элементов $d \in D_{l_1,l_2}$ i - идентификатор объекта x, $i \in I$;

B - множество объектов бизнес-логики (объектов, которыми оперирует аналитик);

S - множество интерпретаций данных, $s(d) \in S$; $b = s_t(\overline{D})$.

Необходимо построить функции перевода $F(\langle I, Dx \rangle, S') \to \langle I', B' \rangle$,

где

S' - множество интерпретаций работающего аналитика;

$I' \subset I$ - подмножество идентификаторов объектов отбора (т.е. набор объектов для анализа);

$B' \subset B$ - подмножество объектов бизнес-логики, необходимых для анализа объектов отбора.

Другими словами, задачу, на языке баз данных, можно сформулировать так: построить запрос, который бы позволил отобрать список строк, содержащих поля, необходимые для аналитика.

Для реализации сформулированной задачи, должны быть написаны следующие программные компоненты:

- компоненты управление запросами – набор компонент, которые позволяли бы выбирать, хранить, обновлять запросы и обеспечивающие общее управление запросами;
- компоненты формирования запроса – набор компонент, обеспечивающие дружелюбных и понятный пользователю интерфейс создания и редактирования запросов;
- компоненты выполнения результата – набор компонент, которые обеспечивают выполнение сформированного в системе запроса;
- компоненты форматирования результата выполнения – набор компонент, которые представят полученные данные в удобном для пользователя виде.

iii. Принципы реализации

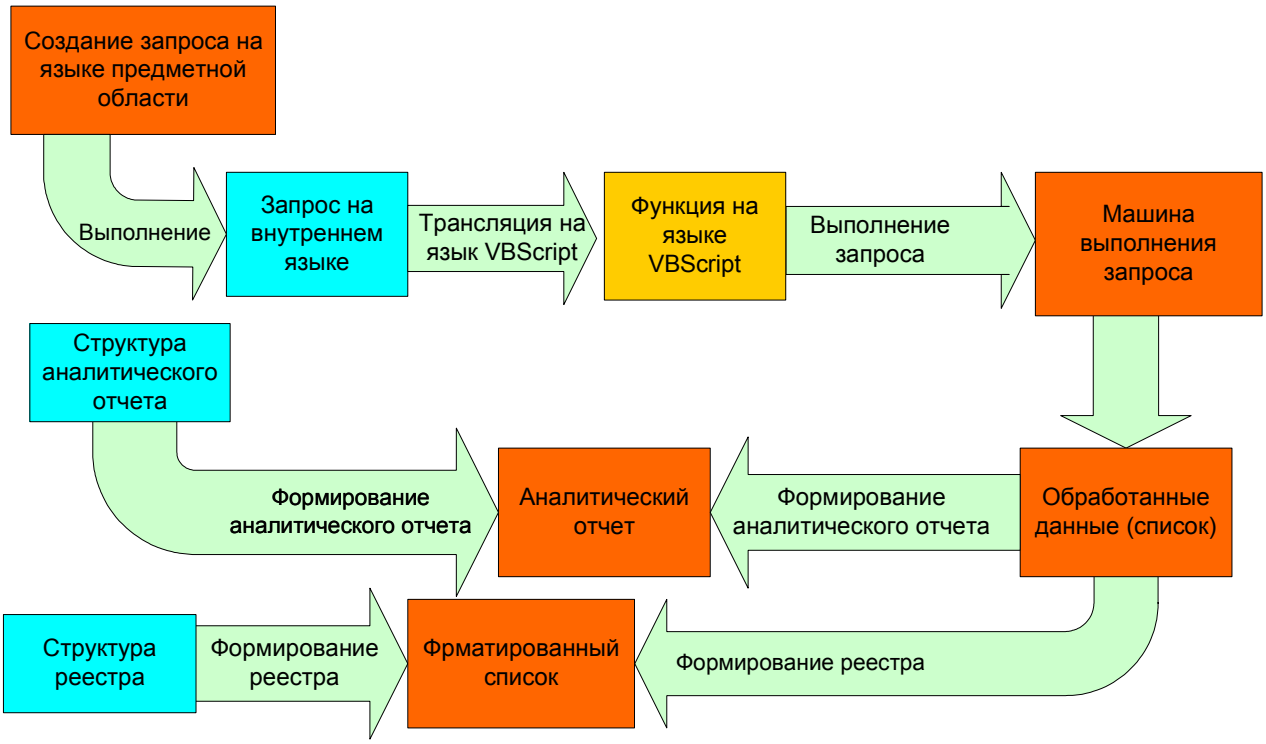
На программную реализацию предложенного подхода, могут налагаться дополнительные условия. Одним из существенных условий, может являться использование трехзвенной архитектуры при построении сложных корпоративных систем. При использовании данной архитектуры, основная масса вычислений происходит на уровне сервера приложений. Таким образом, часть бизнес-объектов сложно получить на уровне СУБД. Так как часть данных содержится только на уровне сервера приложений, то вычисления, выполняемые при отборе данных, также желательно проводить на уровне бизнес-логики и выполнять на сервере приложений [3].

При реализации данной запросной системы описанное условие было учтено следующим образом. Запрос, записанный на внутреннем языке, преобразовывался к языку VBScript, и затем интерпретировалась полученная на VBScript функция. Использование VBScript в качестве внутреннего языка запросов позволило использовать интерпретатор, написанный Microsoft, что облегчало задачу реализации системы. Еще одним преимуществом сделанного выбора явилось то, что язык VBScript позволяет использовать COM-объекты, реализующие бизнес-сервисы. Это позволило использовать COM-объекты системы при выполнении запроса.

Технология выполнения запроса разбивается на несколько этапов:

1. Формирование запроса пользователем.
2. Выполнение запроса.
   2.1. Декодирование запроса на внутренний язык исполнения (VBScript).
   2.2. Выполнение полученной подпрограммы на массиве исходных данных.
3. Форматирование данных.

Данные этапы проиллюстрированы на следующей схеме:



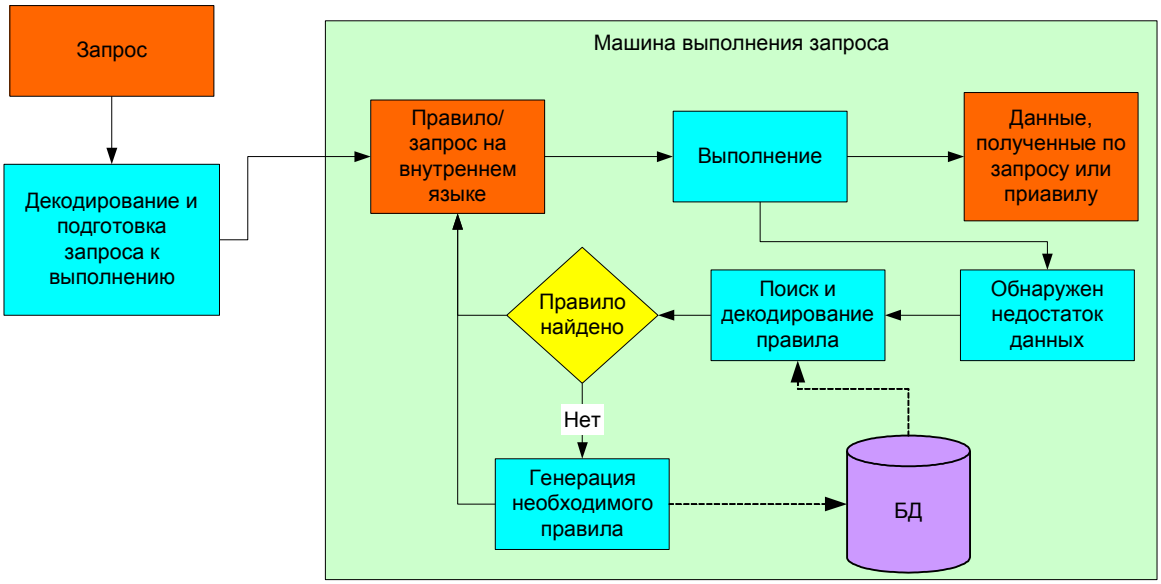
На представленной схеме оранжевым цветом помечены объекты, с которыми работает аналитик. Это: созданный аналитиком запрос, полученные списки, аналитические отчеты, сформированные на основе полученных списков и форматированные списки, которые можно предоставлять в другие отделы.

Голубым цветом помечены объекты, которые представляют настройки администратора системы или промежуточные результаты выполнения системы. То есть различные конфигурационные файлы для выдачи пользователю данных.

Желтым цветом показаны внутренние объекты подсистемы, которые используются для выполнения запросов.

Таким образом, после выполнения сформированного запроса, пользователь системы получит либо аналитический отчет, либо отформатированный список отобранных объектов.

Более подробно, последовательность выполнения запроса можно представить на схеме:



Запрос, сформированный пользователем, декодируется к внутреннему языку. Затем он выполняется. При выполнении запроса могут запрашиваться данные явно не описанные в тексте запроса. Данный недостаток данных разрешается следующим образом: происходит поиск недостающих правил и, если они находятся, то выполняется правило и продолжается выполнение запроса. При невозможности разрешить правило может происходить создание правила специальными компонентами. На выходе мы получаем сформированные по запросу данные.

iv. Выводы

В статье предложена формализованная модель запросной системы по данным периодически изменяющейся отчетности, имеющей табличное представление. Предложены технология, обобщенные алгоритмы работы и принципы реализации запросной системы.

Список литературы

1) Коннолли К., Брегг. М., «Базы данных: проектирование, реализация и сопровождение. Теория и практика, 2-е издание» Т.: издательский дом «Вильямс», 2000 г., - 1120 с.
2) Найгель Пендс. Что следует понимать под термином OLAP? - http://www.olap.ru/basic/fasmi.asp
3) Виноградов В.И., Завьялов П.А. Оптимизация иерархических запросов в информационных системах при работе в трехзвенной архитектуре. Сб. «Теоретические вопросы вычислительной техники и программного обеспечения. Межвузовский сборник научных трудов/ Московский государственный институт

радиотехники, электроники и автоматики (технический университет) ».- М., 2003 -с.224.

Сведения об авторе

Завьялов Павел Александрович, аспирант кафедры Математическая кибернетика Московского авиационного института (государственного технического университета).
e-mail: pavelz@km.ru