

# DOs and DONTs for Indexes [closed]

Asked 8 years, 9 months ago   Active 5 months ago   Viewed 5k times



13



13



It's difficult to tell what is being asked here. This question is ambiguous, vague, incomplete, overly broad, or rhetorical and cannot be reasonably answered in its current form. For help clarifying this question so that it can be reopened, [visit the help center](#).

Closed 8 years ago.

What are some DOs and DONTs for improving database performance using index?

A DO would be a case in which an index should be created, or another indexes related tip that will improve performance.

A DONT will be a case when an index shouldn't be created, or another index related action that can hurt the performance.

mysql

database

performance

database-design

indexing

edited May 23 '11 at 14:40

asked May 23 '11 at 14:29



[Click Upvote](#)

213k ● 239 ● 538 ● 712

- 2 the question is waaaaay too broad. – [Spudley](#) May 23 '11 at 14:32
- 1 From the close reason: "overly broad". It is a question that invites vast, most likely subjective lists, those are generally frowned upon. – [Orbling](#) May 23 '11 at 14:34
- 2 @Orbling I don't see how, I'm asking what are some DOs for indexes that would improve database performance and DONTs that will be harmful, I don't see how there can be any subjective answers for that. I will edit the question and clarify it. – [Click Upvote](#) May 23 '11 at 14:37
- 3 Personally, I think it's a good candidate for a wiki question, and that it should be re-opened. – [Denis de Bernardy](#) May 23 '11 at 14:39
- 2 @Orbling I'm sure there are many general tips which work/don't work in 80% of the situations, and I'd like to know them. – [Click Upvote](#) May 23 '11 at 14:41

## 2 Answers



Generally speaking:

31

**1. Don't add an index unless you actually need it.**

Join Stack Overflow to learn, share knowledge, and build your career.

[Sign up](#)





```
-- index on foo (bar)
select bar from foo where bar = :bar;
```

By the same token it'll be used in foreign key references (on both tables).

```
-- index on foo (bar) if baz (bar) is frequently updated/deleted.
create table foo (bar references baz (bar));
```

### 3. An index will be used for sorting, especially when tied to a limit:

```
-- index on foo (bar)
select bar from foo order by bar limit 10;
```

### 4. Multicolumn indexes are occasionally useful when 2. and 3. both apply.

In this case put the where conditions first, and the sort key last:

```
-- index on foo (baz, bar)
select bar from foo where baz between :baz1 and :baz2 group by bar;
```

### 5. Keep your table statistics up to date.

If the table stats are garbage, there is little chances that the optimizer will use your indexes. Manually vacuum/analyze your database if needed.

### 6. Index usage depends on your table repartition.

Past a certain threshold of rows retrieved, it'll be faster to do a full table scan. If your index is on a boolean field that more or less splits your table in two, it'll never be used.

Likewise, if your data is stored in such a way that the index scan will likely end up randomly accessing nearly ever applicable disk page for that table, the planner will prefer a full table scan.

### 7. Consider partial/expression indexes when available.

If you've a field that has the same value except for 10% of your rows, consider a partial index on it (i.e. where not that value). This results in a much smaller index without hindering its actual usefulness.

If you're constantly querying against an expression applied to your column and you platform offers expression indexes, consider adding an index on it. When used, the expression won't get evaluated for each row.

edited Sep 25 '19 at 9:45



[simonalexander2005](#)  
3,309 ● 4 ● 35 ● 70

answered May 23 '11 at 14:38



[Denis de Bernardy](#)  
62.1k ● 10 ● 100 ● 126



10



1. Measure system throughput as a whole. An index may help some queries but harm insert, update and delete.
2. Create the index.
3. Measure system throughput as a whole.



If performance is better, leave the index there. If performance is worse, delete it.

answered May 23 '11 at 14:30



[S.Lott](#)

338k ● 72 ● 464 ● 739