

数据库索引设计三步曲

设计索引前的必要准备

一个索引设计范例

实际性能测试

设计索引前的必要准备

决定设计数据库索引前，务必先对业务上涉及到的所有查询进行整理并优化，并针对整理结果对表建立索引与复合索引。建立索引的原则是：合理利用复合索引的**最左前缀规则**，在保证尽可能多的覆盖查询筛选条件的情况下，建立尽可能少的索引。

在业务层面就应当避免可能遍历整个表的查询，将最大可能的返回结果控制在可控的数量内。例如：让所有的筛选都必须先选定一个特定的时间，并根据实际情况对这个时间的最大跨度做限制，然后只在这个时间段内筛选数据，这使得数据库总遍历条数和查询时间变得可控。

一个索引设计范例

```
1 select person_role_id from movie
2 where movie_id=100 and role_id=1
3 order by nr_role desc;
```

SQL | 复制代码

对于这句SQL，创建索引的基本思考步骤如下：

- 第一步：评估参与运算的结果集范围，创建 movie_id 和 role_id 的组合索引；
- 第二步：考虑参与排序的字段，创建nr_role索引；

大部分情况下，经过前两步，已经完成索引的创建。有时候，还需要考虑第三步，添加覆盖索引，在索引中添加需要查找的字段，无需回表，以期达到优化目的：

- 第三步：为避免回表查找，添加 person_role_id 覆盖索引。

经过完整的三步，创建出来的组合索引为：index(movie_id, role_id, nr_role, person_role_id)

实际性能测试

取我们的 order 表作为测试用例（大约200W行数据），测试如下查询语句：

SQL | 复制代码

```
1 SELECT `phone` FROM `order` WHERE `type`=3 AND `status`=6 ORDER BY freight;
```

分别建立3中索引，对应的执行过程与执行耗时如下：

Plain Text | 复制代码

1	索引	type status
2	执行计划	SIMPLE order ref type type 2 const,const 38110 Using where; Using filesort
3	耗时	2380 ms
4	说明	只使用索引缩小了结果范围，因为排序字段没有包含在索引中，所以使用了 filesort

Plain Text | 复制代码

1	索引	type status freight
2	执行计划	SIMPLE order ref type type 2 const,const 37846 Using where
3	耗时	32 ms
4	说明	使用索引缩小了结果范围集，并使用索引进行了排序，但因为 select 字段 phone 不在索引中，需要回表读取

Plain Text | 复制代码

1	索引	type status freight phone
2	执行计划	SIMPLE order ref TYPE TYPE 2 const,const 37804 USING WHERE; USING index
3	耗时	3 ms
4	说明	使用索引缩小了结果范围集，并使用索引进行了排序，同时 select 字段 phone 也在索引中被覆盖，可以直接输出

结论：合理设计使用索引，可以将查询效率优化百倍以上。

更详细的索引创建和优化，请参看：

此处为语雀内容卡片，点击链接查看：https://senguo.yuque.com/dev-team/knowledge/blsx2e?view=doc_embed&inner=8hYq2