

Индексы

Индексы в PostgreSQL нужны для ускорения операций чтения, особенно в больших базах данных.

Индекс в PostgreSQL – это специальная структура данных, которая помогает ускорить выборку данных. Он позволяет СУБД находить и извлекать конкретные строки из таблицы быстрее, чем при последовательном поиске.

Индекс содержит ключи (значения одного или нескольких столбцов) и указатели на соответствующие строки в таблице. Поиск по индексу значительно быстрее, чем полное сканирование таблицы.

Пример: Представьте, что база данных – это большая библиотека, а таблица в базе данных – это одна из её книжных полок. У каждой книги (записи в таблице) есть уникальный номер (например, ID). Если вам нужно найти книгу по её номеру, и у вас нет каталога (индекса), вам придётся просмотреть каждую книгу на полке (полное сканирование таблицы).

Теперь представьте, что у вас есть каталог, в котором указано, на какой полке и в каком месте находится каждая книга по её номеру. Этот каталог – это индекс. Вместо того чтобы просматривать каждую книгу, вы быстро находите нужную информацию в каталоге и напрямую идёте к нужной книге.

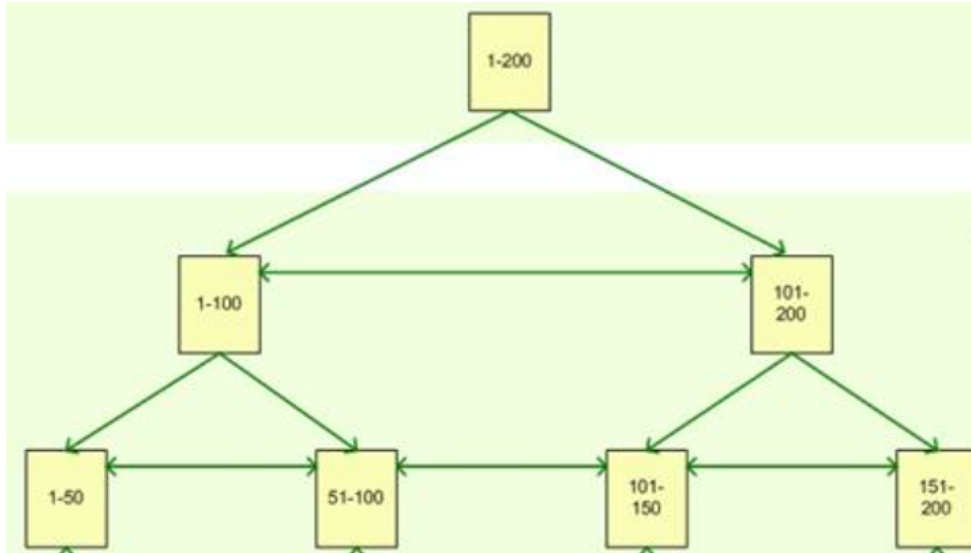
Как это работает:

1. Создание индекса:

- Когда вы создаёте индекс в PostgreSQL, например, на столбец `customer_id` в таблице `orders` , (командой `CREATE INDEX idx_customer_id ON orders(customer_id);`) СУБД создаёт структуру данных для значений `customer_id` . То есть он делит все значения этого поля на группы, а каждой группе выделяет группу строк в таблице (обычный B-tree индекс).

2. Использование индекса:

- При выполнении запроса, такого как `SELECT * FROM orders WHERE customer_id = 123;`, PostgreSQL сначала проверяет индекс. Как на картинке ниже, по дереву спускаясь всё ниже, СУБД ищет ту самую "группу", куда входит значение поля. А потом уже в зависимости от выбранной группы оставляет только нужные строки для выполнения запроса, а не все.



- Если значение `123` найдено в индексе, СУБД использует информацию из индекса, чтобы напрямую обратиться к соответствующим строкам в таблице `orders`, минуя необходимость полного сканирования всей таблицы.

Лучшие практики при использовании индексов:

1. Индексируйте столбцы, которые часто используются в предложениях `WHERE`, `JOIN`, `ORDER BY` и `GROUP BY`.

2. Избегайте избыточной индексации:

- Каждый индекс увеличивает время, необходимое для операций вставки, обновления и удаления.
- Не создавайте индексы на столбцах, которые редко используются в запросах.

3. Используйте составные индексы(из нескольких столбцов) аккуратно:

- Составные индексы полезны, когда запросы фильтруются по нескольким столбцам.

- Порядок столбцов в составном индексе важен.

4. Используйте индексы в столбцах с разными данными:

- Индексы эффективнее для столбцов, которые наполнены различными данными. Например, если у вас есть столбец с полом человека (то есть только два вида значений могут быть), то индекс бесполезен на этом столбце.