

БЛОК 8. ИНДЕКСЫ

ВИДЫ, ПЛЮСЫ И МИНУСЫ ИСПОЛЬЗОВАНИЯ ИНДЕКСОВ



begin



ЦЕЛЬ



01

Узнать, что такое
индексы

02

Понять задачи индексов

05

Познакомиться
с best practice

03

Определить основные виды
индексов

04

Узнать основные
заблуждения

СОДЕРЖАНИЕ УРОКА



1

Задачи индексов

2

Виды индексов

3

Best practice



ИНДЕКСЫ



Что же такое индекс? Самая простая аналогия - оглавление книги:

Оглавление

Об авторе	4
1. PostgreSQL 14. Установка, новые возможности	6
2. Физический уровень	28
3. Работа с консольной утилитой psql	57
4. ACID & MVCC. Vacuum и autovacuum	63
5. Уровни изоляции транзакций	79
6. Логический уровень	89
7. Работа с правами пользователя	105
8. Журналы	122
9. Блокировки	155
10. Настройка PostgreSQL	183
11. Работа с большим объемом реальных данных	206



ИНДЕКСЫ



Мы знаем, на какой странице расположена нужная нам информация и можем сразу туда перейти, нам не нужно перелистывать книгу с начала.



С индексами тоже самое - всегда знаем, где искать ту или иную информацию.

СОЗДАНИЕ ИНДЕКСА



```
CREATE [ UNIQUE ] INDEX [ CONCURRENTLY ] [ [ IF NOT EXISTS ] имя ] ON
имя_таблицы [ USING метод ]
    ( { имя_столбца | ( выражение ) } [ COLLATE правило_сортировки ] [
класс_операторов ] [ ASC | DESC ] [ NULLS { FIRST | LAST } ] [, ...] )
    [ INCLUDE ( имя столбца [,...] ) ] для Btree и Gist
    [ WITH ( параметр_хранения = значение [, ...] ) ]
    [ TABLESPACE табл_пространство ]
    [ WHERE предикат ]
```

Методы: **btree**, **hash**, **gist**, **spgist**, **gin**, **brin**

<https://www.postgresql.org/docs/14/sql-createindex.html>



ВИДЫ ИНДЕКСОВ



Btree (btree).

Сбалансированное дерево (по умолчанию).

Используется в 99% индексов.

Применяется для любого типа, который можно отсортировать в чётко определённом линейном порядке. Работает с операторами сравнения $>$, $<$, $=$, $>=$, $<=$, BETWEEN и IN и условия пустоты IS NULL и IS NOT NULL

Хэш-индекс (hash).

Работает только с условием равенства ($=$). В условиях IS NULL и IS NOT NULL также не используется.

GiST индекс (Generalized Search Tree).

Или обобщённое поисковое дерево. Базовый шаблон, на основе которого могут реализовываться произвольные схемы индексации, например, Btree, R-деревья и другие схемы индексации. Для построения используют один из нескольких алгоритмов, наиболее подходящих под тип индексируемого поля, поэтому набор операторов зависит от типа поля.

Применяется для специфических типов данных: геометрии, сетевые адреса, диапазоны.



ВИДЫ ИНДЕКСОВ



SP-GiST индекс (Space-Partitioned GiST).

Это GiST с разбиением пространства.

Метод поддерживает деревья поиска с разбиением, что позволяет работать с различными несбалансированными структурами данных (деревья квадрантов, k-мерные и префиксные деревья).

Как и GiST, SP-GiST позволяет разрабатывать дополнительные типы данных с соответствующими методами доступа.

GIN индекс (Generalized Inverted Index).

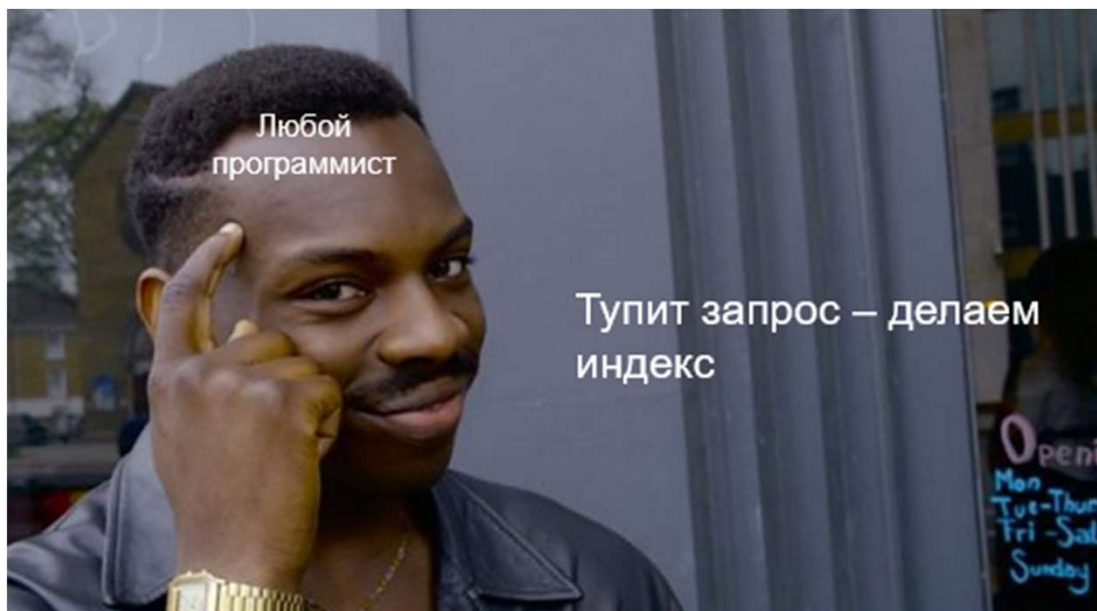
Это обобщённый инвертированный индекс.

Применяется к составным типам, работа с которыми осуществляется с помощью ключей: массивы, jsonb.

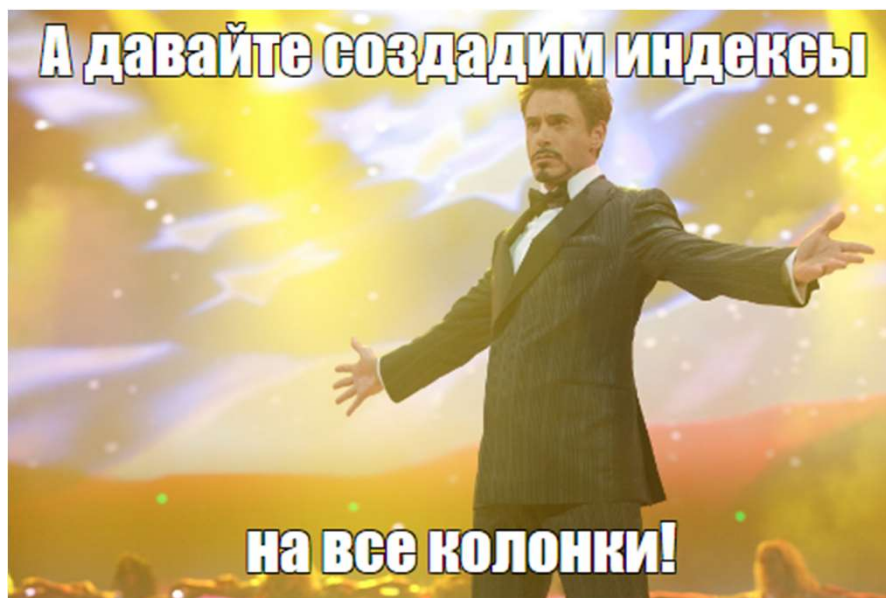
Предназначается для случаев, когда индексируемые значения являются составными, а запросы ищут значения элементов в этих составных объектах.

Самый распространённый вариант использования индексов GIN & GiST - полнотекстовый поиск по аналогии с Google/Yandex.

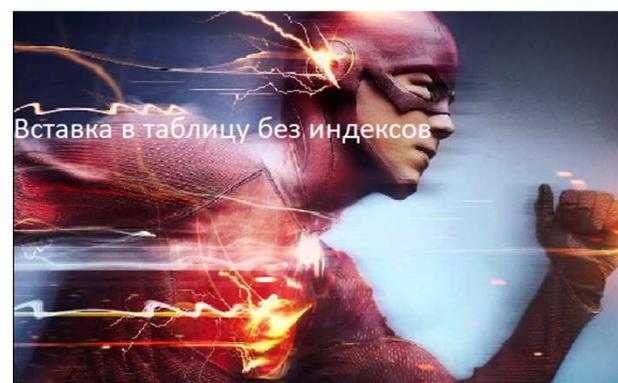
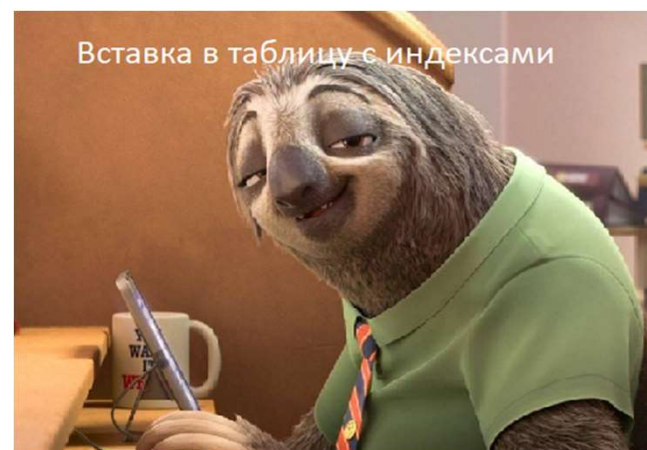
ОТЛИЧНАЯ ИДЕЯ!!! ИЛИ НЕТ?



ОТЛИЧНАЯ ИДЕЯ!!! ИЛИ НЕТ?



НЕДОСТАТКИ ИНДЕКСОВ



ПЛЮСЫ И МИНУСЫ ИНДЕКСОВ

Плюсы индексов:



- Ускоряют выборку в операциях SELECT
- При выборке данных только индексного поля, данные из таблицы не выбираются
- Увеличение скорости сортировки по индексному полю
- Обеспечение уникальности

Минусы индексов:



- Индексы требуют дополнительного места
- Необходимо перестраивать индексы при операциях UPDATE, DELETE, INSERT
- При большом количестве индексов оптимизатору сложно выбрать какой использовать



BEST PRACTICE

01

Выбираем индекс исходя из условий бизнес задачи

02

Возможно стоит попробовать разные индексы - результат может отличаться на разных наборах данных

03

Не забываем про другие типы индексов, которые рассмотрим на следующем занятии - функциональный и покрывающий - помогут нам на аналитических запросах

04

Не забывать анализировать существующие индексы!!!

05

Удаляем неиспользуемые индексы

06

Добавляем необходимые

07

Индексы необходимо обслуживать



ПОДВЕДЕНИЕ ИТОГОВ



ИТОГИ ЗАНЯТИЯ



01

Поняли задачи
индексов



02

Поняли основные виды
индексов



03

Поняли основные
заблуждения



04

Узнали best practice



СПАСИБО!

На следующем занятии мы рассмотрим тему:

- B-tree

end