Николай Лапшин

- Tech Lead в компании AdTech + мендежер инженеров.
- Опыт лет 10+.
- В отусе виду на 9 курсов.

О чем лекция

- Индексы
- Типы индексов
- EXPLAIN
- Практика

Индекс – это структура данных, которая позволяет СУБД быстро находить и получать доступ к данным в таблице. Индексы хранятся отдельно от основных таблиц данных, что позволяет эффективно искать информацию без необходимости просматривать каждую строку таблицы.

Зачем

Принципы

- Каждая строка в таблице имеет идентификатор TID (tuple id), который состоит из номера блока файла и позиции строки внутри блока.
- Вместо сканирования всей таблицы получить только те строки, которые подходят под ключи поиска и получить TID и, следовательно, нужные записи.
- При вставке/обновлении/удалении - индекс перестраивает в той же транзации. Если операция не приводит к перестроению индекса, то это называет HOT(Heap-Only Tuples)
- Оптимизатор просматривает доступные индексы и выбирает оптимальный план запроса.
- Важную роль играет сбор статистики(отдельная лекция), так как позволяте оптимизатору выбрать правильный план запроса.

Структура

```
CREATE INDEX my_index ON my_table USING btree (my_column);
```

```
CREATE UNIQUE INDEX my_unique_index ON my_table (my_column COLLATE
pg_catalog.default DESC NULLS LAST) INCLUDE (other_column);
```

```
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 ]
```

Основная структура

- `CREATE [ UNIQUE ] INDEX [ CONCURRENTLY ] [ [ IF NOT EXISTS ] имя ] ON имя_таблицы [ USING метод ]`:
  - `CREATE INDEX`: команда для создания нового индекса.
  - `UNIQUE`: опционально указывает, что индекс будет уникальным (не допускающим повторяющихся значений). Может быть много NULL.
  - `CONCURRENTLY`: опционально указывает, что индекс будет создан без блокировки таблицы на время его создания. https://postgrespro.ru/docs/postgresql/9.6/sql-createindex#sql-createindex-concurrently
  - `IF NOT EXISTS`: опционально указывает, что индекс будет создан только в том случае, если он еще не существует.
  - `имя`: имя нового индекса.
  - `ON имя_таблицы`: таблица, для которой создается индекс.
  - `USING метод`: метод индексирования, например, BTree или Gist.

Описание индексируемых столбцов и выражений

- `( { имя_столбца | ( выражение ) } [ COLLATE правило_сортировки ] [ класс_операторов ] [ ASC | DESC ] [ NULLS { FIRST | LAST } ] [, ...] )`:
  - `имя_столбца`: столбец, который будет индексироваться.
  - `( выражение )`: выражение для создания индекса на его основе.
  - `COLLATE правило_сортировки`: опционально задает правило сортировки для столбца.
  - `класс_операторов`: опционально указывает класс операторов для столбца.
  - `ASC | DESC`: опционально задает порядок сортировки (возрастающий или убывающий).

- `NULLS FIRST | LAST`: опционально задает положение NULL-значений в индексе.

Дополнительные опции

- `INCLUDE ( имя столбца [, ...] )`: опционально указывает столбцы, которые должны быть включены в индекс для ускорения доступа к данным. Дополнительные атрибуты, которые не участвую в сортировке, но есть в структуре. Если используем в SELECT, но не WHERE/JOIN/ORDER BY
- `WITH ( параметр_хранения = значение [, ... ] )`: опционально задает параметры хранения для индекса. Например, fillfactor = 90 процентов. Количество данных на страницу, когда считается, что индекс заполнен.
- `TABLESPACE табл_пространство`: опционально указывает табличное пространство, в котором будет создан индекс. Куда сохранить индекс
- `WHERE предикат`: опционально создает частичный индекс, который индексирует только строки, соответствующие указанному условию (предикату).
- Обычный индекс по колонке
- Составной индекс по нескольким колонкам
- Функциональный индекс колонка через функцию
- Частичный индекс(с предикатом where).

Costs

```
costs = (чисто чтений диска * seq_page_cost) + (число просканнированых строк
* cpu_tuple_cost)
```

`jit_above_cost` - опция, включает JIT-компиляции

Советы:

- Строить индексы по частым запросам.
- Не строить индексы по полям с низкой кардинальностью(мало разнообразных значений).
- Держать только актуальные индексы(исследовать статистику на частоту использования)