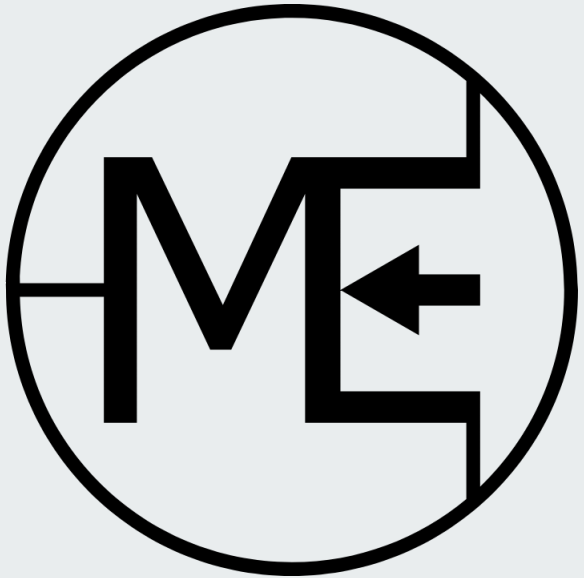
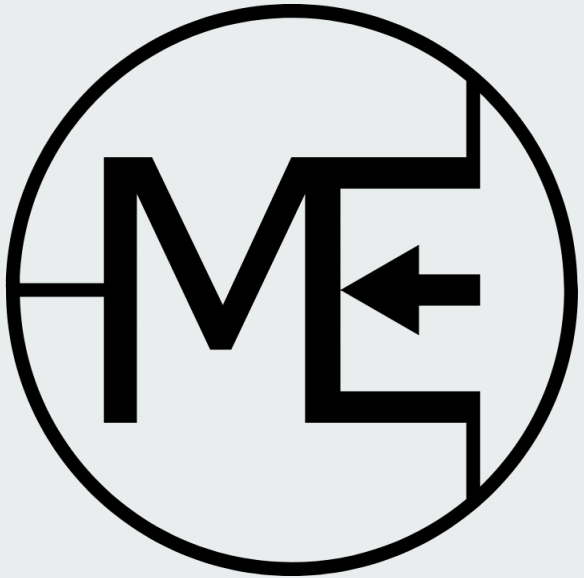
Relationale Datenbanken & SQL



WLAN: TP-Link DA501C
Passwort: 11230113



Agenda

- Was ist eine Datenbank?
- Grundlagen der Datenmodellierung
- Verbindung mit der Datenbank aufbauen
- SQL-Basics
  - Tabellen anlegen
  - Daten eintragen
  - Daten selektieren
  - Daten bearbeiten

Was ist eine relationale Datenbank?

- strukturierte Speicherung großer Datenmengen
- performante Abfragesprache SQL (Structured Query Language)
- Tabellen können untereinander Beziehungen besitzen (=Relationen)
- Transaktionsprinzip (Commit / Rollback)
- ACID-Prinzip
  - Atomarität / Abgeschlossenheit → Transaktion wird “ganz oder gar nicht” ausgeführt
  - Consistency / Konsistenz →Nach Transaktionsende ist Datenzustand konsistent
  - Isolation / Abgrenzung →parallele Transaktionen beeinflussen sich nicht gegenseitig
  - Durability / Dauerhaftigkeit →dauerhaftes Speichern nach Transaktionsende in DB

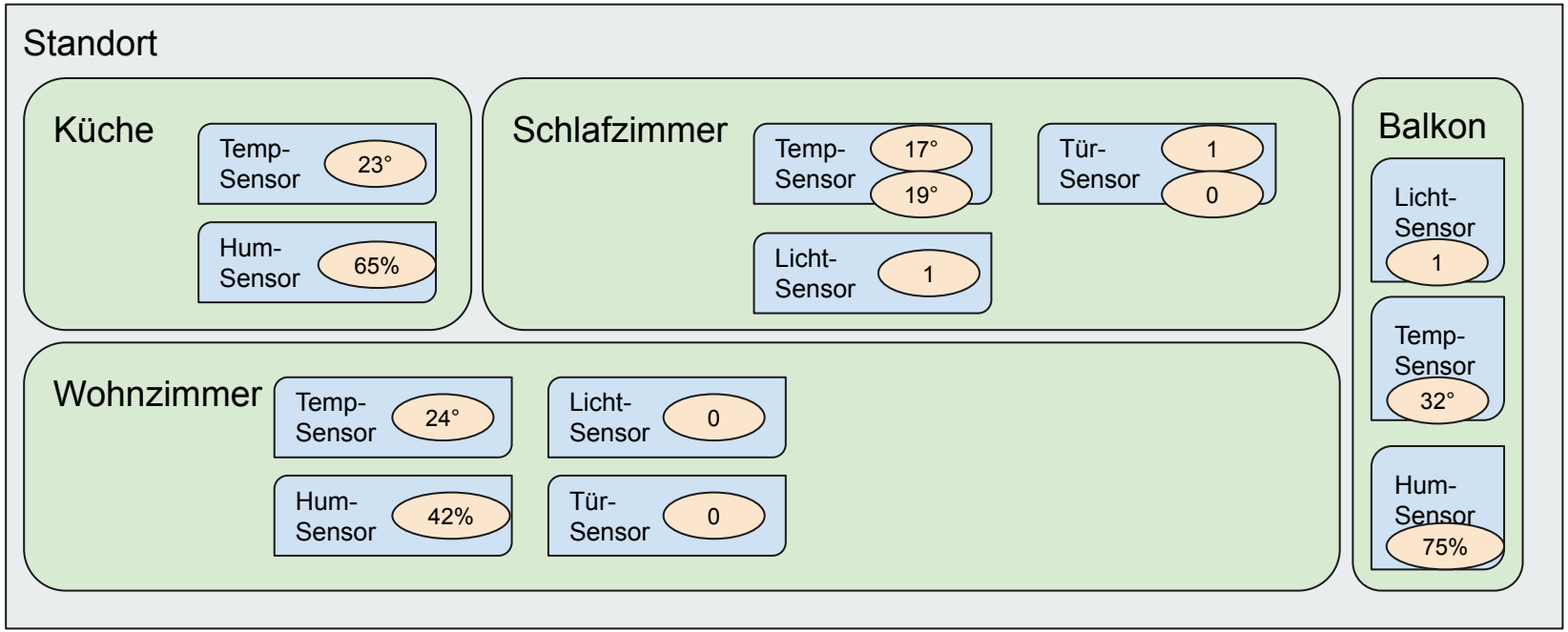
Grundlagen der Datenmodellierung

- Normalisierung der Datenmodelle
  - 1. NF: atomar
    - z.B. Aufteilen eines Messdaten-Sets in die Felder Sensor-ID, Temperatur, Luftfeuchtigkeit, Luftdruck
  - 2. NF: 1. NF + kein nicht-primäres Attribut, das funktional von Teilmenge abhängig ist
    - z.B. mehrere Messwerte eines Sensors in mehreren DB-Zeilen abspeichern
  - 3. NF: 2. NF + keine transitiven Abhängigkeiten
    - z.B. Feld “Sensortyp” in separate Tabelle auslagern
  - Boyce-Codd-Normalform: 3. NF + jede Determinante ist Schlüsselkandidat

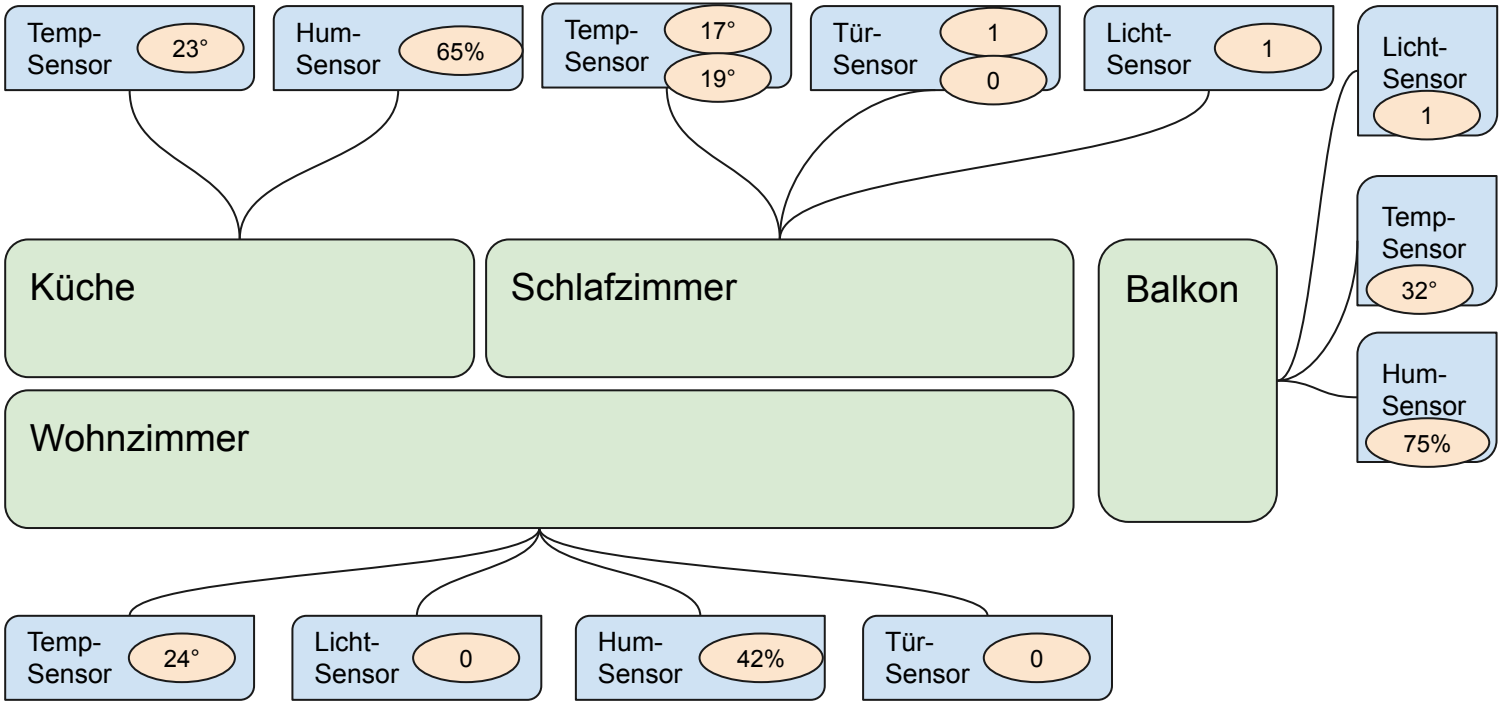
Grundlagen der Datenmodellierung

- Schlüssel-Attribute
  - Primärschlüssel
    - eindeutige Identifizierung eines Datensatzes
  - Fremdschlüssel
    - Verknüpfung von Datensätzen zwischen Tabellen
- Constraints
  - Unique Key-Constraints
    - Sicherstellung, dass Wert in Spalte eindeutig ist
  - Werte-Constraints
    - Prüfung auf Inhalte, z.B. Ja/Nein

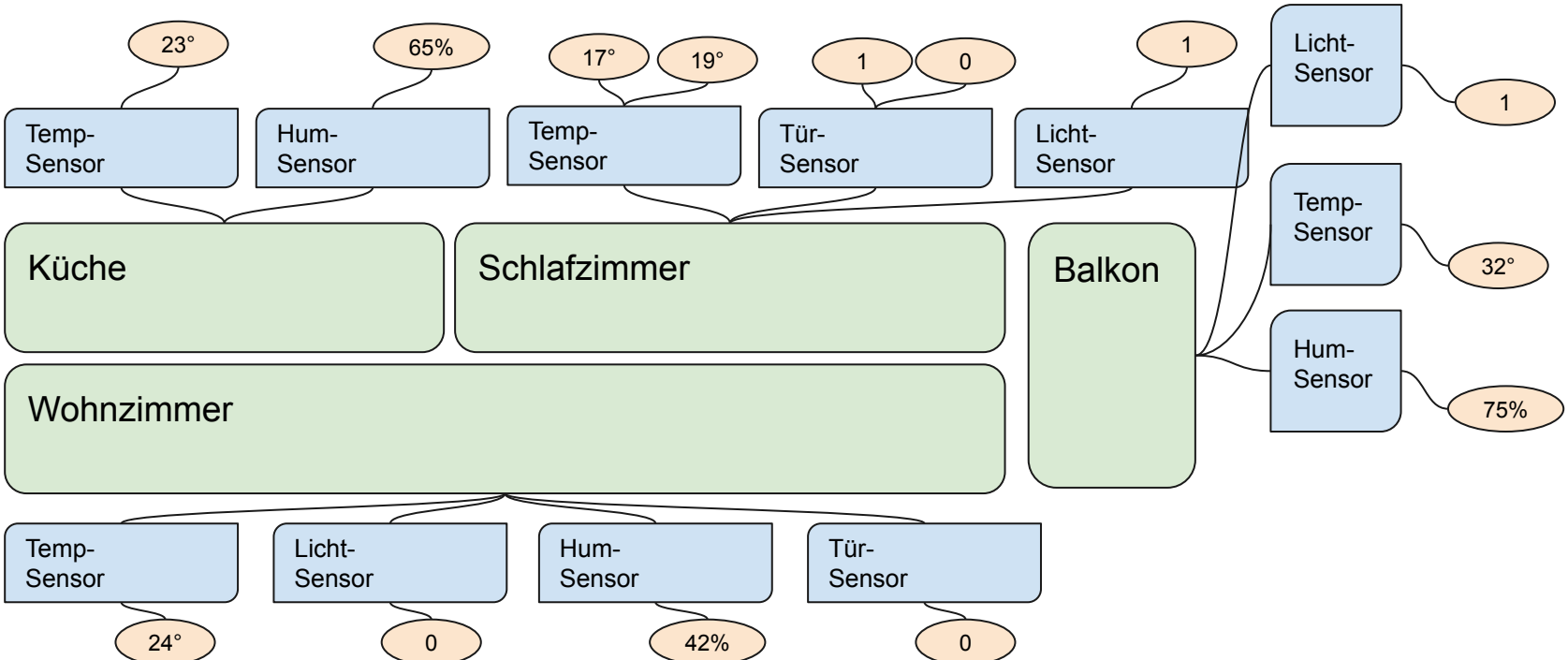
Viel bla bla - mal pragmatisch anschauen..



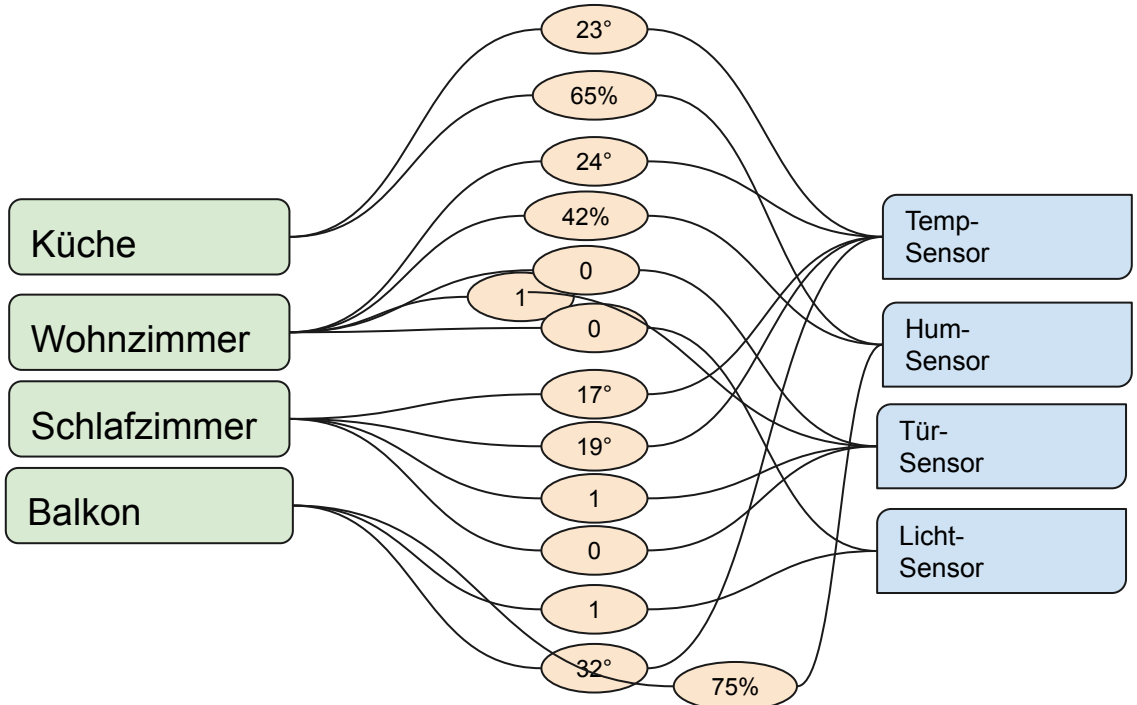
Viel bla bla - mal pragmatisch anschauen..

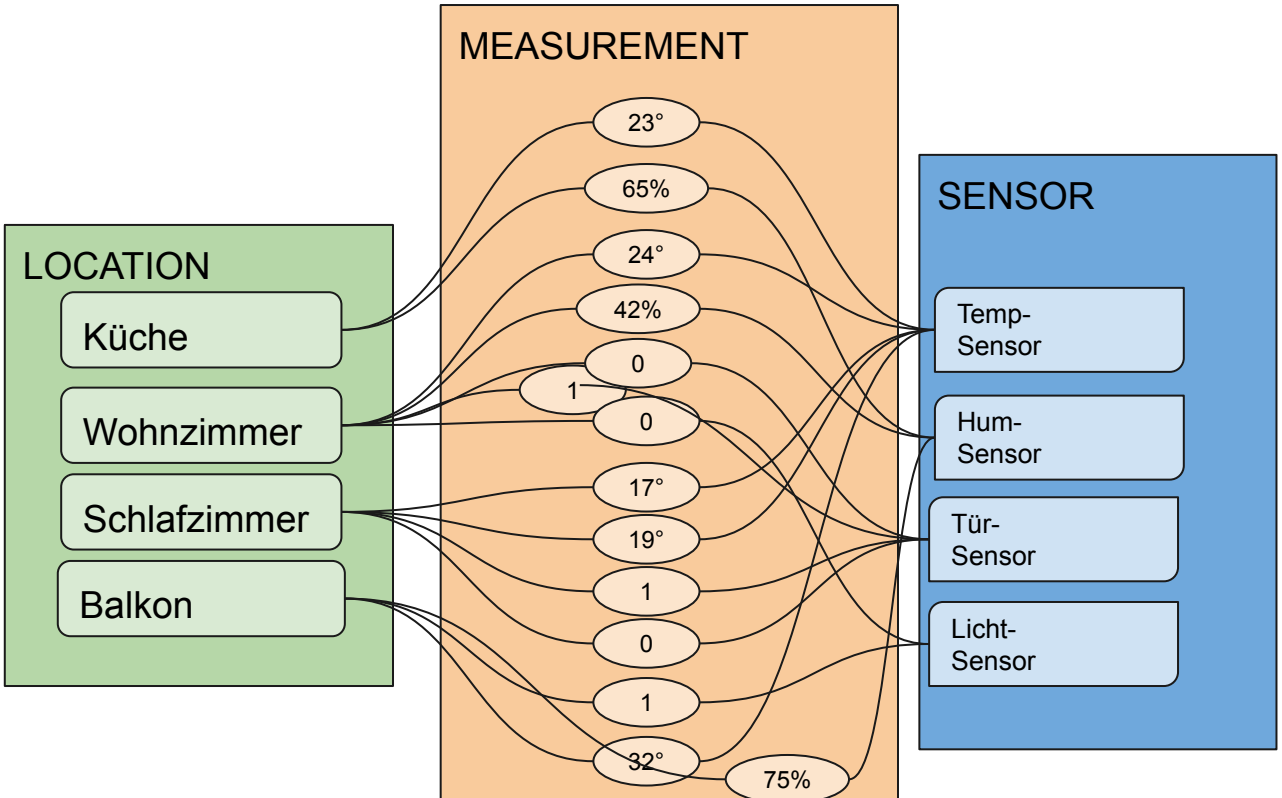


Viel bla bla - mal pragmatisch anschauen..



Viel bla bla - mal pragmatisch anschauen..




MEASUREMENT
LOCATION
Küche
Wohnzimmer
Schlafzimmer
Balkon
23°
65%
24°
42%
0
1
0
17°
19°
1
0
1
32°
75%
SENSOR
Temp-
Sensor
Hum-
Sensor
Tür-
Sensor
Licht-
Sensor

Verbindung mit der Datenbank aufbauen

Entwicklungstools

- Adminer (im Browser 127.0.0.1:8080 aufrufen)
  - einfache webbasierte GUI zur Administrierung
  - einfach SQL-Operationen möglich
- VS Code
  - deutlich komfortableres Entwicklungstool
  - kurzer Einrichtungsaufwand nötig

Verbindungsdaten

- IP: 127.0.0.1
- Port: 5432
- Server: db
- Benutzer: postgres
- Password: example
- Datenbank: postgres

Datenbank erstellen

```
CREATE DATABASE makesworkshop;
```

…danach in VSCode Verbindung hinzufügen

SQL Basics: Tabellen anlegen

```
CREATE TABLE location (
      loca_id SERIAL PRIMARY KEY,
      loca_name VARCHAR (100)
);
```

```
CREATE TABLE sensor (
      sens_id SERIAL PRIMARY KEY,
      sens_type VARCHAR (100)
);
```

```
CREATE TABLE measurement (
      meas_id SERIAL PRIMARY KEY,
      meas_loca_id INT REFERENCES location(loca_id),
      meas_sens_id INT REFERENCES sensor(sens_id),
      meas_value NUMERIC(5,2),
      meas_time TIMESTAMP
);
```

SQL Basics: Datentypen

- SERIAL →numerischer Wert, der automatisch hochgezählt wird
- VARCHAR (x) →Text, Speicherlänge dynamisch
- INT →Integer / Ganzzahlen
- NUMERIC (x,y) →Zahl der Gesamtlänge x mit y Nachkommastellen
- TIMESTAMP →Datum und Zeit

SQL Basics: Daten eintragen

```
INSERT INTO location(loca_name)
    VALUES ('Wohnzimmer');
```

```
INSERT INTO sensor(sens_type)
    VALUES ('Temp');
```

```
INSERT INTO measurement(meas_loca_id,
                        meas_sens_id,
                        meas_value,
                        meas_timestamp)
    VALUES (1,
            1,
            23,
            CURRENT_TIMESTAMP);
```

```
COMMIT;
```

SQL Basics: Daten selektieren

```
SELECT *
  FROM location,
       sensor,
       measurement
WHERE meas_loca_id = loca_id
  AND meas_sens_id = sens_id
  AND sens_type = 'Temp'
ORDER BY meas_time DESC;
```

Links und Ressourcen

https://www.postgresqltutorial.com/

https://hub.docker.com/_/postgres

https://www.postgresql.org/docs/current/index.html