What’s New in MySQL 8.0?

Geir Høydalsvik
Senior Software Development Director
Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
MySQL 8.0 GA Now Available!

- 2-3 years in Development
- 400+ Worklogs
- 5000+ Bugs Fixed
- 500 new tests
- The MySQL Server, but more...

MySQL Document Store  MySQL DevAPI
MySQL InnoDB Cluster  DevAPI based MySQL Shell
DevAPI based MySQL Connectors
Agenda: MySQL 8.0 for Dev & DevOps

Developers
- NoSQL
- SQL
  - JSON
  - GIS
  - Character Sets

The crazy young guys...

DevOps
- Reliability
- Observability
- Manageability
- Security
- Performance

The adults in the room...
MySQL 8.0 = SQL + NoSQL

Relational Database
• SQL
• INSERT, SELECT, UPDATE, DELETE
• Tables
• Schema
• Powerful SELECT (e.g. join)

Document Store
• JavaScript (and more), CRUD
• Create, Read, Update, Delete
• JSON Documents
• Schema-less
• More limited Read/Find

MySQL 8.0 : Mix and Match !
MySQL Document Store

• DevAPI (patterns)
  – Link your existing relational data with your document data in a single query using simple API patterns
• DevAPI based **Connectors**
  – **Node.js**, Python, PHP, .Net, ODBC, C++, Java
• DevAPI based **MySQL Shell**
  – Dev+Admin
• New, async **client-server protocol** (X Protocol)
• And, the “good old MySQL Server”
Agenda: MySQL 8.0 for Dev & DevOps

Developers
• NoSQL
• SQL
• JSON
• GIS
• Character Sets
The crazy young guys...

DevOps
• Reliability
• Observability
• Manageability
• Security
• Performance
The adults in the room...
SQL: Window Functions

```sql
mysql> SELECT employee, date, sale, SUM(sale) OVER (PARTITION BY employee) AS sum FROM sales;
```

```
+------------+--------+---+-----+
<table>
<thead>
<tr>
<th>employee</th>
<th>date</th>
<th>sale</th>
<th>sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>odin</td>
<td>2017-03-01</td>
<td>200</td>
<td>900</td>
</tr>
<tr>
<td>odin</td>
<td>2017-04-01</td>
<td>300</td>
<td>900</td>
</tr>
<tr>
<td>odin</td>
<td>2017-05-01</td>
<td>400</td>
<td>900</td>
</tr>
<tr>
<td>thor</td>
<td>2017-03-01</td>
<td>400</td>
<td>1200</td>
</tr>
<tr>
<td>thor</td>
<td>2017-04-01</td>
<td>300</td>
<td>1200</td>
</tr>
<tr>
<td>thor</td>
<td>2017-05-01</td>
<td>500</td>
<td>1200</td>
</tr>
</tbody>
</table>
+------------+--------+---+-----+
```
SQL: Common Table Expressions

Non-recursive CTEs:
(improved derived tables)

WITH d AS
(SELECT a, b, SUM(c) s
 FROM t1 GROUP BY a, b)
AS d1
SELECT ...
FROM d AS d1
JOIN d AS d2
ON d1.b = d2.a;

Recursive CTEs:

WITH RECURSIVE qn AS
(SELECT 1 AS a
 UNION ALL
SELECT 1 + a FROM qn
WHERE a < 10
)
SELECT * FROM qn;
SQL: NOWAIT and SKIP LOCKED

```sql
SELECT * FROM tickets
WHERE id IN (1,2,3,4)
AND order_id IS NULL
FOR UPDATE
NOWAIT;
```

```
SELECT * FROM tickets
WHERE id IN (1,2,3,4)
AND order_id IS NULL
FOR UPDATE
SKIP LOCKED;
```

- Error immediately if a row is already locked
- Non deterministically skip over locked rows
SQL: Datatypes

• Bitwise operations on binary datatypes

[VAR]BINARY / [TINY|MEDIUM|LONG]BLOB

• IPv6 manipulation

fe80::226:b9ff:fe77:eb17 to/from VARBINARY(16)
INET6_ATON(address) & INET6_ATON(network)

• UUID manipulation

aab5d5fd-70c1-11e5-a4fb-b026b977eb28 to/from VARBINARY(16)
UUID_TO_BIN(+swap), BIN_TO_UUID(+swap), IS_UUID()
Agenda: MySQL 8.0 for Dev & DevOps

Developers
- NoSQL
- SQL
- JSON
- GIS
- Character Sets

The crazy young guys...

DevOps
- Reliability
- Observability
- Manageability
- Security
- Performance

The adults in the room...
JSON Functions
MySQL 5.7 and 8.0

JSON_ARRAY_APPEND()
JSON_ARRAY_INSERT()
JSON_ARRAY()
JSON_CONTAINS_PATH()
JSON_CONTAINS()
JSON_DEPTH()
JSON_EXTRACT()
JSON_INSERT()
JSON_KEYS()
JSON_LENGTH()

JSON_MERGE[ _PRESERVE]()
JSON_OBJECT()
JSON_QUOTE()
JSON_REMOVE()
JSON_REPLACE()
JSON_SEARCH()
JSON_SET()
JSON_TYPE()
JSON_UNQUOTE()

JSON_PRETTY()
JSON_STORAGE_SIZE()
JSON_STORAGE_FREE()
JSON_ARRAYAGG()
JSON_OBJECTAGG()
JSON_MERGE_PATCH()
JSON_TABLE()
JSON Aggregates
From SQL Table to JSON Document

```
mysql> select * from employees;

+----------+------+
| id | name | age |
+----------+------+
| 1    | John | 34  |
| 2    | Mary | 40  |
| 3    | Mike | 44  |
+----------+------+
3 rows in set (0,00 sec)

mysql> set @jsonempl=(SELECT JSON_ARRAYAGG(JSON_OBJECT("id", id, "name", name, "age", age)) FROM employees);

mysql> select * from @jsonempl;

[ { "id": 1, "age": 34, "name": "John"}, {"id": 2, "age": 40, "name": "Mary"}, {"id": 3, "age": 44, "name": "Mike"}]```
### JSON: JSON\_TABLE()

**From JSON Document to SQL Table**

```sql
mysql> SELECT emps.* FROM JSON\_TABLE(@jsonempl, "$[*]" COLUMNS (id INT PATH ".id", name VARCHAR(45) PATH ".name", age INT PATH ".age") emps;
+-------------------------+
<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John</td>
<td>34</td>
</tr>
<tr>
<td>2</td>
<td>Mary</td>
<td>40</td>
</tr>
<tr>
<td>3</td>
<td>Mike</td>
<td>44</td>
</tr>
</tbody>
</table>
+-------------------------+
3 rows in set (0,00 sec)
```
JSON: Partial Update of JSON Values
Performance Improvement

• When apps frequently update small portions of large JSON documents
  JSON_SET(), JSON_REPLACE(), JSON_REMOVE()

• Example:
  UPDATE t SET json_col =
  JSON_SET(json_col, '$.name', 'Geir') WHERE id=123;
• Only write the change back to the database
• Only replicate the change to slaves
• Facilitated by new InnoDB BLOB implementation
Agenda: MySQL 8.0 for Dev & DevOps

Developers
- NoSQL
- SQL
- JSON
- GIS
- Character Sets

The crazy young guys...

DevOps
- Reliability
- Observability
- Manageability
- Security
- Performance

The adults in the room...
GIS: Geography

• Earth’s surface, longitude/latitude
• SQL/MM Part 3 Spatial
• Spatial Reference System (SRS)
  – 5000 SRIDs from EPSG Geodetic Parameter Dataset
  – SRID 4326 = WGS 84 («GPS coordinates»)
  – INFORMATION_SCHEMA.ST_SPATIAL_REFERENCE_SYSTEMS
• SRID aware Spatial Datatypes
  – CREATE TABLE t1 (g GEOMETRY SRID 4326);
  – INFORMATION_SCHEMA.ST_GEOMETRY_COLUMNS
GIS: Geography (cont)

• SRID aware Spatial Indexes
  - `CREATE TABLE t1 (g GEOMETRY SRID 4326 NOT NULL, SPATIAL INDEX (g));`

• SRID aware Spatial Functions
  - `ST_Distance()`, `ST_Length()`, ...
  - `ST_Within()`, `ST_Intersects()`, `ST_Contains()`, ...

• GeoHash(), GeoJSON()
Agenda: MySQL 8.0 for Dev & DevOps

Developers
- NoSQL
- SQL
- JSON
- GIS
- Character Sets
  The crazy young guys...

DevOps
- Reliability
- Observability
- Manageability
- Security
- Performance
  The adults in the room...
Character Sets

• Emoji characters used as input
• MySQL 8.0 defaults to utf8mb4
• Latest Unicode 9.0 Support
• New collations based on DUCET, accent and case sensitive collations, kana sensitive collation, Japanese, Russian
Character Sets : Regular Expressions

• Unicode support
  – Using International Components for Unicode (ICU) library, UTF8MB4

• Additional functionality

  REGEXP_INSTR(), REGEXP_LIKE(), REGEXP_REPLACE(), REGEXP_SUBSTR()

  mysql> SELECT regexp_like ( 'Abba', 'ABBA', 'i' );
  1

  mysql> SELECT regexp_like ( 'Abba', 'ABBA', 'c' );
  0

• Resource control

  reg-exp-stack-limit, reg-exp-time-limit
Agenda: MySQL 8.0 for Dev & DevOps

Developers
• NoSQL
• SQL
• JSON
• GIS
• Character Sets

The crazy young guys...

DevOps
• Reliability
• Observability
• Manageability
• Security
• Performance

The adults in the room...
Reliability
DevOps want their systems to be running or at least to be in a recoverable state

• Meta-data stored in InnoDB
  – From files to a proven transactional storage engine
  – System tables moved from MyISAM to InnoDB

• One source of truth
  – One common data dictionary

• Atomic, crash safe DDL
  CREATE/DROP USER <u1, u2, u3>, DROP TABLE <t1, t2, t3>,...
Agenda: MySQL 8.0 for Dev & DevOps

Developers
• NoSQL
• SQL
• JSON
• GIS
• Character Sets

The crazy young guys...

DevOps
• Reliability
• Observability
• Manageability
• Security
• Performance

The adults in the room...
Observability : Information Schema
DevOps & tools need access to **persistent, static meta-data**

- Re-implemented as views on data dictionary tables, better performance
- **New tables** `INNODB_TABLESPACES, ST_GEOMETRY_COLUMNS, ...`
Observability: Performance Schema

DevOps & tools need access to volatile, dynamic meta-data

- Client Error Reporting
- Query Latency Histograms
- Data Locking Dependency Graph
- Digest Query Sample
- Better Query Performance

(0 - 348ms) 64 |******************************
(348 - 696ms) 7 | .......
(696 - 1044ms) 2 | ..
(1044 - 1393ms) 2 | ..
(1393 - 1741ms) 1 | .
(1741 - 2089ms) 0 | |
(2089 - 2437ms) 0 | |
(2437 - 2785ms) 0 | |
(2785 - 3133ms) 0 | |
(3133 - 3481ms) 1 | .
(3481 - 3829ms) 0 | |
(3829 - 4178ms) 0 | |
(4178 - 4526ms) 0 | |
(4526 - 4874ms) 0 | |
(4874 - 5222ms) 0 | |
(5222 - 5570ms) 0 | |
Total Statements: 78; Buckets: 16; Bucket Size: 348 ms;
Observability: Configuration Variables

DevOps & tools need access to current configuration settings

• New Performance Schema table `variables_info`
  – Variable `name`
  – Current `value`
  – Value Range, `min/max` values
  – `Where` the current value came from
  – `Who` made the change
  – `When` did it happen
Observability : Error Logging
DevOps monitor the overall health of the system (ErrorLog/SysLog)

• Pluggable Error Log
• User can add *filters* (what to write) and *sinks* (where to write)
• Error Numbering [MY-010116]
• System Messages [System][Error][Warning][Note]
• Reduced Default Verbosity (from Notes to Warnings)
• Source [Server][InnoDB][Replic]

2018-03-08T10:14:29.289863Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.5) starting as process 8063

• Continue to improve Error Log Messages:
  – «DevOps/DBAs should only see relevant and actionable messages in the error log»
Agenda: MySQL 8.0 for Dev & DevOps

Developers
- NoSQL
- SQL
- JSON
- GIS
- Character Sets

The crazy young guys...

DevOps
- Reliability
- Observability
- Manageability
- Security
- Performance

The adults in the room...
Manageability: Invisible Indexes

• DBAs reduce risk when creating and dropping indexes
  – Can be very expensive to create an index when tables are large

• Indexes are “hidden” to the MySQL Optimizer
  – Not the same as “disabled indexes”
  – Contents are fully up to date and maintained by DML

• Two use cases:
  – Soft Delete (*Recycle Bin*)
  – Staged Rollout
Manageability: Optimizer Histograms

• DBAs provide more information to the optimizer
  – MySQL makes better choices, typically for non-indexed columns
  – Selectivity of predicates of the form «COLUMN operator CONSTANT»

• ANALYZE TABLE
  UPDATE HISTOGRAM ON column [,column][WITH n BUCKETS]
  DROP HISTOGRAM ON column [,column]

• Sample (or not) based on table size
• Singleton or equi-height based on data distribution and #buckets
• INFORMATION_SCHEMA.COLUMN_STATISTICS
Manageability : New Optimizer Hints

• **After** `SELECT | INSERT | REPLACE | UPDATE | DELETE`

• Enclosed in `/*+ */` style comments

• `INDEX_MERGE, NO_INDEX_MERGE`
  – Control index merge behavior

• `JOIN_FIXED_ORDER, JOIN_ORDER, JOIN_PREFIX, JOIN_SUFFIX`
  – Control table order for join execution

• `SET_VAR`
  – Set the value for a given system variable for the next statement only
Manageability : Instant DDL

• Instant Schema Changes
• `ADD COLUMN` *(Now in Labs, 8.0.12 preview)*
• `RENAME TABLESPACE`
• `RENAME COLUMN`
Manageability : Remote Management

• Configure your server without leaving your client !
• Required privileges:
  – SYSTEM_VARIABLES_ADMIN, PERSIST_RO_VARIABLES_ADMIN, SHUTDOWN

```sql
mysql> SET PERSIST innodb_buffer_pool_size=134217728*2;
mysql> RESET PERSIST innodb_buffer_pool_size;

mysql> SET PERSIST_ONLY innodb_log_file_size=50331648*2;
mysql> RESTART;
```
Manageability: InnoDB Undo Tablespaces

• Upgrade to 8.0 moves the Undo log out of the System Tablespace into Undo tablespaces (default 2)
• Flexible Undo Tablespace Management, user decides:
  – How many tablespaces (min 2, max 127)
  – Where they are placed (e.g. fast storage)
  – How many rollback segments in each tablespace (min 1, max 128)
• Shrink Undo tablespaces **online**
  – Reclaim space used by unusually large transactions
  – Thus, two, one active tablespace while the other being truncated
Manageability: Better Defaults

- Defaults should reflect what is used in practice
  - UTF8MB4 Character Set, Binary Log ON, etc.
- Dedicated MySQL Server \( \text{innodb\_dedicated\_server=} \text{ON} \)
  - Automatically configured based on detected server memory
    - Buffer Pool Size
    - Log File Size
  - Flush Method = \text{O\_DIRECT\_NO\_FSYNC}
Agenda: MySQL 8.0 for Dev & DevOps

Developers
- NoSQL
- SQL
- JSON
- GIS
- Character Sets

The crazy young guys...

DevOps
- Reliability
- Observability
- Manageability
- Security
- Performance

The adults in the room...
Security : OpenSSL

• OpenSSL by Default in Community Edition

• OpenSSL is Dynamically Linked
  – 8.0 uses System provided OpenSSL library by default
  – User can replace the OpenSSL library, comply with customer policy

• OpenSSL FIPS Object Module is Supported
  – For usage in environments which requires FIPS compliance

• Hardware encryption support is enabled (AES)
  – Better encrypt/decrypt performance «out of the box»
Security : New Default Authentication

• Use TLS/SSL for all your network communication!
• Thus, change your old `mysql_native_password` plugin
• To the new `caching_sha2_password` plugin
  – Better security (SHA2 algorithm)
  – High performance (caching)
• Default change both on Client and Server side
• Old authentication can still be used
• Upgrade: Existing user accounts are not affected
Security: More on Authentication

- Slow down *brute force attacks* on user passwords
  - Inject delays in the authentication process
  - Based on consecutive unsuccessful login attempts

- Password *rotation policy*
  - Restrictions on password reuse
  - Configures globally or per user
  - Password history is kept secure
  - Addition to existing *expiration policy* and *allowed password policy*
Security: Roles
Simplifies the management of user access rights

- **SQL Roles**
  - CREATE ROLE, DROP ROLE
  - GRANT, REVOKE, SHOW GRANTS
  - SET DEFAULT ROLE, SET ROLE, CURRENT_ROLE()

- Allow grants and revoke for PUBLIC
  - Config variable mandatory-roles (default roles)
  - GRANT/REVOKE to default roles will affect all users
  - Config variable activate-all-roles-on-login=ON
Security: Administrative Privileges

• Breaking up the **SUPER PRIVILEGE**
  – Limit user access rights to what is needed (and nothing more)
  – Examples:
    • BINLOG_ADMIN
    • CONNECTION_ADMIN
    • ROLE_ADMIN
    • XA_RECOVER_ADMIN

• Other: Resetting the Root Password
  – Disallow remote connections when server is started with skip-grant-tables
Agenda: MySQL 8.0 for Dev & DevOps

Developers
• NoSQL
• SQL
• JSON
• GIS
• Character Sets
The crazy young guys...

DevOps
• Reliability
• Observability
• Manageability
• Security
• Performance
The adults in the room...
Performance

• Scaling Read/Write Loads
  – Re-designing how InnoDB writes to Redo log

• Utilizing IO Capacity
  – Removing file system mutex

• High Contention Loads
  – Contention Aware Transaction Scheduling

• Resource Groups
  – Thread–CPU mapping can be managed

• UTF8MB4
• Partial JSON/BLOB update
• Information Schema
• Performance Schema
• Cost Model (mem/disk aware)
sysbench IO bound read only (point selects)
2x faster than MySQL 5.7

MySQL 8.0
MySQL 5.7
MySQL 5.6

OS: Oracle Linux 7.4
CPU: 48 cores- HT Intel Skylake 2.7Ghz
(2 CPU sockets, Intel(R) Xeon(R) Platinum 8168 CPU)
RAM: 256GB Storage: x2 Intel Optane flash devices
(Intel (R) Optane (TM) SSD P4800X Series)
MySQL 8.0: SysBench Read Write (update nokey)
2x Faster than MySQL 5.7

OS: Oracle Linux 7.4
CPU: 48 cores (HT Intel Skylake 2.7Ghz)
(2CPU sockets, Intel(R) Xeon(R) Platinum 8168 CPU)
RAM: 256GB
Storage: x2 Intel Optane flash devices
(Intel(R) Optane(TM) SSD P4800X Series)
Performance: Resource Groups

System Configuration:
Oracle Linux 7,
Intel(R) Xeon(R) CPU E7-4850 2.27GHz
40cores-HT

- No Resource Group
  (48 Cores Shared)
- With Resource Group
  (48 Cores for Select)
  (10 Cores for Update RG)

Queries per Second

- Select
- Update
Integrated Cloud
Applications & Platform Services