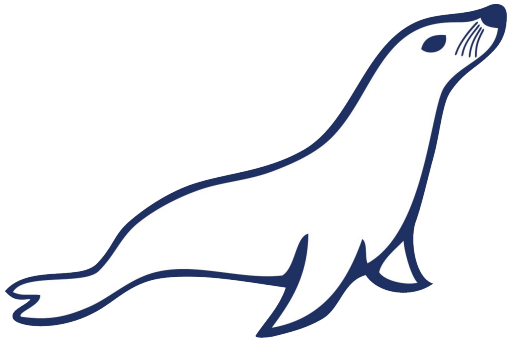


MariaDB®
FOUNDATION

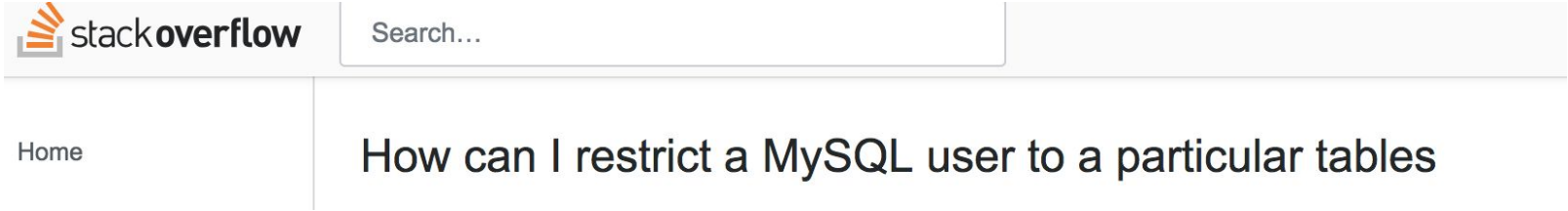
MariaDB 10.4 Reverse Privileges

Vicențiu Ciorbaru
Software Engineer @ MariaDB Foundation
vicentiu@mariadb.org

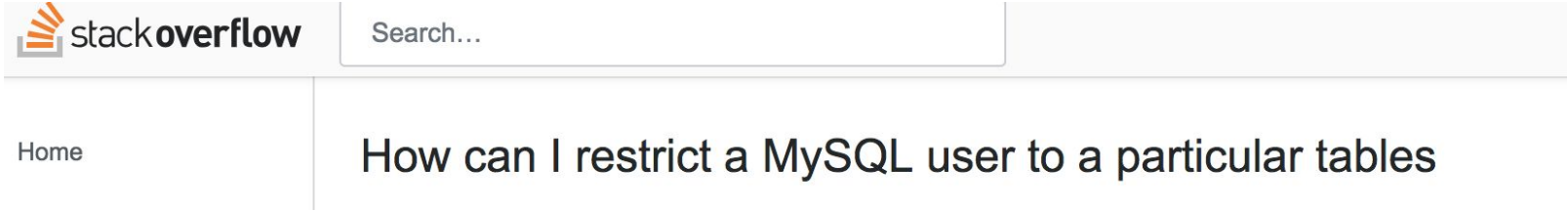
Agenda

- MariaDB privilege system
- Privilege system tables
- Drawbacks
- Reverse privileges

Where did the idea come from?

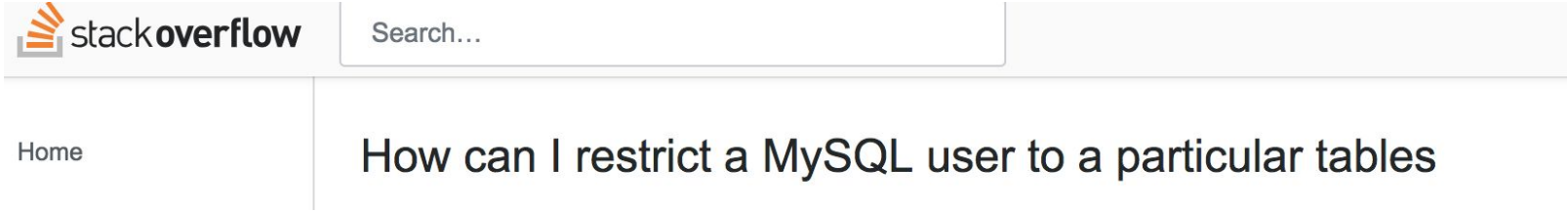


Where did the idea come from?




```
SELECT CONCAT("GRANT UPDATE ON db.", table_name, " TO user@localhost;")
FROM information_schema.TABLES
WHERE table_schema = "YourDB" AND table_name <> "table_to_skip";
```

Where did the idea come from?




```
SELECT CONCAT("GRANT UPDATE ON db.", table_name, " TO user@localhost;")
FROM information_schema.TABLES
WHERE table_schema = "YourDB" AND table_name <> "table_to_skip";
```

REALLY?!

Where did the idea come from?

There has to be a better way!

Where did the idea come from?

There has to be a better way!

Not really :(

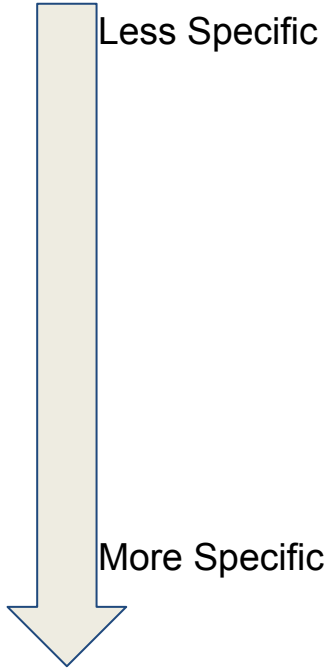
MariaDB access control internally

- MariaDB's tiered privilege system:

- Global Privileges
  - GRANT SELECT ON *.* TO SuperUser;
- Database (Schema) Privileges
  - GRANT SELECT ON wordpress.* TO WPAdmin;
- Table Privileges
  - GRANT UPDATE ON wordpress.users TO WPAdmin;
- Column Privileges
  - GRANT SELECT (user, content) ON wordpress.posts TO WPUser;
- Other Privileges

MariaDB access control internally

- MariaDB's tiered privilege system:



- Global Privileges
  - GRANT SELECT ON *.* TO SuperUser;
- Database (Schema) Privileges
  - GRANT SELECT ON wordpress.* TO WPAdmin;
- Table Privileges
  - GRANT UPDATE ON wordpress.users TO WPAdmin;
- Column Privileges
  - GRANT SELECT (user, content) ON wordpress.posts TO WPUser;
- Other Privileges

MariaDB access control internally

- When the server receives a query
  - Compute the necessary privileges to run the query
  - Are we running with privileges enabled?
  - Check the privilege cache
  - Check the user's global privileges
  - Check the user's database privileges
  - Check the user's table privileges
  - Check for specific column privileges.

  - Repeat the process for current active role, if any

MariaDB access control internally

- All privilege definitions are additive
- If you want to restrict access to a resource
  - You need to grant access to everything else BUT that resource…
- What happens when the schema changes?
  - Update all users…

MariaDB access control internally

- The more objects there are, the bigger the system tables get.
- The bigger the system tables are, the slower privilege checking becomes.
- Contributions to MySQL 8.0 from Eric Herman aimed to address a performance slowdown when things running with too many users.

https://mysql.wisborg.dk/2018/10/26/mysql-server-8-0-13-thanks-for-the-11-facebook-and-community-contributions/

Where did the idea come from?

- We need a solution to "block" access
  - Refactoring the application is not a "solution" in many cases
- Low overhead!
- Make it play nice with the current system and give more control to the DBA.

Where did the idea come from?

- Is there a precedent?

Where did the idea come from?

- Is there a precedent?
- SQL Server has similar functionality!

Where did the idea come from?

- Is there a precedent?
- SQL Server has similar functionality!

```
DENY   { ALL [ PRIVILEGES ] }
     | <permission>  [ ( column [ ,...n ] ) ] [ ,...n ]
    [ ON [ <class> :: ] securable ]
    TO principal [ ,...n ]
    [ CASCADE] [ AS principal ]
[;]
```

SQL Server Syntax

Where did the idea come from?

- Is there a precedent?
- SQL Server has similar functionality!

```
DENY   { ALL [ PRIVILEGES ] }
     | <permission>  [ ( column [ ,...n ] ) ] [ ,...n ]
    [ ON [ <class> :: ] securable ]
    TO principal [ ,...n ]
    [ CASCADE] [ AS principal ]
[;]
```

SQL Server Syntax

- But SQL server behaviour is hard to explain.
  - A grant will undo a deny. ?!

How shall we do it?

- We want to make it easy for users to understand.
- Easy to reason about it!
  - No "hidden" side effects.
  - No Host Table shenanigans
- Goal: be able to explain the solution in one sentence.
- Solution:
  - DENY will trump everything!

How shall we do it?

- Once a DENY command has been run, that user (role) will not have access to the resource, unless the DENY is specifically revoked.
- Any additional grants, aside from explicitly removing the DENY, can not grant access.

Denying privileges

- The project is still under development
  - New foundation developers - Rutuja Surve

- Syntax similar to GRANT / REVOKE:

DENY <priv> ON <resource> TO <user>

- How to undo a deny?

REVOKE DENY <priv> ON <resource> FROM <user>

Denying privileges

- The project is still under development
  - New foundation developers - Rutuja Surve

- Syntax similar to GRANT / REVOKE:

DENY <priv> ON <resource> TO <user>

- How to undo a deny?

REVOKE DENY <priv> ON <resource> FROM <user>

Can we do better?

Denying privileges

- How will the new algorithm work:

- Run a second pass through the user's grants
  - If there is a DENY present, the query will not be allowed (or will return partial results)
- Performance impact?
  - Practically none for GLOBAL, DATABASE and TABLE privileges (checked at the same time)
  - Slightly more work if specific column denies.
    - Offset by fewer column entries overall.

Denying privileges

- Details:
  - Only users with SUPER privilege or READ access to system tables can view denies with SHOW GRANTS.
  - Roles with DENY will block the user's grants too when active.
  - Host table is already removed in 10.4, any entries will be migrated via reverse privileges if possible.

Denying privileges

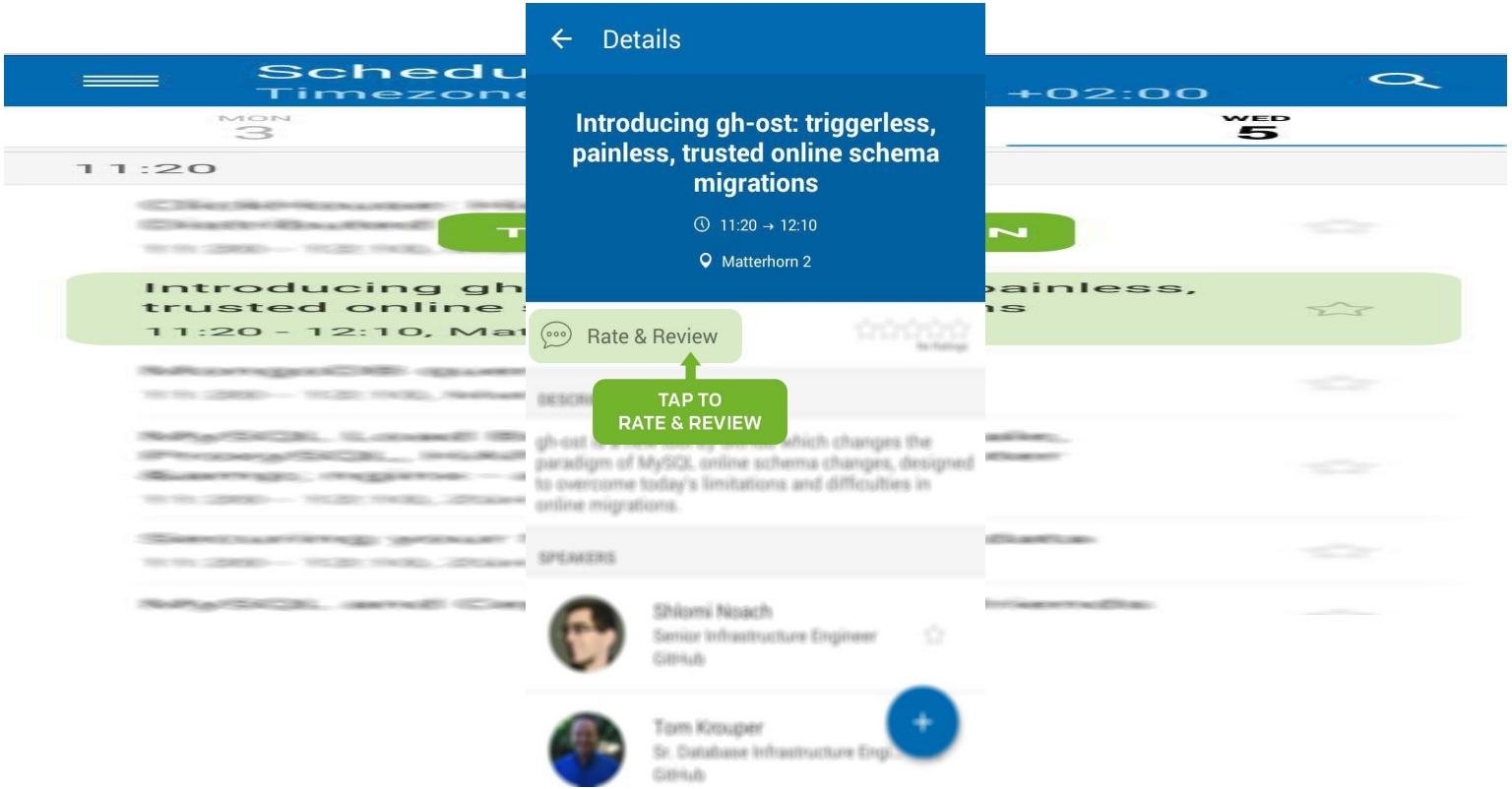
- MariaDB 10.4 is still in Alpha
- Feedback now can still be incorporated before Beta release.

Sponsors

MariaDB Foundation ensuring open development and collaboration.

We cannot fulfil our mission without our members and sponsors!

Rate My Session



Thank You!

Contact me at:
vicentiu@mariadb.org

Blog:
mariadb.org/blog