Администрирование MySQL и оптимизация запросов

6 модуль

Сложность выполнения запросов. EXPLAIN

План выполнения запроса

Описание того, какие отношения и как будут использоваться для выполнения запроса.

Показывает:

- Количество задействованных кортежей
- Использование ключей: фактически и возможные

Стратегия составления плана

- С какой таблицы начать: выбирает ту, где меньше кортежей.
- Какие условия лучше обработать первыми, чтобы отфильтровать больше.
- Какие ключи использовать, чтобы как можно раньше как можно больше отфильтровать.

Индексы

Индексы

- У отношения может быть один первичный ключ, доступ по которому происходит быстро.
- Для остальных полей проверка на условия требует просмотра всех кортежей.
- Индексы выстраивают структуру, позволяющую не просматривать всю таблицу, чтобы найти нужные кортежи.

Как хранятся данные

- Быстрее всего искать данные в упорядоченном списке
- Чтобы изменять данные в упорядоченном списке можно использовать дерево

Индексы

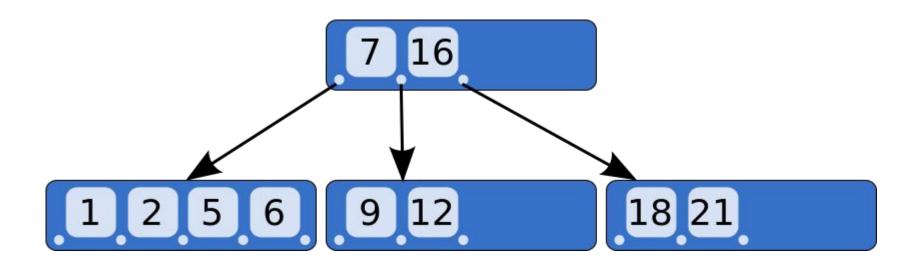
Кластерный индекс:

• Упорядоченное дерево, на листьях которого находятся сами данные

Некластерные индексы:

• Структуры, выстраиваемые вокруг данных со ссылками на кортежи

В-дерево

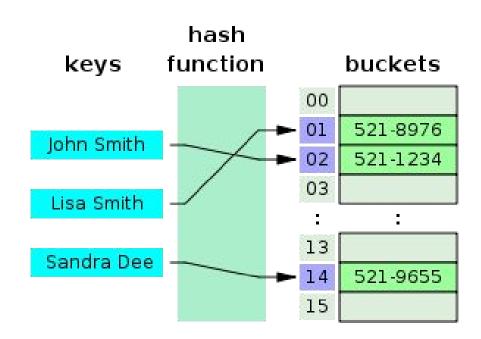


В-дерево

Подходит для:

- Операций сравнения: <, <=, =, >=, >
- Проверки is (not) null
- Сортировки
- Группировки

Хэш-индекс



Хэш-индекс

Подходит для:

- Сравнений = и !=
- WHERE field_name IN (...)

Используется во внешних ключах.

Другие типы индексов

- Геолокация
- Перекрывающие индексы
- Многомерные структуры
- Битовые индексы

Индексы и план запроса

- Индекс создает ключ, который СУБД может использовать в плане запроса.
- СУБД выбирает наиболее подходящий ключ среди доступных, но пользователь может указать, какой ключ использовать лучше.

Стратегия создания индексов

- Проверить, где тип операции ALL и количество выбираемых кортежей, если их
- Проверить условия: если сравнения, то можно также добавить индекс BTREE.
- Если есть операции сравнения на равенство/неравенство, то можно добавить индекс HASH.