

Column Statistics in Hive

- [Introduction](#)
- [HiveQL changes](#)
- [Metastore Schema](#)
- [Metastore Thrift API](#)

Introduction

This document describes changes to a) HiveQL, b) metastore schema, and c) metastore Thrift API to support column level statistics in Hive. Please note that the document doesn't describe the changes needed to persist histograms in the metastore yet.



Version information

Column statistics are introduced in Hive 0.10.0 by [HIVE-1362](#). This is the design document.

Column statistics auto gather is introduced in Hive 2.3 by [HIVE-11160](#). This is also the design document.

For general information about Hive statistics, see [Statistics in Hive](#). For information about top K statistics, see [Column Level Top K Statistics](#).

HiveQL changes

HiveQL currently supports the [analyze command](#) to compute statistics on tables and partitions. HiveQL's analyze command will be extended to trigger statistics computation on one or more column in a Hive table/partition. The necessary changes to HiveQL are as below,

```
analyze table t [partition p] compute statistics for [columns c,...];
```

Please note that table and column aliases are not supported in the analyze statement.

To view column stats :

```
describe formatted [table_name] [column_name];
```

Metastore Schema

To persist column level statistics, we propose to add the following new tables,

```
CREATE TABLE TAB_COL_STATS
```

```
(  
  CS_ID NUMBER NOT NULL,  
  TBL_ID NUMBER NOT NULL,  
  COLUMN_NAME VARCHAR(128) NOT NULL,  
  COLUMN_TYPE VARCHAR(128) NOT NULL,  
  TABLE_NAME VARCHAR(128) NOT NULL,  
  DB_NAME VARCHAR(128) NOT NULL,
```

```
  LOW_VALUE RAW,  
  HIGH_VALUE RAW,  
  NUM_NULLS BIGINT,  
  NUM_DISTINCTS BIGINT,
```

```
  BIT_VECTOR, BLOB, /* introduced in HIVE-16997 in Hive 3.0.0 */
```

```
  AVG_COL_LEN DOUBLE,  
  MAX_COL_LEN BIGINT,  
  NUM_TRUES BIGINT,  
  NUM_FALSES BIGINT,  
  LAST_ANALYZED BIGINT NOT NULL)
```

```
ALTER TABLE COLUMN_STATISTICS ADD CONSTRAINT COLUMN_STATISTICS_PK PRIMARY KEY (CS_ID);
```

```
ALTER TABLE COLUMN_STATISTICS ADD CONSTRAINT COLUMN_STATISTICS_FK1 FOREIGN KEY (TBL_ID) REFERENCES TBLS (TBL_ID)  
INITIALLY DEFERRED ;
```

```
CREATE TABLE PART_COL_STATS
```

```
(  
  CS_ID NUMBER NOT NULL,  
  PART_ID NUMBER NOT NULL,
```

```
DB_NAME VARCHAR(128) NOT NULL,  
COLUMN_NAME VARCHAR(128) NOT NULL,  
COLUMN_TYPE VARCHAR(128) NOT NULL,  
TABLE_NAME VARCHAR(128) NOT NULL,  
PART_NAME VARCHAR(128) NOT NULL,
```

```
LOW_VALUE RAW,  
HIGH_VALUE RAW,  
NUM_NULLS BIGINT,  
NUM_DISTINCTS BIGINT,
```

```
BIT_VECTOR, BLOB, /* introduced in HIVE-16997 in Hive 3.0.0 */
```

```
AVG_COL_LEN DOUBLE,  
MAX_COL_LEN BIGINT,  
NUM_TRUES BIGINT,  
NUM_FALSES BIGINT,  
LAST_ANALYZED BIGINT NOT NULL)
```

```
ALTER TABLE COLUMN_STATISTICS ADD CONSTRAINT COLUMN_STATISTICS_PK PRIMARY KEY (CS_ID);
```

```
ALTER TABLE COLUMN_STATISTICS ADD CONSTRAINT COLUMN_STATISTICS_FK1 FOREIGN KEY (PART_ID) REFERENCES PARTITIONS  
(PART_ID) INITIALLY DEFERRED;
```

Metastore Thrift API

We propose to add the following Thrift structs to transport column statistics:

```
struct BooleanColumnStatsData {  
  1: required i64 numTrues,  
  2: required i64 numFalses,  
  3: required i64 numNulls  
}
```

```
struct DoubleColumnStatsData {  
  1: required double lowValue,  
  2: required double highValue,  
  3: required i64 numNulls,  
  4: required i64 numDVs,  
  
  5: optional string bitVectors  
}
```

```
struct LongColumnStatsData {  
  1: required i64 lowValue,  
  2: required i64 highValue,  
  3: required i64 numNulls,  
  4: required i64 numDVs,  
  
  5: optional string bitVectors  
}
```

```
struct StringColumnStatsData {  
  1: required i64 maxColLen,  
  2: required double avgColLen,  
  3: required i64 numNulls,  
  4: required i64 numDVs,  
  
  5: optional string bitVectors  
}
```

```
struct BinaryColumnStatsData {  
  1: required i64 maxColLen,  
  2: required double avgColLen,  
  3: required i64 numNulls  
}
```

```
struct Decimal {  
  1: required binary unscaled,  
  3: required i16 scale  
}
```

```

struct DecimalColumnStatsData {
1: optional Decimal lowValue,
2: optional Decimal highValue,
3: required i64 numNulls,
4: required i64 numDVs,
5: optional string bitVectors
}

struct Date {
1: required i64 daysSinceEpoch
}

struct DateColumnStatsData {
1: optional Date lowValue,
2: optional Date highValue,
3: required i64 numNulls,
4: required i64 numDVs,
5: optional string bitVectors
}

union ColumnStatisticsData {
1: BooleanColumnStatsData booleanStats,
2: LongColumnStatsData longStats,
3: DoubleColumnStatsData doubleStats,
4: StringColumnStatsData stringStats,
5: BinaryColumnStatsData binaryStats,
6: DecimalColumnStatsData decimalStats,
7: DateColumnStatsData dateStats
}

struct ColumnStatisticsObj {
1: required string colName,
2: required string colType,
3: required ColumnStatisticsData statsData
}

struct ColumnStatisticsDesc {
1: required bool isTblLevel,
2: required string dbName,
3: required string tableName,
4: optional string partName,
5: optional i64 lastAnalyzed
}

struct ColumnStatistics {
1: required ColumnStatisticsDesc statsDesc,
2: required list<ColumnStatisticsObj> statsObj;
}

```

We propose to add the following Thrift APIs to persist, retrieve and delete column statistics:

```

bool update_table_column_statistics(1:ColumnStatistics stats_obj) throws (1:NoSuchObjectException o1,
2:InvalidObjectException o2, 3:MetaException o3, 4:InvalidInputException o4)
bool update_partition_column_statistics(1:ColumnStatistics stats_obj) throws (1:NoSuchObjectException o1,
2:InvalidObjectException o2, 3:MetaException o3, 4:InvalidInputException o4)

ColumnStatistics get_table_column_statistics(1:string db_name, 2:string tbl_name, 3:string col_name) throws
(1:NoSuchObjectException o1, 2:MetaException o2, 3:InvalidInputException o3, 4:InvalidObjectException o4)
ColumnStatistics get_partition_column_statistics(1:string db_name, 2:string tbl_name, 3:string part_name,
4:string col_name) throws (1:NoSuchObjectException o1, 2:MetaException o2,
3:InvalidInputException o3, 4:InvalidObjectException o4)

bool delete_partition_column_statistics(1:string db_name, 2:string tbl_name, 3:string part_name, 4:string col_name) throws
(1:NoSuchObjectException o1, 2:MetaException o2, 3:InvalidObjectException o3,
4:InvalidInputException o4)
bool delete_table_column_statistics(1:string db_name, 2:string tbl_name, 3:string col_name) throws
(1:NoSuchObjectException o1, 2:MetaException o2, 3:InvalidInputException o3,
4:InvalidInputException o4)

```

Note that delete_column_statistics is needed to remove the entries from the metastore when a table is dropped. Also note that currently Hive doesn't support drop column.

Note that in V1 of the project, we will support only scalar statistics. Furthermore, we will support only static partitions, i.e., both the partition key and partition value should be specified in the analyze command. In a following version, we will add support for height balanced histograms as well as support for dynamic partitions in the analyze command for column level statistics.