

Datenbanksysteme (DBS) SQL-Grundlagen

WS 2020/21

SQL - Abfragen

2. Übung - 1. Pflichtübung

Prof. Dr. Frank Kammer
TH Mittelhessen, FB MNI

Frank.Kammer@mni.thm.de

1. Testat-Übung

Übungsnr.	Thema	Art	Ausgabedatum	Spätestes Abgabedatum	Gruppengröße
2. Übung	SQL-Abfragen	Testat	ab Mo, 09.11.2020 im Moodle-Kurs	Fr, 12.12.2020 18:00 über das Feedbacksystem Präsentation im Tutorium: 01./02.12. oder 08./09.12.	1

Die **2. Übung ist eine Pflichtübung**, die **von jedem Studierenden selbstständig** durchzuführen ist.

Ihre Lösungen reichen Sie ins Feedbacksystem über folgende Seite:

<https://feedback.mni.thm.de/courses/157>

ein. Bitte beachten Sie, in Ihren Lösungen darf **kein Datenbankname** vorkommen.

Teil 1 und 2 sind für das Erlangen des Testates vollständig von jedem Studierenden so zu lösen, dass das System Ihre Lösung akzeptiert. Es erfolgen auf dem System automatische Plagiat-Tests, die sowohl die Originallösung als auch die Kopie disqualifiziert. Geben Sie also Ihre Lösung nicht weiter.

Zum Erstellen Ihrer Lösungen sind geeignete eigene Tabellen in ihrer Datenbank anzulegen. Achten Sie auf die korrekten Namen der Tabellen und Attribute.

Für das Lösen von Teil 3 können Bonuspunkte für die Klausur erworben werden.

Für je 4 gelöste Aufgaben aus Teil 3 bekommen Sie einen Bonuspunkt. Maximal können so 4 Bonuspunkte erworben werden. Es wird nicht aufgerundet!

Beispiel: Sie haben 9 Aufgaben aus Teil 3 gelöst: 2 Bonuspunkte.

Während den Übungsterminen vor dem Abgabetermin können Fragen zu den Aufgaben bzw. zu ihren Lösungsansätzen gestellt werden. Es empfiehlt sich daher, die Übungen so früh als möglich zu bearbeiten. Zu den Präsentationsterminen (siehe Ablaufplan) sind die Ergebnisse dem Dozenten zum Erlangen des Testats zu erläutern.

Sie dürfen nach der Präsentation bis zur Deadline weiterarbeiten. Stellen Sie jedoch sicher, dass Sie den Großteil der Aufgaben gelöst haben.

Sollten Sie Bonuspunkte erwerben wollen, so müssen Sie zur Präsentation bereits ein paar der Aufgaben aus Teil 3 gelöst haben.

Zusatzinformation:

Bei einigen Aufgaben kann es sein, dass MySQL Workbench das Ausführen von Befehlen verhindert. Dabei handelt es sich um ein Sicherheitsfeature, welches sich umgehen lässt. Details hierzu stehen am Ende der Installationsanleitung von MySQL Workbench.

Teil 1: Richten Sie die nachfolgenden Tabellen in der Datenbank für die Reisetätigkeiten eines Unternehmens ein und lösen Sie die nachfolgenden Fragestellungen mit SQL-Abfragen.

Tabellen

mitarbeiter (#PNr, Name, *AbtNr, Gehalt)
abteilung (#AbtNr, AbtName)
hotel (#HNr, HName, Kategorie, PLZ, Ort)
reisen (*Mitarbeiter, #*Hotel, #Beginndatum, Dauer, Kosten)
(#=Primärschlüssel, *=Fremdschlüssel)

Vorarbeiten: Legen Sie die vier Datentabellen mit deren Primär- und Fremdschlüssel an. Wählen Sie *Kosten als DECIMAL mit 4 Vorkomma- und 2 Nachkommastellen*. Füllen Sie die Tabellen mit eigenen Datensätzen. Folgendes Mengengerüst müssen Sie mindestens zum Testen anlegen: **5 Mitarbeiter, 10 Hotels, 4 Abteilungen**. Jeder Mitarbeiter hat **mindestens drei Reisen** getätigt. Befüllen sie die Tabelle am besten so, dass die Aufgabenstellungen Ergebnisse liefern.

Erstellen Sie folgende SQL-Anweisungen (diese sollen unabhängig vom Beispieldatenbestand gültig sein und alle Spalten müssen sinnvolle Überschriften erhalten):

1. Liste aller Mitarbeiter aus der *Softwareentwicklung*, mit Personal-Nr. und Namen aufsteigend sortiert nach Personal-Nr.
2. Ermitteln Sie die Mitarbeiter mit dem niedrigsten Gehalt unter Angabe von Personal-Nr., Name und Gehalt.
3. Suchen Sie alle Hotels mit dem Namensbestandteil „City“ im Postleitbezirk 70000 bis 78000 aus der Datenbank heraus und gegeben Sie die Felder HNr, HName, Kategorie, PLZ und Ort aus.
4. Nach Namen absteigend sortierte Liste aller reisenden Mitarbeiter mit Name, Personal-Nr., Anzahl der Reisen und den Durchschnittskosten ihrer Reisen.
5. Namen aller Mitarbeiter, die schon einmal in *Hamburg* waren (jeder Name nur einmal).
6. Name, Personal-Nr und Abteilungsname des Mitarbeiters, der die teuerste Reise gemacht hat (mehrere Mitarbeiter möglich) - inkl. der angefallenen Kosten und des Beginndatums.
7. Angabe aller Paare von Mitarbeitern nur mit Personal-Nr, Hotelname und Beginndatum, die gemeinsam im gleichen Hotel *zum gleichen Datum ihre Übernachtungen begonnen haben*. Die Datensätze sollen nach den Paaren aufsteigend sortiert sein. Beachten Sie auch, dass jedes Paar i,j einmal in einem Datensatz (i,j,...) und einmal in (j,i,...) vorkommt → ein Paar = zwei Zeilen.
8. Anzeige der Hotels mit HNr u. HName, die *mindestens zweimal gebucht* wurden.
9. Geben Sie eine Liste aller bisher gebuchten Hotels mit HNr, HName, Gesamtanzahl der verbrachten Nächte und die durchschnittlichen Kosten pro Nacht, aufsteigend sortiert nach HNr, aus.
10. Anzahl der *Hotels, die nie gebucht wurden*.

Teil 2: Richten Sie die nachfolgenden Tabellen in der Datenbank für die Verwaltung der Projektmitarbeit in einem Unternehmen ein und lösen Sie die nachfolgenden Fragestellungen mit SQL-Abfragen.

Tabellen

mitarbeiter (#PNr, Name, Abteilung)
projektmitarbeit (*PNr, *ProjektNr, Taetigkeit, Stunden)
projekt (#ProjektNr, ProjektName, Beginndatum, Budget)
(#=Primärschlüssel, *=Fremdschlüssel)

Vorarbeiten: Legen Sie die drei Datentabellen mit deren Primär- und Fremdschlüssel an. Füllen Sie die Tabellen mit eigenen Datensätzen. Folgendes Mengengerüst müssen Sie mindestens zum Testen anlegen: **6 Mitarbeiter, 12 Projektmitarbeiten, 5 Projekte**. Befüllen sie die Tabelle am besten so, dass die Aufgabenstellungen Ergebnisse liefern.

Erstellen Sie folgende SQL-Anweisungen (diese sollen unabhängig vom Beispieldatenbestand gültig sein und alle Spalten müssen sinnvolle Überschriften erhalten):

1. Liste aller Projekte mit Namen und Budget, deren *Budget mindestens 175.000, -- EUR* beträgt, sortiert nach fallendem Budget.
2. Ermitteln Sie das Projekt mit dem höchsten Budget und dann das, mit dem geringsten Budget. Geben sie in einer Spalte den *Namen* des Projekts und in einer zweiten Spalte „Budget“, als Text „Groesstes Budget“ und „Kleinstes Budget“ für das entsprechende Projekt an. Hierbei soll der erste Datensatz „Groesstes Budget“ enthalten.
3. Gesamtbudget aller Projekte.
4. Namen aller Mitarbeiter der *Abteilung „IT-Anwendungen“*, dies am *Projekt „Mobile Business Intelligence“* beteiligt sind.
5. Liste der Projekte mit Projektnamen und Anzahl der beteiligten Mitarbeiter.
6. Namen aller Projekte, für die *noch kein Projektleiter benannt* ist.
7. Angabe der Projektleiter mit ihren Gesamtstunden, die aus Projekten mit einem *Budget größer 250.000,00 EUR* resultieren. Gehen Sie davon aus, dass ein Mitarbeiter auch in mehreren Projekten als Projektleiter tätig sein kann. Ein Projektleiter, der in keinem solchen Projekt gearbeitet hat, ist nicht zu nennen.
8. Anzeige der Namen aller *Mitarbeiter, die mit „N“ oder „R“* beginnen
9. Einzelne Projekte sollen einen Budget Puffer erhalten. Zeigen Sie das Projekt "Mobile Business Intelligence" mit zusätzlichen 8%, sowie alle anderen Projekte, die im Jahr 2016 starten mit zusätzlichen 4% Budget an. Geben Sie den Projektnamen, das ursprüngliche Budget und das neue Budget, sortiert nach aufsteigendem Projektnamen, aus.
10. Zeigen Sie für alle Projekte die zugeteilten Mitarbeiter mit Projektnamen, Mitarbeiternamen und Taetigkeit, aufsteigend sortiert nach Projektname und Mitarbeiternamen, an.

Teil 3: Richten Sie die nachfolgenden Tabellen in der Datenbank für eine Kochrezepte-Website ein und lösen Sie die nachfolgenden Fragestellungen mit SQL-Abfragen.

Tabellen

zutat	(#id_zutat, bezeichnung)
rezept	(#id_rezept, *id_benutzer, titel, portionen, dauer)
rezept_schritt	(#*id_rezept, #nr_schritt, titel, anweisung)
schritt_zutat	(#*id_rezept, #*nr_schritt, #*zutat, menge, einheit)
benutzer	(#id_benutzer, benutzername, email)
bewertung	(#*id_rezept, #*id_benutzer, anzahl_sterne, datum)

(# = Primärschlüssel, * = Fremdschlüssel)

Vorarbeiten: Legen Sie die sechs Datentabellen mit deren Primär- und Fremdschlüssel an, wobei Sie für die Spalte „dauer“ in der Tabelle „rezept“ den Datentyp *TIME* verwenden sollen. Die Spalte „einheit“ der Tabelle „schritt_zutat“ soll den Datentyp *ENUM* haben. Mögliche Einheiten sind: g, kg, ml, l, tl und el (Gramm, Kilogramm, Milli-Liter, Liter, Teelöffel und Essöffel)

Füllen Sie die Tabellen mit eigenen Datensätzen. Folgendes Mengengerüst müssen Sie mindestens zum Testen anlegen:

10 Zutaten, 5 Benutzer, 5 Rezepte, 10 Rezeptschritte, 20 Schritt_Zutaten und 5 Bewertungen. Befüllen sie die Tabelle am besten so, dass alle Aufgaben Ergebnisse liefern.

Erstellen Sie folgende SQL-Anweisungen (diese sollen unabhängig vom Beispieldatenbestand gültig sein und alle Spalten müssen sinnvolle Überschriften erhalten):

1. Liste aller Rezepte mit Titel und Dauer in Minuten, die mindestens 4 Portionen produzieren. Ausgabe aufsteigend sortiert nach Benutzer-ID.
2. Rezepttitel und Benutzername aller Rezepte, die in den letzten zwei Jahren *3 oder mehr Bewertungen* und in diesem Zeitraum im Schnitt *mind. 4 Sterne* erhalten haben. Ausgabe aufsteigend sortiert nach Titeln.
3. Eine nach Durchschnittsbewertung absteigend sortierte Übersicht der Rezepte mit Titel, Benutzername, Durchschnittsbewertung und der Anzahl der Bewertungen.
4. Nach Bezeichnung aufsteigende Liste aller Zutaten, welche in keinem Rezept je verwendet wurden.
5. Liste aller Rezepte in denen eine *Zutat in mehreren Schritten* vorkommt. Titel und Gesamtzahl der Schritte, aufsteigend sortiert nach Rezept-ID.
6. Angabe der Rezepte mit Titel und Benutzername, für die *mehr als 200 Milli-Liter Milch* benötigt wird. Gehen Sie davon aus, dass Milch immer in ml angegeben wird. Sortierung aufsteigend nach Rezept-ID.
7. Liste aller Benutzernamen, die *mehr als drei* Rezepte mit *vier oder mehr Sternen* bewertet haben. Sortierung aufsteigend nach Benutzer-ID.

8. Id und Bezeichnung aller Zutaten, die **nicht** im Rezept „Pfannkuchen mit Sahne“ vorkommen. Sortierung aufsteigend nach Zutaten-ID.
9. Geben Sie die Zutaten-ID, Bezeichnung und Gesamtmenge der benötigten Zutaten des Rezepts Nutellapizza aus. Die Mengen sollen von *Liter*, *Teelöffel* und *Esslöffel* in *Milli-Liter* und *Kilogramm* in *Gramm* umgerechnet werden. 1tl = 5ml, 1el = 15ml. Sortieren Sie aufsteigend nach Zutaten-ID.
10. Geben Sie die Anzahl von Benutzer-Paaren an die eine gleiche E-Mail Adresse haben. Gibt es z.B. 4 Benutze A,B,C und D mit der gleichen E-Mail Adresse, so gibt es die Paare (A,B), (A,C), (A,D), (B,C), (B,D) und (C,D), d.h. 6 Paare. Gibt es keine solche Benutzer, dann geben Sie 0 aus.
11. Geben Sie eine Liste mit Bezeichnung, Anzahl aus die für jede Zutat ausgibt, in wie vielen Rezepten diese Zutat verwendet wird. Aufsteigend sortiert nach Zutaten-ID.
12. Geben Sie die Gesamtdauer aller Rezepte von Benutzer „chef123“ im Format Time aus.
13. Berechnen Sie die durchschnittliche Anzahl der Rezeptschritte.
14. Berechnen Sie für jedes Paar an Rezepten die Anzahl gleicher Zutaten. Ausgabe wie folgt: Rezept-ID 1, Rezept-ID 2, Anzahl gleicher Zutaten. Sortierung aufsteigend nach ID1 und ID2.
15. Berechnen Sie die Paare an Rezepten mit den meisten gleichen Zutaten. Ausgabe wie Aufgabe 14.
16. Bestimmen Sie die Zutat(en), die in den meisten Rezepten verwendet werden. Ausgabe mit Zutat-ID und Bezeichnung. Sortieren Sie die Ausgabe nach Zutaten-ID aufsteigend.