**Липецкий государственный технический университет**

Факультет автоматизации и информатики Кафедра автоматизированных систем управления

Задание №4

по дисциплине «Расширенный курс по базам данных»

Индексы. Планировщик.

Студент Титов В.А.

Группа АС-20-1

Руководитель Мирошников А.И.

Липецк 2023 г.

Задание

**Общая часть:** ознакомиться с теоретическими сведениями о возможностях создания индексов и ограничений в PostgreSQL. Добавить в какую-либо из таблиц, созданных в предыдущих лабораторных работах 10000 записей (таблица не должна содержать полей типа date, но должно быть хотя бы одно поле типа char).

**Индивидуальная часть:**

‒ выполнить 6 запросов с сортировкой и фильтрацией (см. пример) для собственных таблиц;

‒ отобразить планы выполнения запросов;

‒ создать различные типы индексов (B-дерево, хеш, GIN и BRIN) + частичные индексы;

‒ отобразить планы выполнения запросов с созданными индексами;

‒ создать таблицу с полем типа JSON, создать GIST индекс по этому полю, убедиться, что он используется при выборке. Пример предметной области - таблица, хранящая ответы API в формате json. Найти записи с ошибками (в json-объекте присутствует ключ “error”).

Ход работы

**Общая часть**

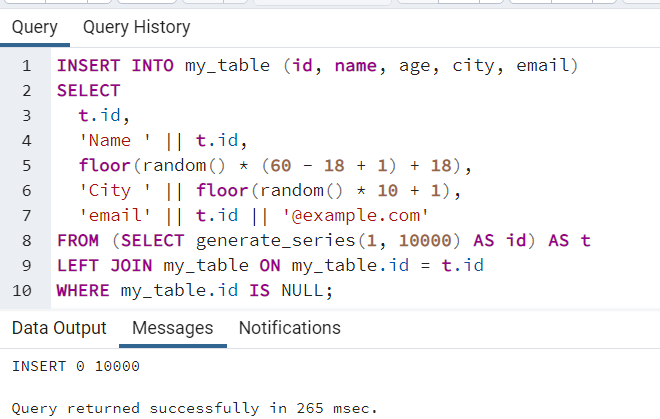


Рис.1. Добавляем 10000 записей в таблицу

Используем функцию `generate\_series()` для создания уникальных идентификаторов для каждой записи в таблице. Также используем функцию `floor(random() \* (60 - 18 + 1) + 18)` для генерации случайных значений для поля `age`. Для полей `name`, `city` и `email` мы используем значения из `id`, чтобы создать уникальные строки.

Также используем оператор `LEFT JOIN` и условие `WHERE my\_table.id IS NULL`, чтобы исключить дубликаты и добавить только те записи, которых еще нет в таблице.

**Индивидуальная часть**

1. Выполнить 6 запросов с сортировкой и фильтрацией для собственных таблиц.
2. Выбрать все записи из таблицы `my\_table`, где возраст больше 30, отсортировать по возрастанию идентификатора

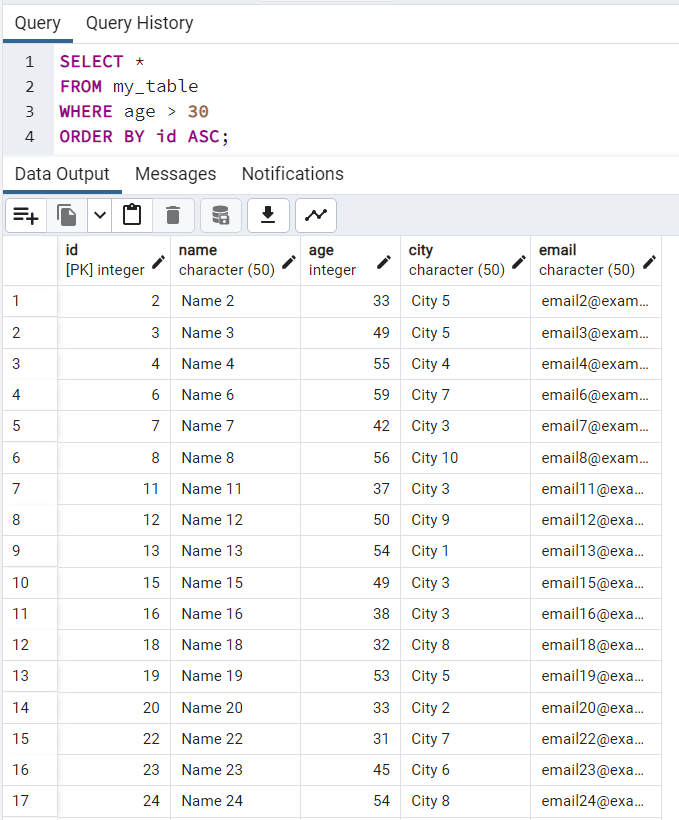


Рис.2. Первый запрос

1. Выбрать все записи из таблицы `my\_table`, где город равен 'City 5', отсортировать по убыванию возраста

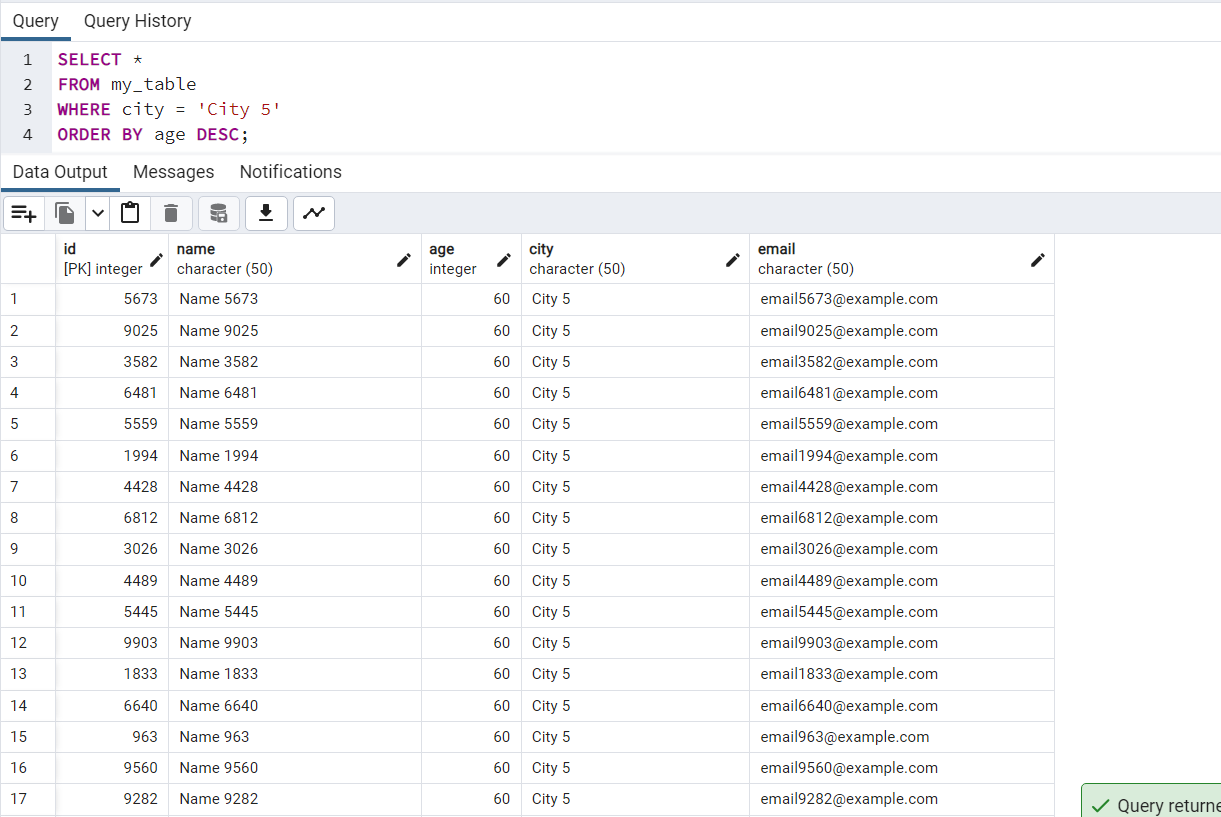


Рис.3. Второй запрос

1. Выбрать последние 10 записей из таблицы `my\_table`, отсортировать по убыванию идентификатора

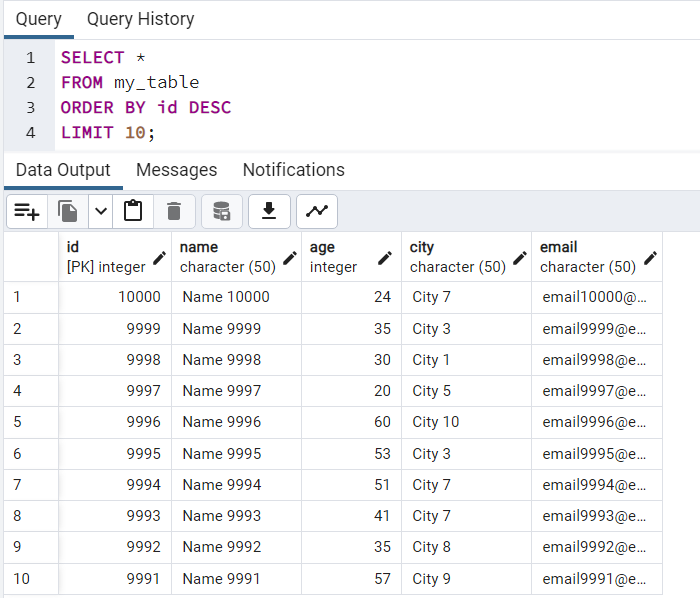


Рис.4. Третий запрос

1. Сортировка по имени по возрастанию и фильтрация по возрасту

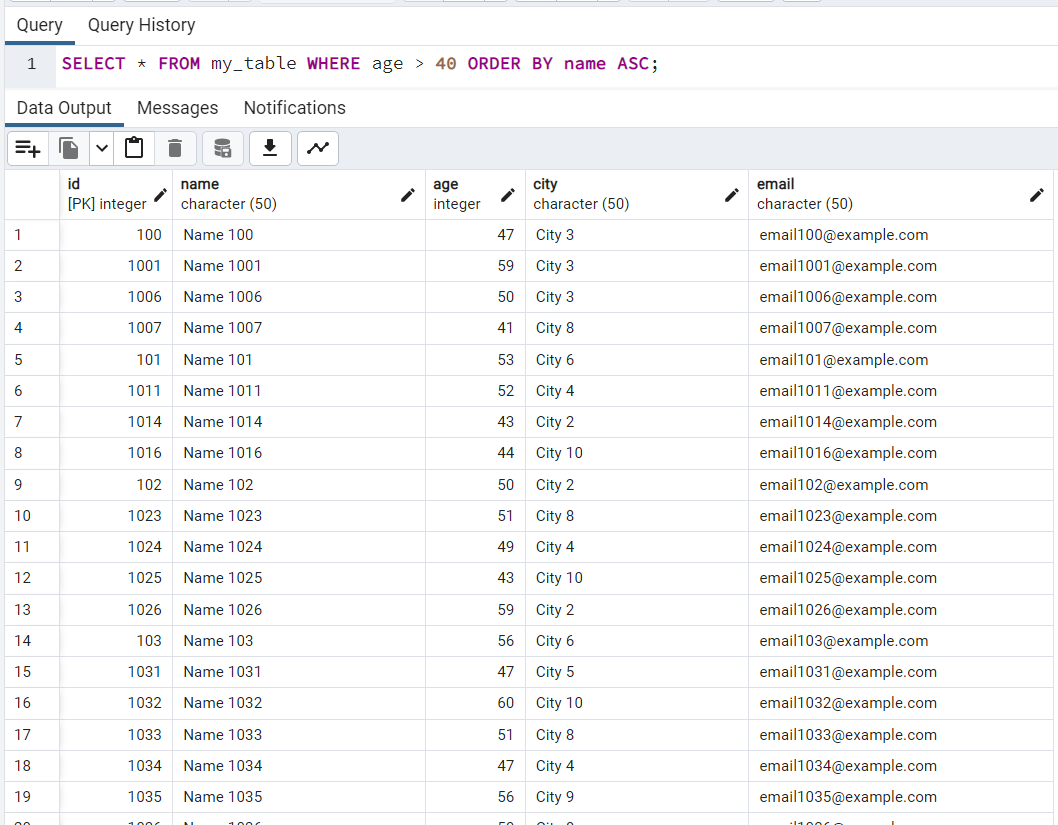


Рис.5. Четвертый запрос

1. Сортировка по возрасту по убыванию и фильтрация по имени

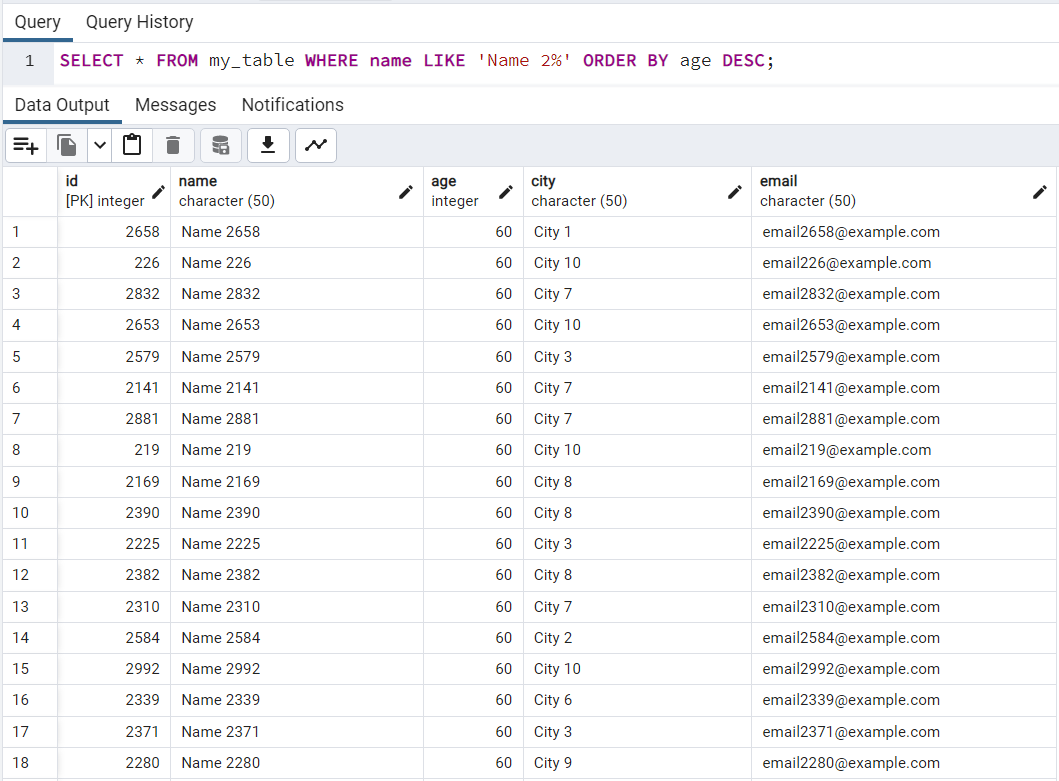


Рис.6. Пятый запрос

1. Сортировка по городу по возрастанию и фильтрация по возрасту

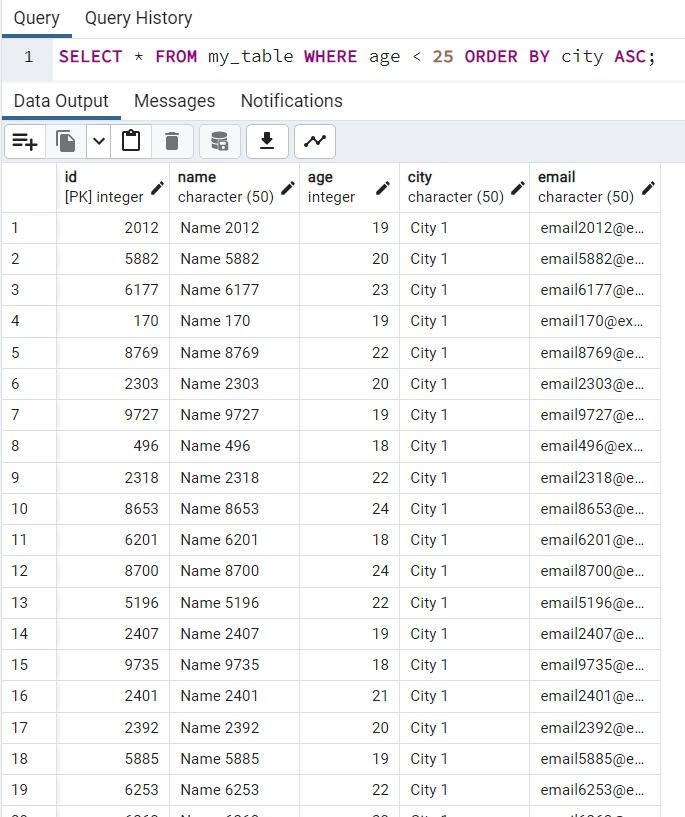


Рис.7. Шестой запрос

1. Отобразить планы выполнения запросов
2. План первого запроса

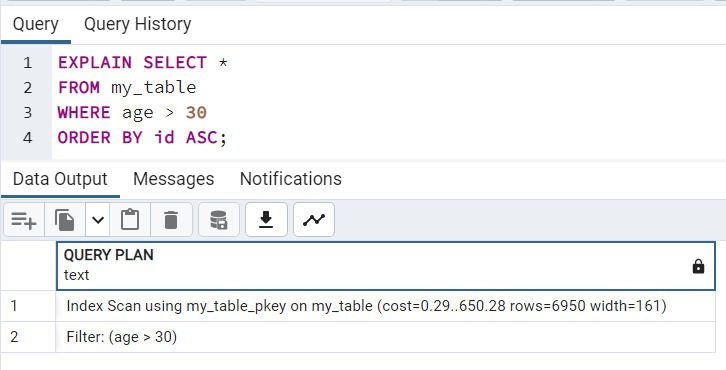


Рис.8. Первый план

1. План второго запроса

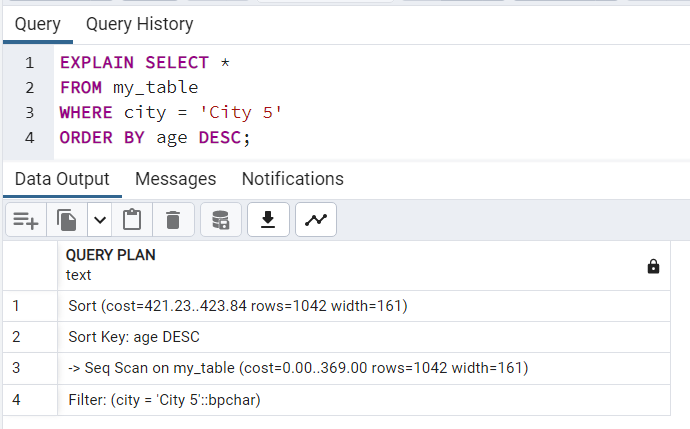


Рис.9. Второй план

1. План третьего запроса

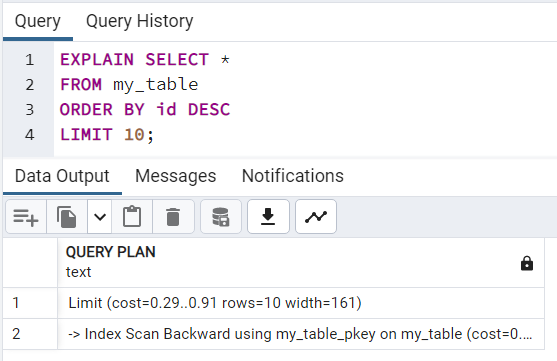


Рис.10. Третий план

1. План первого запроса

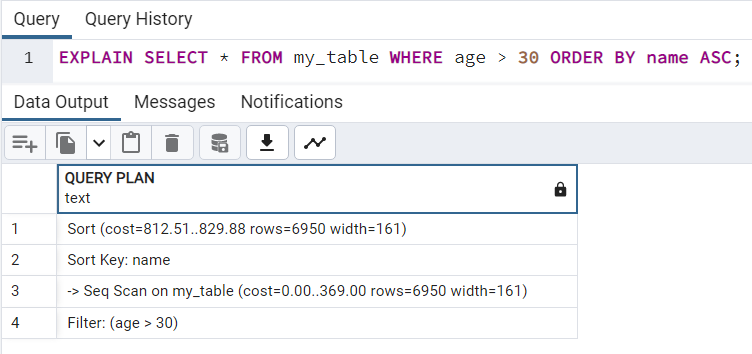


Рис.11. Четвертый план

1. План первого запроса

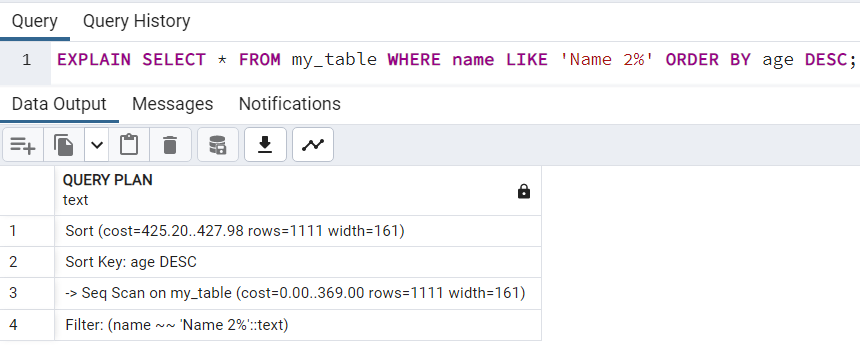


Рис.12. Пятый план

1. План первого запроса

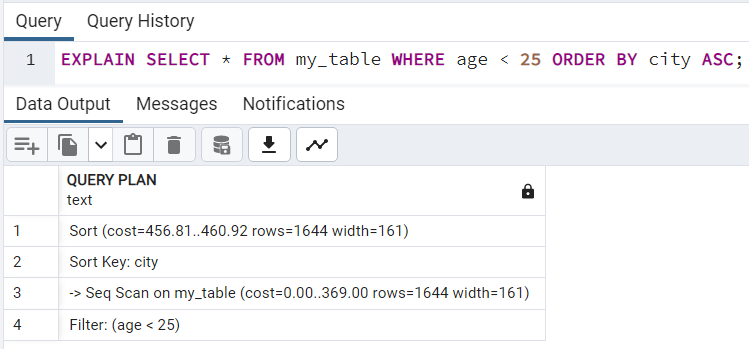


Рис.13. Шестой план

1. Создать различные типы индексов (B-дерево, хеш, GIN и BRIN) + частичные индексы
2. Создание B-дерево индекса на столбце "age"

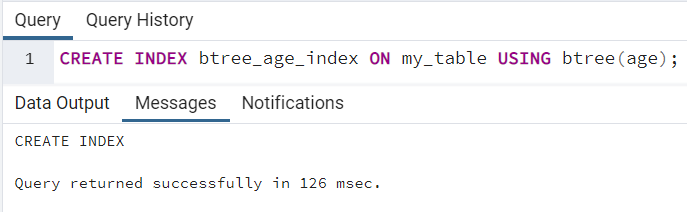


Рис.14. В-дерево

1. Создание хеш индекса на столбце "city"

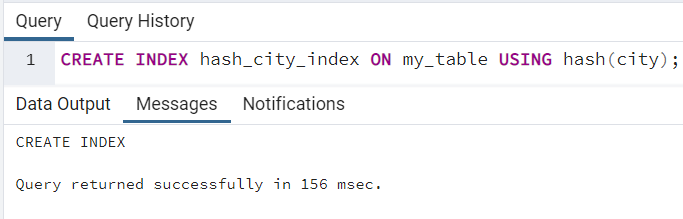


Рис.15. Хеш

1. Создание GIN индекса на столбце "name"

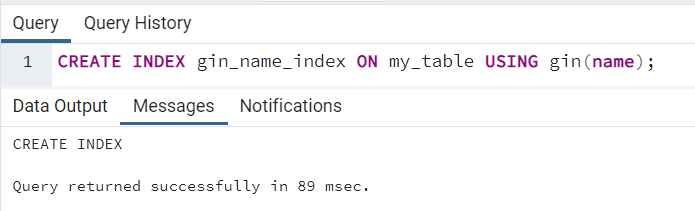


Рис.16. GIN

1. Создание BRIN индекса на столбце "id" с шагом 1000

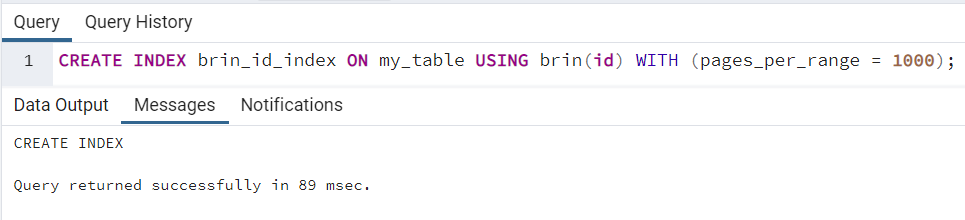


Рис.17. BRIN

1. Создание частичного индекса на строках, где значение столбца "age" больше или равно 30

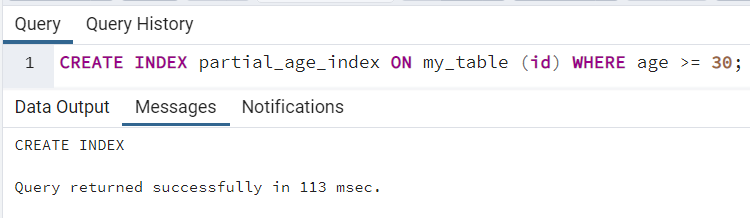


Рис.18. Частичный

1. Отобразить планы выполнения запросов с созданными индексами
2. Поиск строк по столбцу "age" с использованием B-дерево индекса

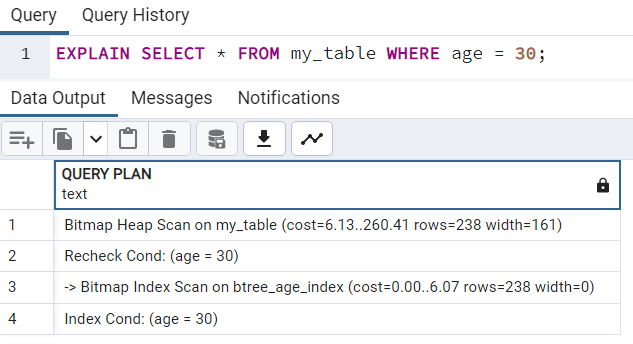


Рис.19. Первый план

1. Поиск строк по столбцу "city" с использованием хеш индекса

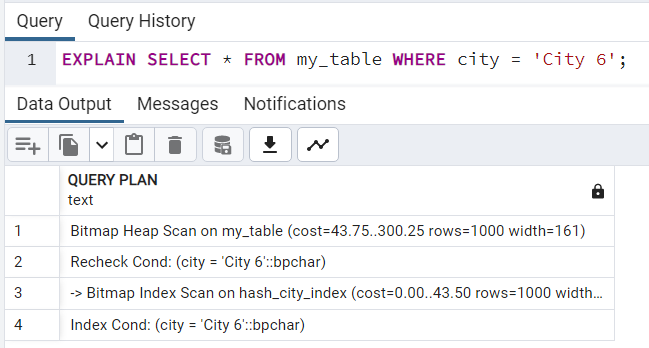


Рис.20. Второй план

1. Поиск строк по столбцу "name" с использованием GIN индекса

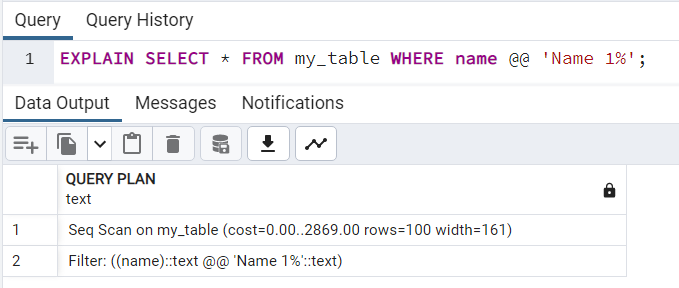


Рис.21. Третий план

1. Поиск строк по столбцу "id" с использованием BRIN индекса

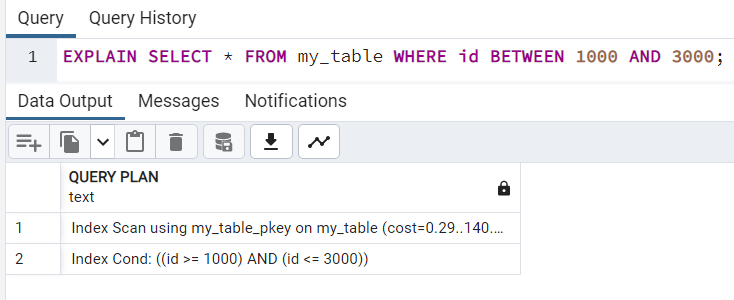


Рис.22. Четвертый план

1. Поиск строк по частичному индексу на столбце "age"

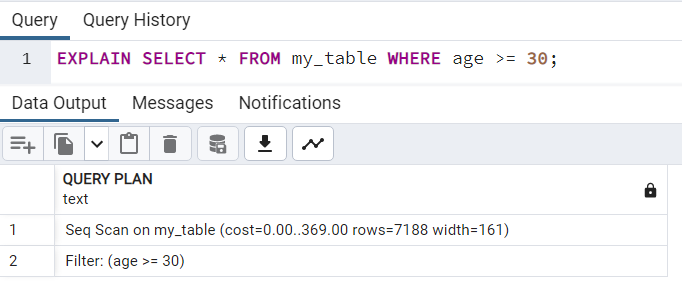


Рис.23. Пятый план

1. Создать таблицу с полем типа JSON, создать GIST индекс по этому полю, убедиться, что он используется при выборке. Пример предметной области - таблица, хранящая ответы API в формате json. Найти записи с ошибками (в json-объекте присутствует ключ “error”)
2. Создадим таблицу

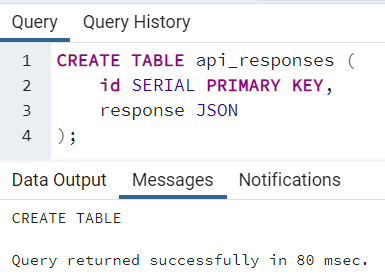


Рис.24. Создание таблицы

1. Заполним таблицу

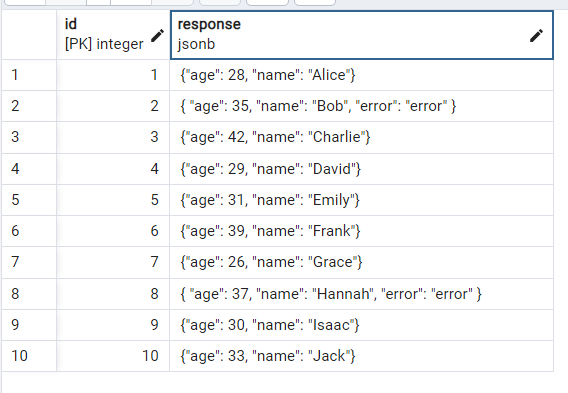


Рис.25. Заполнение таблицы

1. Далее нужно создать GIST индекс по полю response, чтобы ускорить выборку записей

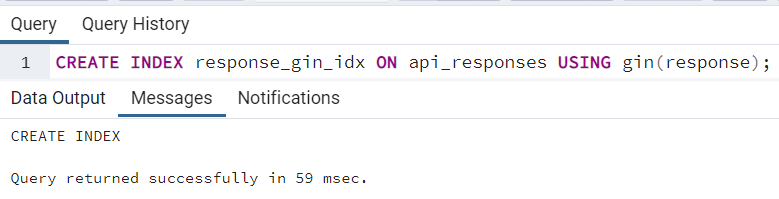


Рис.26. Индекс GIST

1. Найти записи с ошибками (в json-объекте присутствует ключ “error”)

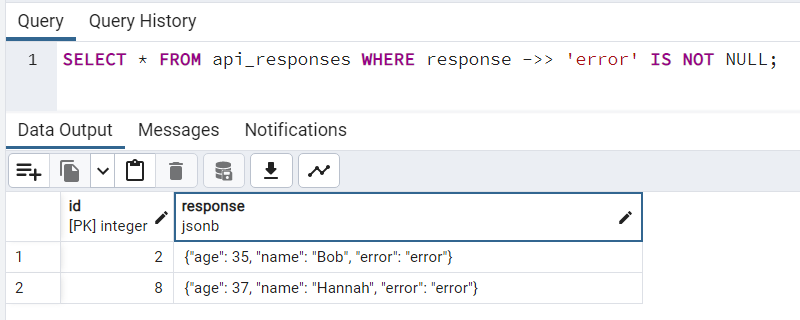


Рис.27. Поиск ошибок