# MySQL的 where 1=1会不会影响性能？看完官方文档就悟了！

你好，我是猿java。

在日常业务开发中，会通过使用`where 1=1`来简化动态 SQL语句的拼接，有人说`where 1=1`会影响性能，也有人说不会，到底会不会影响性能？本文将从MySQL的官方资料来进行分析。

## 动态拼接 SQL的方法

在 Mybatis中，动态拼接 SQL最常用的两种方式：使用 where 1=1 和 使用`<where>`标签。

### 使用where 1=1

使用过 iBATIS的小伙伴应该都知道：在 iBATIS中没有`<where>`标签，动态SQL的处理相对较为原始和复杂，因此使用`where 1=1`这种写法的用户很大一部分是还在使用 iBATIS 或者是从 iBATIS过度到 Mybatis。

如下示例，通过`where 1=1`来动态拼接有效的 if语句：

```
<select id="" parameterType = "">
    SELECT * FROM user
    WHERE 1=1
    <if test="name != null and name != "">
        AND name = #{name}
    </if>
    <if test="age != null ">
        AND age = #{age }
    </if>
</select>
```

使用`<where>`标签
---

Mybatis提供了`<where>`标签，`<where>`标签只有在至少一个 if条件有值的情况下才去生成 where子句，若 AND或 OR前没有有效语句，where元素会将它们去除，也就是说，如果 Mybatis通过`<where>`标签动态生成的语句为`where AND name = '111'`，最终会被优化为`where name = '111'`。

`<where>`标签使用示例如下：

```
<select id="" parameterType = "">
    SELECT * FROM user
    <where>
        <if test="name != null and name != "">
            AND name = #{name}
        </if>
        <if test="age != null">
            AND age = #{age}
        </if>
    </where>
</select>
```

`<where>`标签是在 MyBatis中引入的，所以，很多一开始就使用 MyBatis的用户对这个标签使用的比较多。

性能影响
====

`where 1=1`到底会不会影响性能？我们可以先看一个具体的例子：

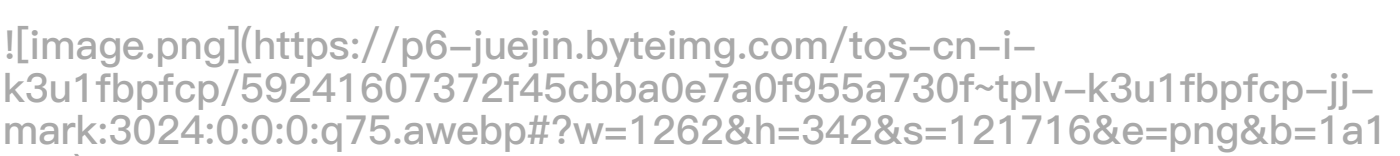
> 说明：示例基于 MySQL 8.0.30

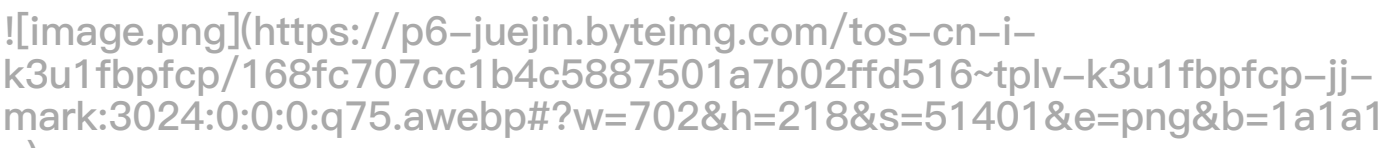
可以使用如下指令查看 MySQL版本：

```
SELECT VERSION();
```


场景：基于一张拥有 100多万条数据的user表，根据name进行查询，

查看表结构和表的总数据，如下图：




下面，通过执行两条 SQL查询语句（一条带有 1=1）：

```
select * from user where name = 'name-96d1b3ce-1a24-4d47-b686-6f9c6940f5f6';
select * from user where 1=1 and name = 'name-f692472e-40de-4053-9498-54b9800e9fb1';
```

对比两条 SQL执行的结果，可以发现它们消耗的时间几乎相同，因此，看起来`where 1=1`对整体的性能似乎并不影响。

为了排除一次查询不具有代表性，我们分别对两条 SQL语句查询 100遍，然后计算平均值：

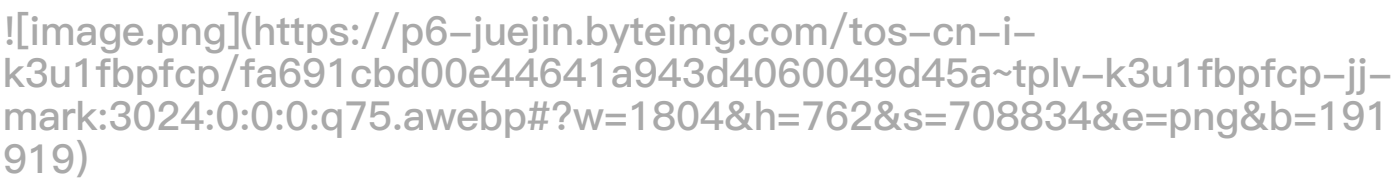
```
SET PROFILING = 1;
DO SLEEP(0.001); -- 确保每次查询之间有足够时间间隔

SET @count = 0;
WHILE @count < 100 DO
select * from user where name = 'name-96d1b3ce-1a24-4d47-b686-6f9c6940f5f6';
-- or
select * from user where 1=1 and name = 'name-f692472e-40de-4053-9498-54b9800e9fb1';
SET @count = @count + 1;
END WHILE;

SHOW PROFILES;
```

两条 SQL分别执行 100次后，最终也发现它们的平均值几乎相同，因此，上述示例似乎证明了 `where 1=1` 对整体的性能并没有不影响。

为什么没有影响？是不是 MySQL对 1=1进行了优化？

为了证明猜想，我们借助`show warnings`命令来查看信息，在 MySQL中，`show warnings`命令用于显示最近执行的 SQL语句产生的警告、错误或通知信息。它可以帮助我们了解语句执行过程中的问题。如下示例：

```
explain select * from user where 1=1 and name = 'name-f692472e-40de-4053-9498-54b9800e9fb1';
show warnings;
```




将上述示例的 warnings信息摘出来如下：

```
/* select#1 */ select `yuanjava`.`user`.`id` AS `id`,
       `yuanjava`.`user`.`name` AS `name`,
       `yuanjava`.`user`.`age` AS `age`,
       `yuanjava`.`user`.`sex` AS `sex`,
       `yuanjava`.`user`.`created_at` AS `created_at`
from `yuanjava`.`user`
where (`yuanjava`.`user`.`name` = 'name-f692472e-40de-4053-9498-
54b9800e9fb1')
```

从 warnings信息可以看出：`1=1`已经被查询优化器优化掉，因此，对整体的性能影响并不大。

那么，有没有 MySQL的官方资料可以佐证 `where 1=1`确实被优化了？

答案：有！MySQL有一种 Constant-Folding Optimization（常量折叠优化）的功能。

Constant-Folding Optimization
==========

MySQL的优化器具有一项称为 Constant-Folding Optimization（常量折叠优化）的功能，可以从查询中消除重言式表达式。Constant-Folding Optimization 是一种编译器的优化技术，用于优化编译时计算表达式的常量部分，从而减少运行时的计算量，换句话说：Constant-Folding Optimization 是发生在编译期，而不是引擎执行期间。

对于上述表达的"重言式表达式"又是什么呢？

重言式

---

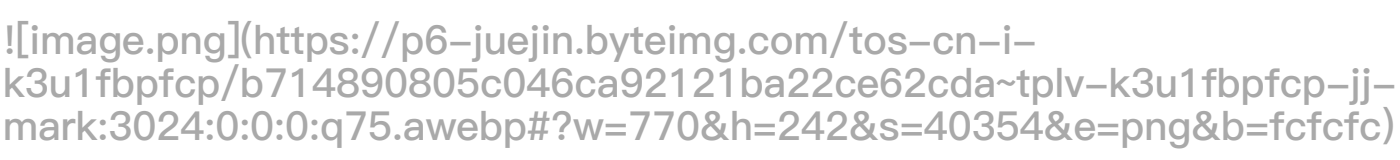
重言式（Tautology ）又称为永真式，它的汉语拼音为：[Chóng yán shì]，是逻辑学的名词。命题公式中有一类重言式，如果一个公式，对于它的任一解释下其真值都为真，就称为重言式（永真式）。

其实，重言式在计算机领域也具有重要应用，比如”重言式表达式”（Tautological expression），它指的是那些总是为真的表达式或逻辑条件。

在 SQL查询中，重言式表达式是指无论在什么情况下，结果永远为真，它们通常会被优化器识别并优化掉，以提高查询效率。例如，如果 where中包含 1=1 或 A=A 这种重言式表达式，它们就会被优化器移除，因为对查询结果没有实际影响。如下两个示例：

```
SELECT * from user where 1=1 and name = 'xxx';
-- 被优化成
SELECT * from user where name = 'xxx';

SELECT id, name, salary * (1 + 0.05 * 2) AS real_salary FROM employees;
-- 优化成(1 + 0.05 * 2 被优化成 1.1)
SELECT id, name, salary * 1.1 AS real_salary FROM employees;
```

另外，通过下面 MySQL架构示意图可以看出：优化器是属于 MySQL的Server层，因此，Constant-Folding Optimization功能支持受 MySQL Server的版本影响。




查阅了 MySQL的官方资料，Constant-Folding Optimization 从MySQL5.7版本开始引入，至于 MySQL5.7以前的版本是否具备这个功能，还有待考证。

## 如何选择?

`where 1=1` 和 `<where> 标签` 两种方案，该如何选择?

* 如果 MySQL Server版本大于等于 5.7，两个随便选，或者根据团队的要求来选；
* 如果 MySQL Server版本小于 5.7，假如使用的是 MyBatis，建议使用`<where> 标签`，如果使用的还是比较老的 iBATIS，只能使用`where 1=1`；
* 如果 MySQL Server版本小于 5.7，建议升升级

> 信息补充：2009年5月，iBATIS从 2.0版本开始更名为 MyBatis， 标签最早出现在MyBatis 3.2.0版本中

总结
==

`where 1=1`和`<where> 标签`到底会不会影响性能，这个问题在网上已经出现了很多次，今天还是想从官方文档来进行说明。本文通过 MySQL的官方资料，加上百万数据的表进行真实测试，得出下面的结论：

* 如果 MySQL Server版本大于等于 5.7，两个随便选，或者根据团队的要求来选；
* 如果 MySQL Server版本小于 5.7，假如使用的是 MyBatis，建议使用`<where> 标签`，如果使用的还是比较老的 iBATIS，只能使用`where 1=1`；

最后，遇到问题，建议首先查找官方的一手资料，这样才能帮助自己在一条正确的技术道路上成长!

参考资料
====

[MySQL8.0 Constant-Folding Optimization](http://cxyroad.com/”https://dev.mysql.com/doc/refman/8.0/en/constant-folding-optimization.html”)

[MySQL5.7 WHERE Clause Optimization](http://cxyroad.com/”https://docs.oracle.com/cd/E17952_01/mysql-5.7-en/where-optimization.html”)

[What’s New in MySQL 5.7](http://cxyroad.com/”https://dev.mysql.com/blog-archive/whats-new-in-mysql-5-7-generally-available/”)

学习交流
====

最后，把我的座右铭送给你：`投资自己才是最大的财富`。 如果你觉得本文章对你有帮助，点赞，收藏不迷路，公众号：猿java，持续为你输出更多的硬核文章和面试经。

原文链接: https://juejin.cn/post/7374238289107648551