

# Язык SQL, дз – 7

Бурмашев Григорий, БПМИ-208

17 ноября 2022 г.

## Номер 1

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.

Ориентируясь на следующий отрывок из учебника:

Важно, что в уникальных индексах допускается наличие значений NULL, поскольку они считаются не совпадающими ни с какими другими значениями, в том числе и друг с другом. Если уникальный индекс создан по нескольким атрибутам, то совпадающими считаются лишь те комбинации значений атрибутов в двух строках, в которых совпадают значения всех соответствующих атрибутов.

Я полагаю, что операция вставки пройдет успешно, поскольку дело мы имеем с NULL, где по сути `NULL != NULL`. Т.е. два объекта из условия будут сохранять свойство уникальности. Для подтверждения я решил просимулировать условия задачи и посмотреть на результат:

Today - 17.11.2022

▶ SELECT \* FROM testing

22:22:05

▶ INSERT INTO testing (column1) VALUES ('ABC');

22:21:58

▶ CREATE UNIQUE INDEX uniq\_index ON testing (column1, column2);

22:21:32

▶ INSERT INTO testing (column1) VALUES ('ABC');

22:20:57

▶ CREATE TABLE testing (column1 text, column2 text);

22:20:28

Data Output Messages Notifications



	column1 text	column2 text
1	ABC	[null]
2	ABC	[null]

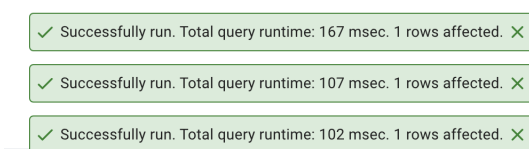
Вставка действительно прошла успешно

## Номер 3

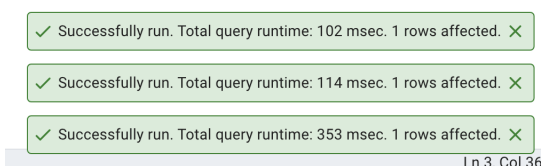
3. Известно, что индекс значительно ускоряет работу, если при выполнении запроса из таблицы отбирается лишь небольшая часть строк. Если же эта доля велика, скажем, половина строк или более, то большого положительного эффекта от наличия индекса уже не будет, а возможно даже, что не будет практически никакого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

Для начала я запустил запросы без индекса и посмотрел на их время выполнения, для решения использовал отображение времени в утилите pgAdmin 4.

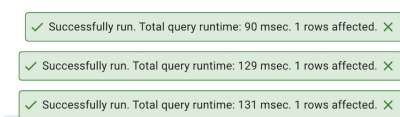
- Запрос с типом **Comfort**:



- Запрос с типом **Business**:



- Запрос с типом **Economy**:



Различия получились действительно незначительные, только 1 из запусков запроса для типа Business обрабатывался 350 мс, могу посчитать это выбросом. После я создал индекс по fare\_conditions, время получилось:

- Запрос с типом **Comfort**:

```

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

Time: 1,996 ms
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Comfort';
count
-----
17291
(1 row)

Time: 1,972 ms
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Comfort';
count
-----
17291
(1 row)

Time: 1,707 ms
demo=#

```

- Запрос с типом **Business**:

```

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

Time: 9,533 ms
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Business';
count
-----
107642
(1 row)

Time: 19,537 ms
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Business';
count
-----
107642
(1 row)

Time: 20,719 ms

```

- Запрос с типом **Economy**:

```

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

Time: 28,785 ms
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Economy';
count
-----
920793
(1 row)

Time: 26,797 ms
demo=# SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Economy';
count
-----
920793
(1 row)

Time: 22,873 ms

```

Очевидный результат, запросы стало выполняться намного быстрее. Для класса **Economy** время выполнения запросов немного выше, я думаю это связано с бОльшим количеством строк с этим классом, поскольку селективность меньше, а индексы наиболее хорошо работают при высокой селективности.