

# MySQL でのパフォーマンス測定

---

- 普段パフォーマンスを気にしながらインデックスを貼ったりクエリを投げているわけですが、無頓着だとどれくらい差がでるのか気になったので計測してみます
- 実務のデータ数でこれまで体感できなかった為、実際に差が出るのか？を試してみたかったというのが経緯です

## 計測

---

### 計測方法

- 単純に **Query OK, xxx rows affected (x min x.xx sec)** の時間を見えています
- ただ、時間が短くシビアなものについては **performance\_schema** database を確認しています

## INSERT

### テーブル

#### index なし

```
CREATE TABLE item
(
  id    INT PRIMARY KEY AUTO_INCREMENT,
  num   INT UNSIGNED,
  str   VARCHAR(30),
  str2  VARCHAR(30),
  str3  VARCHAR(30)
);
```

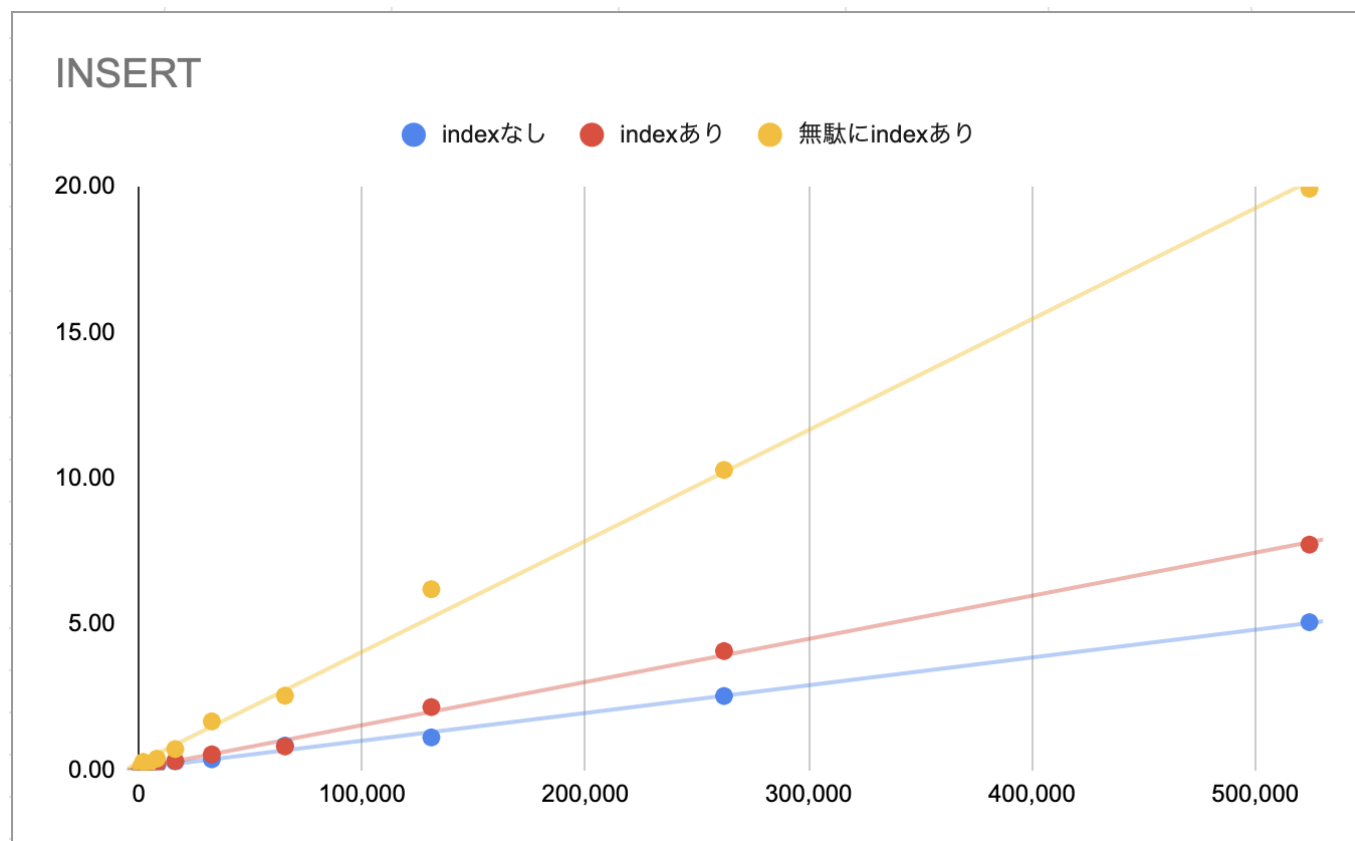
#### index あり

```
CREATE TABLE item_index
(
  id    INT PRIMARY KEY AUTO_INCREMENT,
  num   INT UNSIGNED,
  str   VARCHAR(30),
  str2  VARCHAR(30),
  str3  VARCHAR(30),
  index (num),
  index (str)
);
```

#### 無駄に index あり

```
CREATE TABLE item_index_2
(
  id    INT PRIMARY KEY AUTO_INCREMENT,
  num   INT UNSIGNED,
  str   VARCHAR(30),
  str2  VARCHAR(30),
  str3  VARCHAR(30),
  index (num),
  index (str),
  index (str2),
  index (str3),
  index (str, str2),
  index (str, str3),
  index (str2, str3),
  index (str, str2, str3),
  index (str, str3, str2),
  index (str2, str, str3),
  index (str2, str3, str),
  index (str3, str2, str)
);
```

結果



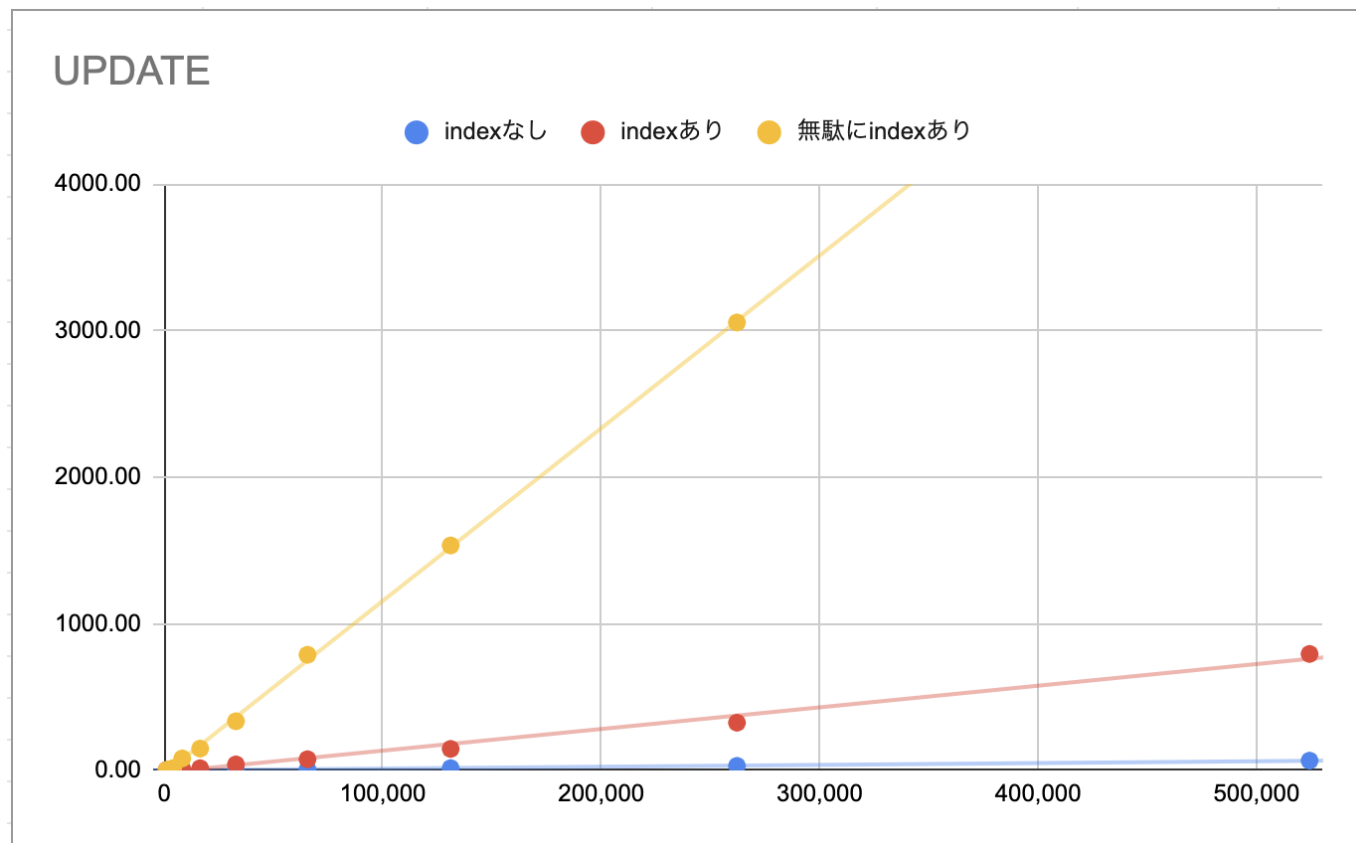
	indexなし	indexあり	無駄にindexあり
1,024	0.02	0.05	0.12
2,048	0.04	0.09	0.28
4,096	0.08	0.10	0.17
8,192	0.15	0.23	0.39
16,384	0.29	0.31	0.72
32,768	0.37	0.54	1.67
65,536	0.84	0.81	2.55
131,072	1.12	2.16	6.20
262,144	2.54	4.08	10.29
524,288	5.07	7.73	19.93

index なし	
傾き	0.00000967077
	1.00
indexあり	
傾き	0.00001496247
	1.55
無駄にindexあり	
傾き	0.00003873739
	4.01
※切片 0 の回帰直線の傾き	

## UPDATE

- 同じテーブルに対して UPDATE かけました。
- テストデータの id が綺麗じゃなかったので LIMIT を使って UPDATE 件数を指定しています(通常よりパフォーマンスは落ちていると思われます)

結果



	indexなし	indexあり	無駄にindexあり
1,024	0.30	2.59	3.53
2,048	0.66	2.82	6.19
4,096	0.49	3.96	15.87
8,192	1.48	9.43	81.37
16,384	2.80	14.56	148.56
32,768	4.55	40.74	335.07
65,536	7.20	76.17	788.84
131,072	15.34	146.52	1534.83
262,144	30.79	325.38	3056.28
524,288	66.52	794.48	(省略)

index なし	
傾き	0.00012452899
	1.00
indexあり	
傾き	0.00143983528
	11.56
無駄にindexあり	
傾き	0.01165850322
	93.62
※切片 0 の回帰直線の傾き	

絞り込むタイミングが早い時と遅い時

1条件目であまり絞り込めていない場合

```
mysql> SELECT EVENT_ID,
->          TRUNCATE(TIMER_WAIT / 1000000000000, 6) AS Duration,
->          SQL_TEXT
-> FROM performance_schema.events_statements_history_long
-> WHERE SQL_TEXT = "select * from item where str like '%2%' and str2
like '%525%'";
```

```
+-----+-----+-----+
| EVENT_ID | Duration | SQL_TEXT |
+-----+-----+-----+
| 532 | 0.758868 | select * from item where str like '%2%' and str2 |
like '%525%' |
| 627 | 0.734291 | select * from item where str like '%2%' and str2 |
like '%525%' |
| 646 | 0.731736 | select * from item where str like '%2%' and str2 |
like '%525%' |
| 665 | 0.784295 | select * from item where str like '%2%' and str2 |
like '%525%' |
| 684 | 0.699200 | select * from item where str like '%2%' and str2 |
like '%525%' |
| 817 | 0.726403 | select * from item where str like '%2%' and str2 |
like '%525%' |
| 836 | 0.665074 | select * from item where str like '%2%' and str2
```

```

like '%525%' |
|      855 | 0.669399 | select * from item where str like '%2%' and str2
like '%525%' |
|      874 | 0.718087 | select * from item where str like '%2%' and str2
like '%525%' |
|      893 | 0.731762 | select * from item where str like '%2%' and str2
like '%525%' |
+-----+-----+-----+
-----+
10 rows in set (0.00 sec)

```

→ 平均 0.7219115s

1条件目で絞り込めている場合(上記と where の順番を逆にした)

```

mysql> SELECT EVENT_ID,
->         TRUNCATE(TIMER_WAIT / 1000000000000, 6) AS Duration,
->         SQL_TEXT
-> FROM performance_schema.events_statements_history_long
-> WHERE SQL_TEXT = "select * from item where str2 like '%525%' and
str like '%2%'";
+-----+-----+-----+
-----+
| EVENT_ID | Duration | SQL_TEXT
|
+-----+-----+-----+
-----+
|      551 | 0.610941 | select * from item where str2 like '%525%' and str
like '%2%' |
|      570 | 0.568627 | select * from item where str2 like '%525%' and str
like '%2%' |
|      589 | 0.619817 | select * from item where str2 like '%525%' and str
like '%2%' |
|      608 | 0.630084 | select * from item where str2 like '%525%' and str
like '%2%' |
|      703 | 0.627935 | select * from item where str2 like '%525%' and str
like '%2%' |
|      722 | 0.613024 | select * from item where str2 like '%525%' and str
like '%2%' |
|      741 | 0.563686 | select * from item where str2 like '%525%' and str
like '%2%' |
|      760 | 0.567389 | select * from item where str2 like '%525%' and str
like '%2%' |
|      779 | 0.595388 | select * from item where str2 like '%525%' and str
like '%2%' |
|      798 | 0.675684 | select * from item where str2 like '%525%' and str
like '%2%' |
+-----+-----+-----+
-----+
10 rows in set (0.01 sec)

```

→ 平均 0.6072575s

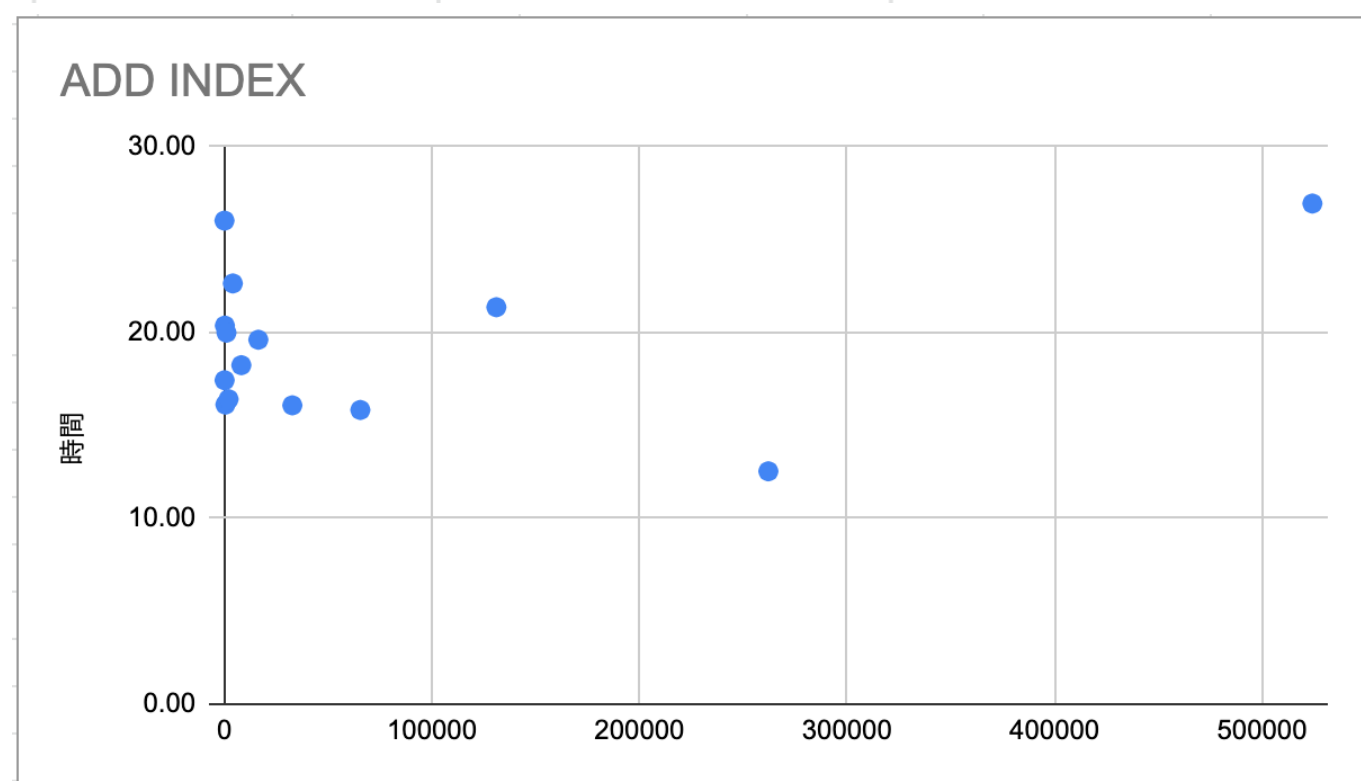
→このケースでは 15% 程度パフォーマンスが改善

## 失敗しました (学び)

後からインデックスを貼ったとき の時間を計測したかったわけだが、100万レコードから DELETE でレコードを減らしながら計測していったが、なぜか結果が以下のように。



レコード数	時間
64	26.02
128	17.43
256	20.37
512	16.12
1,024	19.99
2,048	16.41
4,096	22.64
8,192	18.24
16,384	19.61
32,768	16.08
65,536	15.83
131,072	21.36
262,144	12.53
524,288	26.94



レコード数			
-------	--	--	--

比例しないものなんですかね？  
不審に思い試しに6レコードしかない状態ですが `SELECT * FROM item;` をやってみると...

```
mysql> select * from item;
+-----+-----+-----+-----+
| id      | num      | str                                     | str2 |
| str3    |          |          |          |
+-----+-----+-----+-----+
| 1310688 | 98135764 | 6bff0ca187f6a99c99daf21799282a |      |
4187a80d597ac9855519b8c1d90c2e | b40a8a182d9b7d67e0e3062ead701d |
| 1310689 | 10926723 | 01b18c93949530fe2e7d870f8aca24 |      |
f80b962ec99c28c1d7a37c5af31786 | 4f80cc563c4e6ca7842692629b744f |
| 1310690 | 98869803 | 145d8de154d23367713cbdca066f7f |      |
b77d25c46f0ad94b398c16614e4f75 | a1b28b4e59ffea131727eec6834281 |
| 1310691 | 27278484 | 80ac1361cbe1abd80bb31cdbf5e525 |      |
2ac1774a89828415501144a148c088 | e1ef45687fa3c80eed91e9363612a1 |
| 1310692 | 48066717 | d86bd1608477460813a76884f0456a |      |
f6940106536b08dd3bf035ed769a4e | 507d17133d4913b2cc5b8f604bc9d9 |
| 1310693 | 77327447 | a130de047f9183a922cc3bc3f9fc26 |      |
0873044106d59a9b9fa89d712b0e09 | 4eb3b32e3845385b3054d8e92e4c4b |
+-----+-----+-----+-----+
6 rows in set (32.40 sec)
```

```
__人人人人人人人人人人人人人人__
> 6 rows in set (32.40 sec) <
-YY^YY^YY^YY^YY^YY^YY^YY^YY^YY^Y-
```

物理削除をし続けるとパフォーマンスが悪くなる

MySQL の InnoDB では、フラグメンテーション(断片化) が起こるとのこと。言われてみると、そうなるのが自然な気がしますね...

断片化のせいでパフォーマンスが極端に悪化していたものと推測。

`ALTER TABLE テーブル名 ENGINE INNODB;` で最適化できるとのことなのでやってみると、

```
mysql> select * from item;
+-----+-----+-----+-----+
| id      | num      | str                                     | str2 |
| str3    |          |          |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

```
| 1310688 | 98135764 | 6bff0ca187f6a99c99daf21799282a |  
4187a80d597ac9855519b8c1d90c2e | b40a8a182d9b7d67e0e3062ead701d |  
| 1310689 | 10926723 | 01b18c93949530fe2e7d870f8aca24 |  
f80b962ec99c28c1d7a37c5af31786 | 4f80cc563c4e6ca7842692629b744f |  
| 1310690 | 98869803 | 145d8de154d23367713cbdca066f7f |  
b77d25c46f0ad94b398c16614e4f75 | a1b28b4e59ffea131727eec6834281 |  
| 1310691 | 27278484 | 80ac1361cbe1abd80bb31cdbf5e525 |  
2ac1774a89828415501144a148c088 | e1ef45687fa3c80eed91e9363612a1 |  
| 1310692 | 48066717 | d86bd1608477460813a76884f0456a |  
f6940106536b08dd3bf035ed769a4e | 507d17133d4913b2cc5b8f604bc9d9 |  
| 1310693 | 77327447 | a130de047f9183a922cc3bc3f9fc26 |  
0873044106d59a9b9fa89d712b0e09 | 4eb3b32e3845385b3054d8e92e4c4b |  
+-----+-----+-----+-----+-----+  
-----+-----+  
6 rows in set (0.00 sec)
```

無事復活しました。

パーティションとか、他にも色々やりたかったのですが今回はやりきれませんでした。

また機会があればやろうとおもいます。