**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 9**

1. Индексирование таблиц. Связывание таблиц

Определение ключа для таблицы означает автоматическую сортировку записей, контроль отсутствия повторений значений в ключевых полях записей и повышение скорости выполнения операций поиска в таблице. Для реализации этих функций в СУБД применяется индексирование.

Под **индексом**понимают средство ускорения операции поиска записей в таблице, а, следовательно, и других операций, использующих поиск: извлечение, модификация, сортировка и т.д. Таблицу, для которой используется индекс, называют индексированной.

Индекс выполняет роль оглавления таблицы, просмотр которого предшествует обращению к записям таблицы. В некоторых системах индексы хранятся в индексных файлах, хранимых отдельно от табличных файлов.

Варианты решения проблемы организации физического доступа к информации зависят в основном от следующих факторов:

* вида содержимого в поле ключа записей индексного файла;
* типа используемых ссылок (указателей) на запись основной таблицы;
* метода поиска нужных записей.

В поле ключа индексного файла можно хранить значения ключевых полей индексируемой таблицы либо свертку ключа (так называемый хэш-код). Преимущество хранения хэш-кода вместо значения состоит в том, что длина свертки независимо от длины исходного значения ключевого поля всегда имеет некоторую постоянную и достаточно малую величину (например, 4 байта). Что существенно снижает время поисковых операций. Недостатком хэширования является необходимость выполнения свертки (требует определенного времени), а также борьба с возникновением коллизий (свертка различных значений может дать одинаковый хэш-код).

Для организации ссылки на запись таблицы могут использоваться три типа адресов: абсолютный (действительный), относительный и символический (идентификатор).

На практике чаще всего используются два метода поиска данных: последовательный и бинарный (основан на делении интервала поиска пополам).

Пример одноуровневой схемы индексации.

Пусть ОС поддерживает прямую организацию данных на магнитных дисках. Основные таблицы и индексные файлы хранятся в отдельных файлах. Информация файлов хранится в виде совокупности блоков фиксированного размера, например, целого числа кластеров.

При **одноуровневой схеме**в индексном файле хранятся короткие записи, имеющие два поля: поле содержимого старшего ключа (хэш-кода ключа) адресуемого блока и поле адреса начала этого блока. В каждом блоке записи располагаются в порядке возрастания значения ключа или свертки.

Старшим ключом каждого блока является ключ его последней записи.

Если в индексном файле хранятся хэш-коды ключевых полей индексированной таблицы, то алгоритм поиска нужной записи (с указанным ключом) в таблице включает в себя следующие три этапа.

1. Образование свертки значения ключевого поля искомой записи.
2. Поиск в индексном файле записи о блоке, значение первого поля которого больше полученной свертки (это гарантирует нахождение искомой свертки в этом блоке).
3. Последовательный просмотр записей блока до совпадения сверток искомой записи и записи блока файла. В случае коллизий сверток ищется запись, значение ключа которой совпадает со значением ключа искомой записи.

Основным недостатком одноуровневой схемы является то, что ключи (свертки) записей хранятся вместе с записями. Это приводит к увеличению времени поиска записей из-за большой длины просмотра (значения данных в записях приходится пропускать).

**Двухуровневая схема**в ряде случаев оказывается более рациональной. В ней ключи (свертки) записей отделены от содержимого записей (рис.1). В этой схеме индекс основной таблицы распределен по совокупности файлов: одному файлу главного индекса и множеству файлов с блоками ключей.

На практике для создания индекса для некоторой таблицы БД пользователь указывает поле таблицы, которое требует индексации. Ключевые поля таблицы во многих СУБД как правило индексируются автоматически. Индексные файлы, создаваемые по ключевым полям таблицы, часто называют файлами первичных индексов.

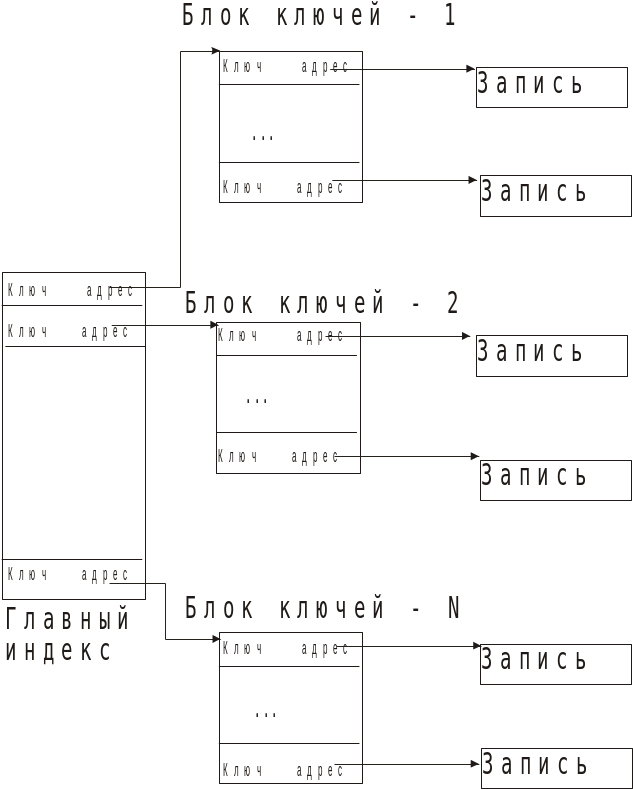


Рис. 1. Двухуровневая схема индексации

Индексы, создаваемые пользователем для неключевых полей, иногда называют **вторичными (пользовательскими) индексами.**Введение таких индексов не изменяет физического расположения записей таблицы, но влияет на последовательность просмотра записей. Индексные файлы, создаваемые для поддержания вторичных индексов таблицы, обычно называются файлами вторичных индексов.

Связь вторичного индекса с элементами данных базы может быть установлена различными способами. Один из них – использование вторичного индекса как входа для получения первичного ключа, по которому затем с использованием первичного индекса производится поиск необходимых записей (рис. 2).

Некоторыми СУБД, например Access, деление индексов на первичные и вторичные не производится. В этом случае используются автоматически создаваемые индексы и индексы, определяемые пользователем по любому из не ключевых полей.

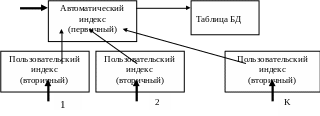


Рис. 2. Способ использования вторичных индексов

Главная причина повышения скорости выполнения различных операций в индексированных таблицах состоит в том, что основная часть работы производится с небольшими индексными файлами, а не с самими таблицами. Наибольший эффект повышения производительности работы с индексированными таблицами достигается для значительных по объему таблиц. Индексирование требует небольшого дополнительного места на диске и незначительных затрат процессора на изменение индексов в процессе работы. Индексы в общем случае могут изменяться перед выполнением запросов к БД, после выполнения запросов к БД, по специальным командам пользователя или программным вызовам приложений.

**Связывание таблиц**

При проектировании БД информацию обычно размещают в нескольких таблицах, связанных семантикой информации. В реляционных СУБД для указания связей таблиц производят операцию их связывания.

Многие СУБД при связывании таблиц автоматически выполняют контроль целостности вводимых в базу данных в соответствии с установленными связями. Это повышает достоверность хранимой в БД информации.

Установление связи между таблицами облегчает доступ к данным. Связывание таблиц при выполнении таких операций, как поиск, просмотр, редактирование, выборка и подготовка отчетов, обычно обеспечивает возможность обращения к произвольным полям связанных записей. Это уменьшает количество явных обращений к таблицам данных и число манипуляций в каждой из них.

**Основные виды связи таблиц**

Между таблицами могут устанавливаться бинарные (между двумя таблицами), тернарные (между тремя таблицами) и, в общем случае, n-арные связи.

При связывании двух таблиц выделяют основную и дополнительную (подчиненную) таблицы. Логическое связывание таблиц производится с помощью **ключа связи.**

Ключ связи, по аналогии с обычным ключом таблицы, состоит из одного или нескольких полей, которые в данном случае называют **полями связи**(ПС).

Суть связывания состоит в установлении соответствия полей связи основной и дополнительной таблиц. Поля связи основной таблицы могут быть обычными и ключевыми. В качестве полей связи подчиненной таблицы чаще всего используют ключевые поля.

В зависимости от того, как определены поля связи основной и дополнительной таблиц (как соотносятся ключевые поля с полями связи), между двумя таблицами в общем случае могут устанавливаться следующие четыре основных вида связи (табл. 4.2):

* Один – один (1:1);
* Один – много (1:М);
* Много – один (М:1);
* Много – много (М:М).

## Связь 1:1

Связь вида 1:1 образуется в случае, когда все поля связи основной и дополнительной таблиц являются ключевыми. Поскольку значения в ключевых полях обеих таблиц не повторяются, обеспечивается взаимно-однозначное соответствие записей из этих таблиц. Сами таблицы по сути становятся равноправными.

# Связь вида 1:м

Связь вида 1:М имеет место в случае, когда одной записи основной таблицы соответствует несколько записей вспомогательной таблицы.

## Связь вида м:1

Связь вида М:1 имеет место в том случае, когда одной или нескольким записям основной таблицы ставится в соответствие одна запись дополнительной таблицы.

# Связь вида м:м

Самый общий вид связи М:М возникает в случаях, когда нескольким записям основной таблицы соответствует несколько записей дополнительной таблицы.

1. Потенциальные ключи

Потенциальный ключ – подмножество атрибутов переменной отношения, удовлетворяющее требованиям уникальности и минимальности (несократимости).

* Уникальность означает, что в отношении не может быть двух кортежей, в которых значения этого подмножества атрибутов совпадают (равны).
* Минимальность (несократимость) означает, что в составе потенциального ключа отсутствует меньшее подмножество атрибутов, удовлетворяющее условию уникальности. Иными словами, если из потенциального ключа убрать любой атрибут, он утратит свойство уникальности.

Виды потенциальных ключей:

По составу атрибутов

* простой (Name);
* составной (First Name, Last Name);

По «осмысленности» значений

* естественный, натуральный, интеллектуальный (Name)
* искусственный, суррогатный (ID).

Первичный ключ – один из потенциальных ключей отношения, выбранный в качестве основного ключа (или ключа по умолчанию). Если в отношении имеется единственный потенциальный ключ, он является и первичным ключом. Если потенциальных ключей несколько, один из них выбирается в качестве первичного, а другие называют «альтернативными». С точки зрения РМД все потенциальные ключи отношения эквивалентны.

**Альтернативными ключами**(alternate key) называют потенциальный ключ, не выбранный первичным ключом.