

Programmierung von Systemen

Blatt 2

Institut für Softwaretechnik und Programmiersprachen | Sommersemester 2024
Matthias Tichy, Raphael Straub und Florian Sihler

Abgabe (Moodle) bis
12. Mai 2024

Erste Schritte in SQL

2 ★ 9 ■ 3 ●

- Manipulationen via Relationenalgebra Ausdrücke definieren
- SQL Ausdrücke definieren und mit Relationenalgebra vergleichen

Willkommen zum zweiten Übungsblatt! Diese Woche beschäftigen wir uns mit SQL!

Aufgabe 1: Relationenalgebra

Auf dem Letzten Übungsblatt ging es darum ein ER-Diagramm zu modellieren und Relationenalgebra Ausdrücke zu definieren. Dabei haben wir folgendes Relationenschema erzeugt:

Teile				Lieferant			Bestellung			
TNR	Name	Preis	Bestand	LiefNr	LiefName	LiefStadt	TNR	LiefNr	KdName	KdStadt

Wir möchten die Attribute KdName und KdStadt in ein extra Relationenschema Kunde auslagern.

a) Änderungen am Schema ★

Wie müssen hierfür die obigen Schemata abgeändert werden, so dass man trotzdem noch sinnvoll die Anfragen des letzten Übungsblattes in Relationenalgebra ausdrücken kann?

b) Eine weitere Anfragen ★

Drücken Sie folgende Anfrage aus:

*Gib die Namen aller Kunden aus, welche entweder nichts, oder **nur** Teile mit einem Preis von weniger als 100€ bestellt haben.*

Aufgabe 2: SQL I



In dieser Aufgabe sollen SQL-Anfragen an die aus der Vorlesung bekannte LegoTrailer-Datenbank gestellt werden. *Beachten Sie dazu folgende Hinweise:*

1. **Datenbank:** Als Datenbanksystem verwenden wir HSQL (<http://hsqldb.org>). Im Moodle wird ein Archiv [LegoDB.zip](#) bereitgestellt, welche die Datenbank und alle zur Verwendung notwendigen Dateien beinhaltet.
2. **Starten der Datenbank:** In dem Archiv befinden sich zwei Skriptdateien für Windows und Linux (`rundb.bat` bzw. `rundb.sh`), welche einfach direkt in dem Verzeichnis ausgeführt werden können und eine Benutzeroberfläche zur Interaktion mit der Datenbank öffnen (Hierzu sollten Sie, falls nicht schon geschehen, Java installiert haben! Achten Sie außerdem auf benötigte Ausführungsrechte). Der Befehl zum Starten kann natürlich auch direkt auf der Kommandozeile abgesetzt werden:

```
java -classpath ./hsqldb.jar org.hsqldb.util.DatabaseManagerSwing
--driver org.hsqldb.jdbcDriver --url jdbc:hsqldb:file:lego.db
```

3. Verwendung: SQL-Statements lassen sich im oberen Inputfeld eingeben und mittels *Execute SQL* ausführen.

Geben Sie die entsprechenden SQL-Statements für folgende Anfragen an (ohne die resultierende Ausgabe). Geben Sie ebenfalls an, welche Konzepte aus der Relationenalgebra für die jeweilige Anfrage in SQL angewendet werden und wo die Unterschiede beziehungsweise Limitierungen liegen:

1. Alle Lieferanten, die nicht aus München kommen.
2. Farbe, TeileNr und Preis aus der Preisliste-Tabelle. Es sollen nur die Einträge ausgegeben werden, bei denen der Preis zwischen 300 und 430 Euro liegt.
3. Die Bestellungen, die vor dem 15.02.2013 oder nach dem 23.02.2013 getätigt wurden.
4. Alle Informationen zu Kunden, die aus einer Stadt kommen, welche mit einem 'S' beginnt.
5. Den Preis des jeweiligen Teils in der Teile-Tabelle mit ihrer Nummer und Farbe.
6. Das Gehalt aller Mitarbeiter mit Funktion „Montage II“ aus Ulm; aufsteigend geordnet.
7. Alle Informationen zu Aufträgen sowie das berechnete „Skontodatum“. Das Skontodatum liegt 3 Monate nach Auftragsdatum.
8. Alle Aufträge mit Nummer und Datum, die an einem geraden Tagesdatum getätigt wurden (z.B. 02.xx, 04.xx, 06.xx, ...).
9. Geben Sie eine Bestandsbewertung für die vorrätigen Teile in der Teilerrelation. Geben Sie dazu die Attribute zu den Teilen aus, sowie eine Bewertung nach folgendem Schema: Wenn der Bestand größer als der Mindestbestand ist, geben Sie „Überbestand“ aus, wenn beide gleich sind „Mindestbestand“, wenn zu wenig Teile da sind „Mangel“.

Aufgabe 3: SQL II



Geben Sie die entsprechenden SQL-Statements für folgende Anfragen an (ohne die resultierende Ausgabe):

1. Alle Teile eines Lieferanten, der aus Ulm kommt und eine Bewertung besser als 0 erhalten hat. Gib jeweils den Lieferantennamen und seine Bewertung mit aus.
2. Alle Auftragspositionen, die vom Kunden *Dehling* in einem Auftrag geordert wurden. Gib jeweils ihre Anzahl, Farbe, Auftrags- und Teilenummer mit aus.
3. Alle Kunden, die mehr als eine Bestellung aufgegeben haben. (Tipp: Sieh dir das HAVING Schlüsselwort an.)
4. Das durchschnittliche Gehalt aller Mitarbeiter aus der Abteilung *Lager*.
5. Den Mitarbeiter mit dem höchsten Gehalt. Gib jeweils alle seine Attribute mit aus. Mitarbeiter, die Manager einer Abteilung sind, sind hierbei ausgeschlossen.
6. Die Kosten eines jeden Auftrages mit Betrag und Auftragsnummer, aufsteigend nach Preis geordnet.

7. Gib alle Lieferanten namentlich aus, die nicht gleichzeitig Kunden bei der Firma sind. Verwende einmal eine **mengenbasierte Abfrage**(EXCEPT/UNITON/INTERSECT) und einmal eine frei wählbare **andere Anfrage**.
8. Alle Lieferantennummern, die jeweils die Teile K14, K16 und K18 liefern. (Tipp: Versuche eine mengenbasierte Abfrage zu verwenden.)
9. Bestimme alle Teile mit Nummer und Farbe, die zur Zeit in keiner Auftragsposition auftauchen. Verwende dazu den Outer-Join.

Aufgabe 4: SQL III



Für diese Aufgabe verwenden wir wieder die bekannte LegoTrailer-Datenbank. Bestimmen Sie die nötigen Statements in SQL um folgende Datenmanipulationen auf der Datenbank durchzuführen:

1. Der neue Lieferant „Elsholz und Spann GmbH“ hat sich Ulm niedergelassen. Fügen Sie ihn mit Lieferantenummer 28 und Bewertung 2 in die Lieferantentabelle ein.
2. Alle Lieferanten, die Teil K18 in beliebiger Farbe liefern, können nun auch Teil TL134 liefern in genau derselben Farbe, aber zum dreifachen Preis wie K18.
3. Löschen Sie alle Lieferanten aus der Liefert-Relation, die nur ein Teil liefern können.
4. Die Gehälter aller Manager sollen um 20% gekürzt werden.
5. Der Kunde Babel hat seine Meinung zu seinen Bestellungen geändert. Er möchte, dass alle Teile, die in seinen Bestellungen blau sind (Farbcode 2), nun doch in rot (Farbcode 1) geliefert werden sollen.