Лабораторная работа №4.

1. Добавление 100000 записей в таблицу Product (генерация в текстовый файл с помощью Python)

```python
import random
measure = ['kg', 'l', 'ml', 'piece', 'g']

with open('products.txt', 'w') as f:
    for id in range(11, 100_001):
        print('(', id, ", ", "\'Product" + str(id) + "\'", ", \'", random.choice(measure),
         "\', ", str(round(random.random()*10000, 3)), "),", sep='', file=f)
```

products – Блокнот

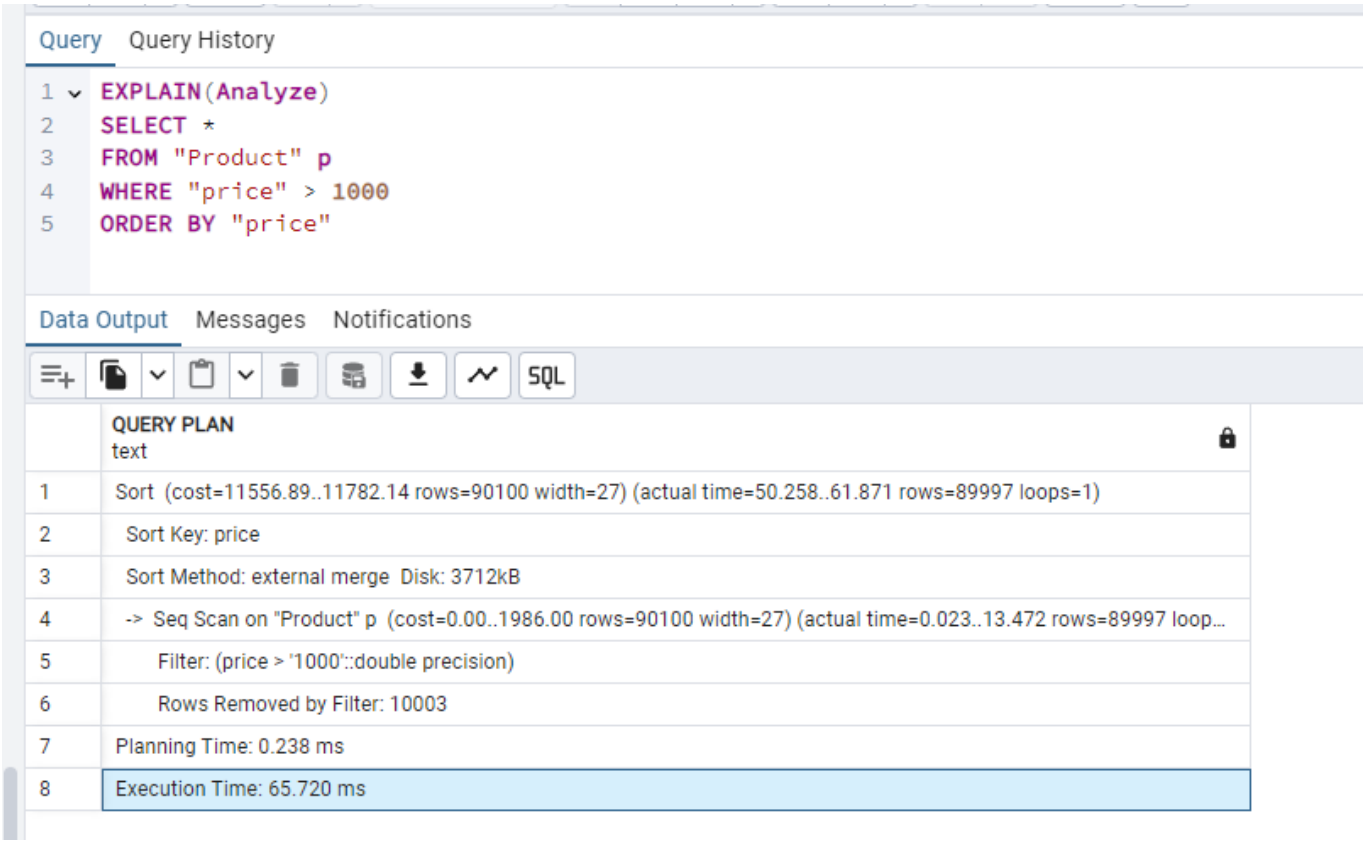
Файл Правка Формат Вид Справка

```
(83837, 'Product83837', 'ml', 2015.943),
(83838, 'Product83838', 'g', 1310.452),
(83839, 'Product83839', 'ml', 4498.342),
(83840, 'Product83840', 'ml', 1180.292),
(83841, 'Product83841', 'kg', 5373.973),
(83842, 'Product83842', 'kg', 958.628),
(83843, 'Product83843', 'piece', 5821.209),
(83844, 'Product83844', 'g', 9777.092),
(83845, 'Product83845', 'kg', 3287.463),
(83846, 'Product83846', 'l', 4648.664),
(83847, 'Product83847', 'ml', 3053.44),
(83848, 'Product83848', 'piece', 4082.592),
(83849, 'Product83849', 'ml', 103.37),
(83850, 'Product83850', 'l', 5289.113),
(83851, 'Product83851', 'l', 447.006),
(83852, 'Product83852', 'ml', 6957.721),
(83853, 'Product83853', 'kg', 603.85),
```

	id [PK] integer	name character varying (100)	measure character varying (100)	price double precision
1	99991	Product99991	ml	8579.466
2	99992	Product99992	g	7138.945
3	99993	Product99993	l	3513.283
4	99994	Product99994	g	2836.752
5	99995	Product99995	kg	5522.623
6	99996	Product99996	ml	2893.281
7	99997	Product99997	l	6301.439
8	99998	Product99998	g	6605.046
9	99999	Product99999	piece	91.812
10	100000	Product100000	kg	1140.198

(выведены последние 10 строк таблицы)

2. Выполнение запроса до создания индекса. Запрос – вывод продуктов с ценой за единицу более 1000 руб., время выполнения – 65 миллисекунд



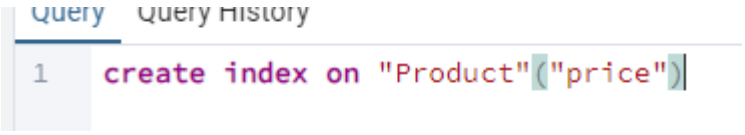
Query | Query History

```
1  EXPLAIN(Analyze)
2  SELECT *
3  FROM "Product" p
4  WHERE "price" > 1000
5  ORDER BY "price"
```

Data Output | Messages | Notifications

	QUERY PLAN text
1	Sort (cost=11556.89..11782.14 rows=90100 width=27) (actual time=50.258..61.871 rows=89997 loops=1)
2	Sort Key: price
3	Sort Method: external merge Disk: 3712kB
4	-> Seq Scan on "Product" p (cost=0.00..1986.00 rows=90100 width=27) (actual time=0.023..13.472 rows=89997 loop...
5	Filter: (price > '1000'::double precision)
6	Rows Removed by Filter: 10003
7	Planning Time: 0.238 ms
8	Execution Time: 65.720 ms

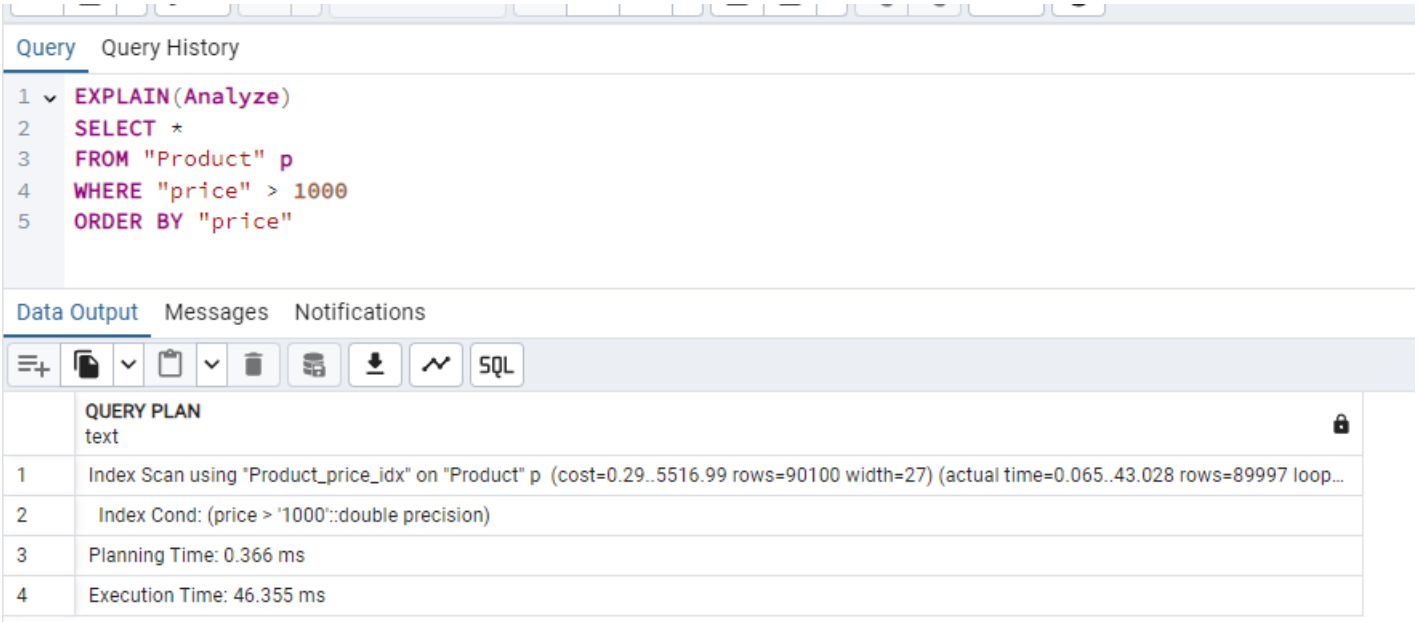
3. Создание индекса по полю price



Query | Query History

```
1  create index on "Product"("price")
```

4. Выполнение запроса после создания индекса. Время выполнения – 46 миллисекунд



Query | Query History

```
1  EXPLAIN(Analyze)
2  SELECT *
3  FROM "Product" p
4  WHERE "price" > 1000
5  ORDER BY "price"
```

Data Output | Messages | Notifications

	QUERY PLAN text
1	Index Scan using "Product_price_idx" on "Product" p (cost=0.29..5516.99 rows=90100 width=27) (actual time=0.065..43.028 rows=89997 loop...
2	Index Cond: (price > '1000'::double precision)
3	Planning Time: 0.366 ms
4	Execution Time: 46.355 ms