Anpassung der Daten, um sie nutzbar zu machen. Es wurde dafür ein Python Skript geschrieben, welches die Yelp Daten mit den Inspektionsdaten verknüpft hat. Dafür wurden allen eindeutig zuordenbaren Restaurants in dem Inspektionsdatensatz eines der meistbewerteten Businesses aus Yelp mit der Kategorie Restaurant zugeordnet und in einer mapping Tabelle dargestellt.

fakingScript.py

Danach müssen die Daten bereinigt werden. Lokale ohne eindeutigen INSPECTION DATE (hier 01/01/1900) müssen entfernt werden, sowie die den zugehörigen

Database-Name: PortfolioPruefung

Yelp: yelp\_business

Inspektionen: inspection\_results

1. fakingScript (ändert yelp daten und verbindet yelp und inspections)
2. consistencyCheck (prüft ob alle Datensätze dieselben Datenfelder hat)
3. consistencyEnforcement (fügt fehlende Datensätze hinzu
4. deleteRestaurantsWithoutInspections (Restaurants ohne Inspection werden gelöscht)

(dataTypesCheck unnötig wenn oben genannte Methoden richtig implementiert)

1. searchDuplicates (sucht nach Duplikate -> findet keine)
2. weitere Skripte notwendig: daten ohne geodaten löschen, consistency check für andere datentypen

121 daten ohne geodaten sind statistisch irelevant -> gelöscht

Zwei SQL-Tabellen benötigt

1. Yelp Daten
2. Inspection results

Yelp Daten: Object ID = Primärschlüssel

Business Id, Name, Address, Postal Code, Borough, latitude, longitude, Phone, Stars, Reviews, Cuisine Description

Inspection results: Inspection (Object) ID = Primärschlüssel, KAMI (Business ID) = Fremdschlüssel

Inspection Date, Action, Violation Code, Violation Description, Critical Flag, Score, Grade, Inspection Type, Community Board, Council District, Census Tract, BIN, BBL, NTA

Daniel: SQL-Abfragen in MySQL

Niklas: Präsentation

Silas: Python-Skripte