

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

Лабораторная работа №7
по курсу «Проектирование баз данных»

Выполнила: Прудникова А. А.
Группа: М8О-114СВ-24
Преподаватель: Моргунов Е. П.

Москва, 2024

Задание 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.

Запрос

```
CREATE TABLE new_table (  
    column1 VARCHAR(50),  
    column2 VARCHAR(50),  
    UNIQUE (column1, column2)  
);  
  
INSERT INTO new_table (column1, column2) VALUES ('ABC', NULL);  
  
INSERT INTO new_table (column1, column2) VALUES ('ABC', NULL);  
  
SELECT * FROM new_table;
```

Результат

	column1	column2
1	ABC	<null>
2	ABC	<null>

Решение

Операция успешна, так как PostgreSQL допускает наличие нескольких строк с NULL в уникальных индексах.

Задание 3

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.

Создайте индекс по столбцу `fare_conditions`. Конечно, в реальной ситуации такой индекс вряд ли целесообразен, но нам он нужен для экспериментов. Прodelайте те же эксперименты с таблицей `ticket_flights`. Будет ли различаться среднее время выполнения запросов для различных значений атрибута `fare_conditions`? Почему это имеет место?

Запрос

```
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';
```

```
CREATE INDEX idx_fare_conditions ON  
ticket_flights(fare_conditions);
```

```
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';
```

Результат

Без индексации

```
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Comfort'  
[2024-10-22 14:34:45] 1 row retrieved starting from 1 in 380 ms (execution: 323 ms, fetching: 57 ms)  
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Business'  
[2024-10-22 14:34:45] 1 row retrieved starting from 1 in 125 ms (execution: 80 ms, fetching: 45 ms)  
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Economy'  
[2024-10-22 14:34:46] 1 row retrieved starting from 1 in 142 ms (execution: 96 ms, fetching: 46 ms)  
  
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Comfort'  
[2024-10-22 14:39:06] 1 row retrieved starting from 1 in 112 ms (execution: 95 ms, fetching: 17 ms)  
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Business'  
[2024-10-22 14:39:06] 1 row retrieved starting from 1 in 100 ms (execution: 81 ms, fetching: 19 ms)  
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Economy'  
[2024-10-22 14:39:06] 1 row retrieved starting from 1 in 112 ms (execution: 93 ms, fetching: 19 ms)
```

```

demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Comfort'
[2024-10-22 14:39:34] 1 row retrieved starting from 1 in 98 ms (execution: 81 ms, fetching: 17 ms)
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Business'
[2024-10-22 14:39:35] 1 row retrieved starting from 1 in 100 ms (execution: 77 ms, fetching: 23 ms)
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Economy'
[2024-10-22 14:39:35] 1 row retrieved starting from 1 in 111 ms (execution: 92 ms, fetching: 19 ms)

demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Comfort'
[2024-10-22 14:39:58] 1 row retrieved starting from 1 in 99 ms (execution: 85 ms, fetching: 14 ms)
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Business'
[2024-10-22 14:39:58] 1 row retrieved starting from 1 in 96 ms (execution: 81 ms, fetching: 15 ms)
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Economy'
[2024-10-22 14:39:58] 1 row retrieved starting from 1 in 114 ms (execution: 95 ms, fetching: 19 ms)

```

С индексацией

```

demo: bookings, public> CREATE INDEX idx_fare_conditions ON ticket_flights(fare_conditions)
[2024-10-22 14:35:23] completed in 503 ms
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Comfort'
[2024-10-22 14:35:43] 1 row retrieved starting from 1 in 34 ms (execution: 6 ms, fetching: 28 ms)
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Business'
[2024-10-22 14:35:43] 1 row retrieved starting from 1 in 57 ms (execution: 11 ms, fetching: 46 ms)
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Economy'
[2024-10-22 14:35:44] 1 row retrieved starting from 1 in 70 ms (execution: 52 ms, fetching: 18 ms)

demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Comfort'
[2024-10-22 14:42:03] 1 row retrieved starting from 1 in 31 ms (execution: 5 ms, fetching: 26 ms)
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Business'
[2024-10-22 14:42:03] 1 row retrieved starting from 1 in 43 ms (execution: 10 ms, fetching: 33 ms)
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Economy'
[2024-10-22 14:42:03] 1 row retrieved starting from 1 in 76 ms (execution: 58 ms, fetching: 18 ms)

demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Comfort'
[2024-10-22 14:42:23] 1 row retrieved starting from 1 in 22 ms (execution: 5 ms, fetching: 17 ms)
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Business'
[2024-10-22 14:42:24] 1 row retrieved starting from 1 in 46 ms (execution: 9 ms, fetching: 37 ms)
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Economy'
[2024-10-22 14:42:24] 1 row retrieved starting from 1 in 73 ms (execution: 48 ms, fetching: 25 ms)

demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Comfort'
[2024-10-22 14:42:42] 1 row retrieved starting from 1 in 27 ms (execution: 4 ms, fetching: 23 ms)
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Business'
[2024-10-22 14:42:42] 1 row retrieved starting from 1 in 34 ms (execution: 10 ms, fetching: 24 ms)
demo: bookings, public> SELECT count(*) FROM ticket_flights WHERE fare_conditions = 'Economy'
[2024-10-22 14:42:42] 1 row retrieved starting from 1 in 80 ms (execution: 47 ms, fetching: 33 ms)

```

Условия/Среднее время выполнения, мс	Без индекса	С индексом
Comfort	172.25	28.5
Business	105.25	45
Economy	119.75	74.75

Решение

Для всех значений индекс существенно уменьшает среднее время выполнения запросов. Это подтверждает, что индекс действительно ускоряет выборку строк.

Отметим также, что индекс приносит больше пользы для значений, которые встречаются реже в таблице, так как наибольшая разница в среднем времени выполнения была достигнута при `fare_conditions = 'Comfort'`, и количество именно таких строк было наименьшим.