Prüfungsleistung: Projektarbeit

Im Rahmen des Moduls Neue Datenbankkonzepte (NDBK) sind als Prüfungsleistung die folgenden Projektaufgaben zu erfüllen.

Teamarbeit

Die Projektarbeit wird i.d.R. im 2er-Team bearbeitet. Arbeitsteilung ist hierbei sinnvoll und erwünscht. Trotzdem muss jedes Teammitglied während der Präsentationen zu allen Aspekten der Projektaufgaben auskunftsfähig sein, d.h. die eigene Implementierung erklären können.

Die Bearbeitung der Projektaufgaben erfolgt weitgehend außerhalb der Labortermine im Selbststudium. Der Zeitplan im Moodle-Kurs weist Labortermine aus, bei denen die Laborbetreuer bezüglich Fragen und Problemen behilflich sind.

Der individuelle Beitrag zum Teamerfolg wird geprüft. Bei einer klaren Ungleichverteilung der Beiträge im Team kann die Benotung der Teammitglieder voneinander abweichen.

Anwesenheitspflicht

Zu den Laborterminen mit Präsentationen besteht Anwesenheitspflicht. Bei Verhinderung, z.B. aufgrund von Krankheit, ist der zuständige Laborbetreuer zu informieren. Nach Rücksprache kann ggf. ein alternativer Labortermin wahrgenommen werden

Anwendungsszenarien

Wähle nach Absprache mit dem Laborbetreuer eines der angebotenen Anwendungsszenarien für dein Team und bearbeite die folgenden Projektaufgaben (PA) hinsichtlich dieses Anwendungsszenarios. Die Anwendungsszenarien werden im Detail in einem separaten Foliensatz vorgestellt.

Zu jedem Anwendungsszenario stehen die folgenden Informationen zur Verfügung:

- Überblick über das vereinfachte Datenmodell in der Notation eines UML-Klassendiagramms
- Wireframe eines ausgewählten Screens in einer zugehörigen mobilen Anwendung
- CSV-Dateien, deren Datensätze in eine Datenbank (DB) zu importieren sind
- SQL-Skript, über das die CSV-Dateien in eine relationale MariaDB-/MySQL-Datenbank geladen werden können
- Anforderungen an die zu entwickelnde API

PA1 Indizes in einer relationalen Datenbank

- Importiere die CSV-Dateien mit Hilfe des SQL-Skripts in eine relationale MariaDB- oder MySQL-Datenbank.
- Die Information, die auf dem Wireframe dargestellt ist, soll möglichst schnell mit einer SQL-Abfrage aus der DB gelesen werden, d.h. mit einer Zugriffszeit i.d.R. < 200 ms. Dazu fehlt der DB nach dem Import über das gegebene SQL-Skript mind. 1 sinnvoller Index. Ergänze sinnvolle Indizes explizit über `create index` oder implizit über Fremdschlüssel.

- Je Anwendungsszenario gibt es eine *zusätzliche statistische Auswertung*, die in PA1 bis PA3 nur für den sehr guten Notenbereich zu bearbeiten ist und deren Zugriffszeit durch Hinzufügen eines Indexes in den Bereich < 5 s reduziert werden soll.

PA2 Entwicklung einer REST-API

- Entwerfe und implementiere für das Anwendungsszenario eine API als *REST-basierter Web Service* in einer der folgenden Technologien:
  - Java mit dem Framework Spring Data[1]
  - JavaScript/TypeScript mit dem Framework Express und Sequelize als ORM
  - Python mit dem Framework Flask und SQLAlchemy als ORM
  - andere Vorschläge nach Absprache mit dem Laborbetreuer

  Die konkreten Anforderungen an die API hinsichtlich ihrer Endpunkte sind im Anwendungsszenario genauer beschrieben.

- Prüfe, welche Indizes aus PA1 bei den lesenden Zugriffen auf die API genutzt werden und ob sich angemessene Antwortzeiten ergeben.

PA3 Migration der Daten in eine nicht-relationale DB

- Installiere ein nicht-relationales Datenbanksysteme (DBS) und setze dich mit seinen grundlegenden Konzepten auseinander. Zur Verfügung stehen z.B. folgende DBS: MongoDB, Neo4J, Redis, ScyllaDB. Alternativ sind eigene Vorschläge für ein nicht-relationales DBS willkommen, wobei dieses mind. eingeschränkt unter einer Open Source-Lizenz verfügbar sein muss.
- Migriere die Daten aus der relationalen DB von PA1 in eine nicht-relationale DB, die über das gewählte DBS zur Verfügung gestellt wird. Die Datenstrukturen sind dabei sinnvoll anzupassen, was häufig nicht einer 1:1-Abbildung des relationalen Modells entspricht.
- Passe die API aus PA2 dahingehend an, dass sie nun auf die nicht-relationale DB zugreift.[2]
- Bereite dich darauf vor, die wesentlichen Unterschiede des gewählten DBS im Vergleich zu einem relationalen DBS zu präsentieren.

PA4 Verarbeitung von Datenströmen mit Kafka

- In der Vergangenheit wurden an dieser Stelle Tweets von der ehemaligen Twitter-Plattform als Streaming-Daten verarbeitet. Die Streaming API von Twitter ist aber aktuell kostenpflichtig. Daher werden folgende alternative Quellen für Streaming-Daten empfohlen:
  - Die Änderungen an sämtlichen Wikipedia-Artikeln werden als Stream über eine EventSource-API zur Verfügung gestellt.
  - Unter https://awesomedata.stream/ gibt es aktuell zu vier unterschiedlichen Themen Streaming-Daten, die über einen öffentlichen Kafka-Broker abonniert werden können.

  Alternativ kann ein beliebiger anderer Stream über eine entsprechende API konsumiert und sinnvoll verarbeitet werden. Ideen gibt es z.B. hier.

[1] Hilfreich zum Einstieg in Spring Data: Building a RESTful Web Service, Accessing Data with JPA
[2] Spring Data unterstützt mit jeweils einem entsprechenden Modul die oben genannten DBS.

Die Daten sollen in ein Kafka-Topic in einem selbst bereitgestellten Kafka-Broker eingetragen werden. Die dazu verwendete Programmiersprache zum Zugriff auf die Datenquelle ist freigestellt.

- Es sollen anschließend die eingehenden Ereignisse nach einem beliebigen Attribut gruppiert und anschließend für bestimmte Zeitfenster (z.B. 10 Min.) gezählt, summiert, gemittelt werden o.ä. Dafür soll ksqlDB oder Kafka Streams verwendet werden. Die Ausgabe soll in ein zweites Kafka-Topic erfolgen.
- Das zweite Kafka-Topic mit den aggregierten Daten soll wiederum von einem Konsumenten abonniert und in die API aus PA2 integriert werden. Die Daten sollen über eine `EventSource` bereitgestellt werden (d.h. bei Spring Data wäre die Rückgabe des Endpunkts vom Typ `Flux`).
- Es soll ein Beobachter auf der `EventSource` registriert werden und die aggregierten Daten in einer Zeitreihe, die laufend aktualisiert wird, als Balkendiagramm, Liniendiagramm o.ä. darstellen, z.B. in einer einfachen Web-Anwendung. Auf der vertikalen Achse soll die Anzahl der Erwähnungen abgetragen werden, auf der horizontalen Achse die Zeit.

Bewertung

Die konkreten Labortermine zur Präsentation der Projektaufgaben PA1 bis PA4 sind dem Zeitplan im Moodle-Kurs zu entnehmen. Die Projektaufgaben sind jeweils vor dem jeweiligen Labortermin umzusetzen und die zugehörige Implementierung in einem Git-Repository, auf das der Laborbetreuer zugreifen kann, zu hinterlegen.

Neben der Vorstellung der eigenen Lösung zur Projektaufgabe müssen auch inhaltliche Fragen zur Implementierung durch den Laborbetreuer beantwortet werden können.

Die Projektaufgaben werden nach dem letzten Labortermin durch den Laborbetreuer hinsichtlich der folgenden Kriterien bewertet:

Kriterium	**Gewicht**	**← sehr gut**				**mangelhaft →**
Erfüllungsgrad PA1	25%					
Erfüllungsgrad PA2	25%					
Erfüllungsgrad PA3	25%					
Erfüllungsgrad PA4	25%					