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

Типы индексов:

Hash, B-tree, GiST, SP-GiST, GIN и RUM, BRIN и Bloom.

Индекс (метод доступа) — это отсортированный список значений полей (конкретной колонки в частности), предназначенный для ускорения поиска в базе данных.

Индекс устанавливает соответствие между ключом (значение индексируемого столбца) и строками таблицы, в которых этот ключ встречается. Строки идентифицируются с помощью номера блока файла и позиции строки внутри блока.

То есть индекс это отдельно взятая отсортированная по значению колонка (её значения) с идентификацией нахождения связанных данных по нему.

Обновление полей таблицы, по которым не создавались индексы,

не приводит к перестроению индексов.

Этот механизм называется HOT (Heap-Only Tuples).

На скорость работы индекса влияет селективность.

-- Seq Scan Последовательное сканирование. Читается последовательно вся таблица.

-- Index Scan Индексное сканирование. Используется индекс для условий WHERE, читает таблицу при отборе строк. Если данных не много.

-- Bitmap Index Scan Сканирование по битовой карте. сначала Index Scan, затем контроль выборки по таблице. Эффективно для большого количества строк. Чтобы не обращаться к одной странице данных несколько раз.

-- Index Only Scan Покрывающий индекс. Самый быстрый. Читается только индекс.

Покрывающий индекс это когда сам индекс содержит все необходимые данные которые нужно получить. В таком случае сканирования страниц таблицы на предмет получения данных как правило не происходит. (Зависит от очистки vacuum)

Не все типы индексов (методы доступа) поддерживают создание индексов по нескольким столбцам (составной индекс).

Индексы по выражениям или Функциональные индексы - это индексы по выражениям, или индексы с использованием функций. Пример: create index on t(lower(b));

Частичный индекс. Или индекс на часть строк (с наименьшим числом уникальных значений)

Такой индекс строиться по удовлетворяющему условию. Пример: create index on t(c) where c;

Колонка `c` в данном случае имеет тип bool.

999 360 имеет значение false

640 имеет значение true

(вот на строки имеющие такое значение можно построить частичный индекс)

Преимущество частичного индекса что он занимает меньше места и работает быстрее.

Сортировка. Использование индекса в запросах с сортировкой.

Из всех методов доступа только btree умеет возвращать данные в отсортированном виде,

Блокировки при создании индексов.

При создании индекса на таблицу устанавливается блокировка SHARE.

Такая блокировка разрешает читать данные из таблицы. Но запрещает их изменять до завершения создания индекса (update, insert, delete)

create index concurrently on t(a);

позволяет создавать индекс параллельно изменениям в таблице. Но это не рекомендуется…

### Индекс Btree

При создании индекса можно задать направление сортировки.

Актуально для запросов с разной сортировкой по нескольким полям (ASC, DESC).

Порядок полей в составном индексе очень важен для эффективного использования.

Не определённые значения (NULL)

Порядок не определённых значений в индексе тоже важен при использовании в запросах.

Пример если нужно поставить значения null в начале индекса:

create index flights\_nulls\_first\_idx on flights(actual\_arrival nulls first);

Уникальные индексы с дополнительными столбцами.

Это решение для создание покрывающего индекса.

Для ускорения запросов в необходимых решениях.

Создавать индексы лучше после заполнения таблицы данными.

Если есть такая возможность.

### Индекс GiST

Данный тип индекса используется в геометрии. направление сортировки, гео локации.

Индексирование географических координат. Индексирование двухмерных массивов.

### Индекс GIN

Данный тип индекса используется для полнотекстового поиска.

Для индексирования JSON колонок.