聚积宝
51JUJIBAO.COM

MySQL 8 新特性体验

简朝阳 @聚积宝

PHPCON
www.phpconchina.com

MySQL 8 新特性体验

聚积宝
51JUJIBAO.COM

About me

简朝阳

Oracle ACE Director

ACOUG 核心成员

聚积宝联合创始人 & CTO

《MySQL 性能调优与架构设计》作者

MySQL 8 新特性体验

聚积宝
51JUJIBAO.COM

- 数据字典 InnoDB 化带来的改变
- 如何利用Invisible Index（隐式索引）调优
- Role（角色）如何让管理更方便
- Histgram（直方图）的引入将带来哪些影响
- 其他部分重要特性

聚积宝
51JUJIBAO.COM

MySQL 8：Integrated Data Dictionary

表象：

- frm, par, trg, db.opt 等消失
- SDI 增加
- mysql数据库全部变成innodb表

本质：

- **MyISAM** 引擎的退出
- DDL **原子化**

MySQL 8：Integrated Data Dictionary

演示环境：

```
root@localhost : (none) 04:28:10> show variables like '%version%';
+-------------------------+-------------------------------+
| Variable_name           | Value                         |
+-------------------------+-------------------------------+
| innodb_version          | 8.0.1                         |
| protocol_version        | 10                            |
| slave_type_conversions  |                               |
| tls_version             | TLSv1,TLSv1.1                 |
| version                 | 8.0.1-dmr-log                 |
| version_comment         | MySQL Community Server (GPL) |
| version_compile_machine | x86_64                        |
| version_compile_os      | macos10.12                    |
+-------------------------+-------------------------------+
```

聚积宝
51JUJIBAO.COM

MySQL 8：Invisible Index

- 什么是Invisible index（隐藏索引）
  - ✓ 通过 VISIBLE|INVISIBLE 关键字控制索引是否对Optimizer可见
  - ✓ 可以在 创建/修改 过程指定
- 哪些使用场景
  - ✓ 删除索引
  - ✓ 新索引替换老索引

MySQL 8：Invisible Index

```
root@localhost : mysql8test 06:45:10> show indexes from tbl_idx_test;
+--------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+
| Table        | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible |
+--------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+
| tbl_idx_test | 0          | PRIMARY  |            1 | a           | A         |       10305 | NULL     | NULL   |      | BTREE      |         |               | YES     |
| tbl_idx_test | 1          | idx_c_b  |            1 | c           | A         |        6567 | NULL     | NULL   | YES  | BTREE      |         |               | YES     |
| tbl_idx_test | 1          | idx_c_b  |            2 | b           | A         |       10305 | NULL     | NULL   | YES  | BTREE      |         |               | YES     |
| tbl_idx_test | 1          | idx_d_b  |            1 | d           | A         |        6513 | NULL     | NULL   | YES  | BTREE      |         |               | YES     |
| tbl_idx_test | 1          | idx_d_b  |            2 | b           | A         |       10305 | NULL     | NULL   | YES  | BTREE      |         |               | YES     |
+--------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+
5 rows in set (0.00 sec)

root@localhost : mysql8test 06:45:13> alter table tbl_idx_test alter index idx_c_b invisible;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

root@localhost : mysql8test 06:48:15> show indexes from tbl_idx_test;
+--------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+
| Table        | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible |
+--------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+
| tbl_idx_test | 0          | PRIMARY  |            1 | a           | A         |       10305 | NULL     | NULL   |      | BTREE      |         |               | YES     |
| tbl_idx_test | 1          | idx_c_b  |            1 | c           | A         |        6567 | NULL     | NULL   | YES  | BTREE      |         |               | NO      |
| tbl_idx_test | 1          | idx_c_b  |            2 | b           | A         |       10305 | NULL     | NULL   | YES  | BTREE      |         |               | NO      |
| tbl_idx_test | 1          | idx_d_b  |            1 | d           | A         |        6513 | NULL     | NULL   | YES  | BTREE      |         |               | YES     |
| tbl_idx_test | 1          | idx_d_b  |            2 | b           | A         |       10305 | NULL     | NULL   | YES  | BTREE      |         |               | YES     |
+--------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+
5 rows in set (0.00 sec)
```

聚积宝
51JUJIBAO.COM

MySQL 8：Roles

- Role 就是一系列权限组合在一起成为一个**权限集合**
- 创建一个Role在MySQL中就是创建了一个具备一系列权限的**特殊用户**(需要指定登录域)
- 将 Role 以授权方式授予给普通用户后，用户将具有**Role的整个权限集合**

MySQL 8：Roles

```
root@localhost : mysql8test 06:20:30> CREATE ROLE role_test@localhost;
Query OK, 0 rows affected (0.00 sec)

root@localhost : mysql8test 06:20:35> select user,host from mysql.user;
+-----------+-----------+
| user      | host      |
+-----------+-----------+
| mysql.sys | localhost |
| role_test | localhost |
| root      | localhost |
+-----------+-----------+
```

```
root@localhost : mysql8test 06:28:18> show grants for role_test@localhost;
+-----------------------------------------------------------------------------+
| Grants for role_test@localhost                                              |
+-----------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `role_test`@`localhost`                               |
| GRANT SELECT ON `mysql8test`.`tbl_role_test1` TO `role_test`@`localhost`    |
| GRANT SELECT ON `mysql8test`.`tbl_role_test2` TO `role_test`@`localhost`    |
+-----------------------------------------------------------------------------+
3 rows in set (0.00 sec)

root@localhost : mysql8test 06:28:33> grant role_test@localhost to user_role_test1@localhost;
Query OK, 0 rows affected (0.00 sec)

root@localhost : mysql8test 06:29:05> show grants for user_role_test1@localhost;
+-------------------------------------------------------------+
| Grants for user_role_test1@localhost                        |
+-------------------------------------------------------------+
| GRANT USAGE ON *.* TO `user_role_test1`@`localhost`         |
| GRANT `role_test`@`localhost` TO `user_role_test1`@`localhost` |
+-------------------------------------------------------------+
```

MySQL 8：Histogram

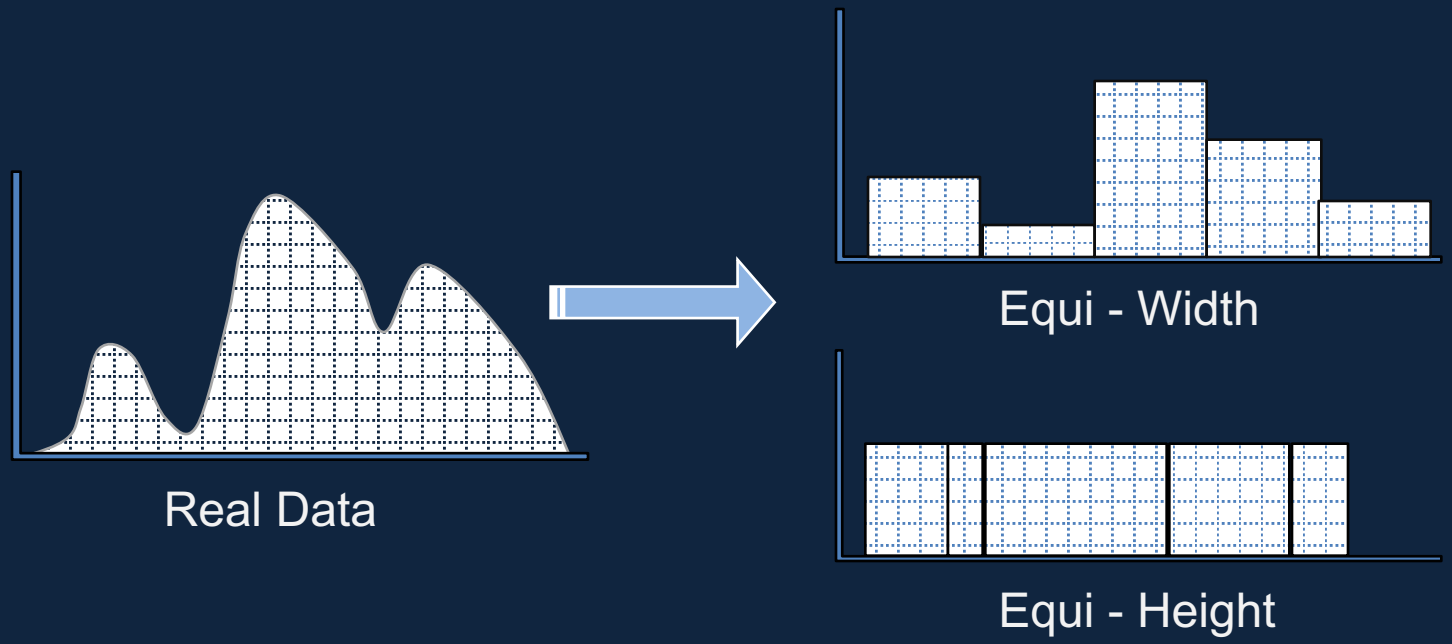
- 什么是 Histogram（直方图）
  - ✓ 一种统计信息，统计字段内各值的分布情况
  - ✓ MySQL的Histogram将有以下2种
    - ➢ 等宽直方图：单值桶，值与频率
    - ➢ 等高直方图：多值桶，上 / 下界，**累积频率**及不同值个数
  - ✓ MySQL的Histogram信息存储在mysql.column_stats中：

```
root@localhost : mysql8test 06:48:17> desc mysql.column_stats;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| database_name | varchar(64) | NO   | PRI | NULL    |       |
| table_name    | varchar(64) | NO   | PRI | NULL    |       |
| column_name   | varchar(64) | NO   | PRI | NULL    |       |
| histogram     | json        | NO   |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
```

聚积宝
51JUJIBAO.COM

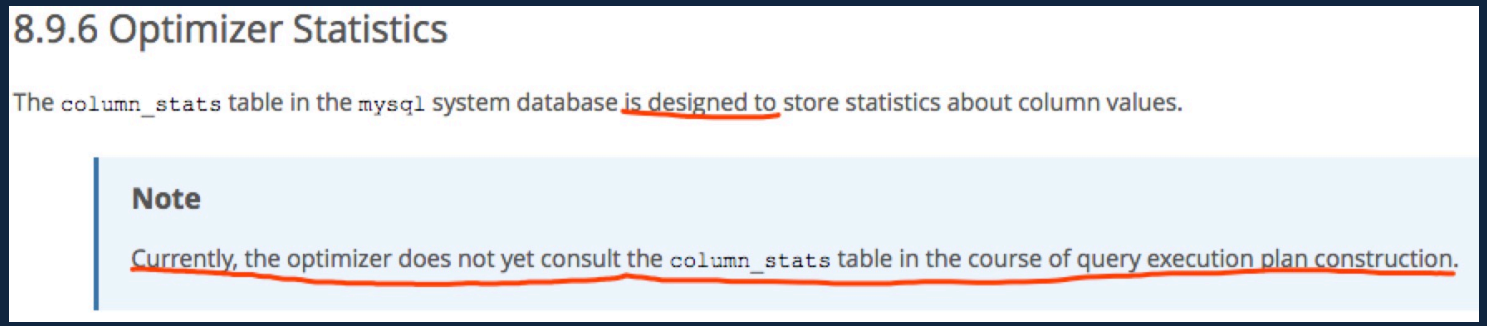
MySQL 8：Histogram

- MySQL Histogram 种类
  - ✓ 等宽直方图：单值桶，值与频率
  - ✓ 等高直方图：多值桶，上 / 下界，**累积频率**及不同值个数



MySQL 8：Histogram

- 对我们有什么用
  - ✓ Optimizer 可以根据统计数据估算查询条件中谓词的选择率来**优化执行计划**
- 如何使用Histogram
  - ✓ 暂时还得等等：



8.9.6 Optimizer Statistics

The `column_stats` table in the `mysql` system database is designed to store statistics about column values.

> **Note**
>
> Currently, the optimizer does not yet consult the `column_stats` table in the course of query execution plan construction.

聚积宝
51JUJIBAO.COM

MySQL 8：the ohers ...

- 全局参数持久化
  - ✓ 借鉴Oracle对于配置参数的管理，在数据库Instance运行过程中修改全局参数变量的时候可以通过关键字（PERSIST）来控制将参数写入配置文件，而不只是在当前Instance运行期间有效。
  - ✓ 持久化参数不会修改my.cnf,而是写入MySQL数据目录下的mysqld-auto.cnf
  - ✓ mysqld-auto.cnf 比 my.cnf 具有更高优先级
  - ✓ Mysqld-auto.cnf 是否被读取受persisted_globals_load 参数控制

```
root@localhost : (none) 11:27:16> show variables like 'sync_binlog';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| sync_binlog   | 0     |
+---------------+-------+
1 row in set (0.01 sec)

root@localhost : (none) 11:27:31> set persist sync_binlog=1;
Query OK, 0 rows affected (0.01 sec)

root@localhost : (none) 11:27:47> show variables like 'sync_binlog';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| sync_binlog   | 1     |
+---------------+-------+
1 row in set (0.01 sec)

root@localhost : (none) 11:27:50> exit
Bye
Sky-MBPro:mydata Sky$ ls
auto.cnf                mysql8test              mysqld-auto.cnf         performance_schema      sys_4.SDI
mysql                   mysql8test_5.SDI        performance_sche_3.SDI  sys
Sky-MBPro:mydata Sky$ cat mysqld-auto.cnf
{ "mysql_server": {"sync_binlog": "1" } }
```

MySQL 8：the ohers ...

- 自增序列持久化（InnoDB）
  - ✓ MySQL 8 以前
    - ➢ 自增序列在系统重启的时候重新计算出表上最大值作为下一次分配起始点
    - ➢ 若存在大量数据被删除的场景会出现自增序列重用的情况
  - ✓ MySQL 8 及以后
    - ➢ 自增序列写入Redo Log
    - ➢ 重启后从Redo Log中读取上一次最后分配值
    - ➢ 仅针对 InnoDB 有用

MySQL 8 新特性体验

Q & A