ДЗ 7. Глава 8 (упражнения 1, 3).

1. Предположим, что для какой-то таблицы создан уникальный индекс по двум столбцам: `column1` и `column2`. В таблице есть строка, у которой значение атрибута `column1` равно ABC, а значение атрибута `column2` — NULL. Мы решили добавить в таблицу еще одну строку с такими же значениями ключевых атрибутов, т. е. `column1` — ABC, а `column2` — NULL.

   Как вы думаете, будет ли операция вставки новой строки успешной или завершится с ошибкой? Объясните ваше решение.

```
demo=# CREATE TABLE ts (c1 text, c2 text);
CREATE TABLE
demo=# INSERT INTO ts VALUES ('ABC');
INSERT 0 1
demo=# \d ts
              Table "bookings.ts"
 Column | Type | Collation | Nullable | Default
--------+------+-----------+----------+---------
 c1     | text |           |          |
 c2     | text |           |          |

demo=# SELECT * FROM ts;
 c1  | c2
-----+----
 ABC |
(1 row)

demo=# CREATE UNIQUE INDEX unique_rest ON ts (c1, c2);
CREATE INDEX
demo=# \d ts
              Table "bookings.ts"
 Column | Type | Collation | Nullable | Default
--------+------+-----------+----------+---------
 c1     | text |           |          |
 c2     | text |           |          |
Indexes:
    "unique_rest" UNIQUE, btree (c1, c2)

demo=# INSERT INTO ts VALUES ('ABC');
INSERT 0 1
demo=# SELECT * FROM ts;
 c1  | c2
-----+----
 ABC |
 ABC |
(2 rows)
```

Вставка проходит успешно, это происходит из-за того, что NULL не может равняться NULL.

3. Известно, что индекс значительно ускоряет работу, если при выполнении запроса из таблицы отбирается лишь небольшая часть строк. Если же эта доля велика, скажем, половина строк или более, то большого положительного эффекта от наличия индекса уже не будет, а возможно даже, что не будет практически никакого эффекта. Наша задача — проверить это утверждение на практике.

   Обратимся к таблице «Перелеты» (`ticket_flights`). В ней имеется столбец «Класс обслуживания» (`fare_conditions`), который отличается от остальных тем, что в нем могут присутствовать лишь три различных значения: `Comfort`, `Business` и `Economy`.

   Если секундомер в утилите psql выключен, то включите его.

   Выполните запросы, подсчитывающие количество строк, в которых атрибут `fare_conditions` принимает одно из трех возможных значений. Каждый из запросов выполните три-четыре раза, поскольку время может немного изменяться, и подсчитайте среднее время. Обратите внимание на число строк, которые возвращает функция `count` для каждого значения атрибута. При этом среднее время выполнения запросов для трех различных значений атрибута `fare_conditions` будет различаться незначительно, поскольку в каждом случае СУБД просматривает все строки таблицы.

```
SELECT count( * )
  FROM ticket_flights
  WHERE fare_conditions = 'Comfort';

SELECT count( * )
  FROM ticket_flights
  WHERE fare_conditions = 'Business';


SELECT count( * )
  FROM ticket_flights
  WHERE fare_conditions = 'Economy';
```

   Создайте индекс по столбцу `fare_conditions`. Конечно, в реальной ситуации такой индекс вряд ли целесообразен, но нам он нужен для экспериментов.

   Проделайте те же эксперименты с таблицей `ticket_flights`. Будет ли различаться среднее время выполнения запросов для различных значений атрибута `fare_conditions`? Почему это имеет место?

   В завершение этого упражнения отметим, что в случае ошибки планировщика при использовании индекса возможно не только отсутствие положительного эффекта, но и значительный отрицательный эффект.

```
demo=# \timing
Timing is on.
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Comfort';
 count
-------
 17291
(1 row)

Time: 167.970 ms
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Comfort';
 count
-------
 17291
(1 row)

Time: 36.252 ms
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Comfort';
 count
-------
 17291
(1 row)

Time: 24.973 ms
```

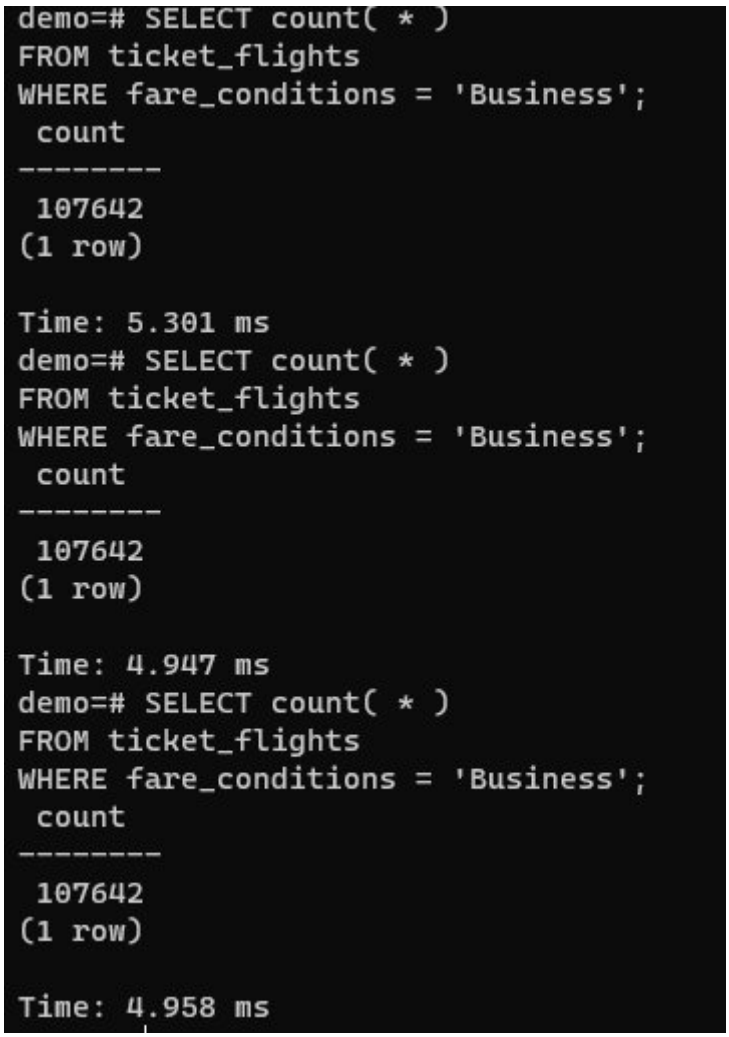
```
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Business';
 count
--------
 107642
(1 row)

Time: 25.398 ms
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Business';
 count
--------
 107642
(1 row)

Time: 30.722 ms
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Business';
 count
--------
 107642
(1 row)

Time: 30.232 ms
```

```
Time: 30.232 ms
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Economy';
 count
--------
 920793
(1 row)

Time: 31.162 ms
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Economy';
 count
--------
 920793
(1 row)

Time: 34.657 ms
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Economy';
 count
--------
 920793
(1 row)

Time: 35.667 ms
```

```
demo=# CREATE INDEX index ON ticket_flights (fare_conditions)
demo-# ;
```

```
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Economy';
 count
--------
 920793
(1 row)

Time: 21.701 ms
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Economy';
 count
--------
 920793
(1 row)

Time: 20.510 ms
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Economy';
 count
--------
 920793
(1 row)

Time: 25.854 ms
```

```
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Comfort';
 count
-------
 17291
(1 row)

Time: 1.278 ms
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Comfort';
 count
-------
 17291
(1 row)

Time: 1.122 ms
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Comfort';
 count
-------
 17291
(1 row)

Time: 1.042 ms
```



Как видно после создания индекса скорость выборки из таблицы увеличивается в разы, это связано из-за того, что поиск по индексам имеет другую логику, чем поиск по значению.