# Common Schema

A Framework for
MySQL Server Administration

2012-04-12

http://code.google.com/p/common-schema/

# Welcome!

Shlomi Noach
http://code.openark.org/blog/shlomi-noach

Common Schema founder

Roland Bouman
http://rpbouman.blogspot.com/
@rolandbouman

Common Schema Contributor

# Agenda

- What's `common_schema`?
- Installation
- Monitoring
- Security
- Schema & Query Object Analysis
- Function library
- Scripting
- How to contribute

# What's common_schema?

- MySQL DBA Toolkit
- Self-contained database schema
  - Tables
  - Views
  - Stored Routines
- MySQL >= 5.1
- BSD Licensed

# Getting common_schema

- Download
  - http://code.google.com/p/common-schema/downloads
- Code: SQL Script
  - `common_schema_mysql_51-r<nnn>.sql`
  - `common_schema_innodb_plugin-r<nnn>.sql`
  - `common_schema_percona_server-r<nnn>.sql`
- Documentation: HTML
  - `common_schema_doc_r<nnn>.tar.gz`

# Installing common_schema

```
mysql> source ~/Downloads/common_schema_innodb_plugin-r218.sql
Database changed

Query OK, 0 rows affected (0.10 sec)

Query OK, 0 rows affected (0.10 sec)

...

Query OK, 0 rows affected (0.10 sec)

+-----------------------------------------------------------+
| complete                                                  |
+-----------------------------------------------------------+
| Installation complete. Thank you for using common_schema! |
+-----------------------------------------------------------+
1 row in set (0.00 sec)
```

# Installing common_schema

```
mysql> select attribute_name, substr(attribute_value, 1, 50)
    -> from   common_schema.metadata;

+-------------------------+----------------------------------------------------+
| attribute_name          | substr(attribute_value, 1, 50)                     |
+-------------------------+----------------------------------------------------+
| author                  | Shlomi Noach                                       |
| author_url              | http://code.openark.org/blog/shlomi-noach          |
| license                 |
Copyright (c) 2011 - 2012, Shlomi Noach
All right |
| license_type            | New BSD                                            |
| project_home            | http://code.google.com/p/common-schema/            |
| project_name            | common_schema                                      |
| project_repository      | https://common-schema.googlecode.com/svn/trunk/    |
| project_repository_type | svn                                                |
| revision                | 218                                                |
+-------------------------+----------------------------------------------------+
9 rows in set (0.01 sec)
```

# Built-in help

```
mysql> desc help_content;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| topic        | varchar(32) | NO   | PRI | NULL    |       |
| help_message | text        | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> select topic from help_content;
+----------------------------+
| topic                      |
+----------------------------+
| auto_increment_columns     |
.                            .
.  ...                       .
.                            .
| variables                  |
+----------------------------+
107 rows in set (0.00 sec)
```

# Built-in help

```
mysql> select help_message from help_content where topic = 'help' \G
*************************** 1. row ***************************
help_message:
NAME

help(): search and read common_schema documentation.

TYPE

Procedure

DESCRIPTION

help() is a meta routine allowing access to documentation from within
common_schema itself.
The documentation, including, for example, this very page, is embedded within
common_schema's tables, such that it can be searched and read using standard
SQL queries.
help() accepts a search term, and presents a single documentation page which
best fits the term. The term may appear within the documentation's title or
description. It could be the name or part of name of one of common_schema's
components (routines, views, ...), or it could be any keyword appearing within
the documentation.
The output is MySQL-friendly, in that it breaks the documentation into rows of
text, thereby presenting the result in a nicely formatted table.
```

# Monitoring

- Status variables
  - `global_status_diff`
  - `global_status_diff_nonzero`
- Transactions and locks
  - `innodb_locked_transactions`, `innodb_simple_locks`
  - `innodb_transactions`, `innodb_transactions_summary`
- Processlist
  - `processlist_grantees`, `processlist_per_userhost`
  - `processlist_summary`, `processlist_top`

# Monitoring Status Variables

```
mysql> select variable_name          name,
    ->        variable_value_0       prev,
    ->        variable_value_1       curr,
    ->        variable_value_diff    diff,
    ->        variable_value_psec    psec,
    ->        variable_value_pminute pmin
    -> from   global_status_diff_nonzero;
+----------------------+---------+---------+------+------+------+
| name                 | prev    | curr    | diff | psec | pmin |
+----------------------+---------+---------+------+------+------+
| handler_read_rnd_next | 1276830 | 1277453 |  623 | 62.3 | 3738 |
| handler_write        | 79880   | 80812   |  932 | 93.2 | 5592 |
| open_files           | 37      | 35      |   -2 | -0.2 |  -12 |
| qcache_not_cached    | 2303    | 2304    |    1 |  0.1 |    6 |
| select_full_join     | 33      | 34      |    1 |  0.1 |    6 |
| select_scan          | 2593    | 2595    |    2 |  0.2 |   12 |
+----------------------+---------+---------+------+------+------+
6 rows in set, 4 warnings (10.03 sec)
```

# Monitoring Processlist

```
mysql> select * from processlist_summary \G
*************************** 1. row ***************************
        count_processes: 1
      active_processes: 0
    sleeping_processes: 1
         active_queries: 0
 num_queries_over_1_sec: 0
num_queries_over_10_sec: 0
num_queries_over_60_sec: 0
    average_active_time: 0.0000
1 row in set (0.00 sec)

mysql> select * from processlist_per_userhost \G
*************************** 1. row ***************************
              user: root
              host: localhost
    count_processes: 1
   active_processes: 0
average_active_time: NULL
1 row in set (0.00 sec)
```

# Security

- Grants
  - `sql_grants`
  - `sql_show_grants`

# Object Analysis: Schema and Storage Engine

- Data Size
  - `data_size_per_engine`
  - `data_size_per_schema`

# Object Analysis: Data Size per Engine

```
mysql> select engine,
    ->        count_tables tabs,
    ->        data_size,
    ->        index_size,
    ->        total_size,
    ->        largest_table,
    ->        largest_table_size largest_size
    -> from   data_size_per_engine;
+--------+------+-----------+-----------+-----------+-------------------------+--------------+
| engine | tabs | data_size | index_size | total_size | largest_table           | largest_size |
+--------+------+-----------+-----------+-----------+-------------------------+--------------+
| CSV    |    2 |         0 |         0 |         0 | `mysql`.`slow_log`      |            0 |
| InnoDB |   80 |  39354368 |   5734400 |  45088768 | `hibernate`.`PRO_FILES` |     19398656 |
| MyISAM |   66 | 4767659576 | 2226202624 | 6993862200 | `pgn`.`t_ply`           |   4862971088 |
+--------+------+-----------+-----------+-----------+-------------------------+--------------+
3 rows in set (0.17 sec)
```

# Object Analysis: Data Size per Schema

```
mysql> select *
    -> from   data_size_per_schema
    -> where  table_schema = 'sakila' \G
*************************** 1. row ***************************
      TABLE_SCHEMA: sakila
      count_tables: 16
       count_views: 7
  distinct_engines: 2
         data_size: 4297536
        index_size: 2647040
        total_size: 6944576
     largest_table: rental
largest_table_size: 2850816
1 row in set (0.14 sec)
```

# Schema Object Analysis: Tables

- DDL scripts
  - `sql_alter_table`
  - `sql_foreign_keys`

# Schema Object Analysis: Tables

```
mysql> select alter_statement
    -> from   common_schema.sql_alter_table
    -> where table_schema = 'sakila';
+-------------------------------------------------+
| alter_statement                                 |
+-------------------------------------------------+
| ALTER TABLE `sakila`.`actor` ENGINE=InnoDB      |
| ALTER TABLE `sakila`.`address` ENGINE=InnoDB    |
| ALTER TABLE `sakila`.`category` ENGINE=InnoDB   |
| ALTER TABLE `sakila`.`city` ENGINE=InnoDB       |
| ALTER TABLE `sakila`.`country` ENGINE=InnoDB    |
| ALTER TABLE `sakila`.`customer` ENGINE=InnoDB   |
| ALTER TABLE `sakila`.`film` ENGINE=InnoDB       |
| ALTER TABLE `sakila`.`film_actor` ENGINE=InnoDB |
| ALTER TABLE `sakila`.`film_category` ENGINE=InnoDB |
| ALTER TABLE `sakila`.`film_text` ENGINE=MyISAM  |
| ALTER TABLE `sakila`.`inventory` ENGINE=InnoDB  |
| ALTER TABLE `sakila`.`language` ENGINE=InnoDB   |
| ALTER TABLE `sakila`.`payment` ENGINE=InnoDB    |
| ALTER TABLE `sakila`.`rental` ENGINE=InnoDB     |
| ALTER TABLE `sakila`.`staff` ENGINE=InnoDB      |
| ALTER TABLE `sakila`.`store` ENGINE=InnoDB      |
+-------------------------------------------------+
16 rows in set (0.00 sec)
```

# Schema Object Analysis: Tables

```
mysql> select alter_statement
    -> from   common_schema.sql_alter_table
    -> where table_schema = 'sakila';
+-------------------------------------------------+
| alter_statement                                 |
+-------------------------------------------------+
| ALTER TABLE `sakila`.`actor` ENGINE=InnoDB      |
| ALTER TABLE `sakila`.`address` ENGINE=InnoDB    |
| ALTER TABLE `sakila`.`category` ENGINE=InnoDB   |
| ALTER TABLE `sakila`.`city` ENGINE=InnoDB       |
| ALTER TABLE `sakila`.`country` ENGINE=InnoDB    |
| ALTER TABLE `sakila`.`customer` ENGINE=InnoDB   |
| ALTER TABLE `sakila`.`film` ENGINE=InnoDB       |
| ALTER TABLE `sakila`.`film_actor` ENGINE=InnoDB |
| ALTER TABLE `sakila`.`film_category` ENGINE=InnoDB |
| ALTER TABLE `sakila`.`film_text` ENGINE=MyISAM  |
| ALTER TABLE `sakila`.`inventory` ENGINE=InnoDB  |
| ALTER TABLE `sakila`.`language` ENGINE=InnoDB   |
| ALTER TABLE `sakila`.`payment` ENGINE=InnoDB    |
| ALTER TABLE `sakila`.`rental` ENGINE=InnoDB     |
| ALTER TABLE `sakila`.`staff` ENGINE=InnoDB      |
| ALTER TABLE `sakila`.`store` ENGINE=InnoDB      |
+-------------------------------------------------+
16 rows in set (0.00 sec)
```

# Schema Object Analysis: Columns

- Column overviews
  - `auto_increment_columns`
  - `text_columns`

# Schema Object Analysis: Auto Increment Columns

```
mysql> select table_name,
    ->        column_name,
    ->        data_type,
    ->        max_value,
    ->        auto_increment value,
    ->        auto_increment_ratio ratio
    -> from   auto_increment_columns
    -> where  table_schema = 'sakila';
+------------+--------------+-----------+------------+-------+--------+
| TABLE_NAME | COLUMN_NAME  | DATA_TYPE | max_value  | value | ratio  |
+------------+--------------+-----------+------------+-------+--------+
| actor      | actor_id     | smallint  |      65535 |   201 | 0.0031 |
| address    | address_id   | smallint  |      65535 |   606 | 0.0092 |
| category   | category_id  | tinyint   |        255 |    17 | 0.0667 |
| city       | city_id      | smallint  |      65535 |   601 | 0.0092 |
| country    | country_id   | smallint  |      65535 |   110 | 0.0017 |
| customer   | customer_id  | smallint  |      65535 |   600 | 0.0092 |
| film       | film_id      | smallint  |      65535 |  1001 | 0.0153 |
| inventory  | inventory_id | mediumint |   16777215 |  4582 | 0.0003 |
| language   | language_id  | tinyint   |        255 |     7 | 0.0275 |
| payment    | payment_id   | smallint  |      65535 | 16050 | 0.2449 |
| rental     | rental_id    | int       | 2147483647 | 16050 | 0.0000 |
| staff      | staff_id     | tinyint   |        255 |     3 | 0.0118 |
| store      | store_id     | tinyint   |        255 |     3 | 0.0118 |
+------------+--------------+-----------+------------+-------+--------+
13 rows in set (0.08 sec)
```

# Schema Object Analysis: Indexes

- Keys and Indexes
  - `candidate_keys`
  - `candidate_keys_recommended`
  - `no_pk_innodb_tables`
  - `rendundant_keys`

# Schema Object Analysis: Dependencies

- Dependency Routines
  - `get_event_dependencies(schema, name)`
  - `get_routine_dependencies(schema, name)`
  - `get_view_dependencies(schema, name)`
  - `get_sql_dependencies(sql, schema)`

# Schema Object Analysis: Dependencies

```
mysql> call get_view_dependencies('sakila', 'actor_info');
+-------------+---------------+-------------+--------+
| schema_name | object_name   | object_type | action |
+-------------+---------------+-------------+--------+
| sakila      | actor         | table       | select |
| sakila      | category      | table       | select |
| sakila      | film          | table       | select |
| sakila      | film_actor    | table       | select |
| sakila      | film_category | table       | select |
+-------------+---------------+-------------+--------+
5 rows in set (0.15 sec)


mysql> call get_routine_dependencies('sakila', 'rewards_report');
+-------------+-------------+-------------+--------+
| schema_name | object_name | object_type | action |
+-------------+-------------+-------------+--------+
| sakila      | customer    | table       | select |
| sakila      | payment     | table       | select |
| sakila      | tmpCustomer | table       | create |
| sakila      | tmpCustomer | table       | drop   |
| sakila      | tmpCustomer | table       | insert |
| sakila      | tmpCustomer | table       | select |
+-------------+-------------+-------------+--------+
6 rows in set (0.15 sec)
```

# Function Library

- Date/Time
  - `start_of_()` (hour, month, quarter, week, year)
  - `easter_day()`
- Text
  - `get_num_tokens(text, delim)`
  - `split_token()`
- Dynamic SQL
  - `exec(), exec_file(), exec_single()`

# Function Library: Date/Time

```
mysql> select now(),
    ->        start_of_hour(now()),
    ->        start_of_week(now()),
    ->        start_of_week_sunday(now()),
    ->        start_of_month(now()),
    ->        start_of_quarter(now()),
    ->        start_of_year(now()),
    ->        easter_day(now())
    -> \G
*************************** 1. row ***************************
                      now(): 2012-04-12 10:55:55
       start_of_hour(now()): 2012-04-12 10:00:00
       start_of_week(now()): 2012-04-09
start_of_week_sunday(now()): 2012-04-08
      start_of_month(now()): 2012-04-01
    start_of_quarter(now()): 2012-04-01
       start_of_year(now()): 2012-01-01
          easter_day(now()): 2012-04-08
1 row in set (0.00 sec)
```

# Scripting

- General
  - `eval(sql)`
  - `repeat_exec(interval, sql, condition)`
  - `foreach(collection, script)`
  - `$(collection, script)`

# Scripting: eval()

```
mysql> call eval('
    '>   select  concat(
    '>               \'create table test.\', table_name,
    '>               \' as select * from sakila.\', table_name
    '>           )
    '>   from    information_schema.tables
    '>   where   table_schema = \'sakila\'
    '> ');
Query OK, 0 rows affected (3.32 sec)

mysql> show tables in test;
+----------------------------+
| Tables_in_test             |
+----------------------------+
| actor                      |
. ...                        .
| store                      |
+----------------------------+
23 rows in set (0.00 sec)

mysql> call eval('
    '>   select  concat(\'drop table test.\', table_name)
    '>   from    information_schema.tables
    '>   where   table_schema = \'test\'
    '> ');
Query OK, 0 rows affected (1.41 sec)

mysql> show tables in test;
Empty set (0.00 sec)
```

# Scripting: foreach(range, script)

```
mysql> call $('1:3', 'select \'${1}\'');
+---+
| 1 |
+---+
| 1 |
+---+
1 row in set (0.19 sec)

+---+
| 2 |
+---+
| 2 |
+---+
1 row in set (0.34 sec)

+---+
| 3 |
+---+
| 3 |
+---+
1 row in set (0.50 sec)
```

# Scripting: foreach(multi-range, script)

```
mysql> call $('1:2,5:6', 'select \'${1}\', \'${2}\'');
+---+---+
| 1 | 5 |
+---+---+
| 1 | 5 |
+---+---+
1 row in set (0.20 sec)

+---+---+
| 1 | 6 |
+---+---+
| 1 | 6 |
+---+---+
1 row in set (0.39 sec)

+---+---+
| 2 | 5 |
+---+---+
| 2 | 5 |
+---+---+
1 row in set (0.57 sec)

+---+---+
| 2 | 6 |
+---+---+
| 2 | 6 |
+---+---+
1 row in set (0.73 sec)
```

# Scripting: foreach(select, script)

```
mysql> call $('select name from sakila.category', 'create table test.${1}(id int)');
mysql> show tables in test;
+----------------+
| Tables_in_test |
+----------------+
| Action         |
| Animation      |
| Children       |
| Classics       |
| Comedy         |
| Documentary    |
| Drama          |
| Family         |
| Foreign        |
| Games          |
| Horror         |
| Music          |
| New            |
| Sci-Fi         |
| Sports         |
| Travel         |
+----------------+
16 rows in set (0.01 sec)
```

# Scripting: foreach(table in schema, script)

```
mysql> call common_schema.$('table in test', 'drop table test.`${1}`');
Query OK, 0 rows affected (0.18 sec)

mysql> show tables in test;
Empty set (0.00 sec)
```

# Queryscript Language

- Execute
  - `run()`
  - `run_file()`

# How to Contribute

- Install and try it out
- Blog about it
- code.google.com/p/common-schema/issues/list

# Resources

- code.google.com/p/common-schema
- code.openark.org/blog/tag/common_schema