Meet the 10.2

Sergei Golubchik
MariaDB Corporation
10.2

- **Facts**
  - About a year in active development
  - Currently at 10.2.2-beta

- **Plans**
  - Gamma soon
  - GA by the end of the year
• Analytical queries
• Removing historical limitations
• Client and protocol
• Optimizations
• MySQL compatibility

• GSoC!
Analytical queries

Common Table Expressions
Subquery in the FROM clause

```
SELECT * FROM t2,
   (SELECT a FROM t1 WHERE b >= 'c') AS sq
WHERE t2.c=sq.a;
```
WITH sq AS (SELECT a FROM t1 WHERE b >= 'c')
SELECT * FROM t2, sq WHERE t2.c=sq.a;
SELECT * FROM t2,
    (SELECT a FROM t1 WHERE b >= 'c') AS sq
WHERE t2.c=sq.a;

WITH sq AS (SELECT a FROM t1 WHERE b >= 'c')
SELECT * FROM t2, sq WHERE t2.c=sq.a;
WITH RECURSIVE ancestors AS (
    SELECT * FROM folks WHERE name = 'Sergei'
    UNION ALL
    SELECT folks.* FROM folks, ancestors
    WHERE folks.id = ancestors.father
    OR folks.id = ancestors.mother
)
SELECT * FROM ancestors;
### Recursive CTE: factorial

```sql
WITH RECURSIVE fact AS
    (SELECT 1 AS n, 1 AS `n!` UNION ALL
     SELECT n+1, `n!`*(n+1) FROM fact WHERE n < 20)
) SELECT * FROM fact;
```

<table>
<thead>
<tr>
<th>n</th>
<th>n!</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>120</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>2432902008176640000</td>
</tr>
</tbody>
</table>

+-----+---------------------+
Recursive CTE: Fibonacci

WITH RECURSIVE fib (a, b) AS
    (SELECT 1, 1 UNION ALL
        SELECT b, a+b FROM fib WHERE b < 200000
    )
SELECT a FROM fib;

+---+
| a |
+---+
| 1 |
| 1 |
| 2 |
| 3 |
| 5 |
| 8 |
| 13 |
...

+--------+
| a      |
+--------+
|      1 |
|      1 |
|      2 |
|      3 |
|      5 |
|      8 |
|     13 |
...
Analytical queries

Window functions
The schema

```sql
SELECT * FROM information_schema.profiling;
```

<table>
<thead>
<tr>
<th>QUERY_ID</th>
<th>SEQ</th>
<th>STATE</th>
<th>DURATION</th>
<th>CPU_USER</th>
<th>CPU_SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>17</td>
<td>query end</td>
<td>0.000054</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>starting</td>
<td>0.000322</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>10</td>
<td>23</td>
<td>updating status</td>
<td>0.000083</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>13</td>
<td>22</td>
<td>closing tables</td>
<td>0.000046</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>preparing</td>
<td>0.000110</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>checking permissions</td>
<td>0.000068</td>
<td>0.001000</td>
<td>0.000000</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>Table lock</td>
<td>0.000079</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>Creating sort index</td>
<td>0.000044</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>Sending data</td>
<td>0.000127</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>8</td>
<td>18</td>
<td>query end</td>
<td>0.000053</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

...
### Running total

```sql
SELECT query_id, seq, state, duration,
  (SELECT SUM(duration) FROM information_schema.profiling AS p
   WHERE p.query_id=pp.query_id AND p.seq <= pp.seq) AS rt
FROM information_schema.profiling AS pp
ORDER BY query_id, seq;
```

<table>
<thead>
<tr>
<th>query_id</th>
<th>seq</th>
<th>state</th>
<th>duration</th>
<th>rt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>starting</td>
<td>0.000304</td>
<td>0.000304</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>checking permissions</td>
<td>0.000063</td>
<td>0.000367</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>27</td>
<td>updating status</td>
<td>0.000085</td>
<td>0.002489</td>
</tr>
<tr>
<td>1</td>
<td>28</td>
<td>cleaning up</td>
<td>0.000081</td>
<td>0.002570</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>starting</td>
<td>0.000397</td>
<td>0.000397</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>checking permissions</td>
<td>0.000069</td>
<td>0.000466</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

336 rows in set (3.55 sec)
Running total

```
SELECT query_id, seq, state, duration,
       SUM(duration) OVER (PARTITION BY query_id ORDER BY seq) AS rt
FROM information_schema.profiling ORDER BY query_id, seq;
```

```
+----------+-----+----------------------+----------+----------+
| query_id | seq | state                | duration | rt       |
|----------+-----+----------------------+----------+----------|
|         1 |   2 | starting             | 0.000304 | 0.000304 |
|         1 |   3 | checking permissions | 0.000063 | 0.000367 |
|         2 |   2 | starting             | 0.000397 | 0.000397 |
|         2 |   3 | checking permissions | 0.000063 | 0.000367 |
|         1 |  27 | updating status      | 0.000085 | 0.002489 |
|         1 |  28 | cleaning up          | 0.000081 | 0.002570 |
|         2 |   2 | starting             | 0.000397 | 0.000397 |
|         2 |   3 | checking permissions | 0.000069 | 0.000466 |
|         ...|    |                      |          |          |
|         ...|    |                      |          |          |
+----------+-----+----------------------+----------+----------+

336 rows in set (0.04 sec)
SELECT query_id, state, duration,
    NTILE(10) OVER (ORDER BY duration) nt
FROM information_schema.profiling
WHERE query_id=30 ORDER BY nt DESC;

+----------+----------------------+----------+------+
| query_id | state                | duration | nt   |
+----------+----------------------+----------+------+
|       30 | Filling schema table | 0.012040 |   10 |
|       30 | Creating sort index  | 0.005448 |   10 |
|       30 | Sending data         | 0.004479 |   10 |
|       30 | Sending data         | 0.004017 |    9 |
|       30 | Opening tables       | 0.000545 |    9 |
|       30 | removing tmp table   | 0.000354 |    9 |
|       30 | starting             | 0.000346 |    8 |
|       30 | removing tmp table   | 0.000138 |    8 |

...
Running average

```sql
SELECT query_id, seq, AVG(duration) OVER
    (ORDER BY query_id ROWS BETWEEN 5 PRECEDING AND 5 FOLLOWING) x
FROM information_schema.profiling
WHERE state='end' ORDER BY query_id;
```
Removing limitations

Temporary tables
Before 10.2: self-join

```
CREATE TEMPORARY TABLE employees (  
   id INT PRIMARY KEY,  
   name VARCHAR(100),  
   reports_to INT);  
Query OK, 0 rows affected (0.00 sec)

INSERT employees VALUES (1, 'Rasmus', NULL), (2, 'Sergei', 1);  
Query OK, 1 rows affected (0.00 sec)

SELECT e.name, m.name AS manager FROM employees e, employees m  
 WHERE e.reports_to=m.id;  
ERROR 1137 (HY000): Can't reopen table: 'employees'
```
Temporary tables in 10.2

```sql
CREATE TEMPORARY TABLE employees (
    id INT PRIMARY KEY,
    name VARCHAR(100),
    reports_to INT);
INSERT employees VALUES (1, 'Rasmus', NULL), (2, 'Sergei', 1);

SELECT e.name, m.name AS manager FROM employees e, employees m
    WHERE e.reports_to=m.id;
```

<table>
<thead>
<tr>
<th>name</th>
<th>manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sergei</td>
<td>Rasmus</td>
</tr>
<tr>
<td>Sergei</td>
<td></td>
</tr>
</tbody>
</table>
Removing limitations

CHECK constraint
CREATE TABLE t1 (a INT, b INT CHECK (b > 10), CHECK (a > b));
Query OK, 0 rows affected (0.00 sec)

INSERT t1 VALUES (5, 6);
Query OK, 1 row affected (0.00 sec)

Wait, what?
**CHECK constraint in 10.2**

```sql
CREATE TABLE t1 (a INT, b INT CHECK (b > 10), CHECK (a > b));

INSERT t1 VALUES (5, 6);
ERROR 4025 (23000): CONSTRAINT `b` failed for `test`.`t1`

INSERT t1 VALUES (5, 16);
ERROR 4025 (23000): CONSTRAINT `CONSTRAINT_1` failed for `test`.`t1`
```
Removing limitations

DEFAULT clause
DEFAULT clause in 10.2

- Expressions
- DEFAULT for BLOBs

```sql
CREATE TABLE defs (  uid CHAR(32) DEFAULT UUID(),  c1 INT, c2 INT DEFAULT (c1 + 1),  data BLOB DEFAULT 'foo')
```
Removing more limitations: generated columns

- Up to 64K per expression (was 252 bytes)
- Can use constant expressions
- Can refer to other virtual columns
- Can use non-deterministic functions (UDFs, server variables, ...)

```sql
CREATE TABLE t1 (  
a INT GENERATED ALWAYS AS (10),
b INT GENERATED ALWAYS AS (a+1),
c TIMESTAMP GENERATED ALWAYS AS (NOW() + INTERVAL 1 HOUR)
);
```
Removing even more limitations

- longer DECIMAL

```sql
CREATE TABLE t1 (a DECIMAL(65,38));
```

- Views and subqueries in the FROM clause

```sql
CREATE VIEW v1 AS
SELECT * FROM (SELECT a+1 FROM t1) x;
```
Other features
### New user management commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CREATE USER foo@bar REQUIRE SSL WITH MAX_QUERIES_PER_HOUR 10;</strong></td>
<td>Creates a user named foo@bar with SSL requirement and a maximum of 10 queries per hour.</td>
</tr>
<tr>
<td></td>
<td>Query OK, 0 rows affected (0.00 sec)</td>
</tr>
<tr>
<td><strong>ALTER USER foo@bar IDENTIFIED VIA pam WITH MAX_USER_CONNECTIONS 3;</strong></td>
<td>Changes the authentication method for user foo@bar to pam and sets the maximum number of connections to 3.</td>
</tr>
<tr>
<td></td>
<td>Query OK, 0 rows affected (0.00 sec)</td>
</tr>
</tbody>
</table>

```sql
SHOW CREATE USER foo@bar;
```

```sql
CREATE USER for foo@bar
```

```sql
CREATE USER 'foo'@'bar' IDENTIFIED VIA pam REQUIRE SSL WITH MAX_QUERIES_PER_HOUR 10 MAX_USER_CONNECTIONS 3
```

```sql
+------------------------------------------------------------
| CREATE USER 'foo'@'bar' IDENTIFIED VIA pam REQUIRE SSL |
| WITH MAX_QUERIES_PER_HOUR 10 MAX_USER_CONNECTIONS 3      |
+------------------------------------------------------------
```
Even more

- NO PAD collations
- INFORMATION_SCHEMA.USER_VARIABLES
- mysqlbinlog continuous backup
- mysqld --version=xxx

[server]
version=5.6.10-fake-as-a-three-dollar-bill
InnoDB 5.7

you have just heard all about it
New client library
and protocol enhancements
Client library

- MariaDB Connector/C
- for MySQL and MariaDB
- libmysqlclient API compatible
- LGPL
- OpenSSL/GnuTLS/schannel
Protocol Enhancements

- EXECUTE DIRECT
- session trackers
- bulk operations
Optimizer

Condition pushdown into non-mergeable derived tables and views
Using views: MERGE is good

CREATE TABLE t1 (a INT UNIQUE, b INT);
CREATE TABLE t2 (a INT UNIQUE, b INT);
CREATE VIEW v1 AS SELECT a+b AS c FROM t1 WHERE a>2;
EXPLAIN EXTENDED SELECT * FROM v1, t2 WHERE a=c AND c<100;

Note (Code 1003): select t1.a + t1.b AS c,t2.a,t2.b from t1 join t2
where t2.a = t1.a + t1.b and t1.a + t1.b < 100 and t1.a > 2
Using views: no MERGE is not good

```sql
CREATE VIEW v2 AS SELECT a+b AS c FROM t1 WHERE a > 2 GROUP BY c;
EXPLAIN EXTENDED SELECT * FROM v2, t2 WHERE a=c AND c<100;
```

```
+----+-------------+------------+------+----+--------------------------------------------+
<table>
<thead>
<tr>
<th>id</th>
<th>select_type</th>
<th>table</th>
<th>type</th>
<th></th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PRIMARY</td>
<td>t2</td>
<td>ALL</td>
<td></td>
<td>Using where</td>
</tr>
<tr>
<td>1</td>
<td>PRIMARY</td>
<td>&lt;derived2&gt;</td>
<td>ref</td>
<td></td>
<td>Using where</td>
</tr>
<tr>
<td>2</td>
<td>DERIVED</td>
<td>t1</td>
<td>ALL</td>
<td></td>
<td>Using where; Using temporary</td>
</tr>
</tbody>
</table>
+----+-------------+------------+------+----+--------------------------------------------+

3 rows in set, 1 warning (0.00 sec)

Note (Code 1003): select v2.c,t2.a,t2.b from v2 join t2 where t2.a = v2.c and v2.c < 100
Condition pushdown into non-mergeable views

```
SET optimizer_switch='condition_pushdown_for_derived=on';
EXPLAIN FORMAT=JSON SELECT * FROM v2, t2 WHERE a=c AND c<100;
...
  "materialized": {
    "query_block": {
      "temporary_table": {
        ...
        "filtered": 99.8,
        "attached_condition": "t1.a>2 and t1.a+t1.b<100"
      }
    }
  }
...
```
Performance
Performance improvements

- Fast connect
- CRC32 on P8
- Partitioned auto-scaling table cache
- Thread pool with prioritization
- Non-blocking ANALYZE TABLE
GSoC
GSoc

- NO PAD collations (Daniil Medvedev)
- Condition pushdown into non-mergeable views (Galina Shalygina)
- CREATE AGGREGATE FUNCTION (Varun Gupta)
- Long UNIQUE constraint (Shubham Barai, Sachin Setia)
- Invisible columns (Sachin Setia)
You too can contribute!

- **Code**: https://github.com/mariadb/server
- **KnowledgeBase**: https://mariadb.com/kb/
- **Bugs**: https://jira.mariadb.org/
- **IRC**: #maria on Freenode
- **Mailing list**: maria-discuss@lists.launchpad.net
- **Feedback plugin**: enable-feedback in my.cnf
Questions?
10.2.3

- JSON functions
- Indexes on virtual columns
- Multiple triggers (per time per event per table)