



# **FRAGEN ZUR KLAUSUR**



# WIEDERHOLUNG

# CAP-THEOREM

AP

CA

CP



# NOSQL

# KEY-VALUE-STORE

```
redis> ping
PONG
redis> set foo bar
OK
redis> get foo
"bar"
redis> incr mycounter
(integer) 1
redis> incr mycounter
(integer) 2
redis> get mycounter
"2"
```

# **COLUMN STORE**

## **(SPALTENORIENTIERTE DB)**

# GRAPH DATABASES

# **DOKUMENTEN-ORIENTIERT**

# BEISPIEL: COUCHDB-DOKUMENT

```
{
  "_id" : "00a271787f89c0ef2e10e88a0c0001f4"
  "_rev": "5509377776",
  "name": "Peter Lustig",
  "address": {
    "street": "Teststr.",
    "city": "Hamburg"
  }
}
```





**LOS GEHT'S**

**\o/**

# DATENSCHUTZ

# RECHTEVERGABE IN SQL

# BENUTZERSTRUKTUR

# **BENUTZER ANLEGEN**

```
create user tina identified by 'my-secret-pw';
```

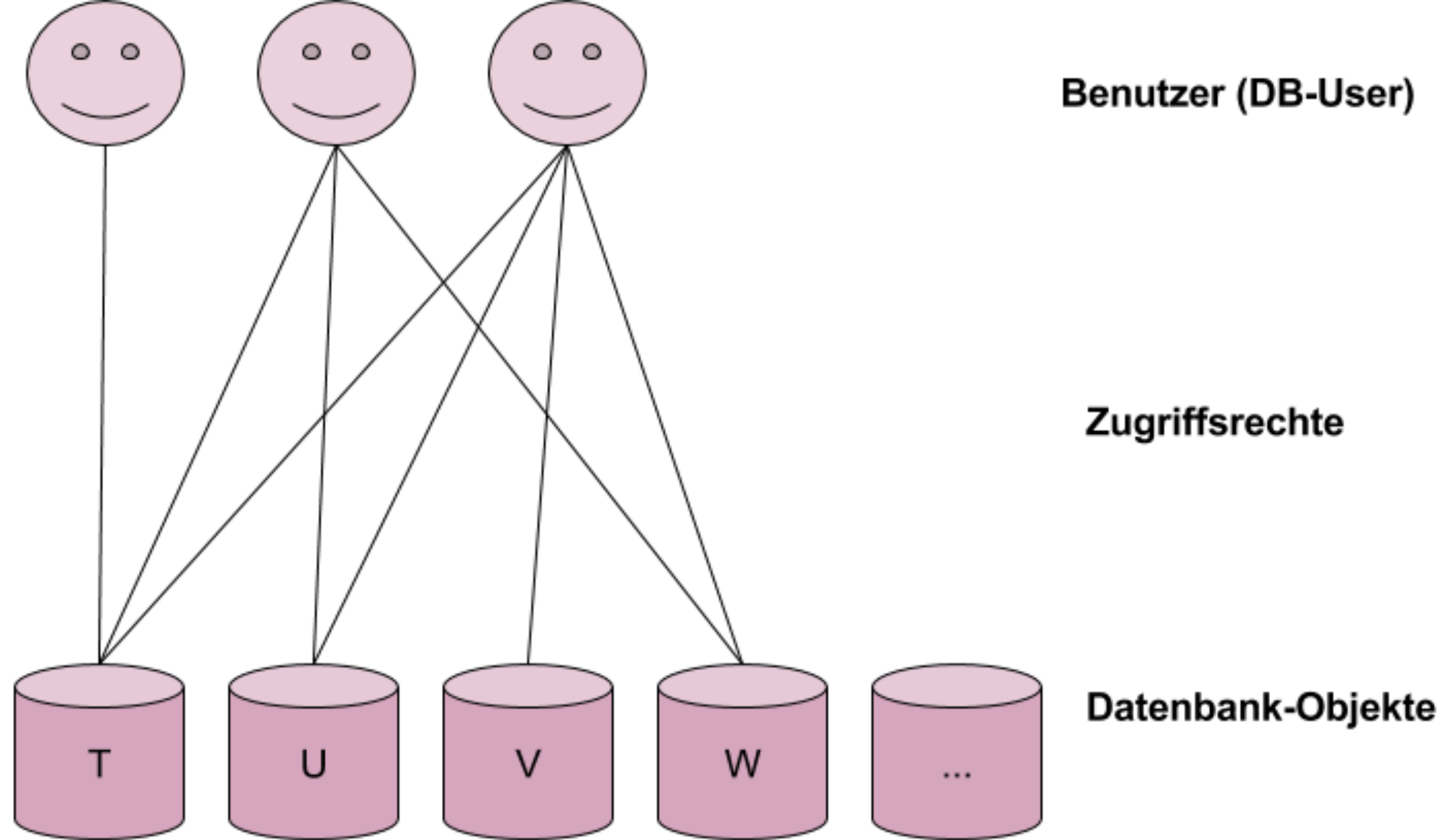
```
CREATE PROFILE app_user LIMIT
  SESSIONS_PER_USER          UNLIMITED
  CPU_PER_SESSION            UNLIMITED
  CPU_PER_CALL                3000
  CONNECT_TIME                45
  LOGICAL_READS_PER_SESSION  DEFAULT
  LOGICAL_READS_PER_CALL     1000
  PRIVATE_SGA                 15K
  COMPOSITE_LIMIT             5000000;
```

```
CREATE PROFILE app_user_pw LIMIT  
  FAILED_LOGIN_ATTEMPTS 5  
  PASSWORD_LIFE_TIME 60  
  PASSWORD_REUSE_TIME 60  
  PASSWORD_REUSE_MAX 5  
  PASSWORD_VERIFY_FUNCTION verify_function  
  PASSWORD_LOCK_TIME 1/24  
  PASSWORD_GRACE_TIME 10;
```



```
CREATE USER sidney  
  IDENTIFIED BY out_standing1  
  DEFAULT TABLESPACE example  
  QUOTA 10M ON example  
  TEMPORARY TABLESPACE temp  
  QUOTA 5M ON system  
  PROFILE app_user  
  PASSWORD EXPIRE;
```

# ZUGRIFFSRECHTE



	T	U	V	W
User A	A	-	-	-
User B	S	S, I	-	A
User C	S	S	S	S, I
...	...	...	...	...

**(AutorisierungsID, DB-Ausschnitt, Operation)**

Datenbanknutzer

Relationen, Sichten etc.

select, insert, delete, update, ...

# RECHTEVERGABE

```
grant [select | insert | update | delete | all]  
on <table | view>  
to <username | public> [with grant option]
```

# SYSTEMRECHTE

CREATE, ALTER UND DROP Z.B. VON  
TABLE, INDEX, USER, ROLE, ETC.



# OBJEKTRECHTE

SELECT, INSERT, UPDATE,  
DELETE, ETC. UND BEZIEHEN SICH  
ZUSÄTZLICH NOCH AUF EIN  
DATENBANKOBJEKT (TABELLE, VIEWS,  
TRIGGER ETC.).

```
GRANT INSERT ON PERSONAL TO leonhard_euler;
```

```
GRANT DROP TABLE TO ada_lovelace;
```

```
GRANT ALL ON MASCHINE TO hedy_lamarr;
```

GRANT . . . WITH GRANT OPTION;

**GRANT . . . WITH ADMIN OPTION;**

```
GRANT SELECT ON PERSONAL TO hedy_lamarr WITH ADMIN OPTION;
```

# ZUR ÜCKNAHME VON RECHTEN

```
REVOKE [SELECT | INSERT | UPDATE | DELETE | ALL]  
ON <table | view>  
FROM <username | public>;
```



```
REVOKE INSERT ON PERSONAL TO leonhard_euler;
```

```
REVOKE DROP TABLE TO ada_lovelace;
```

```
REVOKE ALL ON MASCHINE TO hedy_lamarr;
```



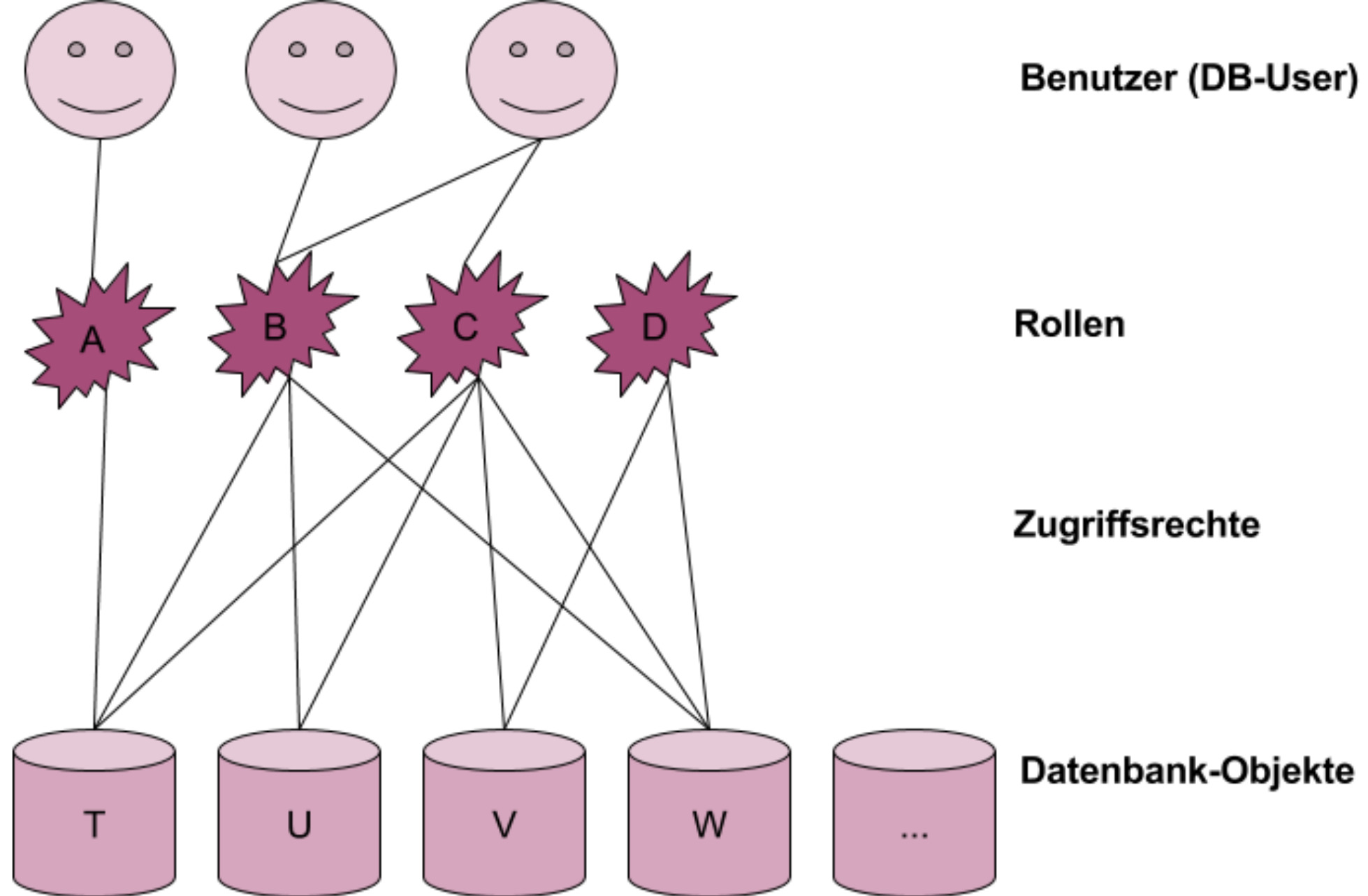
BEISPIEL

# **BEARBEITUNGSWORKFLOW FÜR DOKUMENT IN EINEM UNTERNEHMEN**

## PROBLEM

**MITARBEITER IST KRANK BZW. VERLÄSST DAS UNTERNEHMEN UND  
NUR DIESER HAT DIE RECHTE, EIN DOKUMENT ZU ÄNDERN**

# ROLLEN



	T	U	V	W
Rolle A	A	-	-	-
Rolle B	S	S, I	-	A
Rolle C	S	S	S	S, I
...	...	...	...	...

	Rolle A	Rolle B	Rolle C	Rolle D
User A	X	-	-	-
User B	-	X	-	-
User C	-	X	X	-
...	...	...	...	...

# ROLLEN ERSTELLEN

```
CREATE ROLE reports;
```

```
GRANT reports TO leonhard_euler;
```

```
GRANT reports TO hedy_lamarr;
```

# ROLLEN DEFINIEREN

```
GRANT SELECT ON PERSONAL TO reports;
```

```
GRANT SELECT ON MASCHINE TO reports;
```



# ROLLEN ROLLEN ZUWEISEN

```
CREATE ROLE finance_reports;
```

```
GRANT SELECT ON PRAEMIE TO finance_reports;
```

```
GRANT finance_reports TO reports;
```

# EIGENSCHAFTEN

**BERECHTIGUNGEN WERDEN IN ROLLEN  
GEEIGNET ZUSAMMENGEFASST**

# VEREINFACHUNG UND VERBESSERUNG DER RECHTEVERWALTUNG

VORTEILE

# NACHVOLLZIEHBARKEIT

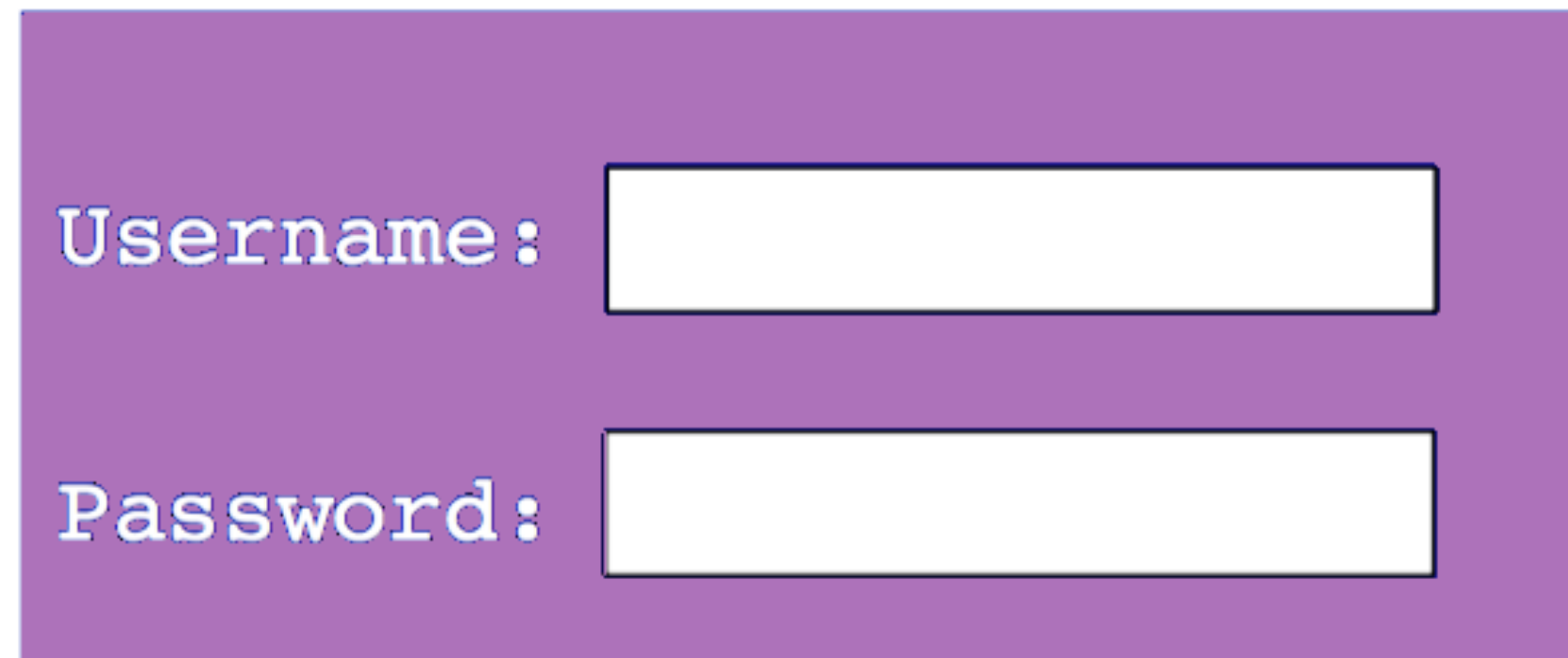
# BESSERE WARTBARKEIT

**„SINGLE POINT OF ADMINISTRATION“**



# SQL-INJECTION UND IHRE VERMEIDUNG

```
conn = pool.getConnection( );
String sql = "select * from user2 where username='" +
    username + "'" and password='" + password + "'";
stmt = conn.createStatement();
rs = stmt.executeQuery(sql);
if (rs.next()) {
    loggedIn = true;
    out.println("Successfully logged in");
} else {
    out.println("Username and/or password not recognized");
}
```



Username:

Password:

```
username = "admin' OR '1'='1";
```

```
String sql = "select * from user where username='" +  
    username + "' and password='" + password + "'";
```

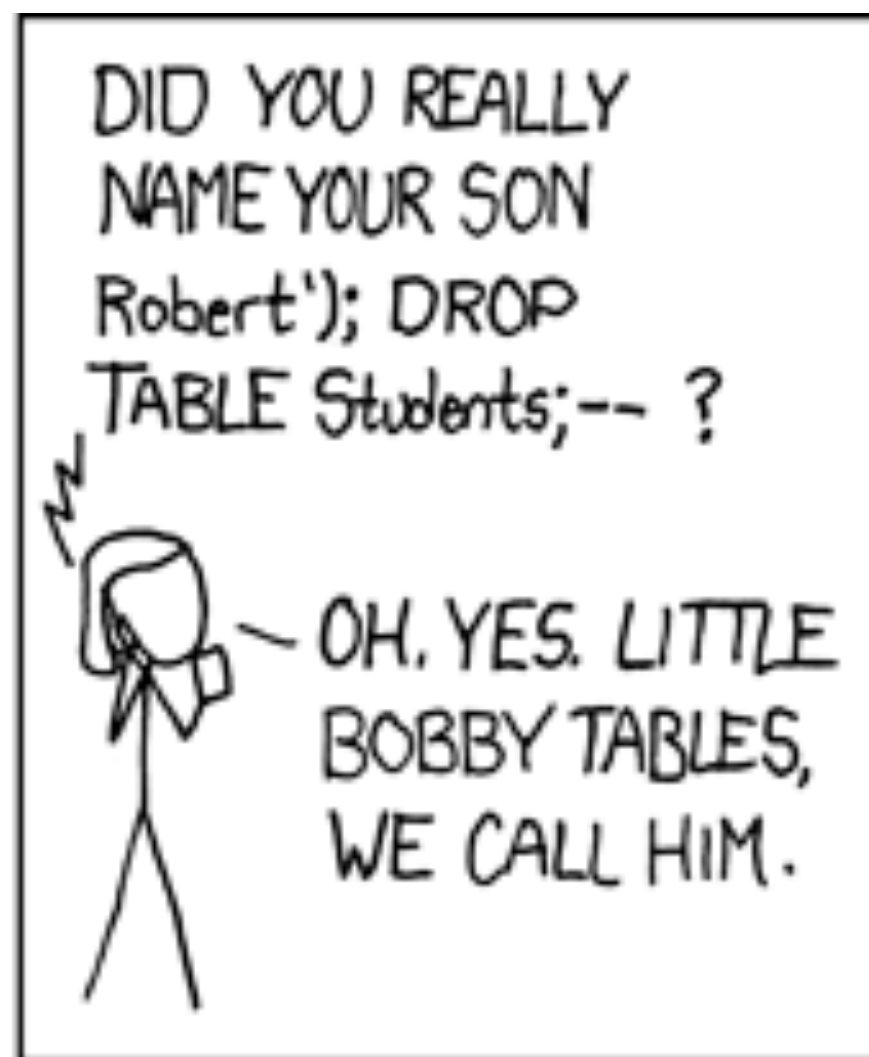
```
==>
```

```
sql = "select * from user where username='admin'  
    OR '1'='1' and password=' '";
```

Username:

admin' OR '1'='1

Password:



```
name = "Robert'); DROP TABLE STUDENTS;--";
```

```
String sql = "select * from student where (name='" + name + "')";
```

```
==>
```

```
sql = "select * from student where (name='" + name + "')";
```

```
sql = "select * from student where (name='Robert'); DROP TABLE STUDENTS; --')"
```



# GEGENMASSNAHMEN

# 'DIREKTES' JDBC-STATEMENT

```
final Statement stmt = con.createStatement();  
final ResultSet rset = stmt.executeQuery("SELECT pnr FROM PERSONAL WHERE nachname = '" + user_nachname + "';");
```

```
final Statement stmt = con.createStatement();
final ResultSet rset = stmt.executeQuery("SELECT pnr FROM PERSONAL WHERE nachname = '" + user_nachname + "';");

final PreparedStatement pstmt = con.prepareStatement("SELECT pnr FROM PERSONAL WHERE nachname = ?");
pstmt.setString(1, user_nachname);
final ResultSet rset = pstmt.executeQuery();
```



