Hochschule Landshut
Fakultät Informatik

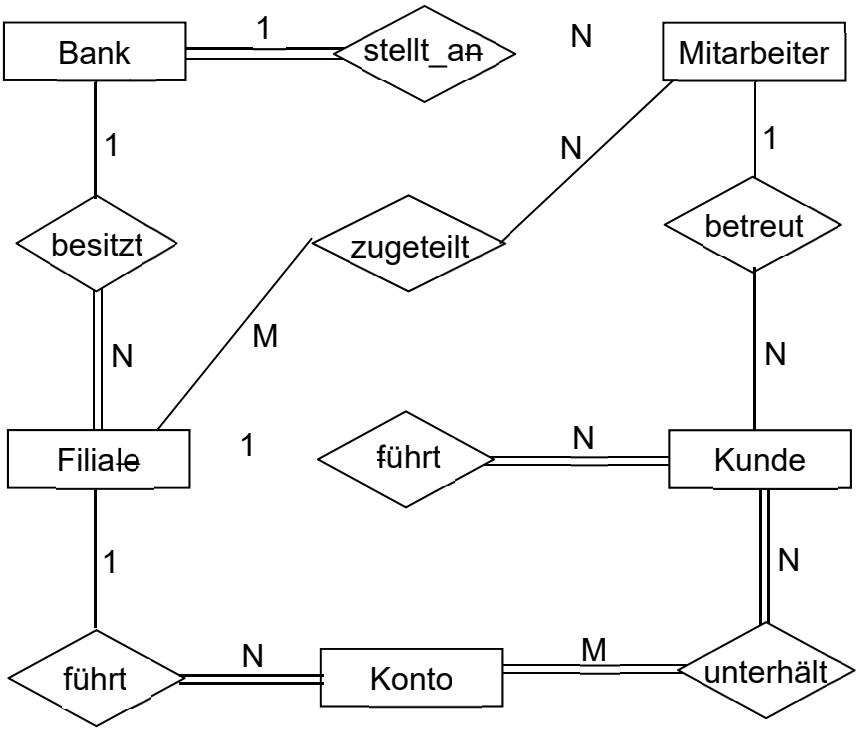
Prof. Dr. Studt, Hr. Franzke
Hr. Surner
WS23/24

Praktikum Datenbanken

Aufgabe 2: Datenbank-Abfrage unter Postgres

In dieser Aufgabe sollen Sie einige Anfragen an eine vorgegebene relationale Datenbank über Banken und ihre Filialen stellen. Hierfür benutzen Sie das relationale Open-Source-Datenbank-Management-System *PostgreSQL*. *PostgreSQL* steht in unseren Praktikumsräumen zur Verfügung, kann aber auch unter www.postgresql.org heruntergeladen werden.

Das der Aufgabe zugrundeliegende ER-Diagramm in Chen-Notation:



Aufgabenstellung:

Holen Sie also mit je einem SQL-Select-Statement die folgenden Informationen aus der Datenbank heraus:

1. Alle Informationen über alle Banken in der Tabelle Bank.
2. Alle Informationen über die Banken mit Geschäftssitz *Hamburg* in der Tabelle Bank.
3. Die Namen und Wohnorte der Mitarbeiter, die ab dem 28.4.1998 angestellt wurden, geordnet nach Namen.
4. Join in zwei Notationen (Komma-Schreibweise und Join-Schreibweise): Die Namen und Wohnorte der Bank-Mitarbeiter, mit dem Namen und der Bankleitzahl

ihrer Bank, aber nur, wenn die Mitarbeiter ein niedriges Monatsgehalt (< € 2500) haben. Auf die Join-Bedingung achten!

5. Join über drei Tabellen: Die Nummern und Arten der Konten, die Kunden unterhalten, die durch den Mitarbeiter Olafson betreut werden.

6a. Join über viele Tabellen ("Riesenjoin 1"):

Die Besitzer, Nummern und Guthaben der Tagesgeldkonten, die Kunden unterhalten, die durch Mitarbeiter von Frankfurter Banken betreut werden, geordnet nach den Kontobesitzern, bei gleichem Kontobesitzer nach Kontonummer.

6b. Noch ein Join über viele Tabellen ("Riesenjoin 2"):

Die Besitzer, Nummern und Guthaben der Tagesgeldkonten, die Kunden unterhalten, die bei Filialen von Frankfurter Banken geführt werden, wiederum geordnet nach den Kontobesitzern, bei gleichem Kontobesitzer nach Kontonummer.

7. Die Anzahl der Mitarbeiter aus *Hamburg*.

8. Die Namen der Mitarbeiter mit überdurchschnittlichem Monatsgehalt, und ihre Monatsgehälter.

9. Geschachteltes Select: Die Namen der Mitarbeiter und ihrer Bank, die das folgende Kriterium erfüllen: sie betreuen Kunden aus *Leipzig.*

10. Zwei äquivalente Selects: ein geschachteltes Select und ein Select mit einem Join über zwei Tabellen: Die Kontonummern und Guthaben aller Girokonten der Filialen der *BlackBank*, absteigend geordnet nach Kontonummer.

Materialien:

Zur Bearbeitung dieser Aufgabe stehen auf Moodle zur Verfügung:

- eine Datei *creates.sql* mit Tabellendefinitionen (also *Create Table* Statements),
- eine Datei *inserts.sql* mit Einfügestatements für diese Tabellen (also *Insert* Statements).

Hinweis:

Formulieren Sie Ihre SQL-Anfragen zunächst in einem Editor und kopieren Sie sie anschließend in das Postgres-Fenster. Dann müssen Sie leicht fehlerhafte Anfragen nach Korrektur nicht noch einmal komplett neu eintippen.

Abgabe:
Am Ende des jeweiligen Praktikumstermins

Anhang: Postgres

Für die Lösung dieser Aufgabe verwenden Sie das relationale Open-Source-Datenbank-Management-System Postgres (www.postgresql.org). Es besitzt u.a. eine einfache textuelle Oberfläche, die aus einer Linux-Shell heraus folgendermaßen zu bedienen ist:

Datenbank anlegen: createdb -h vm3-1 *db_name*
Datenbank öffnen: psql -h vm3-1 *db_name*
Datenbank löschen: dropdb -h vm3-1 *db_name*
(*db-name* ist ein von Ihnen erzeugter Name für Ihre Datenbank.)

Wenn eine Postgres-Datenbank mit createdb ... angelegt bzw. mit psql ... geöffnet wurde, lassen sich die folgenden Postgres-Kommandos eingeben:

Kommando	**Erläuterung**
\?	**Hilfe für diese Postgres-Kommandos**
\connect <dbname | -> <user>	mit neuer Datenbank verbinden
\d <tabelle>	**Details einer Tabelle auflisten (Spalten, Indexe, ...)**
\d <index>	Details von Indexen auflisten
\dd [<object>]	Kommentar für Objekt zeigen
\df	Funktionen auflisten
\di	Indexe auflisten
\do	Operatoren auflisten
\ds	Sequenzen auflisten
\dS	Systemtabellen und -views auflisten
\dt	**Tabellen auflisten**
\dT	Datentypen auflisten
\dv	Views auflisten
\h [<cmd>]	**Syntaxhilfe für SQL-Kommandos, \h* für alle SQL-Kommandos**
\i <datei>	**Befehle aus Datei ausführen (Datei kann über Pfadnamen angegeben werden)**
\l	**alle Datenbanken auflisten**
\o [<fname>] [|<cmd>]	Anfrageergebnisse in Datei oder |Pipe schreiben
\p	aktuellen Inhalt des Anfragepuffers zeigen
\q	**quit: psql beenden**
\r	Anfragepuffer löschen
SQL-Kommandos	**Create Table, Insert, Select, Update, Delete, ...**

Für das Anlegen der Datenbank über die Banken und ihre Filialen ist das Kommando \i.... geeignet, angewandt auf die zur Verfügung gestellten Dateien mit den entsprechenden Create Table ... und Insert... Statements. Alternativ lassen sich zuerst die Create-Table-Kommandos und anschließend die Insert-Kommandos auch einfach in das Postgres-Fenster hineinkopieren und anschließend ausführen.

Anschließend lassen sich SQL-Kommandos wie z.B. select ... from ... where ... eingeben.

Postgres Tutorial: *www.postgresqltutorial.com*