# MySQL - Berechtigungssystem

**MySQL** hat ein fortschrittliches, aber vom Standard abweichendes Sicherheitssystem. Die Zugriffssteuerung auf einen MySQL – Server erfolgt immer in 2 Phasen.

| | |
|---|---|
| **PHASE 1**<br>**Verbindungsprüfung** | Der Server überprüft, ob Sie das Recht haben, sich verbinden zu dürfen (wird nur einmal bei Verbindungsaufbau durchgeführt). |
| **PHASE 2**<br>**Anfrageprüfung** | Der Server überprüft, ob Sie ausreichende Rechte haben, diese Abfrage auszuführen (wird bei jeder Abfrage durchgeführt). |

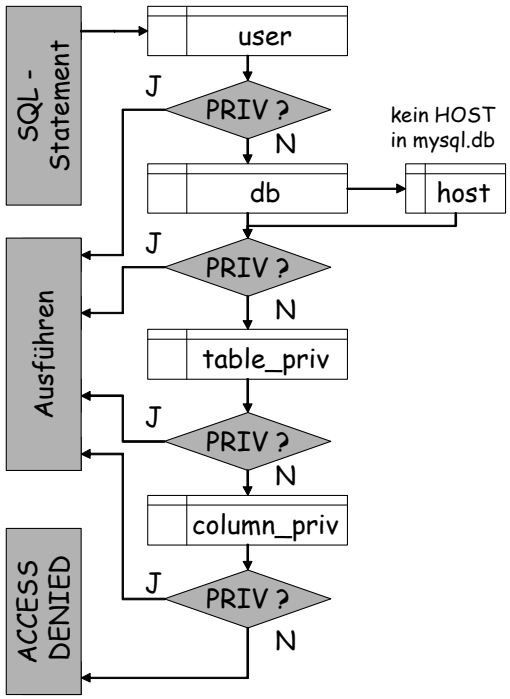
Die Rechte (Privileges) werden in einer MySQL-Datenbank gespeichert. Die wesentlichen Tabellen heißen **mysql.user**, **mysql.db**, **mysql.host** bzw. **mysql.tabl_priv** und **mysql.column_priv**. Der Server benutzt diese Tabellen in beiden Phasen der Zugriffskontrolle. Wesentliche PRIVILEGES auf Datenbanken, Tabellen und Spalten in diesen Berechtigungs-tabellen sind nebenstehend dargestellt.

| Berechtigung | Kontext |
|---|---|
| select | Tabellen |
| insert | Tabellen |
| update | Tabellen |
| delete | Tabellen |
| index | Tabellen |
| alter | Tabellen |
| create | Datenbanken, Tabellen oder Index |
| drop | Datenbanken oder Tabellen |
| references | Datenbanken oder Tabellen |
| reload | Serververwaltung |
| shutdown | Serververwaltung |
| process | Serververwaltung |

Jeder Zugriff erfolgt mit der Benutzeridentifizierung der Form „user@host“ (mit Passwort)

**Verbindungsprüfung:** In der 1.Phase werden nur die Informationen aus der **mysql.user** genutzt. Da theoretisch mehrere Einträge für eine Useridentikation zutreffen können wird sie vor dem Zugriff sortiert (Einträge, welche den HOST am genauesten spezifizieren zuerst). Der 1. passende Eintrag zum user@host wird zur Verbindungsprüfung herangezogen.

**Anfrageprüfung:** Bei jeder Anfrage (SQL-Statement) werden die Tabellen nach dem neben-stehenden Schema auf passende Einträge (PRIVILEGES) durchsucht.



## Hinweise :

- PRIVILEGES können durch direkte Manipulation (INSERT, DELETE bzw. UPDATE) in den entsprechenden Tabellen geändert werden (was aber komplizierter ist).
- Globale PRIVILEGES werden vor lokalen ausgewertet.
- Alle Tabellen werden beim Server-Start sortiert ins RAM geladen, so dass alle Änderungen mit FLUSH PRIVILEGES aktualisiert werden müssen.
- Auf LINUX/UNIX - Systemen wird zwischen TCP/IP - und SOCKET (localhost) - Verbindungen unterschieden ,weshalb für diese USER auch 2 Einträge notwendig sind.
- Datenbanken 'test_%' sind grundsätzlich für alle berechtigten User (Phase 1) frei zugänglich.
- Nach einer Neuinstallation ist die Rechte-Einstellung (systemabhängig) sehr freizügig.

  WINDOWS: User 'root' hat (ohne Passwort) alle Rechte von überall (lokal und Netz).

  LINUX: User 'root' hat (ohne Passwort) alle Rechte von 'localhost' und 'hostname'. Alle 'user' des Systems können sich (ohne Rechte) anmelden.

## SQL Statements der DCL (Data Control Language) zur Rechte-Vergabe

```
① GRANT PRIVILEGES
② ON OBJECT
③ TO USER
④ [IDENTIFIED BY] 'PASSWORD'
⑤ [WITH GRANT OPTION];
```

GRANT – Syntax

ALL (alle RECHTE) / USAGE (kein RECHT)
INSERT,SELECT, . . .

Database.Tabelle / Database.* / *.*

user@host / ' '@host / user@'%' / ' '@'%'

Erstellt (ggf.) User und erteilt PRIVILEGES.

---

```
① REVOK PRIVILEGES
② ON OBJECT
③ FROM USER ;
```

REVOKE – Syntax

Entzieht PRIVILEGES entsprechend der Spezifikation.

---

```
① SHOW GRANTS FOR USER ;
```

SHOW – Syntax

Zeigt aktuelle Rechte des Users an.

---

```
① FLUSH PRIVILEGES;
```

FLUSH – Syntax

Sortiert und lädt die mysql-DB neu ins RAM.

---

```
① SET PASSWORD
② [FOR ' USER@HOST' ]
③ = PASSWORD ('NEUES_PASS' ) ;
```

SET – Syntax

Ändert Passwort des angemeldeten (oder angegebenen) users.