Gruppe G Erklärung der CA 3 Lessons Learned

Datenbank

Für die Umsetzung der CA-Abgabe mussten wir zunächst eine geeignete NoSQL-Datenbank finden, die die benötigten Features bereitstellt. Hierfür haben wir uns drei verschiedene Datenbanken angeschaut. Nach der Recherche viel unsere Wahl auf die Couchbase NoSQL-Datenbank.

DynamoDB von Amazon:

Lessons-Learned

Die Datenbank DynamoDB von Amazon bietet zwar die Funktionalität, mehrere Nodes zu verwenden, leider wird diese Funktion nur bei einem kostenpflichtigen Abo bereitgestellt.

Riak:

Lessons-Learned

- Riak wurde 2009 entwickelt und fühlte sich nach genauerer Recherche nicht mehr nach der State of the Art an. Weshalb wir bedenken hatten diese Datenbank zu verwenden.
- Docker Images recht alt.
- Letztes Update gab es vor 5 Jahren.

Docker-Compose

Beim Hochfahren des Servers mittels Docker-Compose-File werden den einzelnen Nodes IP-Adressen zugewiesen. Anschließend müssen beim Erstellen des Clusters diese IP-Adressen auf der Startseite des Webinterfaces von Couchbase hinterlegt werden.

Wurde in dem Compose-File die Startreihen der einzelnen Nodes nicht fest vorgegeben, kann es passieren, dass die Nodes beim nächsten Start in einer anderen Reihenfolge hochgefahren werden. Hierdurch können Konflikte mit dem angelegten Cluster auftreten da die IP-Adressen nicht mehr übereinstimmen. → Nodes können nicht mehr angesprochen werden.

Lessons-learned

- Es muss eine festen Startreihenfolge vergeben werden, damit jeder Node wieder die richtige IP-Adresse zugewiesen bekommt.
- Logging kann den Nutzer in die Irre führen.
- Der Docker Container meldet wenige Sekunden nach dem Start dem Nutzer, dass er bereit ist. Jedoch kann es passieren, dass dieser erst nach ein paar Minuten wirklich erreichbar ist.

Docker Port-Mapping

Lessons-learned

- Normalerweise wird nur eine Instanz eines Docker Container pro Rechner verwendet,
- Bei der Verwendung von mehreren Instanzen auf einem Rechner muss das Portmapping von Nutzer vorgenommen werden. **Sehr aufwendig und kompliziert.**

Vorverarbeitung der Daten

Für die Aufbereitung der Daten, um diese in unser gewählten Datenmodelle zu bringen. Werden die Daten in der Python Datei **preprocessing.py** vorverarbeitet.

Zuvor hatten wir versucht die Daten während der Laufzeit für den Upload anzupassen. Für die Überprüfung, ob die Daten bereits vorhanden sind, wurden bei jedem Element die kompletten Daten geladen und jedes Mal geprüft, ob sich das Element bereits also Value abgelegt wurde.

➔ **Verursacht sehr lange Verarbeitungszeiten!**

Lessions-learned

Daten sollten auf jeden Fall vorher schon in das passende Format gebracht werden, um vielfachen Zugriffe auf die Datenbank zu umgehen.

Flask - Web Development

Lessions-learned

- Für das builden des Flask images wird wheel benötigt, es kann passieren, dass das Builden abstürtzt.
- Pip install wheel schafft hier abhilfe.

Couchbase – NoSQL Abfragen

Lessions-learned

- Die Couchbase Datenbank besitzt eine eigene SQL-Like-Query-Sprache für JSON namens N1QL.
- N1QL sehr ähnlich zu SQL.
- Queries dadurch intuitiv und leichter umstieg von SQL zu NOSQL möglich.

Tabel-Plus

Lessions-learned

Während unserer Recherchen für das Aufsetzen des Couchbase Clusters, sind wir auf die Anwendung TablePlus gestoßen. TablePlus ist eine moderne, native App mit einer sauberen Benutzeroberfläche, die es Entwicklern ermöglicht, Datenbanken gleichzeitig sehr schnell und sicher zu verwalten. TablePlus unterstützt die meisten gängigen Datenbanken NO-SQL wie Mongo DB, Cassandra und viele mehr. Leider wird Couchbase nicht unterstützt.

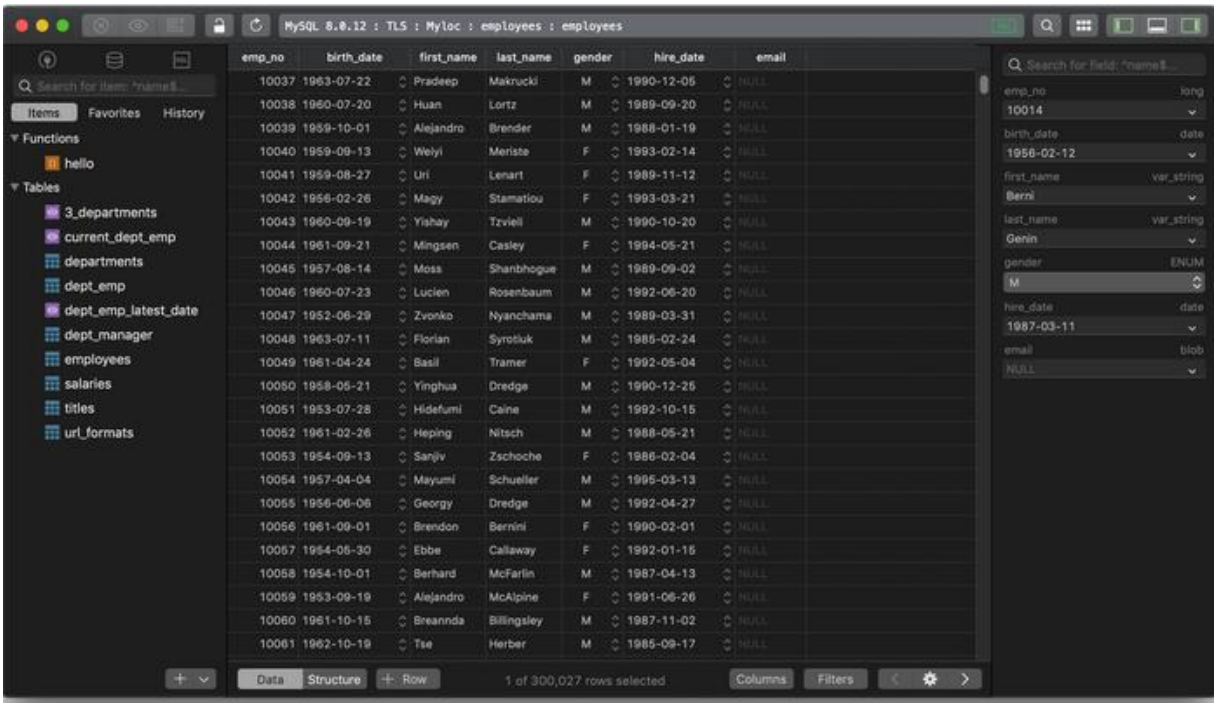

Abbildung 1: Ausschnitt von TablePlus