Datenbanken Laborbericht

Raimund Rothammel – Andreas Hofmeier

Auftraggeber:	Prof. Dr.-Ing. Uwe Meyer, Fachhochschule Bremen
Abgabe am:	17.05.2004

Andreas Hofmeier _____

Raimund Rothammel _____

Zusammenfassung

.
Dieser Bericht dokumentiert die Erstellung einer Datenbank durch Andreas Hofmeier und Raimund Rothammel im Zuge des Labors zur Datenbankvorlesung von Prof. Dr.-Ing. Meyer.

Das Szenario:

Sinn und Zweck dieser Datenbank soll es sein, den Erdkunde-Unterricht an allgemeinbildenden Schulen zu unterstützen. Dies soll erreicht werden, in dem viele allgemeine Informationen (z.B. Währung, Religion) und spezielle Informationen (z.B. Landkarte, Planquadrat) über Länder abgespeichert werden. Des weiteren sollen Namen (z.B. Flüssen, besonderen Gebäuden) mit einem Land in Verbindung gebracht werden können, bzw. umgekehrt.

Inhaltsverzeichnis

1 Das Szenario

1.1 Hintergrund

Im Erdkundeunterricht ist es ein ständiges Ärgernis die richtige Karte zu finden. Ist es möglich ein Land, eine Stadt oder einen Fluss auf der Karte zu Zeigen, ohne 20 Karten durchsuchen zu müssen und dann das Objekt der Begierde nur halb und im unteren rechten Eck der Karte zu sehen?

Sehr Hilfreich wäre ein Assistent, der einem nach Eingabe eines Namens (z.B. eines Berges) wichtige Daten, die Lage und die Nummer einer passenden Karte ausgibt.

1.2 Konkrete Anforderungen

Die Datenbank soll Länder und Informationen über diese verwalten. Lehrer, Schüler und allgemein Interessierte sollen in die Lage versetzt werden Informationen über ein bestimmtes Land abzurufen:

- Wo liegt das Land? (Geographisch und Kartenbezogen)
- Welche Nachbarn hat das Land?
- Was liegt in dem Land? (Flüsse, Berge, Städte, etc)
- Was zeichnet das Land aus?
    - Welche Sprachen werden in diesem Land gesprochen?
    - Womit wird bezahlt?
    - Lebenserwartung?

2 Realisierung

Um diesen Ansprüchen gerecht zu werden, müssen eine ganze Reihe von Realweltobjekten in der Datenbank abgebildet werden:

2.1 Objekte

1. Kontinente, Ozeane, Länder
2. Berge, Flüsse, Seen, Städte, Wälder, Wüsten

3. besondere Bauwerke / Sehenswürdigkeiten

2.1.1 Beziehungen

1. Länder liegen auf Kontinenten.
2. Länder liegen an Ländern oder Ozeanen
3. Flüsse, Berge, Seen, Wüsten, Wälder, Städte liegen in Ländern
4. besondere Bauwerke liegen in Städten

2.1.2 Eigenschaften

1. Alle diese Objekte haben zwei Namen (einen deutschen und einen englischen, falls einer existiert und vom deutschen abweicht), eine Position (Ausdehnung/Fläche) und sind auf mindestens einer Karte abgebildet.
2. Länder haben folgende Eigenschaften:
    - (Amts-)Sprache(n)
    - Arbeitslosigkeit
    - Bevölkerung
    - Bevölkerungswachstum
    - Bruttosozialprodukt
    - Einwohner pro Arzt
    - Fläche
    - Hauptstadt
    - Infrastruktur
    - Inflation
    - Länderdomain (Top-Level-Domain im Internet)
    - Ländervorwahl
    - Lebenserwartung
    - Mitgliedschaft in Bündnissen
    - Präsident
    - Religion(en)
    - Wirtschaftsform(en)
    - Währung(en)
    - Zeitzone(n)

3. Flüsse
    - durchschnittliche Tiefe
    - durchschnittliche Breite
    - Durchfluss in Liter pro Tag
    - Länge
    - Mündet in Meer
4. Berge
    - max Höhe
    - Typ [Vulkan, aktiver Vulkan]
5. Seen
    - max Tiefe
    - durchschnittliche Tiefe
    - Salz-/Süßwasser
6. Wüsten
    - Wüstenart [Sandwüsten entstehen durch Erosion von Kieswüsten, Kieswüsten entstehen durch Erosion von Stein- und Felswüsten, Stein- und Felswüsten (Hammada), Salzwüsten (Salztonebene)]
    - Temperatur Tag Durchschnitt
    - Temperatur Tag min
    - Temperatur Tag max
    - Temperatur Nacht Durchschnitt
    - Temperatur Nacht min
    - Temperatur Nacht max
    - Regenwahrscheinlichkeit
7. Wälder
    - Waldart [tropischer Regenwald, regengrüner tropischer und subtropischer Trockenwald, subtropischer Hartlaubgehölzwald, Monsunwald, Auwald, winterkahler Laubwald und Mischwald der gemäßigten Zone, immergrüner borealer Nadelwald, Bergwald, Nebelwald]
    - Temperatur Tag Durchschnitt
    - Temperatur Tag min
    - Temperatur Tag max
    - Temperatur Nacht Durchschnitt
    - Temperatur Nacht min

- Temperatur Nacht max
- Regenwahrscheinlichkeit

8. Städte

   - Einwohner
   - Infrastruktur
   - Alter

9. besondere Bauwerke / Sehenswürdigkeiten

   - Alter
   - max Höhe

2.2 Konventionen

In den folgenden Tabellen werden folgende Abkürzungen und Zeichen verwenden:

NW	Nullwert
D.	Default
Con.	Constraint Bemerkung
→	Feld verweist auf vorhandenen Eintrag in Tabelle
←	... wird von Tabelle verwiesen
PK	Primary Key
FK	Foreign Key
STRING	Steht für CHARACTER VARYING in der Datenbank
NUM	Steht für NUMERIC in der Datenbank
SERIAL	Automatisches wählen einer neuen Nummer (letzte plus eins) beim Einfügen eines neuen Datensatzes.

2.2.1 Ein Wort zu Positionen

Der im folgenden zur Positionsangabe eines Objektes verwendete String enthält eine approximierte Abbildung des Objektes in Form eines Kreises oder Rechtecks. Zu diesem Zweck wird auf eine externe Bibliothek zugegriffen. Die Angaben werden entweder als Punkt mit Radius oder als “links oben”-Punkt und “rechts unter”-Punkt eines Rechtecks gespeichert.

Das Feld `winfo` ist vorgesehen um weiterführende Verweise anzugeben. Zum Beispiel Internetadressen oder Buchnamen.

2.3 Tabellen/Entitys

2.3.1 laender

← vlaenderkontinente
← vlaenderozeane
← vlaenderberge_gebirge
← vlaenderfluesse
← vlaenderseen
← vlaenderstaedte
← vlaenderwaelder
← vlaenderwuesten
← vlaenderbuendnisse
← vlaenderexportgueter
← vlaenderklimatypen
← vlaenderreligionen
← vlaenderrohstoffe
← vlaendersprachen
← vlaenderwaehrungen
← vlaenderwirtschaftformen
← vlaenderzeitzohnen
← vlaenderzfalaender
← nachbarlaender
← zufindenauf

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
position	STRING	Nein	-	-	-
flaeche	NUM	Ja	-	-	in Quadratmetern
winfo	STRING	Ja	-	-	-
kennzeichen	STRING	Nein	-	-	<= 3 Zeichen
domain	STRING	Ja	-	-	<= 3 Zeichen, INet
vorwahl	STRING	Ja	-	-	Telefon
bevoelkerung	BIG INT	Ja	-	-	Einwohnerzahl
bevoelkerungswachstum	NUM	Ja	-	-	in Prozent
bruttosozialprodukt	NUM	Ja	-	-	in Euro
amtssprache	INT	Nein	-	FK	→ sprachen
arbeitslosigkeit	NUM	Ja	-	-	in Prozent
einwohner_pro_arzt	NUM	Ja	-	-	-
hauptstadt	INT	Nein	-	FK	→ staedte
infrastruktur	STRING	Ja	-	-	Kommentar
inflation	NUM	Ja	-	-	in Prozent
lebenserwartung	NUM	Ja	-	-	in Jahren
staatschef	STRING	Ja	-	-	-

2.3.2 kontinent, vlaenderkontinente

Ein Land liegt normalerweise auf einem Kontinent, kann aber auch über mehrere Kontinente verteilt sein. Auf einem Kontinent können mehrere Länder liegen.

kontinente:

← vlaenderkontinente ← zufindenauf

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
position	STRING	Nein	-	-	-
flaeche	NUM	Ja	-	-	in Quadratmetern
winfo	STRING	Ja	-	-	-

vlaenderkontinente: kontinent $-\overset{n:m}{\Diamond}-$ laender

Feldname	Datentyp	NW	D.	Con.	Bemerkung
kontinent_nr	INT	Nein	-	FK	→ kontinente
land_nr	INT	Nein	-	FK	→ laender

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

2.3.3 ozeane, vlaenderozeane

Ein Land kann an einem oder mehreren Ozean liegen. Es können mehrere Länder an ein und demselben Ozean liegen.

ozeane:

← vlaenderozeane ← zufindenauf

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
position	STRING	Nein	-	-	-
flaeche	NUM	Ja	-	-	in Quadratmetern
winfo	STRING	Ja	-	-	-
max_tiefe	NUM	Ja	-	-	in Metern
ds_tiefe	NUM	Ja	-	-	in Metern

vlaenderozeane: ozeane $\overset{n:m}{-\Diamond-}$ laender

Feldname	Datentyp	NW	D.	Con.	Bemerkung
ozean_nr	INT	Nein	-	FK	→ ozeane
land_nr	INT	Nein	-	FK	→ laender

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

2.3.4 berge_gebirge, vlaenderberge_gebirge, vberge_gebirge

Ein Berg oder ein Gebirge liegt in/an einem Land, kann sich allerdings auch über Landesgrenzen erstrecken. Berge und Gebirge werden mit Hilfe des Feldes Typ unterschieden. Ein Berg oder ein Gebirge kann an oder in einem Gebirge liegen, was durch eine extra Verknüpfungstabelle (vbergegebirge) abgebildet wird.

berge_gebirge:

← vberge_gebirge ← zufindenauf
← vlaenderberge_gebirge

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
position	STRING	Nein	-	-	-
flaeche	NUM	Ja	-	-	in Quadratmetern
winfo	STRING	Ja	-	-	-
typ	STRING	Nein	-	-	Berg/Gebirge
vtyp	STRING	Nein	-	-	aktiver Vulkan?
max_hoehe	NUM	Ja	-	-	in Metern

vlaenderberge_gebirge: berge_gebirge $\overset{n:m}{-\Diamond-}$ laender

Feldname	Datentyp	NW	D.	Con.	Bemerkung
bg_nr	INT	Nein	-	FK	→ berge_gebirge
land_nr	INT	Nein	-	FK	→ laender

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

vberge_gebirge: berge_gebirge $-\overset{n:m}{\Diamond}-$ berge_gebirge

Feldname	Datentyp	NW	D.	Con.	Bemerkung
bg1_nr	INT	Nein	-	FK	→ berge_gebirge
bg2_nr	INT	Nein	-	FK	→ berge_gebirge

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

2.3.5 fluesse, vlaenderfluesse

Ein Fluss fließt durch ein oder mehrere Länder. In einem Land können beliebig viele Flüsse fließen. Ein Fluss kann ein Land allerdings auch nur tangieren, das heißt, dass Land wird durch den Fluss begrenzt.

fluesse:

← vlaenderfluesse ← zufindenauf

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
position	STRING	Nein	-	-	-
flaeche	NUM	Ja	-	-	in Quadratmetern
winfo	STRING	Ja	-	-	-
ds_tiefe	NUM	Ja	-	-	in Metern
ds_breite	NUM	Ja	-	-	in Metern
durchfluss	NUM	Ja	-	-	in Liter pro Tag
laenge	NUM	Ja	-	-	in Metern
muendet_in_meer	STRING	Nein	-	-	Ja/Nein

vlaenderfluesse: fluesse $-\overset{n:m}{\Diamond}-$ laender

Feldname	Datentyp	NW	D.	Con.	Bemerkung
fluss_nr	INT	Nein	-	FK	→ fluesse
land_nr	INT	Nein	-	FK	→ laender

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

2.3.6 seen, vlaenderseen

Ein See kann in oder an einem Land liegen oder die Landesgrenzen überschreiten. In einem Land kann mehr als einen See liegen.

seen:

← vlaenderseen ← zufindenauf

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
position	STRING	Nein	-	-	-
flaeche	NUM	Ja	-	-	in Quadratmetern
winfo	STRING	Ja	-	-	-
wasserart	STRING	Ja	-	-	Salz/Süßwasser
max_tiefe	NUM	Ja	-	-	in Metern
ds_tiefe	NUM	Ja	-	-	in Metern

vlaenderseen: seen $-\overset{n:m}{\Diamond}-$ **laender**

Feldname	Datentyp	NW	D.	Con.	Bemerkung
see_nr	INT	Nein	-	FK	→ seen
land_nr	INT	Nein	-	FK	→ laender

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

2.3.7 staedte, vlaenderstaedte, bbauwerke

Eine Stadt liegt normalerweise nur in einem Land und wird auch zu diesem zugeordnet. Ausnahmen werden nicht berücksichtigt. Einem Land können mehrere Städte zugeordnet sein.

Die Hauptstadt eines Landes wird vom Datensatz des jeweiligen Landes direkt verwiesen. Zusätzlich muss die Stadt allerdings auch über die Verknüpfungstabelle `vlaenderstaedte` zugeordnet sein. (Andernfalls wäre eine Zuordnung der Stadt zu einem Land mit der `get_l()`-Funktion nicht möglich.

Das Gründungsjahr bezieht sich auf Christus – also unserer "normalen" Zeitrechnung. Wurde die Stadt vor Christus gegründet, so wird dies durch einen negativen Wert zum Ausdruck gebracht.

staedte:

← vlaenderstaedte ← zufindenauf
← bbauwerke

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
position	STRING	Nein	-	-	-
flaeche	NUM	Ja	-	-	in Quadratmetern
winfo	STRING	Ja	-	-	-
infrastruktur	STRING	Ja	-	-	Kommentar
gruendungsjahr	INT	Ja	-	-	Jahr der Gründung

vlaenderstaedte: staedte $-\overset{n:1}{\Diamond}-$ **laender**

Feldname	Datentyp	NW	D.	Con.	Bemerkung
stadt_nr	INT	Nein	-	FK	→ staedte
land_nr	INT	Nein	-	FK	→ laender

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

bbauwerke

Es wird angenommen, dass Sehenswürdigkeiten oder "besondere Bauwerke" jeweils in einer Stadt liegen oder zumindest einer Stadt zugeordnet werden können. Es wird fernen davon ausgegangen, dass Bauwerke keine Länder- oder Städtegrenzen überschreiten. Bauwerke können nur über den Umweg über die Stadt einem Land zugeordnet werden.

Das Entstehungsjahr bezieht sich auf Christus – also unserer "normalen" Zeitrechnung. Wurde das Objekt vor Christus gebaut, so wird dies durch einen negativen Wert zum Ausdruck gebracht.

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
position	STRING	Nein	-	-	-
flaeche	NUM	Ja	-	-	in Quadratmetern
winfo	STRING	Ja	-	-	-
max_hoehe	NUM	Ja	-	-	Höhe des Bauwerkes
entstehungsjahr	NUM	Ja	-	-	Jahr der Entstehung
stadt_nr	INT	Nein	-	FK	→ staedte

2.3.8 waelder, vlaenderwaelder, waldtypen

Ein Wald kann sich über mehrere Länder erstrecken oder nur in oder an einem liegen. Ein Land kann mehr als einen Wald beheimaten.

waelder:

← vlaenderwaelder ← zufindenauf

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
position	STRING	Nein	-	-	-
flaeche	NUM	Ja	-	-	in Quadratmetern
winfo	STRING	Ja	-	-	-
typ	INT	Nein	-	FK	→ waldtypen
temp_tag_ds	NUM	Ja	-	-	in °C
temp_tag_max	NUM	Ja	-	-	in °C
temp_tag_min	NUM	Ja	-	-	in °C
temp_nac_ds	NUM	Ja	-	-	in °C
temp_nac_max	NUM	Ja	-	-	in °C
temp_nac_min	NUM	Ja	-	-	in °C
regenwahrscheinlichkeit	NUM	Ja	-	-	in %

vlaenderwaelder: waelder $\overset{n:m}{-\Diamond-}$ laender

Feldname	Datentyp	NW	D.	Con.	Bemerkung
wald_nr	INT	Nein	-	FK	→ waelder
land_nr	INT	Nein	-	FK	→ laender

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

waldtypen:

← waelder

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
beschreibung	STRING	Nein	-	-	-

2.3.9 wuesten, vlaenderwuesten, wuestentypen

Eine Wüste kann sich über mehrere Länder erstrecken oder nur in oder an einem liegen. Ein Land kann mehr als eine Wüste beheimaten.

wuesten:

← vlaenderwuesten ← zufindenauf

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
position	STRING	Nein	-	-	-
flaeche	NUM	Ja	-	-	in Quadratmetern
winfo	STRING	Ja	-	-	-
typ	INT	Nein	-	FK	→ wuestentypen
temp_tag_ds	NUM	Ja	-	-	in °C
temp_tag_max	NUM	Ja	-	-	in °C
temp_tag_min	NUM	Ja	-	-	in °C
temp_nac_ds	NUM	Ja	-	-	in °C
temp_nac_max	NUM	Ja	-	-	in °C
temp_nac_min	NUM	Ja	-	-	in °C
regenwahrscheinlichkeit	NUM	Ja	-	-	in %

vlaenderwuesten: wuesten $-\overset{n:m}{\Diamond}-$ laender

Feldname	Datentyp	NW	D.	Con.	Bemerkung
wueste_nr	INT	Nein	-	FK	→ wuesten
land_nr	INT	Nein	-	FK	→ laender

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

wuestentypen:

← wuesten

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
beschreibung	STRING	Nein	-	-	-

2.3.10 buendnisse, vlaenderbuendnisse

Jedes Land kann beliebig vielen Bündnissen (EU, NATO, etc) zugeordnet werden. Ein Bündnis enthält mehrere Länder.

buendnisse:

← vlaenderbuendnisse

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
winfo	STRING	Ja	-	-	-

vlaenderbuendnisse: buendniss $\overset{n:m}{-\Diamond-}$ laender

Feldname	Datentyp	NW	D.	Con.	Bemerkung
buendnissname_nr	INT	Nein	-	FK	→ buendnisse
land_nr	INT	Nein	-	FK	→ laender

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

2.3.11 exportgueter

Wenn ein Land nennenswert exportiert, wird dies hier aufgeführt.

exportgueter:

← vlaenderexportgueter

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
winfo	STRING	Ja	-	-	-

vlaenderexportgueter: exportgueter $\overset{n:m}{-\Diamond-}$ laender

Feldname	Datentyp	NW	D.	Con.	Bemerkung
exportgut_nr	INT	Nein	-	FK	→ exportgueter
land_nr	INT	Nein	-	FK	→ laender

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

2.3.12 klimatyp, vlaenderklimatypen, klimazonen, vklimatypenzonen

Das Klima eines Landes wird beschrieben durch den Klimatyp. Der Klimatyp wiederrum setzt sich aus mehreren Klimazonen zusammen.

kimatypen:

← vlaenderklimatypen ← vklimatypenzonen

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ

vlaenderklimatypen: klimatypen $-\overset{1:n}{\Diamond}-$ laender

Feldname	Datentyp	NW	D.	Con.	Bemerkung
klimatyp_nr	INT	Nein	-	FK	→ klimatypen
land_nr	INT	Nein	-	FK	→ laender

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

klimazonen:

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
beschreibung	STRING	Nein	-	-	-

vklimatypenzonen: klimatypen $-\overset{n:m}{\Diamond}-$ klimazonen

Feldname	Datentyp	NW	D.	Con.	Bemerkung
klimatyp_nr	INT	Nein	-	FK	→ klimatypen
klimazone_nr	INT	Nein	-	FK	→ klimazonen

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

2.3.13 religion, vlaenderreligionen

In einem Land können beliebig viele Religionen ausgeübt werden. Ein Religion kann aber auch in mehr als einem Land vorkommen.

religionen:

← vlaenderreligionen

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
winfo	STRING	Ja	-	-	-

vlaenderreligionen: religion $\overset{n:m}{-\Diamond-}$ **laender**

Feldname	Datentyp	NW	D.	Con.	Bemerkung
religion_nr	INT	Nein	-	FK	→ religionen
land_nr	INT	Nein	-	FK	→ laender

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

2.3.14 rohstoffe, vlaenderrohstoffe

In einem Land sind möglicherweise Rohstoffe zu finden. Nennenswerte Rohstoffe werden hier aufgelistet und mit den jeweiligen Ländern verknüpft.

rohstoffe:

← vlaenderrohstoffe

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
winfo	STRING	Ja	-	-	-

vlaenderrohstoffe: rohstoffe $\overset{n:m}{-\Diamond-}$ **laender**

Feldname	Datentyp	NW	D.	Con.	Bemerkung
rohstoff_nr	INT	Nein	-	FK	→ rohstoffe
land_nr	INT	Nein	-	FK	→ laender

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

2.3.15 sprachen, vlaendersprachen

Es wird unterschieden zwischen der landesweiten Amtssprache, welche direkt von der Land-Tabelle auf die Sprache-Tabelle zeigt und den in einem Land gesprochenden Sprachen, welche über die Verknüpfungstabelle `vlaendersprachen` zugeordnet werden. Die Amtssprache muss zusätzlich in der Verknüpfungstabelle aufgeführt sein, damit eine Zuordnung mit Hilfe von `get_l()` möglich ist.

sprachen:

← laender ← vlaendersprachen

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Nein	-	-	UNIQ
winfo	STRING	Nein	-	-	-

vlaendersprachen: sprache $-\overset{n:m}{\Diamond}-$ laender:

Feldname	Datentyp	NW	D.	Con.	Bemerkung
sprache_nr	INT	Nein	-	FK	→ sprachen
land_nr	INT	Nein	-	FK	→ laender

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

2.3.16 waehrungen, vlaenderwaehrungen

Gezahlt wird in einem Land mit einer oder beliebig vielen Währungen. Ein und die selbe Währung kann in mehr als einem Land vorkommen.

waehrungen:

← vlaenderwaehrungen

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
winfo	STRING	Ja	-	-	-

vlaenderwaehrungen: waehrungen $-\overset{n:m}{\Diamond}-$ laender

Feldname	Datentyp	NW	D.	Con.	Bemerkung
waehrung_nr	INT	Nein	-	FK	→ waerungen
land_nr	INT	Nein	-	FK	→ laender

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

2.3.17 wirtschaftsform, vlaenderwirtschaftformen

In einem Land können beliebig viele Wirtschaftsformen aufweisen. Eine Wirtschaftsform kann in mehr als einem Land vorkommen.

wirtschaftsformen:

← vlaenderwirtschaftformen

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
winfo	STRING	Ja	-	-	-

vlaenderwirtschaftformen: wirtschaftsform $-\overset{n:m}{\Diamond}-$ laender

Feldname	Datentyp	NW	D.	Con.	Bemerkung
wirtschaftsform_nr	INT	Nein	-	FK	→ wirtschaftsformen
land_nr	INT	Nein	-	FK	→ laender

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

2.3.18 zeitzonen, vlaenderzeitzonen

Ein Land kann sich über mehr als eine Zeitzone erstrecken. In einer Zeitzone sind allerdings meinst mehrere Länder beheimatet.

zeitzonen:

← vlaenderzeitzonen

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
abkuerzung	STRING	Ja	-	-	Offizielle
zeitverschiebung	NUM	Nein	-	-	in Stunden

vlaenderzeitzonen: zeitzonen $-\overset{n:m}{\Diamond}-$ laender

Feldname	Datentyp	NW	D.	Con.	Bemerkung
zeitzone_nr	INT	Nein	-	FK	→ zeitzonen
land_nr	INT	Nein	-	FK	→ laender

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

2.3.19 nachbarlaender

Ein Land kann an beliebig viele andere Länder grenzen.

nachbarlaender: laender $-\overset{n:m}{\Diamond}-$ laender

Feldname	Datentyp	NW	D.	Con.	Bemerkung
land1_nr	INT	Nein	-	FK	→ laender
land2_nr	INT	Nein	-	FK	→ laender

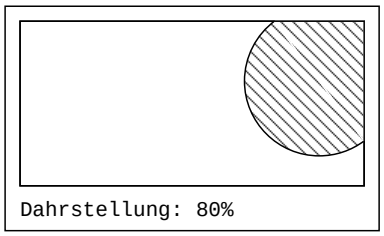
Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

2.3.20 zufindenauf, karten, kartentyp, kartenhersteller, tabellen

In diesen Tabellen sollen alle nötigen Informationen abgelegt werden, welche das wiederfinden des Objektes auf verschiedenen Karten ermöglichen. Ein Objekt kann auf mehreren Karten abgebildet sein. Eine Karte kann auch in Form eines Buches (Atlasses) vorliegen, daher ist die Seite zu nennen. Auf eine Seite bzw. Karte können mehrere Teilabschnitte enthalten sein, was es notwendig macht die Abbildungsnummer (abbnr) der Karte mit abzuspeichern.

Des weiteren ist es hilfreich zu wissen, ob das Objekt der Begierde ganz oder nur halb auf der Karte abgebildet ist und welches Teil abgeschnitten ist. Um

dies zu bewerkstelligen und um sich ein Bild von der Lage des Objektes auf der Karte machen zu können, stellt das Klientprogramm eine kleine Übersichtskarte (Kreis/Rechteck=Objekt in Rechteck=Karte) da:



Um diese Darstellung zu generieren werden Lageinformationen (Mittelpunkt und Radius des approximierenden Kreises ODER obere linke und untere rechte Ecke des approximierenden Rechtecks) des Objektes und die Position der oberen rechten und der unteren linken Ecke der Karte ausgelesen. Die Angabe Darstellung in Prozent wird aus der Datenbank entnommen. Sie kann aus den approximativen Lageinformationen nicht genau genug gerechnet werden.

zufindenauf:

← kontinente
← ozeane
← berge_gebirge
← fluesse
← seen
← staedte
← bbauwerke
← waelder
← wuesten

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
karte	INT	Nein	-	FK	→ karten
planquadrat	STRING	Nein	-	-	5 Zeichen (158AZ)
darstellung	INT	Nein	-	-	in Prozent (100=ganzes Objekt)
tabellen_nr	INT	Nein	-	FK	→ intabelle
objekt_nr	INT	Nein	-	FK	→ (tabellen_nr → intabelle)

Im Feld `tabellen_nr` wird auf den Namen der Tabelle verwiesen, in welcher zu diesem Datensatz gehörige Objekt zu finden ist. `object_nr` verweist auf das Objekt selber, in der eben verwiesenen Tabelle. Es handelt sich so zu sagen um einen Doppelverweiß.

intabelle:

← zufindenauf

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
tabellenname	STRING	Nein	-	-	UNIQ

karten:

← zufindenauf

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
hersteller_nr	INT	Nein	-	FK	→ kartenhersteller
typ	INT	Nein	-	FK	→ kartentyp
version	INT	Nein	-	-	-
seite	INT	Nein	-	-	-
abbnr	INT	Ja	-	-	-
massstab	INT	Nein	-	-	1:X
erscheinungsdatum	DATE	Nein	-	-	-

kartentyp

← karten

Feldname	Datentyp	NW	D.	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ
beschreibung	STRING	Nein	-	-	-
zu finden	STRING	Nein	-	-	z.B. Raumnummer

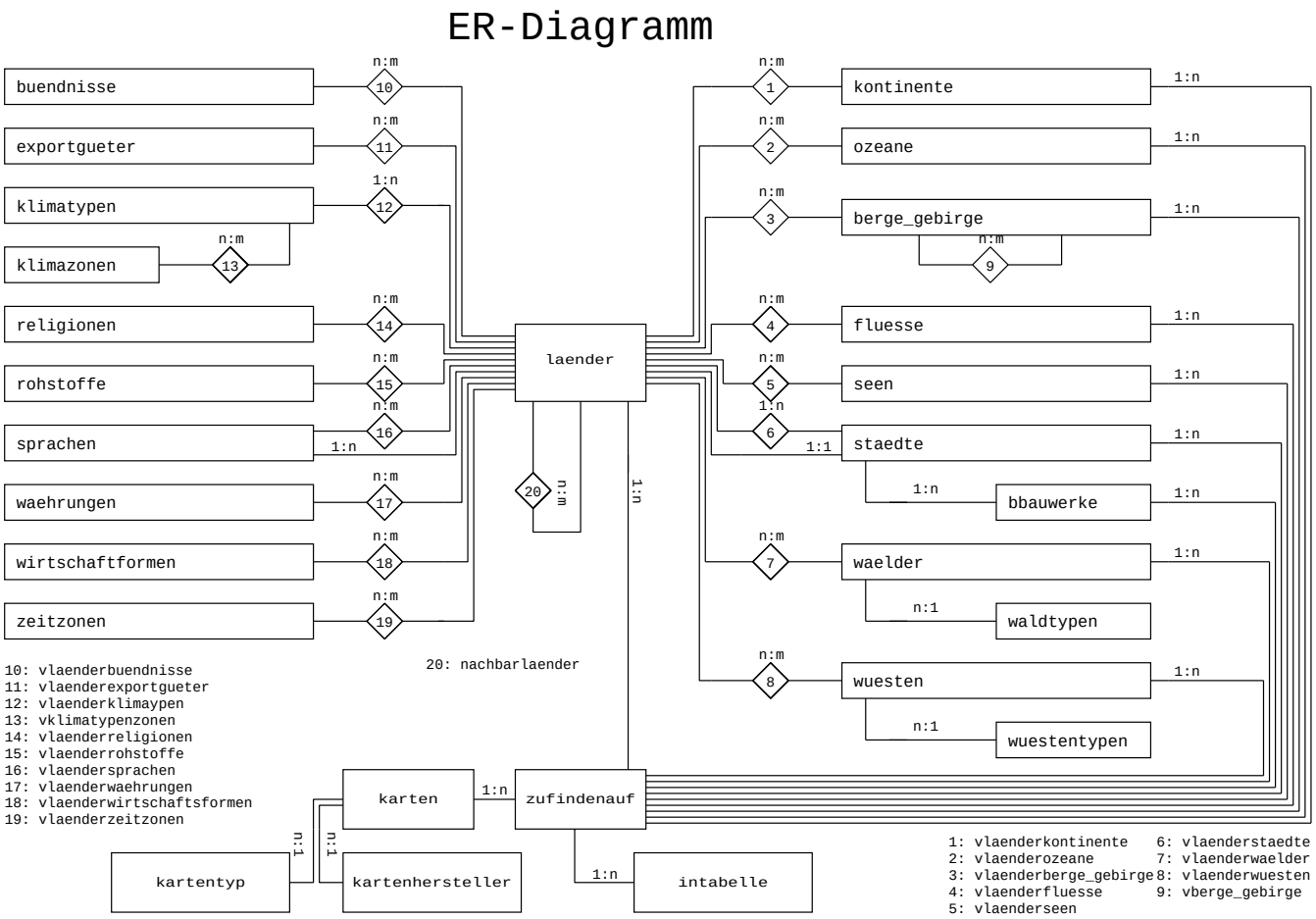
kartenhersteller

← karten

Feldname	Datentyp	NW	Def	Con.	Bemerkung
nr	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	UNIQ

2.4 ER-Diagramm

-



2.5 Implementierung, Normalisierung

Im Zuge der Normalisierung, welche wir bereits im Verlauf des Entwurfes berücksichtigt haben, haben wir zum Beispiel die Wald- und Wüstentypen aus den entsprechenden Tabellen herausgelöst um eine Redundanz zu vermeiden.

Die für weiterführende Informationen vorgesehenden Felder `winfo` stellen eine Schwachstelle der Normalisierung dar. Eine Informationsquelle könnte zum Beispiel zweimal auftreten, was zu Redundanzen führt. Des weiteren ist nicht richtig berücksichtigt, dass auch mehrere Informationsquelle angegeben werden können. (Gut, man könnte mit Semikolon trennen, aber sauber ist das natürlich nicht.)

Die 1:n-Beziehungen zu Städten und Klimatypen wurden genau wie fast alle Verknüpfungen in extra Verknüpfungstabellen gespeichert, um eine einfacherere Verarbeitung zu ermöglichen.

Die Eingrenzung der Zahlen, zum Beispiel auf 3 Vorkomma- und 2 Nachkommastellen bei Prozentangaben mussten wir wegen Problemen mit den Funktionen entfernen.

2.6 Funktionen zum Zugriff auf die Datenbank

2.6.1 search_all(suchstring)

Die Funktion `search_all(suchstring)` durchsucht alle Tabellen der Datenbank nach einem Vorkommen von `suchstring` und gibt Treffer folgendermaßen zurück:

Feldname	Datentyp	NW	D.	Con.	Bemerkung
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ
tabellenname	STRING	Nein	-	-	in welcher gefunden

Es werden die englischen sowie die deutschen Namen mittels der `LIKE`-Anweisung untersucht. Es ist daher möglich reguläre Ausdrücke zu verwenden:

Regulärer Ausdruck	Bedeutung
%	Steht für kein oder beliebig viele beliebige Zeichen

2.6.2 search_table_<tabellenname>(suchstr)

Die Funktion(en) `search_table_<tabellenname>(suchstr)` durchsucht die Tabelle `tabellenname` nach einem Vorkommen von `suchstr` in den Feldern `name` und `name_egl` ab. Zum Verglichen wird die `LIKE`-Anweisung verwendet, daher sind auch hier reguläre Ausdrücke (siehe bei Funktion `search_all`) zugelassen.

Die Funktion gibt die passenden Datensätze aus `tabellenname` komplett zurück, wobei direkte `n:1`-Beziehungen aufgelöst werden. So würden zum Beispiel die Felder `amptssprache` und `hauptstadt` in der Tabelle `laender` automatisch eingesetzt.

Im Parameter `tabellenname` erlaubte Werte:

- laender
- kontinent

- ozeane
- berge_gebirge
- fluesse
- seen
- staedte
- waelder
- wuesten
- buendnisse
- exportgueter
- klimatyp
- religion
- rohstoffe
- sprachen
- waehrungen
- wirtschaftsform
- zeitzonen
- nachbarlaender

2.6.3 get_l(tabellenname, suchstr)

Die Funktion `get_l(tabellenname, suchstr)` gibt alle Länder zurück, welche über die Tabelle `vlaender<tabellenname>` mit dem auf `suchstr` passtenden Eintrag in der Tabelle `tabellenname` verknüpft sind.

Beispiel: `get_l(■religionen■, ■Buddismus■)` würde eine Liste mit den Namen (Deutsch und Englisch) der Ländern ausgeben, in denen der Buddismus praktiziert wird. Reguläre Ausdrücke sind hier nicht zugelassen. Es kann also nur nach einer Religion zur Zeit gesucht werden.

Im Parameter `tabellenname` erlaubte Werte:

Wert	Verknüpfung
laender	Nachbar(land) zu Land
kontinent	Land liegt auf Kontinent
ozeane	Land liegt an Ozean
berge_gebirge	Berge oder Gebirge liegt in/an Land
fluesse	Durch/an Land fließt Fluss
seen	In/an Land liegt See
staedte	Städte liegen in/an Land
waelder	Wälder liegen in/an Land
wuesten	Wüsten liegen in/an Land
buendnisse	Land ist Mitglied in Bündnis
exportgueter	Land exportiert ...
klimatyp	Land weißt Klimatyp auf
religion	In Land tritt Religion auf
rohstoffe	In Land ist Rohstoff zu finden
sprachen	In Land wird Sprache gesprochen (nicht Amtssprache)
waehrungen	In Land wird mit Währung bezahlt
wirtschaftsform	Wirtschaftsform herrscht in Land
zeitzonen	Land liegt in Zeitzone

Rückgabetabelle:

Feldname	Datentyp	NW	D.	Con.	Bemerkung
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ

2.6.4 get_v(tabellenname, suchstr)

Die Funktion `get_v(tabellenname, suchstr)` macht das gleicht wie `get_l`, nur anders herum. Sie gibt alle Namen (Englisch und Deutsch) aus Tabelle `tabellennamen` zurück, welche über die Tabelle `vlaender<tabellenname>` mit den auf `suchstr` passtenden Eintrag in der Tabelle `laender` verknüpft sind.

Beispiel: `get_l(■religionen■, ■Deutschland■)` würde eine Liste mit den Namen (Deutsch und Englisch) aller Religionen zurückgeben, welche in Deutschland praktiziert werden (Falls diese in der Datenbank eingetragen sind). Reguläre Ausdrücke sind hier nicht zugelassen. Es kann also nur nach den Religionen in einem Land zur Zeit gesucht werden.

Im Parameter `tabellenname` erlaubte Werte:

Siehe Funktion `get_l()`.

Rückgabetabelle:

Feldname	Datentyp	NW	D.	Con.	Bemerkung
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ

2.6.5 get_berge_in_gebirgen(name)

Die Funktion `get_berge_in_gebirgen(name)` liefert eine Tabelle mit den Namen (Deutsch, Englisch) aller Berge/Gebirge in/an dem Berg/Gebirge `name` liegt.

Rückgabetabelle:

Feldname	Datentyp	NW	D.	Con.	Bemerkung
name	STRING	Nein	-	-	UNIQ
name_egl	STRING	Ja	-	-	UNIQ

2.6.6 get_liegtauf(tabellenname, name)

Die Funktion `get_liegtauf(tabellenname, name)` sucht alle Informationen heraus, die zum Wiederfinden eines Objektes (z.B. Land, Stadt, Fluss) notwendig sind. Das Objekt wird beschirieben durch seinen Namen `name` und dem Namen `tabellenname` der Tabelle in dem es liegt. Reguläre Ausdrücke sind nicht erlaubt.

Die Funktion sucht zu erst die zu dem Objekt (z.B. Land, Fluss) mit dem Namen `name` gehörige ID in der Tabelle `tabellenname`. Dann wird die zu der Tabelle `tabellenname` gehörige ID in `intabelle` gesucht. Diese beiden IDs werden dann mit den Feldern `tabellen_nr` und `objekt_nr` jedes Datensatzes in Tabelle `zufindenauf` verglichen. In passenden Datensätzen werden die Verseise auf diese vier Tabellen `zufindenauf`, `karten`, `kartentyp` und `kartenhersteller` weiter aufgelöst und dann zurückgegeben.

Rückgabetabelle:

Feldname	Datentyp	NW	D.	Con.	Bemerkung
karten_nr	INT	Nein	-	PK	SERIAL
hersteller	STRING	Nein	-	-	-
typ	STRING	Nein	-	-	-
version	INT	Nein	-	-	-
seite	INT	Nein	-	-	-
abbnr	INT	Ja	-	-	-
massstab	INT	Nein	-	-	1:X
erscheinungsdatum	DATE	Nein	-	-	-
planquadrat	STRING	Nein	-	-	5 Zeichen (158AZ)
darstellung	INT	Nein	-	-	in Prozent (100=ganzes Objekt)

Im Parameter `tabellenname` erlaubte Werte:

- laender
- kontinent
- ozeane
- berge_gebirge
- fluesse
- seen
- staedte
- waelder
- wuesten

2.6.7 get_bbauwerke(suchstr)

Die Funktion `get_bbauwerke(suchstr)` liefert eine Liste mit den Namen (Deutsch, Englisch) von besonderen Bauwerken, welche in der Stadt Namens `sname` liegen.

Beispiel:

Suche alle in der Datenbank verzeichneten Sehenswürdigkeiten (besondere Bauwerke, auch ein Wasserfall kann in diesem Fall ein "besonderes Bauwerk sein") welche in Deutschland liegen.

1. Schritt: Suche alle Städte in Deutschland: `result = get_v(■staedte■, ■Deutschland■)`
2. Schritt: Suche alle Sehenswürdigkeiten in den gefundenen Städten. Jede Stadt muss einzeln abgefragt werden. `for i in result: res2[i] = get_bbauwerke(result[i]);`
3. Schritt: Weitere Informationen zu den gefundenen Sehenswürdigkeiten abrufen. `for i in result: for j in res2[i]: searchtable(■bbauwerke■,`

```
res[i][j])
```

2.7 Sicherheitskonzept

Das Sicherheitskonzept ist hier relativ simpel: Jeder darf alles lesen, dazu dienen diese Funktionen:

- `search_all()`
- `search_table()`
- `get_l()`
- `get_v()`
- `get_berge_in_gebirgen()`
- `get_liegtauf()`
- `get_bbauwerke()`

Und die Systemverwalter können etwas verändern, dazu wurden folgende Funktionen vorgesehen:

- `put_table_line()`,
- `remove_table_line()`

3 Quellangaben

- Das Wissen um die Datenbank erstellen zu können bezogen wir aus der Vorlesung Datenbanken von Prof. Dr.-Ing. U. Meyer und der dazugehörigen Laborveranstaltung an der Hochschule Bremen.
- Als Hilfsmittel dienten uns `Das offizielle PostgreSQL-Handbuch` von `Peter Eisentraut` (1. Auflage 2003) sowie `SQL - Der Schlüssel zu rela- tionalen Datenbanken` von `Gregor Kuhlmann` und `Friedrich Müllmer-stadt` (Auflage 2001).
- `www.erdkunde-online.de` lieferte einige Anregungen und beantwortete Fragen rund um das Thema Erdkunde.
- Unter `http://www.m-forkel.de/klima/koeppen.html` fanden wir Informationen zur Katerlogisierung vom Klima.