Relationale Datenbanken

Unterrichtsdokumentation

Modulname:	Relationale Datenbanken (RDB) in der Sek 2

Stand: 16. Sep 2020

BSW Berufliches Schulzentrum Wangen

© BS-Wangen

Inhaltsverzeichnis

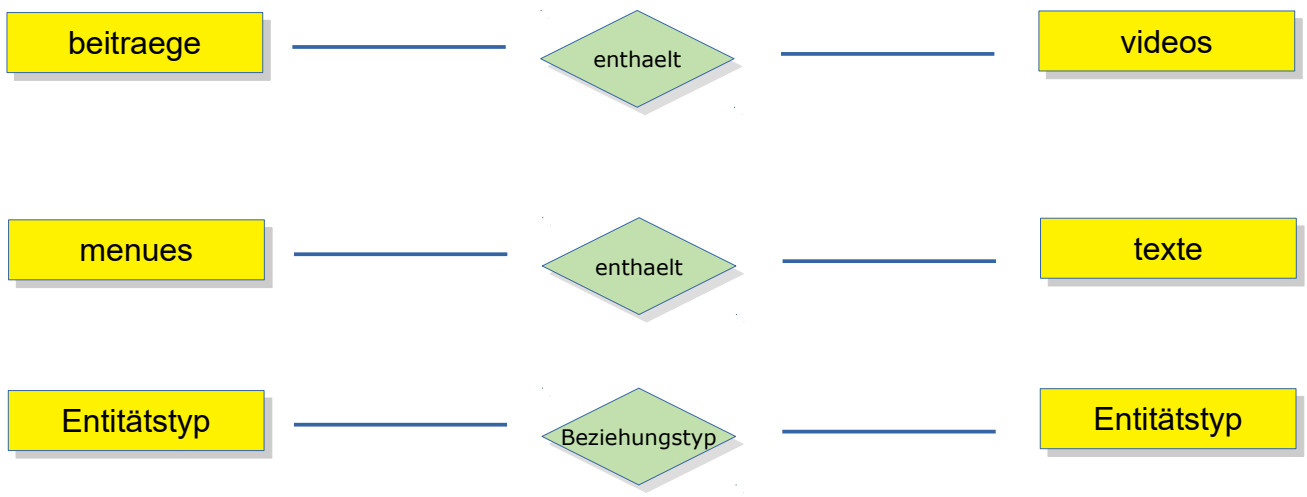
1 Grundlagen....3
1.1 Einführung in die Softwareentwicklung....3
1.2 Einführung in Relationale Datenbanken....5
1.3 Datenbank mit einer Tabelle modellieren....6
1.4 Datenbanktabellen und Beziehungen – Kardinalität N:M....8
1.5 Datenbanktabellen und Beziehungen – Kardinalität N:1 und 1:N....11
1.6 Datenbanktabellen und Beziehungen – Kardinalitäten im Onlineshop....13
1.7 Datenbanktabellen und Beziehungen – Kardinalitäten im Schulbistro....17
1.8 Einführung SQL....19
1.9 SQL: Data Definition Language - DDL....20
1.10 SQL:Data Query Language (DQL)....22
1.11 SQL:Data Query Language: WHERE, ORDER BY, GROUP BY, HAVING....26
1.12 SQL:Data Query Language: Gruppierungsfunktionen (Aggregatsfunktionen). 29
1.13 Data Query Language: Auswahlabfragen über mehrere Tabellen....30
1.14 Checkliste SQL....37

2 Normalisierung....38
2.1 Übung: Normalisierung – Filmchen aus Lernfortschritt 4....39
2.2 Referentielle Integrität....41
2.3 Redundanzfreiheit (Keine Widersprüche in Daten)....42

1 Grundlagen

1.1 Einführung in die Softwareentwicklung

Welche Daten verwaltet das ZDF in der Mediathek?	Datensammlung: Benutzerprofile Beiträge Videos Logs Bilder Stream Texte Links/Verweise Überschriften Menü Partner
In welchem Zusammenhang stehen diese Daten?	**Beitrag** enthält **Video** **Menü** besitzt **Texte** **Texte** enthalten **Links** **Texte** enthalten **Überschriften** **Links** verweisen auf **Beiträge** **Beiträge** enthalten **Texte** **Streams** enthalten **Videos**
Wie erfolgt Heutzutage die Verwaltung vielen Daten?	Relationale Datenbanken Es werden nicht nur die Daten abgespeichert, sondern auch die Beziehung zwischen den Daten. Damit werden die Daten zur Information. Grundoperationen: Löschen Einfügen/Speichern Ändern Verknüpfen
Klären Sie Ihnen unbekannte Begrifflichkeiten!	Datum → ist die Einzahl von Daten



Merke:
Entitätstypen stehen ~~immer~~ im ~~Singular~~ Plural und werden ~~großgeschrieben~~ kleingeschrieben. So sind die Vorgaben (→ Namenskonventionen für das schriftliche Abitur!!!)

(VORGABE lt. Lehrplan prüfen!!!)
→ **Namenskonventionen Datenbankentwicklung**

Hinweis: oben verbessern!!

Z.B.: Kundendaten → Tabelle *kunden*; – Daten der Vereinsmitglieder → Tabelle *mitglieder*
(Für die Schreibweise der Tabellennamen gilt die Konvention:
→ Kleinbuchstaben, Plural, keine Umlaute/Sonderzeichen, keine Leerstellen

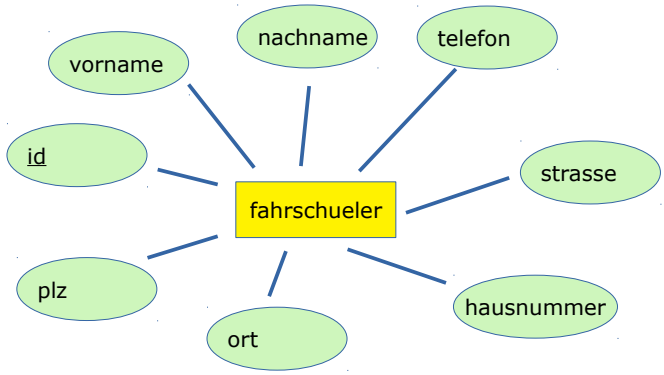
Beziehungstypen sind in der Regel Adjektive.

1.2 Einführung in Relationale Datenbanken

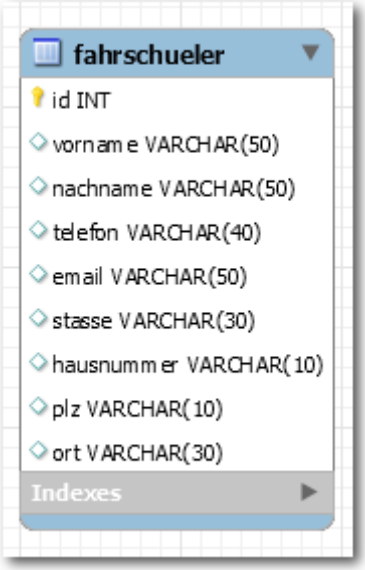
Tabellennamen: fahrschueler									
Attributna-me	id	vorname	nachna-me	telefon	email	strasse	haus-nummer	plz	ort
Datentyp	INT	VARCHAR(n)	VARCHAR(n)	VARCHAR(n)	VARCHAR(n)	VARCHAR(n)	VARCHAR(n)	VARCHAR(n)	VARCHAR(n)
Max. Zei-chenzahl	4 Bytes	50	50	40	50	30	10	10	30

Attribut: Attributname/-wert	Bezeichnung bzw. Wert für Eigenschaft (Feld) eines Systems bzw. der Daten die gespeichert werden sollen → Attributname (Spaltenbezeichnung) → Attributwert (das was in einer konkreten Zelle steht)
atomar	Einwertig → Bedingung für die 1. Normalform Beispiel: Feld „adresse" → nicht einwertig → da Straßenangabe, Hausnummer, PLZ, Ort. → mehrere Felder definieren: strasse, hnr, plz, ort
Datensatz (Tupel)	Zeile in der Datenbanktabelle
Datentyp	Der Datentyp legt die Art des Wertes und die Größe des Wertes fest. Wird für jedes Attribut festgelegt
Primärschlüssel	Ist ein Attribut das den Datensatz in einer Datenbanktabelle eindeutig identifiziert. → einfacher Primärschlüssel → vielfach ein INTEGER dessen Werte beim Einfügen ei-nes Datensatzes automatisch hochgezählt (inkrementiert) wird.
Entitätstyp	Datenbanktabelle bestehend aus Zeilen und Spalten. Die Spaltenbezeichnungen sind die Attributnamen. → Relation

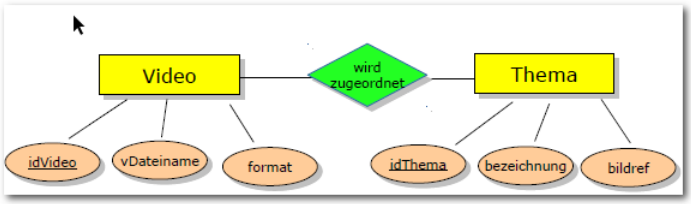
1.3 Datenbank mit einer Tabelle modellieren

1 Welche Aufgaben hat ein Entity-Relationship-Diagramm?	ERD steht für Entity-Relationship-Diagramm ist ein Standard für die Modellierung einer relationalen Datenbank und deren Beziehungen. Damit kann also der logische Aufbau (die Datebankstruktur) konzeptionell entwickelt werden. → in der Praxis wird meist eine Entwicklungsumgebung genutzt um die Modellierung elektronisch umzusetzen. Wir nutzen dazu die abiturrelevante Software MySQL Workbench. → E(lektronisch)ERD
2 Erstellen Sie ein Entity-Relationship-Diagramm für die Daten der Fahrschüler.	
3 Welche Aufgaben erfüllt ein Relationenmodell?	Beschreibt die Relationen (Tabellen), deren Attribute, festgelegten Datentypen und Schlüssel.

4 Erstellen Sie auf der Grundlage des entwickelten Entity-Relationship-Diagramms ein entsprechendes Relationenmodell.	fahrschueler{
	id INT,
	vorname varchar(50),
	nachname varchar(50),
	telefon varchar(40),
	email varchar(50),
	strasse varchar(30),
	hausnummer varchar(10),
	plz varchar(10),
	ort varchar(30)
	}

5 Erstellen Sie ein Datenmodell für die Datenbank fahrschule mit der Tabelle fahr schueler softwaregestützt mit Hilfe der MySQL Workbench.	

1.4 Datenbanktabellen und Beziehungen – Kardinalität N:M

Welche Eigenschaften können wir den Datenbanktabellen (Entitätstypen) Produkt und Hersteller eindeutig zuordnen? → ERD	

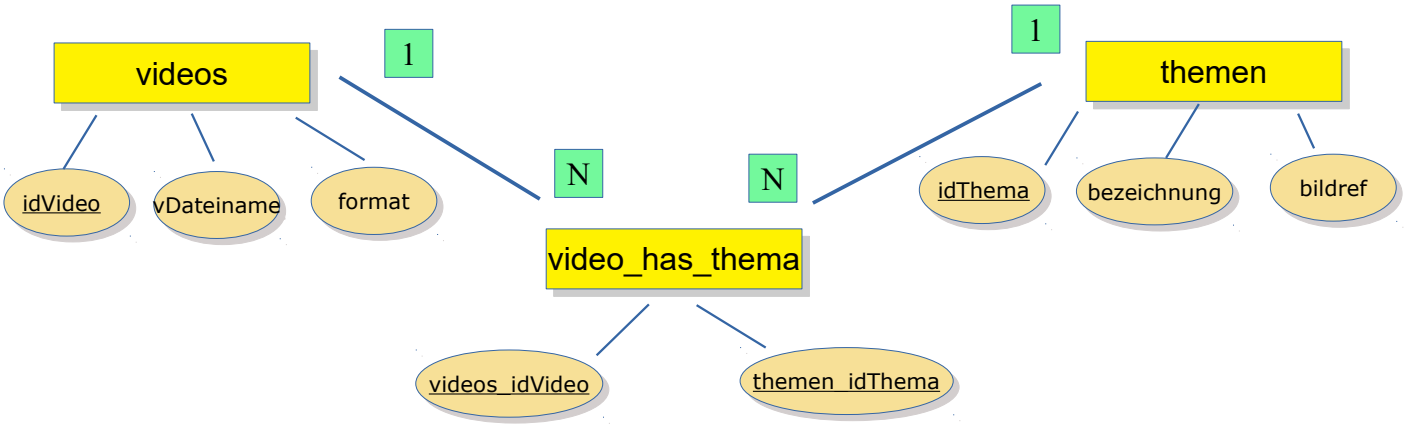
Welche Art von Werten sollen gespeichert werden? → Relationenmodell	videos{
	idVideo INT,
	vDateiname varchar (30),
	format varchar(10)
	}
	
	themen{
	idThema INT,
	bezeichnung varchar (60),
	bildref varchar(256)
	}

Welcher mengenmäßige Zusammenhang besteht zwischen den Datensätzen in den beiden Tabellen? → Kardinalität	Die Kardinalität. Es geht um den Zusammen zwischen Datenbanktabellen und den enthaltenen Datensätzen. Die Textarbeit und der Lösungsweg führen zu einem richtigen Ergebnis. Benotet wird deshalb die Textarbeit und der Lösungsweg. Eine-Sätze: 1. Ein Video wird einem oder **mehren Themen** zugeordnet. [**N**] 2. Ein Thema wird einem oder **mehreren Videos** zugeordnet [**M**]

→ Kardinalität: Thema – Video → N:M

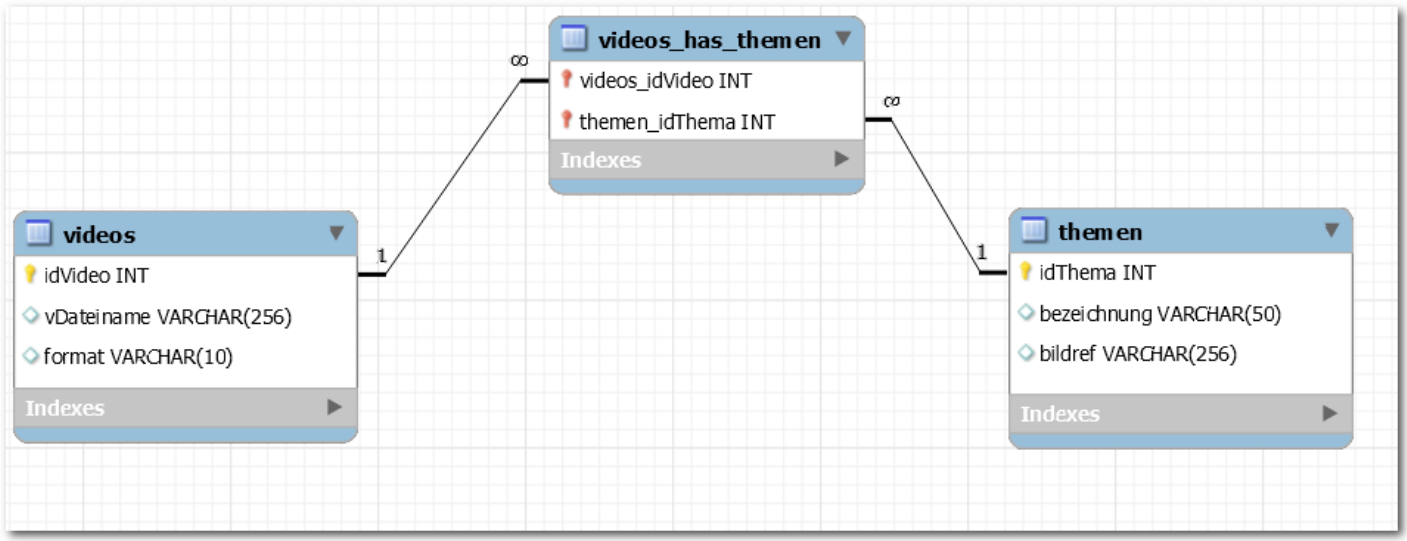
Bei N:M wird eine weitere Datenbanktabelle benötigt!!!

Im ERM ändert sich die Datenbankstruktur wie folgt:



Aus N:M wird 1:N und N:1

Modellieren Sie diesen Zusammenhang mit Hilfe mit einer geeigneten Entwicklungsumgebung.
→ Workbench



Das Relationenmodell ändert sich dann eben auch:

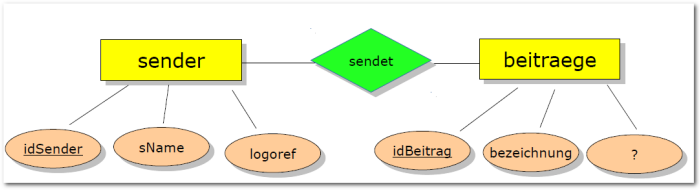
videos{
idVideo INT,
vDateiname varchar (30),
format varchar(10)
}

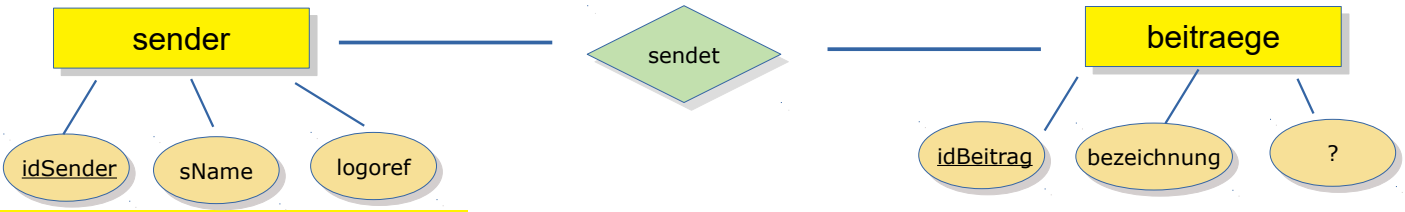
```
themen{
        idThema INT,
        bezeichnung varchar (60),
        bildref varchar(256)
}

video_has_themen{
        videos_idVideo INT,
        themen_idThema INT
}
```

Merke:
In der Zusammenhangstabelle „video_has_themen" besteht der Primärschlüssel aus nun zwei Attributen, deshalb heißt er auch kombinierter Primärschlüssel. Eindeutig ist jetzt nicht mehr der Wert eines Attributs, sondern die Kombination der beiden Attributwerte ist eindeutig.

1.5 Datenbanktabellen und Beziehungen – Kardinalität N:1 und 1:N

Welche Eigenschaften können wir den Datenbanktabellen (Entitätstypen) Produkt und Hersteller eindeutig zuordnen? → ERD





Bestandteil des Lösungsweges!!!

Welche Art von Werten sollen gespeichert werden? → Relationenmodell

```
sender{
        idSender INT,
        sName varchar(50),
        logoref varchar(256)
}

beitraege{
        idBeitrag INT,
        bezeichnung varchar(256),
        zeitpunkt timestamp(256)
}
```

Welcher mengenmäßige Zusammenhang besteht zwischen den Datensätzen in den beiden Tabellen? → Kardinalität

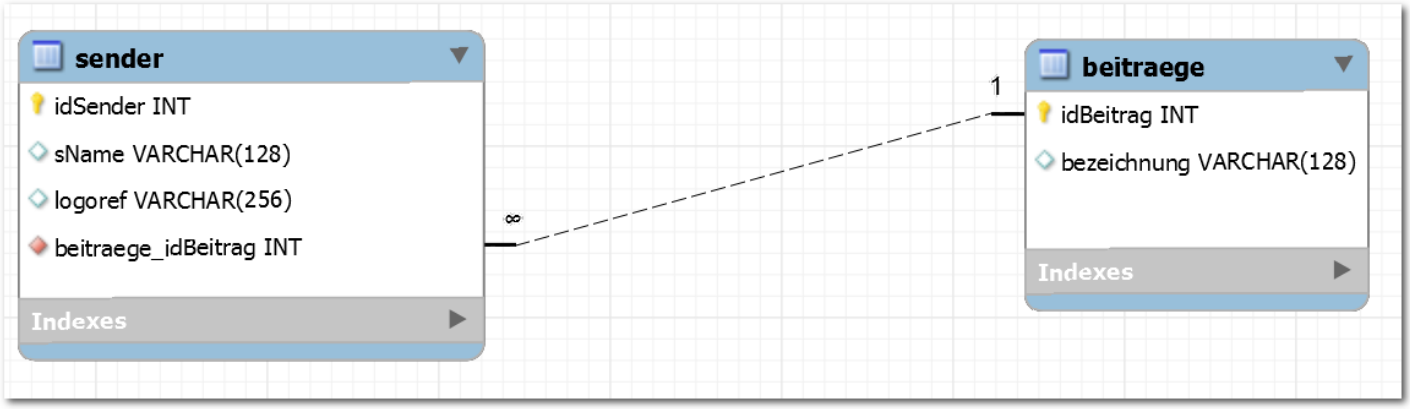
Bestimmung der Kardinalität:

Lösungsweg - Eine-Sätze bilden:

1. Ein Sender sendet (zum Zeitpunkt) **genau einen Beitrag**. [**1**]
2. Ein Beitrag wird auf einem oder **mehreren Sendern gesendet** [**N**]

Kardinalität: sender – beitraege → N : 1

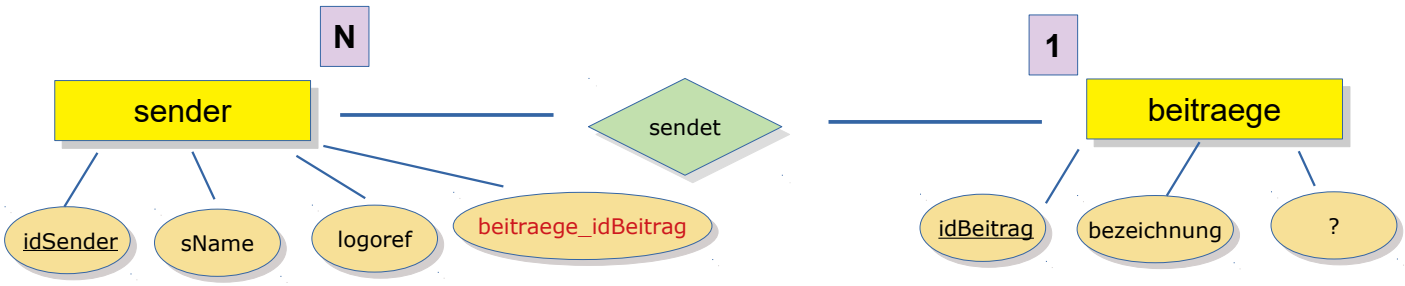
Annahme: Zum Zeitpunkt kann nur ein Beitrag gesendet werden, es sei denn der Beitrag läuft gleichzeitig auf mehreren Sendern.



EERM → Enhanced (Erweitert) Entity Relationship Modell

ERM – Entity Relationship Modell (ERD → Diagram)

Optimiertes / Normalisiertes Model



Relationenmodell

```
sender{
        idSender INT,
        sName varchar(128),
        logoref varchar(256),
        beitraege_idBeitrag
}

beitraege{
        idBeitrag INT,
        bezeichnung varchar(128)
}
```

1.6 Datenbanktabellen und Beziehungen – Kardinalitäten im Onlineshop

Anwendungsfall: Online-Shop für Mobiltelefone

In einem Onlineshop für Mobiltelefone beraten Mitarbeiter jeden Tag viele Kunden über eine Telefonhotlein. Die Kunden werden dazu mit dem nächsten freien Mitarbeiter verbunden.

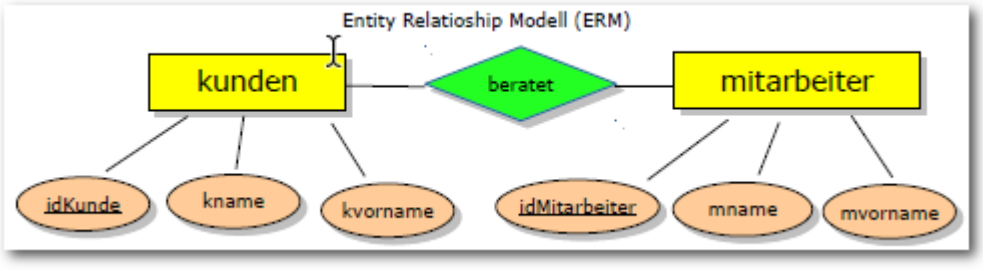
Die Mobiltelefone sind immer mit einem Betriebssytem ausgestattet.

Kunden können die Mobiltelefone bestellen.

Die Mobiltelefone werden direkt beim Hersteller bestellt.

Nutzen Sie die folgenden Angaben um die Aufgabenstellung zu lösen.

Textarbeit	Entitätstypen: • mobiltelefone → produkte • mitarbeiter • kunden • betriebssysteme • hersteller Beziehungstypen: • beraten • ausstatten • bestellen



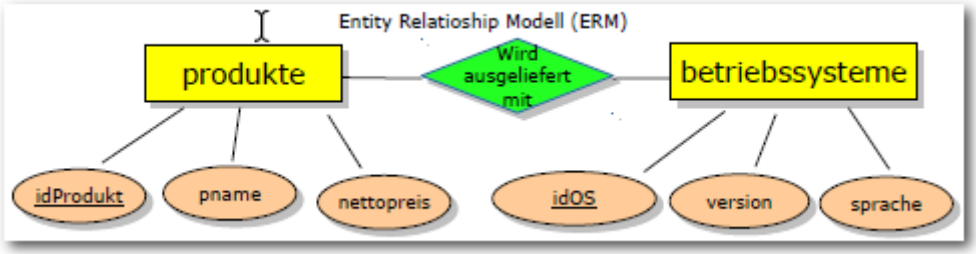
Eine-Sätze:

1. Ein Kunde wird von genau einem Mitarbeiter beraten. [1]
2. Ein Mitarbeiter berät genau einen Kunden. [1]

Annahmen:

Zu einem bestimmten Zeitpunkt kann ein Mitarbeiter nur einen Kunden am Telefonhotline beraten. Ein Kunde kann zu einem bestimmten Zeitpunkt nur ein Telefonat führen, also nur mit einer Person kommunizieren.

Kardinalität: kunden – mitarbeiter → **1:1**

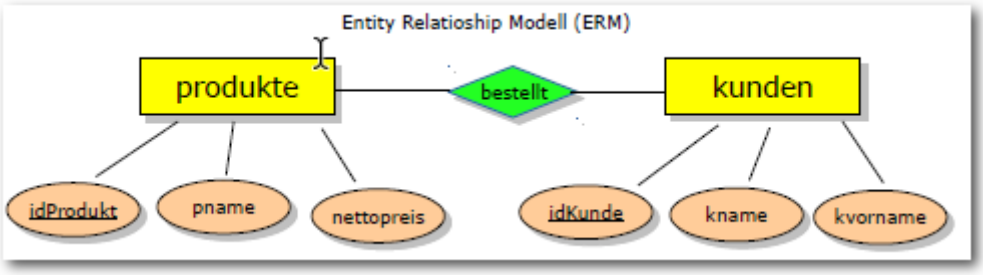


Eine-Sätze:

1. Ein Produkt wird mit genau einem Betriebssystem ausgeliefert.[1]
2. Ein Betriebssystem wird mit mehreren Produkten ausgeliefert. [N]

Annahmen:
Zu einem Zeitpunkt befindet sich nur ein Betriebssystem auf dem Mobiltelefon (nicht bei Zubehör). Zum Zeitpunkt der Auslieferung ist ein Betriebssystem ist auf mehreren Mobiltelefonen (nicht Zubehör) installiert.

Kardinalität: produkte – betriebssysteme → **N:1**

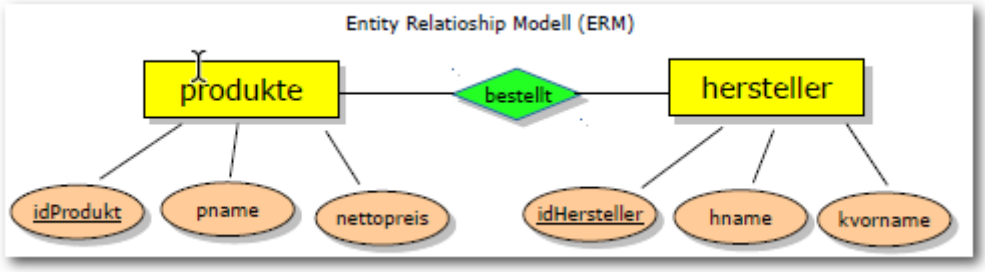


Eine-Sätze:

1. Ein Produkt dürfen mehrere Kunden bestellen [N]
2. Ein Kunde kann ein oder mehrere Produkte bestellen. [M]

Annahmen:
Ein Produkt kann zu einem Zeitpunkt von mehreren Kunden bestellt werden. Zu einem Zeitpunkt kann ein Kunde mehrere Produkte bestellen.

Kardinalität: kunden – produkte → **N:M**



Eine-Sätze:

1. Ein Produkt kann bei **genau einem Hersteller** [1]
2. Bei einem Hersteller können **mehrere Produkte** bestellt werden [N].

Annahmen:
Zu einem Zeitpunkt können bei einem Hersteller mehrere Produkte bestellt werden. Ein Produkt wird von genau einem Hersteller hergestellt.

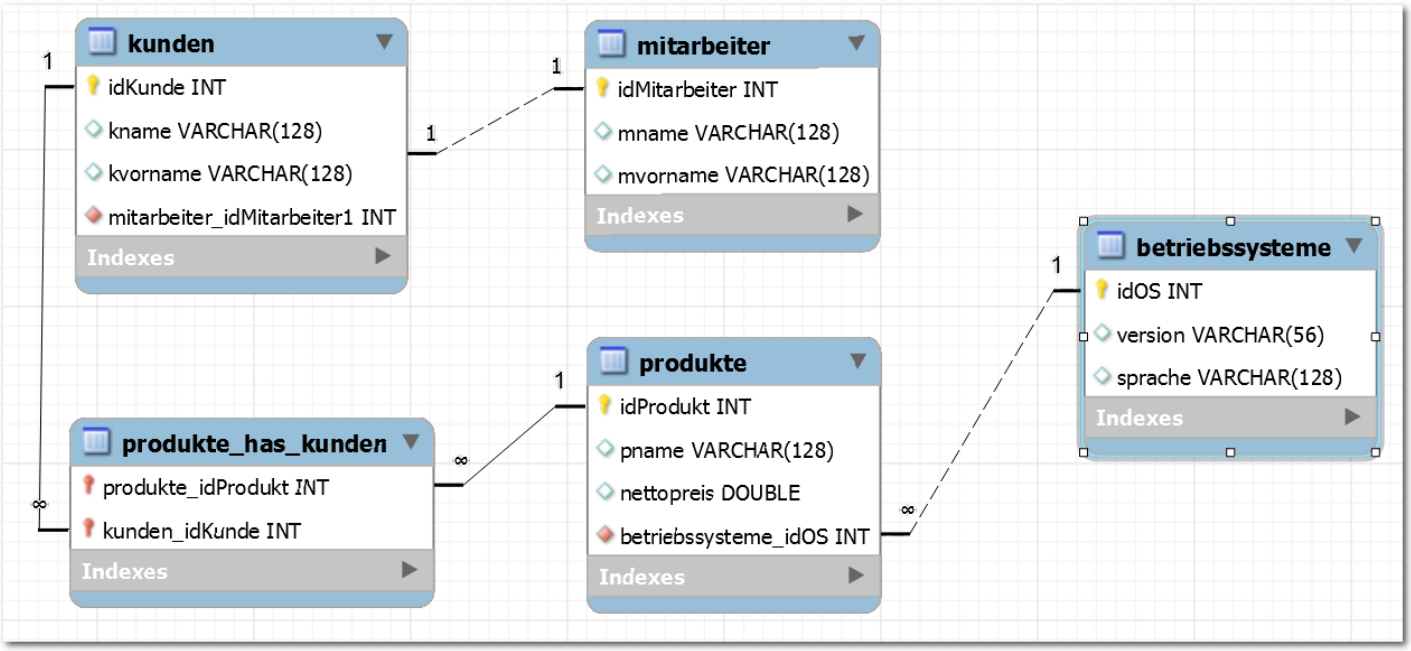
Kardinalität:
produkte – hersteller → **N:1**

Relationenmodell:

```
kunden{
        idKunde INT,
        mitarbeiter_idMitarbeiter INT,
        kname VARCHAR(128),
        kvorname VARCHAR(128)
}

mitarbeiter{
        idMitarbeiter INT,
        mname VARCHAR(128),
        mvorname VARCHAR(128)
}

produkte{
        idProdukt INT,
        betriebssysteme_idOS INT,
        pname VARCHAR(128),
        nettopreis DOUBLE
}

kunden_has_produkte{
        kunden_idKunde INT,
        produkte_idProdukt INT
}
```

betriebssysteme{
idOS INT,
version VARCHAR(56),
sprache VARCHAR(128)
}

EERM



1.7 Datenbanktabellen und Beziehungen – Kardinalitäten im Schulbistro

Kunden erteilen Aufträge

1. Ein Kunden kann ein oder mehrere Aufträge erteilen. [N]
2. Ein Auftrag kann von genau einem Kunden erteilt werden. [1]

Annahmen:
Von Kunden können zu verschiedenen Zeitpunkten mehrere Aufträge erteilt werden. Bezieht sich der Auftrag auf genau einen Kunden mit einer Anschrift (Lieferadresse), kann ein Auftrag zum Zeitpunkt von nur einem Kunden erfolgen.
Kardinalität: kunden – auftraege → 1:N

Aufträge bestehen aus Auftragspositionen

1. Ein Auftrag besteht aus mehreren Auftragspositionen [M]
2. Eine Auftragsposition ist Bestandteil eines oder mehrerer Aufträge [N]

Annahmen:
Zu einem Zeitpunkt kann ein Auftrag aus mehreren AP bestehen.
Eine AP kann zu einem Zeitpunkt Bestandteil eines oder mehrerer Aufträge sein.

Kardinalität: auftraege – auftragspositionen → N:M

Auftragspositionen enthalten Snacks

1. Eine AP enthält genau ein Snack [1]
2. Ein Snack ist in einem oder mehreren Auftragspositionen enthalten [N]

Annahmen:
Die Auftragsposition bezieht sich immer nur auf genau einen Snack x (einen Artikel aus der Datenbank). Ein Snack (Erdnüsse) kann in mehreren APs enthalten (gesalzen, mit Honig)

Kardinalität: auftragspositionen - snacks → N:1

Snacks werden Kategorien zugeordnet

1. Ein Snack wird einer Kategorie zugeordnet [1]
2. Zu einer Kategorie werden eine oder mehrere Snacks zugeordnet.[N]

Annahmen:
Ein Snack soll hier nur in einer Kategorie aufgeführt werden. In ein Kategorie sollen aber alle ähnlichen oder vergleichbaren Snacks aufgeführt sein.

Kardinalitäten: snacks – kategorien → N:1

Relationenmodell

kunden{
kdnr INT,
handynummer VARCHAR()
}

auftraege{
nummer INT,
eingangsdatum DATE,
kunden_kdnr INT
}

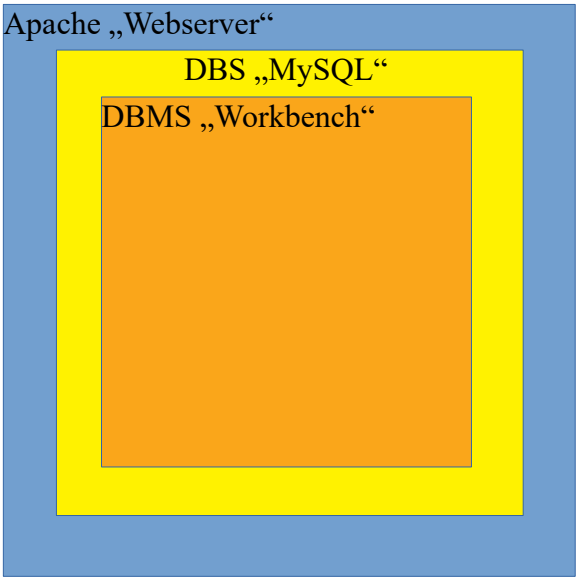
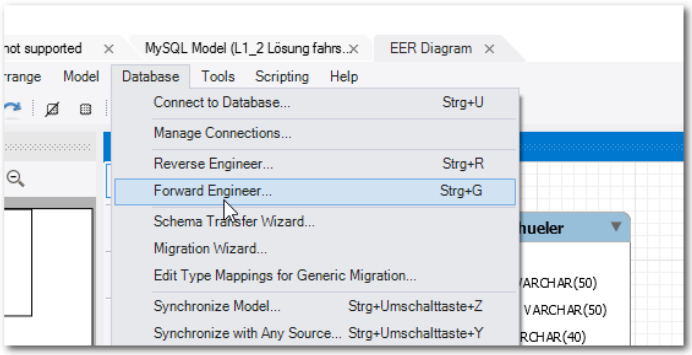
auftraege_has_auftragspositionen{
nummer INT,
ap_nr INT
}

auftragspositionen{
ap_nr INT,
snackmenge VARCHAR(),
nummer VARCHAR(),
snacks_snacknr INT
}

snacks{
snacknr INT,
bezeichnung VARCHAR(),
preis VARCHAR(),
kategorie_kat_nr INT
}

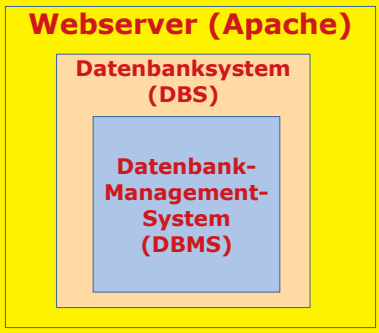
kategorien{
kat_nr INT,
kbezeichnung VARCHAR()
}

1.8 Einführung SQL

<table>
<tr>
<td> 
</td>
<td>Starten den Apache „Webserver“
Starten wir den Dienst MySQL (DBS)
Starten wir das DBMS „Workbench“</td>
</tr>
<tr>
<td> 
</td>
<td>Modell (EERM) exportiert und dann importiert

Modell in der Workbench öffnen → Menüleiste → Database → Forward Engenieer → dem Assistenten folgen → die Datei abspeichern und den Create-Befehl auf dem Datenbanksystem ausführen.</td>
</tr>
</table>

1.9 SQL: Data Definition Language - DDL



Nachtrag!

Vorgehensweise:

1. Digitale Tasche öffnen
2. Webserver Apache starten
3. Datenbanksystem (DBS)→ MySQL starten
4. Datenbankmanagementsystem (DBMS) → Workbench starten

Datenbank erzeugen

```
-- -----------------------------------------------------
-- Schema fahrschule
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `fahrschule` ;
USE `fahrschule` ;
```

Befehl erzeugt die Datenbank (Schema) „fahrschule" und nutzt diese für alle folgenden Befehle.

Datenbanktabelle erzeugen

```
-- -----------------------------------------------------
-- Table `fahrschule`.`fahrschueler`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `fahrschule`.`fahrschueler` (
  `id` INT NOT NULL,
  `vorname` VARCHAR(50) NULL,
  `nachname` VARCHAR(50) NULL,
  `telefon` VARCHAR(40) NULL,
  `email` VARCHAR(50) NULL,
  `stasse` VARCHAR(30) NULL,
  `hausnummer` VARCHAR(10) NULL,
  `plz` VARCHAR(10) NULL,
  `ort` VARCHAR(30) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;
```

Befehl erzeugt die Datenbanktabelle „fahrschueler" und fügt sie in die Datenbank „fahrschule" ein.

Aufgeführt sind alle Attribute mit dem Datentyp und der Definition des Primärschlüssels.

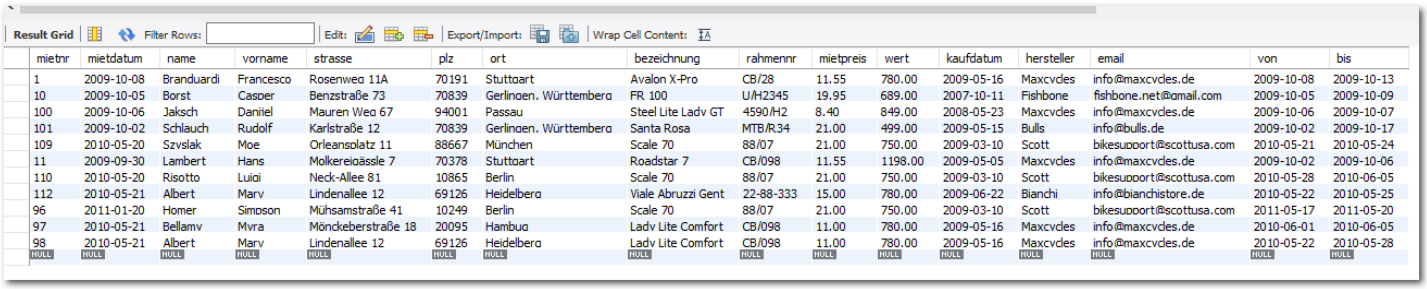
Nachtrag

DDL: Data Definition Language Klauseln	CREATE SCHEMA

	→ erzeugt Datenbank CREATE TABLE → erzeugt Datenbanktabelle ALTER TABLE → verändert die Datenbanktabellenstruktur DROP TABLE → entfernt Datenbanktabelle RENAME → umbenennen einer Datenbanktabelle CREATE VIEW → erzeugt eine veränderte Ansicht auf die Daten

1.10 SQL:Data Query Language (DQL)

Abfragen auf der Datenbank dbfahrrad01 → Tabelle Vermietung



mietnr	mietdatum	name	vorname	strasse	plz	ort	bezeichnung	rahmennr	mietpreis	wert	kaufdatum	hersteller	email	von	bis
1	2009-10-08	Branduardi	Francesco	Rosenweg 11A	70191	Stuttgart	Avalon X-Pro	CB/28	11.55	780.00	2009-05-16	Maxcycles	info@maxcycles.de	2009-10-08	2009-10-13
10	2009-10-05	Borst	Casper	Benzstraße 73	70839	Gerlingen, Württemberg	FR 100	U/H2345	19.95	689.00	2007-10-11	Fishbone	fishbone.net@gmail.com	2009-10-05	2009-10-09
100	2009-10-06	Jaksch	Daniel	Mauren Weg 67	94001	Passau	Steel Lite Lady GT	4590/H2	8.40	849.00	2008-05-23	Maxcycles	info@maxcycles.de	2009-10-06	2009-10-07
101	2009-10-02	Schlauch	Rudolf	Karlstraße 12	70839	Gerlingen, Württemberg	Santa Rosa	MTB/R34	21.00	499.00	2009-05-15	Bulls	info@bulls.de	2009-10-02	2009-10-17
109	2010-05-20	Szvslak	Moe	Orleansplatz 11	88667	München	Scale 70	88/07	21.00	750.00	2009-03-10	Scott	bikesupport@scottusa.com	2010-05-21	2010-05-24
11	2009-09-30	Lambert	Hans	Molkereiässle 7	70378	Stuttgart	Roadstar 7	CB/098	11.55	1198.00	2009-05-05	Maxcycles	info@maxcycles.de	2009-10-02	2009-10-06
110	2010-05-20	Risotto	Luigi	Neck-Allee 81	10865	Berlin	Scale 70	88/07	21.00	750.00	2009-03-10	Scott	bikesupport@scottusa.com	2010-05-28	2010-06-05
112	2010-05-21	Albert	Marv	Lindenallee 12	69126	Heidelberg	Viale Abruzzi Gent	22-88-333	15.00	780.00	2009-06-22	Bianchi	info@bianchistore.de	2010-05-22	2010-05-25
96	2011-01-20	Homer	Simpson	Mühsamstraße 41	10249	Berlin	Scale 70	88/07	21.00	750.00	2009-03-10	Scott	bikesupport@scottusa.com	2011-05-17	2011-05-20
97	2010-05-21	Bellamy	Myra	Mönckebergstraße 18	20095	Hamburg	Lady Lite Comfort	CB/098	11.00	780.00	2009-05-16	Maxcycles	info@maxcycles.de	2010-06-01	2010-06-05
98	2010-05-21	Albert	Marv	Lindenallee 12	69126	Heidelberg	Lady Lite Comfort	CB/098	11.00	780.00	2009-05-16	Maxcycles	info@maxcycles.de	2010-05-22	2010-05-28

1. Der Name, Vorname und Fahrradbezeichnung aller im Jahr 2010 durchgeführten Vermietungen.

Kurzform:

```
SELECT name,vorname,bezeichnung
FROM Vermietung
WHERE YEAR(mietdatum) = 2010;
```

Langform:

```
1  SELECT vermietung.name,vermietung.vorname,vermietung.bezeichnung
2  FROM dbfahrrad01.Vermietung
3  WHERE YEAR(mietdatum) = 2010;
```

name	vorname	bezeichnung
Szvslak	Moe	Scale 70
Risotto	Luigi	Scale 70
Albert	Marv	Viale Abruzzi Gent
Bellamy	Myra	Lady Lite Comfort
Albert	Marv	Lady Lite Comfort

Ergebnis der Abfrage

Der Zeitraum, Fahrradbezeichnung und Hersteller der Vermietungen an Marv Albert.

Kurzform:

```
SELECT von,bis,bezeichnung,hersteller
FROM  Vermietung
WHERE name = 'Albert' AND vorname = 'Marv';
```

Langform:

```
1  SELECT vermietung.von,vermietung.bis,vermietung.bezeichnung,vermietung.hersteller
2  FROM dbfahrrad01.Vermietung
3  WHERE vermietung.name = 'Albert' AND vermietung.vorname = 'Marv';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

von	bis	bezeichnung	hersteller
2010-05-22	2010-05-25	Viale Abruzzi Gent	Bianchi
2010-05-22	2010-05-28	Ladv Lite Comfort	Maxcycles

Ergebnis der Abfrage

Der Name, Vorname, Fahrradbezeichnung und Hersteller aller Vermietungen deren Tagesmietpreis hoher ist als 11 €.

Kurzform:

```
SELECT name,vorname,bezeichnung,hersteller
FROM  Vermietung
WHERE mietpreis > 11.00;
```

Langform:

```
SELECT vermietung.name,vermietung.vorname,vermietung.bezeichnung,vermietung.hersteller
FROM dbfahrrad01.Vermietung
WHERE vermietung.mietpreis > 11.00;
```

Result Grid | Filter Rows: | Export: | Wrap

name	vorname	bezeichnung	hersteller
Branduardi	Francesco	Avalon X-Pro	Maxcycles
Borst	Casper	FR 100	Fishbone
Schlauch	Rudolf	Santa Rosa	Bulls
Szvslak	Moe	Scale 70	Scott
Lambert	Hans	Roadstar 7	Maxcycles
Risotto	Luigi	Scale 70	Scott
Albert	Marv	Viale Abruzzi Gent	Bianchi
Homer	Simpson	Scale 70	Scott

Ergebnis der Abfrage

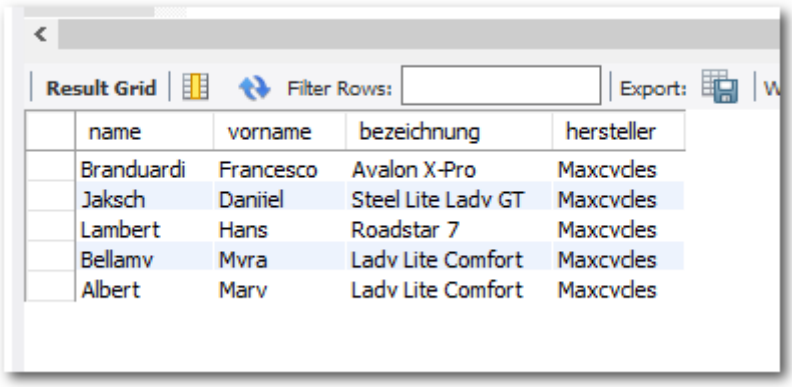
Der Name, Vorname, Fahrradbezeichnung und Hersteller aller Vermietungen der Marke „Maxcycles“

Kurzform:

```
SELECT name,vorname,bezeichnung,hersteller
FROM  Vermietung
WHERE hersteller = 'Maxcycles';
```

Langform:

```
SELECT vermietung.name,vermietung.vorname,vermietung.bezeichnung,vermietung.hersteller
FROM dbfahrrad01.Vermietung
WHERE vermietung.hersteller = 'Maxcycles';
```



Ergebnis der Abfrage

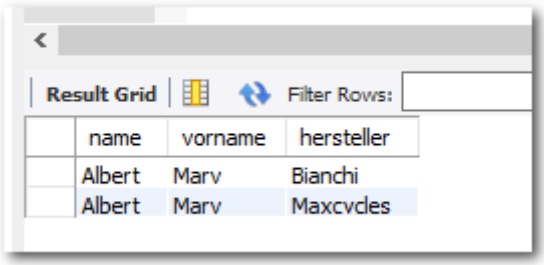
Der Name, Vorname und Hersteller aller Vermietungen des Kunden „Albert“

Kurzform:

```
SELECT name,vorname,hersteller
FROM  Vermietung
WHERE name = 'Albert';
```

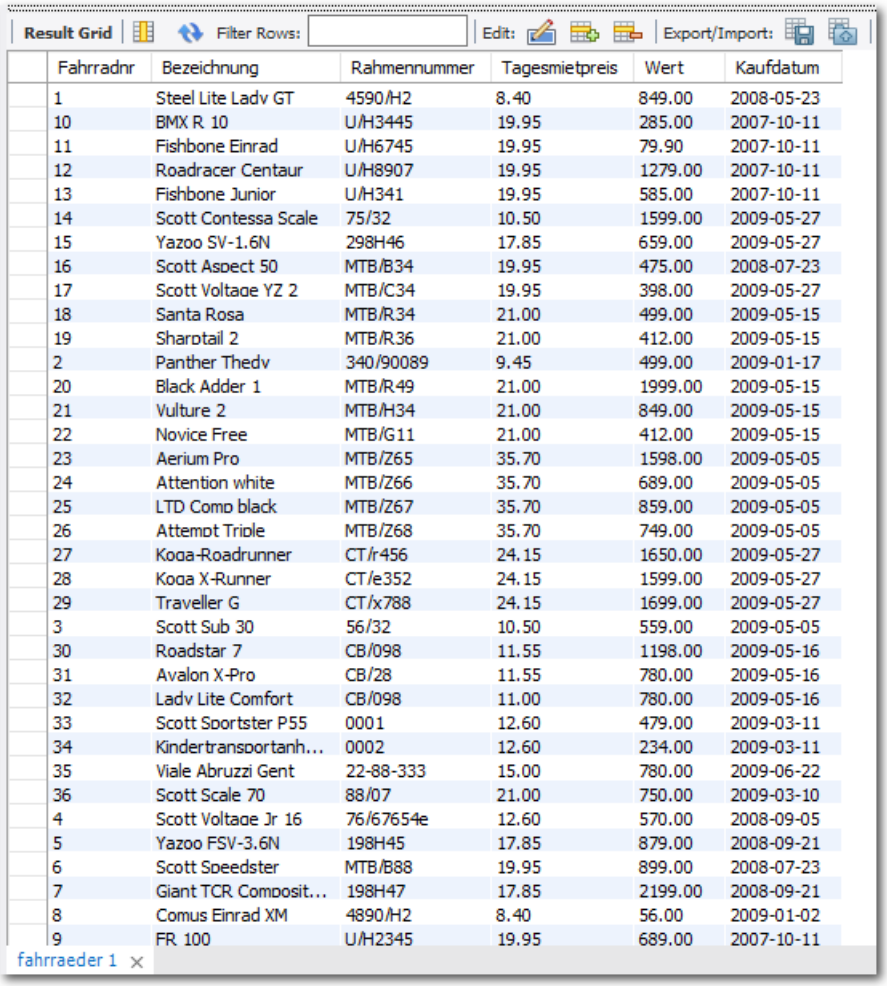
Langform:

```
SELECT vermietung.name,vermietung.vorname,vermietung.hersteller
FROM dbfahrrad01.Vermietung
WHERE vermietung.name = 'Albert';
```



Ergebnis der Abfrage

1.11 SQL:Data Query Language: WHERE, ORDER BY, GROUP BY, HAVING



Result Grid | Filter Rows: | Edit: | Export/Import:

Fahrradnr	Bezeichnung	Rahmennummer	Tagesmietpreis	Wert	Kaufdatum
1	Steel Lite Ladv GT	4590/H2	8.40	849.00	2008-05-23
10	BMX R 10	U/H3445	19.95	285.00	2007-10-11
11	Fishbone Einrad	U/H6745	19.95	79.90	2007-10-11
12	Roadracer Centaur	U/H8907	19.95	1279.00	2007-10-11
13	Fishbone Junior	U/H341	19.95	585.00	2007-10-11
14	Scott Contessa Scale	75/32	10.50	1599.00	2009-05-27
15	Yazoo SV-1.6N	298H46	17.85	659.00	2009-05-27
16	Scott Aspect 50	MTB/B34	19.95	475.00	2008-07-23
17	Scott Voltage YZ 2	MTB/C34	19.95	398.00	2009-05-27
18	Santa Rosa	MTB/R34	21.00	499.00	2009-05-15
19	Sharptail 2	MTB/R36	21.00	412.00	2009-05-15
2	Panther Thedv	340/90089	9.45	499.00	2009-01-17
20	Black Adder 1	MTB/R49	21.00	1999.00	2009-05-15
21	Vulture 2	MTB/H34	21.00	849.00	2009-05-15
22	Novice Free	MTB/G11	21.00	412.00	2009-05-15
23	Aerium Pro	MTB/Z65	35.70	1598.00	2009-05-05
24	Attention white	MTB/Z66	35.70	689.00	2009-05-05
25	LTD Comp black	MTB/Z67	35.70	859.00	2009-05-05
26	Attempt Triple	MTB/Z68	35.70	749.00	2009-05-05
27	Koga-Roadrunner	CT/r456	24.15	1650.00	2009-05-27
28	Koga X-Runner	CT/e352	24.15	1599.00	2009-05-27
29	Traveller G	CT/x788	24.15	1699.00	2009-05-27
3	Scott Sub 30	56/32	10.50	559.00	2009-05-05
30	Roadstar 7	CB/098	11.55	1198.00	2009-05-16
31	Avalon X-Pro	CB/28	11.55	780.00	2009-05-16
32	Ladv Lite Comfort	CB/098	11.00	780.00	2009-05-16
33	Scott Sportster P55	0001	12.60	479.00	2009-03-11
34	Kindertransportanh...	0002	12.60	234.00	2009-03-11
35	Viale Abruzzi Gent	22-88-333	15.00	780.00	2009-06-22
36	Scott Scale 70	88/07	21.00	750.00	2009-03-10
4	Scott Voltage Jr 16	76/67654e	12.60	570.00	2008-09-05
5	Yazoo FSV-3.6N	198H45	17.85	879.00	2008-09-21
6	Scott Speedster	MTB/B88	19.95	899.00	2008-07-23
7	Giant TCR Composit...	198H47	17.85	2199.00	2008-09-21
8	Comus Einrad XM	4890/H2	8.40	56.00	2009-01-02
9	FR 100	U/H2345	19.95	689.00	2007-10-11

fahrraeder 1

Selektion mit einer Bedingung (einfacher Textvergleich) und einer Ordnung

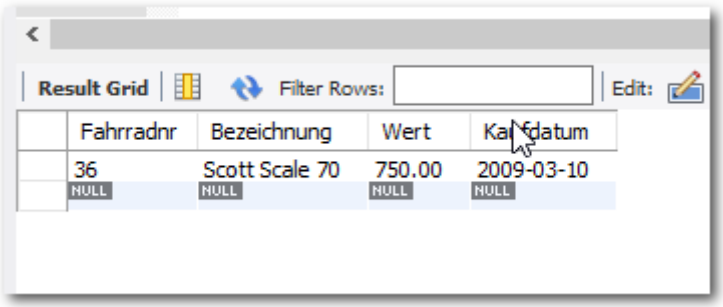
Aufzulisten sind die Attribute Fahrradnr, Bezeichnung, Wert und Kaufdatum des Fahrrads mit der Bezeichnung „Scott Scale 70“.

```
SELECT <Tabellenname>.<Attributname>,...
FROM <Tabellenname>
WHERE <Bedingung>
ORDER BY <Attributname> DESC;

*DESC für Absteigend, ASC für Aufsteigend
```

Kurzform:

Langform:

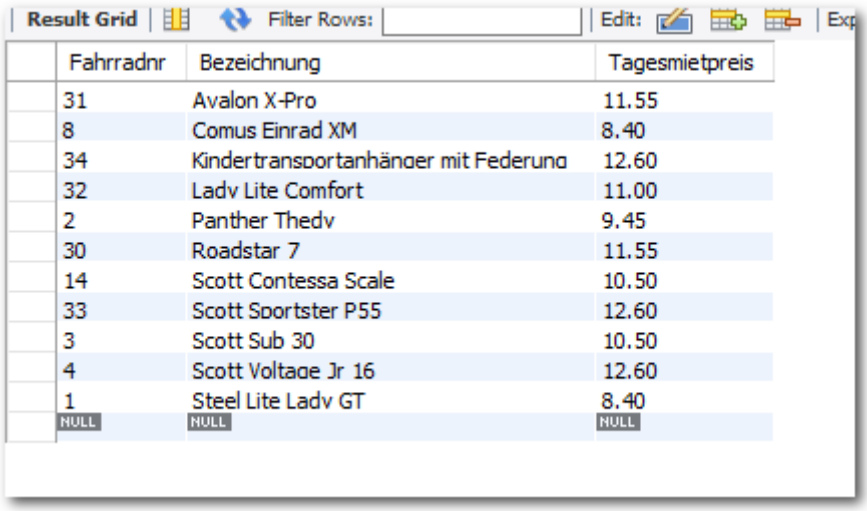
```
1  SELECT  Fahrraeder.Fahrradnr,
2          Fahrraeder.Bezeichnung,
3          Fahrraeder.Wert,
4          Fahrraeder.Kaufdatum
5  FROM    dbfahrrad02.Fahrraeder
6  WHERE   Fahrraeder.Bezeichnung = "Scott Scale 70";
```



Ergebnis der Abfrage

Erweiterung von Aufgabe 1: Selektion mit einer Bedingung (einfacher Textvergleich) und einer Ordnung	

Erweiterung:
Alle Fahrrader, deren Bezeichnung mit ‚Scott' beginnt, soll aufgelistet werden. Die Liste soll nach dem Kaufdatum absteigend sortiert sein.
	Kurzform:
	
	Langform:
	`SELECT Fahrraeder.Fahrradnr, Fahrraeder.Bezeichnung, Fahrraeder.Tagesmietpreis`
	`FROM dbfahrrad02.Fahrraeder`
	`WHERE Fahrraeder.Tagesmietpreis < 15`
	`ORDER BY Fahrraeder.Bezeichnung;`



Ergebnis der Abfrage

Selektion mit einer Bedingung (numerischer Vergleich)	
Gewunscht wird eine Auflistung aller Fahrrader, deren Tagesmietpreis unter 15.00 € liegt (Attribute: Fahrradnr, Bezeichnung, Tagesmietpreis). Die Liste soll aufsteigend sortiert sein.	Kurzform: Langform:

```
SELECT Fahrraeder.Fahrradnr, Fahrraeder.Bezeichnung, Fahrraeder.Tagesmietpreis
FROM dbfahrrad02.Fahrraeder
WHERE Fahrraeder.Tagesmietpreis < 15
ORDER BY Fahrraeder.Bezeichnung;
```

Result Grid | Filter Rows: | Edit:

Fahrradnr	Bezeichnung	Wert	Kaufdatum
14	Scott Contessa Scale	1599.00	2009-05-27
16	Scott Aspect 50	475.00	2008-07-23
17	Scott Voltage YZ 2	398.00	2009-05-27
3	Scott Sub 30	559.00	2009-05-05
33	Scott Sportster P55	479.00	2009-03-11
36	Scott Scale 70	750.00	2009-03-10
4	Scott Voltage Jr 16	570.00	2008-09-05
6	Scott Speedster	899.00	2008-07-23
NULL	NULL	NULL	NULL

Ergebnis der Abfrage

Selektion mit einer Bedingung (Vergleich mit Datumswert)

Alle Fahrrader, die vor dem 01.01.2009 angeschafft wurden, sind nach dem Kaufdatum absteigend sortiert auszugeben. (Fahrradnr, Bezeichnung, Rahmennummer, Kaufdatum).

Kurzform:

Langform:

```
SELECT Fahrraeder.Fahrradnr, Fahrraeder.Bezeichnung, Fahrraeder.Rahmennummer, Fahrraeder.Kaufdatum
From dbfahrrad02.Fahrraeder
WHERE Fahrraeder.Kaufdatum < "2009-01-01"
ORDER BY Fahrraeder.Kaufdatum;
```

Result Grid | Filter Rows: | Edit:

Fahrradnr	Bezeichnung	Rahmennummer	Kaufdatum
9	FR 100	U/H2345	2007-10-11
10	BMX R 10	U/H3445	2007-10-11
11	Fishbone Einrad	U/H6745	2007-10-11
12	Roadracer Centaur	U/H8907	2007-10-11
13	Fishbone Junior	U/H341	2007-10-11
1	Steel Lite Ladv GT	4590/H2	2008-05-23
6	Scott Speedster	MTB/B88	2008-07-23
16	Scott Aspect 50	MTB/B34	2008-07-23
4	Scott Voltage Jr 16	76/67654e	2008-09-05
5	Yazoo FSV-3.6N	198H45	2008-09-21
7	Giant TCR Compos...	198H47	2008-09-21
NULL	NULL	NULL	

Ergebnis der Abfrage

1.12 SQL:Data Query Language: Gruppierungsfunktionen (Aggregatsfunktionen)

pnr	name	vorname	geschl	ortnr	plz	ort	strasse	gebtag	eintritt	stkl	gehalt	kostnr
400000	Schneider	Heinrich	M	29306	73734	Esslingen	Goezstraße 25	1946-10-18	1965-11-01	I	4800.00	02
400001	Schlauch	Franz	M	29675	70599	Stuttgart	Ulmer Weg 56	1952-04-01	1967-05-02	III	4400.00	05
400002	Schlauch	Franz	M	29675	70599	Stuttgart	Ulmer Weg 56	1952-04-01	1967-05-02	III	3675.00	04
400003	Böckle	Jennifer	W	29675	70794	Filderstadt	Hermann-Hesse-Str. 3	1966-05-25	1982-06-01	V	4600.00	03
400004	Hauffe	Johann	M	29306	73734	Esslingen	Seestraße 21	1948-02-01	1991-03-02	III	5400.00	06
400005	Yilmaz	Ali	M	29315	73760	Ostfildern	Wehrstraße 87	1971-05-01	2000-07-01	III	2175.00	06
400006	Berger	Johann	M	30050	71640	Ludwigsburg	Vaihinger Str. 103	1956-10-18	2001-05-01	I	1900.00	05
400007	Schweizer	Anton	M	29303	73730	Esslingen	Schillingstraße 18	1962-01-30	2001-05-01	III	4950.00	05
400008	Lambert	Hans	M	29306	73734	Esslingen	Molkereigässle 7	1948-05-29	2003-04-02	III	3450.00	06
400009	Sautter	Fritz	M	28952	73061	Ebersbach	Karlstraße 23	1984-08-12	2004-10-01	V	1900.00	06
400010	Branduardi	Francesco	M	29740	70794	Filderstadt	Rosenweg 11A	1979-10-31	2005-01-02	I	6150.00	07

Datenbank: muecke_1, Tabelle personal

Selektion mit Aggregatsfunktionen

1. Von allen Mitarbeitern sind die gesamten Monatsgehälter, das durchschnittliche sowie das höchste und das niedrigste Gehalt aufzulisten (siehe Abb.).

Gehaltssumme	durchschnittliches Gehalt	hoechstes Gehalt	geringtes_Gehalt
222537.50	2853.044872	6150.00	1800.00

AVG(Spalte) = Durchschnittswert **COUNT**(Spalte) = Anzahl aller Einträge
MAX(Spalte) = Maximalwert **MIN**(Spalte) = Minimalwert
SUM(Spalte) = Summe aller Einträge in einer Spalte

Gruppierungsfunktionen können nur anstelle eines Spaltennamens direkt hinter der SELECT Anweisung stehen.Sie liefern genau einen Wert, beziehen auf mehrere Tabellenzeilen.

```
SELECT SUM(personal.gehalt) AS "Gehaltssumme",
  AVG(personal.gehalt) AS "durchschnittliches Gehalt" ,
  MAX(personal.gehalt) AS "hoechstes Gehalt",
  MIN(personal.gehalt) AS "geringtes_Gehalt"
FROM muecke_1.personal;
```

Fetched 1 records

Gehaltssumme	durchschnittliches Gehalt	hoechstes Gehalt	geringtes_Gehalt
222537.50	2853.044872	6150.00	1800.00

Selektion mit Aggregatsfunktionen

2. Mit Hilfe einer Abfrage ist die Anzahl der Mitarbeiter, die in der Abteilung mit der Kostenstellennummer 04 beschäftigt sind, zu ermitteln.

kostnr	Anzahl_Mitarbeiter
04	12

AVG(Spalte) = Durchschnittswert **COUNT**(Spalte) = Anzahl aller Einträge
MAX(Spalte) = Maximalwert **MIN**(Spalte) = Minimalwert
SUM(Spalte) = Summe aller Einträge in einer Spalte

Gruppierungsfunktionen können nur anstelle eines Spaltennamens direkt hinter der SELECT Anweisung stehen.Sie liefern genau einen Wert, beziehen auf mehrere Tabellenzeilen.

```
SELECT personal.kostnr, COUNT(*) AS Anzahl_Mitarbeiter
FROM muecke_1.personal
WHERE personal.kostnr = "04";
```

kostnr	Anzahl_Mitarbeiter
04	12

1.13 Data Query Language: Auswahlabfragen über mehrere Tabellen

(Quelle: Zusatzskript - Auswahlabfragen mit mehreren Tabellen ab der Seite 23 (muecke_2)

kundennr	firma	strasse	ortnr	ortnr	ort
270033	Einrichtungshaus Blessing	Bissinger Str. 5	34976	34976	Augsburg, Bay
270036	Meinschneider Einrichtungen	Brunnenweg 88	29303	29303	Esslingen am Neckar
270017	NimmMit Mitnahmemöbel GmbH	Brühlstr. 50	29306	29306	Esslingen am Neckar
270053	Einrichtungshaus Bluthardt	Kastellstr. 14	29303	29303	Esslingen am Neckar
270040	Schober Vollholzmöbel KG	Heilbronner Str. 3	29675	29675	Stuttgart
270002	FIRENCE Möbel & Leuchten GmbH	Königsstraße 12	29521	29521	Stuttgart
270018	Einrichtungshaus Bull e.K.	Hauptstraße	29654	29654	Stuttgart
270008	Bär-Möbel GmbH	Luisenstr. 55	36023	36023	Ulm, Donau

```
SELECT kundennr, firma, strasse, kunden, plz, ort ← (1)
FROM kunden, orte ← (2)
WHERE kunden.ortnr = orte.ortnr ← (3)
AND (ort LIKE "Ulm%" OR ort LIKE "Stutt%") ← (4)
ORDER BY ort; ← (5)
```

(1) Zusammenstellen der anzuzeigenden Attribute (Projektion).
(2) Auswahl der benötigten Tabellen.
(3) Bedingung für die Verbindung der beiden Tabellen. Für das gemeinsame Attribut ortnr müssen in beiden Tabellen übereinstimmende Attributswerte vorhanden sein. Attribute, die in beiden Tabellen vorkommen, müssen mit dem Tabellennamen als „Vorsilbe" angesprochen werden, Beispiel *kunden.ortnr*.
(4) Bedingung für die Auswahl der Datensätze (= Selektion). (Die in der WHERE-Klausel verwendeten Platzhalter sollen sicher stellen, dass auch eine Ortsbezeichnung wie beispielsweise „Ulm, Donau" selektiert wird.)
(5) Sortierung.

Datenbank: muecke_2 Tabellen: Kunden und Orte

Rangfolge	**Parent-Tabelle**	**Primärschlüssel**	**Child-Tabelle**	**Fremdschlüssel**
1.	ORTE	OrtNr		
2.	KONDITIONEN	KondNr		
3.	BANKEN	Blz		
4.	ZULAGEN	PNR		
5.	ZUSCHLAEGE	ZNr		
6.	PERSONAL	PNr	PERSONAL	Blz,OrtNr
7.	KUNDEN	KdNr	KUNDEN	KondNr,OrtNr,PNr
8.	AUFTRAEGE	AufNr	AUFTRAEGE	KdNr
9.	FE	FENr	FE	ZNr
10.	AUF_POS	Aufposnr	AUF_POS	AufNr,FENr
11.	GEHAELTER	PNr	GEHAELTER	PNr
12.	LOEHNE	PNr	LOEHNE	PNr

Aufgabe

Situation:

Für eine Werbeaktion wird eine Adressliste aller Kunden aus Stuttgart und Ulm benötigt. Die Liste die Attribute *name, plz, strasse* und *ort* enthalten und nach den Kundennamen alphabetisch sortiert sein.

Abfrage

```
1  SELECT kunden.KdNr, kunden.Firma, kunden.Strasse, orte.PLZ, orte.OrtName
2  FROM muecke_2.kunden, muecke_2.orte
3  WHERE kunden.ortNr = orte.ortNr
4  AND (orte.OrtName LIKE "Ulm%" OR orte.OrtName LIKE "Stutt%")
5  ORDER BY orte.OrtName;
```

Angezeigte Lösung

Overview | Output | Snippets | Result (1) | Fetched 4

KdNr	Firma	Strasse	PLZ	OrtName
270040	Schober Vollholzmöbel KG	Heilbronner Str. 3	70599	Stuttgart
270018	Einrichtungshaus Bull e.K.	Hauptstraße	70555	Stuttgart
270002	FIRENCE Leuchten GmbH	Calwer Str.	70173	Stuttgart
270008	Bär-Möbel GmbH	Luisenstr. 55	89073	Ulm, Donau

Aufgabe

Abfrage 1 Alle Kunden mit einem Rabattsatz von 10% sollen mit den Feldern *firma, plz, ort* und *rabatt* aufgelistet werden.

#Alle Kunden mit einem Rabattsatz von 10% sollen mit den Feldern firma, plz, ort und rabatt aufgelistet werden.

Abfrage

```
1  SELECT   kunden.KdNr,
2           kunden.Firma,
3           orte.PLZ,
4           orte.OrtName,
5           konditionen.Rabatt
6  FROM     muecke_2.kunden,
7           muecke_2.orte,
8           muecke_2.konditionen
9  WHERE    kunden.KondNr = konditionen.KondNr
10 AND      kunden.OrtNr = orte.OrtNr
11 AND      konditionen.Rabatt = 10;
```

Angezeigte Lösung

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

KdNr	Firma	PLZ	OrtName	Rabatt
270046	Möbelzentrale OTTO	74385	Pleidelsheim	10
270047	MARKTEX Exclusiveinrichtungen	89518	Heidenheim an der Brenz	10
270049	Möbel-Treffpunkt-Exclusiv	79576	Weil am Rhein	10
270051	ARTEFAKT GmbH	71063	Sindelfingen	10
270052	Möbel Scheufele	77815	Bühl, Baden	10
270053	Einrichtungshaus Bluthardt	73730	Esslingen am Neckar	10
270055	Möbel Schmidt GmbH & Co KG	97070	Würzburg	10

Aufgabe

Die Auftragskopfdaten des Auftrages mit der Auftragsnummer 20370 sind wie in der nachfolgenden Anzeige dargestellt, anzuzeigen.

auftragsnr	auftragsdat	kundennr	firma	strasse	plz	ort
20370	2014-08-20	270015	Schlenz Möbel KG	Bachrain 71	89250	Senden, Iller

Abfrage

```
1  SELECT  Auftraege.AufNr AS 'auftragsnr',
2          Auftraege.AufDat AS 'auftragsdat',
3          Kunden.kdNr AS 'kundennr',
4          Kunden.Firma AS 'firma',
5          Kunden.Strasse AS 'strasse',
6          Orte.PLZ AS 'plz',
7          Orte.OrtName AS 'orte'
8  FROM muecke_2.Auftraege, muecke_2.Kunden, muecke_2.Orte
9  WHERE Kunden.OrtNr = Orte.OrtNr
10 AND Auftraege.KdNr = Kunden.KdNr
11 AND auftraege.AufNr = 20370;
```

Angezeigte Lösung

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

auftragsnr	auftragsdat	kundennr	firma	strasse	plz	orte
020370	2009-09-09	270015	Schlenz Möbel KG	Bachrain 71	89250	Senden. Iller

Aufgabe

3. Die Vertriebsleiterin Frau Angelika Schlauch-Köpf wünscht von der Produktgruppe I der Fertigerzeugnisse eine Auflistung der Attribute erzeugnisnr, bezeichnung, herstellkosten sowie Zuschlagssatznummer (Attribut zuschlagsnr) und Kalkulationszuschlagssatz (Attribut zuschlagssatz)

fertigernr	bezeichnung	herstellkosten	zuschlagsnummer	zuschlagssatz
200030	Stuhl EIFEL	275.50	Z1	16
200032	Stuhl TAUNUS	197.50	Z1	16
200012	Sideboard INN	1125.50	Z2	20
200017	Vertiko BLEIBTREU	995.75	Z2	20
200018	Vertiko SEHNSUCHT	1250.75	Z2	20
200024	Esstisch INNSBRUCK	915.50	Z2	20
200001	Kleiderschrank PARIS	1675.50	Z3	23
200003	Schrankwand ALLGÄU	955.50	Z3	23
200007	Schrankwand TIROL	883.75	Z3	23
200010	Sideboard DONAU	695.50	Z3	23
200013	Sideboard ISAR	755.00	Z3	23
200026	Couchtisch LECH	712.50	Z3	23
200004	Schrankwand LINZ	995.00	Z4	25
200006	Schrankwand OSTSEE	825.75	Z4	25
200002	Kleiderschrank LONDON	1175.75	Z5	30

Abfrage

Query 1 | fe

Limit to 1000 rows

```
1
2  SELECT  fe.FENr AS 'fertigernr',
3      fe.FEBez AS 'bezeichnung',
4      fe.HK AS 'herstellkosten',
5      fe.Znr AS 'zuschlagsnummer',
6      zuschlaege.Z_Satz AS 'zuschlagssatz'
7  FROM muecke_2.fe, muecke_2.zuschlaege
8  WHERE fe.ZNr = zuschlaege.ZNr
9  AND fe.PGruppe = "I";
```

Angezeigte Lösung

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

fertigernr	bezeichnung	herstellkosten	zuschlagsnummer	zuschlagssatz
200030	Stuhl EIFEL	275.50	Z1	16
200032	Stuhl TAUNUS	197.50	Z1	16
200012	Sideboard INN	1125.50	Z2	20
200017	Vertiko BLEIBTREU	995.75	Z2	20
200018	Vertiko SEHNSUCHT	1250.75	Z2	20
200024	Esstisch INNSBRUCK	915.50	Z2	20
200001	Kleiderschrank PARIS	1675.50	Z3	23
200003	Schrankwand ALLGÄU	955.50	Z3	23
200007	Schrankwand TIROL	883.75	Z3	23
200010	Sideboard DONAU	695.50	Z3	23
200013	Sideboard ISAR	755.00	Z3	23
200026	Couchtisch LECH	712.50	Z3	23
200004	Schrankwand LINZ	995.00	Z4	25
200006	Schrankwand OSTSEE	825.75	Z4	25
200002	Kleiderschrank LONDON	1175.75	Z5	30

Aufgabe

4. Die Verkaufsleitung wünscht eine Aufstellung der Herstellkosten, Zuschlagssätze und der Netto-Verkaufspreise aller Fertigerzeugnisse. (Netto-Verkaufspreis = Herstellkosten * Zuschlagssatz/100). Die Aufstellung soll nach den Verkaufspreisen aufsteigend sortiert sein und folgendes Aussehen haben (auszugsweise):

fertigerznr	bezeichnung	herstellkosten	zuschlagssatz	Verkaufspreis
200028	Tisch ALPHA	98.50	16	114.260000
200023	Wandregal BETA	175.00	16	203.000000
200034	Rollcontainer UNIT	175.50	20	210.600000

Abfrage

```
1  SELECT  fe.FENr AS 'fertigerznr',
2          fe.FEBez AS 'bezeichnung',
3          fe.HK AS 'herstellkosten',
4          zuschlaege.Z_Satz AS 'zuschlagssatz',
5          fe.HK+(fe.HK*zuschlaege.Z_Satz/100) AS 'Verkaufspreis'
6  FROM muecke_2.zuschlaege, muecke_2.fe
7  WHERE fe.ZNr = zuschlaege.ZNr
8  ORDER BY Verkaufspreis;
```

Merke:
Werte die berechnet werden können werden selbst nicht in der Datenbank gespeichert. Gespeichert werden die Werte die ich zur Berechnung benötige! → Bestandteile einer Formel. Die Berechnung selbst erfolgt im SELECT!

Angezeigte Lösung

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

fertigerznr	bezeichnung	herstellkosten	zuschlagssatz	Verkaufspreis
200028	Tisch ALPHA	98.50	16	114.260000
200023	Wandregal BETA	175.00	16	203.000000
200034	Rollcontainer UNIT	175.50	20	210.600000
200033	Stuhl TAURUS	185.50	23	228.165000
200032	Stuhl TAUNUS	197.50	16	229.100000
200009	Schrank GAMMA	225.50	16	261.580000
200030	Stuhl EIFEL	275.50	16	319.580000
200035	Hochcontainer WENDY	298.00	20	357.600000
200029	Stuhl COMBRA	275.25	30	357.825000
200031	Stuhl LEGGERA	322.50	30	419.250000
200021	Wandregal UNO	385.00	20	462.000000
200019	Wandregal DUO	495.00	23	608.850000
200020	Wandregal INTENTO	596.00	30	774.800000
200010	Sideboard DONAU	695.50	23	855.465000
200026	Couchtisch LECH	712.50	23	876.375000

357.60

Aufgabe

5. Welche Fertigerzeugnisse hat der Kunde City-Möbel GmbH (kundennr = 270042) im Juni 2009 in Auftrag gegeben? Die Aufstellung soll folgendes Aussehen haben (auszugsweise):

kundennummer	firma	plz	ort	auftragsdat	fertigungsnr	menge	bezeichnung
270042	City-Möbel GmbH	73312	Geislingen an der Steige	2009-06-28	200014	4	Sideboard MENOS

Abfrage

Query 1 | kunden | orte | auftraege | fe

Limit to 100

```
1  SELECT  kunden.KdNr AS 'kundennr',
2          kunden.Firma AS 'firma',
3          orte.PLZ AS 'plz',
4          orte.OrtName AS 'ort',
5          auftraege.AufDat AS 'auftragsdat
6          fe.FENr AS 'fertigerznr',
7          fe.FEBez AS 'bezeichnung',
8          auf_pos.Menge AS 'menge'
9  FROM    muecke_2.kunden,
10         muecke_2.orte,
11         muecke_2.auftraege,
12         muecke_2.fe,
13         muecke_2.auf_pos
14 WHERE   kunden.OrtNr = orte.OrtNr
15 AND     auftraege.KdNr = kunden.KdNr
16 AND     auf_pos.AufNr = auftraege.AufNr
17 AND     auf_pos.FENr = fe.FENr
18 AND     kunden.KdNr = 270042
19 AND     MONTH(auftraege.AufDat) = 6
20 AND     YEAR(auftraege.AufDat) = 2009;
```

Angezeigte Lösung

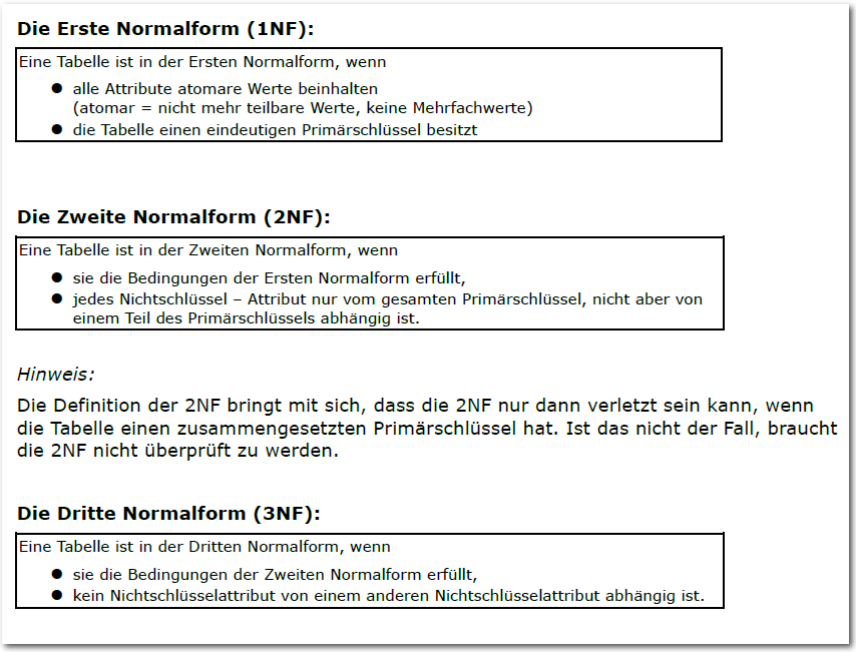
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

kundennr	firma	plz	ort	auftragsdat	fertigerznr	bezeichnung	menge
270042	City-Möbel GmbH	73312	Geislingen an der Steige	2009-06-28	200014	Sideboard MENOS	4

1.14 Checkliste SQL

1. Anführungszeichen für Werte verwenden deren Datentyp VARCHAR oder DATE ist.
2. Klammersetzung überprüfen:
   - Funktion YEAR(...),MONTH(...), DAY(...) sind parameterbehaftet, d.h. Der Attributname steht innerhalb der Klammern. → YEAR(kaufdatum)
   - Funktion NOW() ist parameterlos, d.h. In der Klammer steht nichts!
   - Funktionen können geschachtelt werden → YEAR(NOW()) → liefert das aktuelle Jahr.
3. Verwendete Attributnamen prüfen! (Groß- und Kleinschreibung, Zeichendreher,...)
4. Reihenfolge der Klauseln prüfen.
5. Vergleichsoperatoren überprüfen
6. Semikolon (;) prüfen. Es darf nur einer je Abfrage enthalten sein.
7. In der ausführlichen Form Datenbankname und die Tabellenbezeichnung an den geforderten Stellen nicht vergessen.
8. In Abfragen über mehrere Tabellen den Vergleich von Primär- und Fremdschlüsselattributen nicht vergessen, um das Kreuzprodukt auf das wesentliche zu reduzieren. (→ Projektion)

2 Normalisierung



Die Erste Normalform (1NF):

Eine Tabelle ist in der Ersten Normalform, wenn

- alle Attribute atomare Werte beinhalten
  (atomar = nicht mehr teilbare Werte, keine Mehrfachwerte)
- die Tabelle einen eindeutigen Primärschlüssel besitzt

Die Zweite Normalform (2NF):

Eine Tabelle ist in der Zweiten Normalform, wenn

- sie die Bedingungen der Ersten Normalform erfüllt,
- jedes Nichtschlüssel – Attribut nur vom gesamten Primärschlüssel, nicht aber von einem Teil des Primärschlüssels abhängig ist.

Hinweis:

Die Definition der 2NF bringt mit sich, dass die 2NF nur dann verletzt sein kann, wenn die Tabelle einen zusammengesetzten Primärschlüssel hat. Ist das nicht der Fall, braucht die 2NF nicht überprüft zu werden.

Die Dritte Normalform (3NF):

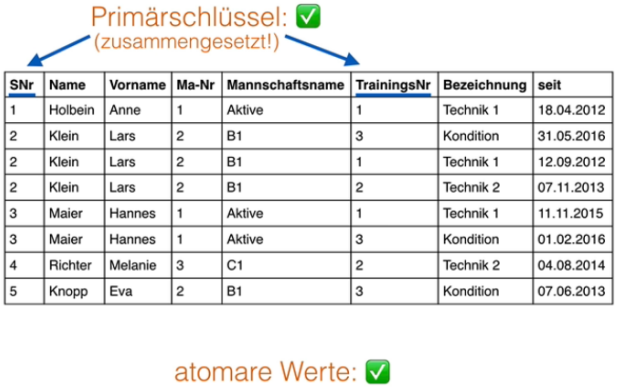
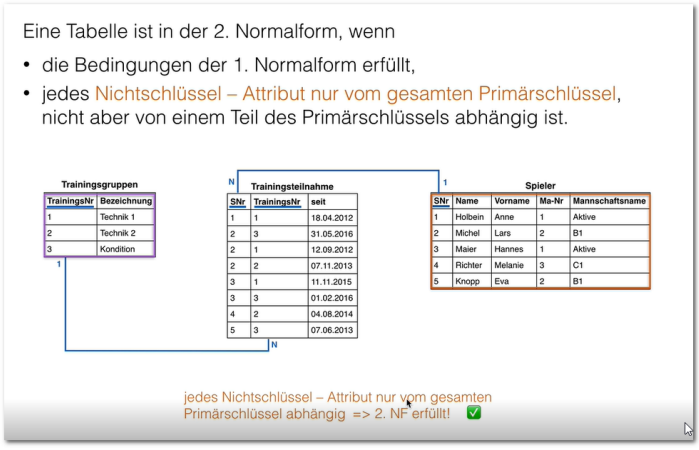
Eine Tabelle ist in der Dritten Normalform, wenn

- sie die Bedingungen der Zweiten Normalform erfüllt,
- kein Nichtschlüsselattribut von einem anderen Nichtschlüsselattribut abhängig ist.

Normalisierung	Die Normalisierung beschreibt eine Vorgehensweise die schrittweise sicherstellt, dass die Struktur der künftigen Datenbank optimal gestaltet ist. Es gibt sechs Normalformen. Nur die ersten drei Normalformen sind für uns relevant.
Anomalien	Wenn Datenbanken nicht normalisiert sind kann dies zu fehlerhaften Abfrageergebnissen oder inkonsistenten Daten führen. Anomalien sind also allgemein betrachtet Fehler. **Folgende Fehler** sind möglich: 1. Abfrage(SELECT)-Anomalie: Daten einer Abfrage sind z.B. unvollständig da bei der Eingabe schon Fehler passiert sind. 2. Aktualisierungs (UPDATE)-Anomalie: Daten wurden nicht aktualisiert da z.B. eine Eingabefehler dazu geführt hat, dass das Datum bei der Aktualisierung unberücksichtigt blieb. 3. Einfüge (INSERT)-Anomalie Beim Einfügen von Daten verursachen z.B. Zeichendreher oder „Meh-

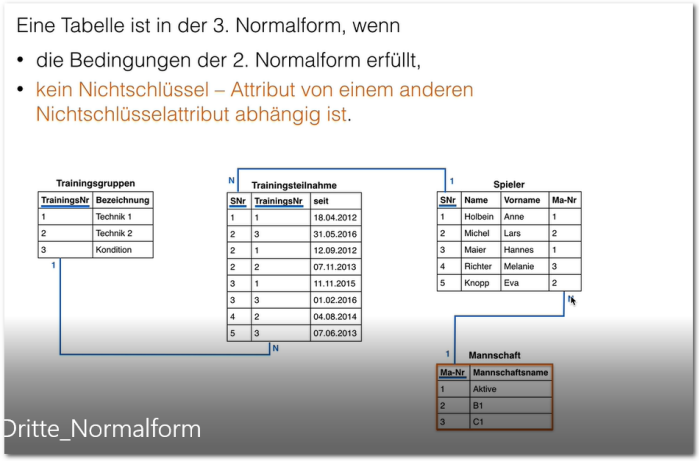
	rere Bezeichnungen für ein und das selbe" verursachen Fehler und damit Inkonsistenzen. (Für eine Abteilung „Personal" und „Human Resources")

2.1 Übung: Normalisierung – Filmchen aus Lernfortschritt 4

L4_2.1 1. Normalform	 SNr | Name | Vorname | Ma-Nr | Mannschaftsname | TrainingsNr | Bezeichnung | seit 1 | Holbein | Anne | 1 | Aktive | 1 | Technik 1 | 18.04.2012 2 | Klein | Lars | 2 | B1 | 3 | Kondition | 31.05.2016 2 | Klein | Lars | 2 | B1 | 1 | Technik 1 | 12.09.2012 2 | Klein | Lars | 2 | B1 | 2 | Technik 2 | 07.11.2013 3 | Maier | Hannes | 1 | Aktive | 1 | Technik 1 | 11.11.2015 3 | Maier | Hannes | 1 | Aktive | 3 | Kondition | 01.02.2016 4 | Richter | Melanie | 3 | C1 | 2 | Technik 2 | 04.08.2014 5 | Knopp | Eva | 2 | B1 | 3 | Kondition | 07.06.2013 atomare Werte: ✅ → Atomare Werte (keine Mehrfachnennungen in einer Zelle, keine teilbaren Werte in einer Zelle) → Prüfung ob der identifizierbare Primärschlüssel nach der Umstrukturierung immer noch eindeutig ist, sonst eventuell kombinieren mit weiteren Attributen.
L4_2.2 2. Normalform	 Stichwort: → Nach Sachverhalten trennen → Zusammenhang mittels Primär- und Sekundär-(Fremd-)schlüsselattribute rekonstruieren.

L4_2.3

3.Normalform



Stichwort:
→ Nach Sachverhalten trennen
→ Zusammenhang mittels Primär- und Sekundär-(Fremd-)schlüsselattribute rekonstruieren.

2.2 Referentielle Integrität

muecke_0

Einfügen von Daten in die Tabelle:
→ Erstmaliges Einfügen eines noch nicht existierenden Datensatzes

INSERT INTO

→ Erst Daten in die Tabellen ohne Fremdschlüsselattribut einfügen.

```
--- muecke_0 -- INSERT

SELECT *
FROM muecke_0.kunden
WHERE kunden.firma = "Meinhardt - Möbel nach Maß" ;

SELECT *
FROM muecke_0.orte
WHERE orte.PLZ = 70599;

INSERT INTO muecke_0.kunden(kundennr, firma, strasse, ortnr, telefon)
VALUES(270057, "Meinhardt - Möbel nach Maß", "Neckarstraße 55","29675", "0711-453423");

INSERT INTO muecke_0.auftraege
    (auftraege.auftragsnr,auftraege.auftragsdatum, auftraege.kundennr)
VALUES (10022,'2012-10-05',270057);

INSERT INTO muecke_0.auftragspositionen(auftragsposnr, auftragsnr, posnr, fertigerznr, menge)
VALUES
(47,10022, 1, 200010,9),
(48,10022, 2, 200016, 2);
```

Die **referentielle Integrität** stellt sicher, dass eine Datenbank frei von Widersprüchen (konsistent) ist und keine verwaisten Datensätze enthält.

Bei der referentiellen Integrität können Datensätze, die einen Fremdschlüssel enthalten (Child-Tabelle) nur dann gespeichert werden, wenn ein entsprechender Wert im Primärschlüsselattribut der in Beziehung stehenden Tabelle (Parent-Tabelle) vorhanden ist.

Konkret:

Die Daten eines Kunden können nur dann gespeichert werden, wenn die zu erfassende Ortsnummer bereits in der Tabelle *orte* als Primärschlüsselwert existiert.

Umgekehrt kann ein Datensatz aus einer Parent-Tabelle nicht gelöscht werden, solange der Wert seines Primärschlüssels als Eintrag im Fremdschlüsselattribut der in Beziehung stehenden Tabelle (Child-Tabelle) vorhanden ist.

Konkret:

Die Daten eines Ortes können nur dann gelöscht werden, wenn kein Kunde den Wert der Ortsnummer in seinem Fremdschlüsselattribut aufweist.

Prinzipiell kann die referentielle Integrität beim Einfügen, Ändern und Löschen von Datensätzen verletzt werden. Solche Verletzungen führen zu Anomalien (Unregelmäßigkeiten) in der Datenbank.

2.3 Redundanzfreiheit (Keine Widersprüche in Daten)

L2_1 Redundanzfreiheit – Information

Ein funktionierendes relationales Datenbankmodell verlangt u.a., dass folgende Kriterien beachtet werden:

Primärschlüsselattribut:	Ein Primärschlüssel wird zur eindeutigen Identifizierung eines Datensatzes verwendet. Der Wert eines Primärschlüssels muss in einer Tabelle einmalig sein, da er jeden Datensatz eindeutig kennzeichnet.
Atomare Attributwerte:	Einem Attribut dürfen nicht mehrere Werte zugewiesen werden, d.h. in einem Datenfeld darf nur ein Wert enthalten sein.
Redundanzfreiheit:	Unter Redundanz versteht man eine doppelte bzw. mehrfache Speicherung gleicher Daten. Redundante Daten führen zu einem unnötigen Zeitaufwand bei der Erfassung, zur Verschwendung von Speicherplatz und zu erhöhten Verarbeitungszeiten. Außerdem können sie zu widersprüchlichen Daten führen, wenn eine Änderung einer Information nicht an allen Stellen der Datenbank vorgenommen wird. In diesem Fall spricht man von einer Änderungs-Anomalie, die zur Inkonsistenz der Daten führt. Ziel der Datenbankmodellierung muss es sein, solche Redundanzen zu vermeiden. Redundanzfreiheit liegt dann vor, wenn sämtliche Daten nur einmal erfasst und gespeichert werden, ohne dass ein Informationsverlust stattfindet. Redundanzfreiheit kann erreicht werden, indem die redundanten Daten in eigene Tabellen ausgelagert werden.