LU12c: MySQL – Rollen & Berechtigungen im Notenbuch

Lernziele dieser Seite

Sie können …

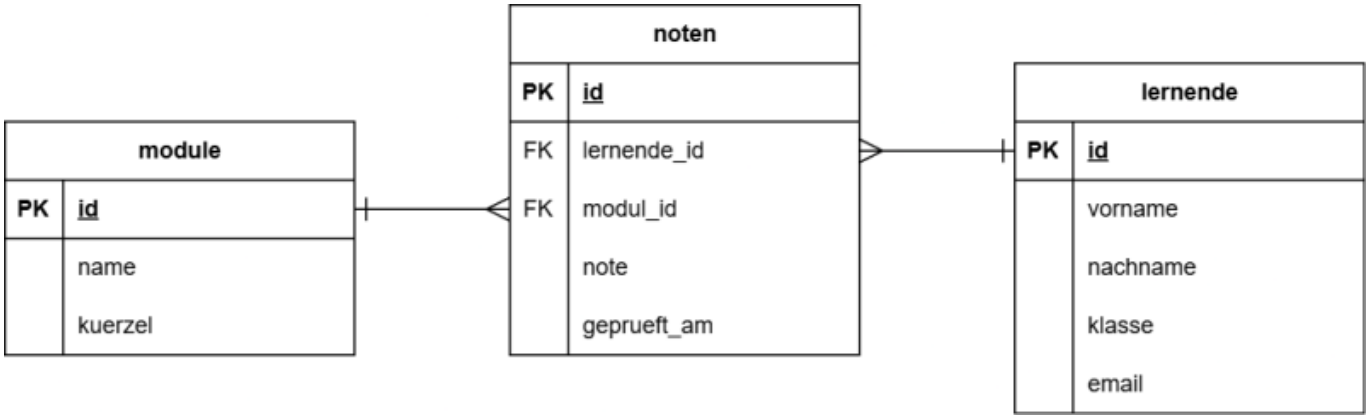
- in einer einfachen Noten-Datenbank zwei sinnvolle Rollen definieren (Administration und Lernende).
- mit den Befehlen `CREATE USER`, `CREATE ROLE`, `GRANT`, `SHOW GRANTS`, `SET PERSIST activate_all_roles_on_login` und `REVOKE` bzw. `DROP USER`/`DROP ROLE` in MySQL umgehen.
- erklären, wie das Prinzip der minimalen Privilegien mit Rollen umgesetzt werden kann.

Ausgangslage: Noten-Datenbank

Zum Nachvollziehen der Übung laden Sie hier das SQL-Skript herunter, welches eine entsprechende Datenbank mit den Tabellen erzeugt.

SQL-File mit Setup einer Noten-Datenbank.

Stellen Sie sich die Datenbank `noten_db` vor, in der die Noten aller Lernenden gespeichert sind. Eine zentrale Tabelle dafür ist `noten`.



Die Datenbank ist mit drei Tabellen aufgebaut: `noten`, `lernende` und `module` und über Fremdschlüssel miteinander verknüpft.

In der Tabelle `noten` liegen besonders schützenswerte Personendaten (Leistungsbeurteilungen). Entsprechend wichtig ist eine saubere Rechtevergabe.

Schritt 1: Zwei Rollen definieren

Wir arbeiten bewusst nur mit **zwei Rollen**:

- eine Admin-Rolle für die vollständige Verwaltung der Noten-Datenbank,
- eine Rolle für Lernende, die alle Noten lesen dürfen (aber nichts verändern).

Rolle	Zweck	Typische Rechte
`db_admin`	Verwaltung der gesamten Noten-Datenbank	alle Rechte auf `noten_db.*`
`lernende_role`	Lernende sehen alle Noten (Leserechte)	SELECT auf Tabellen in `noten_db`

Zuerst werden die Rollen in MySQL angelegt:

```
CREATE ROLE db_admin@localhost;
CREATE ROLE lernende_role@localhost;
```

Schritt 2: Rechte an Rollen vergeben (GRANT)

Nun legen Sie fest, was die beiden Rollen genau dürfen.

Rolle db_admin – Alle Rechte

```
GRANT ALL PRIVILEGES
ON noten_db.*
TO db_admin@localhost
WITH GRANT OPTION;
```

Die Rolle `db_admin` kann damit:

- Tabellen anlegen, ändern, löschen,
- alle Noten lesen, ändern und löschen,
- Rechte an andere vergeben.

Rolle lernende_role – Noten lesen, aber nichts verändern

Alle Lernenden sollen alle Noten sehen können, aber keine Noten verändern oder löschen. Dafür

erhält die Rolle nur Leserechte (SELECT).

```
GRANT SELECT
ON noten_db.*
TO lernende_role@localhost;
```

Damit gilt:

- `lernende_role` darf alle Daten in `noten_db` lesen,
- aber keine Datensätze einfügen, ändern oder löschen,
- und auch keine Tabellenstruktur verändern.

Sie setzen damit das Prinzip der minimalen Privilegien um: Lernende dürfen genau das, was sie für ihren Zweck benötigen – nicht mehr.

Schritt 3: Zwei MySQL-Benutzer erstellen

Jetzt legen Sie genau **zwei MySQL-Benutzer** an:

- `lehrperson_koch` – Lehrperson/Administrator der Noten-Datenbank,
- `lernende_caduff` – Beispiel-Benutzer für Lernende (für das Unterrichtsbeispiel ist es in Ordnung, dass alle Lernenden darüber alle Noten sehen).

```
CREATE USER 'lehrperson_koch'@'localhost'
  IDENTIFIED BY 'Admin!2025';

CREATE USER 'lernende_caduff'@'localhost'
  IDENTIFIED BY 'Lernende!2025';
```

In der Praxis sollten Sie selbstverständlich stärkere und individuelle Passwörter verwenden. Hier geht es um das Prinzip.

Schritt 4: Rollen an Benutzer zuweisen (GRANT)

Im nächsten Schritt weisen Sie den Benutzern ihre Rollen zu:

```
GRANT db_admin@localhost
TO 'lehrperson_koch'@'localhost';

GRANT lernende_role@localhost
```

```
TO 'lernende_caduff'@'localhost';
```

Damit gilt:

- `lehrperson_koch` arbeitet mit allen Rechten der Rolle `db_admin`,
- `lernende_caduff` arbeitet mit den Leserechten der Rolle `lernende_role`.

Wenn Sie später eine weitere Lehrperson oder einen weiteren Admin hinzufügen, müssen Sie nur einen neuen Benutzer erstellen und ihm oder ihr die passende Rolle zuweisen.

Schritt 5: Rollen beim Login automatisch aktivieren

Viele Tools (z. B. WebStorm) öffnen neue Verbindungen, ohne `SET ROLE …` auszuführen. Dann sind Rollen zwar zugewiesen, aber nicht aktiv – die Datenbank wird ggf. nicht angezeigt.

Serverweit alle Rollen automatisch aktivieren

```
-- als root/Administrator:
SET PERSIST activate_all_roles_on_login = ON;
```

Ab jetzt sind für alle Benutzer die zugewiesenen Rollen bei jeder Anmeldung aktiv – auch in neuen Query-Consoles.

Schritt 6: Rechte kontrollieren (SHOW GRANTS)

Mit `SHOW GRANTS` können Sie überprüfen, welche Rechte ein Benutzer effektiv besitzt.

```
SHOW GRANTS FOR 'lehrperson_koch'@'localhost';
SHOW GRANTS FOR 'lernende_caduff'@'localhost';
```

Sie sehen dort, welche Rollen zugewiesen wurden und welche Privilegien diese Rollen enthalten. So können Sie auch im Unterricht gemeinsam kontrollieren, ob die gewünschte Rechtevergabe funktioniert.

Schritt 7: Rechte und Benutzer entfernen (REVOKE & DROP USER)

Wenn sich Aufgaben ändern oder Benutzer das System verlassen, können Sie Rechte oder Rollen wieder entziehen und Benutzer löschen.

Rechte oder Rollen entziehen

```
-- Einzelne Rechte entziehen
REVOKE SELECT ON noten_db.* FROM lernende_role;

-- Eine Rolle von einem Benutzer entfernen
REVOKE lernende_role FROM 'lernende_caduff'@'localhost';
```

Benutzer löschen

Wenn ein Benutzer nicht mehr benötigt wird, können Sie ihn vollständig entfernen:

```
DROP USER 'lernende_caduff'@'localhost';
DROP USER 'lehrperson_koch'@'localhost';
```

Damit entfernen Sie sowohl den Zugriff als auch alle gespeicherten Login-Daten. Dies ist ein wichtiger Schritt bei **Personalwechseln** oder **Projektende**, um den Datenschutz weiterhin sicherzustellen.

Verbindung zu Datenschutz (DSG/DSGVO)

Mit diesen zwei Rollen und zwei MySQL-Benutzern setzen Sie zentrale Datenschutzprinzipien um:

- **Schutz vor unbefugtem Zugriff**
  - Nur die Benutzer `lehrperson_koch` und `lernende_caduff` haben Zugriff auf die Noten-Datenbank.
- **Prinzip der minimalen Privilegien**
  - `lehrperson_koch` darf alles, trägt aber auch volle Verantwortung.
  - `lernende_caduff` darf nur lesen (SELECT), keine Noten verändern.
- **Vertraulichkeit und Integrität**
  - Strukturänderungen und Rechteverwaltung sind auf die Admin-Rolle beschränkt.
  - Lernende können keine Noten manipulieren, sondern nur ansehen.

So entsteht aus wenigen, klar definierten Rollen und Benutzern ein übersichtliches und datenschutzkonformes Berechtigungskonzept für eine Noten-Datenbank.