Understanding How MySQL Works by Understanding Metadata

Sheeri Cabral and Patrick Galbraith

.....working title.....

I Never Metadata I Didn't Like!

About the speakers

Sheeri Cabral

- MySQL Team Lead, Pythian Group
- MySQL Administrators Bible
  - mid-May 2009
  - tinyurl.com/mysqlbible
- 8 years of experience with MySQL
- Podcasts, videos and more at www.technocation.org

About the speakers

Patrick Galbraith

- Principal Software Engineer, Lycos
- 16 Years dabbling in Open Source
- Author of Developing Web Applications using..
- Federated Storage Engine, Memcached Functions for MySQL/UDFs, DBD::mysql...

What is metadata?

Metadata is data about data

- Tutorial content vs. 5682
- Row count
- Data type

When in doubt....

...check it out!

Useful MySQL Commands

- \!
  - (not on Windows)
- \c
- SELECT NOW();
- DO SLEEP(x);

Metadata: Files

- configuration
- data, index, frm files
- temporary files

- logs
  - general
  - slow
  - binary
  - relay

Other metadata

- System variables
- Status variables
- INFORMATION_SCHEMA
- Status commands
  - SHOW SLAVE STATUS

Metadata contained within files

- Data and Index Files
- Table data dictionary file (.frm)
- Log Files

- Tools
  - File size
  - File date
  - File content
    - more
    - file (not on Windows)
    - strings (not on Windows)

.frm files

- .frm data dictionary file for each table, regardless of storage engine
- "Format"

```
CREATE TABLE states (
state_id int unsigned not null,
state_name varchar(100) not null
) ENGINE=MYISAM;
```

.frm files (continued)

- Table definitions
- Column definitions
- Other information

.frm files (continued)

- Table definitions
- Column definitions
- Other information

```
strings states.frm
PRIMARY
MyISAM
)
state_id
state_name
state_id
state_name
```

Federated .frm file

```
CREATE TABLE fed_test (state_id int)
ENGINE=FEDERATED
CONNECTION='mysql://root@127.0.0.1:3306/test/s
tates';
```

Federated .frm file

```
CREATE TABLE fed_test (state_id int)
ENGINE=FEDERATED
CONNECTION='mysql://root@127.0.0.1:3306/test/s
tates';
```

```
strings fed_test.frm
mysql://root@127.0.0.1:3306/test/states
FEDERATED
)
    state_id
state_id
```

MyISAM files

- Observe which files grow, when

```
INSERT INTO states (state_id,state_name)
 VALUES (1,'Alaska');

ALTER TABLE states ADD PRIMARY KEY (state_id);

ALTER TABLE states ADD INDEX (state_name);


INSERT INTO states (state_id, state_name)
 VALUES (2,'Alabama'), (3,'NY'),
 (4,'New Hampshire'),(5,'Hawaii');
```

Storage Engine-Specific Files

- What conclusions can we draw based on what we see the files doing?

- .frm data dictionary file for each table, regardless of storage engine
  - Blackhole, federated, memory only have .frm files
    - No data on local disk

Views

A view is created using .frm file which contains the query and other information of the view

CREATE VIEW odd_states AS SELECT state_name FROM states WHERE state_id % 2 = 1;

Views .frm

```
strings odd_states.frm
TYPE=VIEW
query=select `test`.`states`.`state_id` AS
`state_id`,`test`.`states`.`state_name` AS
`state_name` from `test`.`states` where
((`test`.`states`.`state_id` % 2) = 1)
md5=c11aab2ff14199983ff57fc97ac4c1f9
updatable=1          algorithm=0
definer_user=root       definer_host=localhost
suid=2         with_check_option=0
revision=1          timestamp=2009-04-20 09:44:30
create-version=1
source=SELECT * FROM states WHERE state_id % 2 = 1
client_cs_name=latin1
connection_cl_name=latin1_swedish_ci
view_body_utf8=SELECT * FROM states WHERE state_id
% 2 = 1
```

CSV files

```
CREATE TABLE csv_test (id int) ENGINE=CSV;

file csv_test.CSM

INSERT INTO csv_test (id) VALUES (1),(2),(3);

ls -l csv_test*
```

CSM

CSV does not allow indexes, so CSM is not indexes.

```
INSERT INTO csv_test (id)
 VALUES (4),(5),(6);
```

```
INSERT INTO csv_test (id) VALUES (0);
```

```
strings csv_test.CSM
```

```
file csv_test.CSM
```

CSM (continued)

```
INSERT INTO csv_test (id) VALUES (0);
```

```
strings csv_test.CSM
file csv_test.CSM
```

Suggestions?

CSM (continued)

```
CHECK TABLE csv_test;
```

```
ANALYZE TABLE csv_test;
```

CSM is for CSV metadata such as statistics.

ARCHIVE files

```
CREATE TABLE archive_test (id int,
name VARCHAR(32) DEFAULT NULL)
ENGINE=ARCHIVE;
```

```
strings archive_test.frm
```

```
strings archive_test.ARZ
```

```
INSERT INTO archive_test (id, name) VALUES
(1,'Patrick'),(2,'Sheeri'),(3,'Ronald'),
(4,'Bob');
```

ARZ

```
mysql> \! strings archive_test.ARZ
ARCHIVE
)
name
name
-8#5
sR8A|
```

MERGE files

```
CREATE TABLE mrg1 (id int) ENGINE=MYISAM;
CREATE TABLE mrg2 (id int) ENGINE=MYISAM;


CREATE TABLE merge_test (id int)
UNION=(mrg2,mrg1);

\! strings /var/lib/mysql/test/*.MRG
```

MERGE files (continued)

```
ALTER TABLE merge_test UNION=(mrg2,mrg1);
```

```
\! strings /var/lib/mysql/test/*.MRG
```

The MRG table stores the UNION definition in order.

InnoDB Files

- tablespace files contain data and indexes
  - Single tablespace contains data dictionary
  - File-per-table, data and indexes per table
  - Else, data dictionary + data + indexes

  - ***`ls -1 /usr/local/mysql/var/data1/ibdata*`***
  - ***`/usr/local/mysql/var/data1/ibdata1`***
  - ***`/usr/local/mysql/var/data1/ibdata2`***

InnoDB File-per-table Files

```
 ls -l /usr/local/mysql/var/data1/userdb/*ibd
/usr/local/mysql/var/data1/userdb/cities.ibd
/usr/local/mysql/var/data1/userdb/regions.ibd
/usr/local/mysql/var/data1/userdb/states.ibd
/usr/local/mysql/var/data1/userdb/users.ibd
```

Temporary files

User-Created Temp Tables

```
ls -a /tmp/#sql*

 CREATE TEMPORARY TABLE foo
 (a int, name varchar(32));

INSERT INTO foo VALUES (1, 'one'), (2,'two');

ls -a /tmp/#sql*
```

User-Created Temp Tables

```
ls -a /tmp/#sql*
/tmp/#sql31e7_795f_0.frm
/tmp/#sql31e7_795f_0.MYD
/tmp/#sql31e7_795f_0.MYI
```

On-disk temporary MyISAM tables are created for large intermediate results for queries.

What about ALTER TABLE type temporary tables?

System Variables

- Set at startup
  - command line option
  - option file
- Many can be set dynamically
  - SET statement
- Scoped globally or per session

Getting System Variables

```
INFORMATION_SCHEMA:
SELECT Variable_value FROM
INFORMATION_SCHEMA.GLOBAL_VARIABLES
WHERE Variable_name = 'max_connections';
```

As a variable:

```
SELECT @@global.max_connections;
```

SHOW:

```
SHOW GLOBAL VARIABLES LIKE 'max_connections';
```

Setting System Variables

Configuration file (restart required)
max_connections=1000;

As a variable:
SET @@global.max_connections=1000;

SHOW:
SET GLOBAL max_connections=1000;

Status Variables

- Like system variables, scope is global or session
- Read-only
- Status of running system
- Use system and status variables together

Getting Status Variables

```
INFORMATION_SCHEMA:
SELECT Variable_value FROM
INFORMATION_SCHEMA.GLOBAL_STATUS
WHERE Variable_name = 'max_connections';
```

SHOW:

```
SHOW GLOBAL VARIABLES LIKE 'max_connections';
```

```
mysql> SHOW GLOBAL VARIABLES LIKE 'Query_cache%';
+------------------------------+----------+
| Variable_name                | Value    |
+------------------------------+----------+
| query_cache_limit            | 1048576  |
| query_cache_min_res_unit     | 4096     |
| query_cache_size             | 16777216 |
| query_cache_type             | ON       |
| query_cache_wlock_invalidate | OFF      |
+------------------------------+----------+
5 rows in set (0.00 sec)

mysql> SHOW GLOBAL STATUS LIKE 'Qcache%';
+-------------------------+----------+
| Variable_name           | Value    |
+-------------------------+----------+
| Qcache_free_blocks      | 1        |
| Qcache_free_memory      | 16759704 |
| Qcache_hits             | 0        |
| Qcache_inserts          | 0        |
| Qcache_lowmem_prunes    | 0        |
| Qcache_not_cached       | 17       |
| Qcache_queries_in_cache | 0        |
| Qcache_total_blocks     | 1        |
+-------------------------+----------+
8 rows in set (0.00 sec)
```

Getting Status Information

- SHOW TABLE STATUS;
  - INFORMATION_SCHEMA.TABLES
- SHOW MASTER STATUS;
- SHOW SLAVE STATUS;

Profiling

- SHOW PROFILE
- SHOW PROFILES

SHOW SLAVE STATUS

```
mysql> show slave status\G
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: localhost
                  Master_User: repl
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: bin.000012
          Read_Master_Log_Pos: 2536
               Relay_Log_File: relay.000019
                Relay_Log_Pos: 245
        Relay_Master_Log_File: bin.000012
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
```

InnoDB status

- ***`SHOW ENGINE INNODB STATUS;`***
- semaphores
- transaction information
- file IO
- insert buffer and index
- thread state
- buffer pool and memory state
- log file status
- row operations

Tables in INFORMATION_SCHEMA

CHARACTER_SETS
COLLATIONS
COLLATION_CHARACTER_SET_APPLICABILITY
COLUMNS
COLUMN_PRIVILEGES
ENGINES
EVENTS
FILES
GLOBAL_STATUS
GLOBAL_VARIABLES
KEY_COLUMN_USAGE
PARTITIONS
PLUGINS
PROCESSLIST
REFERENTIAL_CONSTRAINTS
ROUTINES
SCHEMATA
SCHEMA_PRIVILEGES
SESSION_STATUS
SESSION_VARIABLES
STATISTICS
TABLES
TABLE_CONSTRAINTS
TABLE_PRIVILEGES
TRIGGERS
USER_PRIVILEGES
VIEWS

Information Schema tables

Database objects

- SCHEMATA
- TABLES
- VIEWS
- PARTITIONS
- FILES
- COLUMNS
- ROUTINES
- EVENTS
- TRIGGERS
- PLUGINS

Data Objects in the Data Dictionary

- Instead of SHOW commands

SHOW DATABASES;

SELECT SCHEMA_NAME FROM SCHEMATA;

SHOW PROCESSLIST
SELECT Id, User, db, Command, state FROM PROCESSLIST
WHERE User='root';

Checking for Data Fragmentation

```
SELECT TABLE_NAME, DATA_LENGTH, INDEX_LENGTH
FROM TABLES WHERE TABLE_SCHEMA='test' AND
ENGINE!='INNODB';

\! ls -l /var/lib/mysql/test/



\! ls -l /var/lib/mysql/test/
```

InnoDB Data Fragmentation

Shared tablespace for data/indexes:

```
SELECT SUM(DATA_LENGTH+INDEX_LENGTH) FROM
TABLES WHERE ENGINE='INNODB';
```

```
ls -l ibdata*
```

InnoDB Data Fragmentation

File-per-table:

```
SELECT TABLE_NAME, (DATA_LENGTH+INDEX_LENGTH)
as Size FROM TABLES WHERE ENGINE='INNODB';
```

```
ls -l *ibd
```

```
mysql> SELECT * FROM TABLES WHERE TABLE_NAME =
'mrg1'\G
  TABLE_CATALOG: NULL
   TABLE_SCHEMA: test
     TABLE_NAME: mrg1
     TABLE_TYPE: BASE TABLE
         ENGINE: MyISAM
        VERSION: 10
     ROW_FORMAT: Fixed
     TABLE_ROWS: 12
 AVG_ROW_LENGTH: 7
    DATA_LENGTH: 84
MAX_DATA_LENGTH: 1970324836974591
   INDEX_LENGTH: 3072
      DATA_FREE: 0
 AUTO_INCREMENT: NULL
    CREATE_TIME: 2009-04-19 06:24:52
    UPDATE_TIME: 2009-04-19 06:26:22
    CHECK_TIME: NULL
TABLE_COLLATION:
latin1_swedish_ci
       CHECKSUM: NULL
 CREATE_OPTIONS:
  TABLE_COMMENT:
1 row in set (0.00 sec)
```

CHECKSUM

- MyISAM only
- Constant checksum

```
ALTER TABLE test.mrg1 CHECKSUM=1;

SELECT CHECKSUM FROM TABLES WHERE
TABLE_NAME='mrg1';

CHECKSUM TABLE test.mrg1;

INSERT INTO test.mrg1 (id) VALUES (50),(60);
```

Index Objects

- STATISTICS
- TABLE_CONSTRAINTS
- KEY_COLUMN_USAGE
- REFERENTIAL CONSTRAINTS

SHOW INDEXES
SELECT TABLE_NAME, INDEX_NAME,
GROUP_CONCAT(COLUMN_NAME) FROM STATISTICS
WHERE TABLE_SCHEMA='test' GROUP BY INDEX_NAME
ORDER BY INDEX_NAME, SEQ_IN_INDEX;

Key Constraints

TABLE_CONSTRAINTS

- PRIMARY
- UNIQUE
- FOREIGN

```
SELECT DISTINCT CONSTRAINT_NAME FROM
KEY_COLUMN_USAGE WHERE TABLE_SCHEMA NOT IN
('information_schema','mysql');
```

Referential Constraints

"What tables have foreign keys to the address table?"

```
SELECT  CONCAT(TABLE_NAME,
' depends on ', REFERENCED_TABLE_NAME)
FROM REFERENTIAL_CONSTRAINTS
WHERE REFERENCED_TABLE_NAME='address'
ORDER BY REFERENCED_TABLE_NAME;
```

Your Homework

Create a better mysqldump:

Write a wrapper script to make mysqldump foreign-key aware, and export tables in the proper order.

(or a script that echoes mysqldump commands in proper order)

(or modify mysqldump, which is a C file...)

Privilege Information

Privileges

- USER_PRIVILEGES
- SCHEMA_PRIVILEGES
- TABLE_PRIVILEGES
- COLUMN_PRIVILEGES

```
SELECT GRANTEE, GROUP_CONCAT(PRIVILEGE_TYPE)
FROM USER_PRIVILEGES GROUP BY GRANTEE\G
```

System and Status Variables

- SESSION_STATUS
- SESSION_VARIABLES
- GLOBAL_STATUS
- GLOBAL_VARIABLES

PROCESSLIST

- User
- Host
- DB
- Time
-
- Info

PROCESSLIST COMMAND

- Query
- Sleep
- Statistics
- Kill
-
- Field List

Replication COMMANDs

- Connect Out
- Connect
- Register Slave
- Binlog Dump
- Table Dump

Prepared Statement COMMANDs

- `Prepare`
- `Reset stmt`
- `Execute`
- `Fetch`
- `Close stmt`

Less Frequent COMMANDs

```
Delayed insert

Change user
Init DB


Create DB
Drop DB

Refresh
```

Less Frequent COMMANDs

```
Error

Long Data
Ping
Quit
Daemon - internal thread
Debug
Set option
Shutdown
```

Less Frequent COMMANDs

```
Error

Long Data
Ping
Quit
Daemon - internal thread
Debug
Set option
Shutdown
```

General States

```
After create
Analyzing
Checking table
cleaning up
closing tables
converting HEAP to MyISAM
copy to tmp table
Copying to group table
Copying to tmp table
Copying to tmp table on disk
Creating index
Creating sort index
creating table
Creating tmp table
deleting from main table
```

Cluster States

```
Processing events
Committing events to binlog
Syncing ndb table schema operation and
binlog
Processing events from schema table
Shutting down
Waiting for event from ndbcluster
Waiting for first event from ndbcluster
Waiting for ndbcluster binlog update to
reach current position
Waiting for ndbcluster to start
Waiting for schema epoch
Opening mysql.ndb_apply_status
```

InnoDB system variables

SELECT * FROM GLOBAL_VARIABLES WHERE VARIABLE NAME LIKE '%INNO%';

INNODB_FILE_PER_TABLE YES
INNODB_DATA_FILE_PATH ibdata1:10M;ibdata2:10M:autoextend

INNODB_DATA_HOME_DIR /usr/local/mysql/var/data2

INNODB_LOG_GROUP_HOME_DIR /usr/local/mysql/var/data2

InnoDB system variables

INNODB_LOCK_WAIT_TIMEOUT	50
INNODB_LOG_BUFFER_SIZE	8388608
INNODB_BUFFER_POOL_SIZE	67108864
INNODB_ADDITIONAL_MEM_POOL_SIZE	10485760
INNODB_LOG_FILE_SIZE	5242880
INNODB_MAX_DIRTY_PAGES_PCT	90

InnoDB system variables

INNODB_LOCK_WAIT_TIMEOUT	50
INNODB_LOG_BUFFER_SIZE	8388608
INNODB_BUFFER_POOL_SIZE	67108864
INNODB_ADDITIONAL_MEM_POOL_SIZE	10485760
INNODB_LOG_FILE_SIZE	5242880
INNODB_MAX_DIRTY_PAGES_PCT	90

...and so on, and so on....

Questions?

Comments?
Suggestions?

Thank you!

Patrick Galbraith, patg@patg.net

Sheeri K. Cabral, awfief@gmail.com

Twitter: at 5682 tutorial, it's awesome!
#mysqlconf 5682