Mathematisch-Naturwissenschaftliche Fakultät
Wilhelm-Schickard-Institut für Informatik
Datenbanksysteme · Prof. Dr. Grust

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

---

# Datenbanksysteme I

WS 2021/22

Torsten Grust, Christian Duta, Tim Fischer

## Assignment #1

Submission Deadline: November 3, 2021 - 10:00

In this assignment, you will work with JSON and CSV data sets obtained from the Twitter API. If you are not familiar with basic Twitter concepts, please consult Wikipedia. Along with this assignment, we provided you with two JSON files `tweets.json` and `users.json` as well as two CSV files `tweets.csv` and `users.csv`.

**`tweets.json`** An array of tweet objects related to *Tübingen*.

**`users.json`** An array of user objects related to the tweets in `tweets.json`.

**`tweets.csv` and `users.csv`** These contain the same data as the beforementioned JSON files as CSVs.

### Exercise 1: Introduction (0 Points)

1. **Before we are able to grade you** and/or your team, you have to complete one initial task first. In your team's GitHub Classroom repository, there exists a file called `README.md`. Add each team member's name, surname, matrikel number, subject and field of study to the file and commit+push the changes. This task is not part of your submission and does not grant you any points, however this is a **requirement** for your team to be graded in the first place. Please make sure the file always exists with the correct information present.
2. Each assignment submissions in "Datenbankensysteme 1" must be placed into a seperate folder in the `root` directory of your team's GitHub Classroom repository. The name of the folder has to be in the format `solution<number>`. For this assignment, for example, your submission folder would be called `solution01`.
3. In general, the only accepted file formats are PDF of appropriate size (< 2MB), plain text files (.txt) and source files (.sql, .py, ...). Other formats may not be graded, unless stated otherwise. Your submitted code has to compile and work. If compiling, running and/or understanding your source code is particularly complex, please write documentation accordingly.
4. Lastly, the usual rules of academic integrity (plagiarism, in particular) apply.
5. For any further questions about this lecture, assignment and other related topics, visit the forum at

   https://forum-db.informatik.uni-tuebingen.de/c/ws2122-db1.

### Exercise 2: Twitter Mini World (5 Points)

Explore the JSON data sets. Documentation on the semantics of particular fields can be found in the Twitter API documentation.

Then, characterize the data set in terms of the "Mini World" vocabulary (objects, attributes, relationships, constraints) discussed in the lecture. Do not exhaustively describe all attributes (there are many!), but select a few which seem important to you.

### Exercise 3: Structured Information (5 Points)

Describe informally and *briefly* how relationships between objects in the CSV files are encoded in the data. What are the differences in comparison to the JSON files?

### Exercise 4: JSONiq/PyQL Queries (20 Points)

We will now query the provided data sets using the query languages JSONiq and PyQL. Please consider the hints at the end of the assignment sheet **prior** to working out solutions.

Implement each of the following queries in (a) JSONiq and (b) PyQL.

1. Return the text of all tweets having more than 5 retweets (based on `retweet_count`).
2. Return all tweets of the user with the screen name `Tagblatt`.
3. For each user, count the number of the tweets she published. (Not based on `statuses_count`)
   **Hint:** The result size for this task is 397 which exceeds the default result size limit of RumbleDB and triggers a warning. To work around this, simply use `--materialization-cap 1000` when running your JSONiq query in RumbleDB.
4. For every natural language ('de', 'en', ...) calculate the number of tweets that have been published.
5. Return screen names of all users for which no tweet exists in the data set.
   **Hint:** Remember the JSONiq `empty()` function and consider `len([]) == 0` in PyQL.

#### Additional information on JSONiq

Running JSONiq queries requires a JSONiq engine like RumbleDB. Refer to the at documentation on how to install RumbleDB on your system.

If you run into problems refer to the forum and in particular the forum post which describes roughly the steps needed to install RumbleDB.

#### Additional information on PyQL

The provided CSV files can be loaded using the `DB1` Python module found on the course website and queried using the PyQL Python subset as discussed in the lecture.

- The Python documentation and tutorial may be helpful.
- Duplicates can be eliminated by converting a list to a set and then back to a list i.e.: `list(set(xs))`.
- PyQL (i.e. Python) comprehensions can have multiple generators:

```
[ <e> for <x> in <xs> for <y> in <ys> ]
```