

SQL-DDL und SQL-Anfragen

Ausgangspunkt für Aufgabe 1 ist wiederum das aus den vorherigen Blättern bekannte Beispiel der Kontoführung mit folgendem relationalen Schema:

Kategorie: (Bezeichnung:string, Klassifikationskriterium:string)
Kunde: (Vorname:string, Name:string, Gebdat:date, Bezeichnung:string)
Konto: (Kontonr: integer, Betrag:numeric, Gebühr:numeric)
Festgeldkonto: (Kontonr: integer, Fälligkeit:date, Zinssatz: numeric)
Girokonto: (Kontonr: integer, Dispolimit:numeric, Telebanking:boolean)
Kontobewegung: (Datum:date, Vorname:string, Name:string, Gebdat:date, Kontonr:integer, Betrag: numeric, Bewegungsart: string)
Kontoauszug: (Erstelldat:date, Kontonr:integer, Startdat:date, Gesamt:numeric)
besitzt: (Vorname:string, Name:string, Gebdat:date, Kontonr:integer)
listet: (Erstelldat:date, Kontonr:integer, Datum:date, Vorname:string, Name:string, Gebdat:date)

Aufgabe 1: SQL-DDL

Setzen Sie das oben angegebene Relationenschema in ein SQL-Schema um.

Hinweis: Um Werte in die neu angelegten Tabellen einzutragen, verwenden sie den Befehl INSERT INTO Tabelle (Spaltenname, Spaltenname..) VALUES (Wert1,Wert2, ...). Es müssen nur die Spalten tatsächlich angegeben werden, für die Werte bestimmt werden sollen. Um die neue eingetragenen Werte anzusehen, benutzen sie SELECT * FROM Tabellenname.

Gehen Sie dabei in folgenden Schritten vor:

- 1.) Legen Sie die Tabelle Kategorie an. Die Kategoriebezeichnung soll ein String mit wechselnder Länge, maximal jedoch 15 Zeichen sein, das Klassifikationskriterium ein String mit wechselnder Länge (max. 100 Zeichen). Weiterhin soll die Tabelle festlegen, dass das Kategoriekriterium eingegeben werden muss und die Kategoriebezeichnung der Primärschlüssel ist. Fügen Sie dann folgende Tupel in die Tabelle ein: ('U-18', 'Alter unter 18 Jahren'),('G-Kunde', 'Geschäftskunde'), ('U-18', 'Alter unter 18 Jahren')
- 2.) Legen Sie die Tabelle Konto an.
Fügen sie das Tupel (Kontonr, Betrag, Gebühr): (174266,789.50,3) ein, danach das Tupel (Kontonr): (174255). Was passiert im zweiten Fall?
- 3.) Legen Sie ähnlich wie oben die Tabelle Kunde an. Hierbei soll gelten, dass die Kategoriebezeichnung eines Kunden (15 Zeichen, variabel) nicht leer sein darf. Um den zusammengesetzten Schlüssel darzustellen, verwenden Sie das Primary

Key(Spaltenname, Spaltenname, ...)-Konstrukt, das wie eine Spaltendefinition eingetragen wird.

Fügen Sie folgende Tupel ein (VN, NN, Gebdatum, Bezeichnung):

('Heinz', 'Schiller', '05/03/1942', 'P-Kunde') und (VN, NN, Gebdatum)('Anette', 'Müller', '07/25/1953'). Was passiert im zweiten Fall?

- 4.) Setzen Sie die restlichen Relationen analog zu 1-3 um.
- 5.) Oft ist es notwendig, eine Tabelle nachträglich noch zu verändern.
Fügen Sie zur Tabelle Kunde eine neue Spalte Wohnort mit Typ VARCHAR(30) hinzu.
Um Tabellen nachträglich weitere Spalten hinzuzufügen, gibt es die Anweisung ALTER TABLE Tabellename ADD COLUMN Spaltendefinition
Welche Werte sind in der neuen Spalte?
- 6.) Legen Sie einen Index zu den Bezeichnungen in der Tabelle Kunde an.
- 7.) Versuchen Sie, einen Index auf Bezeichnung in der Tabelle Kategorie anzulegen.
Welchen Schluss können Sie aus der Fehlermeldung ziehen?
- 8.) Löschen Sie den Index aus Aufgabe 5.) wieder.

Aufgabe 2: SQL-Anfragen, Kontobeispiel

Formulieren Sie folgende Anfragen basierend auf dem Schema aus Aufgabe 1 in SQL.

- 1.) Finden Sie alle Kunden.
- 2.) Finden Sie alle Kategoriebezeichnungen.
- 3.) Finden Sie alle Kontobewegungen des Kunden Heinz Schiller (geb. am 3.5.1942), bei denen mehr als 500 Euro bewegt wurden.
- 4.) Finden Sie alle Kunden des Kontos 174266.
- 5.) Wieviele Kunden haben mehr als 5000 Euro Guthaben.
- 6.) Finden Sie das Konto mit dem höchsten Guthaben.

Aufgabe 3: SQL-Anfragen, Bibliotheks-DB

Gegeben ist folgende Datenbank, die das Ausleihwesen einer Bibliothek unterstützt:

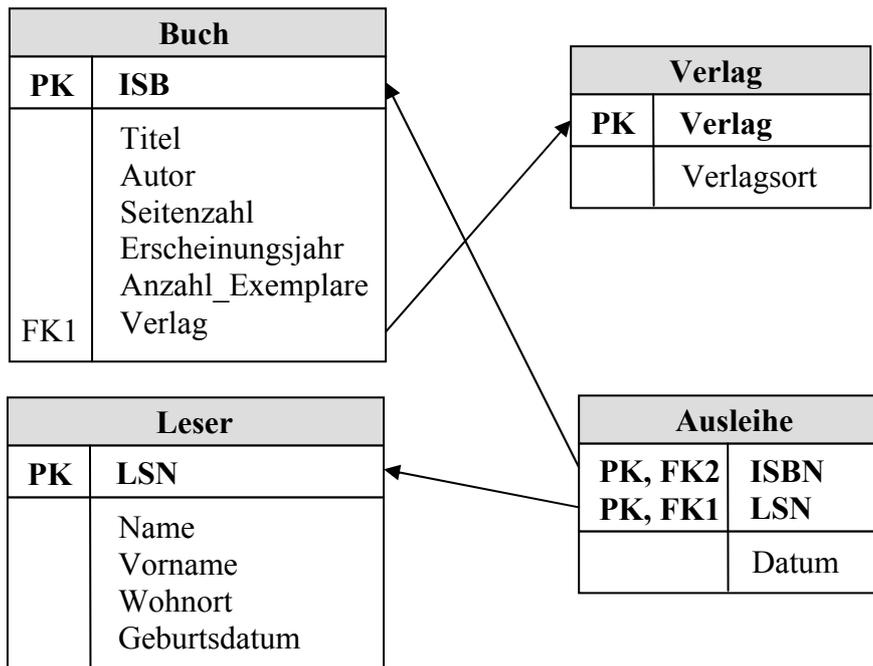
- Für Leser werden die wichtigsten Daten erfasst und jedem Leser eine eindeutige Lesernummer zugeordnet.
- Bei Büchern stellt die ISBN (eine zehnstellige Zahl) einen eindeutigen Schlüssel dar. Sie wird für jede Ausgabe neu vergeben, ändert sich aber für einzelne Exemplare nicht.
- Falls ein Leser ein Buch ausleiht, wird in der Relation „Ausleihe“ ein Tupel mit der ISBN des Buches, seiner Lesernummer und dem Ausleihdatum angelegt. Wir berücksichtigen noch nicht, dass es mehrere Exemplare eines Buches geben kann, d.h. es wird nur festgehalten, dass für diesen Leser und ein solches Buch eine Ausleihe besteht.

LESER (LSNR, NAME, VORNAME, WOHNORT, GEBDATUM)

BUCH (ISBN, TITEL, AUTOR, SEITENZAHL, VERLAG, ERSCHEINUNGSJAHR, ANZAHL_EXEMPLARE)

VERLAG (VERLAG, VERLAGSORT)

AUSLEIHE (ISBN, LSNR, DATUM)



Die Grafik stellt die Relationen noch einmal anschaulicher dar. Primärschlüssel (PK) und Fremdschlüssel (FK) sind als solche markiert, die Pfeile geben die Fremdschlüsselbeziehungen an

Formulieren Sie folgende Anfragen in SQL:

1. Welche Bücher (AUTOR, TITEL) hat der Leser Lemmi Schmöker ausgeliehen?
2. Welche Bücher (AUTOR, TITEL) sind in mehreren verschiedenen Ausgaben vorhanden?
3. Welche Leser (NAME, VORNAME) haben Bücher ausgeliehen, die an ihrem Wohnort verlegt wurden?
4. Welcher Leser (NAME, VORNAME) hat mindestens ein Buch ausgeliehen, das auch Leser Lemmi Schmöker ausgeliehen hat (Lemmi Schmöker soll nicht ausgegeben werden)?

Hinweis: SQL

Unter <http://sunbayer62.informatik.tu-muenchen.de:8000/> steht eine Übungsschnittstelle für SQL zur Verfügung. Mit diesem Interface sollten alle in der Übung angesprochenen SQL-Statements ausprobiert werden.

Ihr Login ist **dbslweb**, das Passwort **uebung**.

Beachten Sie, dass für alle Teilnehmer der Übung nur eine einzige Datenbank zur Verfügung steht. Daher sollten die von Ihnen angelegten Tabellen einen eindeutigen Namen haben, um nicht in Konflikt mit den Tabellen ihrer Kommilitonen zu geraten. Eine Möglichkeit dazu wäre, ihre Tabellennamen mit einem eindeutigen Präfix zu versehen, z.B. MEINNAME.TABELLE.

Beachten sie weiterhin, dass ihre Tabellen jederzeit von jemand anderem gelöscht werden können.

Bei Problemen bitte Mail an bauermi@in.tum.de

