

# Verteilte Datenbanken

Ari Ayvazyan, Martin Haidn

10.10.2014

## Inhaltsverzeichnis

|          |                               |          |
|----------|-------------------------------|----------|
| <b>1</b> | <b>Angabe</b>                 | <b>2</b> |
| 1.1      | Einführung . . . . .          | 2        |
| 1.2      | Anforderungsanalyse . . . . . | 2        |
| <b>2</b> | <b>Datenbank Entwurf</b>      | <b>3</b> |
| 2.1      | Diagramm . . . . .            | 3        |
| 2.2      | Konfiguration . . . . .       | 3        |
| <b>3</b> | <b>Fragmentierung</b>         | <b>3</b> |
| 3.1      | Horizontal . . . . .          | 3        |
| 3.2      | Vertikal . . . . .            | 3        |
| <b>4</b> | <b>Abfragen</b>               | <b>4</b> |
| <b>5</b> | <b>Quellen</b>                | <b>4</b> |

# 1 Angabe

## 1.1 Einführung

Es soll eine Verteilte Datenbank mit Oracle XE realisiert werden. Die Hauptaufgabe ist es herauszufinden wie man eine verteilte Datenbank erstellt. Die einzige Informationsquelle ist das weite Netz, allem voran die Oracle Dokumentation Site. Der komplette Datenbankentwurf für ein verteiltes Oracle-DBS soll erstellt werden. Zu verwenden ist dabei eine geeignete Fragmentierungsmethode! Die einzelnen Schritte sind zu dokumentieren, und Abfragen durchzuführen, die eine Connection auf die verteilte Instanz nötig machen!

## 1.2 Anforderungsanalyse

Die MensaAustria plant die Erstellung einer Datenbank zur Vereinfachung der Arbeitsabläufe. Dabei ist zu beachten, dass jede Universität ihre eigene Verwaltung bekommt, jedoch von einer Stelle geleitet wird.

Eine Speise hat eine eindeutige Nummer, einen Namen und einen Typ (Vorspeise, Hauptspeise, Nachspeise). Eine Speise besteht aus mehreren Zutaten, wobei die Menge der einzelnen Zutaten gespeichert werden muss.

Eine Zutat hat eine eindeutige Nummer, einen Namen, eine Einheit (kg, Liter, etc.), einen Preis pro Einheit und einen bzw. mehrere Lieferanten, von denen die Adresse und eine eindeutige UIDNummer bekannt sind. Weiters wird die Kundennummer der Mensa beim jeweiligen Lieferanten vermerkt. Der Lagerbestand wird ebenfalls modelliert. Dazu sind von jeder Zutat die Anzahl der Einheiten, die momentan vorrätig sind, bekannt. Wenn eine Zutat nicht mehr vorrätig ist, so wird der Bestand auf 0 gesetzt.

Im Falle einer Bestellung werden ein bzw. mehrere Zutaten bei einem bestimmten Lieferanten bestellt. Die Bestellung hat eine eindeutige Bestellnummer, ein Bestelldatum, ein voraussichtliches Lieferdatum. Die einzelnen Zutaten sind die Bestellposten, die eine pro Bestellung eindeutige Nummer haben, und bei denen die bestellte Menge und der verhandelte Preis gespeichert werden. Zu jeder Bestellung gibt es nach erfolgter Lieferung eine Rechnung, die verbucht werden muss. Rechnungen beziehen sich auf genau eine Bestellung. Dabei wird die eindeutige Rechnungsnummer, die Bankverbindung und die Rechnungssumme gespeichert, wobei letztere vom Bestellpreis differieren kann, wenn ein Skonto anfällt.

Ein Menü hat einen eindeutigen Namen, einen Preis und es ist gespeichert, aus welchen (verschiedenen) Speisen ein Menü besteht. Eine Speise kann selbstverständlich in mehreren Menüs vorkommen.

Um eine zu große Eintönigkeit in der Menüabfolge zu vermeiden, wird weiters gespeichert, an welchen Tagen welche Menüs serviert wurden. An jedem Tag werden genau drei Menüs serviert.

## 2 Umsetzung

### 2.1 Diagramm

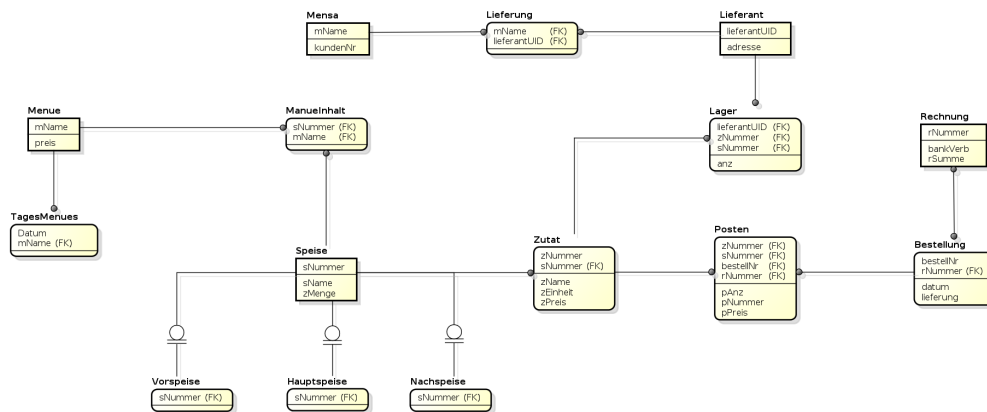


Abbildung 1: ERD

### 2.2 Konfiguration

### 2.3 Fragmentierung

#### 2.3.1 Horizontal

Die Mensa scheint perfekt für eine horizontale Fragmentierung geeignet, hier würden die Daten je nach geografischer Lage des typischen Zugriffsorts horizontal getrennt.

#### 2.3.2 Vertikal

Vertikale Trennung scheint in diesem Beispiel weniger sinnvoll.

### **3 Abfragen**

Database links are the invisible glue that makes location transparency possible.

### **4 Quellen**

<http://oreilly.com/catalog/ordistsys/section/ch01.html>