Unterrichtsmaterial

Als Fundgrube für Abfragen empfehle ich das Begleitmaterial zu „Microsoft Office – Class in a Box“ vom Cornelsen – Verlag. Eine Datenbank „**Biblio**“ wird dort mitgeliefert. Sie enthält über 1500 Eintragungen zu Büchern einer Schülerbücherei. Im Begleitheft finden Sie viele Aufgaben. **ideal für den Einstieg**

Videocenter, eine Videothek – Datenbank mit aktuellen Titeln (http://videocenter.schule.de/)

Das **SQL-Tutorial** des Lichtenberg-Gymnasiums, Darmstadt (unter http://www.schulserver.hessen.de/darmstadt/lichtenberg/SQLTutorial/home.html). Grundlagen sind die „CIA – Datenbank“ mit Informationen zu vielen Ländern der Erde und die „Nordwind – Datenbank“, die bei Microsoft Access als Beispieldatenbank mitgeliefert wird.

Klett: Informatik 2, Kapitel III, IV und V

Schöningh: Informatik 3, Kapitel 3 Datenbanken

23. Datenbanken

Einführung –Beispiele Telefonbuch, Online – Bestellungen (Versandhandel), Google, Handy

Geschichte: Karteikarten (→ Entity), Lochkarten (Hollerith), COBOL, DBase, Access

Eine **Datenbank** (DB) ist eine systematisch strukturierte, langfristig verfügbare Sammlung von Daten.
Eine **Entität** ist ein eindeutig identifizierbares Objekt oder ein eindeutig identifizierbarer Sachverhalt der realen Welt oder der Vorstellungswelt.
Die Entität wird durch ihre **Attribute** / Eigenschaften bestimmt. Die bei einer bestimmten Entität auftretenden Werte sind Attributwerte.

Merkmale des relationalen Modells

- Alle Angaben über Entitätstypen und Beziehungstypen werden in zweidimensionalen Tabellen (Relationen) festgehalten.
- Die Spalten der Tabellen entsprechen den Attributen. Die Anordnung der Spalten ist beliebig. Die Spalten sind durch ihre Namen eindeutig bestimmt.
- Die Zeilen entsprechen den Entitäten. Die Anordnung der Zeilen innerhalb der Tabelle ist beliebig.
- Jede Zeile ist über einen Schlüssel eindeutig identifizierbar.
- Jede Relation besitzt einen eindeutigen Namen.

Aufbau einer Datenbank

Jede Datenbank besteht aus einer oder (meistens) mehreren **Tabellen**.

Eine Tabelle setzt sich aus Spalten und Zeilen zusammen.

Im Gegensatz zu Excel oder Word ist jede Spalten mit einer besonderen Bedeutung verbunden:

Die **Spalten** entsprechen den **Attributen** (Eigenschaften) und werden auch **Felder** genannt; in der Tabellenüberschrift steht eine Bezeichnung für das Attribut – der Feldname, in den Zeilen darunter die Attributwerte für die einzelnen Entitäten.

Eine Tabellenzeile wird **Datensatz** genannt; sie enthält alle Attributwerte zu einer Entität. Hier: alle Eigenschaften eines Lesers.

Feld ↓

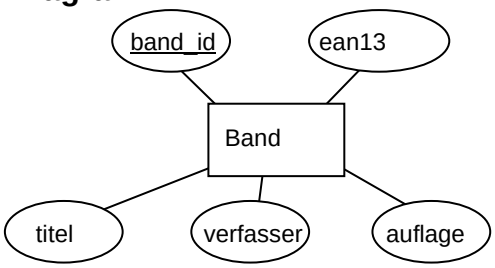
	benutzer_id	nachname	vorname	klasse	Geburtsdatum	Ort	PLZ	Strasse
	1405	Pütz	Jonathan	06D	16.08.1999	Solingen	42659	Balkhausen 74
	1406	Pütz	Nane	06D	05.04.2000	Solingen	42659	Bülowstr. 13
	1407	Richter	Erick	06D	17.10.2000	Solingen	42659	Höhrath 113a
	1408	Schüttler	Maxim	06D	22.04.2000	Solingen	42659	Ohmweg 7
	1409	Seefeld	Rebekka	06D	01.09.2000	Solingen	42659	Schlicker Weg 51
Datensatz →	1410	Semmler	Cornelius	06D	24.05.2001	Solingen	42659	Pfaffenberger Weg 113
	1411	Sliwinski	Katrin	06D	16.04.2000	Solingen	42659	Am Buschberg 27
	1412	Speitmann	Jil	06D	07.01.2000	Solingen	42659	Beckmannstr. 71
	1413	Temmel	Carolin	06D	12.09.2000	Solingen	42659	Balkhauser Weg 138
	1414	Warneke	Johanna	06D	07.04.2000	Solingen	42659	Hertzweg 1
	1415	Weber	Tristan	06D	21.04.2000	Solingen	42659	Eichenstr. 59
	1416	Wehning	Kai	06D	20.02.2000	Solingen	42659	Gabelsberger Str. 85
	1417	Wodicka	Jasmin	06D	13.10.2000	Solingen	42651	Akazienweg 13

23.1 Darstellung

Am Beispiel der Büchertabelle werden die verschiedenen benutzten Darstellungsweisen benutzt und Begriffe verdeutlicht.

Entity -	ein einzelnes Buch – jedes Buch	→ Objekt
Entitytyp	Aufbau der Tabelle, Set von Eigenschaften(Attributen)	→ Klasse

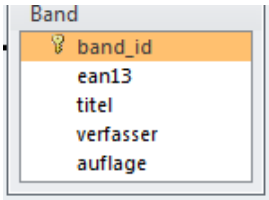
Entity – Relationship – Diagramm:



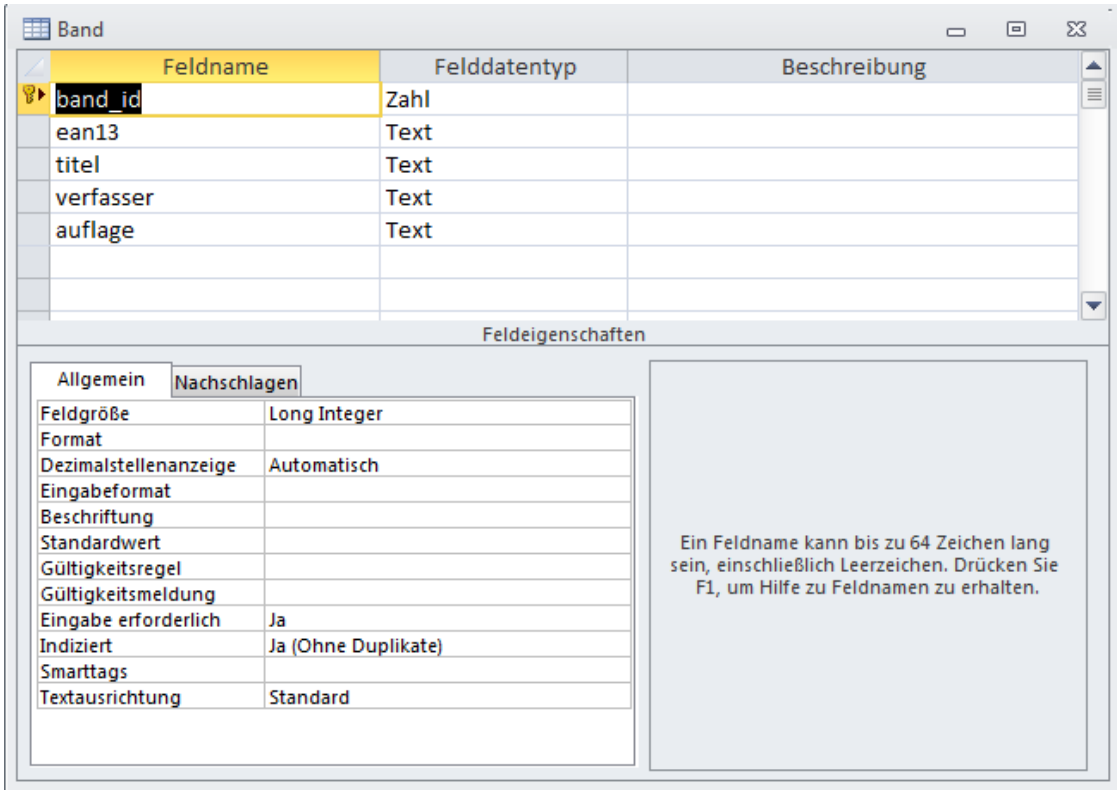
Datenbankschema

Band (band_id, ean13, titel, verfasser, auflage) /Primärschlüssel unterstrichen

Teil eines Beziehungsdiagramms (Open Office / Access)



Tabellenentwurf



Band

Feldname	Felddatentyp	Beschreibung
band_id	Zahl	
ean13	Text	
titel	Text	
verfasser	Text	
auflage	Text	

Feldeigenschaften

Allgemein | Nachschlagen

Feldgröße	Long Integer
Format	
Dezimalstellenanzeige	Automatisch
Eingabeformat	
Beschriftung	
Standardwert	
Gültigkeitsregel	
Gültigkeitsmeldung	
Eingabe erforderlich	Ja
Indiziert	Ja (Ohne Duplikate)
Smarttags	
Textausrichtung	Standard

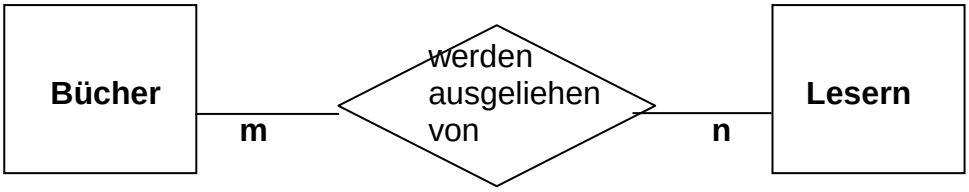
Ein Feldname kann bis zu 64 Zeichen lang sein, einschließlich Leerzeichen. Drücken Sie F1, um Hilfe zu Feldnamen zu erhalten.

und schließlich die Tabelle selbst

Band				
band_id	**ean13**	**titel**	**verfasser**	**auflage**
5	9783125177000	Pons Micro-Robert en poche	Robert, Paul	1. Aufl., Nachdr.1989
6	9783507107229	Physik	Dorn-Bader	2000
7	9783766157645	Der gallische Krieg	Friedrich Maier u. Heinrich Voit	1990
8	9783127322101	Lambacher-Schweizer 11	Manfred Baum	1. Aufl. 2000
9	9783791504544	Drachenreiter	Cornelia Funke	1997
10	9783491756519	Basiswissen Abitur - Philosophie	Michael Wittschier	2006
11	9783458322504	Goethes Schr. zur Naturwiss.	Goethe, Johann Wolfgang von hrsg. von Horst Günther	1. Aufl.1981
12	9783124560902	Historisch-politische Weltkunde	Asmut Brückmann	1. Aufl., 1. Dr. 1993
13	9783464410042	Texte, Themen und Strukturen		2. Aufl., 1. Dr. 2006
14	9783507107205	Physik Gymnasium Sek II 11	Ausgabe Dorn-Bader	Dr. A, 1 1998
15	9783766134110	Chemie 2000+ Bd. 1	Tausch von Wachtendonk	1. Aufl., 1. Dr. 2001

Modellierung

ER – Diagramme und Kardinalitäten

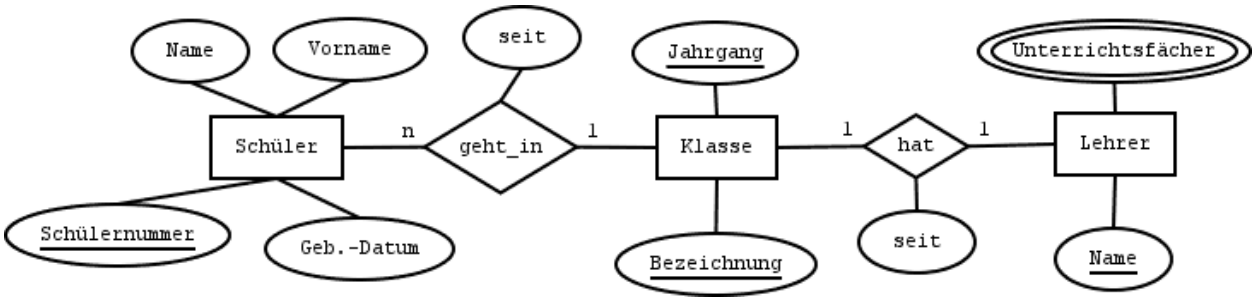


Eine Realisierung durch einfaches Hinzufügen eines Feldes in der Leser- oder Büchertabelle ist jetzt nicht mehr möglich.

m:n – Beziehungen werden durch Beziehungstabellen realisiert.

Aufgabe:

Hier ein ER – Diagramm für ein nahe liegendes Beispiel (Schule)



Was wird hier implizit vorausgesetzt ?
Geben Sie ein sinnvolleres ER – Modell an!

<u>Aufgabe :</u>

Modellieren Sie die folgenden Situationen in ER – Diagrammen. Geben Sie jeweils die zusätzlichen Bedingungen an, die Sie in ihrem Modell voraussetzen. Notieren Sie die wichtigsten Attribute und die Kardinalitäten.

a) Musiker - CD - Song

Notieren Sie in den weiteren Aufgaben nur die Kardinalitäten

b) Fan - Fußballverein
c) Spieler - Mannschaft(Liga) - Verein
d) Arzt - Patient
e) Abgeordneter - Partei
f) Schauspieler - Film - Filmverleih

Projektablauf:

1. Anforderungen

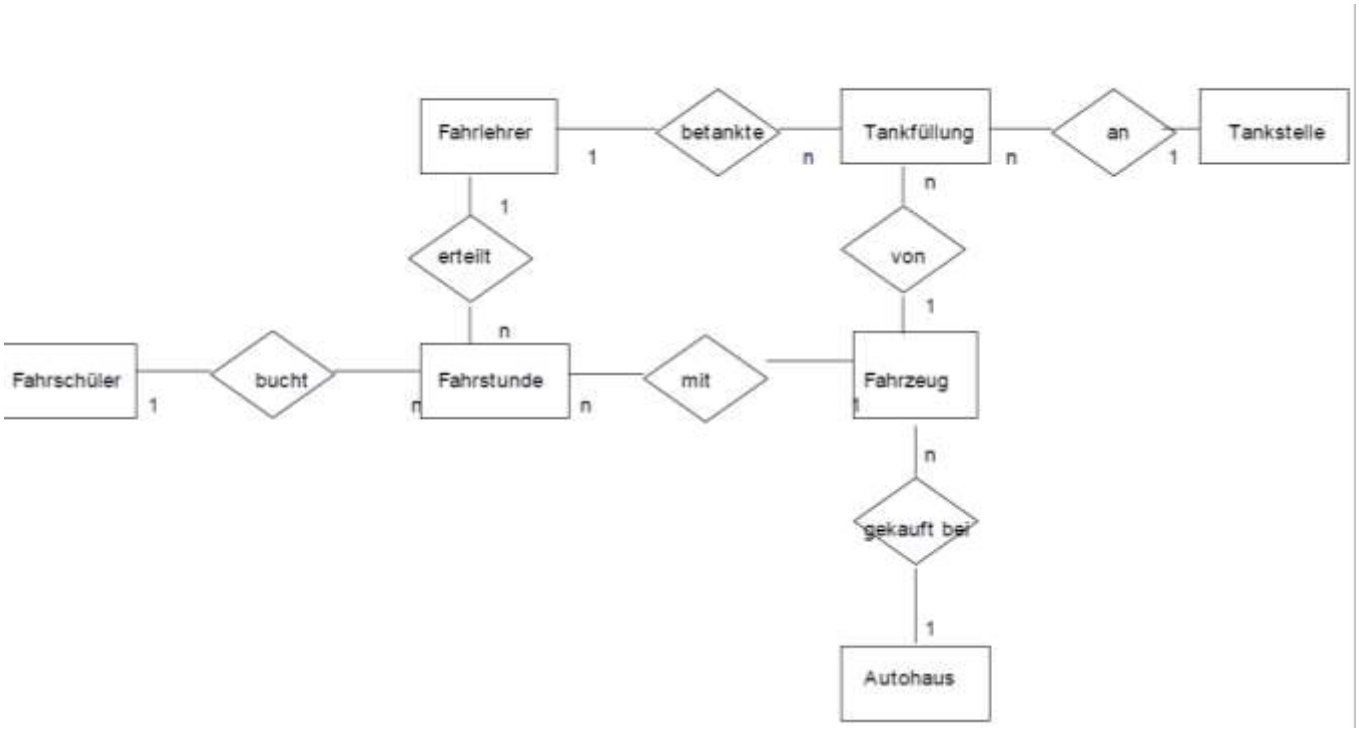
Durch den Zusammenschluss von fünf Fahrschulen in Solingen ist ein Unternehmen mit 11 Fahrlehrern und 15 Fahrzeugen entstanden. Um organisatorisch den Überblick über das Unternehmen zu behalten, soll die Firma Itter – Datensysteme eine Datenbank erstellen, die folgende Anforderung erfüllt:

Zur **Terminplanung** und **Abrechnung** soll festgehalten werden, wann ein Fahrschüler eine Fahrstunde bei einem Fahrlehrer hat und welches Fahrzeug verwendet wird.

Es soll erfasst werden, wann und über welches Autohaus die Fahrzeuge beschafft worden sind.

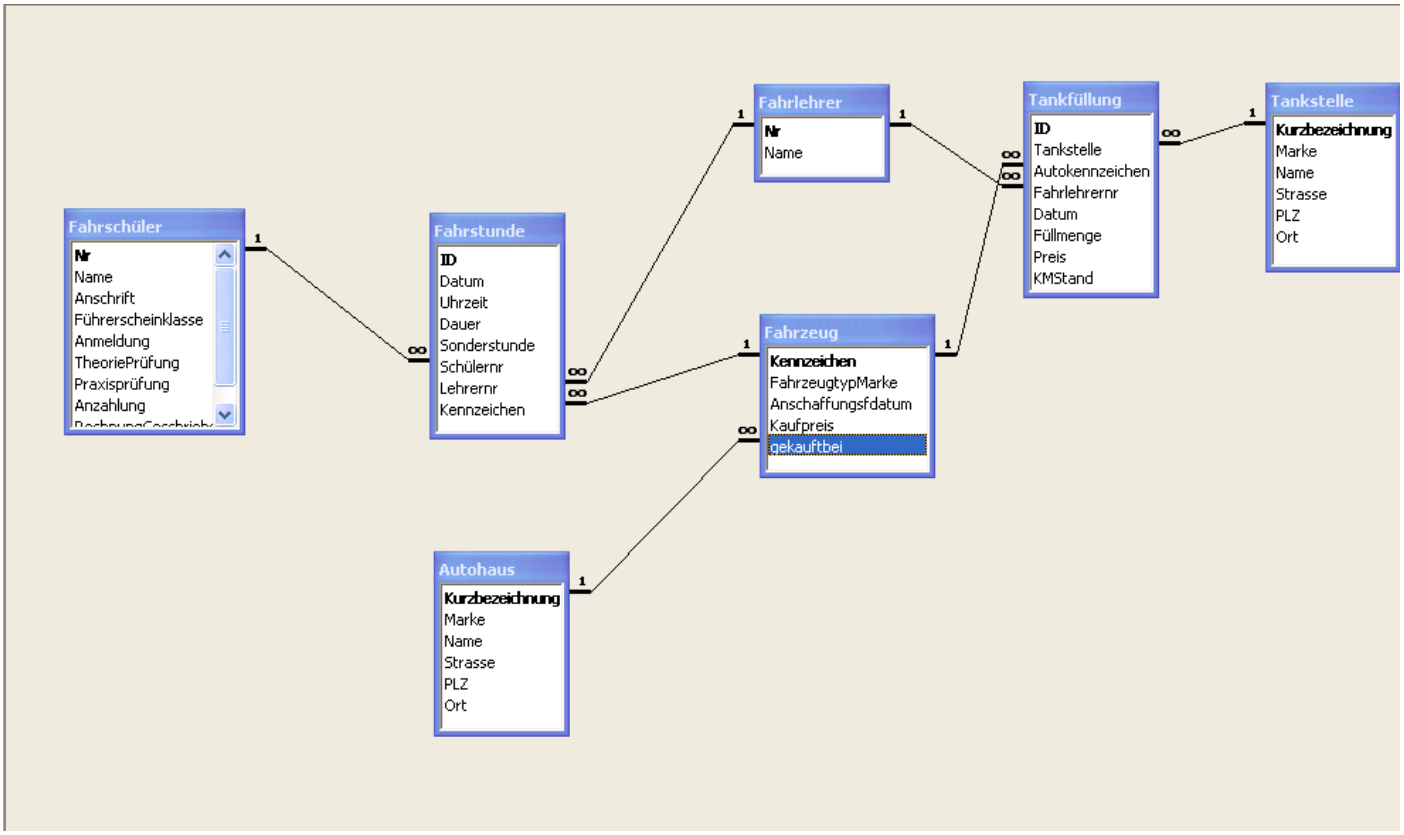
Zur Kostenkontrolle sollen die Daten der einzelnen Betankungsvorgänge (auch der Fahrlehrer, der getankt hat) gespeichert werden.

2. ER - Diagramm



3. Datenbankschema

Das Datenbankschema ergibt sich unmittelbar aus dieser Darstellung: (Access – Beziehungsdiagramm)



Aufgabe dazu:

s. Abi 2010 Teil a)

Normalisierung von Datenbanken

nach: Tino Hempel - Richard-Wossidlo-Gymnasium Ribnitz-Damgarten
(http://www.tinohempel.de/info/info/datenbank/normalisierung.htm)

Begriffe der Normalisierung

Die Relationen (Tabellen) einer konkreten Datenbankanwendung lassen sich auf zwei Wege erzeugen:

- über die Schrittfolge: konzeptioneller →logischer →physischer Entwurf,
  also Diskurs →ERM →relationales Modell →Datenbank,

- über die Normalisierung einer Relation, die alle für die Datenbank relevanten Attribute enthält.

Normalisierung ist somit eine weitere Möglichkeit, relationale Datenbanken zu erhalten.

Ein exakter Modellentwurf (ERM/RM) macht eine Normalisierung eigentlich überflüssig.

26.1 Gründe für die Normalisierung

- Vermeidung von unerwünschten Anomalien (Lösch-, Einfüge- und Änderungsanomalie)
- Vermeidung von überflüssigen Informationen (Redundanz)
- Zwang zum systematischen Entwurf einer Datenbank
- bessere Übersichtlichkeit für Benutzer und Programmierer

26.3 Erste Normalform

Eine Tabelle befindet sich in der ersten Normalform, wenn alle ihre Attribut nur einfache Attributwerte aufweisen, d. h. wenn jeder Kreuzungspunkte von Zeile und Spalte höchstens einen Wert besitzt (Bezeichnung: **atomar**).

Die Datenbank einer Firma als nicht normalisierte Relation:

PersNr	**Name**	**Vorname**	**AbtNr**	**Abteilung**	**Projekt Nr**	**Beschreibung**	**Zeit**
1	Lorenz	Sophia	1	Personal	2	Verkaufspromotion	83
2	Hohl	Tatjana	2	Einkauf	3	Konkurrenzanalyse	29
3	Willschrein	Theodor	1	Personal	1,2,3	Kundenumfrage, Verkaufspromotion, Konkurrenzanalyse	140, 92, 110
4	Richter	Hans-Otto	3	Verkauf	2	Verkaufspromotion	67
5	Wiesenland	Brunhilde	2	Einkauf	1	Kundenumfrage	160

Relation Firma in der ersten Normalform:

PersNr	**Name**	**Vorname**	**AbtNr**	**Abteilung**	**ProjektNr**	**Beschreibung**	**Zeit**
1	Lorenz	Sophia	1	Personal	2	Verkaufspromotion	83
2	Hohl	Tatjana	2	Einkauf	3	Konkurrenzanalyse	29
3	Willschrein	Theodor	1	Personal	1	Kundenumfrage	140
3	Willschrein	Theodor	1	Personal	2	Verkaufspromotion	92
3	Willschrein	Theodor	1	Personal	3	Konkurrenzanalyse	110
4	Richter	Hans-Otto	3	Verkauf	2	Verkaufspromotion	67
5	Wiesenland	Brunhilde	2	Einkauf	1	Kundenumfrage	160

Die Redundanz nimmt zu. Der bisherige Primärschlüssel verliert seine Eindeutigkeit, es muss ein Neuer gefunden werden.
Festlegung: PersNr und ProjektNr bilden den neuen Primärschlüssel.

26.4 Zweite Normalform

Eine Relation befindet sich in der zweiten Normalform, wenn

sie in der ersten Normalform ist und

jedes Nicht-Schlüssel-Attribut vom gesamten Identifikationsschlüssel funktional abhängig ist,
nicht aber bereits von Teilen des Gesamtschlüssels.

Schlüssel der gegebenen Tabelle festlegen, falls dieser nur aus einem Attribut besteht, so liegt bereits 2. NF vor.

Tabelle auf Teilschlüsselabhängigkeiten untersuchen, d. h. folgt aus Schlüsselteilattributen bereits weitere Nichtschlüsselattribute.

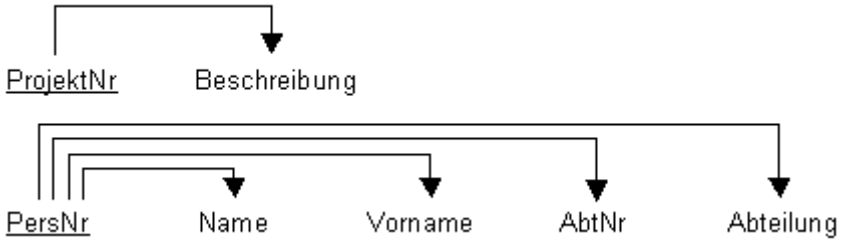
Neue Tabelle bilden, die den Teilschlüssel, der nun Primärschlüssel der neuen Tabelle wird, und die allein von ihm funktional abhängigen Nichtschlüsselattribute enthält. Streichen der abhängigen Nichtschlüsselattribute aus der Ausgangstabelle

Vorgang ab 2. wiederholen, bis alle Nichtschlüsselattribute vom gesamten Schlüssel funktional abhängig sind.

Beispiel:

Aus dem Teilschlüssel ProjektNr lässt sich bereits eindeutig auf Beschreibung schließen. Aus PersNr lässt sich eindeutig auf Name, Vorname, AbtNr und Abteilung schließen.
Somit sind zwei neue Tabellen Projekt und Personal zu erzeugen und das verbleibende Tabelle Firma so zu überarbeiten, dass die Attribute Beschreibung, Name, Vorname, AbtNr und Abteilung gelöscht werden.

Abhängigkeiten:



Projekt

ProjektNr	Beschreibung
2	Verkaufspromotion
3	Konkurrenzanalyse
1	Kundenumfrage

Personal

PersNr	Name	Vorname	AbtNr.	Abteilung
1	Lorenz	Sophia	1	Personal
2	Hohl	Tatjana	2	Einkauf
3	Willschrein	Theodor	1	Personal
4	Richter	Hans-Otto	3	Verkauf
5	Wiesenland	Brunhilde	2	Einkauf

Firma

PersNr	ProjektNr	Zeit
1	2	83
2	3	29
3	1	140
3	2	92
3	3	110
4	2	67
5	1	160

26.5 Dritte Normalform

Eine Relation befindet sich in der dritten Normalform, wenn

sie in der zweiten Normalform ist und

die Nichtschlüsselattribute voneinander nicht funktional abhängig sind, d.h. es existieren keine transitiven Abhängigkeiten.

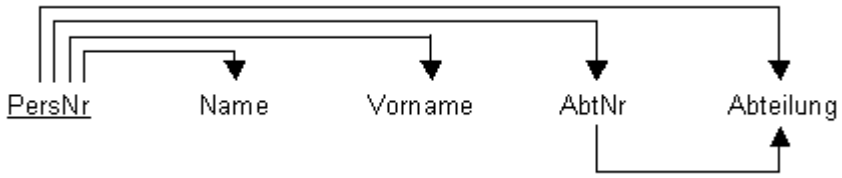
Ein Attribut Z ist vom Attribut X über das Attribut Y genau dann transitiv abhängig, wenn gilt: aus X folgt Y und aus Y folgt nicht X und aus Y folgt Z.

Relation auf transitive Abhängigkeiten untersuchen, d. h. folgt aus einem Nichtschlüsselattribut eindeutig ein weiteres Nichtschlüsselattribut.

Neue Tabelle bilden, die die transitiv abhängigen Attribute enthält.
Schlüssel der neuen Tabelle festlegen
Alle in die neue Tabelle übernommenen Attribute aus der alten löschen, mit Ausnahme des Attributes, das in der neuen Relation Schlüssel ist.

Vorgang ab 2. wiederholen, bis alle Nichtschlüsselattribute vom gesamten Schlüssel funktional abhängig sind.

Schreibweise zur Erkennung von transitiven Abhängigkeiten:



Aus der PersNr. folgt Name, Vorname, AbtNr und die Abteilung.
Aber aus der AbtNr. folgt bereits die Abteilung!
Also werden die Attribute AbtNr. und Abteilung in eine neue Tabelle gebracht.

Abteilung

AbtNr.	Abteilung
1	Personal
2	Einkauf
3	Verkauf

In der Tabelle Personal verschwindet die Spalte „Abteilung“

Personal

PersNr	Name	Vorname	AbtNr.
1	Lorenz	Sophia	1
2	Hohl	Tatjana	2
3	Willschrein	Theodor	1
4	Richter	Hans-Otto	3
5	Wiesenland	Brunhilde	2

Die anderen beiden Tabellen bleiben unverändert:

Projekt

ProjektNr	Beschreibung
2	Verkaufspromotion
3	Konkurrenzanalyse
1	Kundenumfrage

Firma

PersNr	ProjektNr	Zeit
1	2	83
2	3	29
3	1	140
3	2	92
3	3	110
4	2	67
5	1	160

Aufgabe 7:

Die Autovermietung „Dornbusch“ mit Zentrale in Solingen – Schlebusch verwaltet die Entleihdaten in einer Excel – Tabelle. Ein Auszug:

Name	Adresse	Abfahrt	Fahrzeug	Auto marke	Tarif	Rückgabe datum	Rückgabeort	Tage
Fritz Schmitz	42777 Solingen Bartstr. 7	23.11.2011	SG-DB 23	Benz	4	25.11.2011	Solingen	2
Chantal Meyer	43601 Haan Windhövel 1	23.11.2011	SG-DB 77	Wind	2	24.11.2011	Vohwinkel	1
Fritz Schmitz	42777 Solingen BartStr. 7	27.11.2011	SG-DB 23	Benz	4	29.11.2011	Solingen	2
Jacqueline Pauls	42699 Ohligs Baustr. 44	26.11.2011	SG-DB 123	Baguette	3	29.11.2011	Solingen	3
Jacqueline Pauls	43601 Haan Sudetenstr. 1	27.11.2011	SG-DB 77	Wind	2	27.11.2011	Haan	1

Es soll eine Datenbank erstellt werden.
Erstellen Sie zunächst (in mehreren Schritten) ein Datenbankschema.

a) Der Primärschlüssel soll aus Name und Abfahrt bestehen. Erläutern Sie, weshalb der Name hier ungeeignet ist, ergänzen Sie eine weitere Spalte, um die Person eindeutig zu machen und definieren dann einen neuen Primärschlüssel in Kombination mit Abfahrt.

b) Bringen Sie die Tabelle in die erste Normalform!

c) Wo verletzt die Tabelle die zweite Normalform? Ändern Sie das Datenbankschema, so dass die zweite Normalform erfüllt ist.

d) Dasselbe mit der dritten Normalform.

e) Bei der Mietdauer handelt es sich nicht um Grunddaten. Welche Abhängigkeit besteht?

SQL

einfach: Abfragen zu einer Tabelle

```
SELECT titel, verfasser, auflage
FROM Band
WHERE (auflage =„2006“) OR (titel like “Chemie*“)
```

titel	verfasser	auflage
Systematik-Poster: Botanik	Andreas Bresinsky	2006
Chemie : das Basiswissen der Chemie	Charles E. Mortimer	2010
Chemie - einfach alles	Peter W. Atkins	2006
Biologie der Pflanzen	Peter H. Raven	2006

“Vokabeln”

SELECT (DISTINCT) ……FROM ….
WHERE
ORDER BY …. ASC/DESC
NULL
LIKE, BETWEEN, IN, IS NULL

mehrere Tabellen (Beziehung ??)

```
SELECT Leser.nachname, Ausleihe.ausleih_datum
FROM Leser, Ausleihe
```

kartesisches Produkt

(Inner) Join

Die Lösung:
Es wird von vorneherein mit Hilfe einer Bedingung gesucht – meistens Übereinstimmung von Feldinhalten (Äqui – Join) – die den Umfang der Ergebnisrelation in Grenzen hält.

```
SELECT Leser.Nachname, Ausleihe.ausgeliehen
FROM Leser INNER JOIN Ausleihe ON Leser.LeserNr =
Ausleihe.LeserNr
```

Die SQL-Anweisung

```
SELECT Leser.Nachname, Ausleihe.ausgeliehen
FROM Leser, Ausleihe
WHERE Leser.LeserNr = Ausleihe.LeserNr
```

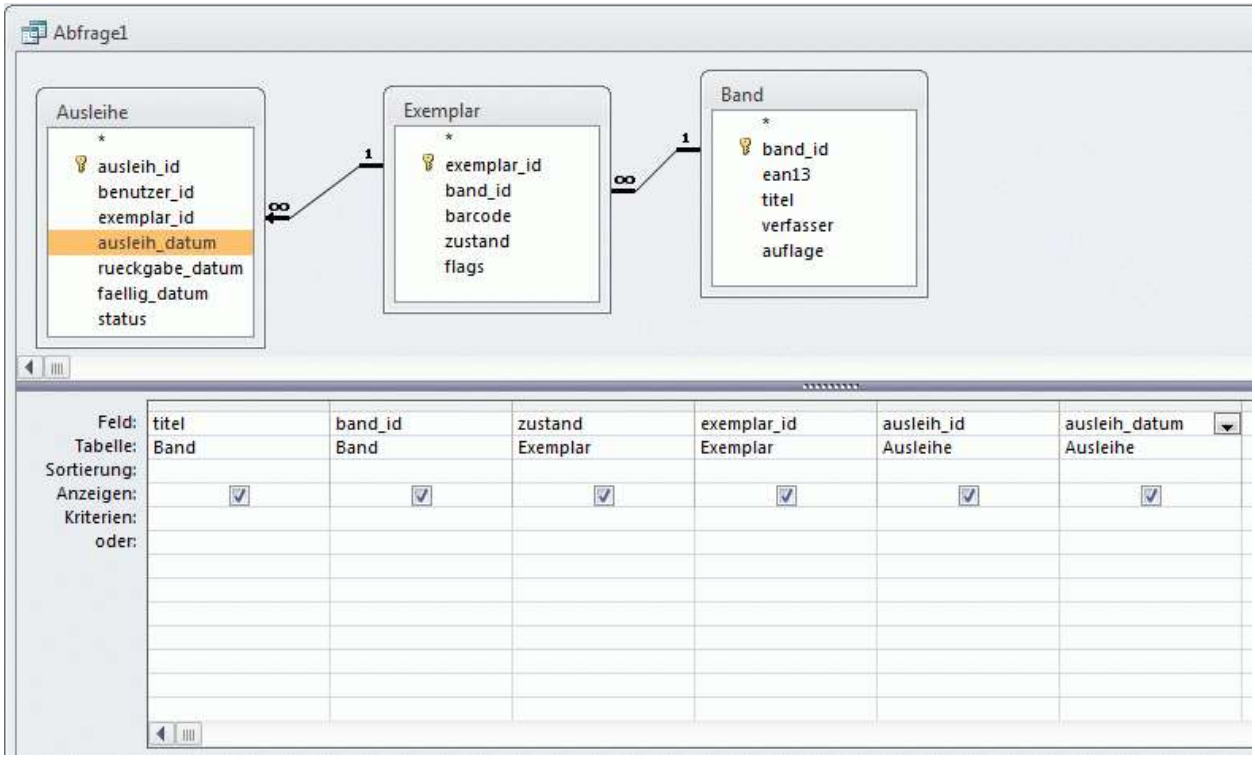
wird übrigens von den DBMS genauso umgesetzt wie der JOIN, ist also in der Praxis gleichwertig.

Weil die m:n – Beziehungen durch eine (hier notwendige) Beziehungstabelle in zwei 1:n – Beziehungen aufgeteilt werden, sind die praktisch realisierten Beziehungen fast alle vom Typ 1:n. Beim Join über eine solche 1:n – Beziehung entscheidet die Seite, auf der das n steht, über die Zahl der Datensätze im Join; diese Tabelle wird auch Detailtabelle genannt. Auf der 1 – Seite ist das verwendete Attribut hingegen durchweg der Primärschlüssel.

25.1 Abfragen über mehrere (verbundene) Tabellen (Handling)

Die zwischen den Tabellen bestehenden Beziehungen werden automatisch berücksichtigt, wenn wir eine Abfrage mit Daten aus mehreren Tabellen erstellen Hier für den „Left Join“ von oben

(Datenbankfenster → Abfragen → Neu → Entwurfsansicht)



Abfrage: Ausgeliehene Exemplare

Die angezeigte Abfrage „Ausgeliehene Bücher“ zeigt alle Buchexemplare an mit dem Titel aus dem zugehörigen (gleiche band_id) „Band“. Wenn ein Ausleihe – Datensatz mit gleicher „exemplar_id“ existiert, werden auch die Ausleihdaten angegeben.

In SQL:

```
SELECT [Band].titel, [Band].band_id, Exemplar.zustand,
Exemplar.exemplar_id, Ausleihe.ausleih_id,
Ausleihe.ausleih_datum

FROM [Band] INNER JOIN (Exemplar LEFT JOIN Ausleihe ON
Exemplar.exemplar_id = Ausleihe.exemplar_id) ON [Band].band_id =
Exemplar.band_id;
```

weitere “Vokabeln”

(LEFT/RIGHT) JOIN … ON …

UNION

AS

COUNT, SUM, MAX, MIN

EXISTS

Aufgabe:

Abitur 2010 Teil c)

Teilaufgabe c)

Die Informationen befinden sich alle in der Tabelle AutovermietungsFirma, so dass es sich bei der Abfrage um eine einfache Selektion handelt.

```
SELECT FirmenName, Ansprechpartner, Email
FROM AutovermietungsFirma
WHERE FirmenNr = 12345;
```

Die Tabellen Angebot und LieferwagenTyp müssen über das Verknüpfungsattribut LieferwagenTypNr verknüpft werden. Darüber erfolgt dann eine Auswahl nach der Kategorie „Kombi“. Durch Anwenden der Funktion min wird dann der niedrigste Preis ausgewählt.

```
SELECT min(a.Angebotspreis)
FROM Angebot AS a, LieferwagenTyp AS t
WHERE a.LieferwagenTypNr = t.LieferwagenTypNr
AND t.Kategorie = “Kombi“;
```

Die verlangten Informationen müssen aus den drei Tabellen Angebot, LieferwagenTyp und AutovermietungsFirma geholt werden. Diese werden über die Verknüpfungsattribute
FirmenNr und LieferwagenTypNr verknüpft. Als Bedingung werden dann noch der Wohnort, die Kategorie und die Marke abgefragt. Dabei ist zu beachten, dass wegen
der Priorität die ODER-Verknüpfung zuerst erfolgen muss, also in Klammern steht.

```
SELECT f.TelefonNr, f.FirmenName,t.Marke, a.Angebotspreis
FROM Angebot AS a, LieferwagenTyp AS t, AutovermietungsFirma AS f
WHERE a.FirmenNr = f.FirmenNr
AND a.LieferwagenTypNr = t.LieferwagenTypNr
AND f.Wohnort = “Köln“
AND t.Kategorie = “Minivan“
AND (t.Marke = “Pudel“
OR t.Marke = “Mops“)
ORDER BY a.Angebotspreis;
```

Die angegebene SQL-Anweisung gibt die Liste aller Marken und Modelle der Lieferwagentypen
aus, deren Zuladegewicht größer ist als 2000 kg und für die es im Moment kein Angebot
gibt.