Лабораторная работа №7

Тема: Глава 8 из книги. Упр 1, 3

Группа: М8О-109СВ-24

Выполнил: **Гимазетдинов Дмитрий Русланович**

вернуться на главную

Упражнение 1

Дано:

Предположим, что для какой-то таблицы создан уникальный индекс по двум столбцам: column1 и column2. В таблице есть строка, у которой значение ат- рибута 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.

Решение:

Операция будет успешной, так как NULL сравнивая с NULL, не вызывает противоречий.

Создадим временную таблицу, и впихнем в нее значения:

```
CREATE TEMP TABLE wt_1 (
    c1 TEXT,     -- Define column c1 as TEXT
    c2 TEXT      -- Define column c2 as TEXT (or other appropriate type)
);
```

```
insert into wt_1 values('abc', null);
```

Теперь создадим индекс по двум столбцам:

```
create index on wt_1 (c1, c2);
```

Вставим эти же значения и получаем:

```
demo=# insert into wt_1 values('abc', null);
INSERT 0 1
demo=# ^C
demo=# create index on wt_1 (c1, c2);
CREATE INDEX
demo=# insert into wt_1 values('abc', null);
INSERT 0 1
```

Наши оправдания подтвердились!!!

Упражнение 3

Дано:

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 эффек та от наличия индекса уже не будет, а возможно даже, что не будет практически икакого 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.

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

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

```sql
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.

Решение:

Выполним запрос для класса - `Comfort`:

Tests	Timings ms
t1	422,164
t2	87,304
t3	83,189
avg	204.219

Выполним запрос для класса - `Business`:

Tests	Timings ms
t1	72,678
t2	79,267
t3	72,613
avg	74.853

Выполним запрос для класса - `Economy`:

Tests	Timings ms
t1	88,011
t2	95,559
t3	92,672
avg	92.080

Добавим индексы по столбцу `fare_conditions`:

```
create index on ticket_flights (fare_conditions);
```

И будем еще перед запуском смотреть план выполнения запроса с помощью `EXPLAIN`:

План:

```
demo=#
explain SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Comfort';
                                                        QUERY PLAN
---------------------------------------------------------------------------------
-----------------------------------------------
 Aggregate  (cost=879.01..879.02 rows=1 width=8)
   ->  Index Only Scan using ticket_flights_fare_conditions_idx on
ticket_flights  (cost=0.43..785.19 rows=37529 width=0)
         Index Cond: (fare_conditions = 'Comfort'::text)
(3 rows)
```

Выполним запрос для класса - `Comfort`:

Tests	Timings ms
t1	6,623
t2	3,724
t3	6,771
avg	5.706

План:

```
demo=#
explain SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Business';
                                                                QUERY
PLAN
---------------------------------------------------------------------------------
-------------------------------------------------------------------
 Finalize Aggregate  (cost=4881.23..4881.24 rows=1 width=8)
   ->  Gather  (cost=4881.02..4881.23 rows=2 width=8)
         Workers Planned: 2
         ->  Partial Aggregate  (cost=3881.02..3881.03 rows=1 width=8)
               ->  Parallel Index Only Scan using
ticket_flights_fare_conditions_idx on ticket_flights  (cost=0.43..3630.15
rows=100347 width=0)
                     Index Cond: (fare_conditions = 'Business'::text)
(6 rows)
```

Выполним запрос для класса - `Business`:

Tests	Timings ms

Tests	Timings ms
t1	16,756
t2	17,261
t3	7,856
avg	13.957

План:

```
demo=#
explain SELECT count( * )
FROM ticket_flights
WHERE fare_conditions = 'Economy';
                                                                QUERY
PLAN
------------------------------------------------------------------------------
------------------------------------------------------------------
 Finalize Aggregate  (cost=34527.05..34527.06 rows=1 width=8)
   ->  Gather  (cost=34526.84..34527.05 rows=2 width=8)
         Workers Planned: 2
         ->  Partial Aggregate  (cost=33526.84..33526.85 rows=1 width=8)
               ->  Parallel Index Only Scan using
ticket_flights_fare_conditions_idx on ticket_flights  (cost=0.43..31358.12
rows=867489 width=0)
                     Index Cond: (fare_conditions = 'Economy'::text)
(6 rows)
```

Выполним запрос для класса - `Economy`:

Tests	Timings ms
t1	53,176
t2	63,178
t3	51,159
avg	55.838

Подведем итоги, составим общую таблицу, в которой покажем кол-во строчек выборки, без префикса `i` - результаты без индекса, и с ним для результатов с индексом.

Class	count(*)	best t	avg t	best it	avg it
Comfort	39154	83,189	204.219	3,724	5.706
Business	242204	72,613	74.853	7,856	13.957
Economy	2078977	88,011	92.080	51,159	55.838