### Ausgangssituation:

Es gibt eine Datenbank **Schule** mit folgenden Tabellen :

* Schueler (id, name, klasse)
* Lehrer (id, name, fach)
* Noten (schueler\_id, fach, note)

Es gibt folgende Benutzer:

* admin
* lehrer1
* lehrer2
* sekretaer
* gast

### Aufgaben:

#### Rechte vergeben (GRANT)

**a)** Erlaube dem Benutzer lehrer1, **nur** die Tabelle Noten zu lesen.

GRANT SELECT ON Schule.Noten TO ‘lehrer1’@’localhost’

**b)** Erlaube sekretaer, **Datensätze** in die Tabelle Schueler **einzufügen** und **anzusehen**.

GRANT SELECT, INSERT ON Schule.Schueler TO ‘sekretaer’@’localhost’

**c)** Erlaube admin, **alle Rechte** auf **alle Tabellen**.

GRANT ALL ON Schule.\* TO ‘admin’@’localhost’

**d)** Erlaube lehrer2, in der Tabelle Noten **nur bestimmte Spalten** (note) zu **aktualisieren**.  
(Hinweis: In Standard-SQL ist Spalten-spezifisches GRANT nicht direkt möglich; aber in einigen DBMS kann man das kombinieren mit Sichten.)

CREATE VIEW NotenNurNoteSpalte AS SELECT schueler\_id, fach, note FROM Schule.Noten;

**e)** Erlaube gast, die Tabelle Schueler **nur zu lesen**, aber verbiete später das Lesen wieder.

#### Rechte entziehen (REVOKE)

**a)** Entziehe lehrer1 das Leserecht auf Noten.

REVOKE SELECT ON Schule.Noten FROM ‘lehrer1’@’localhost’

**b)** Entziehe sekretaer das Recht, Datensätze in Schueler einzufügen.

REVOKE INSERT ON Schule.Schueler FROM ‘sekretaer’@’localhost’

**c)** Entziehe admin **alle Rechte** auf **alle Tabellen** (nur zu Übungszwecken).

### Zusatzaufgabe:

**Erstelle eine View**, die nur die Namen und Klassen der Schüler zeigt, und gib gast nur auf diese View Leserechte.

### Bonus: Theoriefragen

* Was passiert, wenn ein Benutzer ein Recht auf ein Objekt erhält, aber später das Recht durch REVOKE entzogen wird?
* Können GRANT und REVOKE auch auf **Views** angewendet werden?
* Was ist der Unterschied zwischen WITH GRANT OPTION und einem normalen GRANT?