Способы повышения производительности запросов:

* Использование индексов.
* Настройка физических параметров СУБД (способ разделения пространства хранения данных, стратегии работы с транзакциями и т. д.).

Индекс SQL – список всех значений в группе из одного или нескольких столбцов, упорядоченный в некотором приемлемом для данного типа данных смысле (например, в порядке возрастания для чисел или в алфавитном порядке для символьных строк).

Каждое значение имеет указатель на строку в таблицу, где это значение встречается.

Индексы работают **неявно**:

* При выполнении запроса СУБД определяет, какие индексы нужно использовать.
* В рамках того или иного запроса СУБД может не использовать индекс.

*Правила работы с индексами:*

Индексы создаются по таблицам для ускорения операций, включающих:

* предложения WHERE и JOIN
* определение значений MIN() или MAX() по индексированному столбцу;
* сортировку и группировку столбцов таблицы

**Первый недостаток** индексов, что он *занимает память*.

**Второй недостаток** индексов, что при изменении/удалении содержимого индексированного столбца/при добавлении новой строки индекс необходимо обновлять. Эти действия *замедляют операции*.

**!!Индексы неэффективны, если в таблице мало строк!!**

**Четвертый недостаток** индексов, что они могут быть неэффективными, если по условия *выбираются большие объемы данных*.

**Стратегии применения индексов:**

* Какие операции будут применяться к таблицам: запрос данных или обновление таблицы?
* Какие столбцы и как часто будут использоваться в предикатах?
* Как часто столбцы таблиц будут использоваться в соединениях (join)?

**Создание индексов(1):**

CREATE INDEX index\_name on table\_name (column\_name);

CREATE INDEX index\_name (column1\_name, column2\_name);

**Создание индексов(2):**

Индекс B-дерева – это широко используемые структуры данных для индексации. Это метод многоуровневого индексного формата, который сбалансирован бинарными деревьями поиска. Все конечные узлы дерева B обозначают фактические указатели данных.

CREATE INDEX index\_name ON table\_name USING btree(column1, column2);

Дерево – связный ациклический граф.

Связность – означает наличие путей между любой парой вершин.

Ацикличность – отсутствие циклов и то, что между парами имеется только по одному пути.

B-tree (balanced tree index) – индекс сгруппированный по листьям сбалансированного дерева.

tid – tuple identifier, идентификатор записи (номер блока, индекс внутри блока).

2 типа вершин: внутренние (ссылки на дочерние стр.) и листовые страницы (ключ + tid).

**B-tree index:**

* Значения (ключи) внутри каждого узла отсортированы.
* Дерево сбалансировано ключи равномерно распределены по узлам, что позволяет минимизировать количество переходов.
* Полезен при использовании совместно с =,>,>=,<,<=,BETWEEN и подобными операторами.

Hash-index:

Для построение такого индекса используется хэш-функция.

Хэш-функция – функция для преобразования входных данных в результирующие данные фиксированного формата.

* Очень эффективны, когда используется прямое сравнение (на attr – hash index):

… WHERE attr = 1

* Но индекс не будет применен, если:

… WHERE attr = 1 OR A = 10

Индексы в PostgreSQL:

* GiST – обобщенное дерево поиска.
* GIN – обобщенный инвертированный индекс (для ускорения полнотекстового поиска).

**Выполнения запросов и оптимизация.**

Зачем это нужно?

Язык SQL декларативен:

* В запросах указывается, какими должны быть данные, которые необходимо получить.
* Не говорится о том, как система должна выполнить запрос.

**Выполнение запросов:**

1. Разбор запроса (parser) -> строится дерево
2. Преобразование запроса (rewriter)
3. Планировщик + Оптимизатор (planner) -> план выполнения запроса.
4. Выполнение плана -> executor

**План выполнения запроса:**

* Чтобы выполнить SQL-запрос необходимо построить программу – **план выполнения запроса.**
* Таких программ может быть несколько.

**Выбор плана выполнения запроса:**

Критерий: оценочная стоимость выполнения запроса по данному плану.

Компоненты оцениваемой стоимости:

* Число обменов с устройствами внешней памяти, которые потребуются при выполнении плана запроса;
* Среднее время обмена;

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание  
Изображение выглядит как текст, Шрифт, снимок экрана, белый

Автоматически созданное описание  
Тета-соединение (пример):  
Изображение выглядит как текст, снимок экрана, Шрифт, логотип

Автоматически созданное описание

Построение плана (1):

1. Сначала выполнить тета соединение, где мы найдем студентов, которые будут сдавать именно этот экзамен по айди.
2. Сделаем выборку, где группа студентов должна быть равна 3100.

Построение плана (2):

1. 1 пункт продолжение плана (1): делаем выборку, где айди студента должен быть больше или равен 150000.