



Übung zur Vorlesung *Grundlagen: Datenbanken* im WS19/20

Christoph Anneser, Moritz Sichert, Lukas Vogel (gdb@in.tum.de)

<https://db.in.tum.de/teaching/ws1920/grundlagen/>

Blatt Nr. Z1

Dieses **Zusatzblatt** wird nicht in der Übung besprochen!

Tool zum Üben von SQL-Anfragen: <https://hyper-db.com/interface.html>.

Onlineshop-Schema: <http://db.in.tum.de/teaching/ws1920/grundlagen/schema.pdf>

Zusatzaufgabe 1

Diese Aufgabe ist die vierte in einer Reihe von Aufgaben, in denen Sie lernen werden, einen gegebenen Sachverhalt zu analysieren, geeignete Modelle zu entwerfen, diese in ein Datenbankschema zu überführen, das Schema in einer (lokalen) Datenbank aufzusetzen und mit Daten zu füllen. Sie werden ebenfalls lernen, wie Sie die Datenbank für bestimmte Anfragen optimieren können.

In dieser Aufgabe importieren wir zufallsgenerierte Daten für den Onlineshop in unsere lokal installierte Postgres-Datenbank. Laden Sie dafür den auf Moodle bereitgestellten zip-Ordner herunter und entpacken Sie ihn. Importieren Sie dann die Daten mit folgendem SQL-File: <https://db.in.tum.de/teaching/ws1920/grundlagen/shop.sql> (Sie müssen eventuell noch die Pfade anpassen).

Beachten Sie außerdem, dass ein Teil der Daten (Kundenprofile, Namen, Adressen, etc.) mit dem Python-Paket Faker¹ zufällig generiert wurden und sich daher *nicht* auf real existierende Personen oder Unternehmen bezieht. Es handelt sich hierbei um *rein fiktive* Personen, Unternehmen und Adressen.

```
$ psql shop < shop.sql
```

Zusatzaufgabe 2

Formulieren Sie die folgenden Anfragen auf dem Schema des Onlineshops in SQL. Aufgaben, die beispielsweise Schreibrechte auf der Datenbank erfordern, sind durch * kenntlich gemacht und sollten Sie in Ihrer lokalen Datenbank testen, da sie im Webinterface nicht ausgeführt werden können.

- Finden Sie für jedes *Land* den *Kunden*, auf den der höchste Umsatz entfallen ist.
- Finden Sie für jedes *Land* den *Kunden*, auf den der zweithöchste Umsatz entfallen ist. Der zweithöchste Umsatz ist definiert als der nächstkleinere Umsatz verglichen zum höchsten im jeweiligen *Land* angefallenen Umsatz. Geben Sie eine Anfrage *mit* korrelierter Unteranfrage und eine Anfrage *ohne* korrelierter Unteranfrage an.
- Ermitteln Sie die Exporte der einzelnen *Länder* und schlüsseln Sie diese nach Jahr und Importland auf.
- Geben Sie eine Übersicht, wie viele *Kunden* *n* *Bestellungen* getätigt haben und schlüsseln Sie diese nach *Kontinent* auf.

* Exportieren Sie das Ergebnis Ihrer Anfrage als CSV-Datei. Welche Verteilung vermuten Sie hinter den getätigten Bestellungen der Kunden?

¹<https://github.com/joke2k/faker/>

- (e) Finden Sie die *Kunden*, die von mindesten fünf unterschiedlichen *Lieferanten*, die den Sitz im gleichen *Land* haben, *Artikel* erhalten haben, und die insgesamt einen Umsatz von 1000 € oder mehr erzielt haben (unabhängig vom Lieferanten).
- (f) Finden Sie die *Kunden*, die im Jahr 2018 *Artikel* von mindestens zwei *Lieferanten* *a* und *b*, die ihren Sitz in einem der *Länder* {GERMANY, BRAZIL, JAPAN, ROMANIA, UNITED KINGDOM, CANADA} haben, erhielten. Beachten Sie außerdem, dass *a* und *b* ihren Sitz in unterschiedlichen *Ländern* haben müssen.
- (g) Finden Sie für jeden *Artikel* und jedes *Land* den *Lieferanten*, der den günstigsten Lieferpreis verlangt.
- (h) Finden Sie für jeden *Artikel* und jedes *Land* die Preisdifferenz des günstigsten und teuersten Angebots. Geben Sie auch den prozentualen Aufschlag vom günstigeren zum teureren Angebot mit zwei Nachkommastellen an.
- (i) Finden Sie die *Kunden*, die am meisten Geld hätten sparen können, wenn sie beim *Lieferanten* gekauft hätten, der den entsprechenden *Artikel* im Land des *Kunden* am günstigsten anbietet.

Wie die obige Anfrage (i) zeigt, treten negative Einsparungen auf. Dies hat den Grund, dass laut Datenbank *Kunden* von *Lieferanten* beliefert werden, die laut Relation *liefertNach* nicht möglich wären. Diese fehlerhaften Daten werden wir im Folgenden korrigieren.

- (j) * Finden Sie die *Bestellpositionen*, die laut der Relation *liefertNach* gar nicht möglich wären.
 - (1) * Geben Sie **eine** SQL-Anweisung an, um eine neue Tabelle *Bestellpositionen-Korrigiert* zu erstellen, die das gleiche Schema und die gleichen Einträge wie die Tabelle *Bestellpositionen* enthält.
 - (2) * Updaten Sie nun die Tabelle *BestellpositionenKorrigiert*: Jede Position, bei der der *Lieferant* nicht das Land des *Kunden* beliefert, soll durch den günstigsten das Land des Kunden beliefernden Lieferanten (günstig in Bezug auf den in der *Bestellposition* referenzierten *Artikel*) ersetzt werden.
 - (3) * Überprüfen Sie, ob Ihre Anweisungen zum gewünschten Ergebnis geführt haben, indem Sie bei obiger Anfrage überprüfen, dass Kunden keine negativen Einsparungen mehr erhalten würden.

Nachdem wir die Daten nun bereinigt haben, wollen wir sicher gehen, dass nicht erneut fehlerhafte *Bestellpositionen* eingefügt werden können.

- (k) Erörtern Sie unterschiedliche Möglichkeiten, wie man die Relation *Bestellpositionen* modifizieren könnte, damit solche fehlerhaften Daten nicht eingefügt werden können. Gehen Sie dabei insbesondere auf *Fremdschlüssel* und **check**-constraints ein.
- (l) Ermitteln Sie die Entwicklung der Marktanteile für *Länder* *l* innerhalb ihres *Kontinents* *k* für alle verschiedenen Artikeltypen an. Die Entwicklung soll auf Jahresbasis analysiert werden und auf den Preisen der *Bestellpositionen* beruhen. Dabei sollen nur die bestellten Waren berücksichtigt werden, die von *Lieferanten* mit Sitz in *k* geliefert wurden und deren *Kunde* ebenfalls in *k* beheimatet ist. Wenn der umgesetzte Wert im vorherigen Jahr 0 sein sollte, geben sie NULL statt des prozentualen Anstiegs aus.