Appendix C: Other MySQL Reference Data

MySQL information_schema tables and columns

The SQL Standard (ISO/IEC 9075) defines `information_schema` (IS) as a virtual database describing all server databases and their objects, queryable like any other database though not user-modifiable. In MySQL, for many MySQL SHOW commands, IS queries are more flexible for fetching database metadata, but in MySQL 5.x have been slow; 8.0 speeds them up by re-implementing some IS tables (events, parameters, routines, triggers) as Views on `mysql` tables.

MySQL first implemented IS in version 5.0.3. Each subsequent MySQL release has expanded the implementation, *e.g.*, from 5.0 to 5.1, the number of IS tables nearly doubled:

MySQL 5.0 information_schema tables

Character_Sets
Collations
Collation_Character_Set_Applicability
Columns
Column_Privileges
Key_Column_Usage
Profiling
Routines
Schemata
Schema_Privileges
Statistics
Tables
Table_Constraints
Table_Privileges
Triggers
User_Privileges
Views

MySQL 5.1 information_schema tables

Character_Sets
Collations
Collation_Character_Set_Applicability
Columns
Column_Privileges
Engines
Events
Files
Global_Status
Global_Variables
Key_Column_Usage
Partitions
Plugins
Processlist
Profiling
Referential_Constraints
Routines
Schemata
Schema_Privileges
Session_Status
Session_Variables
Statistics
Tables
Table_Constraints
Table_Privileges
Triggers
User_Privileges
Views

And, `information_schema` and its cousin, `performance_schema` (PS),continue to evolve, for example in MySQL 5.7 the global_variables and global_status tables moved from IS to PS.

Because IS is queryable, you can study it in any MySQL GUI query interface, for example *TheUsual* or *MySQL WorkBench*. Now with five different MySQL IS versions of available, we no longer have room to reproduce all versions in a book of this size, but if you want such a listing, it's dead easy to use `information_schema` to document itself: run this query in *TheUsual* or in *MySQL Workbench*, or in whatever is your favourite GUI tool for MySQL queries:

```
SELECT
  LOWER(table_name) AS 'Table',
  LOWER(column_name) AS 'Column',
  ordinal_position As 'Ordinal Position',
  IF(column_default='',' ',column_default) AS 'Default',
  UPPER(data_type) AS 'Type',
  character_maximum_length AS 'Max Len',
  character_octet_length AS 'Octet Len',
  numeric_precision AS 'Precision',
  numeric_scale AS 'Numeric Scale',
  is_nullable AS 'Null',
  character_set_name AS 'Char Set',
  collation_name AS 'Collation'
FROM information_schema.columns
WHERE table_schema = 'information_schema'
ORDER BY table_name,'Ordinal Position';
```

To generate your own self-standing `information_schema` documentation in an HTML file, run the *mysql* client program with the arguments shown, substituting your MySQL username and password for `USR` and `PWD` …

```
mysql -uUSR -pPWD --tee=../scripts/iscols.html --html
```

and in the *mysql* client program run this version of the query:

```
SELECT
  LOWER(table_name) AS 'Table',
  LOWER(column_name) AS 'Column',
  ordinal_position As 'Ordinal Position',
  IF(column_default='',' ',column_default) AS 'Default',
  UPPER(data_type) AS 'Type',
  character_maximum_length AS 'Max Len',
  character_octet_length AS 'Octet Len',
  numeric_precision AS 'Precision',
  numeric_scale AS 'Numeric Scale',
  is_nullable AS 'Null',
  character_set_name AS 'Char Set',
  collation_name AS 'Collation',
  column_key AS 'Key',
  extra AS 'Extra',
  privileges As 'Privileges',
  column_comment As 'Comment'
FROM information_schema.columns
WHERE table_schema = 'information_schema'
ORDER BY table_name,'Ordinal Position';
```

After this runs, the result is in `scripts/iscols.html`.

INNODB tracking tables

As of 5.7, the INNODB engine (Chapters 3, 7) has 30 `information_schema` tables to track INNODB activity. INNODB_CMP, INNODB_CMP_RESET track compression actions and status; `page_size` tracks compressed page size; reading from INNODB_CMP_RESET resets compression statistics so the next read retrieves statistics on actions since the last read. INNODB_CMPMEM, INNODB_CMPMEM_RESET track buffer pool compression status; Each row tracks compression actions for one page size. Reading from INNODB_CMPMEM_RESET resets buffer pool compression statistics so the next read retrieves statistics on actions since the last read. INNODB_TRX tracks current transactions. including whether the transaction is waiting for a lock, when the transaction started, and the SQL statement. For each lock blocked by another transaction, INNODB_LOCKS has rows tracking the blocked and blocking transaction; The mode of a blocking lock always differs from the mode of the blocked lock (e.g., shared/exclusive, read/write). INNODB_LOCK_WAITS has one row or multiple rows for each blocked transaction, identifying the requested and blocking locks. Use INNODB_METRICS to query low-level InnoDB performance, using the new system variables `innodb_monitor_enable, innodb_monitor_disable, innodb_monitor_reset,` and `innodb_monitor_reset_all.` INNODB_BUFFER_PAGE, INNODB_BUFFER_PAGE_LRU, and INNODB_BUFFER_POOL_STATS display InnoDB buffer pool info, useful on a high-performance system with much memory.

Performance_schema

MySQL 5.5 introduced `performance_schema`, a monitoring engine and virtual database for tracking server performance. See the section on it in *Chapter 7* for an introduction. Like `information_schema`, it has grown too large for detailed listing in a book of this size, but again you can use `information_schema` to generate your own HTML `performance_schema` HTML table: run the *mysql* client program with the arguments shown, substituting your MySQL username and password for `USR` and `PWD` …

```
mysql -uUSR -pPWD --tee=../scripts/pscols.html --html
```

and in the *mysql* client program execute this query:

```
SELECT
  LOWER(table_name) AS 'Table',
  LOWER(column_name) AS 'Column',
  ordinal_position As 'Ordinal Position',
  IF(column_default='',' ',column_default) AS 'Default',
  UPPER(data_type) AS 'Type',
  character_maximum_length AS 'Max Len',
  character_octet_length AS 'Octet Len',
  numeric_precision AS 'Precision',
  numeric_scale AS 'Numeric Scale',
  is_nullable AS 'Null',
```

```
  character_set_name AS 'Char Set',
  collation_name AS 'Collation',
  column_key AS 'Key',
  extra AS 'Extra',
  privileges As 'Privileges',
  column_comment As 'Comment'
FROM information_schema.columns
WHERE table_schema = 'performance_schema'
ORDER BY table_name,'Ordinal Position';
```

After this runs, the content of your MySQL version's `performance_schema` is in `scripts/pscols.html`. See *here* for a tabulation of `performance_schema` options and system variables in MySQL 5.7.

Server information returned by SHOW STATUS

This command lists server status variables for the current connection (default or LOCAL) or for all connections (GLOBAL). The latest lists of SHOW STATUS variables can be found at *dev.mysql.com/doc/refman/51/en/server-status-variables.html* for 5.1, at *dev.mysql.com/doc/refman/5.5/en/server-status-variables.html* for 5.5, *http://dev.mysql.com/doc/refman/5.6/en/server-status-variables.html* for 5.6. We omit variables marked for internal use only.

Table C-5: Server Status Variables Returned by SHOW STATUS

Variable	Meaning
Aborted_clients	No. of connections aborted because client did not close connection properly
Aborted_connects	No. of failed MySQL server connect attempts
Binlog_cache_disk_use	No. of transactions overrunning `binlog_cache_size` needing a temp file. Since 4.1.2.
Binlog_cache_use	No. of transactions that used the temporary binary log cache. Since 4.1.2.
Bytes_received	No. of bytes received from all clients
Bytes_sent	No. of bytes sent to all clients
Compression	Does client connection use compression? Since 5.0.16.
Com_*	One row for each * command category--more than 100 of them, each reporting the No. of times a given command has executed. Display them all with SHOW STATUS LIKE 'com%'.
Connection_errors_accept	No of accept() errors
Connection_errors_internal	No of internal errors
Connection_errors_max_connections	No of max_connections errors
Connection_errors_peer_addr	No of errors searching for client peer addresses
Connection_errors_select	No of select() errors
Connection_errors_tycpwrap	No of libwrap library errors
Connections	No. of MySQL server connection attempts
Created_tmp_disk_tables	No. of implicit temporary tables created while executing statements: if big, tmp_table_size may be too small.
Created_tmp_files	No. of temp files server has created

Variable	Meaning
Created_tmp_tables	No. of undeclared temporary tables created while executing statements.
Delayed_errors	No. of INSERT DELAYED row writes that threw errors
Delayed_insert_threads	No. of current INSERT DELAYED thread handlers
Delayed_writes	No. of INSERT DELAYED rws written
Flush_commands	No. of executed FLUSH commands
Handler_commit	No. of internal COMMIT commands
Handler_delete	No. of DELETE requests
Handler_discover	No. of times the server has discovered a named table by querying the NDB Cluster storage engine. Since 4.1.2.
Handler_read_first	No. of times an index first entry was read. Higher with more full index scans.
Handler_read_key	No. of key-based requests to read a row. High if queries use indexes well.
Handler_read_last	No. of requests to read last key in index. Since 5.5.7. Higher with more ORDER BY ... DESC queries.
Handler_read_next	No. of requests to read next row in key order. Higher with query constraints and index scans.
Handler_read_prev	No. of requests to read previous row in key order. Mainly used to optimise ORDER BY ... DESC
Handler_read_rnd	No. of read requests based on fixed row position
Handler_read_rnd_next	No. of read requests for next row based on fixed position. High if there are many whole-table scans and sorts. Proper index use reduces this number.
Handler_rollback	No.mber of internal ROLLBACK commands
Handler_update	No. of update requests
Handler_write	No. of insert requests
Innodb_available_undo_logs	No. of availaable undo storage logs
Innodb_buffer_pool_bytes_data	No. of bytes in the InnoDB buffer pool containing data. Since 5.6.
Innodb_buffer_pool_bytes_dirty	No. of bytes in dirty pages of InnoDB buffer pool. Since 5.6.
Innodb_buffer_pool_dump_status	Progress of recording buffer pool pages triggered by setting innodb_buffer_pool_dump_at_shutdown or innodb_buffer_pool_dump_now
Innodb_buffer_pool_load_status	Progress of warming up buffer pool pages triggered by setting innodb_buffer_pool_dump_at_shutdown or innodb_buffer_pool_dump_now
Innodb_buffer_pool_pages_data	No. of InnoDB buffer pool pages containing data. Since 5.0.2.
Innodb_buffer_pool_pages_dirty	No. of dirty InnoDB buffer pool pages. Since 5.0.2
Innodb_buffer_pool_pages_flushed	No. of InnoDB buffer pool pages requested to be flushed. Since 5.0.2.
Innodb_buffer_pool_pages_free	No. of free buffer pool pages. Since 5.0.2.
Innodb_buffer_pool_pages_latched	No. of InnoDB buffer pool latched (unflushable) pages. Since 5.0.2.
Innodb_buffer_pool_pages_misc	No. of InnoDB buffer pool pages allocated for administrative overhead, eg row locks. Since 5.0.2.
Innodb_buffer_pool_pages_total	No. of pages in InnoDB buffer pool. Since 5.0.2.
Innodb_buffer_pool_read_ahead	No. of pages read into the InnoDB buffer pool by the read-ahead background thread
Innodb_buffer_pool_read-ahead_evicted	No. of read-ahead pages evicted without being accessed by queries
Innodb_buffer_pool_read_requests	No. of logical read requests InnoDB has done. Since 5.0.2.
Innodb_buffer_pool_reads	No. of logical InnoDB reads requiring a single-page read. Since 5.0.2.
Innodb_buffer_pool_read_ahead_seq	No. of InnoDB sequential read-aheads, sequential full table scans. Removed 5.5
Innodb_buffer_pool_read_ahead_rnd	No. of InnoDB random read-aheads InnoDB required by table scan. Remnoved 5.5.
Innodb_buffer_pool_write_requests	No. of writes one to the InnoDB buffer pool. Since 5.0.2.
Innodb_buffer_pool_wait_free	No. of times InnoDB had to wait for a page flush. Small if buffer pool size is correctly set. Since 5.0.2.

Variable	Meaning
Innodb_buffer_pool_write_requests	No. of InnoDB buffer pool writes
Innodb_data_read	No. of bytes of InnoDB data read so far by InnoDB. Since 5.0.2.
Innodb_data_reads	No. of InnoDB data reads so far. Since 5.0.2.
Innodb_data_written	No. of InnoDB bytes written so far. Since 5.0.2.
Innodb_data_fsyncs	No. of InnoDB fsync()s so far. Since 5.0.2.
Innodb_data_pending_fsyncs	Current no. of pending InnoDB fsync() operations. Since 5.0.2.
Innodb_data_pending_reads	Current no. of pending InnoDB reads. Since 5.0.2.
Innodb_data_pending_writes	Current no. of pending InnoDB writes. Since 5.0.2.
Innodb_dblwr_writes	No. of doublewrite InnoDB writes so far. Since 5.0.2.
Innodb_dblwr_pages_written	No. of doublewrite InnoDB pages so far. Since 5.0.2.
Innodb_have_atomic_builtins	InnoDB engine built with atomic instructions for new thread concurrency? 5.5
Innodb_heap_enabled	Use InnoDB memory manager? 5.4.0.
Innodb_log_waits	No. of InnoDB waits for buffer flushing. Since 5.0.2.
Innodb_log_writes	No. of InnoDB physical writes to the log file. Since 5.0.2.
Innodb_log_write_requests	No. of l InnoDB og write requests. Since 5.0.2.
Innodb_os_log_written	No. of bytes written to the InnoDB log file. Since 5.0.2.
Inno*db_os_log_fsyncs	No. of fsyncs writes to the InnoDB log file. Since 5.0.2.
Innodb_os_log_pending_writes	No. of pending InnoDB log file writes. Since 5.0.2.
Innodb_os_log_pending_fsyncs	No. of pending InnoDB log file syncs. Since 5.0.2.
Innodb_os_log_written	No. of bytes written to InnoDB redo log files
Innodb_page_size	Compiled-in InnoDB page size, default 16KB. Since 5.0.2.
Innodb_pages_created	No. of InnoDB pages created. Since 5.0.2.
Innodb_pages_read	No. of InnoDB pages read. Since 5.0.2.
Innodb_pages_written	No. of InnoDB pages written. Since 5.0.2.
Innodb_rows_deleted	No. of InnoDB rows deleted. Since 5.0.2.
Innodb_rows_inserted	No. of InnoDB rows inserted. Since 5.0.2.
Innodb_rows_read	No. of InnoDB rows read. Since 5.0.2.
Innodb_rows_updated	No. of InnoDCb rows updated. Since 5.0.2
Innodb_wakeups	No. of unnecessary mutex wakeups. 5.4.0.
Key_blocks_not_flushed	No. of changed index key blocks not yet flushed to disk. (`not_flushed_key_blocks` before 4.1.1)
Key_blocks_unused	No of blocks unused by key`cache. Since 4.1.2.
Key_blocks_used	No. of blocks used by key cache.
Key_read_requests	No. of requests to read an index key block from cache.
Key_reads	No. of physical reads of an index key block from disk. If big, *key cache* settings may be too small.
Key_write_requests	No. of requests to write an index key block to the cache.
Key_writes	No. of physical writes of an index key block to disk.
Last_query_cost	Query optimiser estimate of the cost of the last compiled subquery-less query in random data reads. Since 5.0.1.
Last_query_partial_plans	No. of optimiser iterations in building query execution plan. Since 5.6.5.

Variable	Meaning
Max_used_connections	Maximum No. of connections that have been open at one time.
Max_used_connections_time	Time when max_used_connections reached current value. Since 5.7.5.
Ndb_cluster_node_id	ID of this cluster node as setting of `node_id` in config file or connection string, or 0
Ndb_config_from_host	Name or IP address of cluster management server, or ''
Ndb_config_from_port	Port No. of connection to cluster server, or 0
Ndb_execute_count	No. of kernel execution round trips, since 6.3.6
Ndb_number_of_storage_nodes	No. of cluster nodes
Not_flushed_delayed_rows	No. of INSERT DELAYED rows waiting to be written.
Open_files	No. of open files
Open_streams	No. of open streams
Open_tables	No. of open tables
Opened_files	No. of tables opened with my_open()
Opened_tables	No. of tables opened in this session; if big, table_cache may be too small
Performance_???_cond_classes_lost	No. of instruments of type ??? lost. Since 5.5.3.
Performance_schema_???_instances_lost	No. of object instances of type ??? lost. Since 5.5.3.
Performance_schema_???_handles_lost	No. of object instances of type ??? could not be opened. Since 5.5.3.
Performance_schema_locker_lost	No. of events lost due to recursion or too-deep nesting. Since 5.5.3.
Prepared_stmt_count	GLOBAL prepared statement count (was a system variable until 5.0.33, 5.1.13)
Qcache_queries_in_cache	No. of queries registered in the cache
Qcache_inserts	No. of queries added to the cache
Qcache_hits	No. of cache hits
Qcache_lowmem_prunes	No. of queries deleted from cache due to low memory
Qcache_not_cached	No. of queries not cached because not cachable or because of *query_cache_type* setting
Qcache_free_memory	Amount of free memory for query cache
Qcache_free_blocks	No. of free memory blocks in query cache
Qcache_total_blocks	Total No. of blocks in query cache
Queries	No. of statements executed by the server including those executed from programs. Since 5.0.77, 5.1.31, 6.0.10
Questions	To 6.0.6, no. of queries sent to the server; since, no. of queries sent to the server from clients rather than programs
Rpl_semi_sync_master_clients	No. of semisynchronous slaves. Since 5.4.4, 6.0.8
Rpl_semi_sync_master_no_tx	No. of commits not acknowledged successfully by a slave. Since 5.4.4, 6.0.8
Rpl_semi_sync_master_status	1 if semisynchronous replication active as master and a commit acknowledgment is outstanding, else 0. Since 5.4.4, 6.0.8
Rpl_semi_sync_master_yes_tx	No. of commits acknowledged successfully by a slave. Since 5.4.4, 6.0.8
Rpl_semi_sync_slave_status	Is semisynchronous replication operational on the slave? Available only in 5.4 since 5.4.4 if plugin is installed
Rpl_status	Status of fail-safe replication. Unused
Select_full_join	No. of joins without keys, should be zero.
Select_full_range_join	No. of joins using range search on reference table.
Select_range	No. of joins using ranges on the first table.
Select_range_check	No. of joins without keys where key usage is checked after each row; should be 0

Variable	Meaning
Select_scan	No. of joins that scanned the first table
Slave_heartbeat_period	Slave replication heartbeat interval in seconds, 5.4 only series, introduced in 5.4.4
Slave_last_heartbeat	Time when slave last received a heartbeat signal, introduced in 5.6.1.
Slave_open_temp_tables	No. of temporary tables open in the slave thread
Slave_received_heartbeats	No. of heartbeats received by slave since last start or reset. 5.4 series only, introduced in 5.4.4.
Slave_retrieved_transactions	No. of retried transactions since last start
Slave_running	ON if this is a slave connected to a master
Slow_launch_threads	No. of threads that required more than slow_launch_time to connect
Slow_queries	No. of queries that required more than long_query_time.
Sort_merge_passes	No. of merges required by a sort; if this is high, increase sort_buffer.
Sort_range	No. of sorts using ranges
Sort_rows	No. of sorted rows
Sort_scan	No. of sorts done by scanning the table
Table_locks_immediate	No. of times a table lock was immediately acquired
Table_locks_waited	No. of times a table lock required a wait. If this is high, optimize queries, split tables, or replicate.
Threads_cached	No. of threads in the thread cache.
Threads_connected	No. of open connections
Threads_created	No. of threads created for connections. If cache hit rate (threads_created/connections) is too big, thread_cache_size may be too small.
Threads_running	No. of threads not sleeping.
Uptime	No. of seconds the server has been up
Uptime_since_flush_status	No. of seconds since last FLUSH. Since 5.1.24, 6.05.

MySQL error codes

MySQL error messages vary by version. From 5.0 to 5.1, message text generation method changed. The source text for error messages in *share/errmsg.txt* defines error output for *include/mysqld_error.h*, *include/mysqld_ ername* and *include/sql_ state.h*. For tabulations of MySQL client and server errors, and notes on dealing with them, see “MySQL Server Errors” and “MySQL Client Errors” at *www.artfulsoftware.com/infotree/mysqltips.php*.

MySQL error handling does not use all conventional SQLSTATE values. For a full list see *here*.