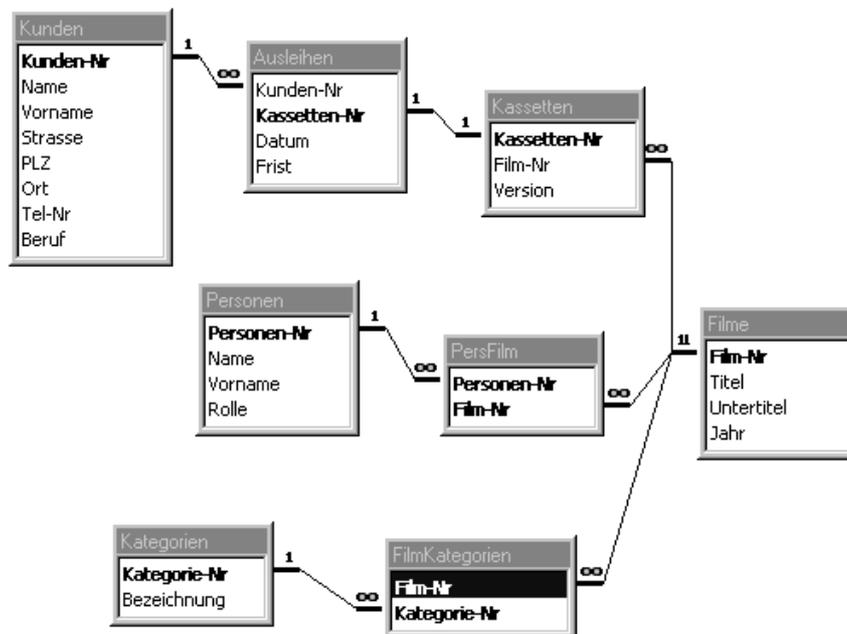


Einsatz von Informatikmitteln

Wintersemester 1999/2000

Datenverwaltung mit MS Access

Tutorial und Übungen



Praxis 4

Vorbemerkung

Dieses Tutorial besteht aus einem Grundmodul und einem Erganzungsmodul:

- Im *Grundmodul* werden der prinzipielle Aufbau der Datenbanksoftware Microsoft Access und drei wichtige Aufgaben beschrieben, die allgemein von Datenverwaltungssystemen unterstutzt werden mussen:
 1. Strukturierte Speicherung von Daten
 2. Eingabe von Daten
 3. Abfragen der Datenbank
- Das *Erganzungsmodul* besteht aus drei Abschnitten:
 1. Im ersten Abschnitt werden Berechnungen als Teil von Abfragen und das Importieren von Daten aus MS Excel, anderen Datenbanken und aus Textdateien vorgestellt.
 2. Der zweite Abschnitt erlautert das Erstellen von Berichten.
 3. Der dritte Abschnitt untersucht noch einmal genauer den Entwurf einer relationalen Datenbank. Sobald Sie einmal selbst eine Datenbank aufbauen wollen, sollten Sie sich unbedingt diesem Thema widmen. Anhand eines Beispiels werden die Normalisierungsschritte beschrieben, die notwendig sind, um Relationen zu erhalten, die weder Redundanz noch unschlussige Abhangigkeiten enthalten.

Hinweis

Dieses Tutorial bezieht sich auf die deutsche Version von MS Access 97. Die wichtigsten der beschriebenen Funktionen sind aber auch mit alteren und der neuen MS Access Version 2000 durchfuhrbar, allerdings treten Unterschiede im Fensteraufbau und in der Struktur der Menus und Befehle auf. Sollten Sie Probleme haben, die Befehle zu finden, wenden Sie am besten die Hilfefunktion von MS Access an oder fragen Sie die Assistierenden.

Zu diesem Tutorial benotigen Sie einige Dateien mit Beispieldaten. Sie finden diese im Format Access 2000 und Access 97 auf der Homepage zur Vorlesung unter <http://food.ethz.ch/info1/> und dort im Verzeichnis zu Praxis 4.

Hinweise zur Formatierung des Textes

1. In diesem Praxisteil werden Sie Schritt fur Schritt durch verschiedene ubungen gefuhrt.
2. Die Reihenfolge der einzelnen Schritte ist so nummeriert, wie Sie es bei den beiden Absatzen, die Sie gerade gelesen haben, sehen.

Bei einigen ubungen sind Erlauterungen zum behandelten Thema notwendig. Um einen solchen Text von den Instruktionen fur die ubung abzuheben, ist er grau hinterlegt. So wie Sie das bei diesem Absatz sehen.

Hervorhebungen und wichtige neue Begriffe werden **fett** gedruckt.

Verweise auf World Wide Web Seiten und andere Internet Dienste (sog. URLs) werden im Text durch eine **serifenlose** Schrift hervorgehoben, so wie dieser URL: <http://food.ethz.ch/info1/>.

Fur Text, den Sie wortlich bei einer ubung am Computer eingeben sollen, wird die Schriftart Courier verwendet.

Platzhalter in Eingabeaufforderungen werden zusatzlich kursiv gedruckt. An solchen Stellen mussen Sie den Platzhalter durch die jeweils fur Ihre Situation gultige Eingabe ersetzen. Beispiel: Um eine E-Mail an NETHZ-Support zu senden, ersetzen Sie `username` in `username@student.ethz.ch` durch `hotline`.

Bei der Beschreibung der Bedienung von Software werden Verweise auf Menupunkte oder andere Bedienungselemente durch **serifenlose fette Schrift** hervorgehoben.

© Institut fur Wissenschaftliches Rechnen, ETH Zurich, 1996-1999

An diesem Tutorial haben mitgearbeitet: Armin Feler, Jakob Lindenmeyer, Bettina Kemme, Alexander Kuckelberg, Defeng Ma, Florian Schlotke und Emil Zeller.

Dieses Tutorial und weitere Informationen und Dateien sind auch auf dem Internet verfugbar:
<http://food.ethz.ch/info1/>

Grundmodul

Ziele

- Sie können eine Datenbank erstellen, die aus mehreren Tabellen besteht.
- Sie können Abfragen an die Datenbank stellen und dabei Daten aus mehreren Tabellen miteinander verknüpfen.
- Sie kennen zwei Arten um Daten in eine Datenbank einzugeben.

Inhalt

1	Einführung	4
1.1	Starten, Speichern und Beenden	4
1.2	Allgemeine Hinweise	5
1.3	Die Hilfefunktion	5
2	Anlegen einer Datenbank	5
2.1	Erstellen einer Tabelle	6
2.2	Festlegen von Beziehungen zwischen Tabellen	7
3	Eingabe von Daten	9
3.1	Eingabe mit Hilfe eines Formulars	9
3.2	Eingabe in Tabellen-Form	10
4	Abfragen	10
4.1	Eine Beispiel-Abfrage	10
4.2	Aufbau von Abfragen	12
4.3	Eine existierende Datenbank abfragen	13
4.3.1	Die Videodatenbank	13
4.3.2	Beispiele von Abfragen	14
5	Testaufgabe	17
5.1	Die Datenbank: Das Landesforstinventar	17
5.2	Aufgabe 1: Einlesen einer existierenden Datenbank	19
5.3	Aufgabe 2: Abfragen stellen	19
5.4	Aufgabe 3: Erweiterung der Datenbank	19

1 Einführung

Immer mehr Information wird digital gespeichert, verarbeitet und analysiert. Um die damit verbundenen Daten in den Griff zu bekommen, werden Datenverwaltungssysteme eingesetzt, die komfortable Methoden bieten, um

- ein konsistentes und redundanzfreies Datenlayout zu entwerfen,
- Daten einzugeben oder aus anderen Anwendungen zu importieren,
- Information aus existierenden Daten zu extrahieren und zu analysieren,
- Analyseergebnisse in Grafiken oder Tabellen weiter zu verarbeiten oder an andere Anwendungen, z.B. MS Excel, zu exportieren.

In der Vorlesung wird das **Relationenmodell** besprochen, mit dessen Hilfe Daten der realen Welt übersichtlich strukturiert werden können. Eine Datenbank ist in Relationen unterteilt, die logische Einheiten der zu repräsentierenden Welt darstellen. Man kann sich eine Relation als **Tabelle** vorstellen. Die **Attribute** einer Relation werden in Spaltenform angeordnet und jede Zeile der Tabelle stellt ein **Tupel** bzw. **Datensatz** oder **Entität** dar. (Die Begriffe Tupel und Datensatz können praktisch synonym verwendet werden.)

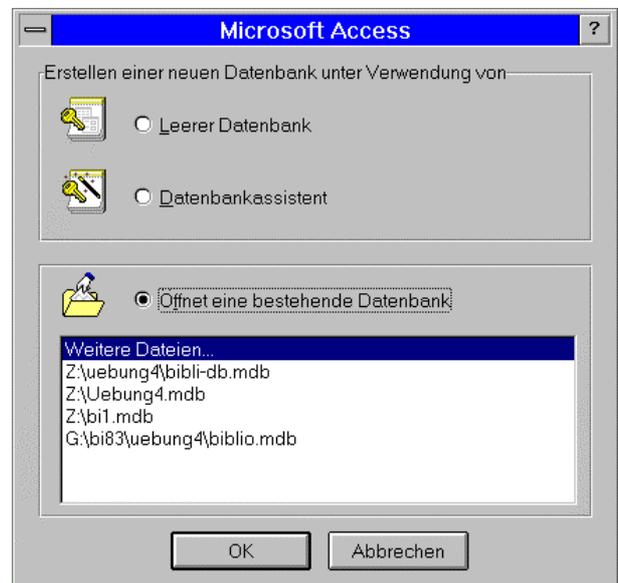
Damit Daten bequem eingegeben werden können, arbeiten Datenbanksysteme oft mit **Masken** bzw. mit **Formularen**. Um Information über existierende Daten zu erhalten, werden **Abfragen** an die Datenbank gestellt. Die Antworten auf diese Abfragen können in Form von **Berichten** weiterverarbeitet werden.

In dieser Übung verwenden wir Microsoft Access als Beispiel eines Datenbanksystems, das die oben beschriebenen Anforderungen erfüllt. MS Access gliedert eine Datenbankanwendung im Wesentlichen in die vier Bausteine **Tabelle**, **Formular**, **Abfrage** und **Bericht**, die wir in diesem Tutorial behandeln.

1.1 Starten, Speichern und Beenden

1.  Starten Sie Das Programm **Microsoft Access**. Sie finden die Software im **Start-Menü** unter **Programme**. Je nach Installation müssen Sie dort noch in das Untermenü **MS Office** wechseln.

2. Wenn Sie MS Access starten, können Sie wählen, ob Sie eine existierende Datenbank öffnen (um sie zu erweitern, Daten einzugeben, Abfragen zu stellen etc.) oder eine neue Datenbank erstellen wollen. In letzterem Fall bietet MS Access in Form des *Datenbankassistenten* Schablonen an, die häufig benutzte Typen von Datenbanken darstellen und somit den Entwurf einer neuen Datenbank erleichtern. Auch wenn Sie MS Access bereits gestartet haben, können Sie jederzeit eine neue Datenbank kreieren oder eine existierende öffnen (Befehle **Neu** und **Öffnen** im Menü **Datei** oder Symbole  und ). Jede Access Datenbank wird in einer Datei mit der Endung **.mdb** gespeichert.



3. Eine Speicherungsoperation (Befehl **Speichern** im Menü **Datei** bzw. Symbol ) bezieht sich immer auf das Objekt, das gerade aktiviert ist, nicht auf die gesamte Datenbank. Es ist ratsam, die an einem Objekt vorgenommenen Modifikationen zu speichern, bevor zu einem anderen Objekt übergegangen wird.

4. Mit dem Befehl **Beenden** im Menü **Datei** verlassen Sie MS Access.

1.2 Allgemeine Hinweise

Die Menüleiste von MS Access ändert sich je nachdem, was Sie gerade bearbeiten. Auch die Befehle, die Ihnen die Menüs zur Verfügung stellen, können unter bestimmten Bedingungen ausgeblendet sein, da sie im entsprechenden Kontext nicht möglich sind.

1.3 Die Hilfefunktion

Ähnlich wie in MS Excel stehen Ihnen zahlreiche Hilfefunktionen zur Verfügung, um MS Access, seine Mächtigkeit und seine Funktionsweise selbständig kennenzulernen.

Hilfeblasen: Wenn Sie kurze Zeit mit dem Mauszeiger über einem Symbol in den Symbolleisten verharren, erscheint eine kleine Sprechblase mit einer Erklärung zur Funktion des Symbols.

Statuszeile: Zu jeder Zeit erhalten Sie in der Statuszeile am unteren Bildschirmrand eine Kurzinformation über die momentan aktivierte Funktion oder über nächste Schritte, die zu tun wären.

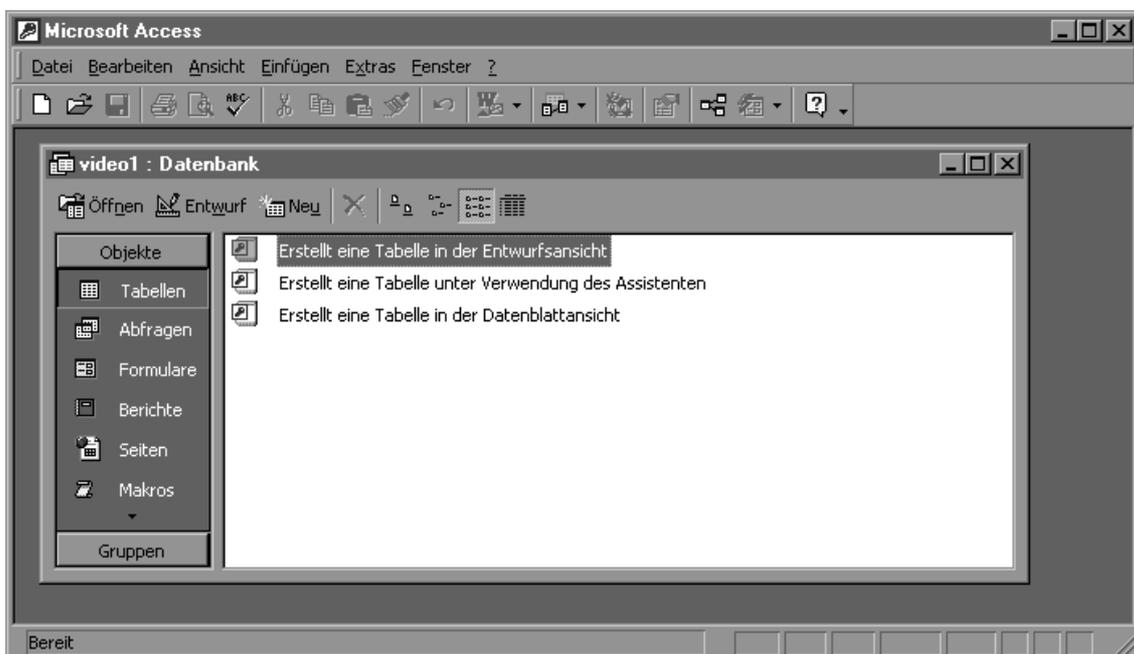
Hilfe: Das eigentliche Hilfesystem starten Sie im Menü **?**. Das Hilfesystem besteht aus einem Inhaltsverzeichnis, einer Index basierten Suche, der Suchmöglichkeit nach Begriffen, die im Hilfesystem verwendet werden und einem Hilfeassistenten. Dieser ermöglicht es Ihnen, die Fragen in beschreibender Form einzugeben. Der Hilfeassistent listet Ihnen dann dazu passende Hilfethemen auf, die schrittweise Anleitungen, Beispiele, Direkthilfe etc. beinhalten. Die Taste **F1** startet ebenfalls das Hilfesystem. Befinden Sie sich gerade in einer Dialogbox, bewirkt **F1**, dass Sie sofort zu der Hilfeinformation gelangen, die mit Ihren momentanen Aktivitäten zu tun hat.

Direkthilfe: Die Direkthilfe gibt kurze Erklärungen zu Befehlen, Dialogfelder und deren Komponenten, Bildschirmbereichen etc. Aktivieren Sie die Direkthilfe, indem Sie auf das Symbol  drücken und danach auf die Komponente, zu der Sie Hilfe benötigen. In manchen Fällen wird statt einer kurzen Erläuterung das Hilfesystem gestartet und umfangreiche Information angeboten. Um die Hilfe zu deaktivieren, klicken Sie nochmals auf das Symbol.

2 Anlegen einer Datenbank

Am Beispiel einer Videodatenbank wollen wir in diesem Abschnitt zeigen, wie man eine neue Datenbank anlegt. Die Videodatenbank verwaltet Daten, die typischerweise in einer Videothek anfallen, wie z.B. Daten über Kunden, Videokassetten, Reservationen, Ausleihen etc.

1. Wählen Sie beim Start von MS Access, dass Sie eine neue Datenbank anlegen wollen, und zwar ohne Verwendung des Datenbankassistenten. Den Assistenten lassen wir beiseite, um die einzelnen Schritte beim Aufbau einer Datenbank besser zu durchschauen. Als Datenbanknamen wählen Sie `video1.mdb`.



- Innerhalb des Access-Fensters erscheint ein **Datenbank**-Fenster, das den Namen `videod1` trägt. Dieses Fenster ist in mehrere Karteikarten unterteilt, wobei eine Karteikarte jeweils für eine Sammlung einer Art von Bausteinen der Datenbank verantwortlich ist. Am Anfang sind alle Karteikarten leer.

2.1 Erstellen einer Tabelle

Eine relational organisierte Datenbank besteht aus einer Menge verschiedener Tabellen. In diesem Beispiel werden Sie die Kunden-Tabelle und die Ausleihe-Tabelle der Videodatenbank definieren.

- Um eine neue Tabelle zu erstellen, wählen Sie die Karteikarte **Tabellen** und drücken Sie den **Neu**-Knopf. Im darauf entstehenden Fenster wählen Sie **Entwurfsansicht**. Sie erhalten nun ein leeres Formular, mit dem Sie die Felder (Attribute) der Tabelle beschreiben können

Datentypen geben an, um was für eine Art von Attribut es sich handelt. Häufig benutzte Datentypen sind **ganze Zahlen**, **Text**, **Datum**, **Währung** etc. Je nach Datentyp können dann verschiedene Operationen auf einem Attribut ausgeführt werden. Zum Beispiel kann getestet werden, ob ein Attribut vom Datentyp **Zahl** kleiner als ein vorgegebener Wert ist, ob ein Attribut vom Typ Text eine bestimmte Wortfolge enthält etc.

Für jedes Feld können bestimmte **Eigenschaften** definiert werden. Diese sind im unteren Teil der Entwurfsansicht angezeigt. Je nach Felddatentyp sind verschiedene Eigenschaften einstellbar. (Mehr Information dazu erhalten Sie in der Online-Hilfe).

Für die Eigenschaften der einzelnen Felder in der Kunden-Tabelle übernehmen Sie einfach die Default-Werte (Default-Werte sind die Werte, die vom System vorgeschlagen werden).

- Ein Feld wird definiert, indem Sie ihm einen Namen geben und ihm einen Datentyp zuordnen. Es stehen mehrere Datentypen zur Auswahl (wählen durch Klicken des Pfeilsymbols, das sichtbar wird, wenn man in die Spalte **Felddatentyp** gelangt). Die **Beschreibung** des Feldes ist nicht obligatorisch und dient lediglich zur Dokumentation Ihrer Attribute.
- Definieren Sie die Felder der Kunden-Tabelle wie in der folgenden Abbildung. Die *Kunden-Nr* ist dabei ein **AutoWert**, d.h. die Nummer wird automatisch vom System vergeben, wenn ein Datensatz eingegeben wird.

Feldname	Felddatentyp	Beschreibung
Kunden-Nr	AutoWert	
Name	Text	
Vorname	Text	
Strasse	Text	
PLZ	Text	
Ort	Text	
Tel-Nr	Zahl	
Beruf	Text	

Feldeigenschaften

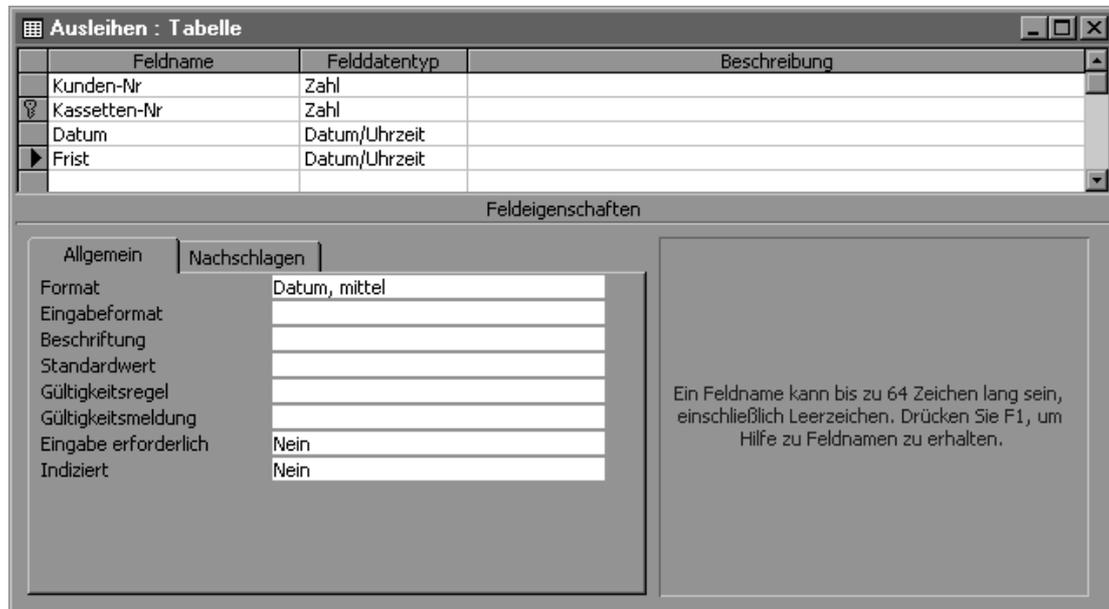
Allgemein | **Nachschlagen**

Feldgröße: Long Integer
 Neue Werte: Inkrement
 Format:
 Beschriftung:
 Indiziert: Ja (Ohne Duplikate)

Der Felddatentyp bestimmt das Format der Werte, die Benutzer in dem Feld speichern können.

Der **Primärschlüssel** ist ein Feld oder eine Kombination von Feldern, dessen Wert jeden Datensatz der Tabelle eindeutig identifiziert.

- Der Primärschlüssel für die Tabelle wird folgendermassen festgelegt: Sie markieren alle zum Primärschlüssel gehörenden Felder (klicken Sie dazu auf das Feld ganz links; für mehrere Felder müssen Sie die Control-Taste gedrückt halten) und klicken dann auf das Symbol  (Schlüsselsymbol). In der Kunden-Tabelle ist die *Kunden-Nr* der Primärschlüssel.
- Um der Tabelle einen bestimmten Namen zu geben, wählen Sie im Menü **Datei** den Befehl **Speichern unter...**. Geben Sie den Namen *Kunden* ein und drücken Sie den **OK**-Knopf.
- Definieren Sie nun auch noch die Tabelle *Ausleihen*, in der gespeichert wird, welche Kassetten jede Kundin bzw. jeder Kunde ausgeliehen hat:



- In dieser Tabelle ist die *Kassetten-Nr* Primärschlüssel. Der Primärschlüssel ist in diesem Fall kein **AutoWert**. Sollten in die Tabelle zwei Datensätze mit derselben *Kassetten-Nr* eingegeben werden, so wird MS Access bei der Eingabe des zweiten Datensatzes eine Fehlermeldung ausgeben.

Ein Attribut einer Tabelle wird als **Fremdschlüssel** bezeichnet, wenn es in einer anderen Tabelle als Primärschlüssel auftritt.

- Das Attribut *Kunden-Nr* tritt auch in dieser Tabelle auf. Allerdings ist es hier kein Primärschlüssel, denn jede Kundin bzw. jeder Kunde kann mehrere Kassetten ausleihen. In diesem Fall wird *Kunden-Nr* als Fremdschlüssel bezeichnet, da es Primärschlüssel einer anderen Tabelle ist.
- Geben Sie für die *Kunden-Nr* unter *Eigenschaften* für *Feldgrösse* den Datentyp *Long Integer* (lange ganze Zahl) an, da das entsprechende Attribut in der Tabelle *Kunden* ein **AutoWert** ist. Der Datentyp **AutoWert** ist nämlich schlussendlich auch ein **Long Integer**.

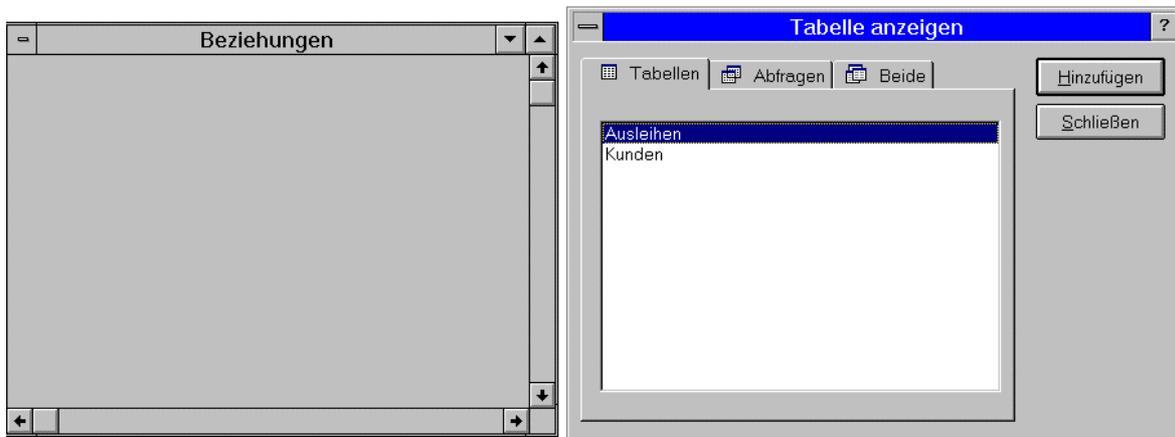
2.2 Festlegen von Beziehungen zwischen Tabellen

Beziehungen werden benutzt, um zu beschreiben, wie Informationen, die über verschiedene Tabellen verteilt sind, gegenseitig logisch verbunden sind. In vielen Fällen gibt es zwischen Tabellen eine sogenannte **1:n**-Beziehung, bei der der Primärschlüssel einer Tabelle als Fremdschlüssel einer zweiten Tabelle auftritt. Das bedeutet, dass es zu einem Datensatz einer Tabelle mehrere Datensätze in einer anderen Tabelle geben kann. Die 1:n-Beziehung wird auch **hierarchische Beziehung** genannt.

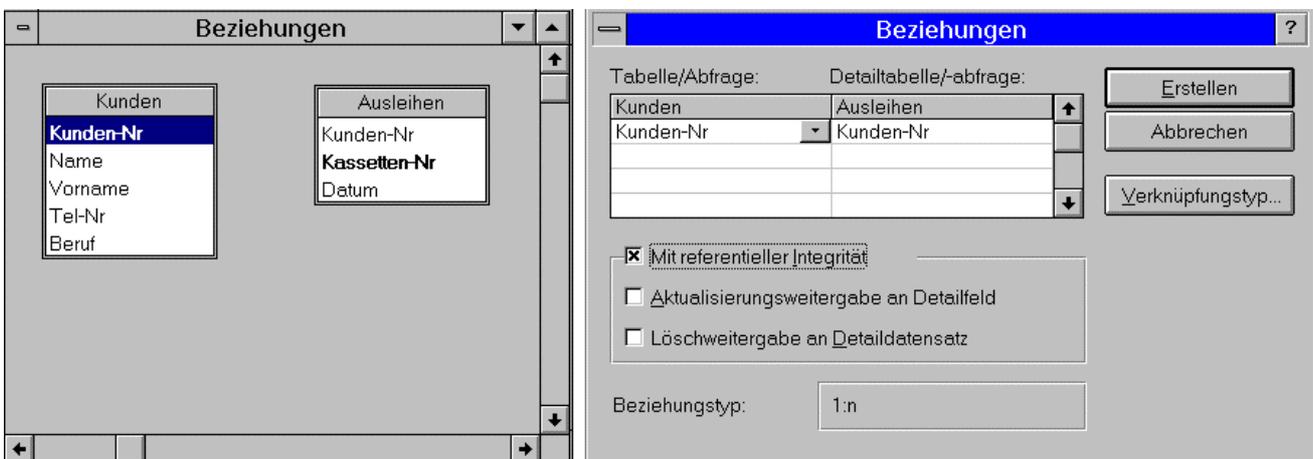
Beziehungen werden vom System benutzt, um Abfragen, die sich auf mehrere Tabellen beziehen, richtig zu beantworten. Ausserdem können sie dazu benutzt werden die sogenannte **referentielle Integrität** sicherzustellen. Diese sorgt dafür, dass zu Werten eines Fremdschlüssels in einer Tabelle, die auf Datensätze in einer anderen Tabelle verweisen, immer auch die entsprechenden Datensätze vorhanden sind. Kurz, dass Referenzen nicht ins „Leere“ zeigen.

In MS Access werden Beziehungen beschrieben, indem Verbindungen zwischen gleichen Feldern in unterschiedlichen Tabellen hergestellt werden. In unserem Beispiel gilt z.B. die 1:n Beziehung, dass ein Kunde/eine Kundin mehrere Videofilme ausgeliehen haben kann. Das heisst, zu **einem** Datensatz in der Tabelle *Kunden* mit der bestimmten *Kunden-Nr X* gibt es **mehrere** Datensätze (d.h. n) in der Tabelle *Ausleihen* mit derselben *Kunden-Nr X*.

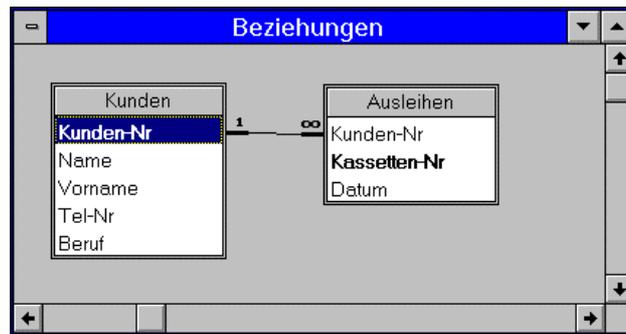
1. Dieses Wissen kann genutzt werden, um die referentielle Integrität der Datenbank zu gewährleisten: Es dürfen in der Tabelle *Ausleihen* mehrere Datensätze mit derselben *Kunden-Nr X* vorliegen (ein Kunde/eine Kundin hat mehrere Videokassetten ausgeliehen) und es muss genau ein Datensatz in der Tabelle *Kunden* vorliegen, der dieselbe *Kunden-Nr X* besitzt (der Kunde/die Kundin ist genau einmal registriert). Beziehungen können wie folgt definiert werden: Wählen Sie den Befehl **Beziehungen** aus dem Menü **Extras** oder wahlweise das Symbol . Es erscheint ein Fenster **Beziehungen** mit allen Tabellen, für die bereits Beziehungen definiert worden sind. Da Sie noch keine Beziehungen für diese Datenbank definiert haben, wird ein Fenster zum Hinzufügen von Tabellen automatisch geöffnet. Sollte sich dieses Fenster nicht automatisch öffnen, können Sie es mit dem Symbol  aktivieren.



2. Wählen Sie zuerst die Tabelle *Kunden* und drücken Sie **Hinzufügen**. Verfahren Sie mit der Tabelle *Ausleihen* ebenso.
3. Jetzt können Sie in dem Fenster **Beziehungen** die gewünschten Beziehungen zeichnen: Sie ziehen bei gedrückter Maustaste den Feldnamen *Kunden-Nr* der Tabelle *Kunden* auf den Feldnamen *Kunden-Nr* der Tabelle *Ausleihen* (vom Primärschlüssel zum Fremdschlüssel). Danach erscheint eine Dialogbox **Beziehungen**. Hier stellen Sie noch ein, dass MS Access automatisch die referentielle Integrität prüfen soll. Damit wird die 1:n Beziehung sichergestellt.



4. MS Access stellt die 1:n Beziehung dar, indem es die beiden betroffenen Felder durch eine Linie verknüpft, deren Enden mit 1 und mit ∞ beschriftet sind. Das Ergebnis sollte wie folgt aussehen:



5. Wollen Sie eine Beziehung löschen, können Sie dies durch einmaliges Anklicken einer Verbindungslinie und anschließendes Drücken der **Delete**-Taste. Wollen Sie eine Beziehung im Nachhinein verändern, erhalten Sie die Dialogbox **Beziehungen**, indem Sie auf die Verbindungslinie doppelklicken.
6. Sie speichern die Beziehungen mit dem Befehl Speichern im Menü Datei.

3 Eingabe von Daten

Die Eingabe von Daten sollte so komfortabel wie möglich gestaltet werden. Komfortabel heisst in diesem Zusammenhang, dass einfach verständlich ist, welche Daten in welche Felder einzugeben sind, und dass eine schnelle Eingabe möglich ist (z.B. ohne Verwendung der Maus, schnelles Springen von einem Datensatz zum anderen etc.). Zu diesem Zwecke bietet MS Access die Möglichkeit Formulare zu erstellen. Alternativ können die Daten direkt in Tabellenform eingegeben werden. Wir wollen jetzt einige Datensätze in die Tabellen *Kunden* und *Ausleihen* eingeben.

3.1 Eingabe mit Hilfe eines Formulars

1. Um Daten in die Tabelle *Kunden* einzugeben, erstellen wir ein Formular. Dazu markieren Sie die Tabelle *Kunden* im **Datenbank**-Fenster und drücken dann auf das Symbol  oder den Befehl **Formular** im Menü **Einfügen**. Wählen Sie **AutoFormular**. Es erscheint ein Formular, mit dem Sie nun Kundendatensätze eingeben können. Die *Kunden-Nr* enthält den vom System bestimmten Wert (**AutoWert**; die *Kunden-Nr* soll ja eindeutig sein und dafür sorgt das System). Sie können mit der **TAB**-Taste oder **Return**-Taste von einem Feld ins nächste springen.
2. Am unteren Ende des Formulars haben Sie eine Feldlaufleiste, mit der Sie zu existierenden Datensätzen springen können. Um den nächsten Datensatz einzugeben, drücken Sie das Symbol .

3. Um das Formular zu sichern, wählen Sie den Befehl **Speichern** im Menü **Datei**. MS Access fragt Sie nach einem Namen für das Formular. Wenn Sie später weitere Eingaben machen möchten, finden Sie das erstellte Formular auf der Karteikarte **Formular** im **Datenbank**-Fenster. Ihre eingegeben Datensätze werden von MS Access automatisch gespeichert. Geben Sie folgende Datensätze ein:

Kunden-Nr	Name	Vorname	Tel-Nr	Beruf
1	Meier	Alois	3682490	Bankler
2	Giger	Bruno	4729208	Maurer
3	Zobrist	Monika	2327282	Professorin

3.2 Eingabe in Tabellen-Form

Wenn die Datensätze einer Tabelle kurz sind, lohnt es sich nicht unbedingt, eigens ein Formular zu erstellen. Um Daten in die Tabelle *Ausleihen* einzugeben, doppelklicken Sie einfach auf die Tabelle *Ausleihen* im **Datenbank**-Fenster. Sie erhalten eine Tabelle, in der die Spalten mit den Feldnamen überschrieben sind. Hier können Sie jetzt einfach die Daten eingeben. Versuchen Sie, eine Kundennummer einzugeben, für die Sie vorher keinen Datensatz in der Tabelle *Kunden* erstellt haben. MS Access beschwert sich, dass die referentielle Integrität verletzt ist. Sie hatten ja vorher eine 1:n Beziehung zwischen den beiden Tabellen festgelegt, das heisst, die Kundinnen und Kunden müssen zuerst registriert werden, bevor sie eine oder mehrere Kassetten ausleihen können.

1. Geben Sie die folgenden Datensätze ein:

Kunden-Nr	Kassetten-Nr	Datum
1	1	02. Dez. 96
3	3	04. Dez. 96
2	12	01. Dez. 96
3	15	01. Dez. 96
0	0	

4 Abfragen

Was sind Abfragen?

Abfragen sind strukturierte Fragen, die Sie an die Datenbank richten, um nach bestimmten Informationen zu suchen:

- Suchen innerhalb einer Tabelle nach Daten, die bestimmten Kriterien entsprechen (z.B. Liste aller Kunden, die in Zürich wohnen)
- Suchen nach Daten, die in mehreren Tabellen gespeichert sind (z.B. Liste aller Kunden und deren momentan ausgeliehenen Kassetten)

Abfragen können gespeichert werden, damit man gleiche Abfragen nicht jedes Mal neu formulieren muss.

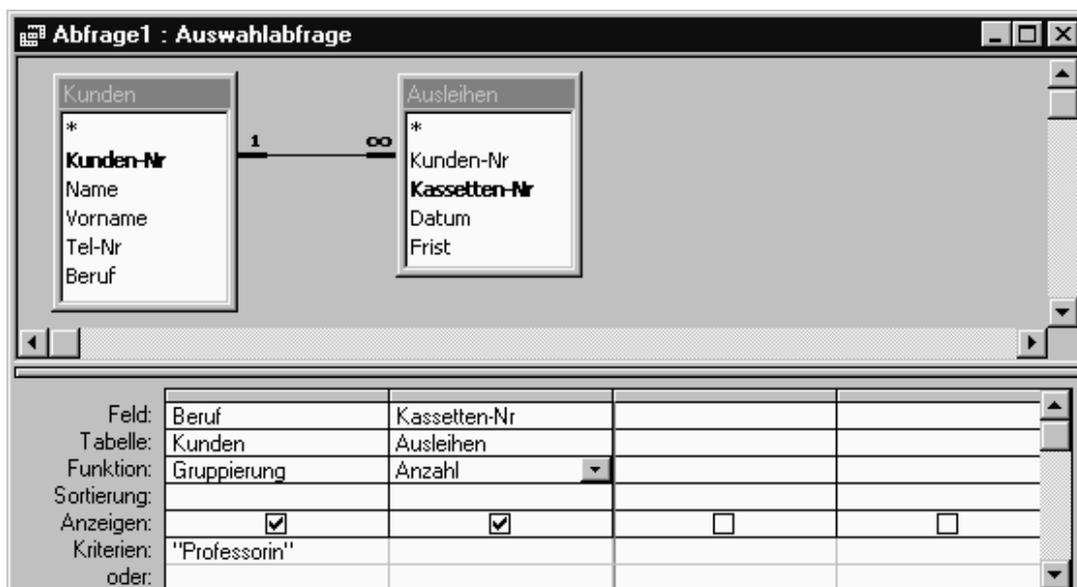
4.1 Eine Beispiel-Abfrage

Wir wollen nun zuerst zeigen, wie man eine Abfrage in MS Access stellt, um dann einige Charakteristiken von Abfragen zusammenzufassen. Als Beispiel nehmen wir die Abfrage "Wie viele Kassetten sind an Kundinnen ausgeliehen, die von Beruf Professorin sind?".

1. Abfragen sind in MS Access wie Tabellen ein zentraler Baustein. Klicken Sie deshalb im **Datenbank**-Fenster auf die Karteikarte **Abfragen**. Bis jetzt ist die Karteikarte leer. Wählen Sie den Befehl **Neu** und in dem erscheinenden Fenster **Entwurfsansicht**. Es erscheint nun eine Liste von Tabellennamen.
2. Wählen Sie die Namen aller Tabellen aus, die Sie für Ihre Abfrage benötigen. In diesem Fall sind das die Tabellen *Kunden* und *Ausleihen*. Klicken Sie nach jedem Namen auf **Hinzufügen**. Dann klicken Sie auf **Schliessen**.



3. Es öffnet sich das Fenster **Auswahlabfrage**, das die gewählten Tabellen und einen Bereich zum Entwerfen der Abfrage enthält. Der Abfrageentwurfsbereich ist zu Beginn natürlich noch leer. Aus den Listen im oberen Teil des Abfragefensters wählen Sie nun jene Feldnamen (Attribute) aus, welche in der Abfrage benötigt werden. Die Auswahl eines Feldes erfolgt, indem Sie auf den Feldnamen klicken und ihn in eine leere Zelle der Feld-Zeile im Abfrageentwurfsbereich ziehen. Ein Feld wird in einer Abfrage benötigt, wenn es in der Ausgabe erscheinen soll, oder wenn die Abfrage Bedingungen beschreibt, die sich auf dieses Feld beziehen. Soll das Feld in der Ausgabe sichtbar sein, so wird das Kreuz im Feld *Anzeigen* gesetzt. Wir ziehen also die Felder *Beruf* und *Kassetten-Nr* in die Feldzellen unten.
4. Jetzt muss noch die Bedingung eingetragen werden, die dafür sorgt, dass nur Professorinnen betrachtet werden. Dazu geben Sie in die mit **Kriterien** beschriftete Zeile unter *Beruf* die Zeichenkette *Professorin* ein.
5. Schliesslich soll in der Abfrage die Anzahl der zutreffenden Datensätze gezählt werden. Dazu müssen Sie eine geeignete Funktion angeben. Klicken Sie deshalb in die Spalte mit *Kassetten-Nr*. Klicken Sie dann auf das Funktionssymbol Σ in der Symbolleiste des Access-Fensters, um eine Funktion auszuwählen. Dann erscheint in der aktuellen Zeile der Begriff **Gruppierung**. Klicken Sie in die Zelle und dann auf das ∇ -Symbol, das erscheint. Jetzt können Sie die Funktion **Anzahl** wählen.



6. Um sich das Ergebnis der Abfrage in Tabellenform anzusehen, wählen Sie den Befehl **Datenblatt** im Menü **Ansicht** oder das Symbol . Das Ergebnis sieht dann in etwa so aus:

Abfrage1 : Auswahlabfrage	
Beruf	Anzahl von Kassetten-Nr
▶ Professorin	2

Datensatz: 1 von 1

7. Um zur Entwurfsmaske zurückzugelangen, wählen Sie den Befehl **Abfrageentwurf** im Menü **Ansicht**. Wenn Sie das Fenster schliessen, werden Sie gefragt, ob Sie die Abfrage speichern wollen. Wählen Sie **Nein**, wenn Sie sicher sind, dass Sie diesen Abfrageentwurf nicht mehr benötigen, also nie wieder diese oder eine ähnliche Frage stellen wollen. Meist jedoch werden Sie der Abfrage einen eindeutigen Namen geben, um Sie später wieder zu verwenden.

4.2 Aufbau von Abfragen

Wir wollen jetzt die Möglichkeiten für Abfragen und den typischen Aufbau einer Abfrage genauer analysieren. Für eine detailliertere Diskussion sei auf den Begleittext zur Vorlesung, Kapitel 12, verwiesen.

In der Vorlesung wurden die relationalen Operatoren SELECT, PROJECT und JOIN definiert:

- Eine **PROJECT**-Operation wählt aus der Menge der Felder einer Tabelle nur bestimmte Felder aus (spaltenorientiert).
- Eine **SELECT**-Operation selektiert mittels der Angabe von Kriterien aus einer Tabelle eine Menge von Datensätzen (zeilenorientiert).
- Eine **JOIN**-Operation fügt verschiedene Relationen zu einer einzigen zusammen. Es entsteht eine neue Relation, die aus den Feldern der beiden verbundenen Tabellen entsteht. Welche Datensätze der beiden Relationen zusammengefügt werden, wird durch die JOIN-Bedingung bestimmt.
- Eine **AGGREGATIONS**-Operation berechnet eine Funktion über eine Gruppe von Datensätzen, z.B. die Funktionen Mittelwert, Anzahl oder Summe.

All diese Operationen sind in der Beispiel-Abfrage enthalten:

- Wir haben nur die Felder *Beruf* und *Kassetten-Nr* ausgewählt (PROJECT). Die Projektion erhalten wir also, indem wir Felder, die wir in der Antwort sehen wollen, explizit auswählen (in den Abfrageentwurfsbereich ziehen) UND das Kreuz im Feld *Anzeigen* setzen.
- Wir haben eine Selektion (SELECT) durchgeführt, da wir die Bedingung gesetzt haben, dass wir nur Datensätze von Professorinnen berücksichtigen. Die Selektion wird also mit Hilfe der *Kriterien*-Zellen durchgeführt.
- Die JOIN-Operation wird quasi automatisch durchgeführt, da die beiden Tabellen mit Hilfe der Beziehung zwischen den *Kunden-Nr* verknüpft sind. Es werden die Datensätze der beiden Tabellen miteinander verbunden, bei denen die *Kunden-Nr* identisch ist (das genau ist die JOIN-Bedingung).
- Wir haben die AGGREGATION *Anzahl* berechnet über die Gruppe der Datensätze der Tabelle *Kunden*, die sich auf Professorinnen bezieht. Die Aggregation wird also mit Hilfe der *Funktion*-Zellen durchgeführt.

In das Kriterium-Feld können Ausdrücke eingegeben werden, die aus folgenden Bestandteilen bestehen:

- Zeichenketten z.B. "Meier", 34, #10.12.1996#
- Operatoren, z.B. +, -, UND, ODER, NICHT, WIE
- Konstanten, z.B. Ja, Nein, Wahr, Falsch, NULL, LEER
- Funktionen, z.B. Datum(), Summe, Maximum, ...

Beispiel: "Professorin" ODER "Vertreter" (zwei Zeichenketten, mit dem Operator "ODER" verbunden).

Sie können sich mit dem Befehl *SQL* des Menüs *Ansicht* den SQL-Ausdruck zur Ihrer Abfrage anschauen.

Abfragen werden in eine für das Datenbanksystem verständliche Sprache umgewandelt. **SQL (Structured Query Language)** ist eine derartige Sprache, die relativ einfach zu verwenden ist. Sie wird von fast allen relationalen Datenbanksystemen verstanden. Im Ergänzungsmodul (<http://food.ethz.ch/info1/>) wird SQL näher vorgestellt.

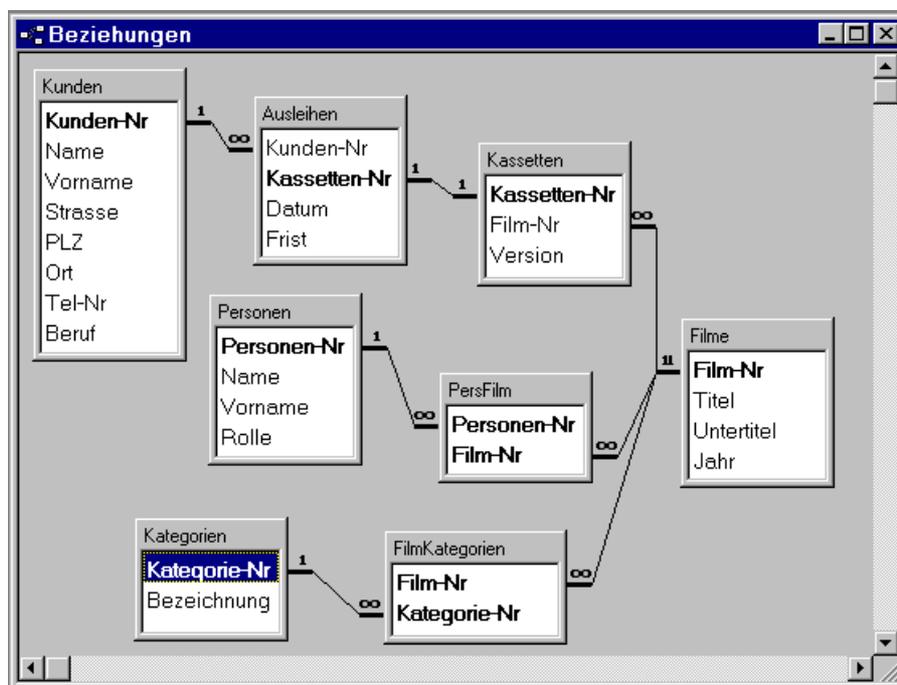
4.3 Eine existierende Datenbank abfragen

4.3.1 Die Videodatenbank

1. Wir haben Ihnen eine vollständige Videodatenbank erstellt, um komplexere Abfragen illustrieren zu können. Die Datenbank heisst **VIDEO2000.MDB** (bzw. **VIDEO97.MDB**) und kann von der Homepage der Vorlesung (<http://food.ethz.ch/info1/>) bezogen werden.
2. Nachdem Sie die Videodatenbank lokal auf Ihrem PC gespeichert haben, können Sie diese mit MS Access öffnen. Wenn Sie die Videodatenbank erfolgreich geöffnet haben, sollten Sie ungefähr folgendes Fenster vor sich haben:



3. Nun sollten Sie sich die folgenden Tabellen ansehen: *Ausleihen*, *Filme*, *FilmKategorien*, *Kassetten*, *Kategorien*, *Kunden*, *PersFilm* und *Personen*. Um die Beziehungen zwischen allen Tabellen anzusehen, wählen Sie den Befehl **Beziehungen** im Menü **Extras** oder das Symbol . Sie erhalten folgendes Diagramm – das **Schema** der Datenbank:



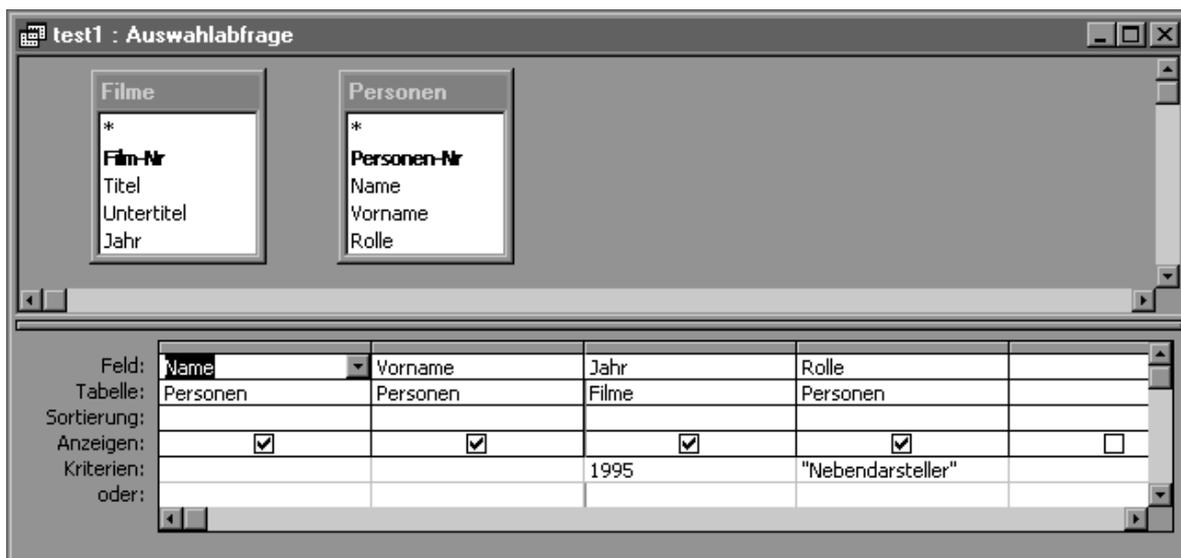
Dieses Schema zeigt nicht nur die Beziehungen zwischen den verschiedenen Tabellen auf, Sie können auch die einzelnen Attribute einer Tabelle sehen. Der Primärschlüssel jeder Tabelle besteht aus einem oder mehreren Attributen. Diese sind fett gedruckt. Bevor Sie mit den Abfragen anfangen, sollten Sie sich zuerst mit der Struktur der Tabellen, deren Beziehungen und den Daten, die in den Tabellen gespeichert sind, vertraut machen. Wie man ein Datenbankschema entwirft (konzeptioneller Entwurf), erfahren Sie am Ende des Ergänzungsmoduls. Es lohnt sich dieses Kapitel zu lesen, wenn man daran interessiert ist, warum gerade dieses Schema für den Zweck einer Videodatenbank dienlich ist. Ein detailliertes Verständnis des Datenbankschemas erleichtert es, Abfragen zu formulieren.

4.3.2 Beispiele von Abfragen

Wenn Sie das Tutorial bis hier durchgearbeitet haben, wissen Sie, wie man eine Abfrage stellen und das Ergebnis in verschiedenen Formen darstellen und speichern kann. Falls Sie noch Probleme mit den folgenden Beispielen haben sollten, können Sie entweder die entsprechenden Teile dieses Tutorials nochmals lesen, die Access Online-Hilfe benutzen oder die Assistierenden fragen.

1. Abfrage 1: Welche Nebendarsteller (Name, Vorname) sind in Filmen von 1995 aufgetreten?

Aus der Beziehungsübersicht der Videodatenbank ist schnell ersichtlich, dass die Information über das Jahr eines Filmes in der Tabelle *Filme* gespeichert ist. Also brauchen Sie die Tabelle *Filme* für die Abfrage. Welche Tabelle brauchen Sie noch? "Nebendarsteller" ist eine Rolle einer Person, die in der Tabelle *Personen* gespeichert ist, weswegen auch die Tabelle *Personen* für die Abfrage notwendig ist. Genügen diese Relationen bereits? Um diese Frage zu beantworten, geben Sie die Kriterien "1995" und "Nebendarsteller" im Abfragefenster ein, speichern Sie Ihre Abfrage unter einem Namen, z.B. "test1".



2. Führen Sie die Abfrage aus. Die Ergebnistabelle sieht etwa so aus:

	Name	Vorname	Jahr	Rolle
	Field	Sally	1995	Nebendarsteller
	Field	Sally	1995	Nebendarsteller
	Field	Sally	1995	Nebendarsteller
	Field	Sally	1995	Nebendarsteller
	Field	Sally	1995	Nebendarsteller
	Thompson	Lea	1995	Nebendarsteller
	Thompson	Lea	1995	Nebendarsteller
	Thompson	Lea	1995	Nebendarsteller
	Thompson	Lea	1995	Nebendarsteller
	Thompson	Lea	1995	Nebendarsteller
	Spacey	Kevin	1995	Nebendarsteller
	Spacey	Kevin	1995	Nebendarsteller
	Spacey	Kevin	1995	Nebendarsteller

Wenn Sie die Ergebnistabelle und die Tabelle *Personen* vergleichen, bemerken Sie vielleicht, dass das Resultat nicht stimmen kann. Es sind alle Nebendarsteller sogar mehrmals aufgeführt, obwohl nur diejenigen Nebendarsteller gezeigt werden sollten, die 1995 in mindestens einem Film aufgetreten sind.

Wo liegt das Problem? Wenn Sie die Beziehungsübersicht anschauen, entdecken Sie, dass die Tabellen *Personen* und *Filme* nur über die Tabelle *PersFilm* verbunden sind. Das heisst, wenn Sie eine Abfrage über die beiden Tabellen machen, müssen Sie die Tabelle *PersFilm* auch in die Abfrage miteinbeziehen. Andernfalls geht die Verbindung zwischen den beiden Tabellen *Personen* und *Filme* verloren. Öffnen

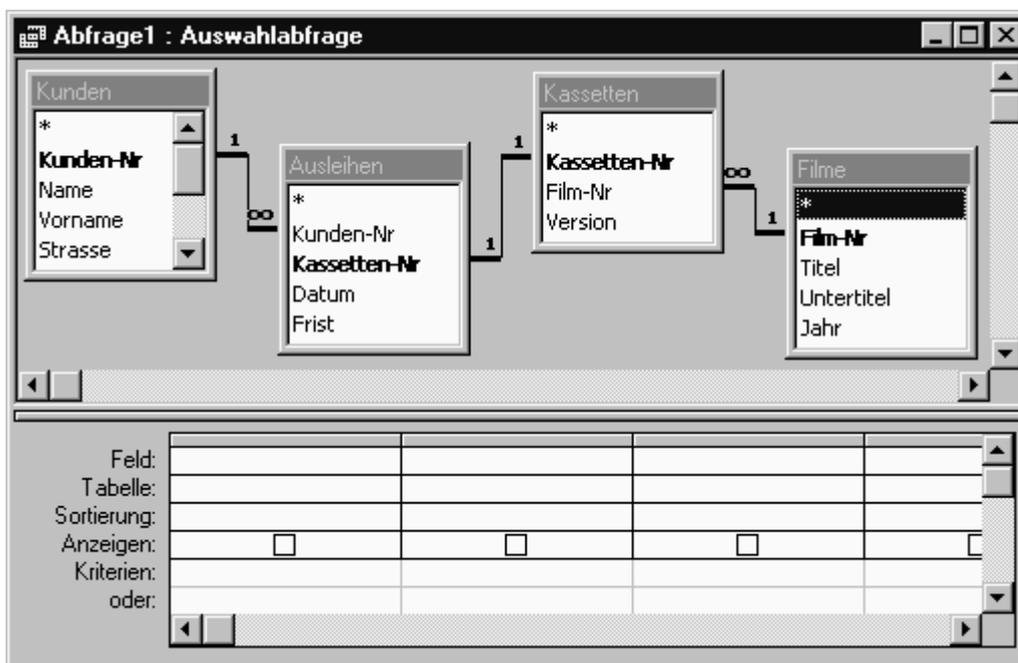
Sie daher die Abfrage *test1* im Entwurfswfenster (markieren Sie die Abfrage *test1* und klicken Sie auf den **Entwurf**-Knopf). Klicken Sie dann die rechte Maustaste neben den beiden Tabellen (*Personen* und *Filme*) und wählen Sie den Befehl **Tabelle Anzeigen**. Sie können jetzt die Tabelle *PersFilm* von der Tabellenliste in die Abfrage einfügen. Beachten Sie, dass die beiden bisherigen Tabellen *Personen* und *Filme* sofort durch 2 Linien mit der Tabelle *PersFilm* verbunden werden. Nun führen Sie die Abfrage *test1* erneut aus. Das Ergebnis ist jetzt korrekt:

Name	Vorname	Jahr	Rolle
Spacey	Kevin	1995	Nebendarsteller
Sorvino	Mira	1995	Nebendarsteller

Dieses Beispiel zeigt, dass eine Abfrage über mehrere Tabellen im Allgemeinen auch die Tabellen mit einschliesst, welche die Tabellen mit den eigentlichen Daten miteinander verbinden. Eine Tabelle darf in der Regel nur dann allein vorkommen, wenn die Abfrage nur diese eine Tabelle betrifft.

3. Abfrage 2: Welche Kunden (Name, Adresse) haben den Film “Leaving Las Vegas” ausgeliehen? Wann müssen sie ihn zurückgeben?

Diese beiden Fragen kann man praktisch durch eine *einzig*e Abfrage lösen. Zuerst sollten Sie sich überlegen, welche Tabellen Sie für diese Abfrage brauchen. Es ist klar, dass Sie die Tabellen *Filme* (wegen des Filmtitels “Leaving Las Vegas”), *Kunden* (wegen Kundenname und Adresse) und *Ausleihen* (wegen der Rückgabefrist) unbedingt brauchen. Öffnen Sie eine neue Abfrage und fügen Sie die Tabellen *Kunden*, *Ausleihen*, und *Filme* ein. Wie Sie sehen, sind die 3 Tabellen noch nicht alle miteinander verbunden. Die Tabelle *Filme* ist noch isoliert. Um diese Lücke zu schliessen, brauchen Sie welche Tabelle? Wenn Sie die Antwort nicht wissen, sollten Sie die Beziehungsübersicht nochmals konsultieren: die Tabelle *Kassette* verbindet die Tabellen *Ausleihen* und *Filme*. (weil ein Kunde nämlich nur die Kassette eines Films ausleihen kann und nicht den Film selbst). Nach dem Einfügen von *Kassetten*, sollte das Auswahlabfragefenster etwa so aussehen:



Wählen Sie nun jene Feldnamen aus, welche in der Abfrage benötigt werden. Von der Fragestellung wissen Sie, dass der Kundenname, die Kundenadresse und die Ausleihfrist gefragt sind. Geben Sie diese Feldnamen ein und tragen Sie die Bedingung der Abfrage ein (Titel = “Leaving Las Vegas”). Die Ausführung dieser Abfrage ergibt folgendes Ergebnis:

Abfrage1 : Auswahlabfrage							
	Name	Vorname	Strasse	PLZ	Ort	Frist	Titel
▶	Bicker	Stefan	Adlisbergstr. 22	8044	Zurich	15. Dez. 97	Leaving Las Vegas
*							

Datensatz: 1 von 1

Nach diesen ersten beiden Abfragen, sollten Sie nun in der Lage sein, einige interessante Fragen selbst zu stellen und von der Datenbank beantworten zu lassen. Versuchen Sie beispielsweise, folgende Fragen zu beantworten:

4. **Abfrage 3:** In wie vielen Filmen (Anzahl) hat jeder Hauptdarsteller mitgewirkt?

Tip: Bei dieser Abfrage müssen Sie **Funktionen** einblenden, damit die Anzahl berechnet wird.

5. **Abfrage 4:** Welche Personen (Name und Vorname von Schauspielern, Regisseuren etc.) wirken bei Filmen mit, die ausgeliehen sind und welche Rollen haben diese Personen?

Abfragen wiederverwenden

Wir haben am Anfang dieses Kapitels gesagt, dass Abfragen gespeichert werden können. Es gibt verschiedene Gründe, wieso man eine Abfrage wiederverwenden möchte, z.B.:

- Sie wollen die Frage zu einem späteren Zeitpunkt wiederholen, wenn sich der Datenbestand geändert hat.
- Sie möchten eine ähnliche Frage stellen und diesen Entwurf als Schablone benutzen.
- Die Antwort auf eine Abfrage kann wieder als eine Tabelle angesehen werden. Da diese nicht modifiziert, sondern nur gelesen werden kann, wird sie als **Sicht** bezeichnet (engl. View). Diese Sicht wird automatisch immer dann aktualisiert, wenn sich die Tabellen ändern, auf die sich die Sicht bezieht.
- Die Antwort auf eine Abfrage kann als Grundlage für neue Abfragen dienen.

Abfragen können sich in MS Access also auf Tabellen der Datenbank und/oder auf bereits existierende Abfragen beziehen. Damit ist es möglich, komplizierte Abfragen in *Teilabfragen* zu zerlegen.

5 Testaufgabe

Das Ziel der Testaufgabe ist es, die bis jetzt beschriebenen Konzepte selbst noch einmal nachzuvollziehen. Wir haben Ihnen eine Videodatenbank vorgestellt, um wichtige Datenbankprinzipien kennenzulernen. Wenden Sie nun die gelernten Konzepte auf die Daten des Landesforstinventars an, die Sie folgendermassen bearbeiten sollen:

- Neue Datensätze erfassen
- Abfragen erstellen
- Ergebnisse erfassen
- Erweitern des Schemas der Landesforstinventars-Datenbank

Abgabe:

Zeigen Sie einer Assistentin oder einem Assistenten Ihre Abfragen und deren Ergebnisse bzw. Erweiterungen am Bildschirm. Die Aufgaben gelten als gelöst, wenn Sie der Assistentin oder dem Assistenten den Aufbau und die Funktionsweise der Abfragen erklären können. Abgaben per e-mail können leider **nicht** entgegengenommen werden.

5.1 Die Datenbank: Das Landesforstinventar

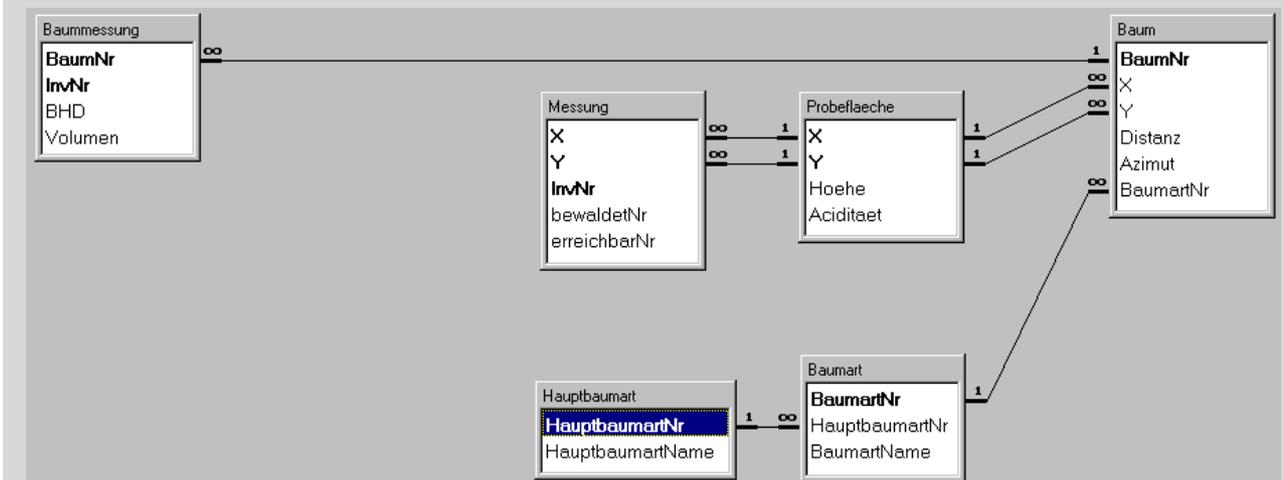
Zunächst ist es wichtig, den Aufbau der Datenbank und die Attribute der Relationen zu verstehen. Das Forstinventar erhebt stichprobenartig Daten über den Zustand des schweizerischen Waldes. Die Stichproben werden in zuvor festgelegten Gebieten, die im Folgenden als *Probeflächen* bezeichnet werden, durchgeführt. Die Probeflächen sind kreisförmig und ihr Mittelpunkt liegt auf einem Kreuzungspunkt eines Gitternetzes mit einer Maschenweite von 1 km, das die gesamte Schweiz überspannt. Die Gitternetzpunkte sind in einem kartesischen Koordinatensystem mit X (östliche Länge) und Y (nördliche Breite) beschrieben.

Eine Probefläche besteht aus zwei konzentrischen Kreisflächen mit einem Inhalt von 200 m^2 (2a) bzw. 500 m^2 (5a). Auf der Probefläche werden alle Bäume aufgenommen, die einen Minstdurchmesser in einer Stammhöhe von 1,3 m aufweisen. Dieser gemessene Durchmesser wird auch als Brusthöhdendurchmesser (BHD) bezeichnet. Im 5a-Kreis werden alle Bäume mit mindestens 36 cm BHD und im 2a-Kreis alle Bäume mit mindestens 12 cm BHD aufgenommen. Jede Messung gehört zu genau einem Inventar.

Zu jedem aufgenommenen Baum wird die Baumart in einer zweistufigen Klassifikation mit Baumart und Hauptbaumart erfasst.

Schließlich wird der Standort des Baumes in der Probefläche genau vermessen, indem die Distanz zum Mittelpunkt der Probefläche bestimmt wird und die Richtung durch eine Winkelabweichung von der Nordrichtung angegeben wird. Der Winkel wird als Azimut bezeichnet. Die nachfolgende Zeichnung verdeutlicht das Messverfahren. Distanz und Azimut bilden die letzten beiden Werte eines Inventar-Datensatzes.

Die restlichen Werte bzw. Attribute und deren Aufteilung auf Relationen entnehmen Sie bitte dem nachfolgendem Schema.



Die Bedeutung der einzelnen Attribute und deren Einheiten bzw. Wertebereiche sind im Folgenden in tabellarischer Form zusammengefasst (gemäss den Konventionen des Relationen-Modells):

Probeflaeche (X, Y, Hoehe, Aciditaet)		Einheit
X	X-Koordinate des Probeflächenzentrums (östl. Länge)	100 m
Y	Y-Koordinate des Probeflächenzentrums (nördl. Breite)	100 m
Aciditaet	Acidität des Muttergesteins (0: keine Angabe, 1: sauer, 2: basisch)	0 – 2
Hoehe	Höhe der Probefläche über Meeresspiegel, aus Luftbild	m

Messung (X, Y, InvNr, bewaldetNr, erreichbarNr)		Einheit
X	X-Koordinate des Probeflächenzentrums (östl. Länge)	100 m
Y	Y-Koordinate des Probeflächenzentrums (nördl. Breite)	100 m
InvNr	Inventarbezeichnung von Landesforst- oder Waldschadensinventaren auf nationaler und kantonaler Ebene	0, 1, ...
bewaldetNr	0: nicht bewaldet, 1: bewaldet, Information stammt aus Luftbild	0 – 1
erreichbarNr	0: unbekannt, 1: unerreichbar, 2: teilweise erreichbar, 3: normal erreichbar, 4: Gebueschwald	0 – 4

Baum (BaumNr, X, Y, Distanz, Azimut)		Einheit
BaumNr	Absolute Baumnummer, für alle Bäume der Schweiz	Number
X	X-Koordinate des entsprechenden Probeflächenzentrums (östl. Länge)	100 m
Y	Y-Koordinate des entsprechenden Probeflächenzentrums (nördl. Breite)	100 m
Distanz	Distanz eines Baumes zu seinem Probeflächenzentrum (Polarkoordinaten)	dm
Azimut	Winkel zwischen dem Baum und der vom Probeflächenzentrum ausgehenden Nordrichtung (Polarkoordinaten)	1-400 (Neugrad)
BaumartNr	Kodierung verschiedener Laub- und Nadelbäume sowie Sträucher	1 – 99

Baummessung (BaumNr, InvNr, BHD, VolumenLiter)		Einheit
BaumNr	Absolute Baumnummer, für alle Bäume der Schweiz	Number
InvNr	Inventarbezeichnung von Landesforst- oder Waldschadensinventaren auf nationaler und kantonaler Ebene	0, 1, ...
BHD	Brusthöhendurchmesser := Baumdurchmesser in 1,3 m Höhe	cm
Volumen	Nutzholzvolumen eines Baumes, bestimmt aus einer Berechnungsformel aufgrund verschiedener Messwerte	Liter(dm ³)

Baumart (BaumartNr, HauptbaumartNr, BaumartName)		Einheit
BaumartNr	Kodierung verschiedener Laub- und Nadelbäume sowie Sträucher	1 – 99
HauptbaumartNr	Kodierung von je 6 Hauptbaumarten verschiedener Laub- und Nadelbäume	0 – 12
BaumartName	Botanische Bezeichnung der erfassten Baumarten mit "Gattung Art" (z.B. "Fagus sylvatica"), gehört zu genau einer Hauptbaumart	[2 Worte]

Hauptbaumart (HauptbaumartNr, BaumartName)		Einheit
HauptbaumartNr	Kodierung von je 6 Hauptbaumarten verschiedener Laub- und Nadelbäume	0 – 12
HauptbaumartName	Name der Hauptbaumart (z.B. 'Buche')	[1-2 Worte]

5.2 Aufgabe 1: Einlesen einer existierenden Datenbank

Unsere Datenbank heisst **LFI2000.MDB** (bzw. **LFI97.MDB**) und kann via WWW auf der Homepage der Vorlesung unter <http://food.ethz.ch/info1/> geholt werden. Speichern Sie die Datenbank lokal ab. Bevor Sie damit beginnen, die Aufgaben zu lösen, schauen Sie sich zuerst die Struktur aller Tabellen des Landesforstinventars und deren Beziehungen an, sowie die Daten, die in den Tabellen gespeichert sind.

5.3 Aufgabe 2: Abfragen stellen

Formulieren Sie folgende Abfragen an die Datenbank:

- Erstellen Sie eine Liste aller Hauptbaumartnamen.
- Erstellen Sie eine Liste aller Baumartnamen der Hauptbaumart "FOEHRE".
- Erstellen Sie eine Liste aller Baumnummern und ihrer zugehörigen Baum- und Hauptbaumartnamen, die sich in Probeflächen befinden, die höher als 1800 m liegen.
- Geben Sie die Baumart (Namen) eines jeden Baumes an, der in einer normal erreichbaren (erreichbarNr = 3) Probefläche liegt.

Vermeiden Sie in Aufgabe d) die mehrmalige Nennung gleicher Datensätze in der Ergebnisrelation. Klicken Sie dazu in der Entwurfsdarstellung der Abfrage zuerst mit der linken und dann mit der rechten Maustaste auf die graue Fläche in der oberen Hälfte des Abfragefensters. Wählen sie im sich öffnenden Kontextmenü den Befehl **Eigenschaften** und setzen Sie dann die Abfrageeigenschaft **Keine Duplikate** auf den Wert **Ja**.

5.4 Aufgabe 3: Erweiterung der Datenbank

Relation *Inventar*

Die Tabellen *Messung* und *Baummessung* haben ein Attribut *InvNr*. Leider wurde beim Erstellen der Datenbank vergessen, die Relation *Inventar* zu erstellen. Holen Sie dies nun nach. Die Tabelle hat die Attribute *InvNr*, *InvName* und *Jahr*.

Tragen Sie nun die Datensätze (0, LFI1, 1995) und (1, LFI2, 1996) in die neue Tabelle ein und vergessen Sie nicht, auch die Beziehungen der neuen Tabelle zu *Messung* und *Baummessung* festzulegen!

Relation *Erreichbar*

In der Relation *Messung* bezeichnet das Attribut *erreichbarNr* den „Grad der Erreichbarkeit“ eines Baumes mittels einer Zahl von 0 bis 4. Um die Bedeutung noch als Text in der Datenbank zu speichern, erstellen Sie eine neuen Relation *Erreichbar*, die neben dem Primärschlüssel-Attribut noch das Text-Attribut *erreichbarName* enthält. Füllen Sie die Tabelle mit den notwendigen Datensätzen.

Abfragen auf der erweiterten Datenbank

Erstellen Sie folgende Abfragen auf der erweiterten Datenbank:

- Welche Baumarten (*BaumartName*) waren in Inventuren ab 1996 bei Baummessungen beteiligt?
- Welche Erreichbarkeit (*Grad*) hatten alle Bäume (*BaumNr*) bei allen Messungen bis zu einer Höhe von 2000m? (Vermeiden Sie die mehrmalige Nennung gleicher Datensätze).

Ergänzungsmodul

(Zugreifbar über die Homepage der Vorlesung: <http://food.ethz.ch/info1/>)

Ziele

- Daten aus MS Excel, ändern Datenbankprogrammen oder Textdateien in MS Access importieren
- Berechnungen in Abfragen durchführen
- Einen Bericht erstellen
- Schrittweise eine Datenbank entwerfen
- SQL (Structured Query Language) kennenlernen

Inhalt

6	Datenimport und Berechnungen in Abfragen	21
6.1	Theorie: Daten einbinden versus importieren	21
6.1.1	Daten einbinden	21
6.1.2	Daten importieren	21
6.2	Praxis: Die Ernährungssituation in der Schweiz	21
6.2.1	Problemstellung	22
6.3	Ausgangsdaten	22
6.4	Datenimport	22
6.4.1	Import aus MS Excel (Datei <i>Naehrwertberechnung.xls</i>)	23
6.4.2	Import aus Textdateien (Datei <i>LMGruppen.txt</i>)	26
6.4.3	Import aus anderen Datenbanken (Datei <i>NWT.dbf</i>)	27
6.5	Theorie: Berechnungen in Abfragen	27
6.5.1	Datenbank versus Tabellenkalkulation	27
6.6	Praxis: Berechnungen in Abfragen über die importierten Daten	28
6.6.1	Bereinigte Nährwert-Tabelle	28
6.6.2	Berechnung der Nährstoffaufnahme pro Tag	28
6.6.3	Gruppieren nach Lebensmittelgruppen	29
6.6.4	Berechnung der Energie pro Lebensmittelgruppe	29
7	Berichte	29
7.1	Erstellen einer berichtsbezogenen Abfrage	30
7.2	Grobentwurf eines Berichtes mit dem Berichtsassistenten	31
7.3	Abschliessendes Design	33
8	Datenbankentwurf	34
8.1	Konzeptioneller Entwurf	35
8.2	Relationaler Entwurf	36
8.3	Fehlerquellen	37
8.4	Tips für die Datenmodellierung	38
8.5	Abfragen mit SQL	38

6 Datenimport und Berechnungen in Abfragen

Ziel dieser Ergänzungsübung ist es, die verschiedenen Unterstützungsmöglichkeiten von MS Access auszuprobieren, um Daten in anderen Formaten (z.B. aus MS Excel, Textdateien oder einer andern Datenbank) in eine neue Datenbank einzulesen. Zusätzlich werden in dieser Ergänzungsübung weitere Datenbanktechniken wie geschachtelte Abfragen und Funktionen in Abfragen vorgestellt. Ein kurzer Theorieblock zu Beginn orientiert über die wichtigsten Unterschiede zwischen dem Einbinden von externen Daten und dem direkten Importieren.

6.1 Theorie: Daten einbinden versus importieren

Bei der Übernahme von Daten in eine Datenbank unterscheidet man zwischen *Import* und *Einbindung*.

6.1.1 Daten einbinden

Daten einbinden bedeutet, dass externe Datenquellen (z.B. Daten aus dem Internet, aus MS Excel oder auch einzelne Tabellen aus einer andern Datenbank, z.B. dBASE) mit einer Datenbank (z.B. MS Access) verbunden werden. Dabei werden die Daten nicht kopiert. Vielmehr wird eine direkte **Verbindung** zu den Originaldaten erstellt. Dies geschieht dadurch, dass die Datenbank die entsprechenden Dateinamen und – Pfade oder die Internetadressen verwaltet.

Vorteile:

- Automatische Aktualisierung bei Änderung der externen Originaldaten.
- Geringerer Speicherbedarf für die Datenbank, da die eingebundenen Daten nur im Original gespeichert werden müssen.

Nachteile:

Beim Löschen des Originals werden auch die entsprechenden Daten in der Datenbank „gelöscht“.

- Wenn man die Datenbank auf einen anderen PC kopieren möchte, muss man die externen Daten separat kopieren (nicht so bei Internetadressen, wenn der zweite PC ebenfalls ans Internet angeschlossen ist).

Beispiel des Einbindens von Daten: Siehe Praxis 2, Abschnitt 7.3 "Excel's Web Queries". Dort wurde am Beispiel von Börsendaten gezeigt, wie Daten online aus dem Internet geholt und in MS Excel importiert werden können, sodass sie automatisch aktualisiert werden.

6.1.2 Daten importieren

Daten importieren bedeutet ebenfalls, dass externe Daten in eine Datenbank (z.B. MS Access) übernommen werden. Dabei werden die Originaldaten aber in die Datenbank kopiert, d.h. es wird ein **lokales Duplikat** der Daten erstellt.

Vorteile:

- Die Datenbank speichert alle Daten, d.h. das Löschen der externen Originaldaten hat keine Auswirkungen auf die Datenbank.
- Die Datenbank besteht nur aus einer Datei, kann also einfacher kopiert und transportiert werden.

Nachteile:

- Werden die Originaldaten geändert, müssen die Daten erneut in die Datenbank importiert werden.

Beispiel des Importierens von Daten: Siehe nachfolgender Abschnitt "6.3 Praxis: Importübung". Dort wird am Beispiel von Schweizer Nährwertdaten der Import aus verschiedenen Dateiformaten in MS Access Schritt für Schritt erklärt.

6.2 Praxis: Die Ernährungssituation in der Schweiz

Um die wichtigsten Importfunktionen von MS Access zu diskutieren, verwenden wir ein reales Beispiel aus dem Gebiet der Lebensmittelwissenschaften (konkret: Nährwertdaten). Die in diesem Beispiel verwendeten Daten wurden vorher *normalisiert*, d.h. die Relationen wurden so angelegt, dass innerhalb einer Tabelle Daten nicht unnötig wiederholt vorkommen. So kann der Schwerpunkt der Übung auf den Datenimport aus

verschiedenen Datenformaten in MS Access gelegt werden. Im Berufsleben kann es jedoch vorkommen, dass Sie in Ihrem Fachgebiet auch mit in grossen Tabellen verpackten nicht relationalen Daten konfrontiert werden, die Sie zuerst einmal normalisieren müssen (siehe dazu Begleittext zur Vorlesung Abschnitt 12.3.5). Die Umwandlung von komplexen Datenstrukturen in ein relationales Modell bringt oft auch Inkonsistenzen in den Daten oder im ursprünglichen Datenmodell zutage.

Datenquelle

Schlotke F., Sieber R. *Berechnung des Verbrauchs an Nahrungsenergie, Energieträgern, Nahrungsfasern, Vitaminen, Mineralstoffen und Spurenelementen*. 4. Schweizerischer Ernährungsbericht, Bundesamt für Gesundheitswesen, 1998, 18-27. Von den Autoren freundlicherweise zur Verfügung gestellt.

6.2.1 Problemstellung

In verschiedenen Dateiformaten stehen Informationen über den schweizerischen Nahrungsmittelverbrauch und über den Nährwert verschiedener Nahrungsmittel zur Verfügung. Ihre Aufgabe besteht darin, mit Hilfe der Datenbankfunktionen von MS Access zu berechnen, wieviel die einzelnen Lebensmittel zur Energieversorgung durch Proteine, Kohlenhydrate, Fette und Alkohol beitragen. Ausserdem sollen Sie berechnen, wie Gross der Anteil jeder Lebensmittelgruppe in unserer Ernährung ist. Eine solche Auflistung kann für die Früherkennung von Unter- bzw. Überversorgung mit einzelnen Nahrungs-Grundbausteinen nützlich sein.

6.3 Ausgangsdaten

Auf der Website der Vorlesung unter: <http://food.ethz.ch/info1/> befinden sich folgende drei Dateien, welche für die Ergänzungsaufgabe benötigt werden:

(**Hinweis:** Falls Sie zuhause arbeiten und dort nicht über einen Internetzugang verfügen, sollten Sie sich diese drei Dateien vorher an der ETH auf eine Diskette kopieren.)

Naehrwertberechnung.xls ist eine MS Excel-Mappe mit den beiden Tabellen *Verbrauch* und *NahrungsmittelNWT*. Das Tabellenkalkulationsprogramm MS Excel kennen Sie ja bereits aus Übung 2. Die Tabelle *Verbrauch* enthält 148 Datensätze über den Verbrauch der einzelnen Nahrungsmittel in der Schweiz in Kilogramm pro Person und Jahr. Die hierzu verwendeten Daten stammen aus den Jahren 1994-1995 und sind in der im Abschnitt 6.9 *Datenquelle* aufgelisteten Quelle näher beschrieben. Die Tabelle *NahrungsmittelNWT* (186 Datensätze) enthält zu jedem Lebensmittel eine oder mehrere Identifikationsnummern, welche als Verknüpfung zum Nährwert des entsprechenden Lebensmittels dienen.

LMGruppen.txt ist eine Textdatei, welche zu jeder der 13 Lebensmittel-Hauptgruppen den entsprechenden Gruppennamen enthält.

NWT.dbf ist eine dBase4-Datei (dBase4 ist ein mit Access vergleichbares Datenbankprogramm eines anderen Herstellers). Diese Datei enthält zu jedem der 186 anhand ihrer Identifikationsnummern aufgelisteten Lebensmitteln den entsprechenden Gehalt an Kohlenhydraten, Proteinen, Fetten und Alkohol in Gramm pro 100 Gramm (also in Gewichts-Prozenten).

6.4 Datenimport

1. Downloaden Sie die drei obigen Dateien von der Homepage der Vorlesung (die Excel-Mappe „*Naehrwertberechnung.xls*“ enthält zwei Tabellen) unter: <http://food.inf.ethz.ch/info1/>. Importieren Sie anschliessend die obigen drei Dateien in vier Tabellen in eine neu erstellte Access-Datenbank namens "*Naehrwertberechnung.mdb*". Gehen Sie dazu so wie in den Abschnitten 6.4.1 bis 6.4.3 beschrieben vor.

6.4.1 Import aus MS Excel (Datei *Naehrwertberechnung.xls*)

1. Starten Sie den Import über den Menüpunkt **Importieren** Dieser befindet sich im Datei-Menü unter **Externe Daten**. Im **Importieren** Dialog wählen Sie den der zu importierenden Datei entsprechenden Dateityp, hier **Microsoft Excel (*.xls)** und starten den Importassistenten mit einem Doppelklick auf die zu importierende Datei.
2. Wählen Sie auf der ersten Seite des Assistenten das zu importierende Tabellenblatt. Wählen Sie *Verbrauch* und klicken Sie auf **Weiter**.



3. Auf der zweiten Seite des Assistenten können Sie festlegen, ob die erste Zeile die Spaltenüberschriften enthält. Da dies der Fall ist, sollten Sie das entsprechende Kästchen markieren und auf **Weiter** klicken.



4. Auf der dritten Seite können Sie wählen wo Sie die importierten Daten speichern wollen. Wählen Sie **In einer neuen Tabelle** und klicken Sie auf **Weiter**.

Wo möchten Sie Ihre Daten speichern? Sie können sie entweder in einer neuen oder in einer bereits bestehenden Tabelle speichern.

Die Daten speichern...

In einer neuen Tabelle

In einer bestehenden Tabelle:

	GruppenNr	Nahrungsmittelarten	Verbrauch
1	1	Hartweizen (in Mehl ber.)	11.95
2	1	Weichweizen (in Mehl ber.)	51.71
3	1	Dinkel (in Mehl berechnet)	0.58
4	1	Roggen (in Mehl berechnet)	1.26
5	1	Mischel (in Mehl berechnet)	0.44
6	1	Gerste (in Mehl berechnet)	0.73
7	1	Hafer (in Mehl berechnet)	1.01
8	1	Mais (in Mehl berechnet)	1.51
9	1	Reis, geschält	4.83
10	2	Kartoffeln, -produkte	47.08

Abbrechen < Zurück Weiter > Fertigstellen

5. Auf der vierten Seite können Sie zusätzlich Feldoptionen angeben. Dies ist aber in unserem Beispiel nicht notwendig. Klicken Sie also direkt auf **Weiter**.

Sie können Informationen zu jedem zu importierenden Feld angeben. Wählen Sie dazu Felder aus dem unten stehenden Bereich aus, und bearbeiten Sie dann die Feldinformationen im Bereich 'Feldoptionen'.

Feldoptionen

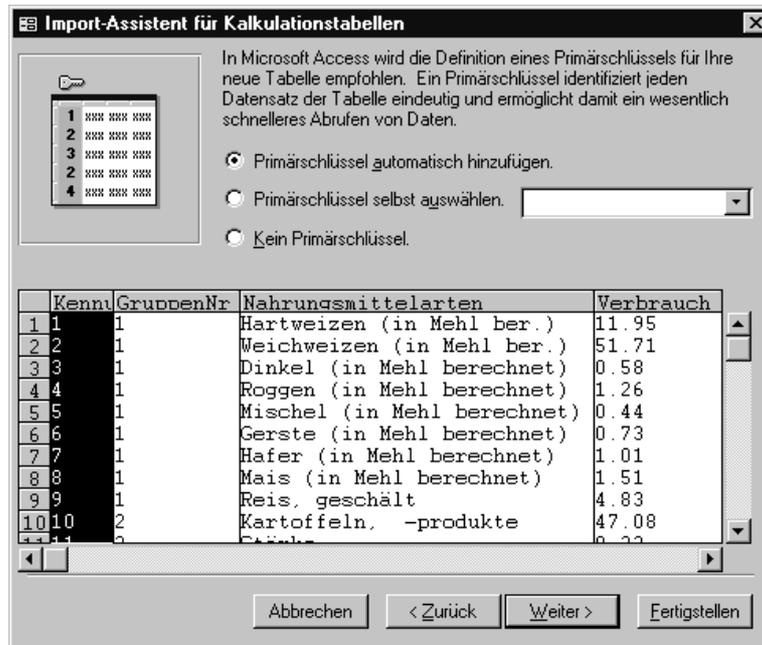
Feldname: Datentyp:

Indiziert: Feld nicht importieren (Überspringen)

	GruppenNr	Nahrungsmittelarten	Verbrauch
1	1	Hartweizen (in Mehl ber.)	11.95
2	1	Weichweizen (in Mehl ber.)	51.71
3	1	Dinkel (in Mehl berechnet)	0.58
4	1	Roggen (in Mehl berechnet)	1.26
5	1	Mischel (in Mehl berechnet)	0.44
6	1	Gerste (in Mehl berechnet)	0.73
7	1	Hafer (in Mehl berechnet)	1.01
8	1	Mais (in Mehl berechnet)	1.51
9	1	Reis, geschält	4.83
10	2	Kartoffeln, -produkte	47.08

Abbrechen < Zurück Weiter > Fertigstellen

6. Auf der fünften Seite können Sie einen Primärschlüssel auswählen oder einen neuen hinzufügen. Da in unserer Tabelle noch kein Primärschlüssel vorhanden ist, sollten Sie **Primärschlüssel automatisch hinzufügen** wählen und auf **Weiter** klicken.



7. Auf der sechsten und letzten Seite des Importassistenten können Sie den Namen der neu erstellten Tabelle festlegen. Geben Sie **Verbrauch** ein und klicken Sie auf **Fertigstellen**, um den Importvorgang zu beenden.



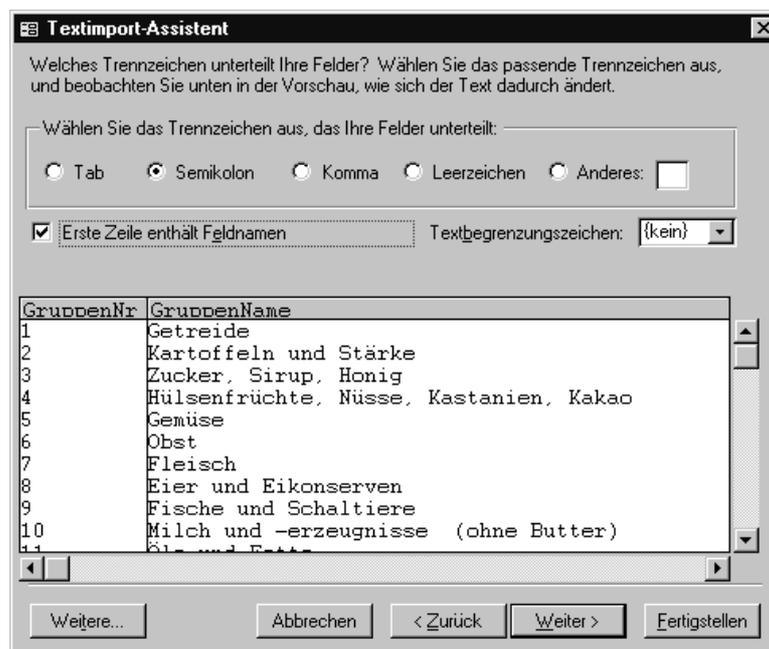
8. Verfahren Sie beim Import des ebenfalls in *Naehwertberechnung.xls* enthaltenen Tabellenblattes *NahrungsmittelNWT* analog wie beim Blatt *Verbrauch*.

6.4.2 Import aus Textdateien (Datei *LMGruppen.txt*)

1. Wählen Sie zum Importieren von *LMGruppen.txt* im Importdialog den Dateityp **Text Files** (*.txt;*.csv;*.tab;*.asc).
2. Auf der ersten Seite des Importassistenten bestimmen Sie, wie die einzelnen Felder in der Textdatei getrennt sind. Wählen Sie die Option **Mit Trennzeichen ...** und klicken Sie auf **Weiter**.



3. Auf der zweiten Seite können Sie das Trennzeichen festlegen. In unserem Fall handelt es sich um ein Semikolon. Ebenfalls geben Sie es auf dieser Seite an, wenn Sie möchten, dass auf der ersten Zeile die Namen der Attribute aufgeführt sind (hier Feldnamen genannt). Mit **Weiter** gelangen Sie zur dritten Seite.



4. Auf der dritten Seite können Sie wählen, wo Sie die importierten Daten speichern wollen. Wählen Sie **In einer neuen Tabelle** und klicken Sie auf **Weiter**.

5. Die vierte Seite ist für zusätzliche Feldoptionen vorgesehen. Dies ist aber in unserem Beispiel nicht notwendig. Klicken Sie direkt auf **Weiter**.
6. Auf der fünften Seite müssen Sie angeben, wie die Primärschlüssel ausgewählt werden sollen. Da die importierten Daten bereits einen Primärschlüssel enthalten, sollten Sie **Primärschlüssel selbst auswählen** anklicken und als Primärschlüssel `GruppenNr` eingeben. Mit **Weiter** gelangen Sie zur letzten Seite.



7. Auf der letzten Seite des Importassistenten legen Sie den Namen der neu erstellten Tabelle fest. Geben Sie `LMGruppen` ein und klicken Sie auf **Fertigstellen**, um den Importvorgang zu beenden.

6.4.3 Import aus anderen Datenbanken (Datei *NWT.dbf*)

1. Zum Importieren von *NWT.dbf* wählen Sie im Importdialog den Dateityp **dBASE IV (*.dbf)**. MS Access kann dBASE Daten direkt übernehmen, d.h. es erscheint kein Importassistent.

6.5 Theorie: Berechnungen in Abfragen

Mit Hilfe von Abfragen lassen sich Berechnungen an Gruppen von Daten durchführen. In MS Access stehen dafür verschiedene Funktionen wie z.B. *Summe*, *Gruppierung*, *Mittelwert*, *Minimum*, *Maximum*, *Anzahl*, *Standardabweichung* oder *Varianz* zur Verfügung. Berechnungen in Abfragen werden auf allen Datensätzen einer Tabelle ausgeführt. Im nachfolgenden Abschnitt 6.7 werden Sie in der Abfrage *BereinigteNWT* z.B. die Funktion *Mittelwert* bei einer Gruppe von Daten berechnen.

Auch die Werte von Attributen in einzelnen Zellen können berechnet werden. Dies geschieht analog dem Berechnen von Zellinhalten in MS Excel, welches Sie bereits aus Praxis 2 kennen. Im Abschnitt 6.7.4 werden Sie in der Abfrage *EnergieProGruppe* mit einer längeren Formel den Feldinhalt des Attributs *Energie* berechnen.

6.5.1 Datenbank versus Tabellenkalkulation

In Praxis 2 lernten Sie das Tabellenkalkulationsprogramm MS Excel kennen und in dieser Übung das Datenbankprogramm MS Access. Beim Anlegen von Datensammlungen stellt sich die Frage nach der geeigneteren Speicherungsform: Datenbank oder Tabellenkalkulation. Im ersten Teil der Ergänzungsübung hatten Sie mit Datenimport aus anderen Datenbanken, Datenverknüpfung und der Abfrage von Daten zu tun: ein Schwerpunktgebiet von Datenbankprogrammen wie MS Access. Im zweiten Teil der Ergänzungsübung ist die Berechnung und das Aufbauen der Berechnungsformel ein wichtiger Bestandteil, darauf wiederum ist MS Excel spezialisiert.

Ein Tabellenkalkulationsprogramm wie MS Excel ist ideal für kurze Berechnungen, Visualisierungen, Formeln, elementare Statistiken sowie auch zur Aufbewahrung von Datensätzen, die sich in einer Tabelle speichern lassen. MS Excel kommt jedoch schnell an seine Grenzen, sobald die Datenstrukturen komplexer werden. Auch ist die Anzahl von Spalten und Zeilen in einer Excel-Tabelle begrenzt. Ebenfalls problematisch wird die Verwendung von MS Excel, wenn mehrere Benutzer gleichzeitig auf Daten zugreifen und diese verändern wollen. Auch die Performance bei Abfragen lässt in MS Excel manchmal zu wünschen übrig.

Für solche Fälle ist der Einsatz eines Datenbankprogrammes wie MS Access vorteilhafter. Dank der relationalen Denkweise können komplexe Datenstrukturen in einem einfachen und universellen Modell gespeichert werden. Datenbankprogramme unterstützen den Benutzer auch mit vielen zusätzlichen Funktionen, um grosse und komplexe Datenmengen überschaubar zu machen z.B. mit effizienten Abfrage- und Suchmöglichkeiten.

Vor dem Anlegen von Datensammlungen lohnt es sich, über die Art der Daten und deren zukünftige Verwendung nachzudenken. Danach kann fallspezifisch eine optimale Aufbewahrungsform gesucht werden, z.B. ein Datenbankprogramm oder ein Tabellenkalkulationsprogramm.

6.6 Praxis: Berechnungen in Abfragen über die importierten Daten

6.6.1 Bereinigte Nährwert-Tabelle

1. In der Tabelle *NahrungsmittelNWT* gibt es Nahrungsmittelarten, die mehrfach vorkommen. So gibt es z.B. vier Einträge *Anderes frisches Obst*. Anhand der Tabelle *NWT* sollen mit einer Abfrage Mittelwerte für die vier Nahrungs-Grundbausteine (Proteine, Kohlenhydrate, Fette und Alkohol) berechnet werden. Dazu gehen Sie wie folgt vor:

- Erstellen Sie eine neue Abfrage *BereinigteNWT*.
- Zeigen Sie die Tabellen *NahrungsmittelNWT* und *NWT* an.
- Verknüpfen Sie die Felder *NWTNr* der beiden Tabellen.
- Fügen Sie folgende Felder ein:

Nahrungsmittelarten (Funktion: Gruppierung). Wählen Sie hierzu in der Zeile **Funktion** aus dem Pull-down-Menü die Funktion **Gruppierung** aus.

- PROT (Funktion: **Mittelwert**).
Wählen Sie hierzu in der Zeile Funktion aus dem Pull-down-Menü die Funktion „Mittelwert“ aus.
- FETT (Funktion: **Mittelwert**)
- KH (Funktion: **Mittelwert**)
- ALK (Funktion: **Mittelwert**)

6.6.2 Berechnung der Nährstoffaufnahme pro Tag

2. In einer weiteren Abfrage soll die Nährstoffaufnahme pro Person und pro Tag berechnet werden. Gehen Sie dazu von der Tabelle *Verbrauch* und der Abfrage *BereinigteNWT* aus. Für diese Abfrage wird die Tatsache ausgenutzt, dass nicht nur Abfragen über Tabellen, sondern auch Abfragen über Abfragen möglich sind. Dazu gehen Sie wie folgt vor:

- Erstellen Sie eine neue Abfrage *TagesNährstoffaufnahme*.
- Zeigen Sie die Tabelle *Verbrauch* an.
- Zeigen Sie die Abfrage *BereinigteNWT* an.
- Verknüpfen Sie die Felder *Nahrungsmittelarten*.
- Fügen Sie die Felder *GruppenNr* und *Nahrungsmittelarten* aus der Tabelle *Verbrauch* ein.
- Fügen Sie vier neue, berechnete Felder für die Nahrungs-Grundbausteine ein. Beispiel Protein:
 $\text{ProteinProTag} = (1000 * [\text{Verbrauch}] / 365) * ([\text{Mittelwert von PROT}] / 100)$
- Die berechneten Felder für Fett, Kohlenhydrate und Alkohol ergeben sich analog.

6.6.3 Gruppieren nach Lebensmittelgruppen

3. Eine weitere Abfrage soll die berechneten Werte aus *TagesNährstoffaufnahme* nach Lebensmittelgruppen gruppieren und innerhalb jeder Gruppierung summieren. Dazu gehen Sie wie folgt vor:

- Erstellen Sie eine neue Abfrage *TagesNährstoffaufnahmeProGruppe*.
- Zeigen Sie die Tabelle *LMGruppen* an.
- Zeigen Sie die Abfrage *TagesNährstoffaufnahme* an.
- Verknüpfen Sie die Felder *GruppenNr*.
- Fügen Sie folgende Felder ein:
 - GruppenNr* aus der Abfrage *TagesNährstoffaufnahme* (Funktion: **Gruppierung**; Sortierung: **Aufsteigend**)
 - GruppenName* aus der Tabelle *LMGruppen* (Funktion: **Gruppierung**)
 - Felder *ProteinProTag*, *FettProTag*, *KohlenhydrateProTag* und *AlkoholProTag* aus der Abfrage *TagesNährstoffaufnahme* (Funktion: **Summe**)

6.6.4 Berechnung der Energie pro Lebensmittelgruppe

4. Mit einer approximativen Formel soll zum Schluss aus dem berechneten Verbrauch die Energie pro Lebensmittelgruppe in Kilokalorien errechnet werden. Damit können Sie aussagen, wieviel jede Lebensmittelgruppe zu unserer Ernährung beiträgt. Gehen Sie wie folgt vor:

- Erstellen Sie eine neue Abfrage *EnergieProGruppe*.
- Zeigen Sie die Abfrage *TagesNährstoffaufnahmeProGruppe* an.
- Fügen Sie die Felder *GruppenNr* und *GruppenName* ein.
- Fügen Sie ein neues Feld *Energie* ein. Geben Sie anschliessend in der Zeile **Feld** die folgende Formel zur Berechnung der Werte des Feldes *Energie* ein:
$$\text{Energie: } 4 * [\text{Summe von ProteinProTag}] + 9 * [\text{Summe von FettProTag}] + 4 * [\text{Summe von KohlenhydrateProTag}] + 7 * [\text{Summe von AlkoholProTag}]$$

7 Berichte

Um Information in leicht verständlicher und anschaulicher Form zu vermitteln, ist es wichtig, sie in ein ansprechendes Layout zu verpacken. Dieses Konzept haben wir bereits bei den Formularen kennengelernt. Allerdings haben wir dort einfach nur den Layout-Vorschlag von MS Access übernommen.

In diesem Abschnitt wollen wir anhand eines Berichtes genauer untersuchen, welche Layout-Möglichkeiten uns mit MS Access zur Verfügung stehen.

Ein Bericht ist eine effektive Methode Daten und Ergebnisse von Abfragen so zu präsentieren, dass die zu vermittelnde Information gut sichtbar ist. Berichte eignen sich besonders für grosse Datenlisten und für den Ausdruck auf Papier.

Ein Bericht besteht aus zwei Teilen. Ein Teil ist das Layout wie z.B. Überschriften, grafisches Design etc.; der zweite Teil sind die Daten, die in den Tabellen der Datenbank gespeichert sind und hier in einem Bericht in gefilterter Form präsentiert werden.

Wir werden nun schrittweise einen Bericht erstellen, der den Lagerbestand einer Videothek darstellt. Das Ergebnis sollte etwa wie folgt aussehen (hier ist nur die erste Seite dargestellt):

<i>Film-Statistik</i>		
<i>Filmkategorie</i>	<i>Titel</i>	<i>Anzahl Kassetten</i>
Action		
	Eraser	4
	Terminator 2: 3 D	2
	True Lies	2
Summe		8
Drama		
	Blue Sky	5
	Dances with Wolves	5
	Dead Man Walking	3
	Leaving Las Vegas	2
	Philadelphia	4
	Rain Man	2
	Schindler's List	3
Summe		24
Humor		
	Asterix erobert Amerika	2
	Back to the Future	2
	Back to the Future (III)	5
	Back to the Future (Par	2

Montag, 6. Januar 2007 Seite 2 von 2

Der Bericht stellt folgende Information in formatierter Form dar:

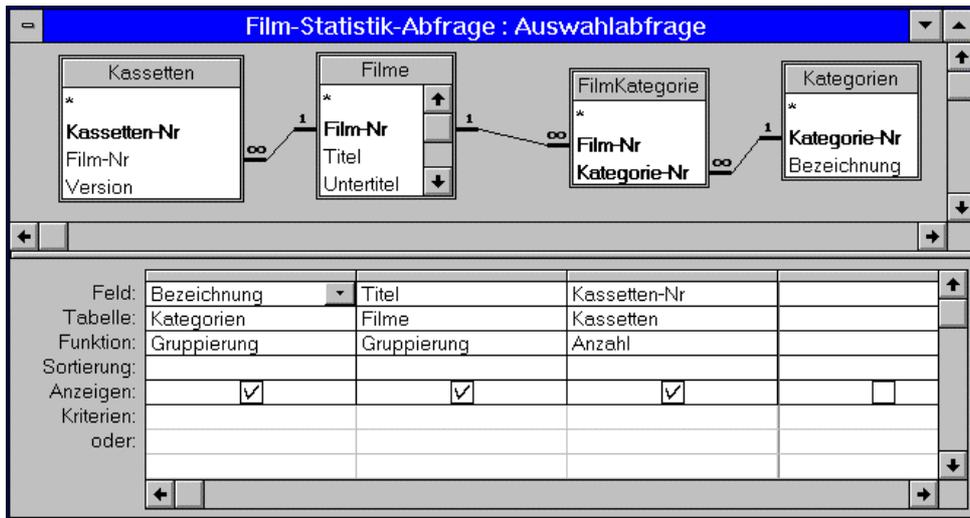
- Für jeden Film in der Videothek wird aufgelistet, wieviele Kassetten es gibt.
- Die Information ist nach Filmkategorien geordnet.
- Für jede Kategorie ist die Summe der existierenden Kassetten aufgelistet.

Das Erstellen eines Berichtes besteht typischerweise aus zwei Teilschritten. Zuerst wird die Information, die in dem Bericht aufgelistet wird, mittels einer Abfrage aus der Datenbank gefiltert. In einem zweiten Schritt wird dann die Information in ein ansprechendes Layout eingebunden.

7.1 Erstellen einer berichtsbezogenen Abfrage

Wir erstellen nun eine Abfrage nach genau demselben Muster, wie wir es in Kapitel 4 kennengelernt haben.

1. Klicken Sie im **Datenbank**-Fenster auf die Karteikarte **Abfragen** und wählen Sie den Befehl **Neu**. Im folgenden Fenster **Entwurfsansicht** fügen Sie die Tabellen *Kassetten*, *Filme*, *FilmKategorie* und *Kategorien* hinzu.
2. Ziehen Sie das Feld *Bezeichnung* aus der Tabelle *Kategorien*, das Feld *Titel* aus der Tabelle *Filme*, und das Feld *Kassetten-Nr* aus der Tabelle *Kassetten* in den Abfrageentwurfsbereich. Die Tabelle *FilmKategorie* benötigen wir nur, um eine Verbindung zwischen Filmen und Kategorien herzustellen. Das Feld *Kassetten-Nr* benutzen wir, um zu berechnen, wieviele Kassetten es von jedem Film gibt. Deswegen müssen wir hier die Funktion *Anzahl* benutzen (Klicken Sie auf das Funktionssymbol Σ in der Symbolleiste des Access-Fensters, falls die Aktionszeile nicht sichtbar ist).
3. Die Join-Operationen, die notwendig sind, um alle Tabellen miteinander logisch zu verbinden, werden vom System automatisch ausgeführt.
4. Die fertige Abfrage sieht in etwa wie folgt aus:



5. Speichern Sie diese Abfrage unter dem Namen *Film-Statistik-Abfrage*.

7.2 Grobentwurf eines Berichtes mit dem Berichtsassistenten

Microsoft Access bietet für viele Vorgänge, wie z.B. die Erstellung von Formularen, Abfragen und Berichten, Unterstützung an. Ein *Assistent* leitet die Benutzerin bzw. den Benutzer durch eine Reihe von Fenstern, in denen man z.B. angibt, welche Attribute einer Tabelle in einem Bericht oder einer Abfrage verwendet werden sollen oder wie das Layout eines Berichts oder eines Formulars in etwa aussehen soll. Mit Hilfe dieser Angaben produziert der Assistent einen ersten Entwurf, der dann manuell verbessert werden kann. (Bemerkung: momentan treten in einigen Versionen von Microsoft Access noch Fehler bei der Verwendung von Assistenten auf. Es ist anzunehmen, dass Microsoft dieses Problem in zukünftigen Versionen beheben wird).

Wir wollen den *Berichtsassistenten* verwenden, um eine erste Fassung unseres Berichts zu erstellen.

6. Klicken Sie im **Datenbank**-Fenster auf die Karteikarte **Berichte** und wählen Sie den Befehl **Neu**. Wählen Sie in dem erscheinenden Fenster den **Berichtsassistenten**. Sie werden nun durch eine Anzahl weiterer Fenster geleitet, mit denen Sie endgültig bestimmen, welche Daten der Bericht enthalten soll und mit denen Sie die Grobstruktur des Layouts festlegen.
7. Im ersten Fenster bestimmen Sie, auf welche Tabelle oder Abfrage sich der Bericht beziehen soll und welche Felder innerhalb dieser Tabelle bzw. Abfrage berücksichtigt werden sollen. Wählen Sie die Abfrage *Film-Statistik-Abfrage* und alle Felder dieser Abfrage:

Welche Felder soll Ihr Bericht enthalten?
 Sie können aus mehr als einer Tabelle oder Abfrage auswählen.

Tabellen/Abfragen:
 Abfrage: Film-Statistik-Abfrage

Verfügbare Felder:

Ausgewählte Felder:
 Bezeichnung
 Titel
 Anzahl von Kassetten-Nr

Abbrechen < Zurück Weiter > Fertigstellen

8. Mit dem nächsten Fenster bestimmen Sie, ob es eine Gruppierung innerhalb des Berichts geben soll. Wählen Sie *Bezeichnung* als Gruppierungsebene:



9. Mit Hilfe des dritten Fensters bestimmen Sie die Sortierreihenfolgen und markieren nötigenfalls welche Art von Summenbildung durchgeführt werden soll. Wählen Sie als Sortierungsmerkmal das Attribut "Titel" und klicken Sie anschliessend auf **Zusammenfassungsoptionen**. Aktivieren Sie für *AnzahlvonKassetten-Nr* das Feld **Summe**:





10. Wählen Sie mit Hilfe des nächsten Fensters (nicht gezeigt) das Grob-Layout (Anordnung der Gruppen und Überschriften). Danach vergeben Sie einen Namen für den Bericht. Falls Sie es wünschen, können sie zu diesem Zeitpunkt auch Dokumente abrufen, die Sie beim Arbeiten mit dem Bericht unterstützen (Display Help):



7.3 Abschliessendes Design

Es erscheint nun die erste Version des Berichtes. Die Gestaltung bedarf aber noch einiger Verbesserungen. Die Befehle **Layout-Vorschau** und **Berichtsentwurf** im Menü **Ansicht**, erlauben es Ihnen das momentane Layout zu betrachten bzw. den Entwurf zu modifizieren.

Film-Statistik : Bericht															
Berichtskopf															
<i>Film-Statistik</i>															
Seitenkopf															
<i>Filmkategorie</i>				<i>Titel</i>				<i>Anzahl Kassetten</i>							
Bezeichnung - Kopfbereich															
Bezeichnung															
Detailbereich															
								Titel				Anzahl von Kasset			
Bezeichnung - Fußbereich															
Summe															
=Summe([Anzahl															
Seitenfuß															
=Jetzt)															
="Seite " & [Seite] & " von " & [Seiten]															
Berichtsfuß															
Summe Tota															
=Summe([Anzahl															

Wir können folgende Unterteilungen in einem Bericht feststellen. Zuerst hat der Bericht als Ganzes einen Kopf, einen Rumpf und einen Fuss. Das gleiche gilt für jede Seite des Berichts. Im Falle, dass Gruppierungen definiert wurden, gibt es noch eine Einteilung jeder Gruppe in Kopf, Rumpf und Fuss. In jedem dieser Bereiche können *Steuerelemente* positioniert werden.

1. Ein Steuerelement kann Daten der zugrundeliegenden Tabelle oder Abfrage enthalten oder einfach nur freier Beschriftungstext sein. Jedes Steuerelement für sich ist durch seine Eigenschaften definierbar (klicken auf das Element, so dass die Umrandung erscheint, klicken auf die rechte Maustaste und anwählen der Eigenschaften).
2. Ein Steuerelement kann optisch verändert werden (z.B. durch Verschieben, Vergrößern, Farbgebung etc.). Zusätzlich kann es inhaltlich modifiziert werden, falls es Daten beinhaltet (z.B. durch Filteroperation oder Auswahl eines anderen Feldes).
3. Ein Bericht kann auch grafische Elemente beinhalten. Mit Hilfe der **Toolbox** (Menü **Ansicht**) können z.B. Linien oder ganze Bilder in den Bericht eingebunden werden. Auch grafische Elemente sind Steuerelemente.
4. Versuchen Sie nach genauerem Studieren des Hilfe-Fensters den Bericht so zu formatieren, wie er am Anfang des Kapitels dargestellt ist. Natürlich können Sie nach Ihrem eigenen Geschmack Änderungen vornehmen ☺.

8 Datenbankentwurf

Eine Datenbank stellt eine Sammlung von Daten der realen Welt dar. Mit Hilfe der Datenbank sollen die Daten in geordneter Form vorliegen, effizient gespeichert werden, vielseitig und langfristig verwendbar und einfach manipulierbar sein. Deswegen muss eine Datenbank sorgfältig entworfen werden. Es kann schwerwiegende Folgen haben, wenn man die Struktur von Tabellen, die bereits Daten beinhalten, verändern will.

Beim **Datenbankentwurf** legt man die Struktur der abspeicherbaren Datenobjekte fest. Es entsteht ein **Datenbankschema**. Die **Modellierung der Daten** wird dabei in zwei Schritten durchgeführt. Im ersten Schritt werden rein **konzeptionell** die Entitätsmengen und die Beziehungstypen zwischen den Entitäten festgelegt, die in der Datenbank festgehalten werden sollen. Sowohl Entitäten als auch Beziehungen können

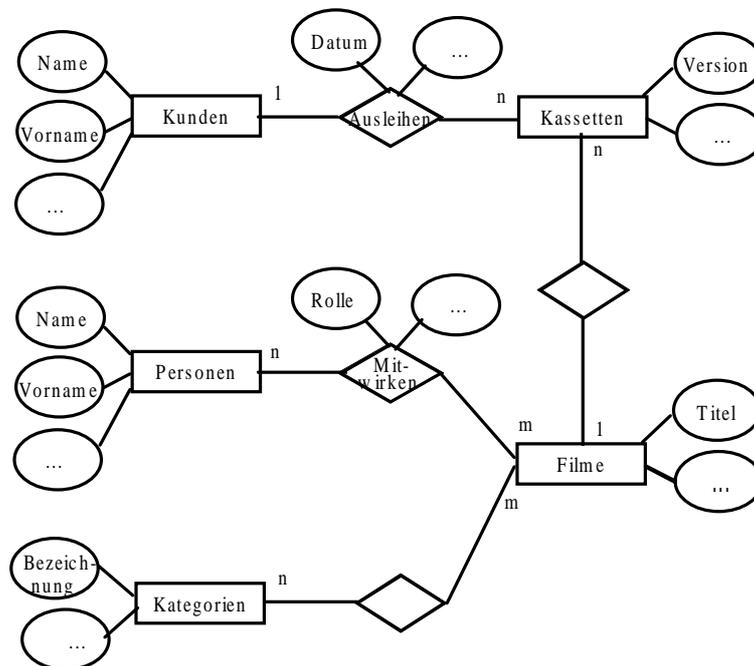
Attribute aufweisen. In einem zweiten Schritt werden dann Entitätsmengen und Beziehungstypen in **Relationen** umgewandelt.

Wir wollen nun die Datenmodellierung anhand der Videodatenbank durchführen. Gleichzeitig wollen wir dabei auf Fehler hinweisen, die häufig auftreten, wenn der Entwurf nicht sorgfältig genug durchgeführt wird. Dazu sind in der Vorlesung Relationennormalisierungen eingeführt worden, durch die diese Fehler vermieden werden. Wir wollen diese Normalisierungen auf informelle Weise beschreiben. Da in MS Access alle Tabellen automatisch in 1. Normalform sind, gehen wir nur auf die 2. und 3. Normalform ein.

8.1 Konzeptioneller Entwurf

In einem ersten Schritt zerlegen wir unsere Video-Welt in Entitäten und Beziehungen. Dabei werden ähnliche Entitäten zu Entitätsmengen (auch als Entitätstypen bezeichnet) und ähnliche Beziehungen zu Beziehungstypen zusammengefasst. Z.B. ist die Kassette mit der Nr. 40 eine Entität. Da alle Kassetten ähnliche Eigenschaften haben, werden sie zur Entitätsmenge *Kassette* zusammengefasst. Ähnlich ist „Alois Meier“ eine Entität der Entitätsmenge *Kunde*. Hat Alois Meier die Kassette Nr. 40 ausgeliehen, so besteht eine Beziehung zwischen beiden. Die Gesamtheit all dieser „Ausleihbeziehungen“ wird in dem Beziehungstyp *Ausleihe* festgehalten.

Die folgende Abbildung zeigt eine mögliche Modellierung der Videodatenbank. Rechtecke stellen Entitätsmengen dar, Rauten repräsentieren Beziehungstypen und Ovale beschreiben die Attribute von Entitäten und Beziehungen. (Diese Form von grafischer Datenmodellierung wird **Entity-Relationship-Modell** genannt). Entitäten sind über Beziehungen miteinander verbunden. Dabei unterscheiden wir in unserem Beispiel 1:n-Beziehungen, bei denen eine Entität einer Menge mit mehreren Entitäten der anderen Mengen verbunden sein kann und n:m Beziehungen, bei der es bei keiner Entitätsmenge eine Einschränkung in der Anzahl der Entitäten gibt. So kann ein Kunde mehrere Kassetten ausleihen, aber eine Kassette kann nicht mehrmals ausgeliehen sein. Hingegen können Personen in mehreren Filmen mitwirken und in jedem Film wirken mehrere Personen mit.

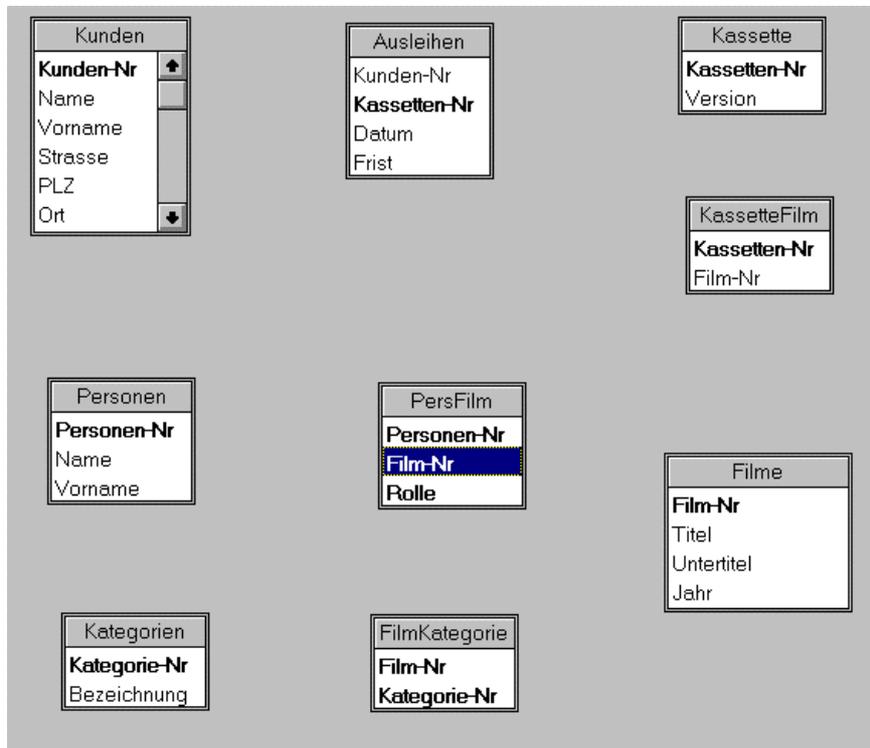


Die Einteilung in Entitätsmengen und Beziehungstypen ist nicht ganz einfach und vor allem nicht eindeutig (es gibt bestimmt noch andere Möglichkeiten die Daten einer Videothek zu ordnen). Dass Kunden eine andere logische Einheit als Kassetten darstellen, mag ja noch einleuchtend sein, aber warum nicht Information über Filme und Kassetten in einer Entitätsmenge unterbringen? Auch Beziehungen sind häufig nicht intuitiv. Das kann man in der Grafik dadurch erkennen, dass teilweise keine sinnvollen Namen für Beziehungstypen gefunden wurden. Deswegen ist es wichtig, das erstellte Schema auf Anomalien zu testen und gegebenenfalls nachzubessern. Auf Anomalien gehen wir im nächsten Abschnitt ein.

Vorsicht: Die hier beschriebenen Beziehungen sind zwar mit den Access-Beziehungen verwandt, aber nicht identisch. Wir werden später den Zusammenhang beschreiben.

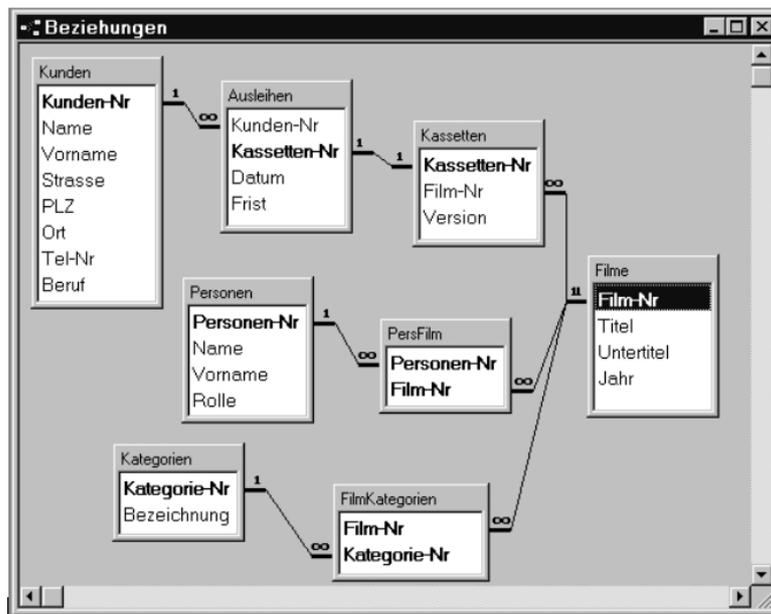
8.2 Relationaler Entwurf

Der konzeptionelle Entwurf muss nun so umgeformt werden, dass MS Access ihn versteht. Dazu wandeln wir in einem ersten Schritt jede Entitätsmenge und jeden Beziehungstyp in eine Tabelle um. Die Attribute werden zu Feldern der Tabellen. Die einzelnen Tabellen sehen in MS Access damit in etwa wie folgt aus:



Dies ist fast das Schema, wie Sie es in der Testaufgabe kennengelernt haben. Jedoch können noch einige Optimierungen durchgeführt werden, um die Anzahl der Tabellen zu vermindern. Im Fall von 1:1-Beziehungen und 1:n-Beziehungen ist es nicht notwendig, Beziehungstypen als eigene Tabellen darzustellen, sondern sie können mit einer der Entitätsmengen zusammengefasst werden. Dabei kann ein Beziehungstyp mit einer Entitätsmenge zusammengefasst werden, wenn beide denselben Primärschlüssel besitzen. Wir könnten also die Beziehungstypen *Ausleihen* und *KasetteFilm* mit der Entitätsmenge *Kasette* zusammenfügen. Dabei erscheinen sowohl die Attribute der Beziehungstypen als auch der Entitätsmenge in der zusammengefassten Tabelle. Das Zusammenfassen von Beziehungen und Entitäten ist aber nicht immer sinnvoll. In unserem Fall war das Zusammenfügen des Beziehungstypen *Ausleihen* und der Entitätsmenge *Kassetten* nicht sinnvoll: eine Kasette kann zwar höchstens einmal ausgeliehen sein, aber sie muss nicht ausgeliehen sein. Trotzdem enthält nun jeder Datensatz der Tabelle *Kassetten* Attribute, die den Ausleihzustand beschreiben. Diese Attribute sind nicht mit Werten gefüllt, wenn eine Kasette nicht ausgeliehen ist. Das kann nicht nur zu Speicherplatzverschwendung führen, sondern zeigt auch, dass eine derartige Modellierung nicht ganz "sauber" ist. Die Tabelle *Kasette* enthält jetzt Attribute, die Eigenschaften ausdrücken, die gar nicht allen Kassetten eigen sind. Ein Zusammenbinden von Beziehungstyp und Entitätsmenge in eine Relation ist also nur dann sinnvoll, wenn diese Beziehung für alle Entitäten zutrifft (wie z.B. bei der Beziehung *Kasette* - *Film*), evtl. auch dann, wenn die Beziehung nur sehr wenige Attribute hat. Wir haben deswegen in unserer Video-Datenbank das *Ausleihen* als eigene Tabelle definiert.

In der folgenden Abbildung sind auch die Access-Beziehungen mit eingezeichnet worden. Die 1:n-Beziehungen zwischen *Kunden* und *Kassetten* und zwischen *Kassetten* und *Filme* sind direkt zu erkennen (1:∞). Bei n:m-Beziehungen muss der Beziehungstyp als eigenständige Tabelle erhalten bleiben und ist durch hierarchische 1:∞- und ∞:1-Verknüpfungen mit seinen Nachbarn verbunden.



8.3 Fehlerquellen

Die beschriebene Vorgehensweise erscheint auf den ersten Blick unnötig umständlich. Wir wollen in diesem Abschnitt zeigen, zu welchen Problemen es kommen kann, wenn bei der konzeptionellen Datenmodellierung nicht sorgfältig genug vorgegangen wird.

Häufig passiert es, dass Beziehungstypen und Entitätsmengen, die nicht zusammengehören, in einer Relation zusammengefasst werden. Wenn wir z.B. Entitäten, die Kunden beschreiben, mit Informationen, welche mit der Ausleihe zu tun haben, kombinieren, dann entsteht eine Tabelle, die etwa so aussieht:

	Kunden-Nr	Name	Vorname	Strasse	PLZ	Ort	Tel-Nr	Kassetten-Nr	Datum	Frist
▶	1	Meier	Alois	Hauptstr. 23	8000	Zurich	3682490	2	03.12.96	08.12.96
	1	Meier	Alois	Hauptstr. 23	8000	Zurich	3682490	3	01.12.96	07.12.96
	2	Giger	Brunhilde	Zwerggasse 4	7270	Davos	4729208	6	02.12.96	07.12.96
	2	Giger	Brunhilde	Zwerggasse 4	7270	Davos	4729208	7	05.12.96	10.12.96
	3	Roten	Melanie	Tannenstr. 17	8006	Zurich	4322323	10	04.12.96	10.12.96
*	0						0	0		

Anhand dieses Beispiels können wir drei *Anomalien* beobachten:

- **Löschanomalie:** Mit dem Zurückgeben von Kassetten werden die entsprechenden Datensätze gelöscht. Geben Kunden alle Kassetten zurück, die sie ausgeliehen haben, dann wird auch die Kundeninformation gelöscht, obwohl das eigentlich nicht das Ziel war.
- **Einfügeanomalie:** Will man Information über eine neue Kundin einfügen obwohl diese noch keine Kassette ausgeliehen hat (wir möchten ihr vielleicht einen Werbebrief schicken), dann müssen die Felder, die das Ausleihen von Kassetten betreffen, mit NULL-Werten besetzt werden. Dieses Problem hatten wir bereits im vorherigen Abschnitt, wenn *Kassetten* und die *Ausleihe* in einer Relation zusammengefügt werden. Ist zudem die *Kassetten-Nr* allein der Primärschlüssel (was sinnvoll wäre, da eine Kassette höchstens einmal ausgeliehen werden kann), dann können Kunden gar nicht eingefügt werden, wenn sie keine Kassetten ausleihen.
- **Änderungsanomalie:** Erhält z.B. ein Kunde oder eine Kundin eine neue Adresse oder Telefonnummer, dann muss diese Änderung evtl. in mehreren Datensätzen durchgeführt werden. Wenn nun einer der betroffenen Datensätze nicht geändert wird, dann enthält die Datenbank widersprüchliche Informationen.

Insbesondere kann man gut erkennen, dass in diesem Beispiel **Redundanz**, in der Form mehrfach gespeicherter Daten auftritt. Zusätzlich kann ein **Informationsverlust** dann eintreten, wenn alle Datensätze eines Kunden oder einer Kundin gelöscht werden. Weil Felder mit NULL-Werten belegt werden verschärft sich das Problem der **Speicherplatzverschwendung**.

Die Probleme entstanden, weil wir nicht zusammenpassende Information in einer Relation gebündelt haben, in diesem Fall Kundeninformation mit Ausleihinformation. Das kann man daran erkennen, dass im Grunde

zwei Primärschlüssel in der Tabelle auftreten (obwohl die *Kassetten-Nr* als Identifikationsschlüssel ausreichen würde). Ein Schlüssel ist die *Kunden-Nr.* und die gesamte Kundeninformation hängt allein von diesem Schlüssel ab (ist allein durch diesen Schlüssel bestimmt). Der zweite Schlüssel ist die *Kassetten-Nr.* und die Ausleihfristen hängen nur von diesem Schlüssel ab.

Man vermeidet diese Anomalien, indem man nur Relationen kreiert, bei denen alle Attribute, die nicht zum Primärschlüssel gehören, von **allen** Attributen des Primärschlüssels abhängen.

Hängen einzelne Attribute nur von Teilen des Primärschlüssels ab, dann muss die Tabelle in mehrere Tabellen zerlegt werden. Eine Tabelle, in der alle Nichtschlüsselattribute von **allen** Attributen des Primärschlüssels abhängen, ist in der **2. Normalform**.

Wir hatten weiter oben beschrieben, dass es auf den ersten Blick nicht unbedingt einleuchtend ist, eine eigene Entitätsmenge *Film* zu haben, anstatt die Information mit der Entitätsmenge *Kassetten* zusammenzufassen. In diesem Fall würden zu jeder Kassette der Filmtitel, Produktionsjahr etc. gespeichert werden. Doch auch in diesem Fall treten falsche Abhängigkeiten in der Tabelle auf. Z.B. hängt das Produktionsjahr eines Filmes nur indirekt von der *Kassetten-Nr* ab. Es ist eigentlich vielmehr durch den Film bestimmt, der sich gerade auf der Kassette befindet. Eine derartige indirekte Abhängigkeit eines Nichtschlüsselattributs vom Primärschlüssel wird auch als **transitiv** bezeichnet. Man sieht schnell, dass es zu Redundanz kommt, falls es mehr als eine Kassette für denselben Film gibt. Sollte ausserdem ein anderer Film auf die Kassette gespielt werden, muss nicht nur der Filmtitel, sondern auch das Produktionsjahr geändert werden. Eine Tabelle ist in **3. Normalform**, wenn sie in **2. Normalform** ist und *keine transitiven* Abhängigkeiten bestehen.

8.4 Tips für die Datenmodellierung

- Sammeln Sie zuerst alle Informationen, die in der Datenbank gespeichert werden sollen. Wird nach der Erstellung der Datenbank erkannt, dass wichtige Informationen fehlen, ist das spätere Einfügen meist sehr zeitaufwendig.
- Erstellen Sie zuerst ein konzeptuelles Schema: Ordnen Sie die gesamte Information in Entitätsmengen und Beziehungstypen an. Listen Sie alle Attribute auf. Überprüfen Sie, ob **alle** Abhängigkeiten zwischen den Daten dargestellt sind. Prüfen Sie auf Vollständigkeit.
- Bestimmen Sie zu jeder Entitätsmenge und jedem Beziehungstyp den Primärschlüssel.
- Klären Sie für jede Entitätsmenge und jeden Beziehungstyp, ob alle Nichtschlüsselattribute von allen Attributen des Identifikationsschlüssels abhängen und keine transitiven Abhängigkeiten existieren.
- Fassen Sie Beziehungstypen nur mit Entitätsmengen zusammen, wenn sie denselben Primärschlüssel haben und die Beziehung für jede Entität der Entitätsmenge gilt (oder der Beziehungstyp wenig Attribute enthält).
- Überprüfen Sie bei den entstandenen Tabellen durch Einfügen von Testdatensätzen, ob
 - a) Die gleiche Information in mehreren Datensätzen gespeichert wird (Redundanz).
 - b) Bei vielen Datensätzen bestimmte Felder mit NULL-Werten gefüllt werden.
 - c) Beim Löschen eines Datensatzes Information verloren geht, die in der Datenbank bleiben sollte.
- Zerlegen Sie die Tabellen gegebenenfalls und wiederholen Sie das Testen.

Literatur

C. A. Zehnder: *Informationssysteme und Datenbanken*. vdf Hochschulverlag Zürich (6. Aufl.) 1998.

8.5 Abfragen mit SQL

Die bisher gezeigte Methode zum Aufbau von Abfragen war graphisch orientiert. Dieses Vorgehen wird auch als **Query by Example (QBE)** bezeichnet. Es gibt weitere standardisierte Verfahren, die Daten einer Datenbank abzurufen. Eine weitverbreitete Form ist **SQL (Structured Query Language)**, eine Abfragesprache, die von allen bedeutenden relationalen Datenbanken verstanden und unterstützt wird.

QBE und SQL sind gleichwertig was ihre Leistungsfähigkeit bezüglich Abfragen angeht. Das heisst aus der Sicht der Datenbanktheorie, dass beide Sprachen Abfragen des sicheren relationalen Kalküls erlauben. Dies sei aber nur am Rande erwähnt. Der Vor- bzw. Nachteil von QBE ist, dass es rein interaktiv bedient wird. So hat MS Access eine graphische Schnittstelle, die den mehr graphisch orientierten Aufbau von Abfragen

zulässt. SQL dagegen ist eine Sprache, die sozusagen "im Hintergrund" Abfragen zulässt ohne eine direkte Interaktion mit dem Benutzer. SQL ist im Gegensatz zu QBE auch für Datenmanipulation und Datendefinition verwendbar. Wir beschränken uns auf einfache SQL-Abfragen, die aber die Verwendung aller relationaler Operatoren zulassen (siehe Begleittext zur Vorlesung, Abschnitt 12.3.5).

Auch mit MS Access kann man SQL-Abfragen stellen. Diese sind nicht grundlegend anders oder schwieriger als die bisher vorgestellten Abfragen, sie unterscheiden sich lediglich in der Art, wie man sie stellt. Wurde bisher eine Abfrage mehr oder weniger mit der Maus zusammengestellt, indem Tabellen und deren Felder in Relation zueinander gesetzt wurden, erlaubt SQL eine textuelle Beschreibung der gewünschten Abfrage. Im Folgenden sollen Analogien zwischen QBE und SQL gezeigt und ein kurzer Einblick in den Umgang mit SQL geben werden.

1. Zwischen der bisher zur Abfrageformulierung benutzten Entwurfsansicht und der SQL-Ansicht (ein Eingabefenster, in das die SQL-Abfrage als Text eingegeben wird) kann man wechseln, indem man den Befehl **SQL-Ansicht** aus dem Menü **Ansicht** auswählt. Zurück in die Entwurfsansicht bzw. Datenansicht gelangt man auf analogem Weg. Um überhaupt eine Abfrage stellen zu können muss man natürlich eine neue Abfrage erstellen bzw. eine bestehende bearbeiten. Dazu klickt man wie bisher im Datenbankfenster auf die Registerkarte **Abfragen** und geht weiter vor wie bisher.

Ist man in der SQL-Ansicht, kann man Abfragen an die Datenbank mittels SQL stellen. Die Sprache SQL unterstützt unter anderem Projektion, Selektion und Join-Bildung (siehe Skript zur Vorlesung, Kapitel 12.3.5).

2. Einige Qualitäten von SQL sollen nun an Beispielen verdeutlicht werden. Für unser Beispiel benutzen wir die Tabellen Kunden und Ausleihen mit folgenden Datensätzen:

Tabelle Kunden:

KubdenNr	Name	Vorname	TeNr	Beruf
1	Meier	Alois	3682490	Bänkler
2	Giger	Bruno	4729208	Maurer
3	Zobrist	Monika	2327282	Professorin

Tabelle Ausleihen:

KundenNr	KassettenNr	Datum
1	1	2. Nov. 1998
3	3	4. Nov. 1998
2	12	12. Nov 1998
3	15	1. Nov. 1998

Eine SQL-Abfrage hat prinzipiell die Form:

SELECT *Projektion* **FROM** *Join* **WHERE** *Selektion*

Es gibt zahlreiche Erweiterungen, die an dieser Stelle nicht alle gezeigt werden können. Bei weiterem Interesse sei hier auf die Hilfestellung von MS Access verwiesen (Aufruf F1, Indexsuche nach SQL).

Die Bedeutung der Teile *Projektion*, *Join* und *Selektion* ist folgende:

- *Projektion* wählt aus der Menge der Feldnamen (Attribute) einer Tabelle (Relation) die Feldnamen aus, die als Ergebnis angezeigt werden sollen, auf die die Ergebnistabelle also "projiziert" werden soll. Sind die gewählten Feldnamen innerhalb der mittels *Join* ausgewählten Tabellen nur einmal vorhanden, können sie direkt angegeben werden. Sollten aber in mehreren Relationen gleiche Feldnamen auftreten, ist der gesamte Name der Relation durch einen Punkt getrennt dem Feldnamen vorweg zu stellen, um MS Access klar zu machen, welchen Feldnamen man wirklich meint. Eine abkürzende Schreibweise für alle Feldnamen ist der Stern *.
- *Join* wählt die Relationen aus, die in die Abfrage einbezogen werden sollen. Es entsteht eine neue Relation, die die Feldnamen aller gewählten Relationen enthält. Zwei Relationen werden hier im Sinn des sogenannten kartesischen Produktes zusammengefügt, d.h. alle Datensätze einer Tabelle werden mit allen Datensätzen der anderen Tabelle kombiniert.
- Die *Selektion* gibt an, welche Einträge der durch *Join* bestimmten (zusammengesetzten) Relation ausgewählt (selektiert) werden und schliesst die anderen aus dem Ergebnis aus. Hierzu sind

Vergleichsoperationen wie >, <, >=, <=, =, **LIKE**, **MATCHES**, **IN** usw. vorhanden, die zusätzlich durch **AND**, **OR** oder **NOT** miteinander verknüpft werden können. Welche darüber hinausgehenden Operationen konkret vom verwendeten Datenbanksystem unterstützt werden und dort zulässig sind, variiert und kann in den zugehörigen Handbüchern nachgeschlagen werden. Die Selektionsbedingung kann entfallen, was zu einem "ungefilterten" Ergebnis führt, in dem alle Zeilen einer Relation mit allen Zeilen der anderen Relationen kombiniert werden: Das kartesische Produkt der Relationen.

3. Der Einsatz von SQL soll nun anhand von Beispielen und der oben angegebenen Tabellen gezeigt werden.

- **SELECT * FROM** Ausleihen

ergibt als Ergebnis die Tabelle Ausleihen selber, da sowohl alle Feldnamen ausgewählt werden (*) und keine Selektionsbedingung vorhanden ist, also alles ausgewählt wird.

Kunden-Nr	Kassetten-Nr	Datum
1	1	2. Nov. 1998
2	12	12. Nov 1998
3	3	4. Nov. 1998
3	15	1. Nov. 1998

- **SELECT * FROM** Ausleihen **WHERE** Datum < '1998-11-04'

selektiert nur Zeilen, bei denen der Wert des Feldes Datum ein Datum vor dem 4. Nov. 1998 hat.

Kunden-Nr	Kassetten-Nr	Datum
1	1	2. Nov. 1998
3	15	1. Nov. 1998

- **SELECT** [Kassetten-Nr], Datum **FROM** Ausleihen **WHERE** Datum < '1998-11-04'

projiziert das Ergebnis zusätzlich nur auf die Feldnamen *Kassetten-Nr* und *Datum*:

Kassetten-Nr	Datum
1	2. Nov. 1998
15	1. Nov. 1998

Bemerkung: Das Attribut *Kassetten-Nr* wurde hier in eckige Klammern [] gesetzt um zu verhindern, dass Access den Bindestrich als Minuszeichen interpretiert und versucht, die beiden nichtvorhandenen Attribute *Kassetten* und *Nr* voneinander zu subtrahieren. Mit den eckigen Klammern wird der Ausdruck als zusammenhängender Name gekennzeichnet.

- **SELECT * FROM** Kunden, Ausleihen

liefert als Ergebnis den (ungefilterten) Join der beiden Relationen *Ausleihen* und *Kunden*, d.h. anschaulich jede Zeile der einen Tabelle mit jeder Zeile der anderen Tabelle kombiniert (das kartesische Produkt).

Kunden. Kunden-Nr	Name	Vorname	TelNr	Beruf	Ausleihen. Kunden-Nr	Kassetten-Nr	Datum
1	Meier	Alois	3682490	Bankler	1	1	2. Nov. 1998
2	Giger	Bruno	4729208	Maurer	1	1	2. Nov. 1998
3	Zobrist	Monika	2327282	Professorin	1	1	2. Nov. 1998
1	Meier	Alois	3682490	Bankler	3	3	4. Nov. 1998
2	Giger	Bruno	4729208	Maurer	3	3	4. Nov. 1998
3	Zobrist	Monika	2327282	Professorin	3	3	4. Nov. 1998
1	Meier	Alois	3682490	Bankler	2	12	12. Nov 1998
2	Giger	Bruno	4729208	Maurer	2	12	12. Nov 1998
3	Zobrist	Monika	2327282	Professorin	2	12	12. Nov 1998
1	Meier	Alois	3682490	Bankler	3	15	1. Nov. 1998
2	Giger	Bruno	4729208	Maurer	3	15	1. Nov. 1998
3	Zobrist	Monika	2327282	Professorin	3	15	1. Nov. 1998

Hier sieht man deutlich, dass die Feldnamen *Kunden-Nr* in beiden Relationen auftreten und deshalb nicht eindeutig sind. Aus diesem Grund setzt MS Access der Darstellung den Feldnamen die zugehörigen Relationennamen vorweg. So wird z.B. der Feldname *Kunden-Nr* der Relation *Kunden* zu *Kunden.Kunden-Nr* (Relation.Feldname).

- **SELECT** Kunden.[Kassetten-Nr], Name, Vorname, [Kassetten-Nr], Datum
FROM Kunden, Ausleihen **WHERE** Kunden.[Kassetten-Nr]=Ausleihen.[Kassetten-Nr]
 selektiert nun alle Zeilen der vorherigen Tabelle, in denen die Kundennummern der Relationen *Kunden* und *Ausleihen* übereinstimmen. Da der Name *Kunden-Nr* als Feldname sowohl in der Relation *Kunden* als auch in der Relation *Ausleihen* vorkommt, muss der Name der Relation im Selektionsteil der SQL-Anweisung verwendet werden (*Kunden.Kunden-Nr=Ausleihen.Kunden-Nr*), um genau zu bestimmen, welcher der jeweiligen Feldnamen gemeint ist.

Kunden-Nr	Name	Vorname	Kassetten-Nr	Datum
1	Meier	Alois	1	2. Nov. 1998
2	Giger	Bruno	12	12. Nov 1998
3	Zobrist	Monika	3	4. Nov. 1998
3	Zobrist	Monika	15	1. Nov. 1998

Da alle Feldnamen im Ergebnis wieder eindeutig sind, zeigt MS Access hier keine Namen von Relationen mehr an.

- **SELECT** Name, Vorname **FROM** Kunden, Ausleihen **WHERE** Kunden.[Kassetten-Nr]=Ausleihen.[Kassetten-Nr]
 liefert dasselbe Ergebnis wie vorher, projiziert auf die Feldnamen *Name* und *Vorname*.

Name	Vorname
Meier	Alois
Giger	Bruno
Zobrist	Monika
Zobrist	Monika

Es fällt auf, dass Zeilen doppelt vorkommen. Dies liegt an der Arbeitsweise, mit der SQL-Abfragen ausgewertet werden. SQL erstellt sogenannte Multimengen (auch als Bags bezeichnet), d.h. Mengen, die doppelte Einträge zulassen. Mit Blick auf das vorangegangene Beispiel ist ersichtlich, warum doppelte Ergebniszeilen auftreten. Sie entstehen durch Ausblenden der Spalten *KundenNr*, *KassettenNr* und *Datum* aus dem Ergebnis (bzw. Mit anderen Worten durch Projektion auf *Name* und *Vorname*), das nun doppelte Zeilen besitzt. Um diese zu eliminieren existiert der Befehl **DISTINCT**, der der **SELECT**-Anweisung nachgestellt werden kann und mehrfache Datensätze auf einen reduziert. Die SQL-Abfrage lautet somit:

- **SELECT DISTINCT** Name, Vorname **FROM** Kunden, Ausleihen
WHERE Kunden.[Kunden-Nr]=Ausleihen.[Kunden-Nr]
 und liefert das Ergebnis:

Name	Vorname
Meier	Alois
Giger	Bruno
Zobrist	Monika

4. Die Hilfestellung von MS Access gibt Auskunft über weitere mögliche Operationen, Aggregationen, Selektionsformeln und die Syntax von SQL-Abfragen. Erstellen Sie zur Übung SQL-Abfragen, die folgendes Ergebnis liefern:

- Welcher Kunde oder welche Kundin (Name/Vorname) hat die Kassette 12 ausgeliehen?
- Welche Kassetten sind von Bankern ausgeliehen?