



Aufgabenblatt 8: SQL I (bis Donnerstag, 18.12.2014)

Hinweis: um die *Studienleistung* für diese Vorlesung zu absolvieren, benötigen Sie 50% der Hausaufgabenpunkte aus diesen Übungen. Um das *Modul RDBI* erfolgreich abzuschließen, müssen Sie die Klausur am Ende des Semesters bestehen **und** die Studienleistung erfolgreich absolvieren. Die Übungen müssen stets **donnerstags vor der Vorlesung** abgegeben werden. Dies kann über unseren **Briefkasten** (Informatikzentrum zweiter Stock, gegenüber vom Fahrstuhl) oder zum **Start der Vorlesung** geschehen. Bitte versehen Sie ihre Abgaben stets mit ihrer **Matrikelnummer** und mit der **Nummer ihrer Übungsgruppe**. Die Lösungen dürfen auf Deutsch oder Englisch eingereicht werden. Benutzen Sie für die Lösungen stets ihre **eigenen Worte**.

Aufgabe 8.1 – Verständnisfragen (10 Punkte)

- Warum wird in SQL eine dreiwertige Logik benötigt? Was für Auswirkungen hat diese auf die booleschen Operatoren \wedge , \vee und \neg ? (2 Punkte)
- Warum wurde bei der Implementierung der meisten RDBMS darauf verzichtet das theoretisch korrektere Mengenmodell (ohne Duplikate) zu betrachten und stattdessen ein Multimengenmodell implementiert? (1 Punkt)
- Erklären Sie den Unterschied zwischen der WHERE-Klausel und der HAVING-Klausel. (2 Punkte)
- Erhöht sich durch die HAVING-Klausel die Mächtigkeit von SQL? Erklären Sie ihre Antwort. (2 Punkte)
- Was ist eine korrelierte bzw. unkorrelierte Subquery? Welche ist allgemein performanter und warum? (2 Punkte)
- Was ist der semantische Unterschied zwischen Ausdruck 1 und Ausdruck 2 in folgendem Beispiel? (1 Punkt)

Ausdruck 1: `SELECT COUNT(email) FROM Kunde`

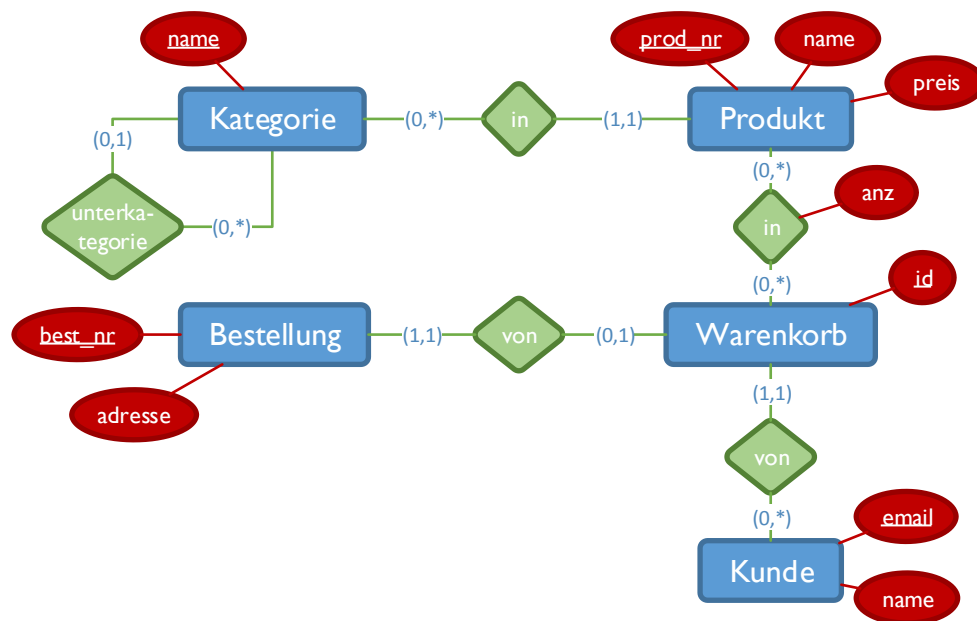
Ausdruck 2: `SELECT COUNT(name) FROM Kunde`

Aufgabe 8.2 – SQL Queries (13 Punkte)

Geben Sie **Ausdrücke in SQL** an, die die folgenden Ergebnisse zurückgeben. Die Anfragen basieren auf dem Schema in Anhang A.

- Alle Produkte (Prodnr, Name), der Kategorie „Konsolen“, die 200€-250€ kosten (1 Punkt)
- Alle Kategorien (Name), die nie bei einem Produkt verwendet wurden (2 Punkte)
- Die Anzahl Produkte, die den Teilstring „Kamera“ im Namen enthalten (2 Punkte)
- Kunden (Email), die mehr als 10 Bestellungen geordert haben (2 Punkte)
- Kunden, die dasselbe Produkt mehrmals in verschiedenen Bestellungen gekauft haben. Es sollen die jeweiligen Kunden (Email, Name) und die von diesen Kunden mehrfach gekauften Produkte (Produktnummern) ausgegeben werden (3 Punkte)
Beispieltupel: `<“barthel@ifis.cs.tu-bs.de“, “Simon Barthel“, 15>` (\Rightarrow Simon Barthel hat den Artikel 15 in mehr als einer Bestellung gekauft)
- Warenkörbe (id), bei denen alle Produkte (mindestens eins) aus derselben Kategorie sind (3 Punkte)

Anhang A:



- Kategorien können über den Relationstyp „unterkategorie“ in Hierarchien angeordnet werden.
- Der Preis des Produkts wird in Eurocent als Ganzzahl angegeben.
- Bestellungen verweisen lediglich auf einen Warenkorb. Sobald ein Warenkorb „gekauft wird“, wird eine Bestellung mit einem Verweis auf den Warenkorb sowie ein neuer leerer Warenkorb für den entsprechenden Kunden angelegt.

Relationales Datenbankschema:

Kunde (email, name)

Warenkorb (id, kunde → Kunde)

Produkt (prodnr, name, preis, kategorie → Kategorie)

Bestellung (bestnr, adresse, warenkorb → Warenkorb)

produkt_in_warenkorb (produkt → Produkt, warenkorb → Warenkorb, anz)

Kategorie (name, parent → Kategorie)