

БЛОК 8. ИНДЕКСЫ

# B-TREE

begin



# ЦЕЛЬ



01

Понять, зачем нужны  
btree индексы

02

Понять архитектура  
btree индекса

03

Узнать их плюсы  
и минусы

04

Научиться создавать  
btree индекс

# СОДЕРЖАНИЕ УРОКА



1

В-tree индекс

2

Плюсы и минусы

3

Практика





# СВОЙСТВА BTREE ИНДЕКСА



Используется в 99% индексов.

Применим для любого типа, который можно отсортировать в чётко определённом линейном порядке.

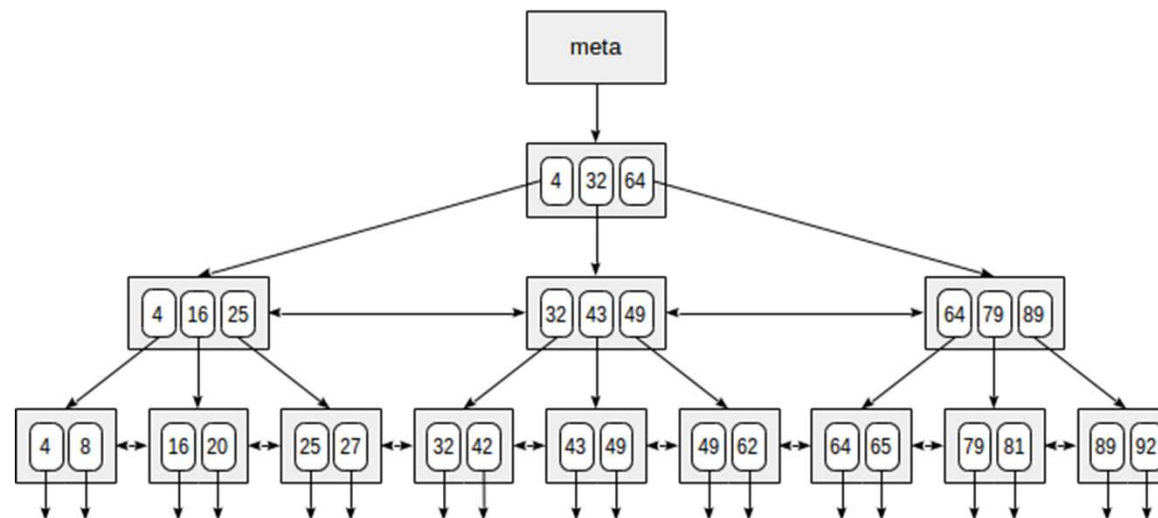
---

## Поддерживает:

- операторы сравнения `>`, `<`, `=`, `>=`, `<=`, `BETWEEN` и `IN`
- условия пустоты `IS NULL` и `IS NOT NULL`
- операторы поиска подстроки `LIKE` и `~`, если искомая строка закреплена в начале шаблона (например `str LIKE 'search%'`)
- регистронезависимые операторы поиска подстроки `ILIKE` и `~*` (но только в том случае, если искомая строка начинается с символа, который одинаков и в верхнем, и в нижнем регистре, например, числа)

<https://habr.com/ru/company/postgrespro/blog/330544/>

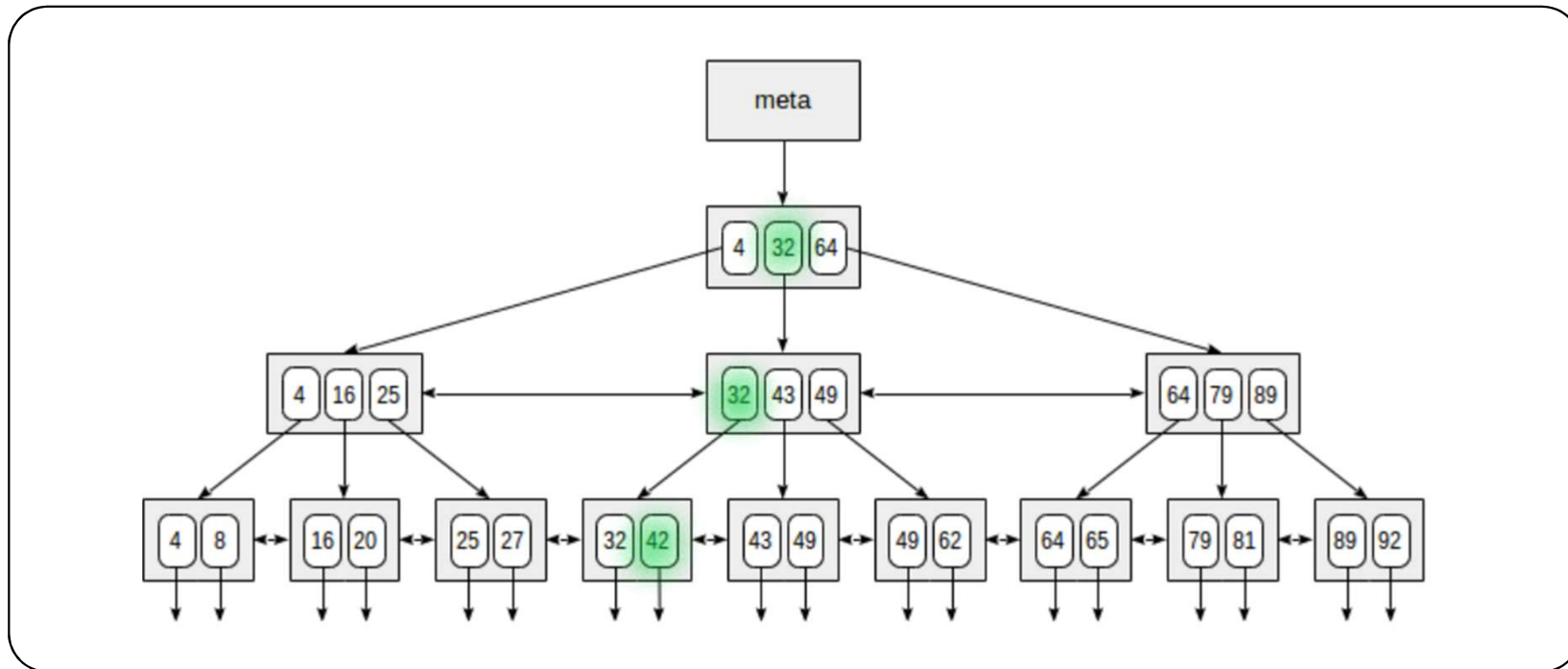
# BTREE - BALANCED TREE, NOT BINARY!



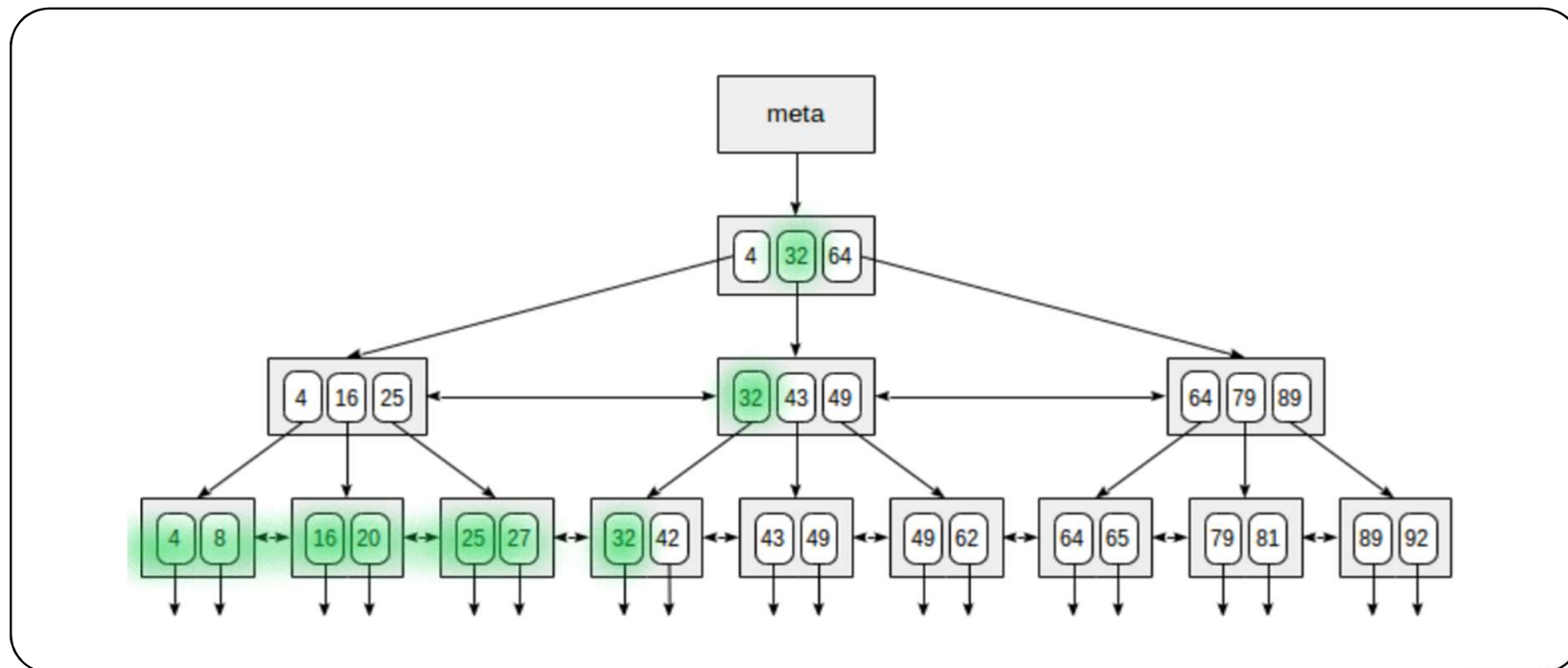
Заметка к новой версии PostgreSQL 13 – Btree может весить меньше! (и по умолчанию он это и делает)

<https://www.cybertec-postgresql.com/en/b-tree-index-deduplication/>

## BTREE. ПОИСК ПО РАВЕНСТВУ = 42



## BTREE. ПОИСК ПО НЕРАВЕНСТВУ $< 33$



# ПЛЮСЫ И МИНУСЫ

## Плюсы :



- Довольно быстрый индекс - сложность  $O(\log n)$  [Вычислительная сложность — Википедия](#)
- Широкий диапазон условий для выборки по условиям

## Минусы:



- Если при очередном добавлении/изменении/удалении значения требуется балансировка - довольно медленно. Именно поэтому изначально плотность заполнения 90%
- Необходимо их обслуживать - будет рассмотрено дальше в курсе
- Работает хорошо только с простыми типами данных (числа, даты, денежные)
- Работает со строками, только если ищем по шаблону с начала строки (%АБВ)





# СТАНДАРТНЫЕ ВИДЫ



## Простой

```
create index uk_test_id on test(id);
```

<https://www.db-fiddle.com/f/97z9hNo3ZqFCXUvN9DNAV5/0>

---

## Уникальный

```
create unique index uk_test_col2 on test(col2);
```

<https://www.db-fiddle.com/f/v98FqExg2Lw7MLeb2PKPhR/0>



Не забываем, что если не добавить описание NOT NULL - возможны аномалии  
NULL != NULL

# ПОДВЕДЕНИЕ ИТОГОВ



## ИТОГИ ЗАНЯТИЯ



01



Понять, зачем нужны  
btree индексы

02



Понять архитектуру btree  
индекса

03



Узнать их плюсы  
и минусы

04



Создали btree  
индекс на таблице



## ЗАДАНИЕ ДЛЯ САМОПРОВЕРКИ



### Цель задания:

Создать B-tree индекс на своей таблице

---

### Пошаговый план выполнения:

1. Взять за основу фидл <https://www.db-fiddle.com/f/97z9hNo3ZqFCXUvN9DNAV5/0>
2. Сгенерировать 100 000 записей
3. Посмотреть план запроса и убедиться, что индекс используется
4. Закомментировать строку создания простого индекса
5. Посмотреть план запроса и убедиться, что индекс НЕ используется

Задание закончено

# СПАСИБО!

На следующем занятии мы рассмотрим тему:

- Составной индекс, по выражениям и частичный

end