АВТОМАТИЗИРОВАННАЯ СИСТЕМА ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ ДЛЯ ПОИСКА ОПТИМАЛЬНЫХ РЕШЕНИЙ

Юрков Д. А.
Восточно-украинский национальный университет им. В.И. Даля,
г. Луганск, Украина

Abstract
Yurkov D.A. Automated system for preliminary processing of information for optimal desisions searching. *In this work we consider methods and principles of preliminary processing of information for realization of an information-searching system in a dataset with different structure. A solution for a problem, that a developer of such system encounters, is offered.*

Анализ состояния проблемы. В настоящее время перед разработчиками программного обеспечения, работающими в сфере автоматизации управленческой деятельности ставятся задачи не только учетного характера (когда и сколько продукции отгружено или получено, кому и за какую цену и т.п.), но и такие задачи, как

- Поиск оптимального решения по закупке продукции. При этом критериями оптимальности считаются требования руководителя организации, который основывается на личном опыте или сложившихся обстоятельствах.
- Возможность поиска решения по разнообразно представленной информации: различные форматы данных, названия и типы полей, кодировка символов и т. п.
- Достичь минимального времени поиска решения, независимого (или практически независимого) от объема перерабатываемой или анализируемой информации. При этом зачастую максимальное время поиска решения строго ограничено различными причинами (ожидание клиентов, психологические факторы и т. п.)
- Удобное и понятное представление полученного результата с возможностью управляемого переноса найденного варианта решения в рабочие документы организации.

Подобные задачи возникают практически всегда при необходимости работы с каталожной продукцией различных производителей и посреднических организаций (автозапчасти, фармакологическая продукция и т. п.). Необходимая для анализа информация содержит, как правило, сотни тысяч записей. Так как каталоги формируются многочисленными поставщиками и производителями для различных групп продукции, то объем перерабатываемой информации может быть несколько миллионов или даже десятков миллионов записей.

Постановка задачи. Реализация подобной информационно – поисковой системы (ИПС) сразу же наталкивается на проблему представления исходных данных, так как на практике сложилась ситуация, при которой в различных организациях созданы базы данных для внутреннего использования и, следовательно, имеют различную структуру. При этом отсутствует какая – либо практическая возможность заставить поставщиков придерживаться единого формата данных и их структуры. Таким образом, в первую очередь необходимо реализовать механизм работы с данными с заранее неизвестной структурой. Следующая проблема заключается в том, что поиске вариантов оптимального решения в базе данных критическим является время ответа. Требования пользователей ИПС – это время ответа не более 5 – 10 секунд вне зависимости от объема исходной информации. Кроме этого, система должна быть достаточно универсальной. Связано это прежде всего с тем, что подобные системы необходимы очень многим организациям, а разные пользователи предъявляют

различные требования к форме информационных запросов, при этом зачастую даже не мотивируя эти требования. И наконец, подобная система должна работать в сетях, не теряя при этом своих функциональных и эксплуатационных характеристик. Таким образом, необходимо создать систему, удовлетворяющую вышеперечисленным требованиям. Таким образом, выделено три основных задачи: представление исходных данных, ограничение времени поиска и возможность настраивать правила информационных запросов.

Результаты исследований. Для того, чтобы принять верное решение по структуре разрабатываемой информационно – поисковой системы, и в частности, базы данных, требуется анализ различных вариантов решения. Рассмотрим один из вариантов решения поставленных задач. Список предоставляемой поставщиками продукции в подавляющем большинстве случаев оформлен в виде файла в DBF – формате или же в формате электронной таблицы Microsoft Excel. В связи с этим можно выделить как минимум следующие основные проблемы:

- Разное наименование полей в DBF – файлах и разные типы данных.
- Невозможно точно определить, в каком поле находится какая информация. Под одним и тем же наименованием поля в различных файлах может находиться различная информация, которая может быть даже излишней для поиска решения.
- Файлы могут быть представлены в различной кодировке

Это приводит к тому, что невозможно создать единую форму запроса к любому из файлов, по крайней мере стандартными средствами. Для решения этих проблем принципиально возможно создавать конфигурационные файлы для каждого исходного файла. При этом в конфигурационных файлах необходимо хранить как минимум образец запроса для исходного файла, а также другую служебную информацию (соответствие полей, кодировка и т. д.). Однако, в данном варианте структуры базы данных невозможно обойти следующие явные ограничения:

- Необходимо реализовать не только поисковый алгоритм, но и другие вспомогательные алгоритмы работы с данными.
- Алгоритм поиска должен работать с каждым исходным файлом по отдельности, что требует отдельного открытия и закрытия файла с данными и конфигурационного файла, а эти операции сами по себе достаточно медленные. Это автоматически приводит к увеличению времени ответа системы. В случае, если количество исходных файлов несколько десятков, то время ответа становится недопустимо большим.
- Работать с различными форматами данных в данной концепции возможно только при реализации поискового алгоритма для каждого из форматов, что практически приводит к усложнению всей системы и увеличению времени на разработку и отладку программного обеспечения.
- Работать напрямую с DBF – файлами в принципе невозможно с использованием SQL - запросов, что также приводит к необходимости реализации специфических алгоритмов поиска и неоправданному, но необходимому усложнению системы.
- В случае, если пользователь системы выдвигает особые требования к виду информационного запроса (например, в коде продукции игнорировать пробелы или другие знаки), то при поиске необходимо проверять каждую запись в файле, не используя при этом существующие индексы. В этом случае время ответа системы становится неприемлемым даже в случае одного достаточно большого исходного файла.
- Нельзя простыми средствами поддерживать целостность базы данных. Элементарное перемещение исходного файла с одного места на другое приводит к потере части информации для поисковой системы и неверному или неполному ответу. Решением этой проблемы может быть перенастройка конфигурационного файла, но это само по себе не удобно.

- При использовании DBF – файлов нельзя работать по технологии клиент – сервер, а это означает, что в сетевом варианте использования системы будет неприемлемым не только время ответа на запрос, но и сама сеть будет излишне загружена, что в конечном итоге приведет к замедлению работы других сетевых приложений.

Из вышеизложенного можно сделать следующие выводы:

- К достоинствам данного подхода, безусловно, следует отнести простоту хранения исходной информации – она практически готовой приходит от поставщиков продукции в виде файлов.
- Недостатки, перечисленные ранее, принципиально не позволяют создать ИПС даже для сравнительно небольших объемов данных с требуемыми эксплуатационными характеристиками. Сложность реализации алгоритмов обработки данных требуют много времени на разработку ИПС, а чувствительность к ошибкам пользователя при конфигурировании системы могут требовать от пользователя специального образования и максимум внимания, что само по себе делает всю ИПС крайне не привлекательной.

В связи с этим можно утверждать, что подобный подход к реализации системы не имеет практического смысла. Другим вариантом решения поставленных задач может быть подход к проектированию базы данных, в которой вся исходная информация представлена в одной базе данных. Это сразу же снимает большинство проблем, изложенных выше, и позволяет получить дополнительные возможности. Однако в этом случае необходимо промежуточное звено для преобразования исходных данных. С одной стороны, это безусловно усложняет реализацию всей системы, но придает ей значительно большую гибкость при использовании и главное, решает следующие задачи:

- Реализация алгоритма поиска. Алгоритм разрабатывается один раз для результирующего набора данных, что значительно сокращает время разработки всей системы в целом.
- Время ответа системы. Время ответа ИПС определяется только характеристиками оборудования, на котором она установлена. Так как обрабатываемые данные хранятся в одной таблице, то появляется возможность индексации необходимых полей.
- Работа в сети. Для управления базой данных возможно использование любой СУБД, работающей по технологии клиент – сервер. Это автоматически решает проблемы разграничения доступа в систему и минимизации сетевого трафика.

Подобный подход также позволяет предложить решение проблемы по изменяемому виду информационного запроса. Для этого достаточно ввести ряд служебных полей, которые являются копией полей, по которым производится поиск, но без ненужной информации. Заполнение этих полей возможно на этапе преобразования данных, что позволяет не изменить скоростных характеристик поискового алгоритма. Таким же образом возможно решить проблему различной кодировки символов в исходных данных.

Практическая реализация. Разработанная система ориентирована на быстрый поиск и анализ информации, собранной из различных источников и имеющую разнообразную структуру. Обобщенная структура реализованной информационно – поисковой системы имеет вид, представленный на рисунке 1.

ИПС обладает следующими свойствами:

- Система не привязывается к конкретным файлам с данными и к их структуре. Все поисковые алгоритмы работают с единой базой данных, что позволяет легко использовать стандартные средства СУБД.
- Система позволяет вести БД по специфическим для каждого исходного файла характеристикам: соответствие полей, кодировки и т.п. Данная информация используется только на этапе преобразования данных.

- Обладает возможностями специфической обработки данных. Спектр этих возможностей достаточно широк, чтобы гарантированно обрабатывать до 95 % исходных файлов. Все вносимые в базу данных изменения доступны в реальном режиме времени.
- Использование в качестве СУБД SQL – сервера дает возможности воспользоваться стандартными средствами по обеспечению целостности и защиты данных от несанкционированного доступа.

Рис. 1. Структура ИПС

Выводы. При необходимости быстрого поиска в больших массивах данных с различной структурой необходима предварительная обработка данных и их хранение в одной базе данных. Качество ответа системы в значительной мере зависит от качества предварительной обработки данных. В рассматриваемой системе реализована возможность предварительной обработки данных по задаваемым параметрам. Это позволяет обрабатывать большинство входящей информации. Однако на практике встречаются случаи оригинально представленной информации, например код продукции и ее наименование находятся в одном поле. Для подобных случаев необходима специальная обработка данных, которую в автоматическом режиме можно осуществить только при наличии встроенного в систему интерпретатора.