



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ

«Информатика и системы управления»

КАФЕДРА

«Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

«Классификация методов построения
индексов в базах данных»

Студент:

ИУ7-73Б

(группа)

(подпись, дата)

М. Д. Маслова

(И. О. Фамилия)

Преподаватель:

(подпись, дата)

А. А. Оленев

(И. О. Фамилия)

2022 г.

РЕФЕРАТ

Расчетно-пояснительная записка 13 с., 3 рис., 0 табл., 4 источн., 1 прил.

Ключевые слова:

Краткое описание

СОДЕРЖАНИЕ

РЕФЕРАТ	3
ВВЕДЕНИЕ	5
1 Анализ предметной области	6
1.1 Основные определения	6
1.2 Типы индексов	7
2 Описание существующих решений	10
2.1 Индексы на основе деревьев поиска	10
2.1.1 В-деревья	10
2.1.2 B^+ -деревья	10
2.1.3 «ИДругие»-деревья	10
2.2 Индексы на основе хеш-таблиц	10
2.3 Индексы на основе битовых карт	10
3 Классификация существующих решений	11
ЗАКЛЮЧЕНИЕ	12
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	13

ВВЕДЕНИЕ

АКТУАЛЬНОСТЬ РАБОТЫ

Целью данной работы является ***классификация методов построения индексов в базах данных.***

Для достижения поставленной цели требуется решить следующие задачи:

- описываются методы построения индексов в базах данных;
- предлагаются и обосновываются критерии оценки качества описанных методов;
- сравниваются методы по предложенным критериям оценки;
- выделяются методы, показывающие лучшие результаты по одному или нескольким критериям.

1 Анализ предметной области

1.1 Основные определения

Индекс — это некоторая структура, обеспечивающая быстрый поиск записей в базе данных [1]. Индекс определяет соответствие значения атрибута или набора атрибутов — *ключа поиска* — конкретной записи с местоположением этой записи [2]. Это соответствие организуется с помощью индексных записей. Каждая из них соответствует записи в *индексируемой таблице* — таблице, по которой строится индекс — и содержит два поля: идентификатор записи или указатель на нее, а также значение индексированного поля в этой записи [3].

Индексы могут использоваться для поиска по конкретному значению или диапазону значений, а также для проверки существования элемента в таблице, однако обеспечение уменьшения времени доступа к записям в общем случае достигается за счет [2]:

- упорядочивания индексных записей по ключу поиска, что уменьшает количество записей, которые необходимо просмотреть;
- а также меньшего размера индекса по сравнению с индексируемой таблицей, сокращающего время чтения одного элемента.

В то же время индекс является структурой, которая строится в дополнение к существующим данным, то есть он занимает дополнительный объем памяти и должен соответствовать текущим данным. Последнее значит, что индекс необходимо изменять при вставке или удалении элементов, на что затрачивается время, поэтому индекс, ускоряя работу СУБД при доступе к данным, замедляет операции изменения таблицы, что необходимо учитывать [4].

Таким образом, индекс может описываться: [2]:

- *типом доступа* — поиск записей по атрибуту с конкретным значением, или со значением из указанного диапазона;
- *временем доступа* — время поиска записи или записей;
- *временем вставки*, включающее время поиска правильного места вставки, а также время для обновления индекса;
- *временем удаления*, аналогично вставке, включающее время на поиск удаляемого элемента и время для обновления индекса;
- *дополнительной памятью*, занимаемая индексной структурой.

1.2 Типы индексов

Индексы могут быть:

- первичные и вторичные (???);
- кластеризованные и некластеризованные;
- плотные и разреженные;
- одноуровневые и многоуровневые;
- а также иметь в своей основе различные структуры, что описывается в следующем разделе, так как исследуется в данной работе.

первичные — по первичному ключу/по уникальным значениям/кластеризованные,

вторичные — по всем остальным атрибутам/по не уникальным значениям/ некластеризованные;

По порядку записей в индексируемой таблице индексы делятся на кластеризованные и некластеризованные. В *кластеризованных* индексах логический порядок ключей определяет физическое расположение записей, а так как строки в таблице могут быть упорядочены только в одном порядке, то кластеризованный индекс может быть только один на таблицу. Логический порядок *некластеризованных* индексов не влияет на физический, и индекс содержит указатели на записи таблицы.

Также индексы делятся на плотные и разреженные. *Плотные* индексы (рисунок 1.1) содержат ключ поиска и указатель на первую запись с заданным ключом поиска. При этом в кластеризованных индексах другие записи с заданным ключом будут лежать сразу после первой записи, так как записи в таких файлах отсортированы по тому же ключу. Плотные некластеризованные индексы должны содержать список указателей на каждую запись с заданным ключом поиска.

В *разреженных* индексах (рисунок 1.2) записи содержат только некоторые значения ключа поиска, а для доступа к элементу отношения ищется запись индекса с наибольшим меньшим или равным значением ключа поиска, происходит переход по указателю на первую запись по найденному ключу и далее по указателям в файле происходит поиск заданной записи. Таким образом, разреженные индексы могут быть построены только на отсортированных последовательностях записей, иначе хранения только некоторых ключей поиска

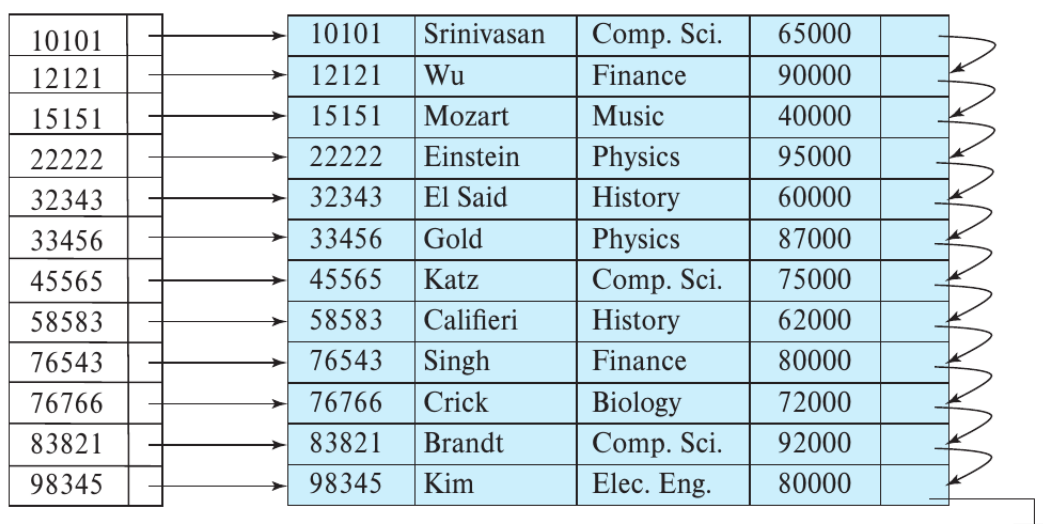


Рисунок 1.1 – Плотный индекс

будет недостаточно, так как будет неизвестно, после записи, с каким ключом будет лежать необходимый элемент отношения.

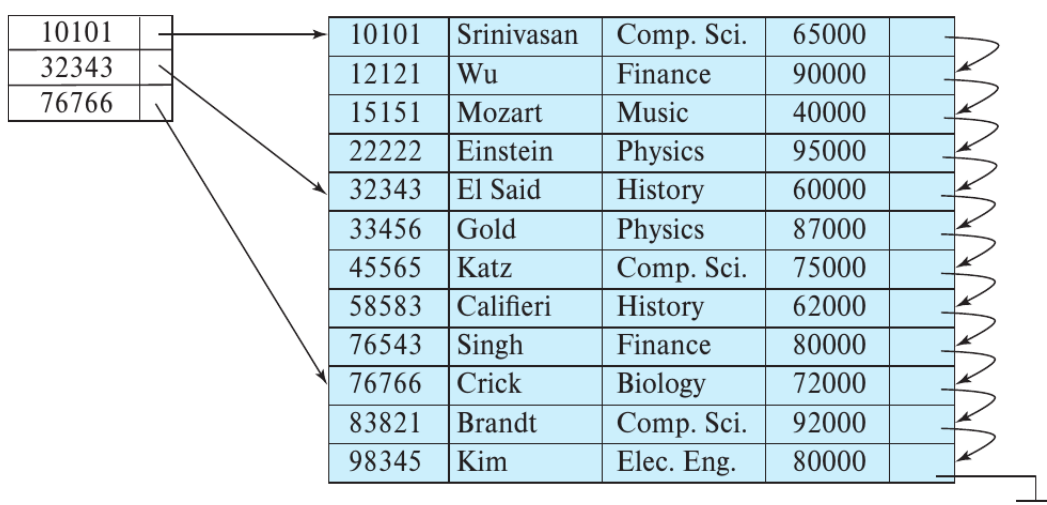


Рисунок 1.2 – Разреженный индекс

Поиск с помощью неразреженных индексов быстрее, так как указатель в записи индекса сразу приводит к необходимым записям. Однако разреженные индексы требуют меньше дополнительной памяти и сокращают время поддержания структуры индекса в актуальном состоянии при вставке или удалении.

Одноуровневые индексы ссылаются на данные таблице, индексы же верхнего уровня *многоуровневой* структуры ссылают на индексы нижестоящего

уровня (рисунок 1.3).

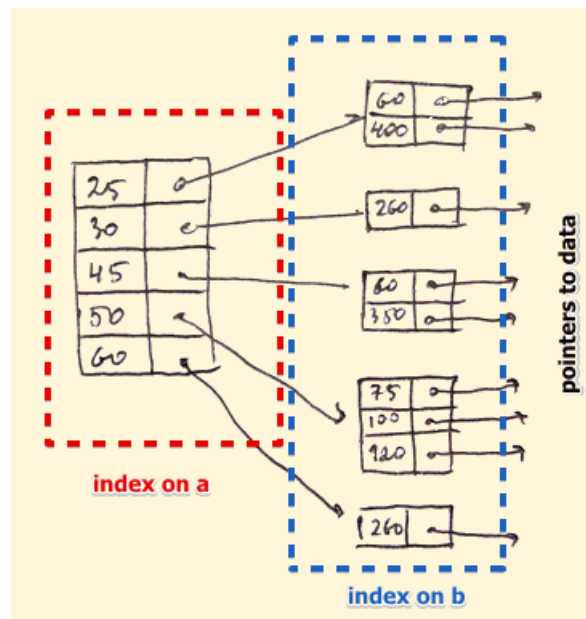


Рисунок 1.3 – Многоуровневый индекс

По структуре индексы подразделяются на

- упорядоченные, на основе деревьев поиска,
- хеш-индексы,
- индексы, на основе битовых карт.

Построение структур каждого из приведенных типов индекса рассматривается в отдельном разделе, так как именно оно исследуется в данной работе.

2 Описание существующих решений

2.1 Индексы на основе деревьев поиска

2.1.1 В-деревья

2.1.2 B^+ -деревья

2.1.3 «ИДругие»-деревья

2.2 Индексы на основе хеш-таблиц

2.3 Индексы на основе битовых карт

3 Классификация существующих решений

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Григорьев Ю. А., Плутенко А. Д., Плужникова О. Ю.* Реляционные базы данных и системы NoSQL: учебное пособие. — Благовещенск : Амурский гос. ун-т, 2018. — С. 424.
2. *Silberschatz A., Korth H. F., Sudarshan S.* Database System Concepts. — New York : McGraw-Hill, 2020. — С. 1344.
3. *Эдвард Сьоре.* Проектирование и реализация систем управления базами данных. — М. : ДМК Пресс, 2021. — С. 466.
4. *Осинов Д. Л.* Технологии проектирования баз данных. — М. : ДМК Пресс, 2019. — С. 498.