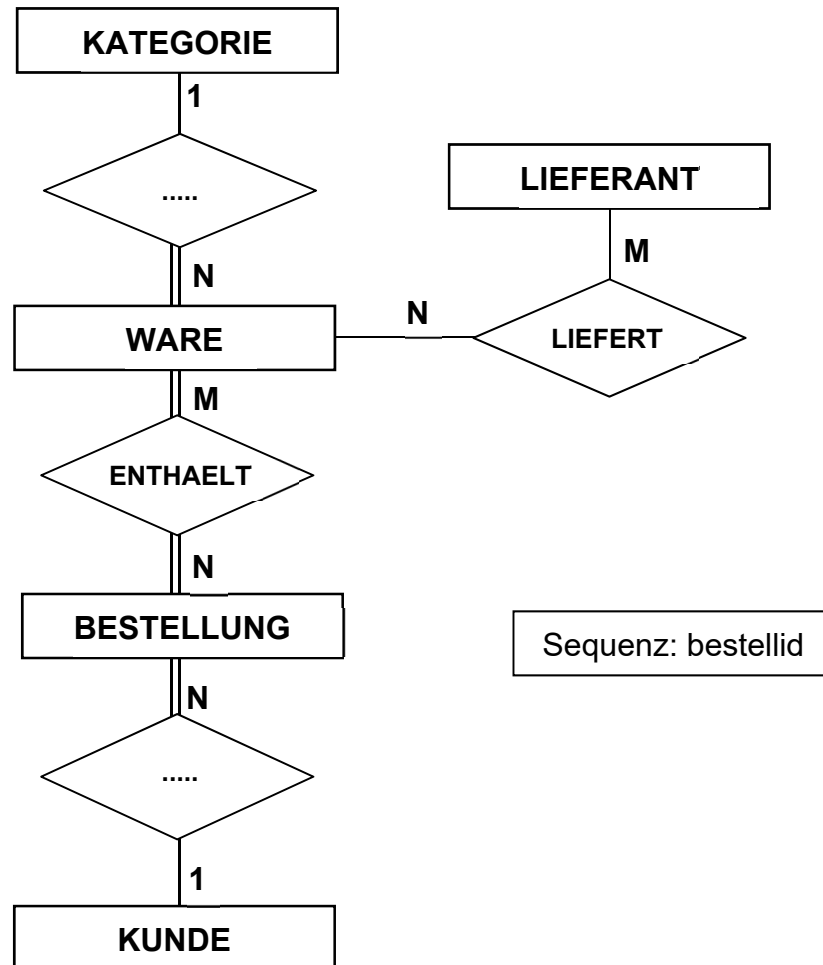


Praktikum Datenbanken

Aufgabe 4: JDBC: Online-Supermarkt

Vorgegeben ist eine Datenbank über Online-Supermärkte mit den Waren, die sie verkaufen und ihren Lieferanten und Kunden:



Zugehöriges relationales Datenbankschema:

KATEGORIE (katID, kname)

LIEFERANT (liefID, lname, standort)

WARE (warenID, bezeichnung, preis, vorrat, *katID*)

KUNDE (kundeID, kname, wohnort)

BESTELLUNG (bestID, datum, status, *kundeID*)

LIEFERT (liefID, warenID)

ENTHAELT (bestID, warenID, anzahl)

Primärschlüssel

Fremdschlüssel

Schreiben Sie für diese Datenbank folgende Methoden unter JDBC:

- **Ausgabe aller Waren** mit Bezeichnung, Preis, Vorrat und dem Namen der Kategorie, zu der sie gehören (*alleWaren()*). Diese Methode soll ein **einfaches** JDBC-Statement verwenden.
- **Anlegen einer Bestellung** (*bestellen()*): Ein Kunde kann eine Bestellung von mehreren Waren tätigen, die Waren können in beliebiger Anzahl bestellt werden (Beispiel: Kunde *Pythagoras* bestellt 10 Zelte und 5 Anoraks). Kundenname, Warenname und Anzahl der bestellten Waren sind einzulesen (z.B. über *readLine()*).

Für eine Bestellung wird ein neuer Eintrag in die Tabelle *Bestellung* erzeugt, mit Bestell-ID (fortlaufende Sequenznummer), Kunden-ID, aktuellem Datum (*CURRENT_DATE*) und Status (0 für noch unerledigte Bestellung). Die ID des Kunden enthält man über seinen eingelesenen Namen.

Zusätzlich wird für den Kunden für jede bestellte Ware ein neuer Eintrag in die Tabelle *enthaelt* erzeugt, mit der betreffenden *Bestell_ID*, der ID der bestellten Ware und der Anzahl der bestellten Exemplare.

Für das Eintragen der Bestellungs-Identifikation in die Tabellen *Bestellung* und *enthaelt* wird eine fortlaufende Nummer verwendet, die über eine sog. *SEQUENCE* implementiert wird, siehe Erläuterungen unten.

- **Ausgabe aller Bestellungen mehrerer nacheinander einzulesender Kunden** jeweils mit Bestell-ID, Datum, Status und Namen des bestellenden Kunden (*alleBestellungen()*). Diese Methode soll ein **prepared** JDBC-Statement verwenden, das nach Erzeugung mehrfach verwendet wird.

Testen Sie Ihre Funktionen an der auf Moodle zur Verfügung gestellten SupermarktDatenbank. Dabei legen Sie für zwei Kunden jeweils eine Bestellung an. Die Bestellung des ersten Kunden soll zwei, die des zweiten Kunden drei Waren enthalten.

Hinweise:

- **Sequenzen: CREATE SEQUENCE**

Außer den Tabellen ist für diese Datenbank eine Sequenz definiert: *CREATE SEQUENCE bestellid*; Eine Sequenz ist ein Zähler (Startwert 1 nach dem ersten Aufruf von *nextval()* s.u. und Inkrement 1, wenn nicht durch Parameter anders initialisiert), der fortlaufende Nummern generiert, um eindeutige Werte in Datenbanken zu erzeugen (z.B. für Primärschlüssel). Diese Sequenz benutzen Sie bei der Erzeugung der Primärschlüsselwerte für neue Bestellungen, also bei Einfügen einer neuen Bestellung in die Relation *Bestellung*:

insert into bestellung values (nextval('bestellid'), ...);

Durch die Verwendung von *nextval()* wird immer der nächste Wert der Zählern berechnet, hier wird also der nächste Wert des Zählers in die Tabelle *bestellung* eingefügt (und zusätzlich der Zähler inkrementiert).

In der Relation *enthaelt*, in die die einzelnen Posten einer Bestellung eingetragen werden, kann man dann im Fremdschlüssel Bezug auf den Wert dieser Sequenz nehmen:

```
insert into enthaelt values (currval('bestellid'), ...);
```

Durch die Verwendung von *currval()* wird immer der aktuelle Wert des Zählers berechnet, hier wird also der aktuelle Wert des Zählers in die Tabelle *enthaelt* eingefügt.

- TREIBERINFORMATION !

Um Ihre JDBC-Anwendung laufen zu lassen, benötigen Sie noch den richtigen JDBC-Treiber. Er findet sich im Verzeichnis `/opt/CLASSPATH`.

Bei Verwendung einer Linux-Shell:

Setzen Sie die CLASSPATH-Variable entsprechend:

```
export CLASSPATH=$CLASSPATH:/opt/CLASSPATH/postgresql-9.1-903.jdbc4.jar
```

Überprüfen von CLASSPATH: **echo \$CLASSPATH**

Außerdem muss im CLASSPATH vorhanden sein: `.` (aktuelles Verzeichnis).

Bei Verwendung der Entwicklungsumgebung Eclipse:

Project -> Properties -> im aufscheinenden Properties Fenster: Java Build Path ->

Libraries -> Classpath -> Add External Jars ... ->

Einfügen von `/opt/CLASSPATH/postgresql-9.1-903.jdbc4.jar`

Rahmenprogramm:

Ein Rahmenprogramm, das Sie als Grundlage verwenden können, steht auf Moodle. Dateiname, Datenbank-Name und Benutzer-Name sind anzupassen.

Materialien:

Diese Aufgabenstellung

CREATE TABLEs: Datenbankdefinition

INSERTs: Datenbankinhalte

Rahmenprogramm

Vorführung / Abgabe:

Am Ende des jeweiligen Praktikumstermins