МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра информационных систем

Отчет   
по лабораторной работе №2  
на тему: «Организация хешированного файла»  
по дисциплине «Базы данных»

Выполнили: Марочкин М.А. Шифр: 170584   
 Шорин В.Д. Шифр: 171406  
 Щекотихин С.Е. Шифр: 170590  
ИПАИТ  
Направление: 09.03.04 «Программная инженерия»  
Группа: 71-ПГ  
Проверил:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Отметка о зачете:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата «\_\_\_\_» \_\_\_\_\_\_ 2019 г.

Орел, 2019 г.

**Цель работы:**

Написать программу, которая работает с хешированным фалом хранящем информацию об отношении «студент».

В программе должны быть реализованы следующие функции:

* добавление информации о студент;
* изменение информации о студенте;
* удаление информации о студенте;
* осуществление поиска информации о студенте.

Отношение студент должно содержать следующие атрибуты: номер зачетки (тип integer), фамилия (тип string(30)), имя (тип string(20)), отчество (тип string(30)), номер группы (тип integer). Атрибут «номер зачетки» выступает в роли первичного ключа.

В качестве хеш-функции необходимо использовать остаток от деления первичного ключа на 4.

**Контрольные вопросы**

1. **Что такое бакет?**

*Бакет* – это участок файла, в котором неупорядоченно хранятся записи, имеющие одинаковое значение хэш-функции для ключа **x**.

1. **Что такое каталог бакетов?**

*Каталог бакетов* – это таблица, включающая в себя **B** указателей (с нумерацией от 0 до **(B-1)**) на первый блок того или иного сегмента. В некоторых случаях указатель на первый блок бакета дополняется ссылкой на последний блок сегмента.

1. **Что такое хеш-функция?**

*Хеш-функция* – это функция, преобразовывающая значения одного или нескольких полей одной записи в целое значение фиксированной длины. Обычно данная функция обозначается буквой **h**, а результат её принадлежит отрезку от 0 до **(B-1)**.

1. **В чем особенности организации хешированных файлов?**

Основная идея хешированных файлов состоит в том, что файл разбивается на бакеты (участки). Каждая запись файла размещается в соответствующем **бакете** в зависимости от значения ключа записи. Ключ в данном случае не обязан быть уникальным. Для каждого файла, организованного таким способом, выбирается hash-функция h(v). Эта функция в качестве параметра получает ключ записи. Выходом такой функции является целое число в диапазоне от 0 до B-1, где B – желаемое число бакетов. Внутри бакета записи организованы также как в файле-куче, т.е. нет никакой упорядоченности.

1. **Причины снижения эффективности хешированных файлов.**

Причиной снижения эффективности хешированных файлов является не верный выбор хеш-функции. Обычно хеш-функция выбирается исходя из знаний о количестве и распределении данных на момент создания файлов, поэтому в процессе эксплуатации она может становиться менее эффективной. Решением данной проблемы является замена статического хеширования динамическим хешированием. Эффективность работы хешированных файлов снижается по мере увеличения числа блоков в бакетах. Наиболее эффективным был бы вариант, когда в каждом бакете находится только один блок, т.е. эф-ть поиска равна единице (+ поиск по каталогу бакетов).

Кроме роста объема данных на снижение эффективности влияет изменение статистики распределения записи по бакетам, т.е. бакеты становятся разной длины. В любом случае причина снижения эф-ти заключается в неэф-ной hash-ф-ции.

1. **Что такое динамическое хеширование?**

Динамическое хеширование позволяет изменить хеш-функцию по мере роста или уменьшения файла. В этом случае изменение размера файла сопровождается соединением или делением бакетов.

1. **Эффективность работы хешированных файлов.**

Основное положение эффективности хешированных файлов следующее. Все операции на хешированных файлах производятся в ***В*** раз быстрее (***В*** – число бакетов), чем на файлах, организованных в виде кучи.

Пусть в файле храниться n записей, помещающихся в R блоков. Хеш-функция выбрана так, что число бакетов равняется B. Если при этом таблица бакетов находится в оперативной памяти, то потребуется следующее число доступов:

1)  ***n/2BR*** при успешном поиске, а также удалении и модификации существующей записи;

2)  ***n/BR*** при неудачном поиске, а также при проверке бакета перед вставкой, удалением и модификацией, если искомая запись отсутствует.

Основанием для таких расчетов служит тот факт, что в бакетах находится по n/R записей. Функция хеширования выбрана так, что записи в бакеты размещаются равномерно.

Если каталог бакетов находится на диске, то все количество доступов по всем операциям увеличивается на единицу, а в том случае, когда требуется внести изменения в каталог бакетов, на два.

Хорошо продуманная организация файлов с хешированным доступом требует лишь незначительного числа обращений к блокам при выполнении каждой операции с файлами. Если функция хеширования выбрана правильно и количество сегментов приблизительно равно количеству записей в файле, деленному на количество записей, которые могут уместиться в одном блоке, тогда средний сегмент состоит из одного блока. Если не учитывать обращения к блокам, которые требуются для просмотра таблицы бакетов, то типичная операция поиска данных по ключу потребует лишь одного обращения к блоку, а операции вставки, удаления или изменения потребуют двух обращений к блокам. Если среднее количество записей в бакете намного превосходит количество записей, которые могут уместиться в одном блоке, можно периодически реорганизовывать таблицу сегментов, удваивая количество бакетов и деля каждый из них на две части.

**Код**