

FH HAGENBERG

PROJEKTARBEIT

Weather Tracer - Dokumentation

Autor:

Daniel ENGLISCH
Andreas ROITHER

Übungsleiter:

Daniel SKLENITZKA

15. November 2018

Version 1.0



Inhaltsverzeichnis

1 Datenbankdesign	2
2 Extractor	4
3 Generator	4
4 Installationsanleitung	5
4.1 Benötigte Programme und Voraussetzungen	5
4.2 Datenbank	5

1 Datenbankdesign

Dieses Projekt wurde mit einer MySQL Datenbank auf Version 5.7.23 realisiert. Es wurden zuerst die geforderten Entitäten aus der Angabe extrahiert und mithilfe eines grafischen Modellierungswerkzeugs namens MySQL Workbench ¹ modelliert.

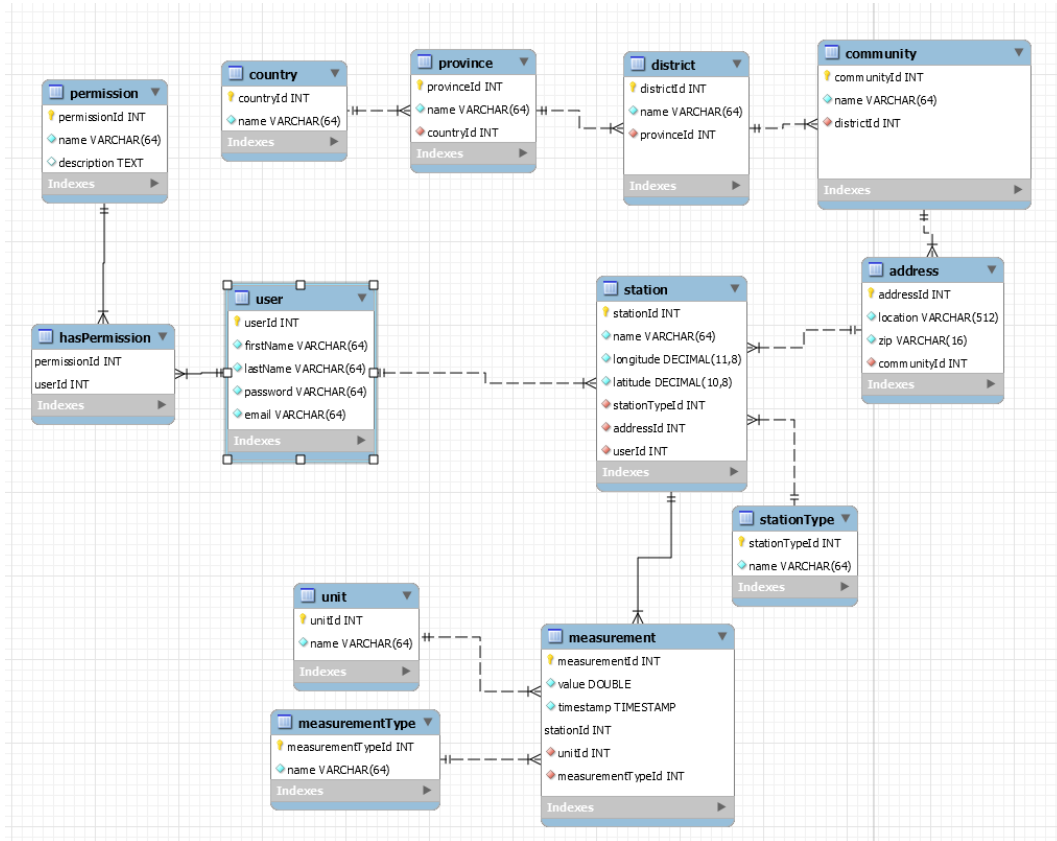


Abbildung 1: Datenbankschema des Wetr-Projekts in der ersten Ausbaustufe.

Wie in Abbildung 1 zu sehen wird zur Verwaltung des Standortes einer Station eine Reihe von abhängigen Entitäten verwendet. Um die Flexibilität zu erhöhen wurde neben zusätzlich ein *Country* modelliert. In einem *Country* befinden sich *Provinces*, welche Bundesländer darstellen. Jede *Province* wird in mehrere *Districts* unterteilt, ähnlich wie Bezirke. In jedem *District* gibt es mehrere *Communities*, welche mit Gemeinden vergleichbar sind. Als kleinste Entität in dieser Kette gibt es die *Address*, welche einen einfachen String zur Angabe von genaueren Adressdaten (rein zur Anzeige oder falls anderswo benötigt) und eine Zuordnung mittels Postleitzahl enthält.

Im Datenbankschema gibt es *User*, welche, falls benötigt, verschiedene *Permissions* zugewiesen haben können. Ein *User* kann mehrere *Stations* betreiben, welche wiederum neben der *Address* auch einen Namen und die Geokoordinaten in Form von Latitude und Longitude gespeichert hat. Der Typ der Station wurde in eine eigene Entität *StationType* ausgelagert.

Jede *Station* kann beliebig viele *Measurements* generieren, welche neben den ebenfalls ausgelagerten Entitäten *MeasurementType* und *Unit*, auch einen Zeitstempel und dazugehörigen Messwert besitzen.

¹<https://dev.mysql.com/downloads/workbench/>

2 Extractor

Der *Extractor* (befindet sich im *extractor* Ordner) wurde mit Python² geschrieben und verwendet die Stationsliste der *Zentralanstalt für Meteorologie und Geodynamik*³. Die “.csv” Datei wird vom *Extractor* eingelesen und für jede *Station* wird eine Insert Anweisung in eine “.txt” Datei geschrieben. Zusätzlich wird anhand des Längen- und Breitengrades der Ort mit einem *geolocator* der Aufenthaltsort der Station ermittelt. Anhand dieser Daten werden SQL Anweisungen für die *Community* und *Address* Tabellen erstellt die von der jeweiligen Stationen referenziert werden.

3 Generator

²<https://www.python.org/>

³<https://www.zamg.ac.at/cms/de/klima/messnetze/wetterstationen>

4 Installationsanleitung

4.1 Benötigte Programme und Voraussetzungen

Für dieses Projekt wird zusätzliche Software benötigt:

- Docker ⁴
- Visual Studio ⁵
- MySql-Connector (optional, nur falls notwendig) ⁶

Um die Datenbank zu erstellen muss Docker gestartet sein. Die Datenbank kann mit dem *Powershell-Script* "run.ps1" automatisch generiert werden. Falls dieses Script wegen fehlender Berechtigungen nicht ausgeführt werden kann, muss eine *Shell* (Git-Bash oder ähnliches) im Ordner mit der Docker Compose Datei "docker-compose.yaml" geöffnet und folgenden Befehle nacheinander ausgeführt werden:

```
docker stop $(docker ps -a -q)
docker rm $(docker ps -a -q)
docker-compose up --build --force-recreate
```

4.2 Datenbank

Für die Datenbank muss die SQL Datei "create-wetr.sql" im Ordner sql/Create ausgeführt werden. Falls das Unit Testing auch ausgeführt werden soll, wird die zweite SQL Datei "create_wetr-unit.testing.sql" auch benötigt.

Für die Füllung der Datenbank kann die SQL Datei "InsertEverythingWithoutMeasurement.sql" verwendet werden. Für die Measurement Daten muss spezieller vorgegangen werden dadurch kann aber die Insert Zeit drastisch verringert werden:

- Wetr.Generator.exe im Ordner sql/Insert starten
- Die resultierenden ".bulk" Dateien in den sql Ordner kopieren (der phpMyAdmin Container wurde mit dem sql Ordner verbunden um diese Dateien verwenden zu können)
- <http://localhost:8080/> im Browser aufmachen (phpMyAdmin sollte starten)
- Benutzer: root Passwort: 0c1cd84e
- Datenbank wetr auswählen
- Zum SQL Tab wechseln
- Die nachfolgenden SQL Anweisungen eingeben und ausführen

⁴<https://www.docker.com/>

⁵<https://visualstudio.microsoft.com/de/>

⁶<https://dev.mysql.com/get/Downloads/Connector-Net/mysql-connector-net-8.0.13.msi>

```
LOAD DATA LOCAL INFILE "/tmp/sql/insert/measurementsDownfall.bulk" INTO
↪ TABLE measurement
FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY '\r\n';

LOAD DATA LOCAL INFILE "/tmp/sql/insert/measurementsHumidity.bulk" INTO
↪ TABLE measurement
FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY '\r\n';

LOAD DATA LOCAL INFILE "/tmp/sql/insert/measurementsTemperature.bulk" INTO
↪ TABLE measurement
FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY '\r\n';

LOAD DATA LOCAL INFILE "/tmp/sql/insert/measurementsWind.bulk" INTO TABLE
↪ measurement
FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY '\r\n';

LOAD DATA LOCAL INFILE "/tmp/sql/insert/measurementsWindDirection.bulk"
↪ INTO TABLE measurement
FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY '\r\n';

LOAD DATA LOCAL INFILE "/tmp/sql/insert/measurementsPreassure.bulk" INTO
↪ TABLE measurement
FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY '\r\n';
```