



ORACLE®

MySQL версии 5.5.3-m3: что нового?

Света Смирнова

Старший инженер технической поддержки SQL

Содержание

- Performance Schema для мониторинга производительности MySQL сервера
- Улучшения Partitioning, такие как:
 - Новые варианты разбиения таблиц
 - Возможность удаления всех строк одной partition
 - Поддержка Key Caches для partitioned MyISAM таблиц
- Новая встроенная функция TO_SECONDS()
- Поддержка полусинхронной репликации (Semi-synchronous replication)

Содержание

- 2 новых оператора SIGNAL И RESIGNAL для использования в хранимых процедурах: теперь можно возвращать ошибки
- Новая команда LOAD XML ... INFILE
- InnoDB Plugin теперь версии 1.0.6 - и это качество RC
- А также более 50 других улучшений

PERFORMANCE_SCHEMA

- Новая служебная база данных
- Предназначена для мониторинга производительности
- Содержит информацию о «событиях сервера», таких как вызов функции, системный вызов, этап выполнения SQL запроса, группа событий
- Сбор информации происходит для «instrumentation points», определённых в коде
- Подобна INFORMATION_SCHEMA, так как все её таблицы существуют только в памяти
- Табличный движок - PERFORMANCE_SCHEMA

PERFORMANCE_SCHEMA: таблицы

```
mysql> show tables;
+-----+
| Tables_in_performance_schema |
+-----+
| COND_INSTANCES |
| EVENTS_WAITS_CURRENT |
| EVENTS_WAITS_HISTORY |
| EVENTS_WAITS_HISTORY_LONG |
| EVENTS_WAITS_SUMMARY_BY_EVENT_NAME |
| EVENTS_WAITS_SUMMARY_BY_INSTANCE |
| EVENTS_WAITS_SUMMARY_BY_THREAD_BY_EVENT_NAME |
| FILE_INSTANCES |
```

PERFORMANCE_SCHEMA: таблицы

```
| FILE_SUMMARY_BY_EVENT_NAME |
| FILE_SUMMARY_BY_INSTANCE |
| MUTEX_INSTANCES |
| PERFORMANCE_TIMERS |
| PROCESSLIST |
| RWLOCK_INSTANCES |
| SETUP_CONSUMERS |
| SETUP_INSTRUMENTS |
| SETUP_OBJECTS |
| SETUP_TIMERS |
+-----+
18 rows in set (0.02 sec)
```

PERFORMANCE_SCHEMA: группы таблиц

- Таблицы в PERFORMANCE_SCHEMA могут быть сгруппированы по следующим категориям:
 - текущие события
 - EVENTS_WAITS_CURRENT
 - история событий
 - EVENTS_WAITS_HISTORY_*,
EVENTS_WAITS_SUMMARY_*, FILE_SUMMARY_*
 - информация о конфигурации
 - SETUP_*, *_INSTANCES, PERFORMANCE_TIMERS

PERFORMANCE_SCHEMA: пример

```
mysql> create table ps_ex1 (f1 varchar(13) CHARACTER
SET latin1 COLLATE latin1_bin DEFAULT NULL);
Query OK, 0 rows affected (0.02 sec)

mysql> insert into ps_ex1 values('a'), ('A'), ('B');
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> flush tables;
Query OK, 0 rows affected (0.62 sec)
```


PERFORMANCE_SCHEMA: пример

```
mysql> select * from FILE_INSTANCES
where FILE_NAME like '%ps\_ex1%';
```

FILE_NAME	EVENT_NAME	OPEN_COUNT
/.../ps_ex1.frm	wait/io/file/sql/FRM	0
/.../ps_ex1.MYI	wait/io/file/myisam/kfile	0
/.../ps_ex1.MYD	wait/io/file/myisam/dfile	0

```
3 rows in set (0.02 sec)
```

PERFORMANCE_SCHEMA: пример

```
mysql> alter table ps_ex1 modify f1 varchar(13)  
CHARACTER SET latin1 COLLATE latin1_general_ci;  
Query OK, 3 rows affected (0.04 sec)  
Records: 3 Duplicates: 0 Warnings: 0
```

PERFORMANCE_SCHEMA: пример

```
mysql> select * from FILE_INSTANCES  
where FILE_NAME like '%ps\_ex1%';
```

FILE_NAME	EVENT_NAME	OPEN_COUNT
/.../ps_ex1.frm	wait/io/file/sql/FRM	1
/.../ps_ex1.MYI	wait/io/file/myisam/kfile	1
/.../ps_ex1.MYD	wait/io/file/myisam/dfile	1

```
3 rows in set (0.02 sec)
```

PERFORMANCE_SCHEMA: таблицы подробнее

- EVENTS_WAITS_CURRENT

- Текущие события
- THREAD_ID – идентификатор thread
- EVENT_ID, EVENT_NAME – событие
- SOURCE – исходный файл где событие определено
- TIMER_START – время начала события
- TIMER_END – время завершения события
- TIMER_WAIT – время, за которое событие выполнялось
- SPINS – для mutex-ов: количество spin rounds
- OBJECT_* - объект, с которым произошло событие
- OPERATION – тип операции: lock, read, write
- NUMBER_OF_BYTES – для read и write: количество байт

PERFORMANCE_SCHEMA: таблицы подробнее

```
$mysql
Welcome to the MySQL monitor.  Commands end with ;
or \g.
Your MySQL connection id is 53285
Server version: 5.6.99-m4-debug Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear
the current input statement.

mysql> select sleep(100);
```

PERFORMANCE_SCHEMA: таблицы подробнее

```
mysql> select * from PROCESSLIST;
```

THREAD_ID	ID	NAME
0	0	thread/sql/main
53286	53285	thread/sql/one_connection
52755	52754	thread/sql/one_connection
1	0	thread/sql/signal_handler

```
4 rows in set (0.14 sec)
```

PERFORMANCE_SCHEMA: таблицы подробнее

```
mysql> select * from EVENTS_WAITS_CURRENT where  
THREAD_ID=53286\G  
***** 1. row *****  
      THREAD_ID: 53286  
      EVENT_ID: 42  
      EVENT_NAME:  
wait/synch/cond/sql/Item_func_sleep::cond  
      SOURCE: item_func.cc:3532  
      TIMER_START: 18444875154570862792  
      TIMER_END: NULL  
      TIMER_WAIT: NULL  
      SPINS: NULL
```

PERFORMANCE_SCHEMA: таблицы подробнее

```
OBJECT_SCHEMA: NULL
OBJECT_NAME: NULL
OBJECT_TYPE: NULL
OBJECT_INSTANCE_BEGIN: 9223372036854775807
NESTING_EVENT_ID: NULL
OPERATION: timed_wait
NUMBER_OF_BYTES: NULL
FLAGS: 0
1 row in set (0.02 sec)
```


PERFORMANCE_SCHEMA: таблицы подробнее

```
mysql> select sleep(100);
+-----+
| sleep(100) |
+-----+
|          0 |
+-----+
1 row in set (1 min 40.25 sec)
```

PERFORMANCE_SCHEMA: таблицы подробнее

```
mysql> select * from EVENTS_WAITS_CURRENT where  
THREAD_ID=53286\G
```

```
***** 1. row *****  
      THREAD_ID: 53286  
      EVENT_ID: 54  
      EVENT_NAME:  
wait/synch/mutex/sql/THD::LOCK_thd_data  
      SOURCE: sql_class.cc:3290  
      TIMER_START: 18444907424123927224  
      TIMER_END: 18444907424125198288  
      TIMER_WAIT: 1271064
```

PERFORMANCE_SCHEMA: таблицы подробнее

- История событий
 - EVENTS_WAITS_HISTORY
 - EVENTS_WAITS_HISTORY_LONG
- Структура обеих таблиц такая же как и EVENTS_WAITS_CURRENT, но они содержат информацию о прошедших событиях.
- Количество строк регулируется переменными
 - performance_schema_events_waits_history_size (default = 10)
 - performance_schema_events_waits_history_long_size (default = 10,000)

PERFORMANCE_SCHEMA: таблицы подробнее

- INSTANCE таблицы
 - COND_INSTANCES
 - NAME
 - OBJECT_INSTANCE_BEGIN
 - FILE_INSTANCES
 - FILE_NAME
 - EVENT_NAME
 - OPEN_COUNT
 - MUTEX_INSTANCES
 - NAME
 - OBJECT_INSTANCE_BEGIN
 - LOCKED_BY_THREAD_ID

PERFORMANCE_SCHEMA: таблицы подробнее

- INSTANCE таблицы
 - RWLOCK_INSTANCES
 - NAME, OBJECT_INSTANCE_BEGIN, WRITE_LOCKED_BY_THREAD_ID, READ_LOCKED_BY_COUNT
- Содержат информацию об объектах, использование которых PERFORMANCE_SCHEMA отслеживает в данный момент

PERFORMANCE_SCHEMA: таблицы подробнее

- Таблицы настроек
 - SETUP_CONSUMERS – какие события отслеживать
 - SETUP_INSTRUMENTS – классы объектов, для которых может быть собрана информация о событиях
 - SETUP_OBJECTS – не используется
 - SETUP_TIMERS – какой TIMER используется в настоящее время
- PERFORMANCE_TIMERS – типы таймеров
 - CYCLE, NANOSECOND, MICROSECOND, MILLISECOND, TICK
 - NULL для неподдерживаемых на текущей платформе
- PROCESSLIST – потоки, запущенные в данный момент

PERFORMANCE_SCHEMA: таблицы подробнее

- Сводные таблицы
 - EVENTS_WAITS_SUMMARY_BY_EVENT_NAME
 - EVENTS_WAITS_SUMMARY_BY_INSTANCE
 - EVENTS_WAITS_SUMMARY_BY_THREAD_BY_EVENT_NAME
 - FILE_SUMMARY_BY_EVENT_NAME
 - FILE_SUMMARY_BY_INSTANCE

PERFORMANCE_SCHEMA: таблицы подробнее

```
mysql> select * from
EVENTS_WAITS_SUMMARY_BY_EVENT_NAME where
COUNT_STAR>1 limit 1\G
***** 1. row *****
EVENT_NAME: wait/synch/mutex/sql/hash_filo::lock
COUNT_STAR: 53454
SUM_TIMER_WAIT: 30713104968
MIN_TIMER_WAIT: 295176
AVG_TIMER_WAIT: 574570
MAX_TIMER_WAIT: 863178960
1 row in set (0.00 sec)
```


PERFORMANCE_SCHEMA: таблицы подробнее

```
mysql> select * from
EVENTS_WAITS_SUMMARY_BY_THREAD_BY_EVENT_NAME where
THREAD_ID=53286 and COUNT_STAR>1 \G
***** 1. row *****
      THREAD_ID: 53286
      EVENT_NAME:
wait/synch/mutex/sql/LOCK_global_system_variables
      COUNT_STAR: 3
      SUM_TIMER_WAIT: 8192640
      MIN_TIMER_WAIT: 530112
      AVG_TIMER_WAIT: 2730880
      MAX_TIMER_WAIT: 4789080
```

PERFORMANCE_SCHEMA: таблицы подробнее

```
mysql> select * from
EVENTS_WAITS_SUMMARY_BY_INSTANCE where COUNT_STAR>1
limit 1 \G
***** 1. row *****
      EVENT_NAME:
wait/synch/mutex/mysys/THR_LOCK::mutex
OBJECT_INSTANCE_BEGIN: 7759404
      COUNT_STAR: 2
      SUM_TIMER_WAIT: 879504
      MIN_TIMER_WAIT: 433728
      AVG_TIMER_WAIT: 439752
      MAX_TIMER_WAIT: 445776
1 row in set (0.00 sec)
```

PERFORMANCE_SCHEMA: таблицы подробнее

```
mysql> select * from FILE_SUMMARY_BY_EVENT_NAME
where COUNT_READ > 1 limit 1\G
***** 1. row *****
      EVENT_NAME: wait/io/file/sql/ERRMSG
      COUNT_READ: 3
      COUNT_WRITE: 0
      SUM_NUMBER_OF_BYTES_READ: 40267
      SUM_NUMBER_OF_BYTES_WRITE: 0
1 row in set (0.00 sec)
```

PERFORMANCE_SCHEMA: таблицы подробнее

```
mysql> select * from FILE_SUMMARY_BY_INSTANCE where  
COUNT_WRITE > 1 limit 1\G  
***** 1. row *****  
          FILE_NAME:  
/Users/apple/Applications/mysql-next-  
mr/data/sbtest/#sql-4d43_5a.frm  
          EVENT_NAME: wait/io/file/sql/FRM  
          COUNT_READ: 14  
          COUNT_WRITE: 22  
          SUM_NUMBER_OF_BYTES_READ: 2256  
          SUM_NUMBER_OF_BYTES_WRITE: 5224  
1 row in set (0.09 sec)
```

PERFORMANCE_SCHEMA: код

- `static PSI_mutex_info all_server_mutexes[] =`
 - `{`
 - `#ifdef HAVE_MMAP`
 - `{ &key_PAGE_lock, "PAGE::lock", 0},`
 - `{ &key_LOCK_sync, "TC_LOG_MMAP::LOCK_sync", 0},`
 - `...`
- `static PSI_rwlock_info all_server_rwlockes[]`
- `static PSI_cond_info all_server_conds[]`
- `static PSI_thread_info all_server_threads[]`
- `static PSI_file_info all_server_files[]`

PERFORMANCE_SCHEMA: код

- Использование:
 - `mysql_mutex_init(key_LOCK_open, &LOCK_open, MY_MUTEX_INIT_FAST);`
- В настоящее время определены только для сервера
- Производители табличных движков должны сами заботиться о поддержке PERFORMANCE_SCHEMA при помощи поддерживаемого инструментария

PERFORMANCE_SCHEMA: дополнительно

- <http://forge.mysql.com/worklog/task.php?id=2360>
- <http://forge.mysql.com/worklog/task.php?id=4513>
- <http://dev.mysql.com/doc/performance-schema/en/index.html>
- <http://marcalff.blogspot.com/>
- http://forge.mysql.com/wiki/Performance_Schema%3A_Instrumenting_Code
- <http://www.markleith.co.uk/?p=112>
- <http://en.oreilly.com/mysql2010/public/schedule/detail/13300>

Улучшения Partitioning

- Новые варианты разбиения таблиц
- Возможность удаления всех строк одной partition
- Поддержка Key Caches для partitioned MyISAM таблиц

Улучшения Partitioning: новые разбиения

- RANGE COLUMNS
 - аналогично RANGE, но поддерживает несколько столбцов
 - не поддерживает выражений
 - поддерживает строки, а также DATE и DATETIME
- LIST COLUMNS
 - аналогично LIST, но поддерживает несколько столбцов
 - не поддерживает выражений
 - поддерживает строки, а также DATE и DATETIME

Улучшения Partitioning: RANGE COLUMNS

```
mysql> create table t1(  
  -> a INT,  
  -> b VARCHAR(255)  
  -> ) PARTITION BY RANGE COLUMNS (a,b) (  
  -> PARTITION p0 VALUES LESS THAN (5, 'G'),  
  -> PARTITION p1 VALUES LESS THAN (MAXVALUE,  
MAXVALUE)  
  -> );  
Query OK, 0 rows affected (0.40 sec)
```

Улучшения Partitioning: RANGE COLUMNS

```
mysql> INSERT INTO t1 VALUES(1, 'Ann');
```

```
Query OK, 1 row affected (0.14 sec)
```

```
mysql> INSERT INTO t1 VALUES(6, 'Ann');
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO t1 VALUES(6, 'Bob');
```

```
Query OK, 1 row affected (0.00 sec)
```

Улучшения Partitioning: RANGE COLUMNS

```
mysql> SELECT PARTITION_NAME, TABLE_ROWS  
-> FROM INFORMATION_SCHEMA.PARTITIONS  
-> WHERE TABLE_NAME = 't1';
```

PARTITION_NAME	TABLE_ROWS
p0	1
p1	2

```
2 rows in set (0.36 sec)
```

Улучшения Partitioning: LIST COLUMNS

```
CREATE TABLE t1 (a int, b int, c int)
PARTITION BY LIST COLUMNS (a,b,c)
(PARTITION p0 VALUES IN
(("a1", "b1", "c1"), ("a2", "b2", "c2")));
```

Улучшения Partitioning: удаление

- Новый синтаксис ALTER TABLE ... TRUNCATE PARTITION
- Позволяет удалить все строки одной partition
- Не работает с subpartitions
- Примеры
 - ALTER TABLE t1 TRUNCATE PARTITION p0;
 - ALTER TABLE t1 TRUNCATE PARTITION p0, p1, p5;
 - ALTER TABLE t1 TRUNCATE PARTITION ALL;

Улучшения Partitioning: Key Caches

- Теперь возможно использовать Key Caches для отдельных partition partitioned MyISAM таблиц
- Новый синтаксис LOAD INDEX INTO CACHE ... PARTITION ...
- Примеры
 - LOAD INDEX INTO CACHE t1 PARTITION (p1);
 - LOAD INDEX INTO CACHE t1 PARTITION (p1,p0);
 - LOAD INDEX INTO CACHE t1 PARTITION (p1) INDEX (`PRIMARY`);
 - LOAD INDEX INTO CACHE t1 PARTITION (ALL);

Улучшения Partitioning: дополнительно

- <http://dev.mysql.com/doc/refman/5.5/en/partitioning-columns.html>
- <http://dev.mysql.com/doc/refman/5.5/en/alter-table.html>
- <http://dev.mysql.com/doc/refman/5.5/en/load-index.html>
- <http://forge.mysql.com/worklog/task.php?id=3352>
- <http://forge.mysql.com/worklog/task.php?id=4444>
- <http://forge.mysql.com/worklog/task.php?id=4571>

TO_SECONDS()

- Возвращает количество секунд с года 0 до аргумента DATE или DATETIME
- Не предсказуемые результаты для дат ранее 1582 года – года введения Грегорианского календаря
- Может использоваться в PARTITIONING
- <http://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.htm>
- <http://forge.mysql.com/worklog/task.php?id=3352>

TO_SECONDS(): пример

```
mysql> SELECT TO_SECONDS(NOW()), TO_SECONDS('0000-00-00');
```

TO_SECONDS(NOW())	TO_SECONDS('0000-00-00')
63438493989	NULL

1 row in set, 1 warning (0.00 sec)

Warning (Code 1292): Incorrect datetime value:
'0000-00-00'

Semi-synchronous replication

- Встроенная репликация в MySQL – асинхронная
- Semi-synchronous репликация работает следующим образом:
 - соответствующая опция должна быть включена на мастере и хотя бы на одном слэйве
 - `rpl_semi_sync_master_enabled`
 - `rpl_semi_sync_slave_enabled`
 - после того как транзакция завершилась на мастере он блокируется до получения ответа с одного слэйва
 - если `timeout` наступает раньше получения ответа со слэйва мастер возвращается к асинхронной репликации
 - если слэйв опять начинает работать semi-synchronous репликация используется снова

Semi-synchronous replication

- Первоначально – mysql-google-tools
- Расширение – вы должны установить plugin
- Не полностью синхронная: ждёт ответа хотя бы от одного слэйва
- Использование semi-synchronous репликации повышает уверенность в целостности данных
- Возможно нарушение целостности только в случае падения мастера в тот момент когда он ожидает ответа со слэйва
- Замедляет производительность DML, потому что ждёт ответа со слэйва. Напрямую зависит от скорости соединения.

Semi-synchronous replication: настройка

- На мастере
 - `INSTALL PLUGIN rpl_semi_sync_master SONAME 'semisync_master.so';`
 - `SET GLOBAL rpl_semi_sync_master_enabled = 1;`
- На слэйве
 - `INSTALL PLUGIN rpl_semi_sync_slave SONAME 'semisync_slave.so';`
 - `SET GLOBAL rpl_semi_sync_slave_enabled = 1;`
 - `STOP SLAVE IO_THREAD;`
 - `START SLAVE IO_THREAD;`

Semi-synchronous replication: мониторинг

На слэйве

```
mysql> SHOW STATUS LIKE 'Rpl_semi_sync%';
```

Variable_name	Value
Rpl_semi_sync_slave_status	ON

```
1 row in set (0.08 sec)
```

Semi-synchronous replication: МОНИТОРИНГ

На мастере

```
mysql> SHOW STATUS LIKE 'Rpl_semi_sync%';
```

Variable_name	Value
Rpl_semi_sync_master_clients	1
Rpl_semi_sync_master_net_avg_wait_time	513
Rpl_semi_sync_master_net_wait_time	2054
Rpl_semi_sync_master_net_waits	4
Rpl_semi_sync_master_no_times	0

Semi-synchronous replication: МОНИТОРИНГ

На мастере

Rpl_semi_sync_master_no_tx	0	
Rpl_semi_sync_master_status	ON	
Rpl_semi_sync_master_timefunc_failures	0	
Rpl_semi_sync_master_tx_avg_wait_time	802	
Rpl_semi_sync_master_tx_wait_time	3209	
Rpl_semi_sync_master_tx_waits	4	
Rpl_semi_sync_master_wait_pos_backtraverse	0	
Rpl_semi_sync_master_wait_sessions	0	
Rpl_semi_sync_master_yes_tx	4	

Semi-synchronous replication: дополнительно

- <http://dev.mysql.com/doc/refman/5.5/en/replication-semisync.html>
- <http://forge.mysql.com/worklog/task.php?id=4398>
- <http://forge.mysql.com/worklog/task.php?id=1720>
- <http://code.google.com/p/google-mysql-tools/wiki/SemiSyncReplicationDesign>
- <http://www.mysqlconf.com/mysql2009/public/schedule/detail/6871>

SIGNAL И RESIGNAL

- SIGNAL
 - способ вернуть ошибку из SQL приложения
 - используйте
 - 01* для предупреждений
 - 02* для not found
 - >02 для исключений
 - SQLSTATE '45000' для generic errors
- Структура соответствует SQL стандарту с дополнением: MYSQL_ERRNO

SIGNAL И RESIGNAL

- RESIGNAL
 - подобен SIGNAL, но используется внутри обработчика события
 - позволяет одновременно обработать ошибку и передать её дальше
 - может изменить передаваемую информацию
 - структура аналогична SIGNAL

SIGNAL И RESIGNAL: структура

CLASS_ORIGIN	VARCHAR(64)
SUBCLASS_ORIGIN	VARCHAR(64)
CONSTRAINT_CATALOG	VARCHAR(64)
CONSTRAINT_SCHEMA	VARCHAR(64)
CONSTRAINT_NAME	VARCHAR(64)
CATALOG_NAME	VARCHAR(64)
SCHEMA_NAME	VARCHAR(64)
TABLE_NAME	VARCHAR(64)
COLUMN_NAME	VARCHAR(64)
CURSOR_NAME	VARCHAR(64)
MESSAGE_TEXT	VARCHAR(128)
MYSQL_ERRNO	SMALLINT UNSIGNED

SIGNAL И RESIGNAL: пример

```
mysql> CREATE TABLE t1 (  
  -> id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  -> procent TINYINT UNSIGNED  
  -> );  
Query OK, 0 rows affected (0.03 sec)  
  
mysql> DELIMITER |
```

SIGNAL И RESIGNAL: пример

```
mysql> CREATE TRIGGER t1_bi BEFORE INSERT ON t1
-> FOR EACH ROW BEGIN
-> IF NEW.procent > 100 THEN
-> SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =
"Invalid value for procent field";
-> END IF;
-> END
-> |
```

Query OK, 0 rows affected (0.18 sec)

```
mysql> DELIMITER ;
```

SIGNAL И RESIGNAL: пример

```
mysql> INSERT INTO t1 (procent) VALUES (10);  
Query OK, 1 row affected (0.17 sec)  
  
mysql> INSERT INTO t1 (procent) VALUES (100);  
Query OK, 1 row affected (0.03 sec)  
  
mysql> INSERT INTO t1 (procent) VALUES (101);  
ERROR 1644 (45000): Invalid value for procesnt field
```

SIGNAL И RESIGNAL: пример 2

```
mysql> SET SQL_MODE='ERROR_FOR_DIVISION_BY_ZERO';  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> SELECT 5/0;
```

```
+-----+
```

```
| 5/0 |
```

```
+-----+
```

```
| NULL |
```

```
+-----+
```

```
1 row in set, 1 warning (0.00 sec)
```


SIGNAL И RESIGNAL: пример 2

```
mysql> DROP FUNCTION IF EXISTS safe_div;
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER |

mysql> CREATE FUNCTION safe_div(divident INT,
    divider INT) RETURNS INT
    -> BEGIN
    -> DECLARE CONTINUE HANDLER FOR SQLSTATE '22012'
    -> BEGIN
```

SIGNAL И RESIGNAL: пример 2

```
-> RESIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =  
"Division by 0", MYSQL_ERRNO = 1365;  
-> END;  
-> RETURN dividend/divider;  
-> END  
-> |
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> DELIMITER ;
```

SIGNAL И RESIGNAL: пример 2

```
mysql> SELECT safe_div(4, 2);
```

```
+-----+  
| safe_div(4, 2) |
```

```
+-----+  
|          2 |
```

```
+-----+  
1 row in set (0.01 sec)
```

SIGNAL И RESIGNAL: пример 2

```
mysql> SELECT safe_div(4, 2);
+-----+
| safe_div(4, 2) |
+-----+
|          2 |
+-----+
1 row in set (0.01 sec)

mysql> SELECT safe_div(5, 0);
ERROR 1365 (45000): Division by 0
```

SIGNAL И RESIGNAL: дополнительно

- <http://dev.mysql.com/doc/refman/5.5/en/signal-resigna>
- <http://forge.mysql.com/worklog/task.php?id=751>
- <http://forge.mysql.com/worklog/task.php?id=2110>
- <http://forge.mysql.com/worklog/task.php?id=2265>

LOAD XML ... INFILE

- Загружает XML файлы в таблицу
- Дополнение к mysql –xml
- Поддерживает 3 формата входных данных:
 - `<row column1="value1" column2="value2" .../>`
 - `<row><column1>value1</column1>
<column2>value2</column2></row>`
 - `<row><field name='column1'>value1</field><field
name='column2'>value2</field></row>`
- Остальные опции аналогично LOAD DATA INFILE
- <http://dev.mysql.com/doc/refman/5.5/en/load-xml.html>

Новые версии InnoDB Plugin

- Вышло 3 новых версии
 - 5.5.3-m3 – с InnoDB Plugin 1.0.6 – RC
 - 5.1.46 – с InnoDB Plugin 1.0.7 - GA
 - 5.5.4-m4 – с InnoDB Plugin 1.1 – Early Adopter

Новые версии InnoDB Plugin: версия 1.0.*

- Динамический plugin
 - Версия по умолчанию начиная с 5.4
- Новый формат файла Barracuda
- Возможность компрессии
- Улучшения производительности
- Создание индексов online
 - больше не нужно копировать таблицу
 - существуют ограничения
- TRUNCATE TABLE возвращает место на диске
- Новые таблицы в INFORMATION SCHEMA
- http://www.innodb.com/doc/innodb_plugin-1.0/

Новые версии InnoDB Plugin: версия 1.1

- Начиная с версии 5.5 – это табличный движок по умолчанию (а не MyISAM как было раньше)
- Поддержка встроенного AIO на Linux (libaio required)
- InnoDB recovery значительно быстрее
- Полная поддержка PERFORMANCE SCHEMA
 - Mutex
 - RWLOCKS
 - File I/O
 - Thread
- Несколько инстанций Buffer Pool

Новые версии InnoDB Plugin: версия 1.1

- Несколько Rollback сегментов
- Улучшенный Change Buffering для буферизации delete и purge
- Улучшенный Purge Scheduling
- Улучшенный Log Sys mutex
- Отдельный Flush List mutex
- Прирост производительности по сравнению с 1.0
 - 188% для read-only
 - 164% для read/write

Новые версии InnoDB Plugin: дополнительно

- <http://dev.mysql.com/doc/innodb-plugin/1.1/en/index.html>
- <http://blogs.innodb.com/wp/>
- <http://dimitrik.free.fr/blog/archives/2010/04/mysql-performance-554-dbstress.html>
- <http://mikaelronstrom.blogspot.com/>

Другие улучшения

- Number-to-string conversions
 - теперь используют информацию о кодировке
 - <http://forge.mysql.com/worklog/task.php?id=2649>

Другие улучшения

- Number-to-string conversions
 - теперь используют информацию о кодировке
 - <http://forge.mysql.com/worklog/task.php?id=2649>

```
mysql-5.1> SELECT @a:=concat(10, ' days'),  
                upper(@a), charset(@a);
```

```
+-----+-----+-----+  
| @a:=concat(10, ' days') | upper(@a) | charset(@a) |  
+-----+-----+-----+  
| 10 days                | 10 days   | binary      |  
+-----+-----+-----+  
1 row in set (0.04 sec)
```

Другие улучшения

- Number-to-string conversions
 - теперь используют информацию о кодировке
 - <http://forge.mysql.com/worklog/task.php?id=2649>

```
mysql-5.5.3-m3> SELECT @a:=concat(10, ' days'),
  upper(@a), charset(@a);
+-----+-----+-----+
| @a:=concat(10, ' days') | upper(@a) | charset(@a) |
+-----+-----+-----+
| 10 days                  | 10 DAYS   | utf8        |
+-----+-----+-----+
1 row in set (0.04 sec)
```

Другие улучшения

- Новые таблицы в INFORMATION SCHEMA
 - ROUTINES
 - PARAMETERS

```
mysql> SELECT ROUTINE_NAME,ROUTINE_BODY,  
ROUTINE_DEFINITION FROM  
INFORMATION_SCHEMA.ROUTINES\G  
***** 1. row *****  
ROUTINE_NAME: p1  
ROUTINE_BODY: SQL  
ROUTINE_DEFINITION: BEGIN  
SET v1=10;  
END
```

Другие улучшения

- 4-byte UTF8, UTF16 и UTF32
 - новая кодировка utf8mb4
 - <http://forge.mysql.com/worklog/task.php?id=1213>
 - <http://www.collation-charts.org/>

```
mysql> SELECT HEX(' *');
+-----+
| HEX(' *') |
+-----+
| F0A080A1  |
+-----+
1 row in set (0.00 sec)
```


Другие улучшения

- Выбор языка для отображения ошибок «на лету»
 - <http://forge.mysql.com/worklog/task.php?id=751>

```
mysql> SET lc_messages=ja_JP;
Query OK, 0 rows affected (0.00 sec)
mysql> CREATE DATABASE mysql;
ERROR 1007 (HY000): 'mysql' データベースが作れません . 既にそのデータベースが存在します
mysql> SET lc_messages=ru_RU;
Query OK, 0 rows affected (0.00 sec)
mysql> CREATE DATABASE mysql;
ERROR 1007 (HY000): Невозможно создать базу данных 'mysql'. База данных уже существует
```

Другие улучшения

- Международный формат для чисел

```
mysql> SELECT FORMAT(12345.67, 2, 'en_US') as en_US,  
        FORMAT(12345.67, 2, 'de_DE') as de_DE,  
        FORMAT(12345.67, 2, 'ru_RU') as ru_RU;  
+-----+-----+-----+  
| en_US   | de_DE   | ru_RU   |  
+-----+-----+-----+  
| 12,345.67 | 12.345,67 | 12 345,67 |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

Другие улучшения

- Replication
 - `SHOW RELAYLOG EVENTS` – аналогично `SHOW BINLOG EVENTS`, но для relay логов
 - Новая переменная `relay_log_recovery` – отбросить relay log, который содержит записи после падения мастера
 - Новая переменная `sync_relay_log=N` – flush relay log после каждых N событий
 - Новая переменная `IGNORE_SERVER_IDS` в дополнение к опции `replicate-same-server-id`
 - Новая переменная `–binlog-direct-non-transactional-updates:` при её использовании изменения нетранзакционных таблиц будут писаться непосредственно в binary log, а не transaction cache

Другие улучшения

- Performance

- Google tcmalloc на Linux
 - <http://code.google.com/p/google-perftools/>
- InnoDB buffer pool разделён на 2 сублиста
- Улучшение производительности встроенных функций
- Увеличение масштабируемости: лучше работает при большом количестве потоков

- Несовместимые изменения

- Много переменных, которые раньше были помечены как deprecated сейчас удалены. Прочитайте Change Log!
- <http://dev.mysql.com/doc/refman/5.5/en/news-5-5-3.html>

Другие улучшения


- mysqldump может сделать dump INFORMATION SCHEMA
 - mysqldump --skip-lock-tables information_schema
- Новые collations: utf8_sinhala_ci и ucs2_sinhala_ci
- Default charset для клиентов теперь LANG или LC_ALL операционной системы, а не latin1
- TRADITIONAL SQL включает NO_ENGINE_SUBSTITUTION
- mysql_upgrade опция --upgrade-system-tables
- Поддержка UNIX_TIMESTAMP() для partitioning

Дополнительно

- <http://dev.mysql.com/doc/refman/5.5/en/news-5-5-x.html>
- <http://en.oreilly.com/mysql2010/public/schedule/proceedings>
- <http://planet.mysql.com/>
- <http://ru.planet.mysql.com/>



?



СПАСИБО!