Professur Datenschutz
und Datensicherheit
Fakultät Informatik
Institut für Systemarchitektur

7. Übung - Datenbanken

„Informatik I für Verkehrsingenieure“

Aufgaben inkl. Beispiellösungen

1. Aufgabe: DBS

a Was ist die Kernaufgabe von Datenbanksystemen?

b Beschreiben Sie kurz die Abstraktionsebenen der 3-Schema-Architektur.

c Aus welchen Komponenten besteht ein Datenbanksystem? Was sind ihre Aufgaben?

Lösung:

a Datenbanksysteme dienen dazu, große Datenmengen effizient, widerspruchsfrei und dauerhaft zu speichern und zu verwalten.

b **interne Ebene:** Beschreibung der physischen Abspeicherung der Daten

konzeptuelle Ebene: Beschreibung der Strukturierung der Daten in der Datenbank (z.B. Tabellen)

externe Ebene: Definition von Sichten für externe Nutzer und Anwendungen

c Ein Datenbanksystem (DBS) besteht aus einem Datenbankmanagementsystem (DBMS) und der eigentlichen Datenbank (bzw. mehreren Datenbanken). In der Datenbank sind die Daten sowie Metainformationen persistent gespeichert. Das DBMS umfasst alle Programme zur Erzeugung, Verwaltung und Manipulation einer Datenbank. Es behandelt die Anfragen an das System, wie das Auslesen von Datensätzen und das Einfügen neuer Daten, erstellt verschiedene Sichten für unterschiedliche Aufgaben und kümmert sich um die Konsistenz der Daten innerhalb der Datenbank.

2. Aufgabe: Schlüssel

Was sind Schlüssel? Welche Arten gibt es? Überlegen Sie sich für die verschiedenen Schlüsselarten jeweils ein geeignetes Beispiel.

Lösung:

Ein Schlüssel dient in der Datenbank dazu, Datensätze in einer Tabelle eindeutig zu identifizieren.

Primärschlüssel Der Primärschlüssel dient der eindeutigen Identifikation eines Datensatzes und besteht aus möglichst wenigen Attributen. Benutzt man beispielsweise den Namen und das Geburtsdatum einer Person als Primärschlüssel, würde man hier annehmen, dass es keine zwei Personen gibt, bei denen diese Daten übereinstimmen. Um Eindeutigkeit zu gewährleisten, wird oft auch ein künstliches neues Attribut eingeführt (Surrogatschlüssel, z.B. Personennummer).

Sekundärschlüssel dienen dazu, die Suche nach weiteren Attributen effizienter zu machen (Index). So könnte man in einer Personendatenbank das Geburtsjahr als Sekundärschlüssel einführen.

Fremdschlüssel Ein Fremdschlüssel verweist auf den Primärschlüssel in einer anderen Tabelle. So kann man zum Beispiel eine Tabelle mit Studenten anlegen und den Schlüssel *ID* einführen. In einer weiteren Tabelle zur Verwaltung von Prüfungsanmeldungen könnte das Attribut *ID* als Fremdschlüssel verwendet werden.

3. Aufgabe: Normalformen

a Was sind die drei Normalformen für Relationen?

b Überführen Sie die Tabelle nacheinander in die drei Normalformen! (Annahmen: *n*:*m*-Beziehung zwischen Mitarbeitern und Aufträgen; pro Abteilung gibt es ein Faxgerät.) Benennen Sie jeweils die vorhandenen Schlüssel!

ID	**Name**	**Vorname**	**Abteilung**	**Fax**	**Aufträge**
123456	Müller	Martin	A01	3765	1255, 7543
387645	Schmidt	Jana	A03	8865	8966
723999	Schulze	Stephan	A01	3765	3378, 1255

Lösung:

a **1. Normalform (1NF)** Alle Attribute sind atomar.

2. Normalform (2NF) Die Relation befindet sich in 1NF und alle Nichtschlüsselattribute sind voll funktional abhängig vom Primärschlüssel.

3. Normalform (3NF) Die Relation befindet sich in 2NF und Nichtschlüsselattribut ist transitiv vom Primärschlüssel abhängig.

b Primärschlüssel: ID; 1NF verletzt, denn das Attribut „Aufträge“ enthält nicht nur atomare Inhalte → Umformung: pro Wert eine Zeile (neuer Primärschlüssel notwendig, z.B. ID+Auftrag)

Tabelle in 1NF:

ID	Name	Vorname	Abteilung	Fax	Auftrag
123456	Müller	Martin	A01	3765	1255
123456	Müller	Martin	A01	3765	7543
387645	Schmidt	Jana	A03	8865	8966
723999	Schulze	Stephan	A01	3765	3378
723999	Schulze	Stephan	A01	3765	1255

2NF verletzt, denn die Attribute Name, Vorname, Abteilung und Fax hängen nur von der ID ab → Aufspaltung in zwei Tabellen

Tabelle 1 (Primarschlüssel: ID):

ID	Name	Vorname	Abteilung	Fax
123456	Müller	Martin	A01	3765
387645	Schmidt	Jana	A03	8865
723999	Schulze	Stephan	A01	3765

Tabelle 2 (Primärschlüssel: Auftrag+ID, Fremdschlüssel: ID):

Auftrag	ID
1255	123456
7543	123456
8966	387645
3378	723999
1255	723999

3NF bei Tabelle 1 verletzt, denn die Faxnummer hängt nur von der Abteilung ab → Aufspaltung in zwei Tabellen

Tabelle 1 (Primarschlüssel: ID, Fremdschlüssel: Abteilung):

ID	Name	Vorname	Abteilung
123456	Müller	Martin	A01
387645	Schmidt	Jana	A03
723999	Schulze	Stephan	A01

Tabelle 2 (Primarschlüssel: Abteilung):

Abteilung	Fax
A01	3765
A03	8865

4. Aufgabe: SQL

Geben Sie die SQL-Anfragen für die folgenden Aufgaben an. Die Aufgaben beziehen sich auf die Geographiedatenbank Mondial (`http://www.dbis.informatik.uni-goettingen.de/Mondial/`). Die SQL-Anfragen können über das Web-Interface der Mondial-Datenbank (`http://www.semwebtech.org/sqlfrontend/`) getestet werden.

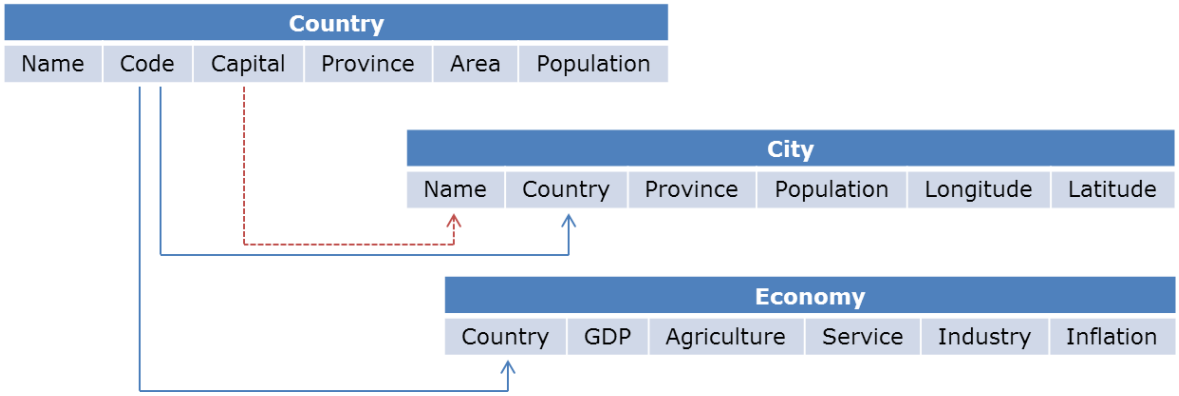

Abbildung 1: Ausschnitt aus den Relationen der Mondial-Datenbank.

a Lassen Sie sich alle in Tabelle Country gespeicherten Datensätze ausgeben.

b Lassen Sie sich die Anzahl der in Tabelle Country gespeicherten Länder ausgeben.

c Lassen Sie sich die größte Einwohnerzahl sowie die größte Fläche der in Tabelle Country gespeicherten Länder ausgeben.

d Lassen Sie sich von der Tabelle Country die Datensätze für alle Länder ausgeben, die mehr als 100000000 Einwohner haben. Die Ausgabe soll aufsteigend nach den Namen der Länder sortiert sein.

e Lassen Sie sich für die Länder mit mehr als 100000000 Einwohnern den Namen des Landes, die Einwohnerzahl und die Hauptstadt mit der zugehörigen Einwohnerzahl anzeigen. Die Ausgabe soll absteigend nach der Einwohnerzahl des Landes erfolgen. (Hinweis: Spalten mit gleichem Namen können mit „as“ einen Aliasnamen zugewiesen bekommen.)

f Bestimmen Sie die Stadt mit der größten Einwohnerzahl und lassen Sie sich den Namen der Stadt, den Namen des Landes, in dem sie liegt, und die Einwohnerzahl der Stadt ausgeben. (Hinweis: Die Anfrage erfordert eine verschachtelte SQL-Anweisung.)

Lösung:

a
```
select *
from country
```

b
```
select count(*)
from country
```

c
```
select max(population), max(area)
from country
```

d
```
select *
from country
where population > 100000000
order by name
```

e
```
select country.name, country.population as populationC, country.capital, city.population
from country,city
where country.population > 100000000 and country.capital=city.name
order by country.population desc
```

f
```
select city.name as stadt, country.name as land, city.population as einwohner
from city,country
where city.population = (
select max(population)
from city) and city.country=country.code
```