

Datenbanksysteme (VL 4.0, 181.186)

Laborübungsteil, WS 2009/2010

Beispiel 2

Ein Händler, der sein Sortiment an Waren über das Internet anbietet, möchte seine Geschäftsabläufe mittels einer Datenbank erfassen und vereinfachen. Diese Datenbank soll folgende Informationen enthalten:

- Jeder Standort des Unternehmens wird durch eine eindeutige Nummer („sid“) identifiziert, dazu werden noch die Bezeichnung („bezeichnung“) und die Adresse („plz“, „ort“, „strasse“, „nummer“) gespeichert. Außerdem wird vermerkt, welche Angestellte an den jeweiligen Standorten arbeiten. Dabei kann man davon ausgehen, dass ein Angestellter nur an genau einem Standort arbeitet. Jeder Standort besitzt außerdem genau einen Angestellten als Leiter. Von den Angestellten werden folgende Daten erfasst: die eindeutige Sozialversicherungsnummer („svnr“), der Name („vname“, „nname“), das Geburtsdatum („gebdatum“), die Adresse sowie das Gehalt („gehalt“). Bei Managern werden außerdem der jährliche Bonus („bonus“) abgespeichert, der diesen zusteht. Die Manager des Unternehmens sind nicht zwangsweise die Leiter der Standorte, die Möglichkeit besteht jedoch.
- Das Sortiment des Händlers wird ebenfalls mit Hilfe der Datenbank verwaltet. Jeder Artikel wird dabei von seiner Europäischen Artikelnummer („ean“) identifiziert, außerdem werden die Bezeichnung („bezeichnung“) sowie der Typ des Artikels („typ“) eingetragen. Manche dieser Artikel werden zu Serien zusammengefasst. Dazu werden jeweils die unmittelbar aufeinanderfolgenden Serienteile gespeichert („vorgaenger“, „nachfolger“), wobei man davon ausgehen darf, dass jeder Serienteil höchstens einen unmittelbaren Nachfolger besitzt.
- Für jeden Artikel aus dem Sortiment müssen tagesaktuelle Preise gespeichert werden. Dabei ist zu berücksichtigen, dass die Preise in den einzelnen Ländern variieren können. Ein Preis gilt also für einen konkreten Artikel in einem konkreten Land an einem fixen Datum („datum“). Zusätzlich wird der Steuersatz („steuer“) in Prozent (Werte zwischen 0 und 100) vermerkt. Um die einzelnen Länder zu erfassen, werden der Ländercode („code“), der Name des jeweiligen Landes („name“) sowie der Kontinent („kontinent“) eingetragen, wobei natürlich nur die existierenden Kontinente (Nordamerika, Südamerika, Europa, Afrika, Asien und Australien) eingetragen werden dürfen.
- Mittels der Datenbank werden auch die Kunden des Versandhändlers erfasst. Diese erhalten eine Kundennummer („knr“), über die sie eindeutig identifiziert werden können, außerdem werden Name („vname“, „nname“) und Adresse abgespeichert. Die Kunden machen laufend Bestellungen aus dem Sortiment des Händlers. Jede Bestellung erhält eine fortlaufende Nummer („bid“), außerdem wird für jede Bestellung der Zeitpunkt des Bestellabschlusses („zeitpunkt“) gespeichert. Jede Bestellung enthält mindestens einen Artikel aus dem Sortiment, wobei für jeden bestellten Artikel die jeweilige Anzahl („anzahl“) eingetragen wird. Dabei ist zu beachten, dass auch gespeichert wird, zu welchem Tagespreis der jeweilige Artikel bestellt wird. Weiters wird gespeichert, ob der Kunde die Bestellung bezahlt hat, und gegebenenfalls, wann („datum“).

Aufgabenstellung

1. Erstellen Sie für die oben beschriebene Datenbank ein Entity-Relationship-Diagramm. Benützen Sie dazu die **(min, max)-Notation** wie im Lehrbuch von Kemper/Eickler bzw. in der VU Datenmodellierung beschrieben. Andere Notationen werden nicht akzeptiert. Erstellen Sie das ER-Diagramm mit einem Grafikprogramm (z. B. Dia, Visio).

Bedenken Sie dabei, dass an drei Stellen (für Standorte sowie bei Angestellten und Kunden) Adressen gespeichert werden müssen. Lösen Sie dieses Problem, indem Sie einen eigenen Entitytypen für Adressen erstellen, wobei jede Adresse von einem künstlichen Schlüssel („aid“) identifiziert wird. Welche Vor- und Nachteile erwachsen aus dieser Vorgehensweise?

- Leiten Sie aus dem ER-Diagramm die Relationen der Datenbank in 3. Normalform so ab, dass sie verbundtreu und abhängigkeitstreu sind. Halten Sie diese Relationen schriftlich fest und machen Sie dabei PRIMARY und FOREIGN KEYS eindeutig kenntlich. Folgende Notation wird empfohlen:

RelX (attr1, attr2, attr3, attr4: RelY.attrZ)

bedeutet, dass die Datenbank eine Relation RelX enthält. Diese Relation hat 4 Attribute attr1, attr2, attr3 und attr4. Der Primary Key dieser Relation besteht aus den Attributen attr1 und attr2. Das Attribut attr4 ist ein Foreign Key auf das Attribut attrZ in der Relation RelY.

Das Entity-Relationship-Diagramm sowie das Relationenmodell sollen gemeinsam in der Datei `entwurf.pdf` gespeichert werden. Für die Erstellung des PDFs kann beispielsweise das Tool [doc2pdf](#) verwendet werden.

- Erstellen Sie eine Datei `create.sql`, in welcher die nötigen CREATE-Befehle gespeichert werden, um die Relationen mittels SQL zu realisieren. Dabei sind folgende Punkte zu beachten:
 - Die Datenbank soll keine NULL-Werte enthalten.
 - Stellen Sie sicher, dass trotz des künstlichen Schlüssels in der Relation „Adressen“ niemals gleichlautende Adresseinträge mehrfach vorkommen können. Stellen Sie dazu sicher, dass die Kombination aus PLZ, Ort, Strasse und Nummer eindeutig ist.
 - Verwenden Sie zur Erfassung des Bestellzeitpunktes den Datentyp `TIMESTAMP`. Verwenden Sie als Default-Wert das aktuelle Systemdatum und die aktuelle Systemzeit. Weitere Informationen dazu finden Sie in der [PostgreSQL 8.4 Dokumentation](#).
 - Realisieren Sie folgende Attribute mit fortlaufenden Nummern mit Hilfe von Sequences: „knr“ von Kunde, wobei die Kundennummern mit 1.000.001 beginnen und anschließend in 10er-Schritten fortgeführt werden sollen (d. h. 1.000.001, 1.000.011, 1.000.021, ...), sowie „bid“ von Bestellungen, wobei diese bei 1.000.000.001 beginnen sollen.
 - Sollten zwischen 2 Tabellen zyklische FOREIGN KEY Beziehungen existieren, so achten Sie darauf, dass eine Überprüfung dieser FOREIGN KEYs erst zum Zeitpunkt eines COMMIT stattfindet.
- Erstellen Sie eine weitere Datei `insert.sql`, welche die INSERT-Befehle für die Testdaten der in Punkt 3 erstellten Tabellen enthält. Jede Tabelle soll zumindest 6 Zeilen enthalten. Sie dürfen die Wahl der Namen, Bezeichnungen etc. so einfach wie möglich gestalten, d. h. Sie müssen nicht „real existierende“ Objekte, Personen, Artikel etc. wählen. Stattdessen können Sie ruhig 'Adresse1', 'Adresse2', 'Person1', 'Person2' etc. verwenden.
- Erstellen Sie eine Datei `drop.sql`, welche die nötigen DROP-Befehle enthält, um alle in Punkt 3 erzeugten Datenbankobjekte wieder zu löschen.

Für die Abgabe bereitzustellen

- Bringen Sie bitte Ihren Studentenausweis zur Abgabe mit. Eine Abgabe ohne Ausweis ist nicht möglich.
- Stellen Sie sicher, dass die SQL-Befehle (laut Punkten 3-5) unter PostgreSQL 8.4 auf dem Server **bordo.dbai.tuwien.ac.at** fehlerlos laufen. Sie müssen in der Lage sein, die von Ihnen bereitgestellten SQL-Dateien im Rahmen des Abgabegesprächs ablaufen lassen.

- Stellen Sie in Ihrem Abgabeverzeichnis eine Listing-Datei mit dem Namen `listing.txt` bereit, die Sie beim Test der SQL-Daten erzeugt haben. Diese Datei soll alle Informationen beinhalten, die beim Ablauf der Dateien `create.sql`, `insert.sql` und `drop.sql` erzeugt werden. Beachten Sie dazu bitte die Hinweise zur Benützung von **psql** am Ende dieses Dokuments.
- Zusammenfassung: In Summe sind also folgende **5 Dateien** zu erstellen:
 - `entwurf.pdf`
 - `create.sql`
 - `insert.sql`
 - `drop.sql`
 - `listing.txt`
- Die Beispiele müssen bis zum Abgabetermin am 09.11.2009 um Mitternacht auf unserem Server (`bordo.dbai.tuwien.ac.at`) im Unterverzeichnis `beispiel2` verfügbar sein (die Dateien werden automatisch abgesammelt und den Tutorinnen und Tutoren zur Verfügung gestellt). Es sind keine Ausdrücke erforderlich.

Bewertung

Für das Übungsbeispiel 2 werden **maximal 10 Punkte** vergeben. Im Rahmen des Abgabegesprächs wird nicht nur die Korrektheit der Lösungen, sondern auch (und vor allem) das **Verständnis** überprüft. Für die einzelnen Aufgaben erhalten Sie die maximal möglichen Punkte nur dann, wenn die Lösung richtig ist **und** wenn Sie in der Lage sind, diese Lösung auch entsprechend zu erklären.

Die Verteilung der Punkte erfolgt nach folgendem Schlüssel:

- Datei `entwurf.pdf`: max. 5 Punkte (max. 3 Punkte für das ER-Diagramm; max. 2 Punkte für das Relationenmodell)
- Datei `create.sql`: max. 2 Punkte
- Datei `insert.sql`: max. 2 Punkte
- Datei `drop.sql`: max. 1 Punkt
- Datei `listing.txt`: Vorhandensein für das Erreichen der vollen Punktezahl notwendig.

Hinweise zur Verwendung von psql

Folgende Befehle können für Ihre Arbeit mit der interaktiven SQL-Shell `psql` von PostgreSQL 8.4 hilfreich sein:

- `\?`: Listet alle `psql`-internen Befehle samt Erklärung auf.
- `\i <dateiname>`: Führt das Skript `<dateiname>` aus. Beispiel: `\i create.sql`
- `\o <dateiname>`: Lenkt die Ausgabe in eine Datei mit dem Namen `<dateiname>` um. Lässt man den Parameter `<dateiname>` weg, so wird dieses Verhalten wieder abgestellt. Beispiel: `\o listing.txt`