

# MYSQL统计信息详解

原创

DBA小强哥

DBA小强

2025年01月14日 16:10

广东

## MYSQL统计信息详解

在 MySQL 中，统计信息是数据库优化器用来生成查询执行计划的重要元数据。它描述了表和索引的基本特性，比如数据分布、索引选择性等。优化器利用这些信息估算查询成本，进而选择最优的查询路径。



DBA小强

文能存储拔电源，武能机房剪网线，进可疯狂删数据，退能后门改权限，加班背锅谁人...>

76篇原创内容

公众号

## 1. 统计信息的主要内容

MySQL 的统计信息主要包括以下几个方面：

### 1.1 表级统计信息

- **\*\*行数 ( `TABLE_ROWS` )\*\***：表中的大致行数。
- **\*\*平均行长度 ( `AVG_ROW_LENGTH` )\*\***：表中每行的平均字节数。
- **\*\*数据大小 ( `DATA_LENGTH` )\*\***：表的总数据大小（字节）。
- **\*\*索引大小 ( `INDEX_LENGTH` )\*\***：表的索引占用的总空间大小（字节）。
- **\*\*空闲空间 ( `DATA_FREE` )\*\***：未使用但分配的空间，可能因删除或更新数据产生。

获取方法：

```
SELECT TABLE_ROWS, AVG_ROW_LENGTH, DATA_LENGTH, INDEX_LENGTH, DATA_FREE
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'your_database' AND TABLE_NAME = 'your_table';
```

### 1.2 索引级统计信息

- **\*\*基数 ( `CARDINALITY` )\*\***：索引中唯一值的估计数量。
- **索引选择性**： `CARDINALITY / TOTAL_ROWS`，表示索引中唯一值的比例。选择性越高，索引越有效。
- **索引类型**：索引的存储方式（BTREE、FULLTEXT、HASH 等）。

获取方法：

```
SHOW INDEX FROM your_table;
```

### 1.3 列级统计信息

- **最小值和最大值**：列中的最小和最大值。
- **数据分布**：用于范围查询的估算，帮助优化器选择索引或全表扫描。
- **\*\*空值计数 ( `NULL_COUNT` )\*\***：列中空值的数量。

这些信息通常不直接可见，而是隐含在优化器的行为中。

## 2. 统计信息的用途

1. 查询优化：

- 优化器利用统计信息选择执行计划（如索引扫描、全表扫描、排序方式等）。
- 根据列的分布和索引的选择性，优化器决定如何访问数据。

2. 估算查询成本：

- 通过行数和索引分布，估算读取的页面数量。
- 比较不同执行路径的代价，选择最低成本的路径。

3. JOIN 操作的优化：

- 根据统计信息估算表的大小，确定 JOIN 的顺序和算法（如嵌套循环、哈希连接）。

### 3. 统计信息的更新

#### 3.1 自动更新

MySQL 默认会自动维护统计信息：

- 当表中的数据发生较大变化（如插入、更新、删除超过 10% 行）时，触发自动更新。
- `innodb_stats_auto_recalc` 参数控制是否自动更新（默认 `ON`）。

#### 3.2 手动更新

可以手动更新统计信息：

- **ANALYZE TABLE**：更新表的统计信息。

```
ANALYZE TABLE your_table;
```

- **OPTIMIZE TABLE**：不仅更新统计信息，还整理表和索引。

```
OPTIMIZE TABLE your_table;
```

- **FLUSH TABLES**：刷新表的统计信息缓存。

```
FLUSH TABLES;
```

### 4. 统计信息的存储与访问

#### 4.1 存储位置

- MySQL 的统计信息存储在内存中，重启后会重新计算。
- 如果启用了 `persistent_stats`，统计信息将持久化存储到磁盘。

启用持久化统计信息：

```
SET PERSIST innodb_stats_persistent = ON;
```

#### 4.2 查询统计信息

统计信息可以通过以下途径查询：

- `INFORMATION_SCHEMA`

:

- `TABLES` 表：表级统计信息。
- `STATISTICS` 表：索引级统计信息。

- **EXPLAIN**

: 查询优化器的执行计划。

```
EXPLAIN SELECT * FROM your_table WHERE column = value;
```

### 5. 统计信息的限制



1. 估算值：
- MySQL 的统计信息是估算的，并不完全精确，特别是在表数据发生频繁变化时。
2. 锁定统计信息：
- 在高并发场景下，统计信息的更新可能会导致性能波动。
3. 统计信息不足：
- 当列的数据分布非常不均时，优化器可能做出次优选择。
  - 可以通过 `ANALYZE` 或手动提示（如 `USE INDEX`）优化查询。

6. 优化统计信息

1. 定期维护：
- 使用 `ANALYZE TABLE` 和 `OPTIMIZE TABLE` 更新统计信息。
  - 监控表的变化量，必要时重新生成统计信息。
2. 持久化统计信息：
- 启用 `innodb_stats_persistent`，减少统计信息重建的开销。
3. 调整采样率：
- ```
SET GLOBAL innodb_stats_sample_pages = 32; -- 默认是 8
```

  - 使用 `innodb_stats_sample_pages` 参数控制采样页面数，提高统计信息的精度。
4. 索引优化：
- 为常用查询添加高选择性索引，提高查询效率。

7. 全库手动优化统计信息方法

MySQL 没有提供直接更新所有数据库中所有表统计信息的单一命令，但可以通过以下几种方式间接实现。

方法 1：使用 INFORMATION\_SCHEMA 动态生成命令

通过 `INFORMATION_SCHEMA.TABLES` 查询所有数据库和表，然后动态生成 `ANALYZE TABLE` 语句。

示例代码：

```
SELECT CONCAT('ANALYZE TABLE ', TABLE_SCHEMA, '`.`', TABLE_NAME, ' ');
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_TYPE = 'BASE TABLE';
```

这将生成一系列 `ANALYZE TABLE` 语句，例如：

```
ANALYZE TABLE `database1`.`table1`;
ANALYZE TABLE `database1`.`table2`;
```

你可以复制这些语句并执行，或者将它们封装为存储过程来自动执行。

方法 2：使用存储过程自动执行

创建一个存储过程，用于更新所有数据库中的表统计信息。

存储过程代码：

```
DELIMITER //
```

```
CREATEPROCEDURE UpdateAllStatistics()
BEGIN
    DECLARE done INTDEFAULTFALSE;
    DECLARE dbName VARCHAR(64);
    DECLARE tblName VARCHAR(64);
    DECLARE cur CURSORFOR
        SELECT TABLE_SCHEMA, TABLE_NAME
        FROM INFORMATION_SCHEMA.TABLES
        WHERE TABLE_TYPE = 'BASE TABLE';

    DECLARE CONTINUE HANDLERFORNOTFOUNDSET done = TRUE;

    OPEN cur;

    read_loop: LOOP
        FETCH cur INTO dbName, tblName;
        IF done THEN
            LEAVE read_loop;
        ENDIF;
        SET @query = CONCAT('ANALYZE TABLE ``, dbName, ``.``, tblName, ``');
        PREPARE stmt FROM @query;
        EXECUTE stmt;
        DEALLOCATEPREPARE stmt;
    ENDLOOP;

    CLOSE cur;
END;
//

DELIMITER ;
```

执行存储过程：

```
CALL UpdateAllStatistics();
```

### 方法 3：使用 Shell 脚本结合 MySQL 客户端

如果你有系统权限，可以编写一个 Shell 脚本，利用 `mysql` 命令执行 `ANALYZE TABLE`。

Shell 脚本示例：

```
#!/bin/bash

# MySQL 连接信息
USER="your_user"
PASSWORD="your_password"
HOST="localhost"

# 获取所有表并执行 ANALYZE TABLE
mysql -u$USER -p$PASSWORD -h$HOST -N -e "SELECT CONCAT('ANALYZE TABLE ``, TABLE_SCHEMA,
```

使用步骤：

- 保存脚本为 `update_statistics.sh`。
- 给脚本赋予可执行权限：`chmod +x update_statistics.sh`。
- 运行脚本：`./update_statistics.sh`。

### 注意事项

- 权限要求：
  - 确保运行 `ANALYZE TABLE` 的用户有足够的权限（`ANALYZE` 和 `SELECT` 权限）。
- 影响性能：
  - 更新统计信息会对正在运行的查询产生影响，建议在业务低峰期进行。

更新统计信息会对正在运行的查询产生影响，建议在生产环境定期运行

3. 适用范围：

ANALYZE TABLE

仅适用于 MyISAM 和 InnoDB 表。其他存储引擎可能不支持此操作。

4. 自动化：

可以结合计划任务（如 `cron`）定期运行统计信息更新脚本，确保统计信息保持最新。

根据你的实际需求和环境，选择适合的实现方式。如果你的表非常多，建议监控更新过程，避免对生产环境造成过大影响。

往期经典:

MySQL集群环境检查与恢复指南

MySQL 日常操作命令

MYSQL8.0权限新特色-Role

MySQL连接数上限问题及解决方案

MySQL主从又双叒异常怎么办？一键主从修复脚本你值得拥有

MySQL数据归档详解

MySQL 5.7.44 升级到 MySQL 8.0.35 详细文档

MySQL如何通过binlog日志恢复数据

MYSQL刷新日志FLUSH LOGS命令详解

MySQL单机部署多个实例方法

MySQL InnoDB集群节点异常后加入失败问题处理

搭建一台高性能的MYSQL服务器相关基础配置

MySQL高可用集群MGR参数详解

MYSQL高可用InnoDB Cluster搭建实战

如何选择MYSQL高可用架构，看这篇你就明白了！

MYSQL原生高可用MGR详解。

MYSQL高可用Percona XtraDB Cluster (PXC) 搭建详解

MYSQL PXC 日常管理维护

数据库工具神器-Percona Toolkit,了解一下

MySQL 高可用架构MHA，你了解吗？

Mysql双主+keepalived实现故障自动切换

MYSQL表分区详解

MYSQL主从复制GTID模式一键搭建

MYSQL快速复制新姿势，克隆插件，了解一下！

MYSQL备份实战

数据库常见的概念和术语







DBA小强哥

👍 喜欢作者

# 高可用 4 # mysql版本 9 # 数据库 48 # 故障处理 10 # MYSQL 53

高可用 · 目录

➤ 上一篇 · Percona Distribution for MySQL 和 Percona Server for MySQL

个人观点，仅供参考