

Praktikum Datenbanksysteme

Sommersemester 2017

Aufgabenblatt 5 – Graphdatenbanken

(Aufgabe 5: max. 7 Punkte; insgesamt 45 Punkte für fünf Praktikumsaufgaben)

Gegeben sei ein ERM für eine Rezepte-Datenbank (s. letzte Seite).

Die beauftragende Firma hat vor, von dem relationalen *MySQL*-DBMS auf das Graph-DBMS von *Neo4j* umzusteigen, da das System in Zukunft dynamisch erweitert wird. Ihre Aufgabe als Datenbankspezialist ist es nun, das relationale Modell zunächst schematisch als Graph-Modell darzustellen. Hierfür bittet sie der Lead-Developer der Firma das „Arrow-Tool“ (<http://www.apcJones.com/arrows/>) zu nutzen, welches sich bereits zur Dokumentation im zuständigen Team etabliert hat. Mit diesem ist es möglich, Knoten und Beziehungen, inklusive Eigenschaften, digital darzustellen. Gefordert ist ein Screenshot des kompletten Graphs im sog. „Bootstrap-Style“ des Tools (0,75 P).

Ergänzend zur Abbildung bekommen sie den Hinweis, dass die Eigenschaften „user_name“, „recipe_name“, „ingredient_name“, „difficulty_name“ und „meal_type_name“ einzigartig sind. Als Codekonventionen werden die etablierten Cypher-Konventionen erwartet sowie alle Eigenschaften im „camelCase“ beginnend mit einem „lower case character“ beschrieben.

Es sind bereits einige Datensätze vom bereits laufenden System vorhanden. Diese liegen als aufbereitete CSV-Dateien vor (auf einem Windows-Betriebssystem erstellt) und müssen in die Datenbank importiert werden (1,5 P; Tipp: Neo4j CSV-Import). Darunter zählen folgende:

- ingredient_to_recipe.csv
- ingredients.csv
- recipes.csv
- user_has_favourite.csv
- user_has_mealplan.csv
- user_has_mealplan_includes_recipe.csv
- userRated_recipe.csv
- users.csv

Hinweise: Beim Einfügen bzw. jeder Änderung an einem Rezept wird die Eigenschaft „last_modified“ mit dem aktuellen Zeitstempel aktualisiert. Die Eigenschaft „rating_average“ wird automatisiert zur Laufzeit berechnet und ist nicht im initialen Import enthalten.

Nach dem Einfügen der Daten sollen folgende Fragen/Aufgaben gelöst werden:

- 1) Finden sie alle Rezept-Knoten (0,25 P)
- 2) Finden sie alle Rezepte und zeigen sie nur die Spalten Name, Preparationszeit, Ruhezeit und ordnen sie diese nach aufsteigender Preparationszeit (0,25 P)
- 3) Finden sie alle Rezepte und zeigen sie den Rezeptnamen und die wirkliche Kochzeit (Preparationszeit und Ruhezeit) absteigend geordnet nach wirklicher Kochzeit an. (0,25 P)
- 4) Finden sie alle Rezepte, die vom Typen "Hauptspeise" sind, als auch den Schwierigkeitsgrad "normal" besitzen und zusätzlich die Ersteller dieser Rezepte. Zeigen sie nur Nutzernamen und Rezeptnamen an. (0,25 P)
- 5) Finden sie alle Nutzer, die ein Rezept erstellt haben und geben sie diese in einer Spalte wieder. Jeder Nutzer soll maximal einmal aufgeführt werden. (0,25 P)
- 6) Updaten sie für alle Rezept-braucht-Zutat Relationen in denen die Einheit "nach Bedarf" ist die Eigenschaft Menge zu "etwas" (Vorsicht mit Update Befehlen, nichts falsches Updaten! Ansonsten muss neu importiert werden) (0,25 P)
- 7) Entfernen sie die Abkürzung für alle Rezept-braucht-Zutat Relationen, in denen die Abkürzung gleich dem Einheitsnamen ist (Vorsicht mit Update Befehlen, nichts falsches Entfernen! Ansonsten muss neu importiert werden) (0,25 P)
- 8) Führen sie ein Update auf alle Rezepte aus, das den Rezepten die Eigenschaft durchschnittliche Bewertung (Integer), berechnet durch die jeweiligen Bewertungen, hinzufügt. (0,25 P)
- 9) Finden sie das Rezept mit der höchsten durchschnittlichen Bewertung und zeigen sie nur den Namen und die durchschnittliche Bewertung an (0,25 P)
- 10) Finden sie alle Zutaten, welche für die Rezepte des Essensplans bzw. der Essenspläne des Nutzers "CooperKilvington" benötigt werden und zeigen sie diese in einer Liste an. Jede Zutat darf nur einmal in der Liste vorhanden sein (0,25 P)
- 11) Finden sie für alle Rezepte, wie oft diese jeweils favorisiert wurden und geben sie die Rezeptnamen sowie die Anzahl der Favorisierungen nach absteigender Favorisierungsanzahl aus. (0,5 P)
- 12) Erweitern sie das System um Freundschaften zwischen Usern. Für den Import der Freundverbindungen steht die CSV-Datei „users_are_friends.csv“ bereit. (Tipp: PERIODIC COMMIT) (0,25 P)
- 13) Finden sie alle Nutzer mit denen "KaliSavile" befreundet ist. (0,25 P)
- 14) Finden Sie alle Rezepte, die in einem Essensplan von einem Nutzer enthalten sind, mit welchem der Nutzer "KaliSavile" über drei Kanten befreundet ist. Außerdem soll der über drei Kanten befreundete Nutzer das Rezept auf dem Essensplan mit einer Wertung von 5 bewertet haben. Geben sie nur den Namen des Rezeptes bzw. der Rezepte aus. (0,5 P)
- 15) Finden Sie alle Rezepte, die in einem Essensplan von einem Nutzer enthalten sind, mit welchem der Nutzer "KaliSavile" über drei Kanten befreundet ist (Freunde von Freunden von Freunden). Außerdem soll der über drei Kanten befreundete Nutzer das Rezept auf dem Essensplan mit einer Wertung von 5 bewertet haben. Diese Gerichte sollen dem Nutzer "KaliSavile" empfohlen werden und dürfen deshalb nicht in einem Essensplan von "KaliSavile" enthalten sein. Geben sie nur den/die Namen des/der Rezepte in einer Liste aus. (0,75 P)

- Kopieren Sie die Lösungen aller Teilaufgaben (Cypher-Code) in eine Textdatei (Endung .txt).
- Die Abgabe Ihrer Textdatei erfolgt in Moodle unter der entsprechenden Aufgabe. Alle Gruppenmitglieder geben dieselbe Datei ab.

Ihre Datei MUSS nach folgendem Schema benannt werden:

db17-pr-aufg5-<Nachname Bearbeiter1><1. Buchstabe Vorname Bearbeiter1>-<Nachname Bearbeiter2><1. Buchstabe Vorname Bearbeiter2>.txt

Der erste Vorname genügt.

Beispiel: db17-pr-aufg5-MüllerP-MeierK.txt

Achten Sie darauf, dass die Reihenfolge der Bearbeiter über alle Aufgabenblätter gleichbleibt! ~~db17-pr-aufg4-MeierK-MüllerP.sql~~

- Reichen Sie Ihre Ergebnisse bis spätestens zum **27.06.2017, 23:55 Uhr (Gruppe 1)**, bzw. **04.07.2017, 23:55 Uhr (Gruppe 2)** als **Textdatei** zur Bewertung **in Moodle** ein.

