

Definisi

GRANT dan **REVOKE** adalah perintah dalam Database Control Language (DCL) yang digunakan untuk mengatur hak akses pengguna terhadap database.

- **GRANT** digunakan untuk memberikan hak akses kepada pengguna tertentu terhadap objek database seperti tabel, view, atau prosedur.
 - **REVOKE** digunakan untuk mencabut hak akses yang sebelumnya telah diberikan kepada pengguna.
-

Fungsi/Luaran

GRANT

- Memberikan hak akses kepada pengguna atau role tertentu.
- Memungkinkan pengelolaan database yang lebih aman dengan membatasi hak akses hanya kepada pengguna yang berwenang.

REVOKE

- Menghapus atau mencabut hak akses yang sebelumnya telah diberikan.
 - Mengontrol keamanan database dengan memastikan hanya pengguna yang berhak dapat melakukan perubahan.
-

Format Query

GRANT

```
GRANT {hak_akses} ON {nama_tabel} TO {nama_pengguna};
```

- **{hak_akses}** bisa berupa **SELECT**, **INSERT**, **UPDATE**, **DELETE**, dll.
- **{nama_tabel}** adalah nama tabel yang diberikan hak akses.
- **{nama_pengguna}** adalah nama pengguna yang menerima hak akses.

REVOKE

```
REVOKE {hak_akses} ON {nama_tabel} FROM {nama_pengguna};
```

- `{hak_akses}` adalah hak akses yang ingin dicabut.
 - `{nama_tabel}` adalah tabel yang terkait dengan hak akses tersebut.
 - `{nama_pengguna}` adalah pengguna yang hak aksesnya akan dicabut.
-

Pembuatan User Baru

```
CREATE USER '{nama_pengguna}'@'{host}' IDENTIFIED BY '{password}';
```

- `{nama_pengguna}` → Nama pengguna yang ingin dibuat (contoh: `user1`).
- `{host}` → Host yang diizinkan mengakses (contoh: `localhost` atau `%` untuk semua host).
- `{password}` → Kata sandi untuk pengguna tersebut (contoh: `password123`).

Contoh

```
CREATE USER 'user1'@'localhost' IDENTIFIED BY 'password123';
```

Pengguna `user1` akan dibuat dengan akses hanya dari `localhost`, menggunakan password `password123`.

Kalau mau cek pengguna yang sudah dibuat

```
SELECT user, host FROM mysql.user;
```

Hapus User

```
DROP USER '{nama_pengguna}'@'{host}';
```

Penjelasan:

- `{nama_pengguna}` → Nama user yang mau dihapus.
- `{host}` → Host pengguna (biasanya `localhost`).

Contoh

Menghapus user `user1` di `localhost`:

```
DROP USER 'user1'@'localhost';
```

Contoh Query Beserta Hasil dan Penjelasan

Tabel Utama

Sebelum menerapkan perintah *GRANT* dan *REVOKE*, kita buat tabel pegawai terlebih dahulu:

```
CREATE TABLE pegawai (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nama VARCHAR(50),  
    jabatan VARCHAR(50),  
    gaji INT  
);
```

```
INSERT INTO pegawai (nama, jabatan, gaji) VALUES  
( 'Andi', 'Manager', 10000000),  
( 'Budi', 'Staff', 5000000),  
( 'Siti', 'Admin', 4000000);
```

Hasilnya jika kita menjalankan perintah:

```
MariaDB [materi]> SELECT * FROM pegawai;  
+----+-----+-----+-----+  
| id | nama | jabatan | gaji |  
+----+-----+-----+-----+  
| 1  | Andi | Manager | 10000000 |  
| 2  | Budi | Staff   | 5000000  |  
| 3  | Siti | Admin   | 4000000  |  
+----+-----+-----+-----+  
3 rows in set (0.032 sec)
```

Pemberian Hak Akses (GRANT)

Misalkan kita memiliki pengguna *user1* dan kita ingin memberinya hak akses hanya bisa melihat data pegawai.

Penjelasan

GRANT digunakan untuk **memberikan izin (privilege)** kepada pengguna untuk melakukan tindakan tertentu pada database atau tabel, seperti membaca data, memasukkan data, atau mengubah data.

Format Query

```
GRANT [jenis_hak_akses] ON [nama_tabel] TO '[nama_pengguna]'@'[host]';
```

Contoh Query

```
GRANT SELECT ON pegawai TO 'user1'@'localhost';
```

Analisis

- **GRANT** → Memberikan izin/hak akses ke pengguna tertentu.
- **SELECT** → Hak akses untuk **membaca data** (melihat isi tabel) tanpa bisa mengubahnya.
- **ON pegawai** → Menentukan tabel yang akan diakses, dalam hal ini tabel bernama pegawai.
- **TO 'user1'@'localhost'** → Menentukan pengguna (user1) yang hanya bisa mengakses database dari **komputer lokal** (localhost).

Hasil

```
ACER@LAPTOP-P323Q91L c:\xampp
# mysql -u user1 -p
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 11
Server version: 10.4.32-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> USE materi;
Database changed
MariaDB [materi]> SELECT * FROM pegawai;
+----+-----+-----+-----+
| id | nama | jabatan | gaji |
+----+-----+-----+-----+
| 1  | Andi | Manager | 10000000 |
| 2  | Budi | Staff   | 5000000  |
| 3  | Siti | Admin   | 4000000  |
+----+-----+-----+-----+
3 rows in set (0.003 sec)
```

disini saya mencoba **INSERT** tetapi gagal itu dikarenakan User1 tidak memiliki Hak Akses.

```
MariaDB [materi]> INSERT INTO pegawai (nama, jabatan, gaji) VALUES ('Dewi', 'Kasir', 4500000);
ERROR 1142 (42000): INSERT command denied to user 'user1'@'localhost' for table 'materi`.`pegawai`
MariaDB [materi]>
```

Artinya, user1 hanya dapat membaca data di tabel pegawai, tetapi tidak dapat menambah, mengubah, atau menghapus data.

Pencabutan Hak Akses (REVOKE)

Jika kita ingin mencabut hak akses `SELECT` dari `user1` , maka perintahnya:

Penjelasan

`REVOKE` digunakan untuk mencabut atau menghapus hak akses dari pengguna tertentu.

Format Query

```
REVOKE [jenis_hak_akses] ON [nama_tabel] FROM '[nama_pengguna]'@'[host]';
```

Contoh Query

```
REVOKE SELECT ON pegawai FROM 'user1'@'localhost';
```

Analisis

- `REVOKE` Menghapus atau mencabut izin/hak akses dari pengguna tertentu.
- `SELECT` Jenis hak akses yang dicabut (dalam hal ini, hak untuk melihat data).
- `ON pegawai` Menentukan tabel yang terkait, yaitu tabel `pegawai` .
- `FROM 'user1'@'localhost'` Menentukan pengguna yang akan dicabut haknya, dalam hal ini `user1` yang terhubung dari `localhost` (komputer server itu sendiri).

Hasil

```
MariaDB [materi]> REVOKE SELECT ON pegawai FROM 'user1'@'localhost';  
Query OK, 0 rows affected (0.006 sec)
```

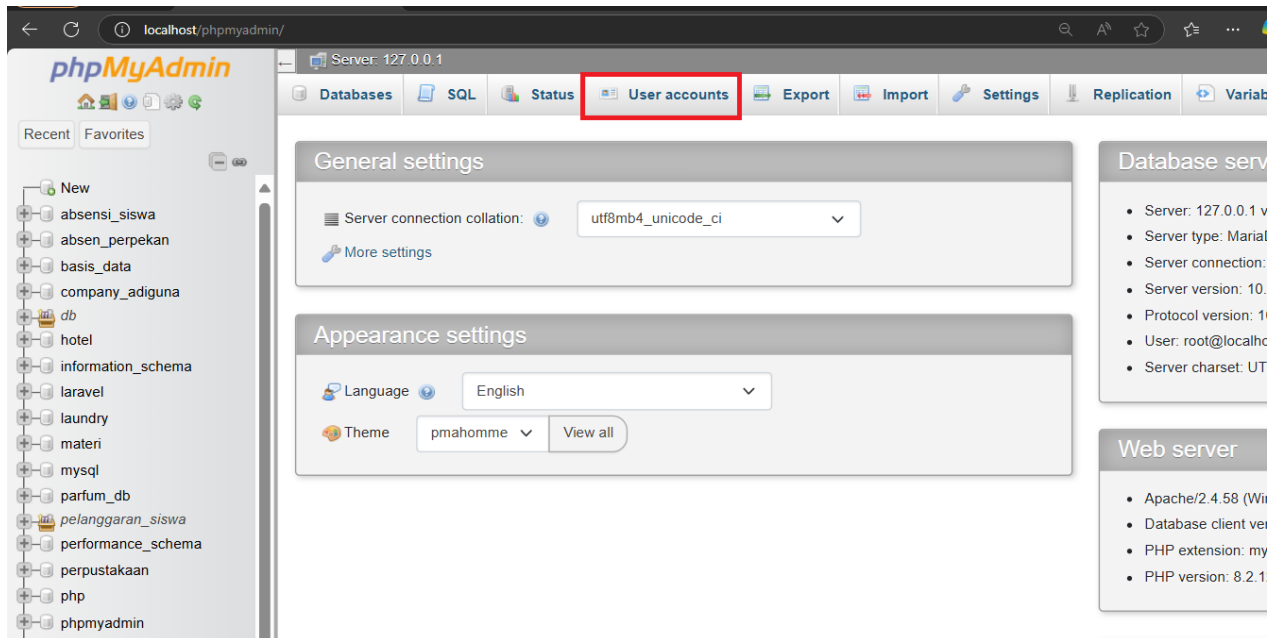
User1 sudah tidak dapat akses untuk melihat tabel `Pegawai` .

```
ACER@LAPTOP-P323Q91L c:\xampp  
# mysql -u user1 -p  
Enter password: *****  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 13  
Server version: 10.4.32-MariaDB mariadb.org binary distribution  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> USE materi  
ERROR 1044 (42000): Access denied for user 'user1'@'localhost' to database 'materi'  
MariaDB [(none)]>
```

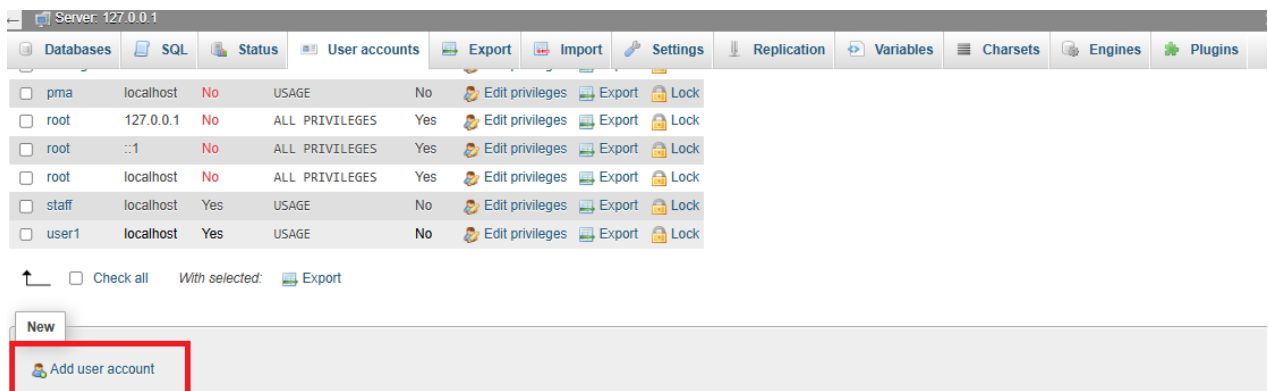
Langkah-Langkah Pembuatan GRANT & REVOKE di phpMyAdmin

Berikut adalah langkah-langkah spesifik untuk memberikan dan mencabut hak akses pengguna di *phpMyAdmin*.

1. Buka browser dan akses localhost/phpmyadmin setelah itu masuk ke User accounts



2. Klik "Add user account".



3. Masukkan detail pengguna baru:
 - *User name*: user1
 - *Host name*: Pilih "Local" (localhost).
 - *Password*: Masukkan password (misalnya password123).
4. Pada bagian *Global Privileges*, biarkan kosong jika hanya ingin memberikan akses ke tabel tertentu.

5. Klik "Go" untuk menyimpan.

Server: 127.0.0.1

Databases SQL Status User accounts Export Import Settings Replication Variables Charsets Engines Plugins

Add user account

Login Information

User name: Use text field manager

Host name: Local localhost

Password: Use text field *** Strength: Extremely weak

Re-type:

Authentication plugin: Native MySQL authentication

Generate password: Generate

6. Setelah kita buat akun kita kembali ke User accounts untuk melihat apakah user yang dibuat sudah ada

Server: 127.0.0.1

Databases SQL Status User accounts Export Import Settings Replication Variables Charsets Engines Plugins

User accounts overview

User name	Host name	Password	Global privileges	Grant	Action
<input type="checkbox"/> manager	localhost	Yes	USAGE	No	Edit privileges Export Lock
<input type="checkbox"/> pma	localhost	No	USAGE	No	Edit privileges Export Lock
<input type="checkbox"/> root	127.0.0.1	No	ALL PRIVILEGES	Yes	Edit privileges Export Lock
<input type="checkbox"/> root	:::1	No	ALL PRIVILEGES	Yes	Edit privileges Export Lock
<input type="checkbox"/> root	localhost	No	ALL PRIVILEGES	Yes	Edit privileges Export Lock
<input type="checkbox"/> staff	localhost	Yes	USAGE	No	Edit privileges Export Lock
<input type="checkbox"/> user1	localhost	Yes	USAGE	No	Edit privileges Export Lock

Check all With selected: Export

New

Add user account

7. Selanjutnya kita buka SQL pada bagian navbar untuk memberikan Hak akses kepada manager, disini manager kita berikan akses hanya dapat melihat (SELECT).

berikut querynya.

Edit privileges: User account 'manager'@'localhost'

The screenshot shows the 'Global privileges' section of the phpMyAdmin interface. At the top, there is a 'Global privileges' header with a 'Check all' button. Below it, a note states: 'Note: MySQL privilege names are expressed in English.' The interface is divided into four main columns: 'Data', 'Structure', 'Administration', and 'Resource limits'. The 'Data' column has a 'Data' sub-header and contains a list of privileges: SELECT (checked), INSERT, UPDATE, DELETE, and FILE. The 'Structure' column has a 'Structure' sub-header and contains a list of privileges: CREATE, ALTER, INDEX, DROP, CREATE TEMPORARY TABLES, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EXECUTE, CREATE VIEW, and EVENT. The 'Administration' column has an 'Administration' sub-header and contains a list of privileges: GRANT, SUPER, PROCESS, RELOAD, SHUTDOWN, SHOW DATABASES, LOCK TABLES, REFERENCES, REPLICATION CLIENT, REPLICATION SLAVE, and CREATE USER. The 'Resource limits' column has a 'Resource limits' sub-header and contains a note: 'Note: Setting these options to 0 (zero) disables the option.' Below the note are four input fields: 'MAX QUERIES PER HOUR' (0), 'MAX UPDATES PER HOUR' (0), 'MAX CONNECTIONS PER HOUR' (0), and 'MAX USER_CONNECTIONS' (0).

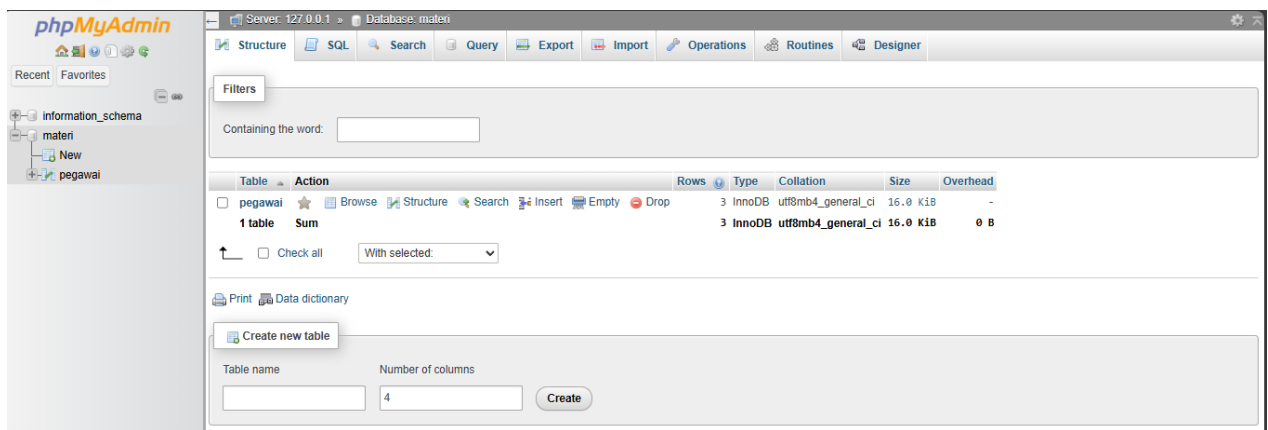
8. Berikut Akses yang di berikan kepada manager, hanya dapat SELECT materi tabel pegawai`

The screenshot shows the 'Grants for manager@localhost' page in phpMyAdmin. It displays the SQL command: 'GRANT SELECT ON *.* TO `manager`@`localhost` IDENT...'. Below the command, there is a 'Query results operations' section.

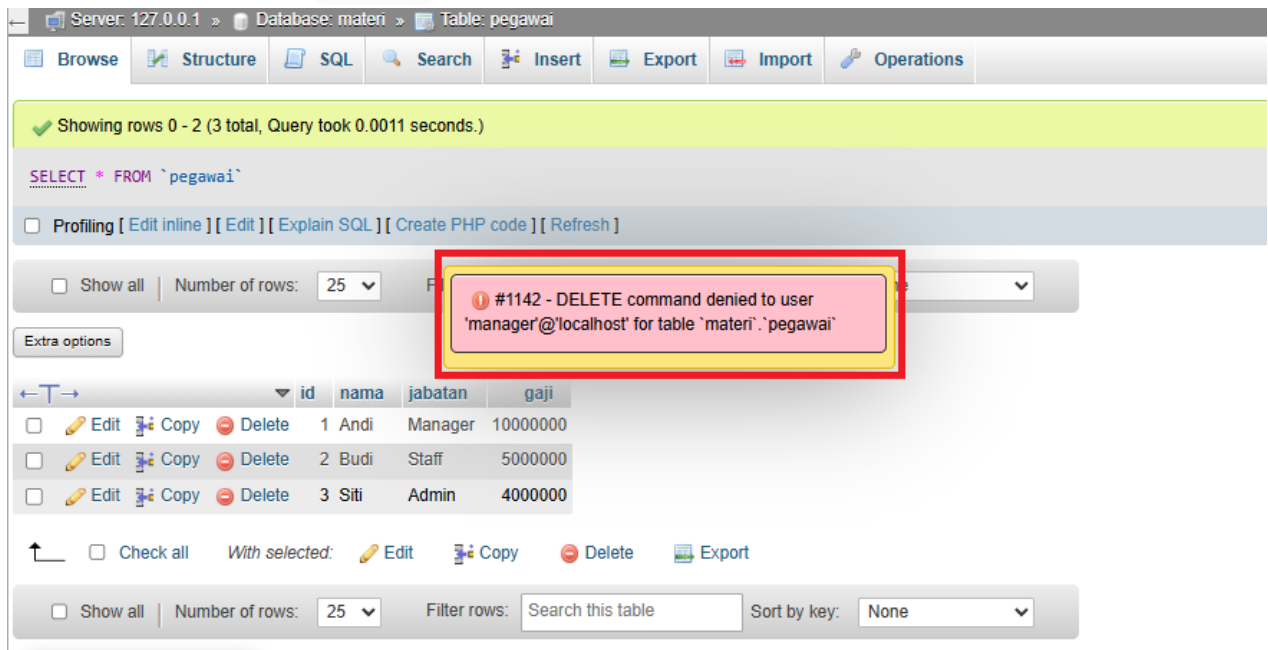
9. Selanjutnya kita login ke phpMyadmin menggunakan User yang sudah dibuat yaitu manager.

The screenshot shows the phpMyAdmin login page. At the top, there is a logo for 'phpMyAdmin' and the text 'Welcome to phpMyAdmin'. Below the logo, there is a 'Language' dropdown menu set to 'English'. Below the language menu, there is a 'Log in' button. Below the 'Log in' button, there are two input fields: 'Username:' with the value 'manager' and 'Password:' with a masked password. At the bottom right, there is a 'Log in' button.

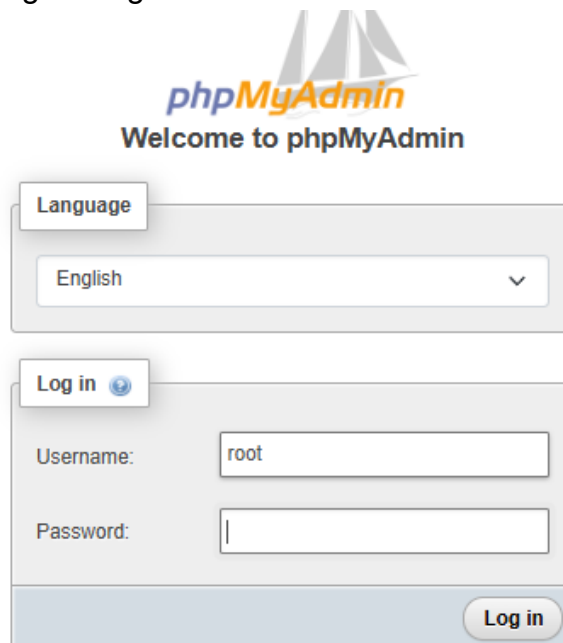
10. Pada halaman Manager hanya terdapat 1 database dan 1 tabel, itu karna kita sudah atur hak akses yang diberikan kepada manager.



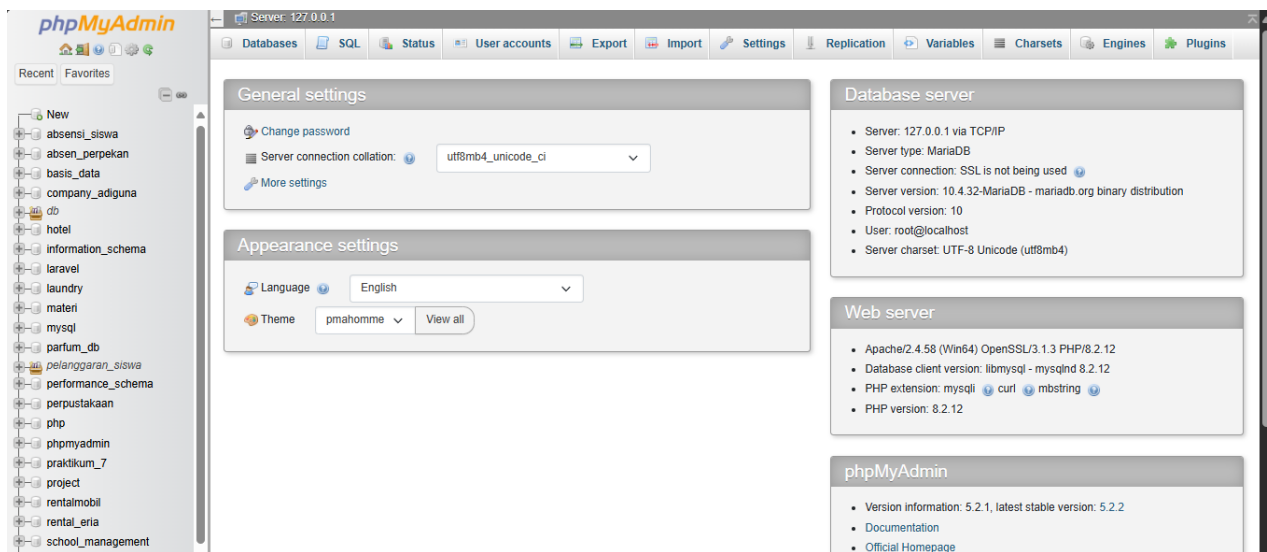
11. Hak akses yang diberikan kepada manager hanya `SELECT`, makanya terdapat pesan error ketika mencoba `INSERT` data.



12. Selanjutnya kita coba login dengan user root.

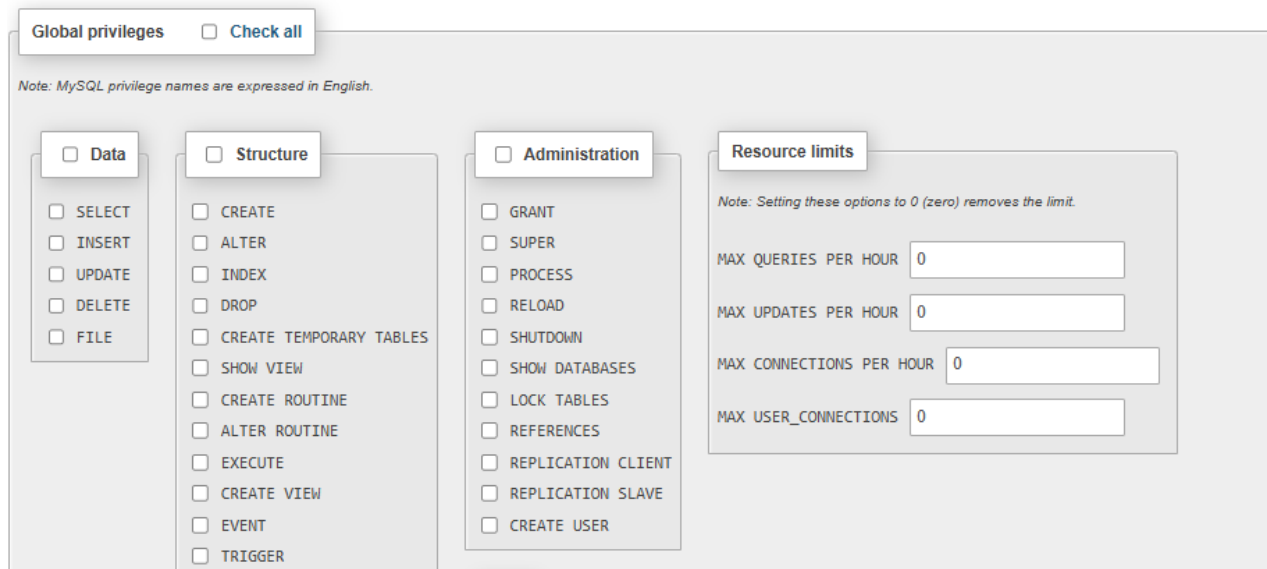


13. Pada halaman User root kita masuk ke bagian `SQL`, untuk menghapus Hak Akses manager yang sudah diberikan sebelumnya.

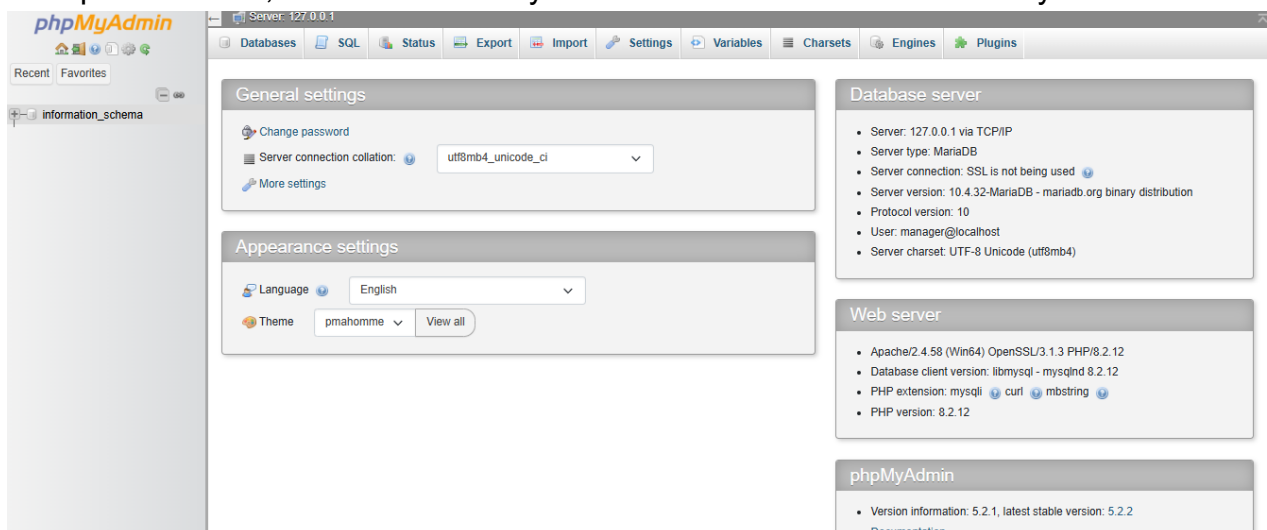


14. Untuk menghapus Hak Akses manager kita hanya perlu kosongkan hak akses apa yang ingin dicabut

Edit privileges: User account '*manager*'@'*localhost*'



15. Selanjutnya kita kembali Masuk ke Manager, pada halaman Manager sudah tidak terdapat database, karena sebelumnya kita sudah mencabut hak aksesnya.



Dengan mengikuti langkah-langkah ini, kita dapat mengatur hak akses database dengan aman menggunakan phpMyAdmin.

Studi Kasus

Kasus: Pengaturan Hak Akses Karyawan dalam Perusahaan

Sebuah perusahaan ingin mengelola akses ke database pegawai:

- *HR Manager* bisa melihat dan mengubah data pegawai.
- *Staff* hanya bisa melihat data pegawai.

Solusi Menggunakan GRANT dan REVOKE

1. Membuat Pengguna Baru

```
CREATE USER 'hr_manager'@'localhost' IDENTIFIED BY 'password123';  
CREATE USER 'staff'@'localhost' IDENTIFIED BY 'password456';
```

- **CREATE USER** : Perintah ini digunakan untuk membuat akun pengguna baru di MySQL.
- **'hr_manager'@'localhost'** : Membuat pengguna dengan nama `hr_manager` , yang hanya bisa login dari `localhost` .
- **'staff'@'localhost'** : Membuat pengguna dengan nama `staff` , juga hanya bisa login dari `localhost` .
- **IDENTIFIED BY** : Mengatur password untuk pengguna.

Ini artinya perusahaan punya dua pengguna: `hr_manager` dan `staff` , masing-masing dengan password mereka sendiri.

2. Memberikan Hak Akses ke HR Manager

```
GRANT SELECT, UPDATE ON pegawai TO 'hr_manager'@'localhost';
```

- **GRANT** : Memberikan hak akses ke pengguna.
- **SELECT** : Mengizinkan pengguna untuk membaca data dalam tabel.
- **UPDATE** : Mengizinkan pengguna untuk mengubah data dalam tabel.
- **ON pegawai** : Menentukan tabel yang diberi hak akses, dalam hal ini tabel `pegawai` .
- **TO 'hr_manager'@'localhost'** : Menentukan pengguna yang akan menerima hak akses tersebut.

Dengan ini, `hr_manager` bisa melihat dan mengubah data pegawai.

3. Memberikan Hak Akses ke Staff

```
GRANT SELECT ON pegawai TO 'staff'@'localhost';
```

- **GRANT SELECT** : Memberikan hak akses hanya untuk membaca data dalam tabel.
- **ON pegawai** : Mengatur bahwa hak akses ini hanya berlaku untuk tabel `pegawai`.
- **TO 'staff'@'localhost'** : Memberikan hak akses ini ke pengguna `staff`.

Artinya, `staff` hanya bisa melihat data pegawai, tanpa bisa mengubahnya.

4. Menghapus Hak Akses Staff

```
REVOKE SELECT ON pegawai FROM 'staff'@'localhost';
```

- **REVOKE** : Menghapus hak akses yang telah diberikan sebelumnya.
- **SELECT** : Menentukan hak akses yang ingin dicabut.
- **ON pegawai** : Menentukan tabel yang hak aksesnya dicabut.
- **FROM 'staff'@'localhost'** : Menentukan pengguna yang hak aksesnya akan dicabut.

Jika ada masalah keamanan atau perubahan kebijakan, perusahaan bisa langsung mencabut akses `staff`.