

Воронов М.П., Часовских В.П. (УГЛТУ, г. Екатеринбург, РФ)

ИССЛЕДОВАНИЕ СТРУКТУРНЫХ ЭЛЕМЕНТОВ СУБД ADABAS ПРИ ПОСТРОЕНИИ КИС В ДЕРЕВООБРАБОТКЕ

DBMS ADABAS STRUCTURE ELEMENTS RESEARCH

Немаловажным фактором, определяющим эффективность корпоративной информационной системы (КИС) в деревообработке является выбор СУБД и среды разработки и функционирования программных элементов информационной системы. В данном разрезе исследование структурных элементов СУБД как средства повышения производительности обработки информации (и, как следствие, повышения конкурентоспособности предприятия) представляют широкий практический и научный интерес.

БД в ADABAS размещается на устройствах прямого доступа. Записи БД запоминаются в блоках устройств прямого доступа. Размер блока выбирается с учетом условия эффективного размещения целого числа блоков на дорожке используемых типов устройств прямого доступа [3].

Структурными элементами базы данных ADABAS на внутреннем уровне являются: накопитель, ассоциатор, рабочий набор и вспомогательные наборы данных. Накопитель, ассоциатор и рабочий набор размещаются в наборах данных ОС ЕС, которые имеют прямую организацию и могут быть многотомными [1].

Накопитель, который занимает не более пяти наборов данных, предназначен для хранения записей файла БД. Каждому файлу выделяется до пяти экстенгов, размещение которых отмечается в таблице размещения файла (ТРФ), находящейся в ассоциаторе.

Ассоциатор содержит сведения о структуре данных концептуального и внутреннего уровней БД в виде таблиц, списков и т. д., помещенных в блоки, и служит для взаимного отображения этих структур и выполнения операций над ними. Основная информация о размещении БД сосредоточена в таблице распределения памяти (ТРП). ТРП занимает 1 блок ассоциатора, имеет постоянный адрес и содержит:

- имя и номер БД;
- максимальное число файлов, которые могут быть загружены в БД;
- число загруженных файлов;
- номера системных файлов;
- адреса экстенгов, выделенных для ассоциатора, накопителя, рабочего набора;
- типы устройств, на которых они размещены, неиспользованные области памяти и т. д.

Инвертированные списки хранятся в блоке набора данных ассоциатора совместно с заголовочной частью, состоящей из значения атрибута, длины этого значения и длины инвертированного списка. В блоке размещается несколько списков, но все они должны соответствовать значениям одного атрибута. Список отсортирован в порядке возрастания значений ВСН, поэтому первый ВСН оказывается младшим элементом.

Доступ к требуемому инвертированному списку осуществляется с помощью многоуровневого индекса, содержащего индекс значений и старшие индексы (рис. 1.).

Нужный блок индекса значения, относящегося к данному атрибуту, отыскивается посредством старшего индекса. Блок старшего индекса состоит из записей. Каждая запись адресует один блок индекса значений. Запись включает код имени атрибута, поле признаков атрибута, длину значений атрибута, ВСН и адрес блока индекса значения. На все поисковые атрибуты заведены записи в старшем индексе.

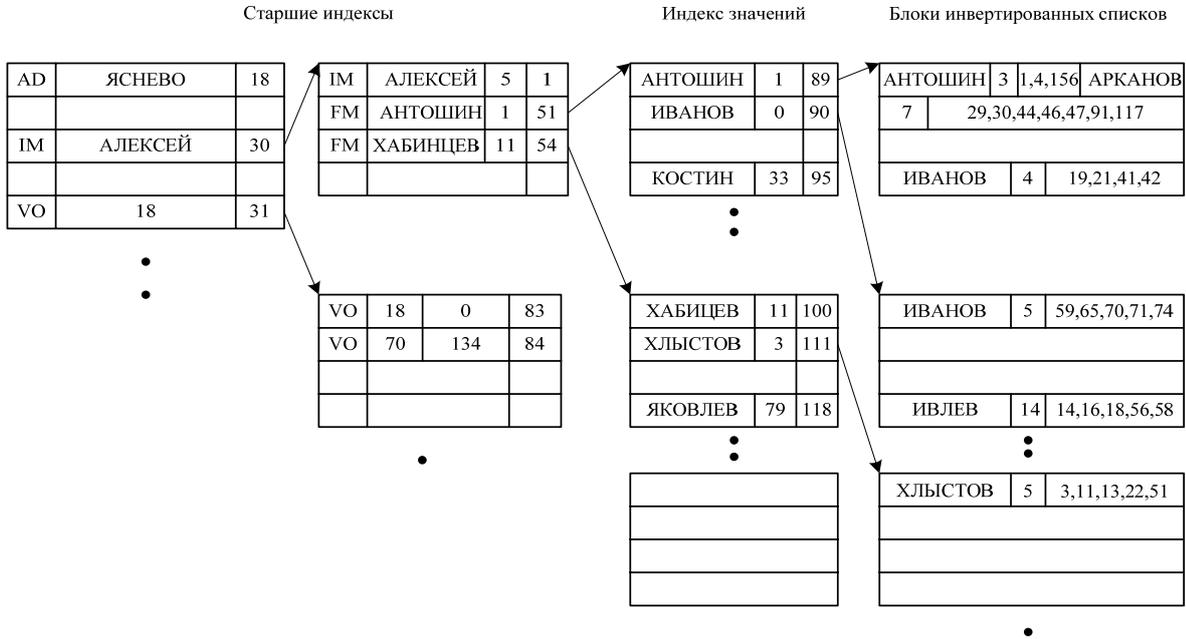


Рисунок 1 – Инвертированные структуры

Поисковые атрибуты, не имеющие значений в записях файла, представлены в записях старшего индекса нулевыми значениями полей длины, ВСН и адреса блока индекса значений. В поле признаков атрибута указываются характеристики атрибута: формат хранения, принадлежность к повторяющейся группе, признак множественного атрибута.

Рабочий набор данных занимает один экстенд памяти и состоит из блоков фиксированной длины. В составе рабочего набора выделяется область оперативного журнала изменений, используемого для поддержания логической целостности БД, области промежуточных списков ВСН, а также области результирующих списков ВСН. Схематически связь ВСН с блоками накопителя посредством преобразователя адреса показана на рисунке 2.

Для работы ряда утилит предназначены вспомогательные наборы данных (набор для сортировки и временный набор данных). Эти наборы имеют прямую организацию фиксированной длиной блока. Набор для сортировки используется при создании и модификации инвертированных списков. Временный набор данных применяется в качестве буферной памяти при загрузке БД для создания инвертированных списков [2].

