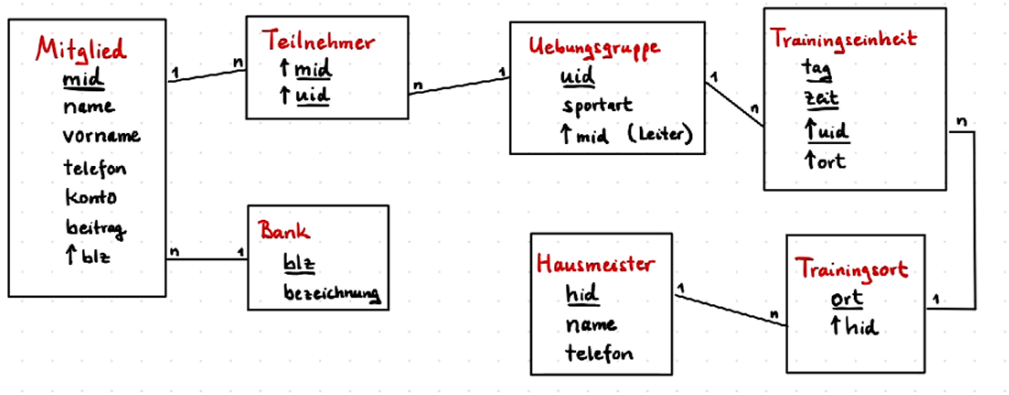
Merkhilfe SQL: Sportverein



Datenbank `sportverein.db`: Primärschlüssel unterstrichen, ↑ Fremdschlüssel, Linien mit 1 und n verbinden Fremdschlüssel und Primärschlüssel

SQL in Python

`cur.fetchall()` Liste von Tupeln

Wenn man nur einen Wert erwartet (z. B. bei `COUNT`), fängt man ihn so auf:

`x = cur.fetchall()[0][0]` erste Zeile, erster Wert

Reihenfolge der Teile

```
SELECT ... FROM ...
  JOIN ... ON ...
  WHERE ...
  ORDER BY ... LIMIT ...
```

Spalten auswählen

`SELECT * FROM Hausmeister` alle Spalten

`SELECT name, telefon FROM Hausmeister`

`SELECT DISTINCT(ort) FROM Trainingseinheit` ohne Doppelte

Zeilen auswählen mit WHERE

`WHERE beitrag > 100` `= <> < > <= >=`

`WHERE tag = 'Montag'` Text in ' '

`AND OR NOT ( )`

`WHERE name LIKE 'Sch%'` % beliebig viele Zeichen

`WHERE name LIKE 'M_ier'` _ genau ein Zeichen

Texte vergleicht SQLite mit = genau, also mit Groß- und Kleinschreibung.

Sortieren und begrenzen

`ORDER BY name` `ORDER BY beitrag DESC`

`ORDER BY name, vorname` mehrere Spalten

`LIMIT 5` nur die ersten 5

Aggregatfunktionen

`SELECT COUNT(*) FROM Bank` Anzahl Zeilen

`COUNT(DISTINCT(a.mid))` ohne Doppelte zählen

`SUM(beitrag)` `AVG(beitrag)` Summe, Durchschnitt

`MIN(beitrag)` `MAX(beitrag)`

Ergebnis ist eine Zeile mit einem Wert, z. B. `(17,)`.

Daten ändern

```
UPDATE Hausmeister
SET telefon = '02381-1234'
WHERE name = 'Krause'
```

Ohne `WHERE` werden **alle** Zeilen geändert! Danach `con.commit()`.

Tabellen verbinden: die Idee

Ein Fremdschlüssel (↑) zeigt auf den Primärschlüssel einer anderen Tabelle. Beim Verbinden folgt man den Linien im Bild und schreibt für jede Linie eine Bedingung *Fremdschlüssel = Primärschlüssel*.

```
Mitglied.blz = Bank.blz
Trainingsort.hid = Hausmeister.hid
```

JOIN ... ON

```
SELECT a.ort, b.name
FROM Trainingsort a
JOIN Hausmeister b ON a.hid = b.hid
```

a, b: Abkürzungen (Alias)

dasselbe ohne JOIN:

```
SELECT Trainingsort.ort, Hausmeister.name
FROM Trainingsort, Hausmeister
WHERE Trainingsort.hid = Hausmeister.hid
```

Mehrere Tabellen verbinden

```
SELECT DISTINCT(c.sportart)
FROM Trainingsort a
JOIN Trainingseinheit b ON a.ort = b.ort
JOIN Uebungsgruppe c ON b.uid = c.uid
WHERE a.ort = 'Halle2'
```

Für jede weitere Tabelle ein weiteres `JOIN ... ON`. Bedingungen mit `WHERE` kommen nach allen JOINs.

Die Wege im Sportverein

Mitglied und Übungsgruppe sind n:m verbunden: der Weg führt über `Teilnehmer`.

```
Mitglied.mid = Teilnehmer.mid
Teilnehmer.uid = Uebungsgruppe.uid
```

`Uebungsgruppe.mid` ist der **Leiter** der Gruppe:

```
Uebungsgruppe.mid = Mitglied.mid
Uebungsgruppe.uid = Trainingseinheit.uid
Trainingseinheit.ort = Trainingsort.ort
Trainingsort.hid = Hausmeister.hid
Mitglied.blz = Bank.blz
```

Gleiche Spaltennamen

`mid`, `uid`, `name`, `telefon`, `ort` gibt es in mehreren Tabellen. Dann den Alias davorschreiben:

`a.name` `f.name` `COUNT(a.mid)`