<https://mp.weixin.qq.com/s/lTk-kQD7_BMnFCRCNIk3Fw>

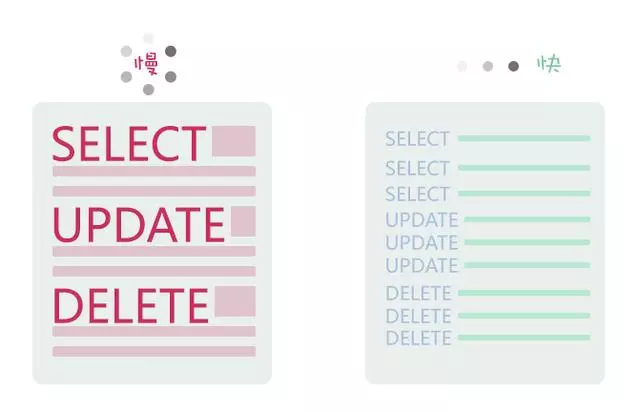
# MySQL 常见优化小技巧

**技巧1  比较运算符能用 “=”就不用“<>”**  
“=”增加了索引的使用几率。

**技巧2  明知只有一条查询结果，那请使用 “LIMIT 1”**  
“LIMIT 1”可以避免全表扫描，找到对应结果就不会再继续扫描了。

**技巧3  为列选择合适的数据类型**  
能用TINYINT就不用SMALLINT，能用SMALLINT就不用INT，道理你懂的，磁盘和内存消耗越小越好嘛。

**技巧4  将大的DELETE，UPDATE or INSERT 查询变成多个小查询**  
能写一个几十行、几百行的SQL语句是不是显得逼格很高？然而，为了达到更好的性能以及更好的数据控制，你可以将他们变成多个小查询。



**技巧5  使用UNION ALL 代替 UNION，如果结果集允许重复的话**

# 因为 UNION ALL 不去重，效率高于 UNION。

**技巧6  为获得相同结果集的多次执行，请保持SQL语句前后一致**  
这样做的目的是为了充分利用查询缓冲。  
比如根据地域和产品id查询产品价格，第一次使用了：



那么第二次同样的查询，请保持以上语句的一致性，比如不要将where语句里面的id和region位置调换顺序。

**技巧7  尽量避免使用 “SELECT \*”**  
如果不查询表中所有的列，尽量避免使用 SELECT \*，因为它会进行全表扫描，不能有效利用索引，增大了数据库服务器的负担，以及它与应用程序客户端之间的网络IO开销。



**技巧8  WHERE 子句里面的列尽量被索引**  
只是“尽量”哦，并不是说所有的列。因地制宜，根据实际情况进行调整，因为有时索引太多也会降低性能。

**技巧9  JOIN 子句里面的列尽量被索引**  
同样只是“尽量”哦，并不是说所有的列。

**技巧10  ORDER BY 的列尽量被索引**  
ORDER BY的列如果被索引，性能也会更好。

**技巧11  使用 LIMIT 实现分页逻辑**  
不仅提高了性能，同时减少了不必要的数据库和应用间的网络传输。

**技巧12  使用 EXPLAIN 关键字去查看执行计划**  
EXPLAIN 可以检查索引使用情况以及扫描的行。  
**其他**  
SQL调优方法有很多种，同样的查询结果可以有很多种不同的查询方式。其实最好的方法就是在开发环境中用最贴近真实的数据集和硬件环境进行测试，然后再发布到生产环境中。