

# MySQL8 New Features: Temptable Engine

Pep Pla

Tue Nov 5th 2018 - Percona Live Europe  
Frankfurt, Germany

Pythian

## Pep Pla

Born in Vinaròs, a small village near the Mediterranean and currently living in Barcelona.

Most of the time I'm busy with my three kids, my partner and our two cats.

And in my spare time I'm a DBC at Pythian, surrounded by some of the most brilliant DBAs in the world.







# ABOUT PYTHIAN

Pythian's 400+ IT professionals  
help companies adopt and  
manage disruptive technologies  
to better compete



**20**

Years in Business



**400+**

Pythian Experts  
in 35 Countries



**350+**

Current Clients  
Globally

# AGENDA



- Introduction
- Sorting
- Temporary space
- Memory engine
- Temptable engine
- Benchmarks

# Sorting

It can change your life

# Sorting

- Definition of “sort”
  - to put a number of things in an order or to separate them into groups
- It is a natural activity
  - Kids are taught how to sort things: colors, sizes, numbers...
  - As adults we classify everything and everybody.
- The results of “sorting” can change your life.

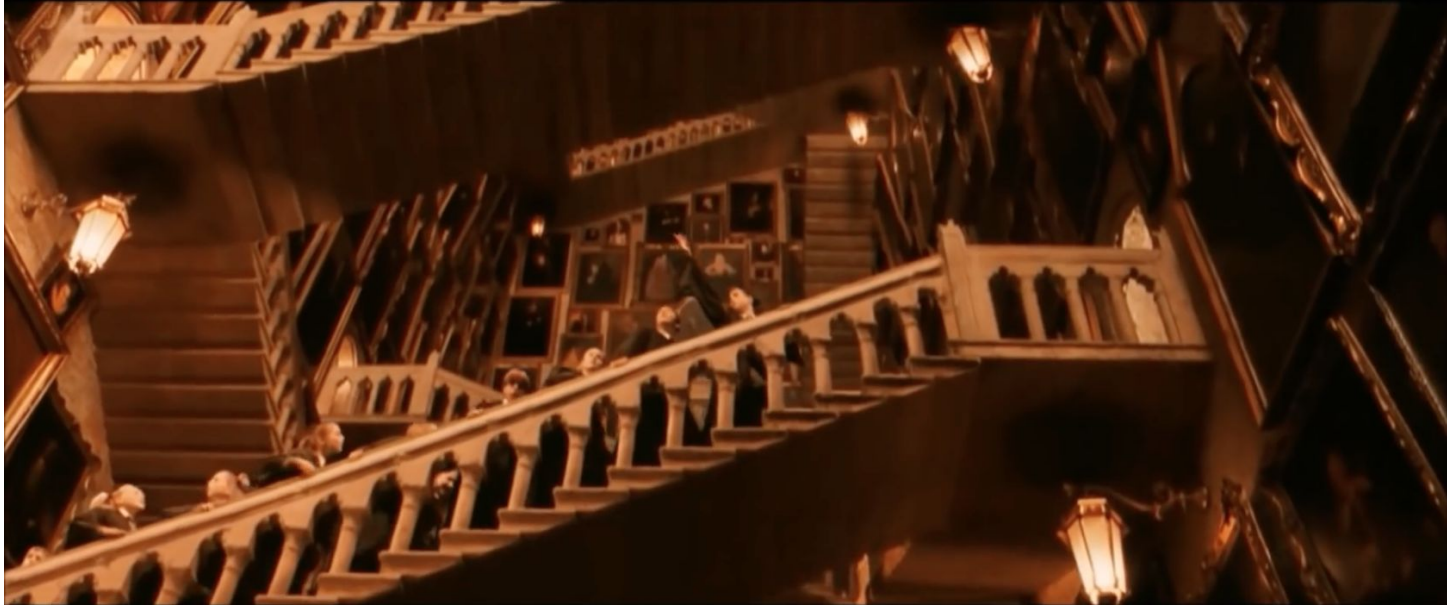


# The Sorting Hat





# The Sorting Hat



# Sorting

- The Sorting Hat method is quite simple.
- What if we want to perform more complex classifications?
  - Place all the students born on the same month together.
- Some methods:
  - Sorting Hat and then order by age.
  - Order by age and then Sorting Hat.
  - Sorting Hat and order by age at the same time.
- Problem: We need a space to keep all the students during the process.

# Sorting


- MySQL Sorting Hat is called Indexes.
- What if we want to perform more complex classifications?
  - MySQL has algorithms to perform complex classifications.
- Problem: We need a space to keep all the data during the process.

# Temporary Space

Time flies

# Temporary Space

## temporary

/ˈtɛmp(ə)rəri/ 

*adjective*

adjective: **temporary**

1. lasting for only a limited period of time; not permanent.

"a temporary job"

*synonyms:* non-permanent, short-term, interim; More

*antonyms:* permanent, lasting

### Origin



mid 16th century: from Latin *temporarius*, from *tempus*, *tempor-* 'time'.



# Temporary Space

- MySQL stores information on tables.
- MySQL has two different types of “temporary tables”.
  - Explicit temporary tables
  - Implicit temporary tables

# Temporary Space

- MySQL explicit temporary tables.
  - Create temporary table statement.
  - Session based
    - Only visible from the session that creates the table
    - Dropped when the session is closed.
  - Create temporary table privilege.

# Temporary Space

- MySQL implicit temporary tables.
  - Created internally without user intervention.
  - Statement based
    - Usually only available during the execution of the statement that required the table
  - Invisible for the user, no privileges required.

# Temporary Space

- Some operations that create an internal temporary table:
  - Evaluation of some UNION statements, of some views, of derived tables, of common table expressions.
  - Subquery or semi-join materialization
  - Evaluation of statements that contain an ORDER BY clause and a different GROUP BY clause, or for which the ORDER BY or GROUP BY contains columns from tables other than the first table in the join queue.
  - Evaluation of DISTINCT combined with ORDER BY may require a temporary table.
  - INSERT ... SELECT statements that select from and insert into the same table, .
  - ....

# Temporary Space

- Some misconceptions
  - Table rebuilds do not create a “temporary” table.
  - Memory engine tables.
- Performance requirements
  - Use the fastest possible storage: memory
  - Overflow to disk if required.



# Memory engine

All alone in the moonlight

# Memory engine

- Default engine for internal temporary tables before MySQL8
- Fixed row format
  - Bad for “oversized” columns
- Configured using per session parameters.
  - `Tmp_table_size` or `(max_heap_table_size if lower)`
- Data migrated to disk table in case of an overflow.
- No additional features required: partitioning, privileges or MVCC

# Memory engine

- Status information:
  - Created\_tmp\_tables
  - Created\_tmp\_disk\_tables
  - Sort\_merge\_passes
  - Sort\_range
  - Sort\_rows
  - Sort\_scan
- Performance schema
  - memory/memory/HP\_SHARE
  - memory/memory/HP\_INFO
  - memory/memory/HP\_PTRS
  - memory/memory/HP\_KEYDEF

# Temptable

No Barbra Streisand songs with Temptable, I'm sorry.

# Temptable engine

- Default engine for internal temporary in MySQL8
- Not “really” an engine.
  - Only for internal temporary tables.
  - It is not shown with `show engines`.
- Variable width row format
  - Great for “oversized” columns
- Configured using global parameters.
  - `temptable_max_ram`
- Data is not migrated to disk table in case of an overflow.  
(Documentation)
- No additional features required: partitioning, privileges or MVCC



# Temptable engine

- Status information:
  - Created\_tmp\_tables
  - Sort\_merge\_passes
  - Sort\_range
  - Sort\_rows
  - Sort\_scan
- Performance schema
  - memory/temptable/physical\_disk
  - memory/temptable/physical\_ram

# Temptable vs. Memory

Fight!

# Temptable vs. Memory

- Memory utilization
  - Temptable memory usage is global.
  - Memory memory usage is controlled per session.
  - Both are allocated and released on demand.
  - Memory can bring your system to its knees.
- Efficiency
  - Temptable uses variable row format.
  - Memory uses a fixed row format.
  - Temptable usually needs far less memory.
- No data migration
  - Temptable does not migrate data.
  - Memory converts the ALL the table to a disk engine if required.

# Benchmarks

BYOB

# Benchmarks

- Not a real production environment
  - One table with 20.000.000 rows.
  - Only one session. No concurrency.
- Three tests
  - One large temporary table without sorting (cursor).
  - One large temporary table with sorting.
  - 500 small temporary tables without sorting.
- Four memory configurations
  - 5Gb, 1Gb, 512Mb and 256Mb.
  - The same values where used for both engines.

# Benchmarks

- One large temporary table without sorting (cursor).
  - Temptable is 13% faster.
  - Temptable needs 1% Write operations.(!)
- One large temporary table with sorting.
  - Temptable is 15% faster.
  - Temptable again needs 1% Write operations.(!)
- Multiple small operations without sorting
  - Temptable is 42% faster.
  - Temptable again needs 1% Write operations

# Benchmarks

- Conclusions:
  - Temptable is faster.
  - Temptable is safer.
  - Concurrent tests are needed (mutex contention could be an issue)
  - More detailed analysis of write operations is also needed.






# THANK YOU

We're hiring!

<https://www.pythian.com/careers/>



# Pythian