count(1)、count(*) 与 count(列名) 的区别

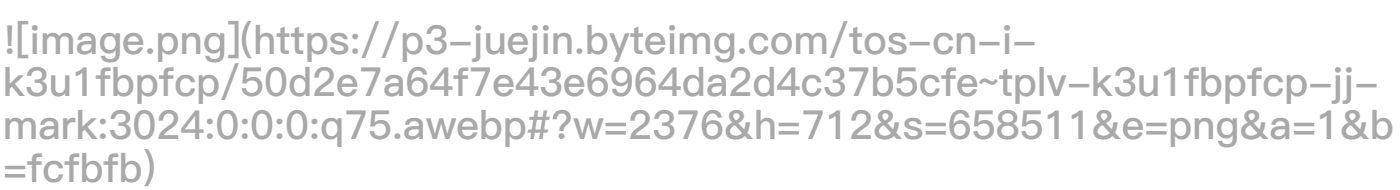
简单来说：

COUNT(1) 和 COUNT(*) 表示的是直接查询符合条件的数据库表的行数。而 COUNT(列名) 表示的是查询符合条件的列的值不为 NULL 的行数。

除了查询得到结果集有区别之外，在性能方面 COUNT(*) 约等于 COUNT(1)，但是 **COUNT(*) 是 SQL92 定义的标准统计行数的语法 **。因为它是标准语法，所以 MySQL 数据库对其进行了很多优化。

COUNT

关于 COUNT 函数，在 MySQL 官网中有详细介绍：



1. COUNT(expr) 返回 SELECT 语句检索的行中 expr 的值不为 NULL 的数量。结果是一个 BIGINT 值。
2. 如果查询结果没有命中任何记录，则返回 0。
3. 但是，值得注意的是，COUNT(*) 的统计结果中会包含值为 NULL 的行数。

即以下表记录:

```
create table #bla(id int,id2 int)
insert #bla values(null,null)
insert #bla values(1,null)
insert #bla values(null,1)
insert #bla values(1,null)
```

```
insert #bla values(null,1)
insert #bla values(1,null)
insert #bla values(null,null)
```

使用语句 count(*),count(id),count(id2)查询结果如下：

```
select count(*),count(id),count(id2)
from #bla
results 7 3 2
```

COUNT(*)的优化

COUNT(*) 在 MySQL 中的优化与所使用的执行引擎密切相关，常见的执行引擎包括 MyISAM 和 InnoDB。

MyISAM 和 InnoDB 之间有许多区别，其中一个关键区别与接下来要讨论的 COUNT(*) 有关：**MyISAM 不支持事务，其锁定级别为表级锁；而 InnoDB 支持事务，并且使用行级锁。**

由于 MyISAM 的表级锁，同一表上的操作需要串行进行。因此，MyISAM 做了一个简单的优化，即单独记录表的总行数。对于不带 WHERE 条件的 COUNT(*) 查询，可以直接返回这个记录的值。

MyISAM 之所以能够记录表中的总行数并供 COUNT(*) 查询使用，是因为其表级锁机制保证了行数查询的准确性，没有并发的行数修改。

然而，对于 InnoDB，这种缓存操作就不可行了，因为 InnoDB 支持事务，其中大部分操作使用行级锁，可能导致表的行数被并发修改，从而使缓存的行数不准确。

尽管如此，InnoDB 也对 COUNT(*) 语句进行了一些优化。

从 MySQL 8.0.13 开始，针对 InnoDB 的 `SELECT COUNT(*) FROM

tbl_name` 查询，在扫表过程中进行了优化，前提是查询语句不包含 WHERE 或 GROUP BY 等条件。

由于 COUNT(*) 只是为了统计总行数，不关心具体值，因此，在扫表过程中选择成本较低的索引可以节省时间。

InnoDB 中的索引分为聚簇索引（主键索引）和非聚簇索引（非主键索引）。非聚簇索引相比聚簇索引更小，因此 MySQL 会优先选择最小的非聚簇索引来扫表。

因此，在建表时，除了主键索引外，创建一个非主键索引也是有必要的。

这些优化的前提是查询语句中不包含 WHERE 和 GROUP BY 条件。

COUNT(*)和 COUNT(1)

MySQL 官方文档对于** COUNT(*) **和 **COUNT(1)** 的性能差异没有做出具体说明。不过，可以从一些实践和理论上推断一些情况。

有些人认为** COUNT(*) **在执行时会转换成** COUNT(1)**，因此 **COUNT(1) **少了转换步骤，所以更快。这个说法在某些情况下可能是正确的，因为** COUNT(*) **会返回表中所有行的数目，而** COUNT(1) **只需要计算行数而不需要检查列值。

另一方面，也有人认为 MySQL 针对 **COUNT(*) **做了特殊优化，因此 **COUNT(*) **更快。这个说法也是有一定道理的，因为** COUNT(*) **是 SQL92 定义的标准语法，MySQL 可能对其进行了一些优化。

综上所述，对于 COUNT(*) 和 COUNT(1) 的性能差异，可能取决于具体的情况和 MySQL 的版本。在实际情况中，可以根据具体的需求和环境选择合适的写法。

> InnoDB handles SELECT COUNT(*) and SELECT COUNT(1) operations in the same way. There is no performance difference.

画重点：**same way, no performance difference**。所以，对于 COUNT(1) 和 COUNT(*)，MySQL 的优化是完全一样的，根本不存在谁比谁快！

那既然** COUNT(*) **和** COUNT(1) **一样，建议用哪个呢?

`建议使用 COUNT(*)！`因为这个是 SQL92 定义的标准统计行数的语法，而且本文只是基于 MySQL 做了分析，关于 Oracle 中的这个问题，也是众说纷纭的呢。

COUNT(字段)

最后，就是我们一直还没提到的 COUNT(字段)，他的查询就比较简单粗暴了，就是进行全表扫描，然后判断指定字段的值是不是为 NULL，不为 NULL 则累加。

相比 **COUNT(*)**，**COUNT(字段) 多了一个步骤就是判断所查询的字段是否为 NULL，所以他的性能要比 COUNT(*) 慢。**

> 如有问题，欢迎搜索【码上遇见你】。

好了，本章节到此告一段落。希望对你有所帮助，祝学习顺利。