

# Datenbanken und Datensicherheit

Prof. Dr. rer. nat. Peter Gerwinski

16. November 2023

# Datenbanken und Datensicherheit

<https://gitlab.cvh-server.de/pgerwinski/dbs>

## 1 Einführung

## 2 Kurzeinführung Unix

## 3 Kurzeinführung TCP/IP

...

### 3.5 Routing

### 3.6 Netzwerkanalyse

### 3.7 SSH

### 3.8 X11

## 4 Relationale Datenbanken

### 4.1 Einführung in DBMS

### 4.2 Einführung in SQL

### 4.3 Normalformen

### 4.4 Verknüpfungen von Tabellen

...

...



Änderungen  
vorbehalten

## 3.8 X11

- Grafik-Bildschirm und Eingabegeräte über's Netz
- `DISPLAY`-Variable: X-Server: Rechner und Bildschirm
- `ssh -X`: X11-Forwarding

# 4 Relationale Datenbanken

## 4.1 Einführung in DBMS

Datenbank-Management-System (DBMS): z. B. PostgreSQL

- Datenbanken „von außen“ verwalten (z. B. auflisten)
- Schnittstelle für Zugriff auf Datenbank
- **in hohem Maße herstellerspezifisch**

```
root@cassini:~# su - postgres
postgres@cassini:~$ psql
psql (15.3 (Debian 15.3-0+deb12u1))
Geben Sie »help« für Hilfe ein.
```

```
postgres=# help
Dies ist psql, die Kommandozeilenschnittstelle für PostgreSQL.
Geben Sie ein:  \copyright für Urheberrechtssinformationen
                 \h für Hilfe über SQL-Anweisungen
                 \? für Hilfe über interne Anweisungen
                 \g oder Semikolon, um eine Anfrage auszuführen
                 \q um zu beenden

postgres=#
```

# 4 Relationale Datenbanken

## 4.1 Einführung in DBMS

Datenbank-Management-System (DBMS): z. B. PostgreSQL

- Datenbanken „von außen“ verwalten (z. B. auflisten)
- Schnittstelle für Zugriff auf Datenbank
- **in hohem Maße herstellerspezifisch**

```
postgres=# \?
```

```
Allgemein
```

|                         |   |
|-------------------------|---|
| \copyright              | PostgreSQL-Urheberrechtsinformationen zeigen                              |
| \crosstabview [SPALTEN] | Anfrage ausführen und Ergebnis als Kreuztabelle anzeigen                  |
| \errverbose             | letzte Fehlermeldung mit vollen Details anzeigen                          |
| \g [(OPT)] [DATEI]      | SQL-Anweisung ausführen (und Ergebnis in Datei oder  Pipe schreiben); ... |
| ...                     | ...   |
| \q                      | psql beenden  |
| ...                     | ...   |

```
postgres=#
```

# 4 Relationale Datenbanken

## 4.1 Einführung in DBMS

Datenbank-Management-System (DBMS): z. B. PostgreSQL

- Datenbanken „von außen“ verwalten (z. B. auflisten)
- Schnittstelle für Zugriff auf Datenbank
- **in hohem Maße herstellerspezifisch**

```
postgres=# \l
```

| Liste der Datenbanken |            |           |              |             |  |
|-----------------------|------------|-----------|--------------|-------------|--|
| Name                  | Eigentümer | Kodierung | Sortierfolge | Zeichentyp  |  |
| postgres              | postgres   | UTF8      | de_DE.UTF-8  | de_DE.UTF-8 |  |
| template0             | postgres   | UTF8      | de_DE.UTF-8  | de_DE.UTF-8 |  |
| template1             | postgres   | UTF8      | de_DE.UTF-8  | de_DE.UTF-8 |  |
| (3 Zeilen)            |            |           |              |             |  |

```
postgres=#
```

# 4 Relationale Datenbanken

## 4.2 Einführung in SQL

Datenbank-Abfragesprache: Structured Query Language (SQL)

- Datenbanken „von innen“ verwalten, z. B. anlegen und wieder löschen
- Daten in der Datenbank bearbeiten
- **in hohem Maße standardisiert**

```
postgres=# create database testdb;  
CREATE DATABASE  
postgres=# create user dbs with password '####';  
CREATE ROLE  
postgres=# GRANT ALL ON DATABASE testdb TO dbs;  
GRANT  
postgres=# \q  
postgres@cassini:~$
```

# 4 Relationale Datenbanken

## 4.2 Einführung in SQL

Datenbank-Abfragesprache: Structured Query Language (SQL)

- Datenbanken „von innen“ verwalten, z. B. anlegen und wieder löschen
- Daten in der Datenbank bearbeiten
- **in hohem Maße standardisiert**

```
$ psql -h localhost -U dbs -W testdb
```

```
Passwort: ####
```

```
psql (15.5 (Debian 15.5-0+deb12u1))
```

```
SSL-Verbindung (Protokoll: TLSv1.3, Verschlüsselungsmethode: TLS_AES_256_G
```

```
Geben Sie »help« für Hilfe ein.
```

```
testdb=> \d
```

```
      Liste der Relationen
```

| Schema | Name | Typ     | Eigentümer |
|--------|------|---------|------------|
| public | tier | Tabelle | postgres   |

(1 Zeile)

```
testdb=>
```



# 4 Relationale Datenbanken

## 4.2 Einführung in SQL

Datenbank-Abfragesprache: Structured Query Language (SQL)

- Datenbanken „von innen“ verwalten, z. B. anlegen und wieder löschen
- Daten in der Datenbank bearbeiten
- **in hohem Maße standardisiert**

```
testdb=> \d tier
```

| Tabelle »public.tier« |               |              |               |             |
|-----------------------|---------------|--------------|---------------|-------------|
| Spalte                | Typ           | Sortierfolge | NULL erlaubt? | Vorgabewert |
| name                  | character(30) |              |               |             |
| tierart               | character(30) |              |               |             |
| id                    | integer       |              |               |             |

```
testdb=>
```

# 4 Relationale Datenbanken

## 4.2 Einführung in SQL

Datenbank-Abfragesprache: Structured Query Language (SQL)

Literatur: z. B. [https://de.wikibooks.org/wiki/Einführung\\_in\\_SQL](https://de.wikibooks.org/wiki/Einführung_in_SQL)

Wichtige SQL-Befehle:

- **CREATE** – Datenbanken, Tabellen usw. anlegen
- **DROP** – Datenbanken, Tabellen usw. löschen
- **SELECT** – Daten abfragen
- **INSERT INTO ... VALUES** – Daten eingeben
- **UPDATE** – Daten ändern
- **DELETE FROM** – Daten löschen

# 4 Relationale Datenbanken

## 4.3 Normalformen

Problem: Schlecht angelegte Datenbanken werden schnell inkonsistent.  
Beliebte Fehler:

- Speichern von mehreren Daten in demselben Tabelleneintrag
- Speichern von denselben Daten in verschiedenen Tabelleneinträgen
- implizite Zusammenhänge
- voneinander unabhängige Zusammenhänge in derselben Tabelle

Literatur: z. B. [https://de.wikipedia.org/wiki/Normalisierung\\_\(Datenbank\)](https://de.wikipedia.org/wiki/Normalisierung_(Datenbank))

# 4 Relationale Datenbanken

## 4.3 Normalformen

Problem: Schlecht angelegte Datenbanken werden schnell inkonsistent.  
Beliebte Fehler:

- Speichern von mehreren Daten in demselben Tabelleneintrag  
→ 1. Normalform
- Speichern von denselben Daten in verschiedenen Tabelleneinträgen  
→ 2. Normalform
- implizite Zusammenhänge  
→ 3. Normalform und Boyce-Codd-Normalform
- voneinander unabhängige Zusammenhänge in derselben Tabelle  
→ 4. und 5. Normalform

Lösung: Normalformen

Literatur: z. B. [https://de.wikipedia.org/wiki/Normalisierung\\_\(Datenbank\)](https://de.wikipedia.org/wiki/Normalisierung_(Datenbank))

## 4 Relationale Datenbanken

### 4.4 Verknüpfungen von Tabellen

Problem: Gut angelegte Datenbanken (→ Normalformen) sind stark aufgesplittet.

Wie kann man sie weiterhin effizient benutzen?

Lösung: Verknüpfungen von Tabellen

SQL-Befehl: JOIN

Literatur: z. B. <https://de.wikipedia.org/wiki/SQL>