Kap 6: Sicherheit in DBMS:

Autorisierung:

- **Daten vor unberechtigtem Zugriff schützen**
- **Daten vor Löschen und Verfälschungen schützen**
- **Rahmenbedingungen**

Vergabe von Objektberechtigungen

```
GRANT     <Right> ON <Object> TO   <User>;
REVOKE    <Right> ON <Object> FROM <User>;
```

Rights: `SELECT, UPDATE, INSERT`

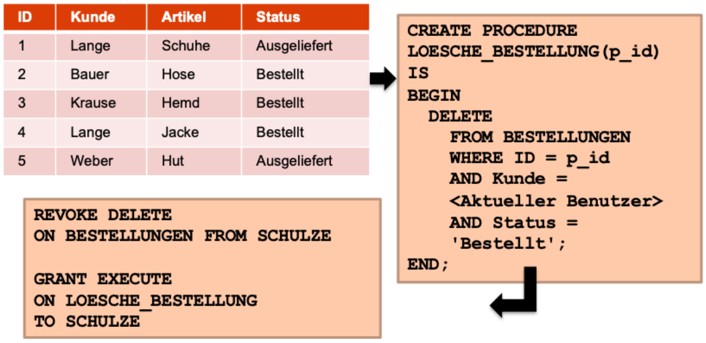
- `GRANT SELECT, UPDATE, INSERT, DELETE ON MITARBEITER TO MAIER;`
- `REVOKE UPDATE ON MITARBEITER FROM SCHULZE;`

Rechte durch Views einschränken

```
CREATE VIEW V_NAME AS
SELECT ... FROM ... WHERE ...

GRANT SELECT ON V_VIEW TO MUELLER
```

Rechte einschränken: Stored Procedures



ID	Kunde	Artikel	Status
1	Lange	Schuhe	Ausgeliefert
2	Bauer	Hose	Bestellt
3	Krause	Hemd	Bestellt
4	Lange	Jacke	Bestellt
5	Weber	Hut	Ausgeliefert

```
CREATE PROCEDURE
LOESCHE_BESTELLUNG(p_id)
IS
BEGIN
  DELETE
    FROM BESTELLUNGEN
    WHERE ID = p_id
    AND Kunde =
    <Aktueller Benutzer>
    AND Status =
    'Bestellt';
END;
```

```
REVOKE DELETE
ON BESTELLUNGEN FROM SCHULZE

GRANT EXECUTE
ON LOESCHE_BESTELLUNG
TO SCHULZE
```

Mit welchen Rechten wird eine Stored Procedure ausgeführt?

Default-Einstellung: Rechte der Benutzerin, die die SP angelegt hat

Mögliche Definition: "Invoker Rights"

- Dann läuft die SP mit den Rechten des aufrufenden Benutzers

Rollen

Schritt 1: Rolle erzeugen

- `CREATE ROLE ANGESTELLTER`

Schritt 2: Recht an Rolle vergeben

- `GRANT SELECT ON V_MITARBEITER TO ANGESTELLTER`

Schritt 3: Rolle an Mitarbeiter vergeben

- `GRANT ANGESTELLTER TO MUELLER`

Weitergabe von Rechten

- `GRANT SELECTON KUNDEN TO MEIER WITH GRANT OPTION;`

Authentifizierung: Wer bin ich?

Thema Authentifikation: Wie identifiziere ich mich?

Klassisch:	**Username / Passwort**	schwache Authentifikation
Alternativen: z.B.	**digitale Zertifikate**	starke Authentifikation

Sicherheitslücken vermeiden: Wie stelle ich das sicher?

Sicherheitsprobleme (Beispiele)

Beispiel: Bekannte Default-Passwörter

- Bei Oracle z.B. sys/change_on_install

SQL Injection

- Wenn der User die Daten/Eingabe so manipuliert, dass diese einen Datenbankcode ergeben, so kann sich der User Zugriff auf die Datenbank verschaffen.

Schutzt vor SQL Injektion

- Prepared Statements