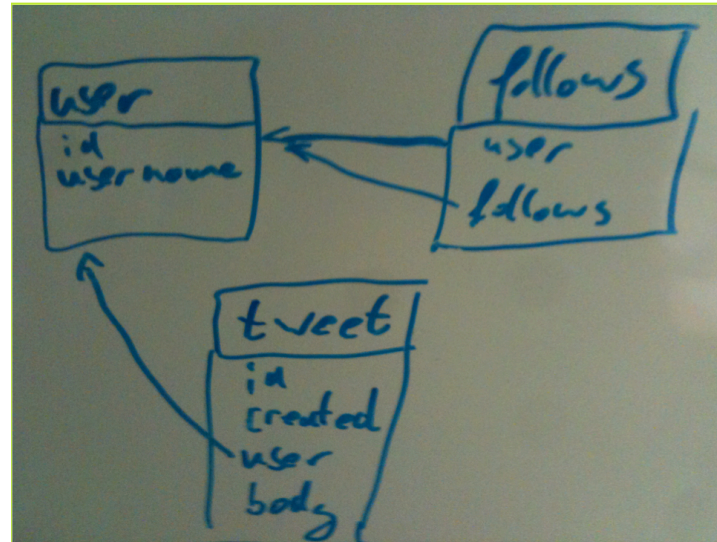




Aufgabe + Szenario

- Arbeit in Teams, wöchentliche Tasks + Präsentation + Feedback
- Jedes Team nutzt eine andere NoSQL-DB
- Einfaches (?) Twitter-Szenario: User, Tweets, Follow-Beziehung



- REST-Interface ist zu Implementieren
 - Bulk-Import
 - Timeline Query
 - (+insert Tweet)
- Relationale Baseline-Implementierung in Postgres+Python ist zu schlagen



HTTP-Post Requests

- JSON-encoded payload

Queries

- /tweets {„username“: „wl“, „start“:0, „limit“: 40} **[Primär]**
- /post_tweet {„username“: „wl“, „body“: „Schon wieder WSDM lesen ...“}
- /create_user {„username“: „wl“, „password“: „xyz“}
- /add_friend {„username“: „wl“, „friend“: „dh“}
- / (Public Timeline)

Für Datenbank-Setup

- /import_friends {„username“: „wl“, „friends“: [„dh“, „mt“]}
- /import_tweets {„username“: „wl“, „bodies“: [„<Text>“, „<Text>“], „timestamps“: [23143252, 2323432]}
- /delete_tweets_afer{„delete_afer “: 2323432}



<http://www.youtube.com/watch?v=b2F-DltXtZs>

Bis Nächste Woche

- Installieren der gewählten Datenbank auf einem lokalen (Linux-)Rechner
 - Wenn nötig: Einrichten einer Entwicklungs-VM
- Installieren des Workload-drivers und des Datenbanksetup-Skripts (next Slide)
- Nächsten Dienstag: 3 Minuten Demo am Testrechner:
 - Verbinden zur Datenbank + Feature-Demo (z.b. Anfragetypen)
- Nötige Software findet sich auf der Lehrveranstaltungsseite



Install requirements

- `sudo apt-get install python-matplotlib` [plotting lib]
- `sudo apt-get install python-mechanize` [simulates browser/user]
- `(sudo apt-get install pip` [python package management])
- `sudo pip install multi-mechanize` [testing framework]

Create the Twitter database

- `python setup_db.py -s 1000 -H 127.0.0.1 -p 5000 load_tweets` [size, host, port, task]
- Tasks: `load_users`, `load_follows`, `load_tweets`

Run the workload driver

- `multimech-run wldriver-r 1000 http://127.0.0.1:5000`
- `http://testutils.org/multi-mechanize/`