

Физическое проектирование БД

Файлы и файловые структуры.

Лекция 13

- Физические модели баз данных определяют способы размещения данных в среде хранения и способы доступа к этим данным, которые поддерживаются на физическом уровне.
- Исторически первыми системами хранения и доступа были файловые структуры и системы управления файлами (СУФ), которые фактически являлись частью операционных систем.
- СУБД создавала над этими файловыми моделями свою надстройку, которая позволяла организовать всю совокупность файлов таким образом, чтобы она работала как единое целое и получала централизованное управление от СУБД.
- Механизмы буферизации и управления файловыми структурами не приспособлены для решения задач собственно СУБД, эти механизмы разрабатывались для традиционной обработки файлов, и с ростом объемов хранимых данных они стали неэффективными для использования СУБД.
- Тогда постепенно произошел переход от базовых файловых структур к непосредственному управлению размещением данных на внешних носителях самой СУБД.
- Пространство внешней памяти уже выходило из-под владения СУФ и управлялось непосредственно СУБД.

- В каждой СУБД по-разному организованы хранение и доступ к данным.
- Существуют файловые структуры, которые имеют общепринятые способы организации и широко применяются практически во всех СУБД.
- В системах БД файлы и файловые структуры, которые используются для хранения информации во внешней памяти, можно классифицировать следующим образом:



- С точки зрения пользователя, *файлом* называется поименованная линейная последовательность записей, расположенных на внешних носителях.

- В файле всегда можно определить текущую запись, предшествующую ей и следующую за ней.

- Всегда существует понятие первой и последней записи файла.



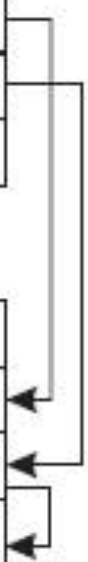
Основная область

Содержание записей	Ссылка на синонимы
Петров	1
Степанов	2

...

Область переполнения

Содержание записей	Ссылка на синонимы
Петрова	
Степановский	3
Степанчиков	



В соответствии с методами управления доступом различают:

- устройства внешней памяти с произвольной адресацией (магнитные и оптические диски)
- Устройства внешней памяти с последовательной адресацией (магнитофоны, стримеры).
- На устройствах с **произвольной адресацией** теоретически возможна установка головок чтения-записи в произвольное место мгновенно. Практически существует время позиционирования головки, которое весьма мало по сравнению со временем считывания-записи.
- В устройствах с **последовательным доступом** для получения доступа к некоторому элементу требуется "перемотать (пройти)" все предшествующие ему элементы информации. На устройствах с последовательным доступом вся память рассматривается как линейная последовательность информационных элементов.

Элемент 1	Элемент 2	Элемент N	Элемент N+1
-----------	-----------	--------	--------	-----------	-------------

- Файлы с постоянной длиной записи, расположенные на устройствах прямого доступа (УПД), являются *файлами прямого доступа*.
- В этих файлах физический адрес расположения нужной записи может быть вычислен по номеру записи (NZ).

Каждая файловая система СУФ — система управления файлами поддерживает некоторую иерархическую файловую структуру, включающую чаще всего неограниченное количество уровней иерархии в представлении внешней памяти



Для каждого файла в системе хранится следующая информация:

- имя файла;
- тип файла (например, расширение или другие характеристики);
- размер записи;
- количество занятых физических блоков;
- базовый начальный адрес;
- ссылка на сегмент расширения;
- способ доступа (код защиты).

- Для файлов с постоянной длиной записи адрес размещения записи с номером K может быть вычислен по формуле:

$$BA + (K - 1) * LZ + 1,$$

где BA — базовый адрес,

LZ — длина записи.

- Если можно всегда определить адрес, на который -необходимо позиционировать механизм считывания-записи, то устройства прямого доступа делают это практически мгновенно, поэтому для таких файлов чтение произвольной записи практически не зависит от ее номера.
- Файлы прямого доступа обеспечивают наиболее быстрый доступ к произвольным записям, и их использование считается наиболее перспективным в системах баз данных.

- На устройствах последовательного доступа могут быть организованы файлы только последовательного доступа.
- Файлы с переменной длиной записи всегда являются файлами последовательного доступа. Они могут быть организованы двумя способами:

1. Конец записи отличается специальным маркером.

Запись 1	X	Запись 2	X	Запись 3	X
----------	---	----------	---	----------	---

2. В начале каждой записи записывается ее длина.

LZ1	Запись 1	LZ2	Запись 2	LZ3	Запись 3
-----	----------	-----	----------	-----	----------

Здесь LZN — длина N-й записи.

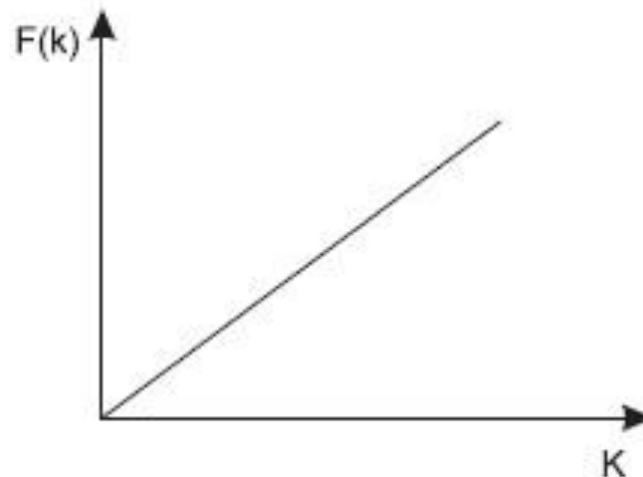
- Файлы с прямым доступом обеспечивают наиболее быстрый способ доступа.
- Доступ по номеру записи в базах данных весьма неэффективен.
- Чаще всего в базах данных необходим поиск по первичному или возможному ключам, иногда необходима выборка по внешним ключам, но во всех этих случаях мы знаем значение ключа, но не знаем номера записи, который соответствует этому ключу.

- При организации файлов прямого доступа в некоторых очень редких случаях возможно построение функции, которая по значению ключа однозначно вычисляет адрес (номер записи файла).

$$NZ = F(K),$$

где NZ — номер записи, K — значение ключа, $F()$ — функция.

- Функция $F()$ при этом должна быть линейной, чтобы обеспечивать однозначное соответствие.



- Не всегда удастся построить взаимно-однозначное соответствие между значениями ключа и номерами записей.
- Часто бывает, что значения ключей разбросаны по нескольким диапазонам.



- В этом случае не удастся построить взаимно-однозначную функцию, либо эта функция будет иметь множество незадействованных значений, которые соответствуют недопустимым значениям ключа.
- В подобных случаях применяют различные методы хэширования (рандомизации) и создают специальные хэш-функции.

- **Суть методов хэширования** состоит в том, что мы берем значения ключа (или некоторые его характеристики) и используем его для начала поиска,
- Мы вычисляем некоторую хэш-функцию $h(k)$ и полученное значение берем в качестве адреса начала поиска.
- Мы не требуем полного взаимно-однозначного соответствия, но, с другой стороны, для повышения скорости мы ограничиваем время этого поиска (количество дополнительных шагов) для окончательного получения адреса.
- Мы допускаем, что нескольким разным ключам может соответствовать одно значение хэш-функции (то есть один адрес).
- Подобные ситуации называются *коллизиями* .
- Значения ключей, которые имеют одно и то же значение хэш-функции, называются *синонимами*.
- Поэтому при использовании хэширования как метода доступа необходимо принять два независимых решения:
 - ✓ выбрать хэш-функцию;
 - ✓ выбрать метод разрешения коллизий.

Стратегия разрешения коллизий с областью переполнения

Область хранения разбивается на 2 части.

Для каждой новой записи вычисляется значение хэш-функции, которое определяет адрес ее расположения, и запись заносится в основную область в соответствии с полученным значением хэш-функции.

• Основная область:

Содержание записей	Ссылка на синонимы
Петров	1
Сахаров	3

• Область переполнения:

Содержание записей	Ссылка на синонимы
Петрова	2
Петровский	
Сахарочкин	

- Если вновь заносимая запись имеет значение функции хэширования такое же, которое использовала другая запись, уже имеющаяся в БД, то новая запись заносится в область переполнения на первое свободное место,
- а в записи-синониме, которая находится в основной области, делается ссылка на адрес вновь размещенной записи в области переполнения.
- Если же уже существует ссылка в записи-синониме, которая расположена в основной области, то тогда новая запись получает дополнительную информацию в виде ссылки и уже в таком виде заносится в область переполнения.

- При поиске записи сначала вычисляется значение ее хэш-функции и считывается первая запись в цепочке синонимов, которая расположена в основной области.
- Если искомая запись не соответствует первой в цепочке синонимов, то далее поиск происходит перемещением по цепочке синонимов, пока не будет обнаружена требуемая запись.
- При удалении произвольной записи сначала определяется ее место расположения. Если удаляемой является первая запись в цепочке синонимов, то после удаления на ее место в основной области заносится вторая (следующая) запись в цепочке синонимов, при этом все указатели (ссылки на синонимы) сохраняются.
- Если же удаляемая запись находится в середине цепочки синонимов, то необходимо провести корректировку указателей: в записи, предшествующей удаляемой, в цепочке ставится указатель из удаляемой записи.
- Если это последняя запись в цепочке, то все равно механизм изменения указателей такой же, то есть в предшествующую запись заносится признак отсутствия следующей записи в цепочке, который ранее хранился в последней записи.

Организация стратегии свободного замещения

- При этой стратегии файловое пространство не разделяется на области, но для каждой записи добавляется 2 указателя:
 - ✓ указатель на предыдущую запись в цепочке синонимов
 - ✓ указатель на следующую запись в цепочке синонимов.
- Отсутствие соответствующей ссылки обозначается специальным символом, например нулем.
- Для каждой новой записи вычисляется значение хэш-функции, и если данный адрес свободен, то запись попадает на заданное место и становится первой в цепочке синонимов.
- Если адрес, соответствующий полученному значению хэш-функции, занят, то по наличию ссылок определяется, является ли запись, расположенная по указанному адресу, первой в цепочке синонимов.

- Если да, то новая запись располагается на первом свободном месте и для нее устанавливаются соответствующие ссылки: она становится второй в цепочке синонимов, на нее ссылается первая запись, а она ссылается на следующую, если таковая есть.
- Если запись, которая занимает требуемое место, не является первой записью в цепочке синонимов, значит, она занимает данное место "незаконно" и при появлении "законного владельца" должна быть "выселена", то есть перемещена на новое место.
- Механизм перемещения аналогичен занесению новой записи, которая уже имеет синоним, занесенный в файл.
- Для этой записи ищется первое свободное место и корректируются соответствующие ссылки: в записи, которая является предыдущей в цепочке синонимов для перемещаемой записи, заносится указатель на новое место перемещаемой записи, указатели же в самой перемещаемой записи остаются прежние.

- После перемещения "незаконной" записи вновь вносимая запись занимает свое законное место и становится первой записью в новой цепочке синонимов.
- Механизмы удаления записей во многом аналогичны механизмам удаления в стратегии с областью переполнения. Однако еще раз кратко опишем их.
- Если удаляемая запись является первой записью в цепочке синонимов, то после удаления на ее место перемещается следующая (вторая) запись из цепочки синонимов и проводится соответствующая корректировка указателя третьей записи в цепочке синонимов, если таковая существует.
- Если же удаляется запись, которая находится в середине цепочки синонимов, то производится только корректировка указателей: в предшествующей записи указатель на удаляемую запись заменяется указателем на следующую за удаляемой запись, а в записи, следующей за удаляемой, указатель на предыдущую запись заменяется на указатель на запись, предшествующую удаляемой.