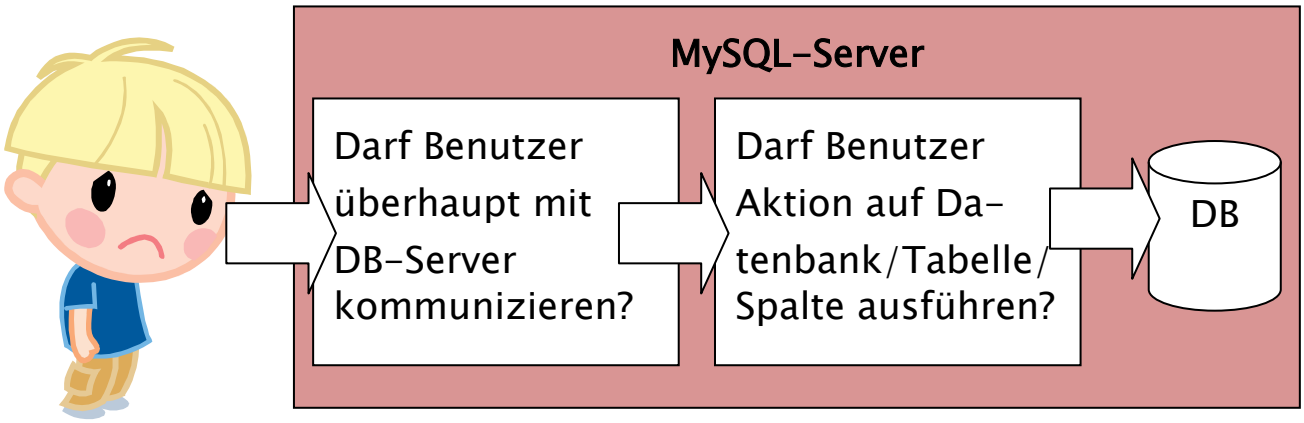
DB: Zugriffsverwaltung, Sicherheit

Ziele

- Wissen nach welchem Konzept die Benutzerauthentifizierung in MySQL erfolgt.
- Wissen wie man am Server Überblick darüber erhält, welche Benutzer wie mit den gespeicherten Daten arbeiten können.
- Das Zusammenwirken zwischen globalen und lokalen Benutzerrechten verstehen.
- Benutzer anlegen, ändern und löschen können.
- Den Benutzern Rechte auf den Datenbank-Server, auf einzelne Datenbanken, auf Tabellen oder nur auf einzelne Datenfelder geben und diese Rechte wieder nehmen können.

Konzept

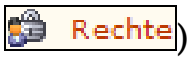


1. Schritt: Darf Benutzer mit DB-Server kommunizieren?

Auswertung von

- *Benutzername*
  maximal 16 Zeichen lang, Groß-/Kleinschreibung wird berücksichtigt.
- *Passwort*
  beliebige Länge, Groß-/Kleinschreibung wird berücksichtigt, verschlüsselt abgespeichert in 45 Bytes, Rekonstruktion unmöglich.
- *Hostname*
  Rechnername von dem aus sich Benutzer anmeldet.

Benutzer und Rechte in Systemdatenbank namens `mysql` gespeichert.

Tabelle `mysql.user` enthält beispielsweise folgende Informationen (aufbereitet in **phpMyAdmin** durch 

):

User	Host	Password	Globale Rechte	Grant
root[1]	localhost	4825d…	ALL PRIVILEGES	Ja
pma[2]	localhost	""	SHUTDOWN	Nein
""[3]	localhost	""	SELECT	Nein
rudi[4]	gobbz.net	545d1…	SELECT, INSERT, UPDATE	Nein
sepp[5]	%	""	USAGE	Nein
sepp[6]	localhost	""	USAGE	Nein

1) `root` darf sich vom Rechner auf dem DB-Server läuft mit Passwort anmelden, hat alle Rechte auf alle Datenbanken und kann seine Rechte weiter geben.

2) `pma` darf sich vom Rechner auf dem DB-Server läuft ohne Passwort anmelden, hat als globales Recht nur das Herunterfahren des Servers und darf seine Rechte nicht weiter geben.

3) Jeder darf sich vom Rechner auf dem DB-Server läuft ohne Passwort anmelden (Benutzername muss nicht in `mysql.user` enthalten sein) und hat das `SELECT`-Recht auf alle Datenbanken (*anonymer Benutzer*).

4) `rudi` darf sich von einem Rechner der Domäne `gobbz.net` aus mit Passwort anmelden und hat angegebene Rechte auf alle Datenbanken (**ANMERKUNG**: Login von `rudi` auf Rechner des DB-Servers mit Passwort funktioniert nicht, Login von `rudi` auf Rechner des DB-Servers ohne Passwort würde [3] hervorrufen).

5) sepp darf sich von allen Rechnern aus – auch vom Rechner des DB-Servers – ohne Passwort anmelden und hat keine Rechte (`USAGE`) auf alle Datenbanken.

6) sepp darf sich vom Rechner des DB-Servers ohne Passwort anmelden und hat keine Rechte auf alle Datenbanken.

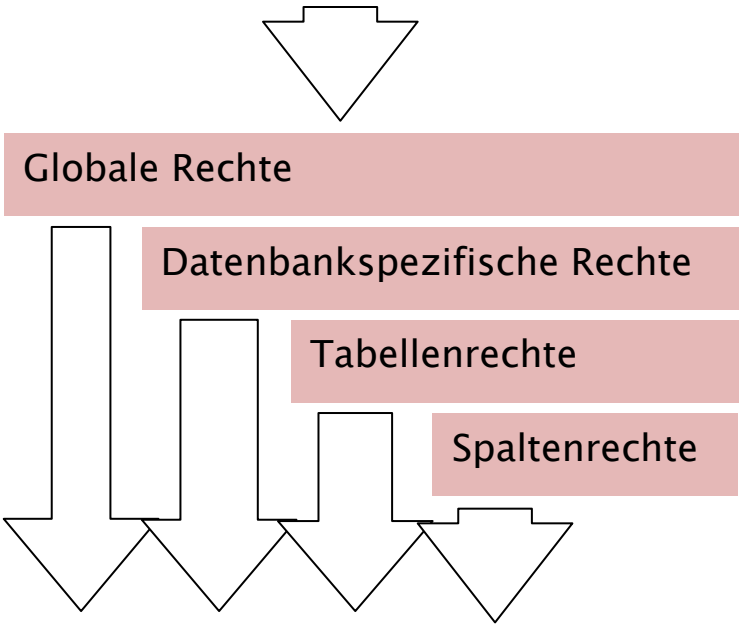
Benutzer anlegen und löschen

```
CREATE USER forumadmin@localhost
  IDENTIFIED BY "forumadmin";
CREATE USER forumuser@localhost
  IDENTIFIED BY "forumuser";
SET PASSWORD
  FOR forumuser@localhost = PASSWORD("forumuserneu");
DROP USER forumuser@localhost;
SHOW GRANTS FOR forumadmin@localhost;
```

Der so angelegte Benutzer hat keinerlei Rechte auf Datenbanken.

Rechte vergeben und nehmen

Mögliche Rechte: **ALL**, **USAGE**, **CREATE**, **ALTER**, **DROP**, **INSERT**, **UPDATE**, **DELETE**, **SELECT**, **LOCK TABLES**, **ALL PRIVILEGES**, **CREATE USER** usw.



Hierarchische Kontrolle:

Was global erlaubt wurde, kann auf Datenbank-/ Tabellen/ Spaltenebene nicht mehr verboten werden

Globale Privilegien

```
GRANT UPDATE
  ON *.*
  TO forumadmin@localhost;
```

Datenbankspezifische Privilegien

```
GRANT ALL
  ON forum.*
  TO forumadmin@localhost WITH GRANT OPTION;
```

```
GRANT SELECT, INSERT, DELETE
  ON forum.*
  TO forumuser@localhost;
```

Tabellenrechte

```
REVOKE SELECT, INSERT, DELETE
  ON forum.privateforen
  FROM forumuser@localhost;
```
[1)]

Spaltenrechte

```
GRANT SELECT(titel, beschreibung)
  ON forum.forenbeitraege
  TO ''@localhost;
```
[2)]

1) Keine Auswirkung, weil Rechte bereits auf Datenbankebene vergeben wurden.

2) Keine Auswirkung, weil Benutzer bereits globales **SELECT**-Recht hat (siehe abgebildete Tabelle `mysql.user`).