

Übung Grundlagen der Datenverarbeitung (EDL 1)

Lektion "Entwurf einer Datenbank"



Übungsziel

Ziel dieser Übung ist es, einige Grundbegriffe des Datenbankentwurfs kennen zu lernen und an einem Beispiel praktisch zu üben. Dies beinhaltet:

- Szenario und Aufgabenstellung 1
- ERM: Entity-Relationship-Modell 2
- Beziehungen 4
- Referenzielle Integrität 5
- Normalisierung 6
- Eigenschaften der Attributwerte 9
- Auswahlabfrage mit SQL 9
- Übung: Auswahlabfrage mit MS Access 12
- Übung: Bericht mit MS Access 14
- Alternativübung: Auswahlabfrage mit OpenOffice.org Base 15
- Alternativübung: Bericht mit OpenOffice.org Base 17



Szenario und Aufgabenstellung



Landschaftspflegeverbände sind nichtstaatliche regionale Aktionsbündnisse aus Vertretern von Naturschutz, Landwirtschaft und Politik. Deren Ziel ist es, Naturschutzbehörden und Kommunen bei der praktischen Umsetzung naturschutzfachlicher und landschaftspflegerischer Zielsetzungen entlastend zu unterstützen. Der Landschaftspflegeverband Dachau e. V. (LPV Dachau) wurde 1993 gegründet. Die Tätigkeit des LPV Dachau umfasst im Wesentlichen Schutz, Entwicklung und Pflege vorhandener, wertvoller Naturflächen sowie die Betreuung neu geschaffener Flächen, die im Zuge umfangreicher Flurneuordnungsverfahren entstanden sind. Die Gesamtfläche des LPV Dachau ist in einzelne Projektbereiche aufgliedert.

1. Aufgabe:

Um einen Überblick über die durchgeführten Maßnahmen für die Flächenpflege hinsichtlich der Kosten zu erhalten, soll für den Bereich des Landschaftspflegeverbandes Dachau e. V. eine relationale Datenbank erstellt werden.

Dabei stellt die Pflegefläche die Grundeinheit dar, die unabhängig von rechtlichen Eigentumsverhältnissen als eigene Größe zu sehen ist. Auf sie bezogen sind alle Detailinformationen sowie die Zugehörigkeit zu einem Maßnahmentyp und zu einem Projekt zu erfassen. Wichtig ist die Verwaltung der Informationen getrennt nach Jahren, da die Einheit „Pflegefläche“ keine statische ist. Sowohl die in einem Jahr gepflegten Flächen können variieren als auch die Eigenschaften ein und derselben Fläche (Größe, Bestand) in unterschiedlichen Jahren. Ebenso können sich Auftragnehmer ändern und mit Sicherheit sind die anfallenden Kosten nicht in jedem Jahr genau dieselben.

2. Aufgabe:

Daten vergangener Jahre sollen zu Vergleichszwecken in naturschutzfachlicher wie wirtschaftlicher Hinsicht erhalten werden. Die jährlich anfallenden Maßnahmen und Kosten sollen in einem Bericht zusammengestellt und ausgedruckt werden können.



ERM: Entity-Relationship-Model



Beim Entwurf einer Datenbank wird ein Ausschnitt der realen Welt als Modell abgebildet und nach bestimmten Kriterien und Anforderungen optimiert:

- Zunächst werden in einer Anforderungsanalyse die Anforderungen der Datenbankbenutzer gesammelt und nach Kriterien klassifiziert
- Anschließend wird festgelegt, welche Daten gespeichert werden sollen
- Als nächstes werden die Daten strukturiert, miteinander in Beziehung gesetzt, optimiert und in relationalen Datenbanken als Tabellen implementiert

Zur Unterstützung dieses Prozesses wird das **Entity-Relationship-Model** (ERM) verwendet.

Informationsobjekte werden als **Entity** bzw. **Entities** (engl.: Wesen, Einheit) bezeichnet. Gleichartige Informationsobjekte werden im ERM zu **Entitytypen** zusammengefasst. Entitytypen werden in relationalen Datenbanken als Tabellen gespeichert.

Die Pflegeflächen „Brücke12“ und „Vierkirchen8“ des LPV Dachau sind solche Entities. Aus diesen Entities kann ein Entitytyp mit dem Namen „Fläche“ abgeleitet werden. Zusätzlich zum Namen besitzt der Entitytyp auch Eigenschaften, diese werden als **Attribute** bezeichnet. Im Übungsbeispiel werden die Attribute „Flächename“, „Beschreibung“, „Maßnahmentyp“, „Projektkürzel“, „Projekttitle“, „Größe“, „Kosten“, „Jahr“ und „Auftragnehmer“ verwendet. Natürlich müssten zur umfassenden Beschreibung des LPV Dachau noch weitere Entitytypen definiert werden, wie z. B. der Entitytyp „Ort“ mit den Attributen „Ortsnummer“, „Gemarkungsnummer“, „Ortsname“ und „Gemarkung“ (Abb. 1).

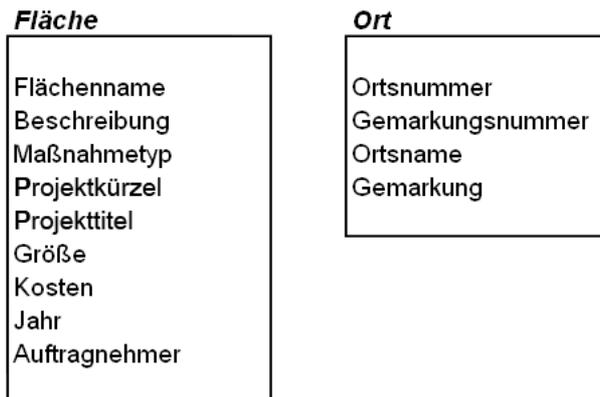


Abb. 1: ERM graphische Darstellung der Entitytypen „Fläche“ und „Ort“ mit Attributen



Jede Entity benötigt zur Identifizierung einen *eindeutigen*¹ Wert, über den die gesamten Attribute (Datensatz) der Entity eindeutig adressiert werden. Dieses Attribut nennt man **Primärschlüssel** (siehe Einführungsskript EDL 1 Modul Datenbank 1).²

Im Fall des Entitytyps „Fläche“ wäre ein möglicher Primärschlüssel der „Flächename“ (z. B. Brücke1), vorausgesetzt, dass Flächennamen keine Duplikate aufweisen.

Beim Entitytyp „Ort“ muss zur eindeutigen Adressierung ein **zusammengesetzter Schlüssel** aus „Ortsnummer“ und „Gemarkungsnummer“ verwendet werden, da eine Gemarkung sich durchaus über mehrere Gemeindegebiete erstrecken kann und deshalb z. B. allein durch die "Gemarkungsnummer" der Ort nicht eindeutig beschrieben ist. Erst in Verbindung mit der "Ortsnummer" lässt sich ein Ort eindeutig identifizieren (vgl. Abb. 2).

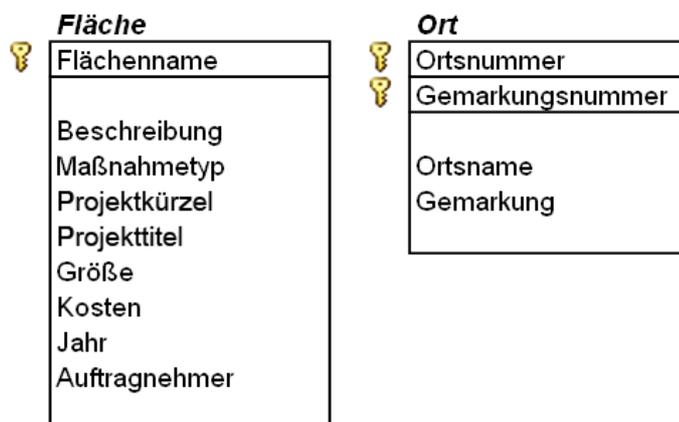


Abb. 2: ERM Darstellung der Entitytypen „Fläche“ und „Ort“ mit Primärschlüsseln

¹ Mathematisch: ein-eindeutig, also unverwechselbar

² Der Primärschlüssel wird in Datenbankprogrammen häufig mit einem Schlüsselssymbol gekennzeichnet



Beziehungen



Zwischen den Entitytypen können nun Beziehungen hergestellt werden. Beziehungen sind Abhängigkeiten (logische Verknüpfungen) zwischen Entities, die immer gelten müssen.

Beziehungen lassen sich aus der realen Situation ableiten. So kann sich z. B. *eine* Fläche immer nur an *einem* bestimmten Ort befinden. An *einem* Ort können aber *mehrere* Flächen liegen. In diesem Fall spricht man von einer **1:n Beziehung**.

Eine **1:1 Beziehung** liegt vor, wenn immer nur ein Datensatz einer Tabelle mit genau einem Datensatz einer anderen Tabelle verknüpft ist. Allerdings werden in der Praxis solche Beziehungen aufgelöst und die Werte *in einer einzigen* Tabelle gespeichert.

Bei der **m:n Beziehung** können mehrere Datensätze verschiedener Tabellen einander zugeordnet sein. So kann *eine* Fläche von *mehreren* Auftragnehmern gepflegt werden, aber auch *ein* Auftragnehmer kann *mehrere* Flächen pflegen. Im ERM wird die m:n Beziehung in zwei 1:n Beziehungen aufgelöst, die über eine neue Entity (Tabelle), z. B. „Pflegeauftrag“, verknüpft werden.

Fläche A => wird gepflegt von => Auftragnehmer 1, 2, 3
 Auftragnehmer 1 => pflegt die => Fläche A, B, C



Für die graphische Darstellung des ERM gibt es verschiedene Notationen, die sich alle an der ERM Datenmodellierung nach Chen orientieren.³

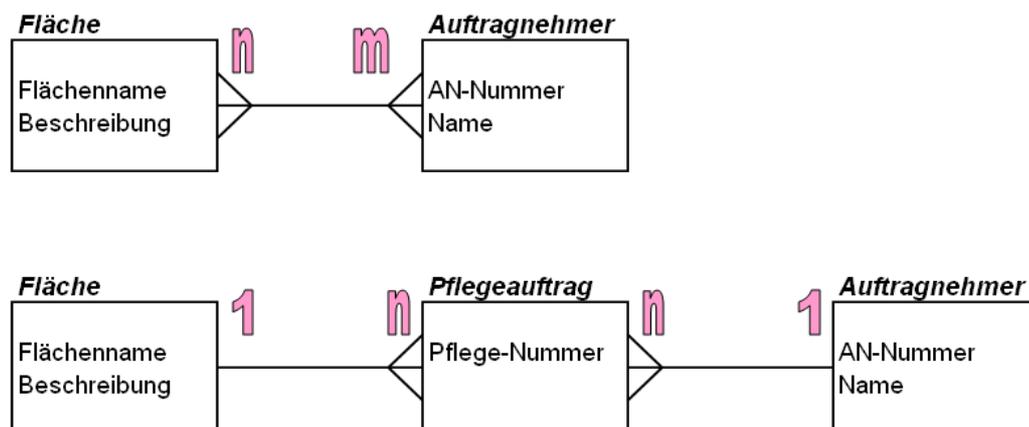


Abb. 3: Auflösung der m:n Beziehung in zwei 1:n Beziehungen

Abb. 3 zeigt die Auflösung der m:n-Beziehung in zwei 1:n-Beziehungen über die zusätzliche Tabelle „Pflegeauftrag“. Zur Darstellung wurde die sog. „Krähenfuß“-Notation verwendet. Dabei wird die n-Seite mit einem einfachen Strich, die 1-Seite mit drei Strichen \leftarrow gekennzeichnet (für eine andere Darstellungsart siehe Abb. 7).

³ <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/datenwissen/Datenmanagement/Daten-/Entity-Relationship-Model-->

Ein Pflegeauftrag bezieht sich immer nur auf *einen* Auftragnehmer und auf *eine* Fläche. Eine Fläche oder ein Auftragnehmer kann aber *mehreren* Pflegeaufträgen zugeordnet werden.

Werden Entitytypen in Beziehung gesetzt, muss grundsätzlich auf der „n-Seite“ der Beziehung (Detailtabelle) der Primärschlüssel der „1er-Seite“ der Beziehung (Mastertabelle) als neues Attribut (Fremdschlüssel, hier unterstrichen) eingefügt werden (Abb. 4).



Abb. 4: Die Primärschlüssel „ Flächenname“ und „ AN-Nummer“ werden im Entitytyp (Tabelle) „Pflegeauftrag“ als Fremdschlüssel eingefügt



Referentielle Integrität



Bei der Aufteilung der Daten in verschiedene Tabellen, die dann über Beziehungen miteinander verknüpft werden, kann es allerdings bei Dateneingabe, -änderung oder -löschung zu Problemen kommen. Ein Beispiel: In einer Datenbank werden die Daten der Kunden und die von ihnen getätigten Bestellungen in zwei verschiedenen Tabellen geführt. Wenn aus irgendeinem Grund ein Kunde gelöscht wird, obwohl noch eine Bestellung von ihm offen ist, sind die Bezüge zwischen den Tabellen nicht mehr schlüssig – also: die *Referenzen* sind nicht mehr *integer*.

Um diese **Referentielle Integrität** gewährleisten zu können, müssen bestimmte Regeln eingehalten werden:

- Wenn ein Kunde gelöscht wird, müssen auch die von ihm getätigten Bestellungen gelöscht werden („*kaskadiertes Löschen*“ oder „*Löschweitergabe*“)
- Wenn ein Kunde eine neue Kundennummer bekommt, müssen in allen darauf Bezug nehmenden Tabellen die Fremdschlüssel (Kundennummern) angepasst werden („*kaskadiertes Ändern*“ oder „*Aktualisierungsweitergabe*“)

PC-Datenbanksysteme können z. T. so eingestellt werden, dass sie die Einhaltung der Referenziellen Integrität überwachen bzw. kaskadiertes Löschen und kaskadiertes Ändern automatisch durchführen.



Normalisierung



Mit welchen Werten die Attribute einer Tabelle belegt sein können, zeigt folgendes Beispiel:

Der Projektbereich Weichser Moos wird vom LPV Dachau unter dem Kürzel NSG_WM geführt. Zu diesem Bereich gehört die Fläche Brücke1. Ein Teil der Fläche Brücke1 wurde im Jahr 2001 vom Auftragnehmer Ulrich Kias für 390 Euro gemäht. Die gesamte Fläche Brücke1 wurde im selben Jahr von Walter Demel für 7500 Euro aufgemessen.

Der obige Text kann auch in Tabellenform dargestellt werden (Tabelle 1).

Tabelle 1: Erster Entwurf zur Tabelle „Weichser Moos“

Fläche	Beschreibung	Maßnahmetyp	Projektkürzel	Projekttitle	Größe	Kosten	Jahr	Auftragnehmer
Brücke1	Im Glonntal	Mähen	NSG_WM	Weichser Moos	13000	390 €	2001	Ulrich Kias
		Messen			15000	7500 €		Walter Demel

Allerdings kann diese Tabelle noch nicht als Tabelle einer relationalen Datenbank umgesetzt werden. Ein wesentliches Ziel des Datenbank-Entwurfes ist die Vermeidung von Redundanz und Inkonsistenz. **Redundanz** bedeutet die mehrfache Speicherung von gleichen Daten. **Inkonsistenz** entsteht, wenn es unterschiedliche Versionen der gleichen Daten in einer Datenbank gibt. Außerdem wird eine einfache Daten- und Datenbankpflege durch einen übersichtlichen und möglichst einfachen Aufbau der Tabellen angestrebt.

Um diese Ziele zu erreichen, wird das Verfahren der **Normalisierung** auf den Datenbankentwurf angewendet. Dabei werden die Daten in mehreren Schritten auf Tabellen aufgeteilt, bis die in der jeweiligen „Normalform“ verlangten Voraussetzungen erfüllt sind:



1. Normalform

Stehen in den Attributen (Feldern) einer Tabelle nur atomare Werte, so ist diese Tabelle in der ersten Normalform (1NF).

Alle Mehrfacheinträge in einem Attribut (Feld) müssen entfernt werden. Jedem Attribut (Feld) eines Datensatzes darf nur ein Wert zugewiesen werden.

In der Beispieltabelle (Tabelle 1) enthalten die Felder Maßnahmetyp, Größe, Kosten und Auftragnehmer jeweils mehrere Werte. Für die 1NF müssen diese Mehrfacheinträge in verschiedene *Datensätze* aufgeteilt werden.

Tabelle 2: Die Tabelle zum NSG Weichser Moos in der ersten Normalform (1NF)

Fläche	Beschreibung	Maßnahmetyp	Projektkürzel	Projekttitle	Größe	Kosten	Jahr	AN_Name	AN_Vorname
Brücke1	Im Glonntal	Mähen	NSG_WM	Weichser Moos	13000	390 €	2001	Kias	Ulrich
Brücke1	Im Glonntal	Messen	NSG_WM	Weichser Moos	15000	7500 €	2001	Demel	Walter

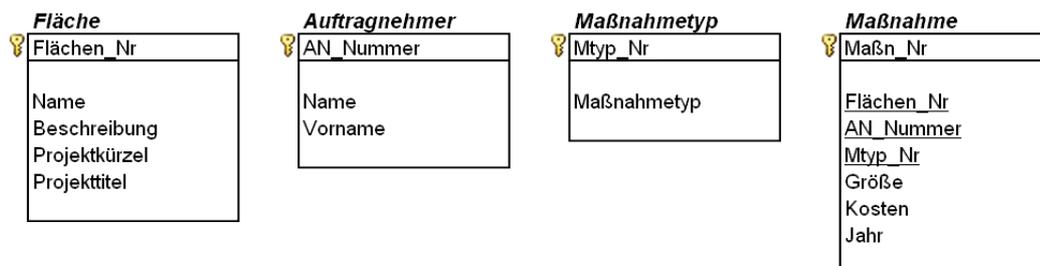


2. Normalform

Eine Tabelle befindet sich in der zweiten Normalform (2NF), wenn sie sich in der ersten Normalform befindet und darüber hinaus die Werte aller Nichtschlüssel-Attribute ausschließlich vom Primärschlüssel bestimmt werden, bzw. von allen Attributen eines zusammengesetzten Primärschlüssels.

Um dies zu erreichen, werden Tabellen in kleinere Tabellen zerlegt, so dass in jeder Tabelle alle Nichtschlüssel-Attribute nur noch vom Primärschlüssel abhängen. Besteht ein Primärschlüssel aus mehreren Attributen, ist zu prüfen, ob es Attribute gibt, die eigentlich nur von einem Teil des Primärschlüssels abhängen. Ist dies der Fall, so ist dieser Teil des Primärschlüssels und die dazugehörigen Attribute in eine neue Tabelle zu bringen.

Betrachten wir die Beispieltabelle in der 1NF (Tabelle 2), so fällt auf, dass das Attribut „Fläche“ nicht mehr als Primärschlüssel verwendet werden kann, da mehrere Datensätze den Wert „Brücke1“ enthalten. Um die einzelnen Attribute eindeutig zuordnen zu können, ist die Schaffung eines neuen Attributes, in unserem Beispiel „Maßn_Nr“, als Primärschlüssel nötig. Von diesem Primärschlüssel sind nun nicht alle Attribute direkt abhängig. Es liegt eine sogenannte „innere Abhängigkeit“ in der Tabelle vor. So sind die Attribute „Beschreibung“, „Projektkürzel“ und „Projekttitle“ vom Attribut „Fläche“ abhängig, und das Attribut „AN_Vorname“ steht in Abhängigkeit vom Attribut „AN_Name“. Diese indirekt vom Primärschlüssel abhängigen Attribute lassen sich nun in Tabellen auslagern und mit Primärschlüsseln versehen. In der Tabelle „Maßnahme“ (neuer Entitytyp) werden die Werte der einzelnen Tabellen über Fremdschlüssel wieder zusammengefasst (Abb. 5).



Maßnahme

Maßn_Nr	Flächen_Nr	AN_Nummer	Mtyp_Nr	Größe	Kosten	Jahr
1001	100	1	14	13000	390 €	2001
1002	101	2	17	15000	7500 €	2001

Abb. 5: Einfügen der Tabelle „Maßnahme“ mit Fremdschlüsseln – zweite Normalform



3. Normalform

Eine Tabelle ist in der dritten Normalform (3NF), wenn sich eine Tabelle in der 2NF befindet, und zusätzlich keine Abhängigkeiten von Nichtschlüssel-Attributen bestehen. Sobald ein Nichtschlüssel-Attribut nur über ein anderes Nichtschlüssel-Attribut identifizierbar ist, wird von transitiver Abhängigkeit gesprochen.

Maßnahme: Entfernen aller transitiven Abhängigkeiten⁴ für die 3NF durch Teilen der Tabelle in mehrere Tabellen, in denen alle Nichtschlüssel-Attribute direkt vom Primärschlüssel abhängig sind.

In diesem Beispiel enthält die Tabelle „Fläche“ noch eine innere Abhängigkeit von „Projektkürzel“ und „Projekttitle“. Werden diese Attribute in eine neue Tabelle „Projekt“ ausgelagert, bleibt die Information auch dann erhalten, wenn z. B. in einem Projektbereich keine Flächen gepflegt werden. Eine Änderung z. B. des Projekttitels muss nur noch in einem Datensatz einer Tabelle durchgeführt werden (Abb. 6).

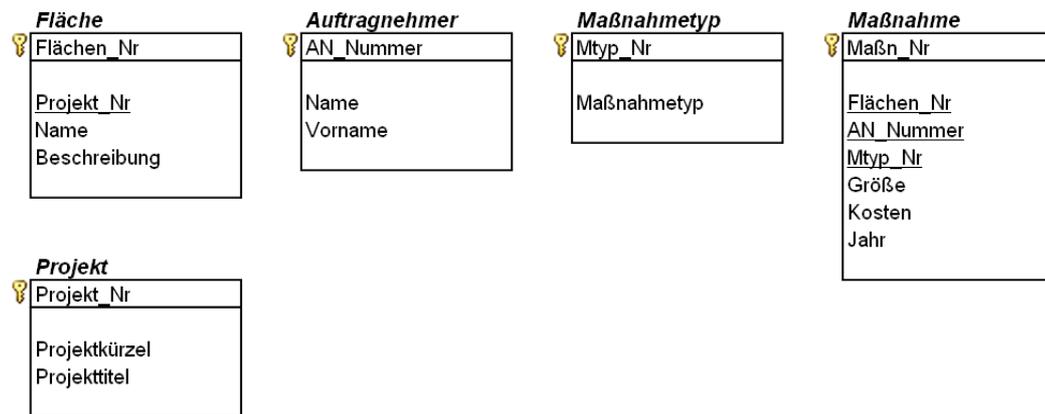


Abb. 6: Einfügen der Tabelle „Projekt“ – dritte Normalform unseres Beispiels

Abb. 7 zeigt das vom Datenbankmanagementsystem (DBMS) MS Access 2010 erzeugte grafische Layout des ERM mit den erstellten Beziehungen zwischen den Tabellen.

MS Access gibt die 1:n-Beziehungen mit der Beschriftung „1-∞“ an, den Primärschlüsseln (1-Seite) ist das Schlüsselssymbol vorangestellt, auf der „∞“-Seite steht das selbe Feld als Fremdschlüssel.

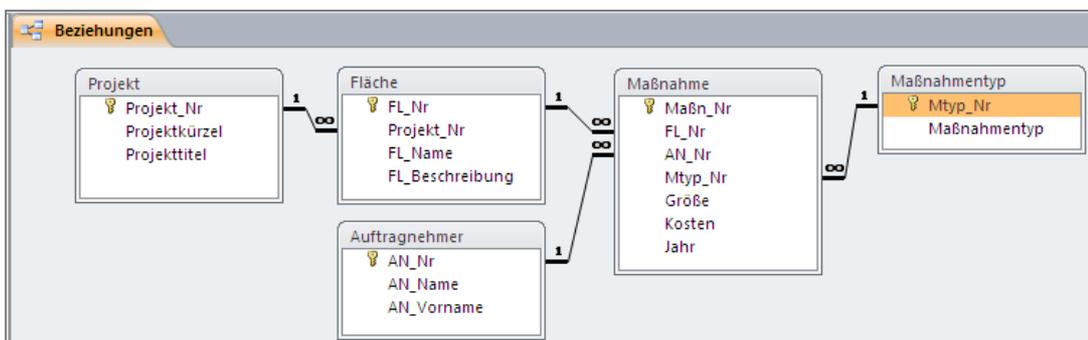


Abb. 7: Darstellung der Tabellenbeziehungen in MS Access 2010

⁴ Anm.: Eine Relation *R* heißt *transitiv*, wenn aus *xRy* und *yRz* folgt, dass *xRz*. Anders ausgedrückt: Eine Relation *R* heißt *transitiv*, wenn gilt: Stehen *x* und *y* und *y* und *z* in Relation, dann steht auch *x* mit *z* in Relation.



Eigenschaften der Attributwerte



Ist der Datenbankentwurf mit dem ERM abgeschlossen, setzen Sie den Entwurf in Tabellen des verwendeten Datenbanksystems um. Bei der Definition der Tabellenfelder müssen Sie die Eigenschaften der Attributwerte genauer beschreiben und eingrenzen. Eingrenzungen sind notwendig zur Bewahrung der Korrektheit der gespeicherten Daten. Die Eingrenzung erfolgt in 3 Stufen:

1. Zuerst wird der **Datentyp** (Zahl, Text, Datum, etc.) definiert. So hat zum Beispiel das Attribut „Kosten“ den Datentyp „Währung“, das Attribut „Vorname“ den Datentyp „Text“ und das Attribut „Maßn_Nr“ den Datentyp „Zahl“.
2. Als nächstes wird die Werte- oder **Datengröße** festgelegt. Bei den Attributen „Name“ und „Vorname“ werden keine längeren Texte eingegeben. Zur Reduzierung des Speicherbedarfs kann also die Länge des Datentyps Text auf z. B. 25 Zeichen beschränkt werden.
3. Die 3. Stufe der Eingrenzung ist die Festlegung von **Gültigkeitsregeln**.⁵ Angenommen die Maßnahmennummern, Attribut „Maßn_Nr“, des LPV Dachau sind immer vierstellig und bestehen nur aus positiven Zahlen. Diese Vorgabe kann über die Gültigkeitsregel „>1000 und <10000“ definiert werden. Die Eingabe einer negativen Zahl oder einer Zahl mit mehr oder weniger als vier Stellen ist nun nicht mehr möglich und damit die Korrektheit der Daten gewährleistet.



Auswahlabfragen mit SQL



Neben der Speicherung und Verwaltung von Daten ist ein weiterer wesentlicher Anwendungszweck von Datenbanken die Selektion, Verknüpfung und Darstellung von Daten. Ein Datenbankmanagementsystem stellt Werkzeuge zur Verfügung, mit denen Attribute ausgewählt werden können. Diese Werkzeuge werden Abfrage oder auch View genannt. Dabei handelt es sich einfach um einen Filter, der eine neue Sicht auf die vorhandenen Daten ermöglicht. Als Ergebnis einer Filterung entsteht wieder eine Tabelle, die auch wieder abgefragt werden kann.

Jedes DBMS besitzt eigene Werkzeuge zur Erstellung von Abfragen oder Views. Die meisten basieren auf der einfachen Abfragesprache SQL (Structured Query Language).⁶



Der Verbund von zwei oder mehr Tabellen über eine Abfrage wird auch als **Join** (engl. verbinden, vereinigen) bezeichnet.

Die gebräuchlichsten Joins sind „Inner Join“, „Left Join“, „Right Join“ und „Outer Join“.

Voraussetzung für Abfragen über mehrere Tabellen sind die festgelegten Tabellenbeziehungen über Primär- und Fremdschlüssel.

⁵ In OpenOffice Base bis Version 3.x nur über die Verwendung von Formularen möglich!

⁶ Für eine intensivere Beschäftigung mit SQL wird auf die weiterführende Literatur des EDL 1 Datenbankmoduls verwiesen.

Beispieltabellen zur Erläuterung der Joins:

Tabelle A: erste, linke Tabelle des Joins

Tabelle B: zweite, rechte Tabelle des Joins

Personal_Nr	Name	Vorname
1	Kias	Ulrich
2	Demel	Walter
3	Ditsch	Michael

Personal_Nr	Telefon
1	4182
7	4000
8	5000

**Inner Join**

Datensätze aus beiden Tabellen werden verbunden, wenn ein oder mehrere Felder den gleichen Wert haben. Datensätze, deren Primärschlüsselwert nicht als Fremdschlüsselwert vorhanden ist, werden auch nicht angezeigt. Der Inner Join ist die meist verwendete Abfrageverknüpfung.

Die Inner Join-Abfrage mit der Beziehung

„TabelleA.Personal_Nr = TabelleB.Personal_Nr“

führt zu folgendem Ergebnis:

Personal_Nr	Name	Vorname	Personal_Nr	Telefon
1	Kias	Ulrich	1	4182

Im Zusammenhang mit Inner Join wird oft der Begriff *Natural Join* verwendet. Dies ist ein Inner Join, der Attribute gleichen Inhalts nur einmal enthält. Im Beispiel entfällt die Spalte „Personal_Nr“ der Tabelle B:

Personal_Nr	Name	Vorname	Telefon
1	Kias	Ulrich	4182

**Left Join (auch Left-Outer-Join)**

Von der ersten in der Abfrage genannten, also der linken Tabelle werden alle Datensätze in die Ergebnistabelle übernommen. Von der in der Abfrage genannten zweiten, der rechten Tabelle werden nur die damit verbundenen Datensätze ausgewählt. Felder der rechten Tabelle werden leer angezeigt bzw. bekommen den Wert „Null“, wenn kein entsprechender Datensatz vorhanden ist.

Die Left Join-Abfrage mit der Beziehung

Abfrage „TabelleA.Personal_Nr = TabelleB.Personal_Nr“

führt zu folgendem Ergebnis:

Personal_Nr	Name	Vorname	Personal_Nr	Telefon
1	Kias	Ulrich	1	4182
2	Demel	Walter	Null	Null
3	Ditsch	Michael	Null	Null



Right Join (auch Right-Outer-Join)

Von der zweiten in der Abfrage genannten, also rechten Tabelle werden alle Datensätze in die Ergebnistabelle übernommen. Von der ersten in der Abfrage genannten, linken Tabelle werden nur die damit verbundenen Datensätze ausgewählt. Felder der linken Tabelle werden leer angezeigt bzw. bekommen den Wert „Null“, wenn kein entsprechender Datensatz vorhanden ist.

Die Right Join-Abfrage mit der Beziehung
 „TabelleA.Personal_Nr = TabelleB.Personal_Nr“
 führt zu folgendem Ergebnis:

Personal_Nr	Name	Vorname	Personal_Nr	Telefon
1	Kias	Ulrich	1	4182
Null	Null	Null	7	4000
Null	Null	Null	8	5000



Outer Join (auch Full-Outer-Join)

Eine Kombination aus Left- und Right Join. Alle Datensätze werden ausgewählt und passende Datensätze verbunden.

Die Outer Join-Abfrage mit der Beziehung
 „TabelleA.Personal_Nr = TabelleB.Personal_Nr“
 führt zu folgendem Ergebnis:

Personal_Nr	Name	Vorname	Personal_Nr	Telefon
1	Kias	Ulrich	1	4182
2	Demel	Walter	Null	Null
3	Ditsch	Michael	Null	Null
Null	Null	Null	7	4000
Null	Null	Null	8	5000



Beispiel Auswahlabfrage mit SQL

Es sollen alle Werte des Attributes „Name“ der Tabelle „TabelleA“ und alle Werte des Attributes „Telefon“ der Tabelle „TabelleB“ angezeigt werden, die der Bedingung „TabelleA.Personal_Nr = TabelleB.Personal_Nr“ entsprechen.

```
SELECT TabelleA.Name, TabelleB.Telefon
FROM TabelleA INNER JOIN TabelleB
ON TabelleA.Personal_Nr = TabelleB.Personal_Nr;
```

„ON“ gehört zum Join-Statement und definiert die Verknüpfungsbedingung über Primär- und Fremdschlüssel.

Dieses SQL-Statement führt zu folgendem Ergebnis:

Name	Telefon
Kias	4182



Übung: Auswahlabfrage mit MS Access



An unserem Beispiel soll eine Tabellenverknüpfungs-Abfrage mit SQL erläutert werden.



Kopieren Sie die für Sie vorbereiteten Dateien auf Ihre Festplatte. Diese stehen im Modul „Datenbank 1“ des Moodle-Kurses „EDL – Einführung in die Datenverarbeitung I“ unter „Übung zum Thema Datenbankentwurf“ zum Download bereit (ZIP-Datei). Entpacken Sie die Übungsdateien aus der Datei dbentwurf.zip in Ihr Arbeitsverzeichnis.



Für das Jahr 2001 und für den Projektbereich Weichser Moos (NSG_WM) sollen sämtliche Maßnahmen aufsteigend sortiert aufgelistet und folgende Attribute angezeigt werden: Maßn_Nr, Projektkürzel, FL_Name, Jahr, AN_Name, AN_Vorname, Größe, Kosten und Maßnahmetyp.

Für diese Abfrage müssen also 5 Tabellen miteinander verknüpft werden. Es werden zuerst zwei Tabellen mit Join verknüpft und eine Ergebnismenge gebildet. Diese wird wieder über einen Join mit der nächsten Tabelle verknüpft usw.



Obige Abfrage in SQL.

Kursiver Text dient der Erläuterung und ist nicht Bestandteil der SQL Syntax:

Auswahl der anzuzeigenden Felder „Tabellenname.Feldname“ mit SELECT:

```
SELECT Auftragnehmer.AN_Name, Auftragnehmer.AN_Vorname,
Fläche.FL_Name, Maßnahme.Maßn_Nr, Maßnahme.Größe, Maßnahme.Kosten,
Maßnahme.Jahr, Maßnahmetyp.Maßnahmetyp, Projekt.Projektkürzel
```

Auswahl und Verknüpfung der Tabellen mit FROM und INNER JOIN. Übersichtlichere Darstellung der INNER JOIN Schachtelung durch die Verwendung von Klammern:

```
FROM
1.Join:      Projekt INNER JOIN
2.Join:      (Maßnahmetyp INNER JOIN
3.Join:      (Fläche INNER JOIN
```

Durch ON wird die Verknüpfungsbedingung über Primär- und Fremdschlüssel definiert:

```
4.Join:      (Auftragnehmer INNER JOIN Maßnahme ON
              Auftragnehmer.AN_Nr = Maßnahme.AN_Nr)
3.Join:      ON Fläche.FL_Nr = Maßnahme.FL_Nr)
2.Join:      ON Maßnahmetyp.Mtyp_Nr = Maßnahme.Mtyp_Nr)
1.Join:      ON Projekt.Projekt_Nr = Fläche.Projekt_Nr
```

Mit WHERE wird die Bedingung für die Auswahl der anzuzeigenden Feldwerte definiert:

```
WHERE (((Projekt.Projektkürzel)="NSG_WM") AND
        ((Maßnahme.Jahr)=2001))
```

Aufsteigende Sortierung der Datensätze durch ORDER BY nach Werten des Attributes „Maßn_Nr“ und Abschluss des SQL Befehls durch Semikolon:

```
ORDER BY Maßnahme.Maßn_Nr;
```



Diese SQL Abfrage können Sie nun mit der Beispiel-Datenbank testen. Die Datenbank enthält vorbereitete und bereits in Beziehung gesetzte Tabellen.



Öffnen Sie die Beispiel-Datenbank.

Legen Sie im Datenbankfenster mit dem Befehl ABFRAGEENTWURF (Menüband ERSTELLEN, Rubrik ABFRAGEN) eine neue Abfrage an. Das dann erscheinende Dialogfenster "Tabelle anzeigen" können Sie schließen.

Wählen Sie anschließend die SQL-ANSICHT bei ANSICHT im Menüband ABFRAGETOOLS, Rubrik ERGEBNISSE.

Geben Sie obigen SQL-Ausdruck ein.⁷

Schließen Sie das Fenster und speichern Sie die Abfrage unter dem Abfragenamen „NSG_WM-Maßnahmen-2001“ ab.

Starten Sie die Abfrage „NSG_WM-Maßnahmen-2001“ durch Doppelklick oder durch Klick auf den Menüpunkt ÖFFNEN im Kontextmenü (rechte Maustaste).

Sie können die Abfrage jederzeit wieder in der SQL-Entwurfsansicht bearbeiten (Menüband START, Rubrik ANSICHT, Icon SQL-ANSICHT). Modifizieren Sie die SQL Abfrage und machen Sie sich durch weiteres Probieren mit den Möglichkeiten von SQL vertraut.



In der Entwurfsansicht (Menüband START, Rubrik ANSICHT, Icon ENTWURFSANSICHT) bietet MS Access die Möglichkeit, Abfragen ohne Eingabe von SQL zu definieren.

⁷

Sollte dies über Kopieren/Einfügen geschehen, entfernen Sie etwa durch die Silbentrennung eingefügte Trennstiche wieder aus dem Statement!



Übung: Bericht erstellen mit MS Access



Mit Hilfe von Berichten können Tabellen oder die Ergebnisse von Abfragen gestaltet und ausgedruckt werden.

Starten Sie den BERICHTS-ASSISTENTEN im Menüband ERSTELLEN, Rubrik BERICHTE.

Wählen Sie bei TABELLEN/ABFRAGEN die Abfrage „NSG_WM-Maßnahmen-2001“ aus und bei VERFÜGBARE FELDER alle Felder durch Klick auf .

Klicken Sie auf die Schaltfläche WEITER.

Bei der Frage „Wie sollen Ihre Daten angezeigt werden?“, markieren Sie „nach Maßnahme“.

Klicken Sie auf die Schaltfläche WEITER.

Fügen Sie *keine* Gruppierungsebene hinzu.

Klicken Sie auf die Schaltfläche WEITER.

Wählen Sie für die Sortierreihenfolge das Feld „Maßn_Nr“.

Klicken Sie auf die Schaltfläche WEITER.

Wählen Sie als Layout „Tabellarisch“ und als Orientierung „Querformat“. Aktivieren Sie auch die Option „Feldbreite so anpassen, dass alle Felder auf eine Seite passen“.

Klicken Sie auf die Schaltfläche WEITER.

Wählen Sie ein Format, das Ihnen gefällt.

Klicken Sie auf die Schaltfläche WEITER.

Geben Sie Ihrem Bericht den Titel „NSG_WM Maßnahmen 2001“ und klicken Sie auf die Schaltfläche FERTIG STELLEN.



Der aktuelle Bericht kann nun ausgedruckt oder in der Entwurfsansicht bearbeitet werden. Im Datenbankfenster unter Objekt „Berichte“ ist der erstellte Bericht jederzeit abrufbar.



Übung: Auswahlabfrage mit OpenOffice Base



An unserem Beispiel soll eine Tabellenverknüpfungs-Abfrage mit SQL erläutert werden.



Kopieren Sie die für Sie vorbereiteten Dateien auf Ihre Festplatte. Diese stehen im Modul „Datenbank 1“ des Moodle-Kurses „EDL – Einführung in die Datenverarbeitung I“ unter „Übung zum Thema Datenbankentwurf“ zum Download bereit (ZIP-Datei). Entpacken Sie die Übungsdateien aus der Datei dbentwurf.zip in Ihr Arbeitsverzeichnis.

Für das Jahr 2001 und für den Projektbereich Weichser Moos (NSG_WM) sollen sämtliche Maßnahmen aufsteigend sortiert aufgelistet werden und folgende Attribute angezeigt werden: Maßn_Nr, Projektkürzel, FL_Name, Jahr, AN_Name, AN_Vorname, Größe, Kosten und Maßnahmetyp.

Für diese Abfrage müssen also 5 Tabellen miteinander verknüpft werden.



Obige Abfrage in SQL, hier im OpenOffice.org Base-Dialekt.

In Base werden die JOINS nicht mit dem Schlüsselwort „JOIN“ angegeben, sondern in WHERE-Anweisungen verpackt. Außerdem muss die Verwendung von Anführungszeichen beachtet werden: *doppelte* Anführungszeichen bei den Feldnamen, aber *einfache* Anführungszeichen bei Zeichenketten in WHERE-Bedingungen ('NSG_WM', s.u.)!

Kursiver Text dient der Erläuterung und ist nicht Bestandteil der SQL Syntax:

Auswahl der anzuzeigenden Felder „Tabellenname.Feldname“ mit SELECT:

```
SELECT "Auftragnehmer"."AN_Name", "Auftragnehmer"."AN_Vorname",  
"Fläche"."FL_Name", "Maßnahme"."Maßn_Nr", "Maßnahme"."Größe", "Maß-  
nahme"."Kosten", "Maßnahme"."Jahr", "Maßnahmetyp"."Maßnahmetyp",  
"Projekt"."Projektkürzel"
```

Auswahl der Tabellen mit FROM. INNER JOIN muss hier nicht angegeben werden. AS legt sog. „Alias-Namen“ für die verwendeten Tabellen fest:

```
FROM "Auftragnehmer" AS "Auftragnehmer", "Maßnahme" AS "Maßnahme",  
"Fläche" AS "Fläche", "Maßnahmetyp" AS "Maßnahmetyp", "Projekt" AS  
"Projekt"
```

Mit WHERE wird die Bedingung für die Auswahl der anzuzeigenden Feldwerte definiert. Hier werden die verschiedenen JOINS durch die Aufzählung der Bezüge zwischen den Primär- und Fremdschlüsselfeldern als WHERE-Bedingungen abgebildet:

```
WHERE "Auftragnehmer"."AN_Nr" = "Maßnahme"."AN_Nr"  
AND "Fläche"."FL_Nr" = "Maßnahme"."FL_Nr"  
AND "Maßnahmetyp"."Mtyp_Nr" = "Maßnahme"."Mtyp_Nr"  
AND "Projekt"."Projekt_Nr" = "Fläche"."Projekt_Nr"  
AND ( "Projekt"."Projektkürzel" ) = 'NSG_WM'  
AND ( "Maßnahme"."Jahr" ) = 2001
```

Aufsteigende Sortierung („ASC“) der Datensätze durch ORDER BY nach Werten des Attributes „Maßn_Nr“. Der Abschluss des SQL Befehls durch ein Semikolon kann hier entfallen:

```
ORDER BY "Maßnahme"."Maßn_Nr" ASC
```



Diese SQL Abfrage können Sie nun mit der Beispiel-Datenbank testen („dbentwurf2009.odt“, enthalten in dbentwurf.zip). Die Datenbank enthält vorbereitete und bereits in Beziehung gesetzte Tabellen.



Öffnen Sie die Beispiel-Datenbank.

Legen Sie im Datenbankfenster unter der Rubrik ABFRAGEN eine **neue Abfrage** an. Wählen Sie dabei die Option ABFRAGE IN SQL-ANSICHT ERSTELLEN.

Geben Sie obigen SQL-Ausdruck ein.⁸

Speichern Sie die Abfrage unter dem Abfragenamen „NSG_WM-Maßnahmen-2001“ ab.

Durch Klick auf das Icon ABFRAGE AUSFÜHREN im Abfrage-Entwurfsfenster können Sie die Abfrage auf die Tabellen anwenden. Das Ergebnis wird im oberen Teil des Abfrage-Fensters angezeigt, im Unteren bleibt das SQL-Statement sichtbar.

Sie können auch das Abfrage-Fenster schließen und die Abfrage „NSG_WM-Maßnahmen-2001“ aus dem Datenbankfenster durch Doppelklick auf den Abfragenamen oder durch Klick auf den Menüpunkt ÖFFNEN im Kontextmenü (Klick mit rechter Maustaste) ausführen.



Sie können die Abfrage jederzeit wieder in der Entwurfsansicht bearbeiten. Diese ist über den Menüpunkt BEARBEITEN oder das Kontextmenü, jeweils unter IN SQL-ANSICHT BEARBEITEN erreichbar. Modifizieren Sie die SQL Abfrage und machen Sie sich durch weiteres Probieren mit den Möglichkeiten von SQL vertraut.



In der Entwurfsansicht – Menü BEARBEITEN und im Kontextmenü, jeweils unter BEARBEITEN – bietet OpenOffice Base die Möglichkeit, Abfragen ohne Eingabe von SQL zu definieren.

⁸ Sollte dies über Kopieren/Einfügen geschehen, entfernen Sie etwa durch die Silbentrennung eingefügte Trennstreiche wieder aus dem Statement!



Übung: Bericht erstellen mit OpenOffice.org Base



Mit Hilfe von Berichten können Tabellen oder die Ergebnisse von Abfragen gestaltet und ausgedruckt werden.



Klicken Sie im Datenbankfenster auf das Objekt „Berichte“ und starten Sie mit Klick auf „Bericht unter Verwendung des Assistenten erstellen...“ den Bericht-Assistenten.

1. Schritt: Felderauswahl:

Wählen Sie bei TABELLEN UND ABFRAGEN die Abfrage „NSG_WM-Maßnahmen-2001“ aus und bei VERFÜGBARE FELDER alle Felder durch Klick auf .

Klicken Sie auf die Schaltfläche WEITER.

2. Schritt: Felder beschriften

Hier haben Sie die Möglichkeit, anstelle der Feldnamen, die ja bisweilen nur aus Abkürzungen bestehen, für den Bericht aussagekräftige Feldbeschriftungen anzugeben.

Klicken Sie auf die Schaltfläche WEITER.

3. Schritt: Gruppierungen

Die Datensätze in einem Bericht lassen sich nach den Werten in einem oder mehreren Feldern in Gruppen gliedern. Lassen Sie für diesen ersten Bericht die Gruppierung weg.

Klicken Sie auf die Schaltfläche WEITER.

4. Schritt: Sortieroptionen

Die Daten lassen sich nach mehreren Feldern auf- oder absteigend sortieren. Sortieren Sie die Daten nach dem Feld „Maßn_Nr“, aufsteigend. ... und WEITER.

5. Schritt: Auswahl des Layouts

In dieser Rubrik lassen sich die Daten einerseits und der Hintergrund sowie Kopf- und Fußzeile andererseits formatieren. Sobald im Assistenten eine Formatierung ausgewählt wird, kann man die Wirkung in der dahinter geöffneten Vorschau sehen. Wählen Sie ein Layout aus! ... und WEITER.

6. Schritt: Bericht erstellen

Nachdem Sie Ihrem Bericht einen Namen gegeben haben, haben Sie die Wahl, einen „statischen“ Bericht abzuspeichern – er wird die Daten enthalten, die zum Zeitpunkt der Erstellung vorhanden waren – oder einen „dynamischen“ Bericht. Wenn „dynamischer Bericht“ gewählt ist, wird der Bericht als Dokumentvorlage gespeichert. Wird er später wieder aufgerufen, enthält der Bericht immer die gerade aktuellen Daten.

Markieren Sie JETZT BERICHT ERSTELLEN und klicken auf FERTIGSTELLEN. Der Bericht wird nun in OpenOffice Writer geöffnet.



Der aktuelle Bericht kann nun ausgedruckt werden. Im Datenbankfenster unter Objekt „Berichte“ ist der erstellte Bericht jederzeit abrufbar und kann in der Entwurfsansicht erneut bearbeitet werden.