



JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN
PROFESSUR BWL – WIRTSCHAFTSINFORMATIK
UNIV.-PROF. DR. AXEL SCHWICKERT

Schwickert, Axel C.; Odermatt, Sven; Müller, Laura;
Patzak, Maximilian; Brauburger, Dhana

Relationale Datenbanken und MySQL – Reader zur WBT-Serie

ARBEITSPAPIERE WIRTSCHAFTSINFORMATIK

Nr. 02/2016

ISSN 1613-6667

Arbeitspapiere WI Nr. 2 / 2016

- Autoren:** Schwickert, Axel C.; Odermatt, Sven; Müller, Laura; Patzak, Maximilian; Brauburger, Dhana
- Titel:** Relationale Datenbanken und MySQL – Reader zur WBT-Serie
- Zitation:** Schwickert, Axel C.; Odermatt, Sven; Müller, Laura; Patzak, Maximilian; Brauburger, Dhana: Relationale Datenbanken und MySQL – Reader zur WBT-Serie, in: Arbeitspapiere WI, Nr. 2/2016, Hrsg.: Professur BWL – Wirtschaftsinformatik, Justus-Liebig-Universität Gießen 2016, 124 Seiten, ISSN 1613-6667.
- Kurzfassung:** Das vorliegende Arbeitspapier dient als Reader zur WBT-Serie „Relationale Datenbanken und MySQL“, die im E-Campus Wirtschaftsinformatik online zur Verfügung steht. Zunächst werden in einer Einführung Datenmanagement und das relationale Datenbankverwaltungssystem „MySQL“ erläutert. Anschließend folgen Planung und Erstellung einer Webshop-Datenbank, welche die Basis für das Einpflegen Abfragen und Verwalten der Produkt- und Kundendaten darstellt. Darauf aufbauen wird erläutert, wie Anpassungen und Optimierungen in einer Webshop-Datenbank vorgenommen werden können und wie komplexe Webshop-Daten abgefragt werden.
- Schlüsselwörter:** Relationale Datenbanken, Datenbank, MySQL, Datenmanagement, Webshop-Datenbank, Produktdaten, Kundendaten, Webshop-Daten, Datenbankmanagementsystem, DBMS, Entity-Relationship-Modell, ER-Modell, Relationales Datenschema, phpMy Admin Designer

A Zur Einordnung der WBT-Serie

Die WBT-Serie richtet sich an Interessenten des Themenbereiches "Relationale Datenbanken und MySQL".

Für Ihr Selbststudium per WBT müssen Sie einen Internet-Zugang haben - entweder auf Ihren eigenen PCs, auf den PCs im JLU-Hochschulrechenzentrum, in den JLU-Bibliotheken oder dem PC-Pool des Fachbereichs.

B Die Web-Based-Trainings

Der Stoff zu diesem Thema ist in Lerneinheiten zerlegt worden und wird durch eine Serie von Web-Based-Trainings (WBT) vermittelt. Mit Hilfe der Web-Based-Trainings (WBT) kann der Stoff im Eigenstudium erarbeitet werden. Die WBT bauen inhaltlich aufeinander auf und sollten in der angegebenen Reihenfolge absolviert werden.

WBT-Nr.	WBT-Bezeichnung	Bearbeitungs- dauer
1	Einführung in das Datenmanagement	45 Min.
2	Erste Schritte mit MySQL	45 Min.
3	Planung der Webshop-Datenbank	45 Min.
4	Erstellen der Webshop-Datenbank	45 Min.
5	Einpflegen der Produkt- und Kundendaten	45 Min.
6	Abfragen der Produkt- und Kundendaten	45 Min.
7	Verwalten der Produkt- und Kundendaten	45 Min.
8	Anpassen der Webshop-Datenbank	60 Min.
9	Abfragen komplexer Webshop-Daten	45 Min.
10	Optimieren der Webshop-Datenbank	45 Min.

Tab. 1: Übersicht WBT-Serie

Die Inhalte der einzelnen WBT werden nachfolgend in diesem Dokument gezeigt. Alle WBT stehen Ihnen rund um die Uhr online zur Verfügung. Sie können jedes WBT beliebig oft durcharbeiten. In jedem WBT sind enthalten:

- Vermittlung des Lernstoffes,
- interaktive Übungen zum Lernstoff,
- abschließende Tests zum Lernstoff.

Inhaltsverzeichnis

	Seite
A Zur Einordnung der WBT-Serie.....	I
B Die Web-Based-Trainings.....	II
Inhaltsverzeichnis.....	III
Abbildungsverzeichnis.....	VII
Tabellenverzeichnis.....	XI
Abkürzungsverzeichnis.....	XII
1 Die WBT-Serie "Relationale Datenbanken und MySQL".....	1
2 Einführung in das Datenmanagement	2
2.1 Einleitung	2
2.2 Der Auftrag.....	2
2.3 Was ist eine Datenbank?	3
2.4 Anwendung und Nutzen der Webshop-Datenbank	4
2.5 Relationale Datenbanken.....	5
2.6 Beziehungen zwischen Tabellen	5
2.7 Der Markt der DBMS.....	6
2.8 Allgemeines über MySQL.....	7
2.9 Zusammenfassung und Ausblick.....	7
3 Erste Schritte mit MySQL.....	8
3.1 Einleitung	8
3.2 Erstellung des Webshops.....	8
3.3 Vorbereitung zur Verwendung von MySQL.....	8
3.4 Das XAMPP Control Panel.....	18
3.5 Ports anpassen	19
3.6 Localhost aufrufen.....	20
3.7 phpMyAdmin	21
3.8 Übung: Erstellen einer Datenbank	22
3.9 Zusammenfassung und Ausblick.....	23
4 Planung der Webshop-Datenbank.....	24
4.1 Einleitung	24
4.2 Übersicht: Schritte der Datenmodellierung.....	24
4.3 Schritt 1: Datenanalyse I	25
4.4 Schritt 1: Datenanalyse II.....	25

4.5	Schritt 2: Konzeptuelles Datenschema.....	26
4.6	ER-Modell: Entitäten und Entitätsmengen.....	26
4.7	ER-Modell: Attribute	27
4.8	ER-Modell: Relationen.....	28
4.9	ER-Modell: Relationen II.....	29
4.10	Schritt 2: ER-Modell des Webshops	29
4.11	Schritt 3: Relationales Datenschema I.....	30
4.12	Schritt 3: Relationales Datenschema II	30
4.13	Zusammenfassung und Ausblick.....	32
5	Erstellen der Webshop-Datenbank	33
5.1	Einleitung	33
5.2	Wiederholung: Erstellen einer Datenbank	33
5.3	Wiederholung: Schritte der Datenmodellierung.....	33
5.4	Schritt 4: Physisches Datenschema	34
5.5	Erzeugen einer Tabelle: Kunden	35
5.6	Einstellungen einer Tabelle: Kunden	35
5.7	Übung: Tabellen erstellen	37
5.8	Kunden und Bestellungen verknüpfen	37
5.9	Übung: Tabellen verknüpfen.....	38
5.10	phpMyAdmin Designer.....	39
5.11	Übung: phpMyAdmin Designer.....	40
5.12	Zusammenfassung und Ausblick.....	41
6	Einpfelegen der Produkt- und Kundendaten	42
6.1	Einleitung	42
6.2	Produktdaten einpflegen mit phpMyAdmin.....	42
6.3	Übung: Produktdaten einpflegen mit phpMyAdmin.....	42
6.4	Datenübertragung I.....	43
6.5	Datenübertragung II	43
6.6	Was ist SQL?.....	44
6.7	Produktdaten einpflegen mit SQL I.....	46
6.8	Produktdaten einpflegen mit SQL II	46
6.9	Übung: Produktdaten einpflegen mit SQL.....	47
6.10	Kundendaten einpflegen.....	47
6.11	Kundendaten einpflegen II	49
6.12	Übung: Kunden- und Ortsdaten einpflegen.....	50
6.13	Zusammenfassung und Ausblick.....	51
7	Abfragen der Produkt- und Kundendaten	52

7.1	Einleitung	52
7.2	Kunden selektieren I.....	52
7.3	Kunden selektieren II	53
7.4	Angebot erstellen.....	54
7.5	Angebot eingrenzen.....	56
7.6	Angebot eingrenzen II	57
7.7	Werbeaktion	58
7.8	Werbeaktion II.....	60
7.9	Grundlage zur Übung	62
7.10	Übung: Abfragen der Produkt- und Kundendaten	62
7.11	Zusammenfassung und Ausblick.....	63
8	Verwalten der Produkt- und Kundendaten.....	64
8.1	Einleitung	64
8.2	Preisänderung eines Produkts I	64
8.3	Preisänderung eines Produkts II	65
8.4	Umzug eines Kunden	66
8.5	Kundenkonto löschen	67
8.6	Sortimentsänderung I	68
8.7	Sortimentsänderung II	70
8.8	Übung: Verwalten der Produkt- und Kundendaten I.....	71
8.9	Grundlage zur Übung	71
8.10	Übung: Verwalten der Produkt- und Kundendaten II	72
8.11	Zusammenfassung und Ausblick.....	73
9	Anpassen der Webshop-Datenbank	74
9.1	Lieferanten- und Kunden-Tabelle anpassen	74
9.1.1	Einleitung	74
9.1.2	Anpassen der Lieferanten-Tabelle	74
9.1.3	Anpassen der Kunden-Tabelle	75
9.1.4	Übung: Lieferanten- und Kunden-Tabelle anpassen.....	76
9.2	Produkte kategorisieren	76
9.2.1	Produkte kategorisieren.....	76
9.2.2	Produktkategorien anlegen.....	77
9.2.3	Anpassen der Produkt-Tabelle	78
9.2.4	Übung: Produkte kategorisieren.....	79
9.2.5	Produkte und Produktkategorien verknüpfen I	80
9.2.6	Produkte und Produktkategorien verknüpfen II.....	81
9.2.7	Übung: Produkte und Produktkategorien verknüpfen.....	82
9.2.8	Grundlage zur Übung.....	82

9.2.9	Übung: Anpassen der Webshop-Datenbank	82
9.2.10	Zusammenfassung und Ausblick	83
10	Abfragen komplexer Webshop-Daten.....	84
10.1	Einleitung	84
10.2	Übung: Datenbank importieren	84
10.3	Sommerangebot.....	84
10.4	Sommerangebot: Kunden- und Ortsdaten abfragen	85
10.5	Sommerangebot: Abfrageergebnis	86
10.6	Sommerangebot: Adressliste erstellen	87
10.7	Bestellübersicht erstellen I	88
10.8	Bestellübersicht erstellen II.....	89
10.9	Marketingmaßnahme planen	90
10.10	Grundlage zur Übung	92
10.11	Übung: Komplexe Abfragen	92
10.12	Zusammenfassung und Ausblick.....	94
11	Optimieren der Webshop-Datenbank	95
11.1	Einleitung	95
11.2	Übung: Datenbank importieren	95
11.3	Problem: Kundenanfragen.....	95
11.4	Optimieren der Kunden-Tabelle.....	96
11.5	Vor- und Nachteile eines Indexes	97
11.6	Übung: Optimieren der Webshop-Datenbank.....	97
11.7	Zusammenfassung und Ausblick.....	98
Anhang		XVI
Literaturverzeichnis.....		XXIII
Eidesstattliche Erklärung		

Abbildungsverzeichnis

	Seite
Abb. 1: Beispielhafte Modellansicht einer Datenbank.....	3
Abb. 2: Zusammenhang Datenbank, DBMS und DBS	4
Abb. 3: 1:1-Beziehung	5
Abb. 4: 1:n-Beziehung	5
Abb. 5: n:m-Beziehung	6
Abb. 6: Kommerzielle DBMS Anbieter (Stand 2011).....	6
Abb. 7: Startseite Apache Friends.....	9
Abb. 8: Abfragefenster bei der Installation von XAMPP	9
Abb. 9: Warnmeldung bei der Installation von XAMPP	10
Abb. 10: XAMPP Control Panel	11
Abb. 11: XAMPP Control Panel.....	11
Abb. 12: Textdatei.....	12
Abb. 13: Apache und MySQL starten.....	12
Abb. 14: Startseite MAMP.....	13
Abb. 15: Download MAMP.....	14
Abb. 16: Warnmeldung bei der Installation von MAMP.....	14
Abb. 17: Abfragefenster bei der Installation von MAMP.....	15
Abb. 18: Spotlight-Suche nach „MAMP“	15
Abb. 19: Suche von MAMP.....	16
Abb. 20: Warnmeldung beim Starten von MAMP	16
Abb. 21: Server starten in MAMP	17
Abb. 22: Warnmeldung beim Starten der Server in MAMP.....	17
Abb. 23: XAMPP Control Panel	18
Abb. 24: Konfig-Button im XAMPP Control Panel	19
Abb. 25: Fehlermeldung im XAMPP Control Panel.....	20
Abb. 26: Localhost	21
Abb. 27: Graphische Benutzeroberfläche phpMyAdmin.....	22
Abb. 28: Informationen zur Datenanalyse	26
Abb. 29: Zusammenhang zwischen Entitätsmengen und Entitäten	26
Abb. 30: Darstellung von Entitätsmengen im ER-Modell	27
Abb. 31: Darstellung von Attributen im ER-Modell.....	27

Abb. 32: Primärschlüssel.....	28
Abb. 33: Darstellung von Relationen und Kardinalitäten im ER-Modell.....	28
Abb. 34: 1:1-Beziehung	29
Abb. 35: 1:n-Beziehung	29
Abb. 36: n:m-Beziehung	29
Abb. 37: ER-Modell des Webshops.....	30
Abb. 38: Auflösung der n:m-Beziehung	30
Abb. 39: Relationales Datenschema des Webshops.....	31
Abb. 40: Dateien der Webshop-Datenbank.....	34
Abb. 41: Tabellenstruktur der Kunden-Tabelle	35
Abb. 42: Einstellungen in der Kunden-Tabelle.....	36
Abb. 43: Tabellen erstellen in der Webshop-Datenbank	37
Abb. 44: Kunden und Bestellungen verknüpfen	37
Abb. 45: Tabellen verknüpfen in der Webshop-Datenbank.....	38
Abb. 46: phpMyAdmin Designer.....	39
Abb. 47: phpMyAdmin Designer öffnen	40
Abb. 48: Datenbank-Tabellen sortieren mit dem phpMyAdmin Designer.....	40
Abb. 49: Produkte der Lemonline AG	42
Abb. 50: Datenbanksprache SQL.....	43
Abb. 51: Datenbanksprache SQL.....	44
Abb. 52: Tabellenstruktur der Produkte-Tabelle.....	44
Abb. 53: Abfrageergebnis des SQL-Befehls.....	45
Abb. 54: Tabellenstruktur der Produkte-Tabelle.....	46
Abb. 55: Produkte der Lemonline AG	47
Abb. 56: Orte- und Kunden-Tabelle	48
Abb. 57: Tabellen ohne Fremdschlüssel	49
Abb. 58: Tabellen mit Fremdschlüsseln.....	50
Abb. 59: Kunden der Lemonline AG	51
Abb. 60: Ergebnis der Abfrage	52
Abb. 61: Tabellenstruktur der Kunden-Tabelle	53
Abb. 62: Ergebnis der Abfrage	54
Abb. 63: Tabellenstruktur der Produkte-Tabelle.....	55
Abb. 64: Ergebnis der Abfrage aufsteigend sortiert.....	55
Abb. 65: Ergebnis der Abfrage absteigend sortiert.....	56

Abb. 66: Ergebnis der Abfrage	57
Abb. 67: Ergebnis der Abfrage	58
Abb. 68: Ergebnis der Abfrage mit dem SQL-Befehl „SELECT DISTINCT“	59
Abb. 69: Tabellenstruktur der Orte-Tabelle	59
Abb. 70: Ergebnis der Abfrage mit dem SQL-Befehl „SELECT“	60
Abb. 71: Ergebnis der Abfrage	61
Abb. 72: Webshop-Datenbank der Lemonline AG	62
Abb. 73: Ergebnis der Abfrage	64
Abb. 74: Ergebnis der Abfrage	65
Abb. 75: Ergebnis der Abfrage	66
Abb. 76: Tabellenstruktur der Kunden-Tabelle	67
Abb. 77: Ergebnis der Abfrage	68
Abb. 78: Ergebnis der Abfrage	69
Abb. 79: Ergebnis der Abfrage	70
Abb. 80: Webshop-Datenbank der Lemonline AG	71
Abb. 81: Lieferanten-Tabelle vor der Anpassung	74
Abb. 82: Lieferanten-Tabelle nach der Anpassung	75
Abb. 83: Kunden-Tabelle vor der Anpassung	75
Abb. 84: Kunden-Tabelle nach der Anpassung	76
Abb. 85: Produkte kategorisieren	77
Abb. 86: Tabellenstruktur der Produktkategorien-Tabelle	77
Abb. 87: Produkte-Tabelle vor der Anpassung	78
Abb. 88: Produkte-Tabelle nach der Anpassung	79
Abb. 89: Produkte und Produktkategorien verknüpfen	81
Abb. 90: Webshop-Datenbank der Lemonline AG	82
Abb. 91: Schnittmenge der Kunden- und Orte-Tabelle	85
Abb. 92: Orte- und Kunden-Tabelle	85
Abb. 93: Ergebnis der Abfrage	86
Abb. 94: Ergebnis der Abfrage	87
Abb. 95: Schnittmenge der Kunden- und Bestellungen-Tabelle	88
Abb. 96: Ergebnisausgabe bei der LEFT JOIN-Abfrage	88
Abb. 97: Ergebnis der Abfrage	89
Abb. 98: Ergebnis der Abfrage	91
Abb. 99: Webshop-Datenbank der Lemonline AG	92

Abb. 100: Vorgehensweise des DBMS	96
Abb. 101: Zusammenhang Kunden-Tabelle und Index	97

Tabellenverzeichnis

	Seite
Tab. 1: Übersicht WBT-Serie.....	II
Tab. 2: Übungsfragen WBT 06.....	63
Tab. 3: Übungsfragen WBT 07.....	73
Tab. 4: Produkte und deren Produktkategorien.....	80
Tab. 5: Übungsfragen WBT 08.....	83
Tab. 6: Übungsfragen WBT 09.....	94
Tab. 7: Lösung zur Übung - WBT 06.....	XVII
Tab. 8: Lösung zur Übung - WBT 07.....	XVIII
Tab. 9: Lösung zur Übung - WBT 08.....	XX
Tab. 10: Lösung zur Übung - WBT 09.....	XXII

Abkürzungsverzeichnis

AB	Aktiebolag
AG	Aktiengesellschaft
DB	Datenbank
DBMS.....	Datenbankmanagementsystem
DBS	Datenbanksystem
E-Mail.....	Electronic Mail
ER.....	Entitiy Relationship
ID.....	Identität (eindeutiger Name)
IT	Information Technology
MS	Microsoft
PLZ.....	Postleitzahl
SQL	Structured Query Language

1 Die WBT-Serie "Relationale Datenbanken und MySQL"

Ziel der WBT-Serie "Relationale Datenbanken und MySQL" ist es, die Funktionsweise von relationalen Datenbanken und MySQL bei der exemplarischen Erstellung einer Webshop-Datenbank zu verdeutlichen. Anhand eines Beispiels wird eine Webshop-Datenbank geplant, erstellt und geändert.

Einführend werden zunächst allgemeine Grundlagen zum Datenmanagement vermittelt und Vorbereitungen zur Verwendung von MySQL getroffen. Anhand des Beispielunternehmens „Lemonline AG“ wird im dritten WBT verdeutlicht, wie eine Webshop-Datenbank geplant wird. Anschließend wird ab dem vierten WBT dargelegt, wie eine Webshop-Datenbank mit Hilfe von phpMyAdmin erstellt und mit Daten befüllt wird.

Ab dem fünften WBT wird aufgezeigt, wie mit Hilfe von SQL-Befehlen Daten in der Webshop-Datenbank eingepflegt, abgefragt, geändert und gelöscht werden können. Zudem wird vermittelt, wie mit Hilfe von SQL-Befehlen Änderungen an der Struktur der Webshop-Datenbank vorgenommen werden können.

Abschließend wird in WBT 10 verdeutlicht, wie die Webshop-Datenbank optimiert werden kann.

2 Einführung in das Datenmanagement

2.1 Einleitung

Hallo! Ich bin Willi und arbeite als selbstständiger Web-Entwickler. Ich freue mich, dass Sie mich in diesem Semester begleiten. Ich möchte Ihnen einige Grundlagen zu relationalen Datenbanken und MySQL näherbringen. Dann können Sie mir bald zur Hand gehen und einige meiner Aufträge übernehmen.

Gestern habe ich einen neuen Auftrag von der Lemonline AG erhalten. Sie benötigt eine MySQL-Datenbank für ihren neuen Webshop. Am besten zeige ich Ihnen anhand dieser Webshop-Datenbank den Umgang mit MySQL.

2.2 Der Auftrag

Vor einigen Tagen habe ich von der Lemonline AG einen neuen Auftrag erhalten. Sie besitzt ein Ladengeschäft auf unserer Einkaufsstraße und vertreibt dort Smartphones und Tablets. Nun möchte sie ihre Artikel auch über einen Webshop vertreiben. Ein erster Schritt auf diesem Weg ist das Erstellen einer Datenbank.

In Ihrer Einarbeitungsphase werden wir gemeinsam eine erste Version der Datenbank für unseren Kunden erstellen. Da Sie bisher keine Erfahrungen in der Entwicklung von Datenbanken haben, werde ich später die Datenbank für die Lemonline AG vervollständigen.

Um mit diesem Auftrag zu beginnen, werden wir uns zunächst mit einigen Grundlagen befassen müssen.

Um einen ersten Eindruck zu vermitteln, sehen Sie eine beispielhafte Modellansicht einer Datenbank (Abb. 1). Das Modell zeigt, dass in Datenbanken Objekte über Beziehungen miteinander verbunden werden.

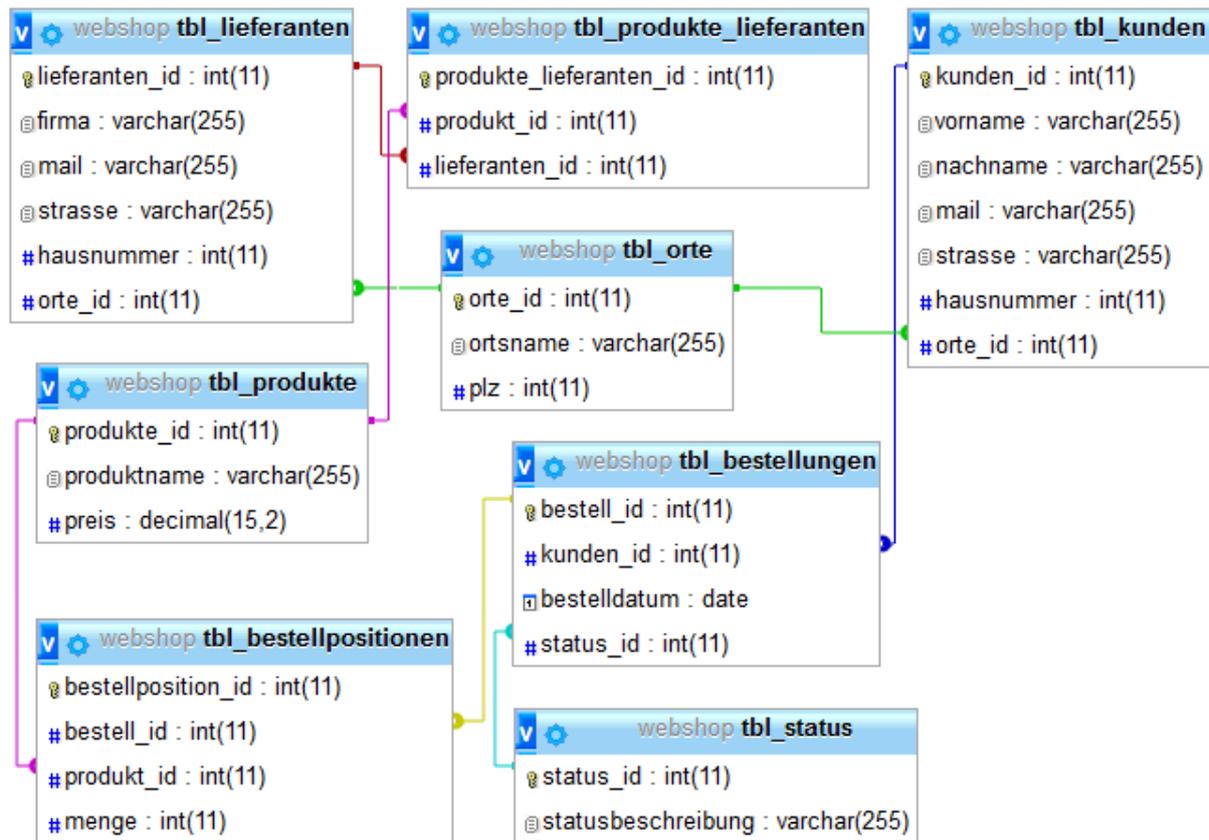


Abb. 1: Beispielhafte Modellansicht einer Datenbank

2.3 Was ist eine Datenbank?

Zum Betrieb eines Webshops müssen große Datenmengen wie Rechnungen, Bestellungen und Kontaktdaten von Kunden und Lieferanten gespeichert werden. Datenbanken werden z. B. dafür genutzt Kundendaten schnell zu finden und zu aktualisieren oder auch Daten aus Bestellungen strukturiert bereitzustellen.

- **Datenbanken** sind Speicherorte, an denen Daten abgelegt und zur Verwaltung bereitgestellt werden können. Für den Webshop werden hier z. B. Kunden- und Lieferantendaten gespeichert.
- Mit einem **DBMS** können Datenbanken erzeugt und verwaltet werden. Ein DBMS ermöglicht es, mehrere Webshops in einem Datenbanksystem zu verwalten. Kleine und auch große Mengen gespeicherter Daten können so leicht verwaltet und ausgewertet werden. Ein DBMS ist beispielsweise MySQL. Mit MySQL werden wir uns in dieser WBT-Serie intensiv befassen.

Grundsätzlich wird zwischen Datenbanken und Datenbankmanagementsystemen (DBMS) unterschieden. Die Kombination aus DBMS und Datenbank wird als Datenbanksystem bezeichnet (Abb.2).

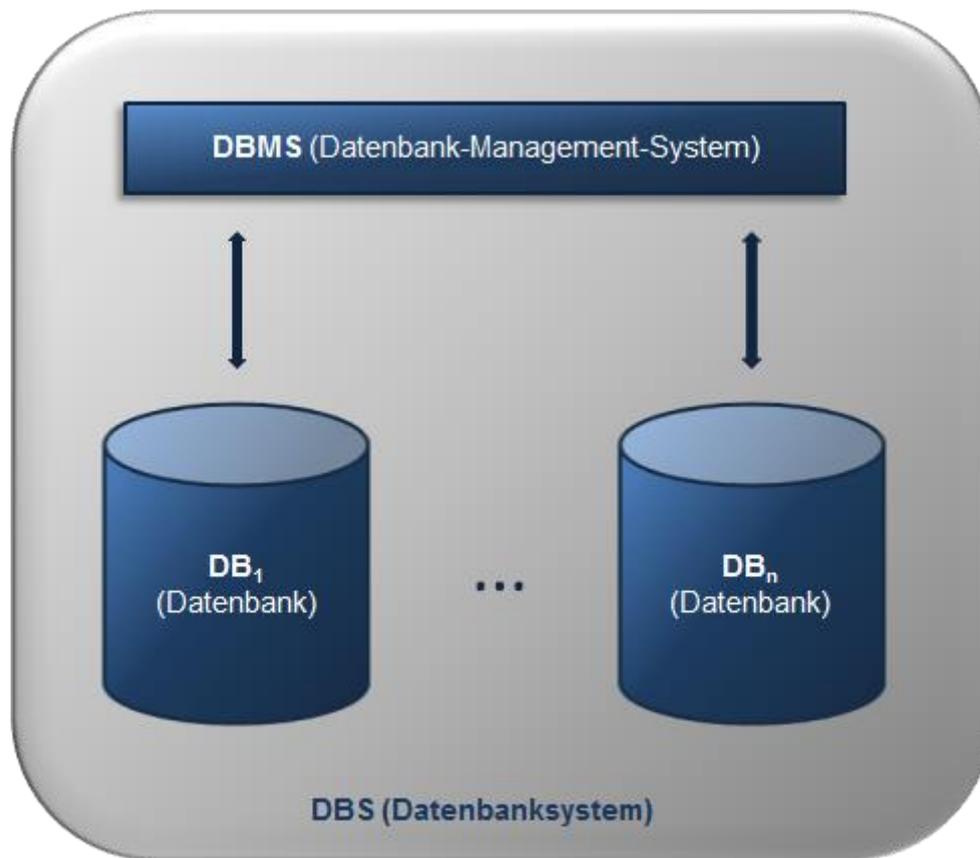


Abb. 2: Zusammenhang Datenbank, DBMS und DBS

Grundlegende Informationen zu Datenbanken erhalten Sie in der WBT-Serie "Datenmodellierung und Datenbanksysteme".

2.4 Anwendung und Nutzen der Webshop-Datenbank

Datenbanken bilden die Grundlage von Webshops. In ihnen werden u. a. alle Kunden und deren Bestellungen gespeichert. Innerhalb der Webshop-Datenbank können Daten gezielt gefiltert werden. Die Lemonline AG könnte so z. B. eine Werbeaktion für alle Kunden gestalten, die mehr als zehn Bestellungen im Jahr tätigen.

Da die Lemonline AG bereits ein eigenes Ladengeschäft hat, könnte sie außerdem alle Webshop-Kunden der Region aus der Datenbank filtern und zu einer Rabattaktion in ihr Ladengeschäft einladen. Unser Kunde wird folglich vom vielfältigen Nutzen der Datenbank profitieren.

Im Gegensatz zu einer unstrukturierten Datensammlung wie z. B. in Textdokumenten, sind die Daten in einem Datenbanksystem dann:

- leicht auffindbar,
- leicht zu ändern,

- mit anderen Daten verknüpfbar und
- redundanzfrei ablegbar.

2.5 Relationale Datenbanken

Im Laufe der Zeit haben sich verschiedene Datenbankmodelle entwickelt. Bei der Entwicklung von Webshops wird relationalen Datenbanken eine große Bedeutung beigemessen. Daher werden wir für unseren Kunden eine relationale Datenbank erstellen.

Relationale Datenbanken zeichnet aus, dass sie aus einzelnen Objekten bestehen, die miteinander in Beziehungen (Relationen) stehen. Bei einer Bestellung im Webshop können so beispielsweise die Kundendaten direkt in der Bestellung verknüpft werden.

2.6 Beziehungen zwischen Tabellen

Objekte werden in Datenbanken als Tabellen umgesetzt. Zwei Tabellen können über die folgenden Beziehungen miteinander verknüpft werden:

In einer 1:1-Beziehung kann eine Ehefrau (1) in Tabelle A genau einem Ehemann (1) in Tabelle B zugeordnet werden (Abb.3).



Abb. 3: 1:1-Beziehung

In einer 1:n-Beziehung können jedem Kunden des Webshops (1) in Tabelle A mehrere Rechnungen (n) in Tabelle B zugeordnet werden (Abb.4).

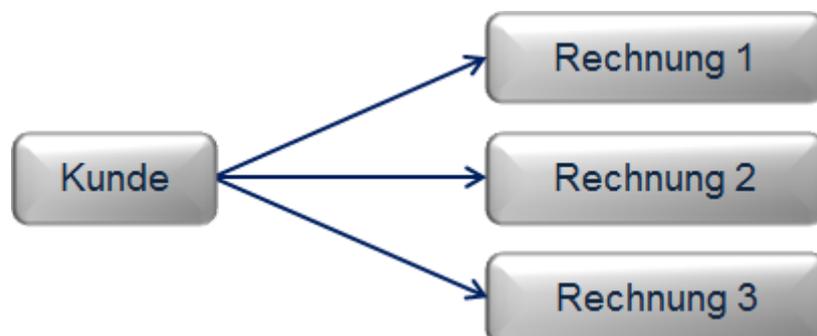


Abb. 4: 1:n-Beziehung

In einer n:m-Beziehung kann jeder Kunde (n) in Tabelle A verschiedenen Webshops (m) in Tabelle B zugeordnet werden, da jeder Kunde in mehreren Webshops einkaufen kann (Abb.5).

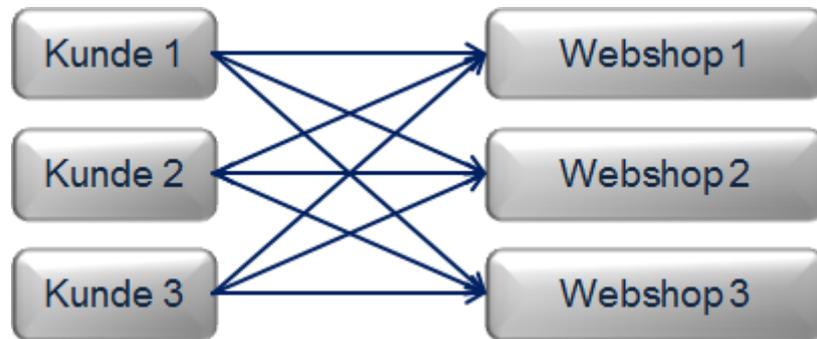


Abb. 5: n:m-Beziehung

2.7 Der Markt der DBMS

Relationale Datenbanken werden mit einem Datenbankmanagementsystem (DBMS) wie z. B. MySQL erstellt. MySQL ist kostenfrei verfügbar und dadurch weltweit stark verbreitet. In dieser WBT-Serie werden wir daher MySQL verwenden.

Kommerzielle Konkurrenten von MySQL sind z. B. "Oracle Database" von Oracle, "MS SQL Server" von Microsoft, "DB2" von IBM sowie "Sybase" von SAP (Abb.6).

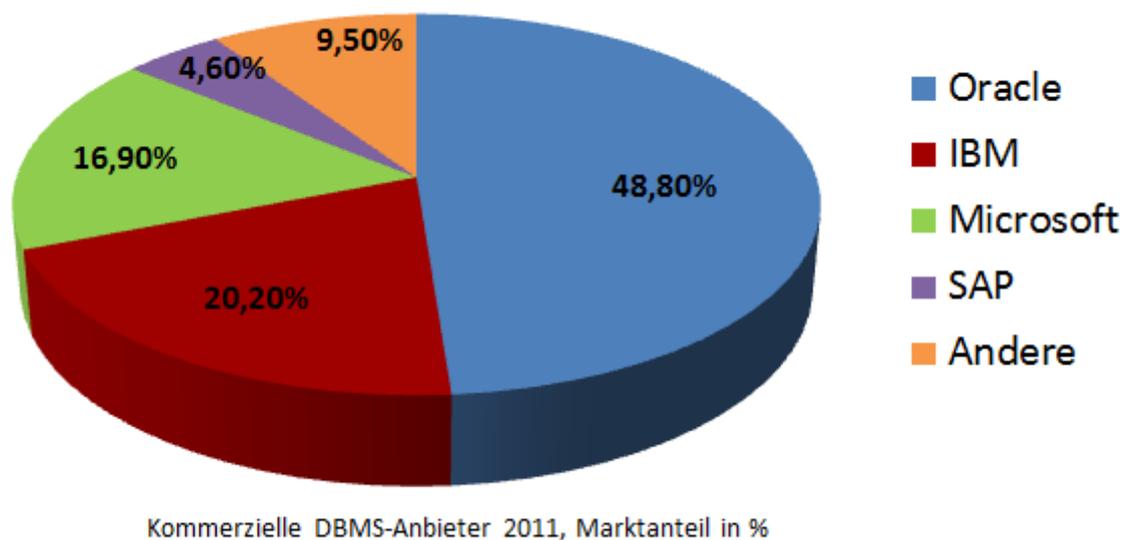


Abb. 6: Kommerzielle DBMS Anbieter (Stand 2011)

Wiederholung: Mit einem DBMS können Datenbanken erzeugt und verwaltet werden. Ein DBMS ermöglicht es, mehrere Webshops in einem Datenbanksystem zu verwalten. Kleine und auch große Mengen gespeicherter Daten können so leicht verwaltet und ausgewertet werden.

2.8 Allgemeines über MySQL

MySQL ist eines der weltweit gängigsten relationalen Datenbankmanagementsysteme. Es wurde 1994 vom schwedischen Unternehmen "MySQL AB" entwickelt, um Datenbanken in Web-Anwendungen verfügbar zu machen. Mit Hilfe von Programmiersprachen wie z. B. PHP oder Java kann MySQL in Web-Anwendungen eingebunden werden. Daher wird MySQL heute für viele Web-Auftritte wie z. B. Facebook, Youtube oder Twitter genutzt.

MySQL kann im privaten und im geschäftlichen Kontext in der Regel kostenfrei verwendet werden. Bei einer bestimmten Nutzung von MySQL können jedoch auch Kosten anfallen.

MySQL wird sowohl als nicht-kommerzielle Open-Source-Software als auch unter einer kommerziellen Lizenz angeboten. Es ist notwendig eine kommerzielle MySQL-Lizenz zu erwerben, wenn Softwareprodukte kommerziell vertrieben werden, die eine MySQL-Datenbank enthalten.

2.9 Zusammenfassung und Ausblick

In diesem WBT haben Sie die Grundlagen zum Datenmanagement kennengelernt. Diese Grundlagen benötigen Sie, damit wir gemeinsam eine Datenbank für den Webshop der Lemonline AG erstellen können.

Im nächsten WBT werden wir hierfür einige Werkzeuge auf Ihrem PC einrichten und anschließend damit eine Datenbank erstellen.

3 Erste Schritte mit MySQL

3.1 Einleitung

Im letzten WBT haben Sie die Grundlagen zum Datenmanagement kennengelernt. Sie wissen nun, was Datenbanken sind und wie sie verwendet werden. Außerdem haben Sie MySQL kennengelernt.

In diesem WBT werden wir einige Werkzeuge auf Ihrem PC einrichten, damit wir gemeinsam die Webshop-Datenbank der Lemonline AG in MySQL erstellen können.

3.2 Erstellung des Webshops

Super, dass Sie schon da sind. Damit wir die MySQL-Datenbank für den Webshop erstellen können, benötigen Sie die folgenden Werkzeuge auf Ihrem PC:

- einen MySQL-Server, auf dem Daten gespeichert werden und
- einen Web-Server für die Nutzung der graphische Benutzeroberfläche (phpMyAdmin) des MySQL-Servers.

Eine einfache Installation dieser Werkzeuge ermöglicht uns die kostenfreie Software XAMPP. Daher werden wir in dieser WBT-Serie mit XAMPP arbeiten.

3.3 Vorbereitung zur Verwendung von MySQL

Die Installation und Konfiguration von XAMPP wird in wenigen Schritten durchgeführt. Führen Sie die folgenden Schritte durch:

1. Laden Sie XAMPP passend für ihr Betriebssystem unter folgendem Link herunter: <https://www.apachefriends.org/de/index.html>
2. Speichern Sie die Installationsdatei. Dieser Vorgang kann einige Minuten dauern.
3. Starten Sie die Installation durch Öffnen der Installationsdatei. Die Installationsdatei erkennen Sie an der Dateierdung ".exe".
4. Folgen Sie dem Installationsassistenten, der Sie durch die Installation führt. Installieren Sie alle XAMPP-Komponenten auf dem vorgegebenen Pfad, da sonst weitere Konfigurationen notwendig werden.

XAMPP ist ein Installationspaket für die verschiedenen Betriebssysteme **X**, das **A**pache, **M**ariaDB (eine Abspaltung von MySQL), **P**erl und **P**HP enthält.

Ausführliche Installationsanleitung für Windows:

XAMPP ist eine kostenlose Software und ermöglicht den lokalen Betrieb eines Webservers inklusive Datenbank.

- a) Besuchen Sie folgende Web Site: <https://www.apachefriends.org/de/index.html>
- b) Klicken Sie auf den „XAMPP for Windows“-Button, um XAMPP herunterzuladen (Abb.7).

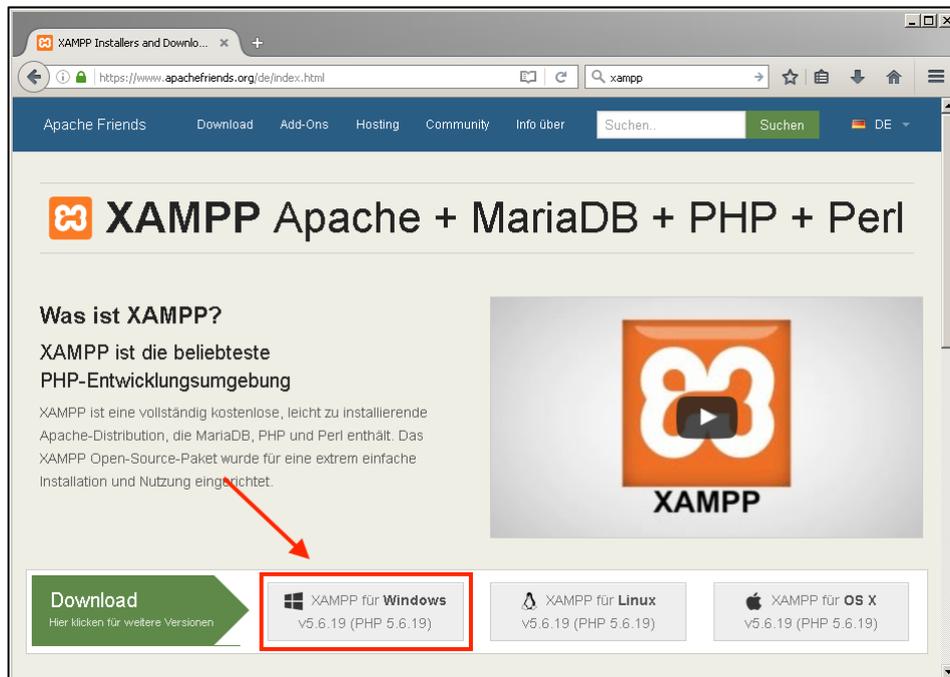


Abb. 7: Startseite Apache Friends

- c) Öffnen Sie die Datei und bestätigen Sie folgendes Fenster mit „Yes“ (Abb.8).



Abb. 8: Abfragefenster bei der Installation von XAMPP

- d) Bestätigen Sie die Warnung mit „OK“ (Abb.9).



Abb. 9: Warnmeldung bei der Installation von XAMPP

- e) Klicken Sie nun im ersten Schritt des Installationsfensters auf „Next >“.
- f) Nun bestätigen Sie auch den Schritt „Select Components“ mit „Next >“.
- g) Im nächsten Installationsschritt „Installation folder“ achten Sie darauf, dass folgender Pfad bei „Select a Folder“ eingetragen ist: „C:\xampp“ (ohne Anführungszeichen) und klicken Sie wieder auf „Next >“.
- h) Bei Schritt „Bitnami for XAMPP“ entfernen Sie den Haken bei „Learn more about Bitnami for XAMPP“ und klicken Sie auf „Next >“.
- i) Im vorletzten Schritt „Ready to Install“ klicken Sie wieder auf „Next >“.
- j) Der letzte Schritt „Completing the XAMPP Setup Wizard“ signalisiert Ihnen, dass die Installation von XAMPP erfolgreich war. Klicken Sie auf „Finish“.
- k) Ein Fenster mit dem Titel „Sprache“ öffnet sich, klicken Sie auf „Speichern“.
- l) Wenn Ihre Firewall Sie nun fragt, ob Sie „Apache“ und/ oder „MySQL“ Zugriff gewähren wollen, klicken Sie „nur private Netzwerke zulassen“ an und bestätigen Sie die Meldung mit „OK“.

m) Es öffnet sich der XAMPP Control Panel. Klicken Sie auf „Konfig“ (4) (Abb. 10).

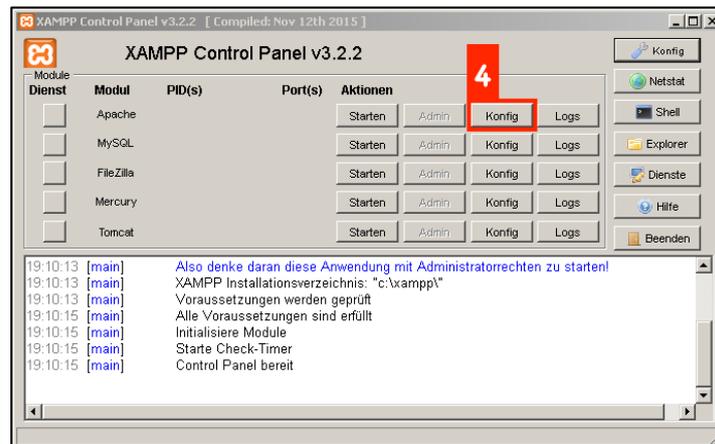


Abb. 10: XAMPP Control Panel

n) Klicken Sie nun auf „Apache (httpd.conf)“ (5) (Abb.11).

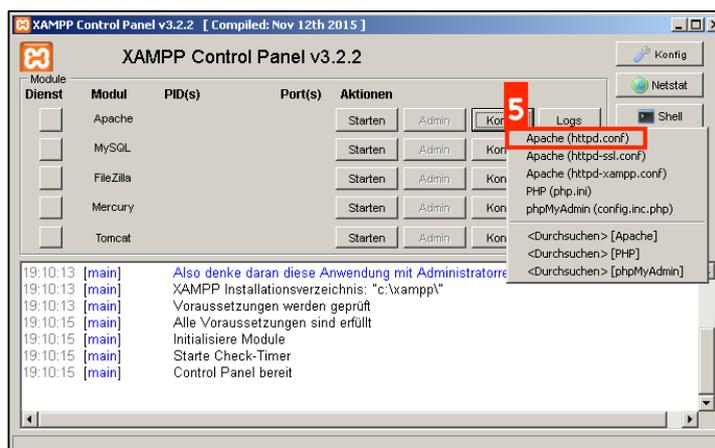


Abb. 11: XAMPP Control Panel

- o) Es öffnet sich eine Textdatei. Suchen Sie nach dem Ausdruck „Listen 80“ und verändern Sie diesen in „Listen 8087“ (ohne Anführungsstriche) und speichern Sie die Datei (Abb.12).

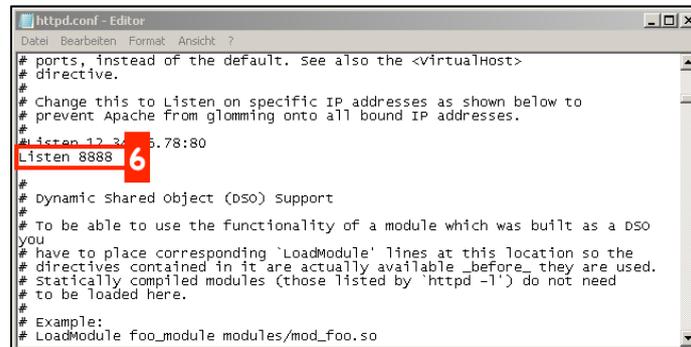


Abb. 12: Textdatei

- p) Klicken Sie im XAMPP Control Panel auf den „Konfig“-Button in der oberen rechten Ecke des Fensters und anschließend auf den Button „Dienste und Ports einstellen“. Verändern Sie nun den „Hauptport“ in der Registerkarte „Apache“ auf „8087“ (ohne Anführungszeichen) und bestätigen Sie Ihre Eingabe mit „Speichern“.
- q) Klicken Sie weiterhin im XAMPP Control Panel auf die „Starten“-Buttons in den Zeilen der Module „Apache“ und „MySQL“ (Abb. 13).

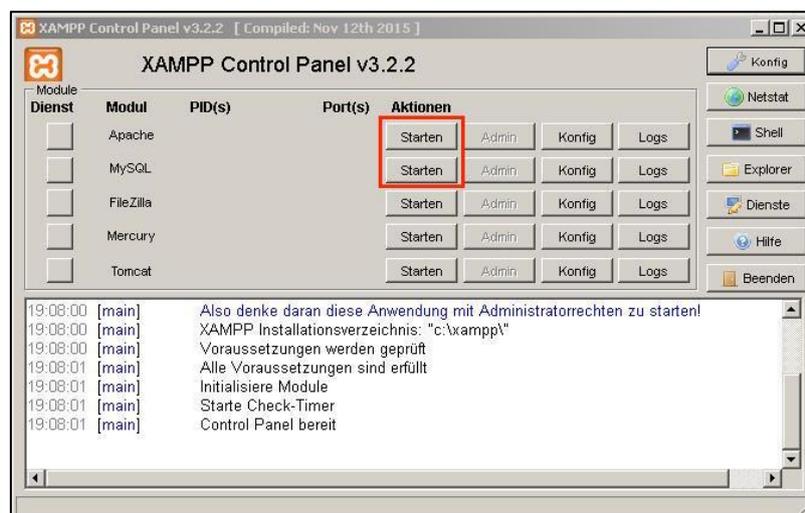


Abb. 13: Apache und MySQL starten

- r) Wenn Ihre Firewall Sie nun fragt, ob Sie „Apache“ und „MySQL“ Zugriff gewähren wollen, klicken Sie „nur private Netzwerke zulassen“ an und bestätigen Sie die Meldung mit „OK“.

Wichtig: Schließen Sie **nicht** das XAMPP Control Panel!

Sie haben nun XAMPP erfolgreich installiert.

Ausführliche Installationsanleitung für OS X:

MAMP ist eine kostenlose Software und ermöglicht den lokalen Betrieb eines Webservers inklusive Datenbank.

- a) Besuchen Sie folgende Web Site: <https://www.mamp.info/de/>
- b) Klicken Sie auf den „Download“-Button (1) unterhalb des grauen MAMP-Logos (Abb.14).

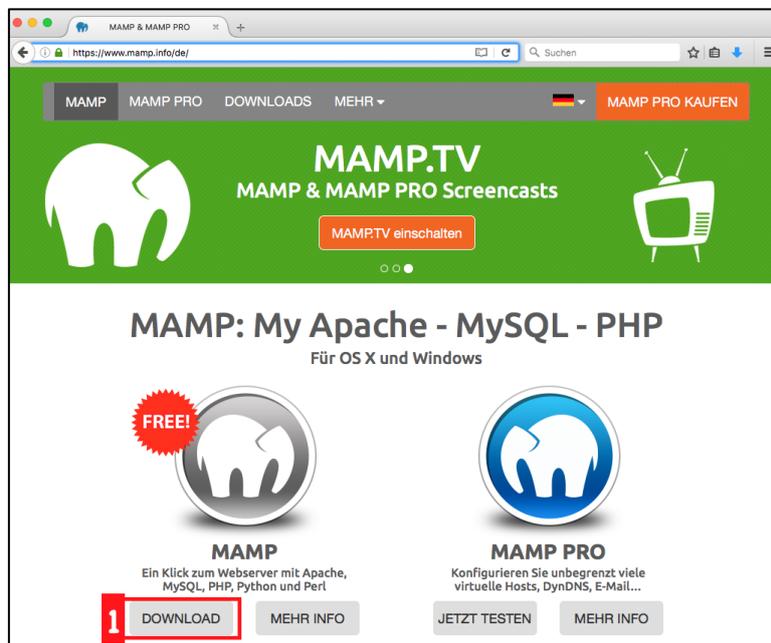


Abb. 14: Startseite MAMP

- c) Klicken Sie auf der folgenden Seite auf den „Download“-Button (2) und speichern Sie die Datei in Ihrem Download-Ordner (Abb.15).

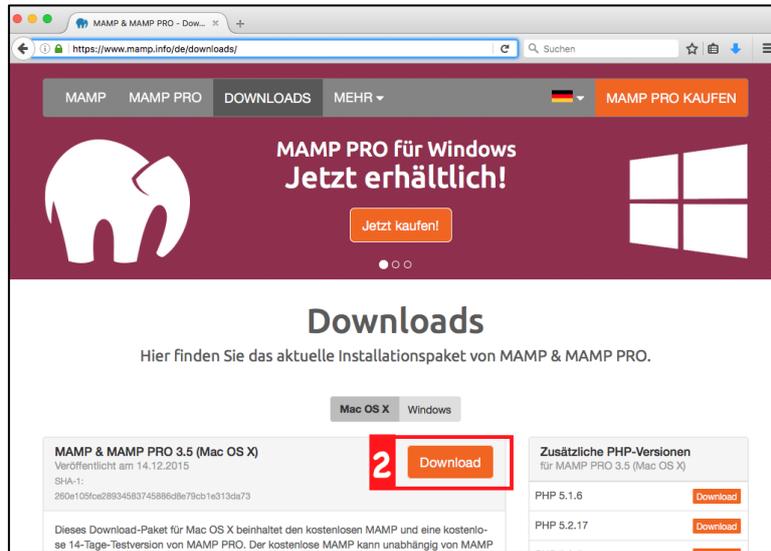


Abb. 15: Download MAMP

- d) Navigieren Sie zu Ihrem Download-Ordner und öffnen Sie die heruntergeladene Datei mit dem Namen „MAMP_MAMP_PRO_3.5.pkg“. Geben Sie zur Ausführung des Programms Ihr Passwort ein, falls dieses abgefragt wird.
- e) Falls Ihnen folgende Meldung gezeigt wird, bestätigen Sie diese mit „Öffnen“ (Abb.16).

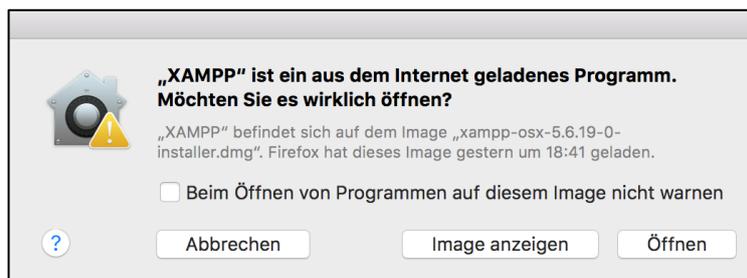


Abb. 16: Warnmeldung bei der Installation von MAMP

- f) Falls Sie nach Ihrem Benutzernamen/ Passwort gefragt werden, geben Sie diese ein und bestätigen Sie mit „OK“ (Abb.17).

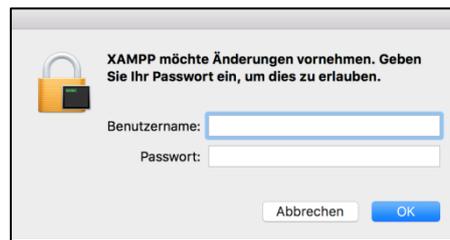


Abb. 17: Abfragefenster bei der Installation von MAMP

- g) Klicken Sie nun im ersten Schritt des Installationsfensters (Einführung) auf „Fortfahren“.
- h) Nun bestätigen Sie auch den nächsten Schritt (Bitte Lesen) mit „Fortfahren“.
- i) Den Installationsschritt (Lizenz) bestätigen Sie ebenfalls mit „Fortfahren“. Und klicken anschließend auf „Akzeptieren“ im sich öffnenden Fenster.
- j) Falls Sie den nächsten Installationsschritt „Zielvolumen auswählen“ angezeigt bekommen. Klicken Sie auf die Zeile „Für alle Benutzer an diesem Computer installieren“ und anschließend auf „Fortfahren“.
- k) Bestätigen Sie den Schritt „Installationstyp“ mit „Installieren“ und geben Sie im sich öffnenden Fenster Ihr Passwort ein und klicken Sie auf „Software installieren“.
- l) Der letzte Schritt „Zusammenfassung“ signalisiert Ihnen, dass die Installation von MAMP erfolgreich war. Klicken Sie auf „Abschließen“.
- m) Öffnen Sie MAMP:

Suchen Sie dazu nach „MAMP“ über die Spotlight-Suche, falls Sie MAMP nicht finden können, fahren Sie alternativ fort (Abb. 18).

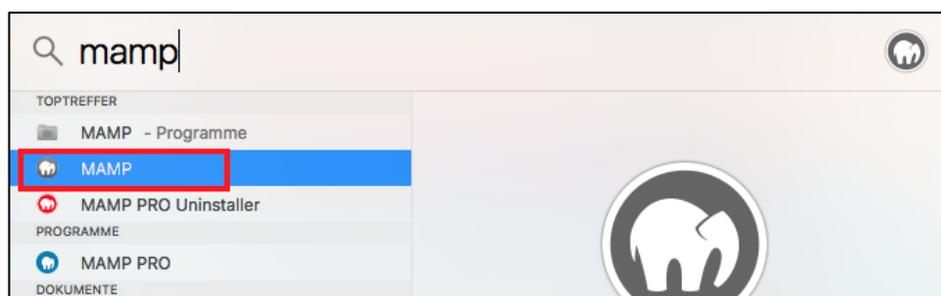


Abb. 18: Spotlight-Suche nach „MAMP“

Alternativ: Öffnen Sie den Finder, klicken Sie auf Programme in der vertikalen Navigationsleiste und scrollen Sie zu „MAMP“ (nicht „MAMP Pro“). Öffnen Sie mit Doppelklick den MAMP-Ordner und klicken Sie mit Doppelklick auf „MAMP“ (Abb.19).

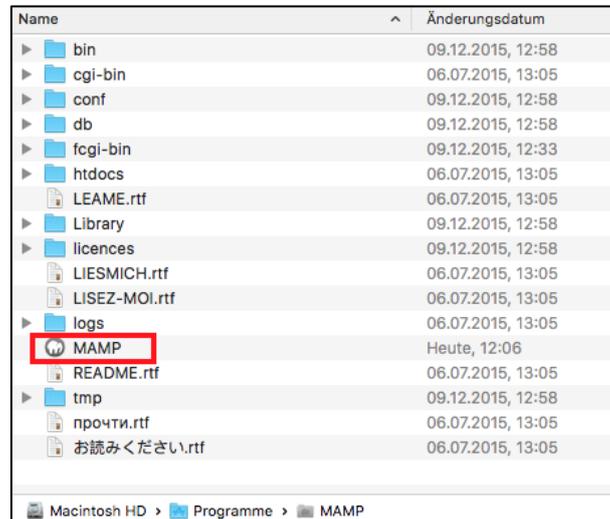


Abb. 19: Suche von MAMP

- n) Es öffnet sich eine Warnung, klicken Sie auf den „Starte MAMP“-Button (nicht „Starte MAMP PRO“) (Abb. 20).



Abb. 20: Warnmeldung beim Starten von MAMP

o) Es öffnet sich MAMP: Klicken Sie auf „Server starten“ (Abb.21).



Abb. 21: Server starten in MAMP

p) Erlauben Sie den Servern eine Netzwerkverbindung, indem Sie auf den „Erlauben“-Button klicken (Abb. 22).

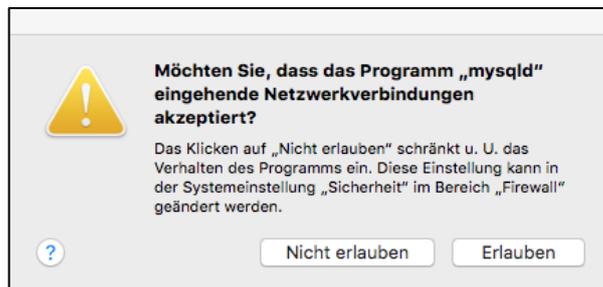


Abb. 22: Warnmeldung beim Starten der Server in MAMP

Wichtig: Sie dürfen MAMP nun **nicht** schließen!

Sie haben nun MAMP erfolgreich installiert.

3.4 Das XAMPP Control Panel

Nach der Installation von XAMPP kann das sogenannte XAMPP Control Panel geöffnet werden (Abb. 23). Das Control Panel ist die "Schaltzentrale" von XAMPP. Module können hier gestartet, konfiguriert und beendet werden.

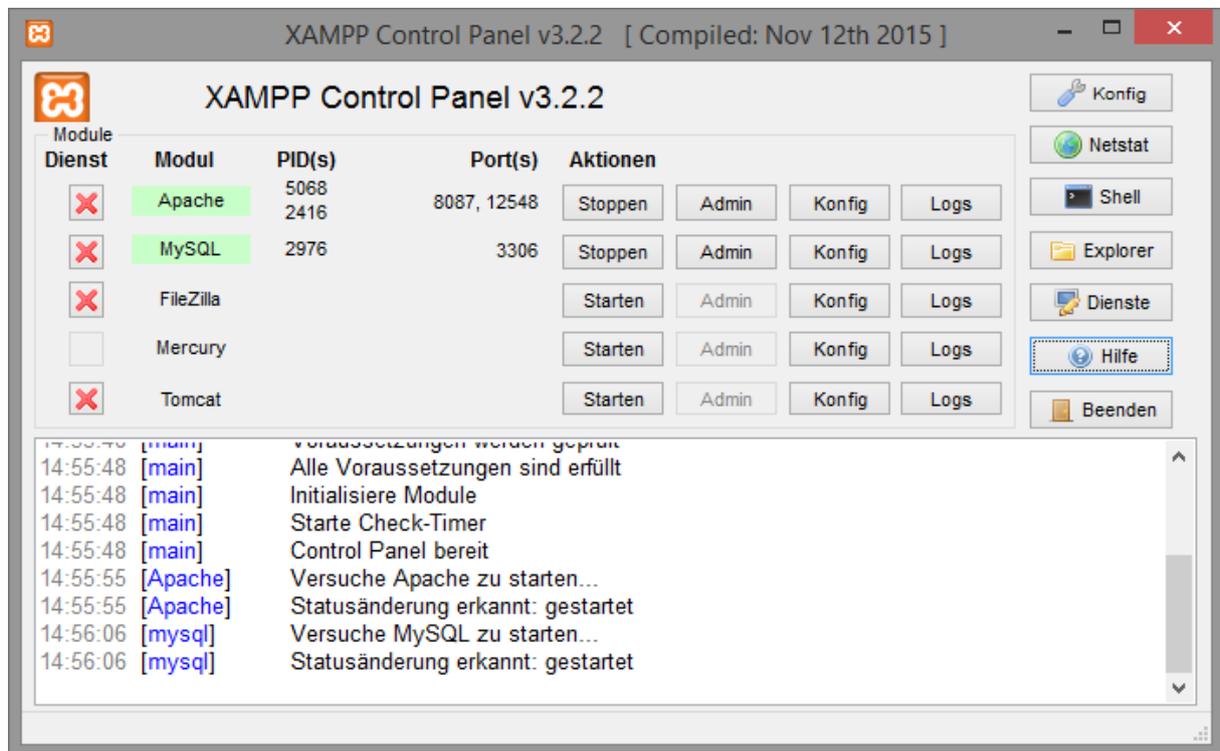


Abb. 23: XAMPP Control Panel

- **Aktionen:** Mit dem XAMPP Control Panel können die Module gestartet, gestoppt, überwacht und konfiguriert werden. Die Server Apache und MySQL wurden hier bereits gestartet.
- **Port(s):** Ein Port wird verwendet, um die Module zu nutzen. Jedes Modul sollte einen eindeutigen eigenen Port haben. So belegt jede Anwendung wie z. B. Skype einen eigenen Port zur Kommunikation. Wenn ein Modul gestartet ist, sind hier die vom Modul verwendeten Ports zu sehen.
- **Dienst:** Das rote X zeigt an, dass die Module nicht automatisch gemeinsam mit dem Windows-Betriebssystem gestartet werden. Diese Einstellung sollten Sie beibehalten.
- **Konsole:** Unten im XAMPP Control Panel befindet sich die Konsole. Dort werden Status- und Fehlermeldungen angezeigt. In den letzten vier Zeilen ist z. B. zu sehen, dass MySQL und Apache gestartet wurden.
- **XAMPP Einstellungen:** Weitere Einstellungsmöglichkeiten und Hilfestellungen zur Verwendung von XAMPP werden auf der rechten Seite des Control Panels zur Verfü-

gung gestellt. Durch Klick auf den "Beenden"-Button kann das Control Panel beendet werden, die gestarteten Server werden dadurch jedoch nicht gestoppt.

- **Modul:** Mit XAMPP wurden fünf Server heruntergeladen, die über das Control Panel verwaltet werden können: Apache, MySQL, FileZilla, Mercury und Tomcat. Die gestarteten Module werden grün angezeigt. Hier wurden Apache und MySQL bereits gestartet.

3.5 Ports anpassen

Die Installation von XAMPP haben Sie abgeschlossen. Als Windows-Nutzer müssen Sie den Port im XAMPP Control Panel für den Web-Server Apache anpassen, um Fehlermeldungen beim Start von Apache zu vermeiden. Führen Sie dazu die folgenden Schritte durch:

1. Öffnen Sie das XAMPP Control Panel
2. Wählen Sie rechts neben dem Modul "Apache" den "Konfig"-Button und wählen Sie nun "Apache (httpd.conf)" (Abb. 24).

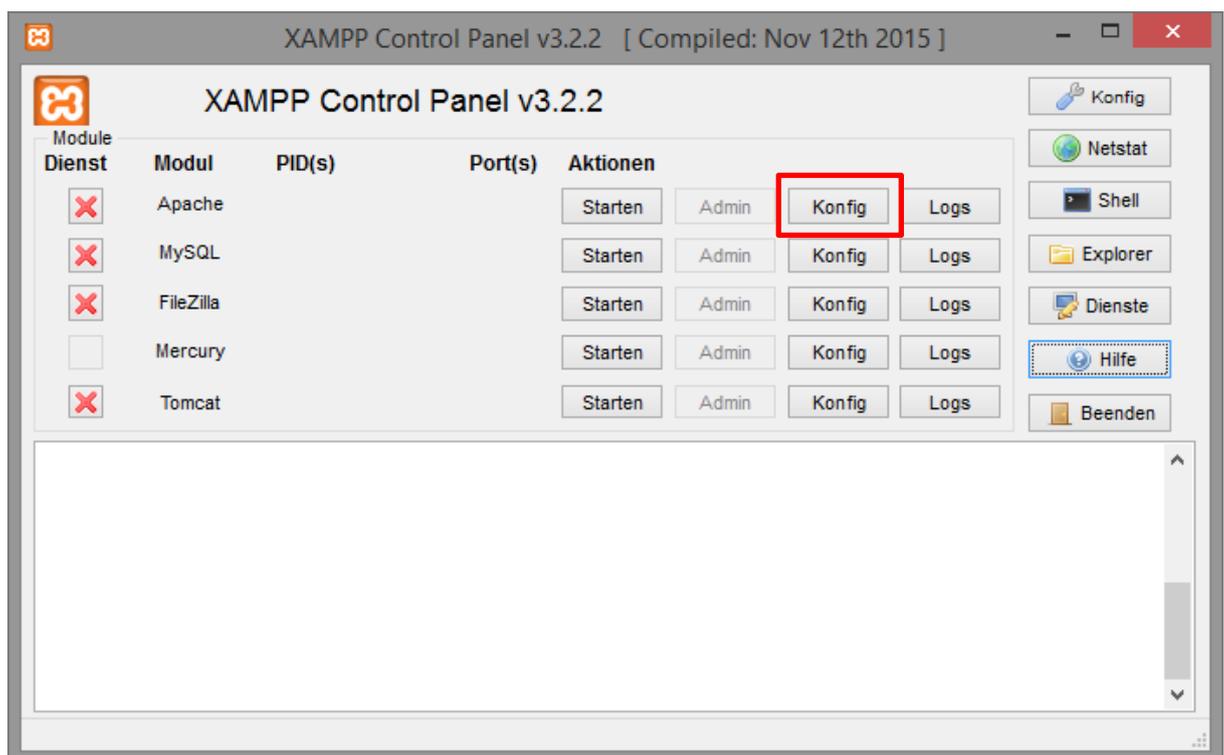


Abb. 24: Konfig-Button im XAMPP Control Panel

3. Die Datei öffnet sich nun im Editor. Suchen Sie im Quelltext nach der Portnummer 80 und ersetzen Sie sie durch die Portnummer 8087. Speichern Sie anschließend die Änderungen im Editor.

Eine Fehlermeldung wie diese kann durch eine Anpassung des Ports behoben werden (Abb. 25).



Abb. 25: Fehlermeldung im XAMPP Control Panel

3.6 Localhost aufrufen

Nun ist die Einrichtung aller Werkzeuge fertig und die Arbeit mit MySQL kann starten. Um mit MySQL zu arbeiten, müssen der Web-Server Apache und der MySQL-Server über das XAMPP Control Panel gestartet werden.

Sobald die beiden Server gestartet wurden, können Sie überprüfen, ob die Installation von XAMPP und die Konfiguration der Server erfolgreich waren. Dazu geben Sie als Windows-Nutzer "http://localhost:8087" und als OS-Nutzer "http://localhost:8888" in die Adressleiste ihres Web-Browsers ein. Sie werden direkt zur Startseite von XAMPP weitergeleitet. Dort haben Sie Zugriff auf das MySQL-Konfigurationstool phpMyAdmin, mit dem wir unsere MySQL-Datenbank erstellen und mit Daten füllen können.

Durch Öffnen des Links sollte sich ihr Localhost öffnen (Abb. 26). Dieser verweist auf ihren lokalen Rechner. Mit dem Doppelpunkt wird der lokale Rechner vom Apache-Port getrennt. Wenn sich eine leere Seite geöffnet hat, überprüfen Sie, ob die Server gestartet sind und ob Sie einen anderen Port als 8087 (OS X: 8888) bei der Installation eingestellt haben. Dieser Port muss statt "8087" bzw. "8888" angegeben werden.

Durch Klick auf die Schaltfläche "phpMyAdmin" gelangen Sie zum MySQL-Konfigurationstool phpMyAdmin.

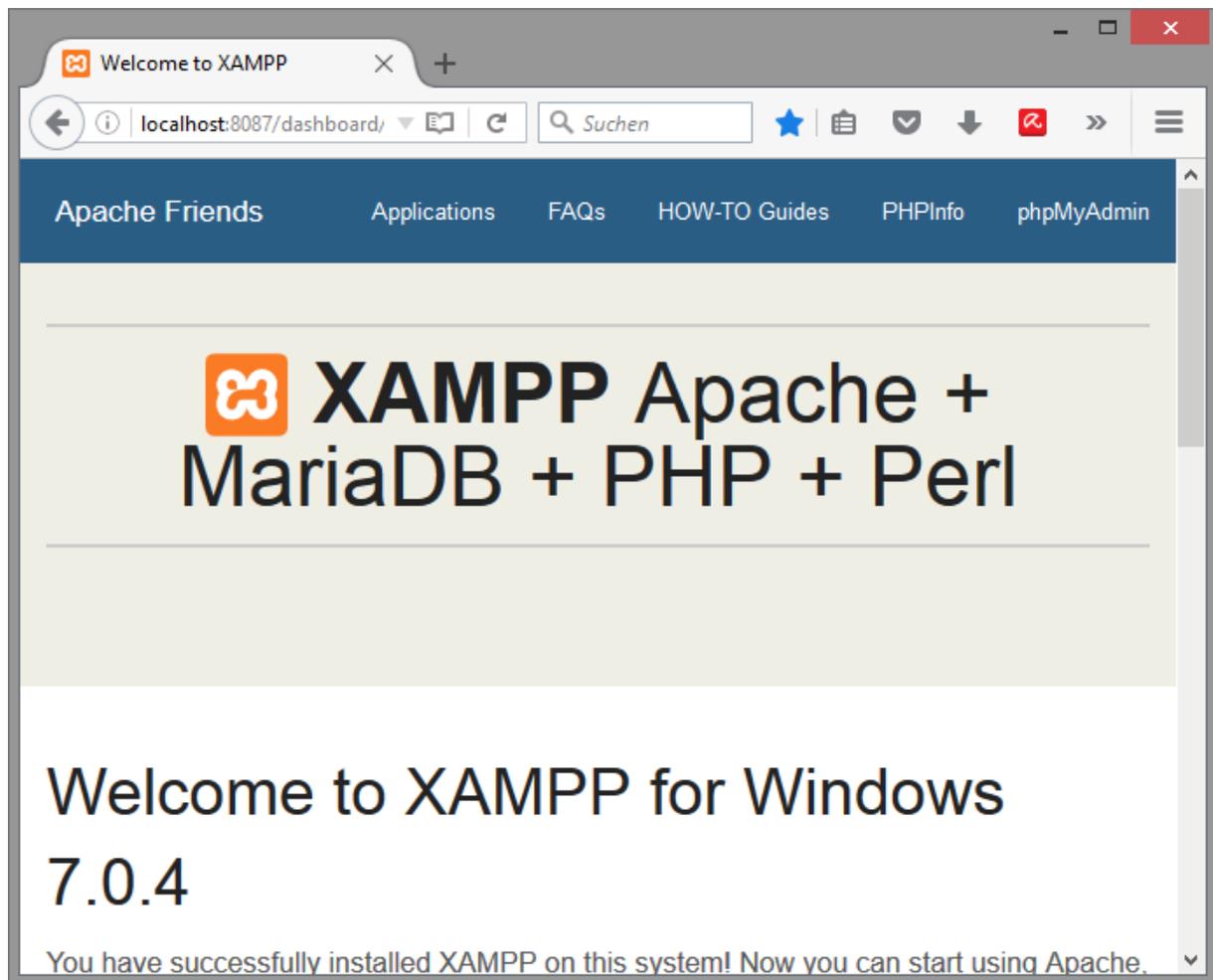


Abb. 26: Localhost

3.7 phpMyAdmin

phpMyAdmin ist eine graphische Benutzeroberfläche für das Datenbankmanagementsystem MySQL (Abb. 27). Es ist in der Programmiersprache PHP geschrieben und kann über einen Web-Browser bedient werden.

Mit phpMyAdmin ist es unter anderem möglich

- Datenbanken anzulegen und zu löschen,
- Tabellen anzulegen und zu löschen,
- Tabellen miteinander zu verknüpfen,
- Daten in Tabellen einzufügen und zu löschen,
- Daten mit SQL abzufragen und Benutzer zu verwalten.

In den nächsten WBT lernen Sie, wie dies funktioniert.

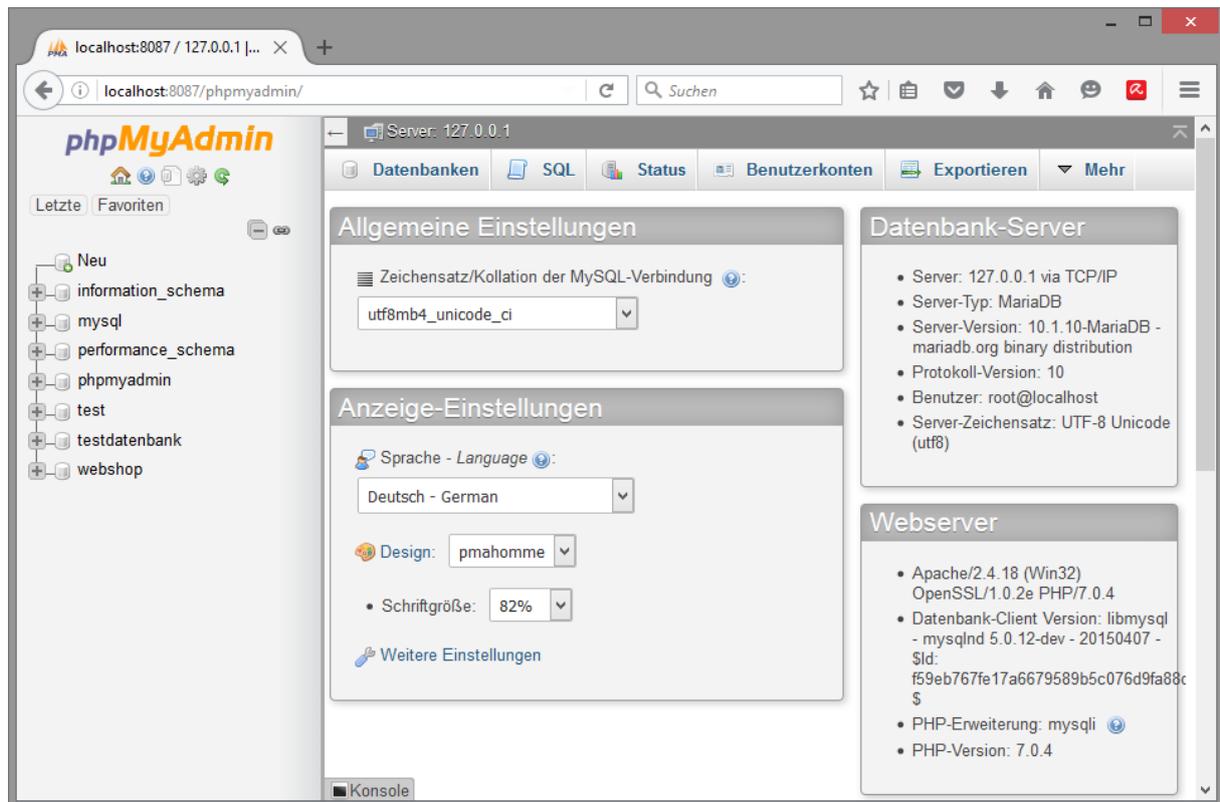


Abb. 27: Graphische Benutzeroberfläche phpMyAdmin

Die Bedienoberfläche von phpMyAdmin ist in zwei Bereiche aufgeteilt. Im linken Bereich des Fensters wird die Navigation für Datenbanken und Tabellen angezeigt. Im rechten Fensterbereich ist der eigentliche Arbeitsbereich zu sehen, in dem Daten bearbeitet und ausgegeben werden können.

3.8 Übung: Erstellen einer Datenbank

Sie haben nun bereits einiges über MySQL gelernt. Um einen Eindruck davon zu bekommen, wie einfach der Umgang mit XAMPP und phpMyAdmin ist, werden wir nun die Datenbank für die Lemonline AG mit phpMyAdmin erstellen.

Starten Sie ihren Apache-Server und ihren MySQL-Server über das XAMPP Control Panel. Rufen Sie mit "http://localhost:8087" (OS X: 8888) die XAMPP-Startseite auf und erstellen Sie die MySQL-Datenbank "webshop" mit phpMyAdmin. Im Video sehen Sie, wie Sie dabei vorgehen müssen.

Im WBT können Sie sich an dieser Stelle ein Video anschauen, was Ihnen die Vorgehensweise zur Erstellung einer Datenbank verdeutlicht.

3.9 Zusammenfassung und Ausblick

In diesem WBT haben wir die ersten Schritte mit MySQL gemacht. Sie haben alle notwendigen Werkzeuge auf Ihrem PC installiert und bereits eine Datenbank für den Webshop unseres Kunden erstellt.

Im nächsten WBT werden wir zunächst planen, wie die Datenbank für den Webshop später aussehen soll. Dazu werden wir uns überlegen, welche Anforderungen an die Datenbank des Webshops gestellt werden und wie wir diese Anforderungen mit MySQL sinnvoll umsetzen können.

4 Planung der Webshop-Datenbank

4.1 Einleitung

Im letzten WBT haben wir einige Werkzeuge auf Ihrem PC eingerichtet, um gemeinsam die Webshop-Datenbank für die Lemonline AG zu erstellen.

In diesem WBT werden wir planen, wie die Datenbank für den Webshop aussehen soll. Dazu werden wir uns überlegen, welche Anforderungen an die Datenbank des Webshops gestellt werden und wie wir diese Anforderungen mit MySQL sinnvoll umsetzen können.

Für gewöhnlich plane und erstelle ich eine Datenbank in vier Schritten. Diese vier Schritte der Datenmodellierung erkläre ich Ihnen zunächst, bevor wir damit starten die Schritte für die Datenbank der Lemonline AG nacheinander abzuarbeiten...

4.2 Übersicht: Schritte der Datenmodellierung

Die Datenbankstruktur für den Webshop der Lemonline AG wird in vier Schritten entwickelt. In diesen vier Schritten wird die Datenbankstruktur immer weiter verfeinert, um den Webshop der Lemonline AG möglichst realistisch darzustellen.

Die einzelnen Schritte werden auf den folgenden WBT-Seiten näher erklärt.

- **Schritt 1 Datenanalyse:** Im ersten Schritt der Datenmodellierung werden zunächst verbal die Tätigkeiten und Daten beschrieben, die mit der Webshop-Datenbank abgebildet werden sollen. Die Lemonline AG legt in diesem Schritt fest, welche Tätigkeiten durch die neue Datenbank unterstützt werden.
- **Schritt 2 Konzeptuelles Datenschema:** Im zweiten Schritt der Datenmodellierung wird aus der unstrukturierten verbalen Beschreibung ein vollständiges, exaktes ER-Modell erzeugt. Das ER-Modell strukturiert die benötigten Daten und stellt die Beziehungen zwischen den Daten her.
- **Schritt 3 Relationales Datenschema:** Im dritten Schritt der Datenmodellierung wird das ER-Modell umgewandelt in eine Datenbankstruktur mit Tabellen und Attributen. Das ER-Modell wird genutzt, um die Datenbank mit dem Datenbankmanagementsystem zu erstellen.
- **Schritt 4 Physisches Datenschema:** Im vierten Schritt der Datenmodellierung werden die Dateien erzeugt, die die Datenbank als Ganzes ausmachen. Die Webshop-Datenbank der Lemonline AG wird dabei mit dem DBMS erstellt.

4.3 Schritt 1: Datenanalyse I

Herr Leo Lemon, einer der Geschäftsführer der Lemonline AG, hat mir in einer E-Mail geschildert, wie die Geschäftsprozesse aussehen, die von der Datenbank unterstützt werden sollen. Jetzt haben wir ein Bild davon, welche Anforderungen die Lemonline AG an ihre Datenbank stellt.

E-Mail von Herr Leo Lemon:

Hallo Herr Weber,

ich freue mich, dass Sie unseren Auftrag angenommen haben. Unsere Webshop-Datenbank soll in unsere Web Site eingebunden werden, um einen Teil unserer Smartphones und Tablets über den Webshop zu vertreiben. Wir stellen uns das wie folgt vor:

Unser Kunde stöbert in unserem Webshop nach Produkten. Wenn ihm etwas gefällt, registriert er sich mit seinen Kontaktdaten im Webshop und sammelt seine Wunschprodukte in einem Einkaufswagen. Am Ende bestellt er alle Produkte gleichzeitig.

Wir sehen in unserer Datenbank, wie viele Bestellungen von wem aufgegeben wurden und welche Bestellungen noch offen sind. Falls Produkte ausverkauft sind, möchten wir direkt über das Produkt zu dessen Lieferanten und seinen Kontaktdaten gelangen, um das Produkt nachzubestellen.

Meine Mitarbeiter haben im letzten Meeting gemeinsam die Informationen strukturiert, die wir später in der Webshop-Datenbank benötigen. Ich habe Ihnen die Excel-Datei mit den Ergebnissen meiner Mitarbeiter an die E-Mail angehängt. Für Rückfragen stehe ich Ihnen jederzeit zur Verfügung.

Mit freundlichen Grüßen

Leo Lemon

4.4 Schritt 1: Datenanalyse II

Leo Lemon hat seiner E-Mail zusätzlich eine Excel-Datei angehängt, in der die notwendigen Informationen strukturiert dargestellt sind (Abb. 28). Diese Datei werden wir nun schrittweise in eine Datenbank umwandeln.

Mit diesem Wissen können wir in einem nächsten Schritt ein konzeptuelles Datenschema erstellen.

	A	B	C	D	E	F	G	H
1								
2		Kunden		Produkte		Bestellungen		
3		Vorname		Name		Bestell-Nr		
4		Nachname		Preis		Datum der Bestellung		
5		Adresse		Lieferant		Status		
6		E-Mail		Adresse Lieferant		Kunde		
7				E-Mail Lieferant		Bestellte Produkte		
8						Bestellmenge		

Abb. 28: Informationen zur Datenanalyse

4.5 Schritt 2: Konzeptuelles Datenschema

Im zweiten Schritt der Datenmodellierung wird aus der Beschreibung von Herr Lemon ein ER-Modell erzeugt. Mit dem ER-Modell kann z. B. dargestellt werden, dass Kunden im Webshop Produkte bestellen. Zudem kann im ER-Modell z. B. verdeutlicht werden, dass zu jedem Produkt der Name, der Preis und der Lieferant hinterlegt sind.

Innerhalb eines ER-Modells werden Entitäten, deren Beziehungen sowie Attribute dargestellt. Auf den nächsten WBT-Seiten erfahren Sie, was das ist.

4.6 ER-Modell: Entitäten und Entitätsmengen

Die Webshop-Kunden "Herr Müller" und "Frau Kleist" stellen zwei Entitäten des Webshops dar. Entitäten zeichnen sich dadurch aus, dass sie individuelle, identifizierbare Dinge oder Personen sind.

Die beiden Webshop-Kunden können in der Entitätsmenge "Kunde" zusammengefasst werden (Abb. 29).

Entitätsmengen	Entitäten
Kunde	Herr Müller, Frau Kleist...
Produkt	Tablet 342, Smartphone X3...
Bestellung	Nr. 291, Nr. 343, Nr. 741...

Abb. 29: Zusammenhang zwischen Entitätsmengen und Entitäten

Entitätsmengen werden im ER-Modell durch Rechtecke dargestellt (Abb. 30).

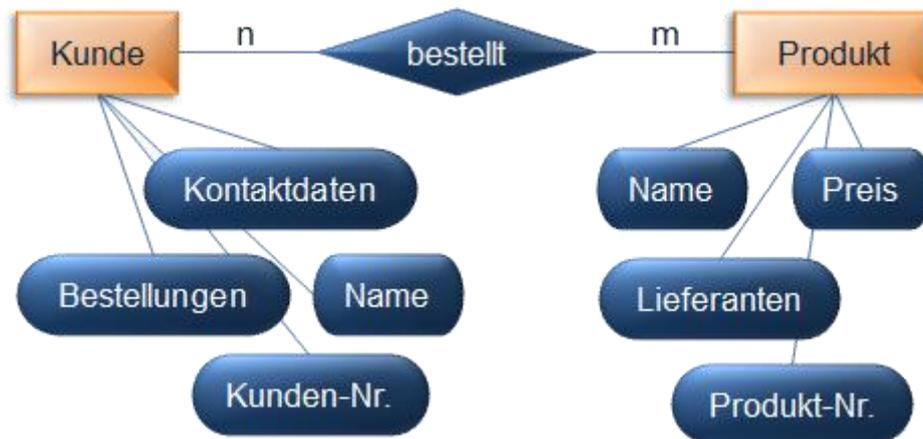


Abb. 30: Darstellung von Entitätsmengen im ER-Modell

4.7 ER-Modell: Attribute

Den einzelnen Entitätsmengen können mehrere Eigenschaften, sogenannte Attribute, zugeordnet werden. So sollen im Webshop später zu allen Kunden Werte für die Attribute Kundennummer, Name, Kontaktdaten und Bestellungen erfasst werden. Attribute werden im ER-Modell durch Ovale dargestellt (Abb. 31).

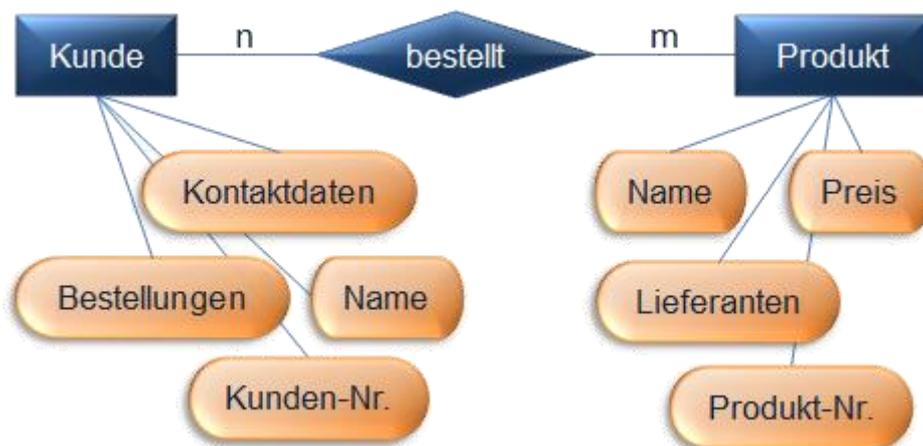


Abb. 31: Darstellung von Attributen im ER-Modell

Die Kundennummer eines Kunden wird dafür erzeugt, einen Kunden eindeutig zu identifizieren. Ein Attribut, das eine Entität eindeutig identifiziert, wird "Primärschlüssel" genannt. Der Primärschlüssel der Entitätsmenge "Produkt" ist in diesem Beispiel die Produktnummer (Abb. 32).

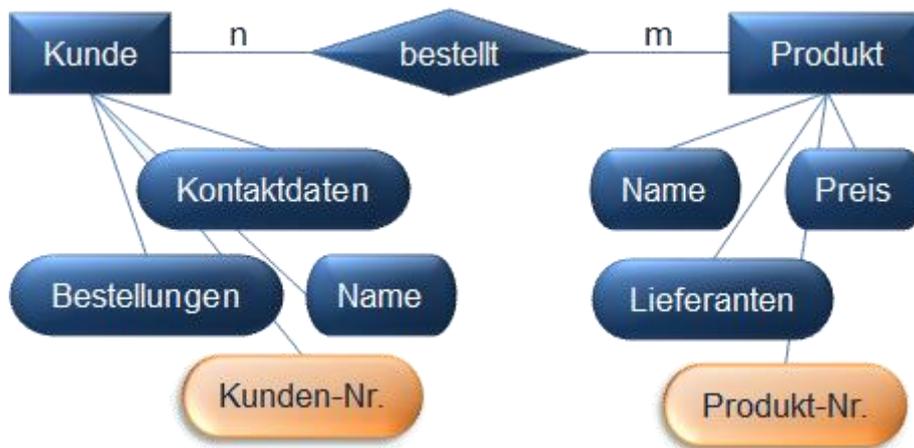


Abb. 32: Primärschlüssel

4.8 ER-Modell: Relationen

In der Webshop-Datenbank sind Daten wie Kunden und Produktdaten miteinander verbunden. Diese sogenannten Relationen werden im ER-Modell durch Rauten dargestellt und durch Kardinalitäten näher beschrieben (Abb. 33). Eine Relation muss dabei immer aus beiden Richtungen gelesen werden.

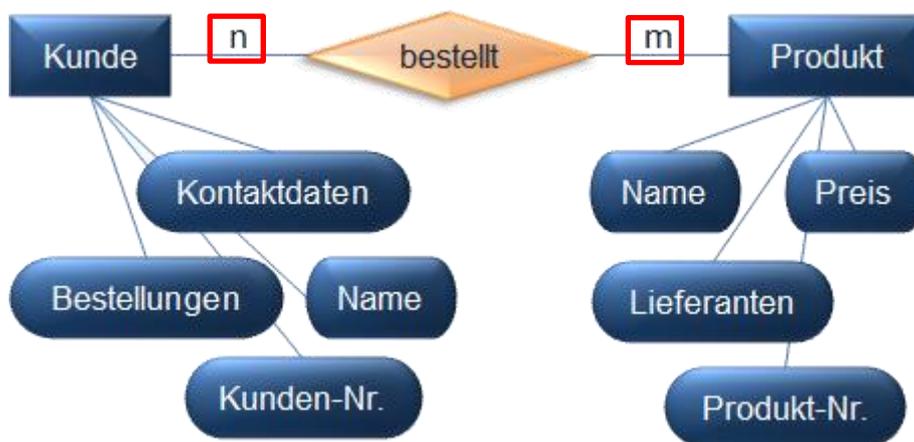


Abb. 33: Darstellung von Relationen und Kardinalitäten im ER-Modell

Für die dargestellte Relation bedeutet dies beispielsweise: Ein Kunde kann mehrere Produkte bestellen und ein Produkt kann von mehreren Kunden bestellt werden.

4.9 ER-Modell: Relationen II

Zwei Entitätsmengen können über die folgenden Kardinalitäten miteinander verknüpft werden:

- **1:1-Beziehung:** Ein Mann heiratet eine Frau. Eine Frau heiratet einen Mann (Abb.34).



Abb. 34: 1:1-Beziehung

- **1:n-Beziehung:** Die Bestellung eines Kunden kann mehrere Bestellpositionen umfassen. Eine Bestellposition gehört zu genau einer Bestellung (Abb. 35).



Abb. 35: 1:n-Beziehung

- **n:m-Beziehung:** Ein Kunde kann mehrere Produkte bestellen (a). Ein Produkt kann von mehreren Kunden bestellt werden (Abb. 36).



Abb. 36: n:m-Beziehung

4.10 Schritt 2: ER-Modell des Webshops

Sie verstehen nun, was die Symbole des ER-Modells bedeuten. Die Erstellung eines ER-Modells ist eine komplexe Angelegenheit. Daher hat einer meiner Mitarbeiter bereits das ER-Modell für die Webshop-Datenbank in Rücksprache mit der Lemonline AG erstellt (Abb. 37). Dieses ER-Modell müssen wir nun in einem nächsten Schritt in ein relationales Datenschema übertragen.

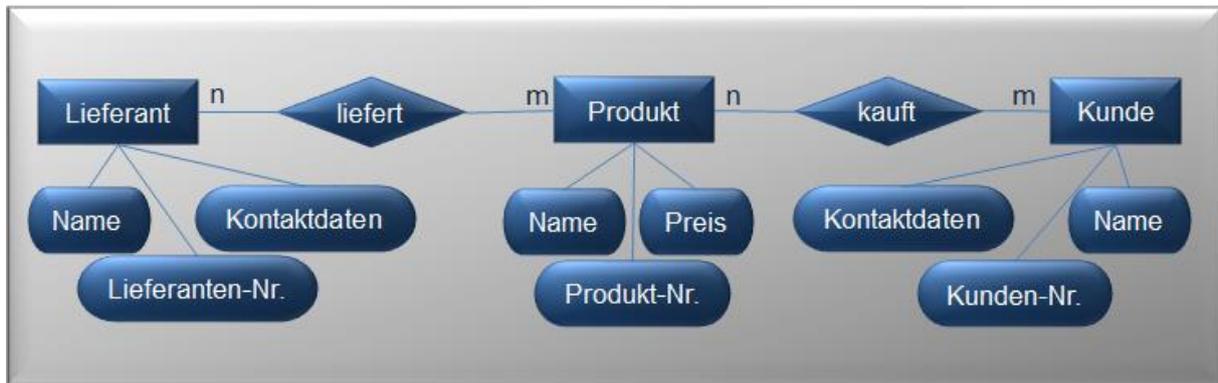
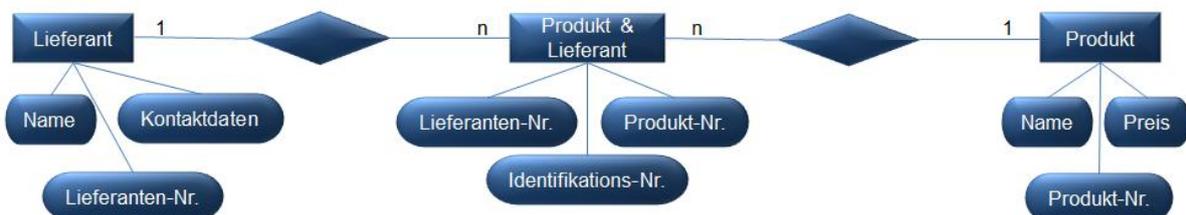


Abb. 37: ER-Modell des Webshops

4.11 Schritt 3: Relationales Datenschema I

Im dritten Schritt der Datenmodellierung wird das ER-Modell der Webshop-Datenbank in ein Relationenmodell überführt. Da viele relationale Datenbanken $n:m$ -Beziehungen nicht umsetzen können, müssen in diesem Schritt zunächst alle $n:m$ -Beziehungen in $1:n$ -Beziehungen aufgelöst werden.

Zur Auflösung der $n:m$ -Beziehung zwischen Lieferanten und Kunden wird so z. B. eine zusätzliche Entität erstellt, die die Primärschlüssel beider Tabellen enthält (Abb. 38).

Abb. 38: Auflösung der $n:m$ -Beziehung

4.12 Schritt 3: Relationales Datenschema II

Bei der Umwandlung des ER-Modells in ein Relationenmodell müssen neben der Auflösung der $n:m$ -Beziehungen die folgenden Kriterien beachtet werden:

- keine Mehrfachspeicherung von Daten
- Widerspruchsfreiheit der Daten
- sinnvolle Abbildung der Realität
- Aufteilung der Attribute in unteilbare Pakete (z. B. werden statt dem Attribut "Adresse" die Attribute "Straße", "PLZ" und "Ort" gebildet.)

Im relationalen Datenschema werden die Entitäten und deren Attribute in Tabellen dargestellt. Außerdem werden die Beziehungen zwischen den Tabellen über Primär- und Fremdschlüssel hergestellt (Abb. 39).

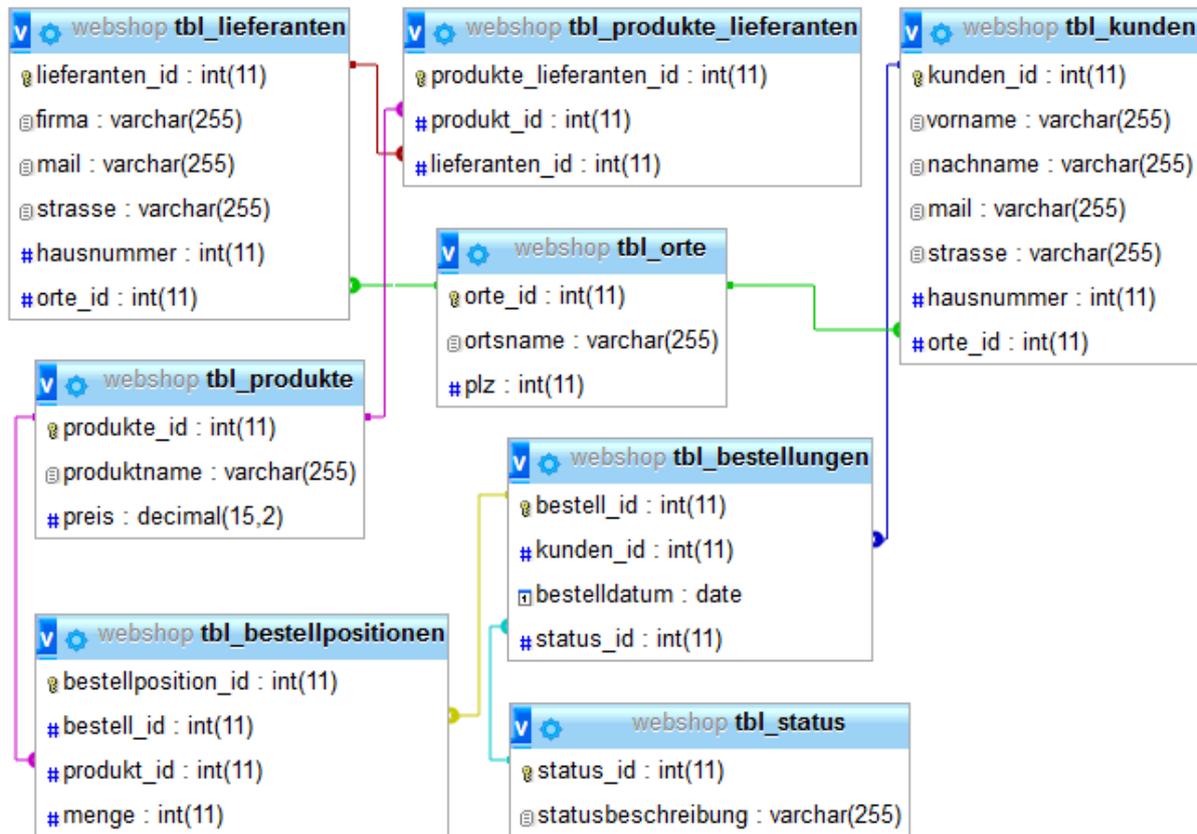


Abb. 39: Relationales Datenschema des Webshops

Um einen Kunden eindeutig zu identifizieren, wird eine Kundennummer verwendet. Daher ist in der Tabelle "tbl_kunden" die "kunden_id" (id = identifizier) der Primärschlüssel.

Bei Fremdschlüsseln handelt es sich um Attribute, die in einer Tabelle vorkommen und gleichzeitig in einer anderen Tabelle Primärschlüssel sind. In der Tabelle "tbl_bestellungen" ist das Attribut "kunden_id" daher ein Fremdschlüssel.

4.13 Zusammenfassung und Ausblick

In diesem WBT haben wir geplant, wie die Datenbank für den Webshop der Lemonline AG aussehen soll. Uns ist jetzt klar, welche Anforderungen die Datenbank erfüllen muss. Die ersten drei Schritte der Datenmodellierung haben wir nun abgeschlossen.

Im nächsten WBT werden wir im Rahmen des vierten Schritts der Datenmodellierung damit starten, unser relationales Datenschema in der MySQL-Datenbank umzusetzen. Dazu werden wir die Tabellen für den Webshop mit phpMyAdmin erstellen und über Primär- und Fremdschlüssel miteinander verknüpfen.

5 Erstellen der Webshop-Datenbank

5.1 Einleitung

Im letzten WBT haben wir die Webshop-Datenbank für die Lemonline AG in drei Schritten geplant.

In diesem WBT werden wir den letzten Schritt der Datenmodellierung, das physische Datenschema, ausführen und die Webshop-Datenbank der Lemonline AG erstellen. Dazu werden wir uns zunächst noch einmal anschauen, wie in "WBT 02 - Erste Schritte mit MySQL" die Webshop-Datenbank mit phpMyAdmin erstellt wurde.

Danach werden wir das relationale Datenschema, das wir im letzten WBT gemeinsam erstellt haben, mit phpMyAdmin umsetzen. Wir werden uns dafür ansehen, wie die Tabellen für den Webshop der Lemonline AG erstellt und miteinander verbunden werden.

5.2 Wiederholung: Erstellen einer Datenbank

In "WBT 02 - Erste Schritte mit MySQL" haben Sie bereits die Datenbank für die Lemonline AG mit phpMyAdmin erzeugt. Im Rahmen der Übung haben Sie dabei die folgenden Schritte ausgeführt:

Starten Sie ihren Apache-Server und ihren MySQL-Server über das XAMPP Control Panel. Rufen Sie mit "http://localhost:8087" (OS X: 8888) die XAMPP-Startseite auf und erstellen Sie die MySQL-Datenbank "webshop" mit phpMyAdmin. Im Video sehen Sie, wie Sie dabei vorgehen müssen.

Im WBT können Sie sich an dieser Stelle ein Video anschauen, was Ihnen die Vorgehensweise zur Erstellung einer Datenbank verdeutlicht.

5.3 Wiederholung: Schritte der Datenmodellierung

Die Datenbankstruktur für den Webshop der Lemonline AG wird in vier Schritten entwickelt. In diesen vier Schritten wird die Datenbankstruktur immer weiter verfeinert, um den Webshop der Lemonline AG möglichst realistisch darzustellen.

Die ersten drei Schritte wurden in "WBT 03 - Planung der Webshop-Datenbank" behandelt. In diesem WBT wird Schritt 4 thematisiert.

- **Schritt 1 Datenanalyse:** Im ersten Schritt der Datenmodellierung werden zunächst verbal die Tätigkeiten und Daten beschrieben, die mit der Webshop-Datenbank abgebildet werden sollen. Die Lemonline AG legt in diesem Schritt fest, welche Tätigkeiten durch die neue Datenbank unterstützt werden.

- **Schritt 2 Konzeptuelles Datenschema:** Im zweiten Schritt der Datenmodellierung wird aus der unstrukturierten verbalen Beschreibung ein vollständiges, exaktes ER-Modell erzeugt. Das ER-Modell strukturiert die benötigten Daten und stellt die Beziehungen zwischen den Daten her.
- **Schritt 3 Relationales Datenschema:** Im dritten Schritt der Datenmodellierung wird das ER-Modell umgewandelt in eine Datenbankstruktur mit Tabellen und Attributen. Das ER-Modell wird genutzt, um die Datenbank mit dem Datenbankmanagementsystem zu erstellen.
- **Schritt 4 Physisches Datenschema:** Im vierten Schritt der Datenmodellierung werden die Dateien erzeugt, die die Datenbank als Ganzes ausmachen. Die Webshop-Datenbank der Lemonline AG wird dabei mit dem DBMS erstellt.

5.4 Schritt 4: Physisches Datenschema

Im letzten Schritt der Datenmodellierung wird das relationale Datenschema umgesetzt. In diesem Schritt werden mit dem Datenbankmanagementsystem die Dateien erzeugt, welche die Datenbank physisch auf einem Speichermedium abbilden (Abb. 40).

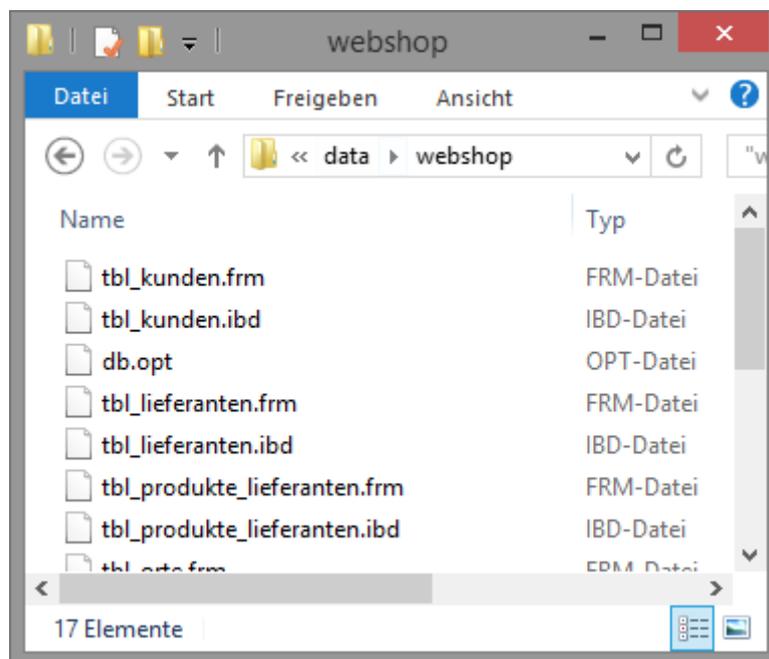


Abb. 40: Dateien der Webshop-Datenbank

Zunächst werden die einzelnen Tabellen erstellt und anschließend zueinander in Beziehung gesetzt.

Um etwas bestellen zu können, müssen sich die Kunden des Webshops auf der Web Site der Lemonline AG registrieren. Diese Kundendaten werden in der Webshop-Datenbank in der Tabelle "tbl_kunden" gespeichert.

Damit diese Kundenregistrierung möglich wird, muss nun zunächst die Tabelle "tbl_kunden" mit den bereits festgelegten Attributen in der Webshop-Datenbank angelegt werden.

5.5 Erzeugen einer Tabelle: Kunden

Damit es künftig möglich ist, Kundendaten in der Webshop-Datenbank der Lemonline AG zu speichern, muss dort zunächst eine Kunden-Tabelle angelegt werden (Abb. 41). Die Kundennummer (kunden_id) wird dabei als Primärschlüssel der Tabelle eingetragen.



Abb. 41: Tabellenstruktur der Kunden-Tabelle

Die Kunden-Tabelle wird dabei mit der Hilfe von phpMyAdmin erstellt. Im Video sehen Sie, wie Sie dabei vorgehen müssen. Auf der folgenden WBT-Seite werde ich Ihnen die Einstellungen erklären, die bei Erstellung der Kunden-Tabelle im Video vorgenommen wurden.

Im WBT können Sie sich an dieser Stelle ein Video anschauen, was Ihnen die Vorgehensweise zur Erstellung der Kunden-Tabelle verdeutlicht.

5.6 Einstellungen einer Tabelle: Kunden

Beim Erzeugen der Kunden-Tabelle muss für jedes Feld (Attribut) zunächst ein Datentyp eingestellt werden. Außerdem wird zu jedem Feld eine maximale Feldlänge angegeben. Abschließend kann ein Feld der Tabelle als Primärschlüssel eingestellt werden. Damit in der Datenbanktabelle alle internationalen Zeichen dargestellt werden können, wird am Ende die Kollation "utf-8_general_ci" eingestellt (Abb. 42).

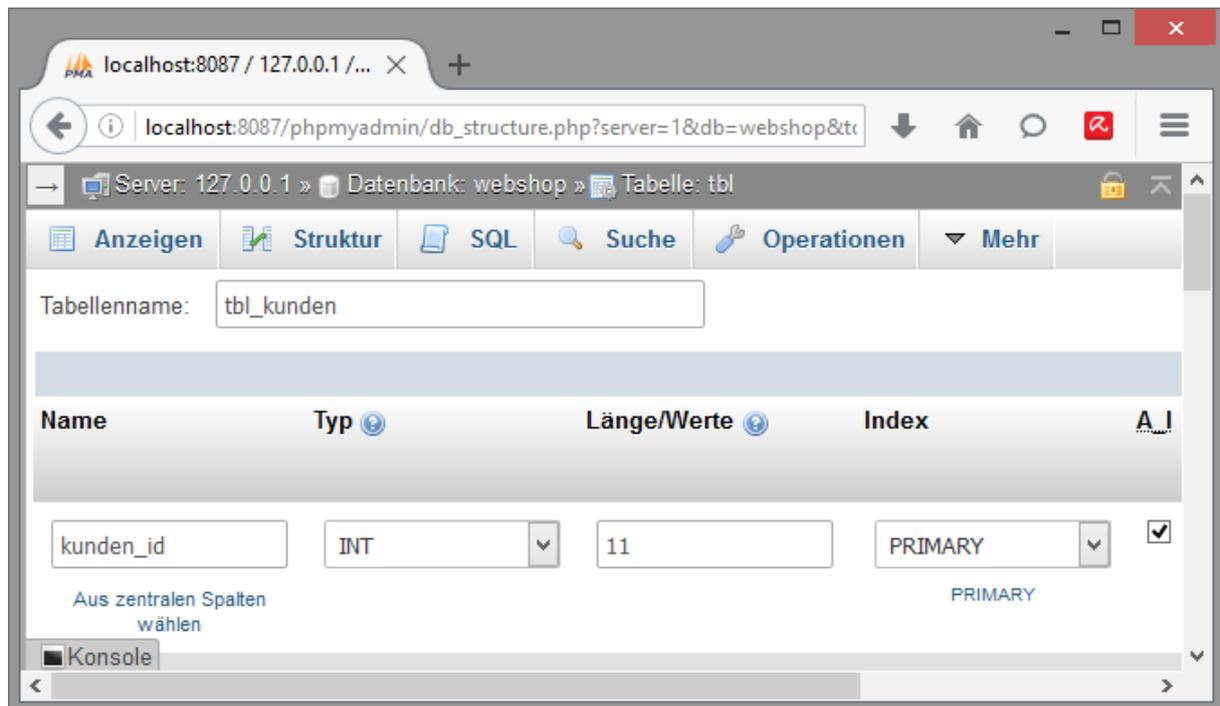


Abb. 42: Einstellungen in der Kunden-Tabelle

Häufige Datentypen sind:

- **VARCHAR:** Der Datentyp VARCHAR legt fest, dass nur Buchstaben und Zeichen in diesem Feld gespeichert werden.
- **INT:** Der Datentyp INT legt fest, dass nur ganze Zahlen in diesem Feld gespeichert werden.
- **DECIMAL:** Der Datentyp DECIMAL legt fest, dass nur Dezimalzahlen in diesem Feld gespeichert werden.

Weitere Datentypen und deren Erklärungen finden Sie in der MySQL-Dokumentation.

Primärschlüssel: Die Kundennummer (kunden_id) identifiziert einen Kunden eindeutig und soll durch das System automatisch erstellt werden. Dazu wird die Einstellung "AUTO_INCREMENT" verwendet. Diese Einstellung führt automatisch zur Abfrage, ob dieses Feld zusätzlich der Primärschlüssel der Tabelle sein soll. Alternativ kann der Primärschlüssel über den Wert "Primary" im Auswahlfeld "Index" eingestellt werden.

5.7 Übung: Tabellen erstellen

Erstellen Sie in der Datenbank "webshop" mit phpMyAdmin die folgenden Tabellen. Stellen Sie dabei für die Attribute die gezeigten Datentypen und Primärschlüssel ein (Abb. 43).



Abb. 43: Tabellen erstellen in der Webshop-Datenbank

5.8 Kunden und Bestellungen verknüpfen

Damit die inhaltliche Beziehung zwischen den Kunden- und Bestelldaten in der Webshop-Datenbank besteht, muss mit phpMyAdmin eine Beziehung zwischen der Kunden- und der Bestellungen-Tabelle erstellt werden.

Die beiden Tabellen werden dazu über den Primär- bzw. Fremdschlüssel "kunden_id" miteinander verbunden (Abb. 44). In phpMyAdmin wird eine Beziehung immer in der Tabelle eingestellt, in der sich der Fremdschlüssel befindet (hier tbl_bestellungen). Im Video sehen Sie, wie Sie dabei vorgehen müssen.



Abb. 44: Kunden und Bestellungen verknüpfen

Im WBT können Sie sich an dieser Stelle ein Video anschauen, was Ihnen die Vorgehensweise zur Erstellung der Kunden-Tabelle verdeutlicht.

5.9 Übung: Tabellen verknüpfen

Verknüpfen Sie mit phpMyAdmin die folgenden Tabellen miteinander wie in der Graphik dargestellt (Abb. 45). In phpMyAdmin müssen Beziehungen immer in der Tabelle erstellt werden, in der sich der Fremdschlüssel befindet.

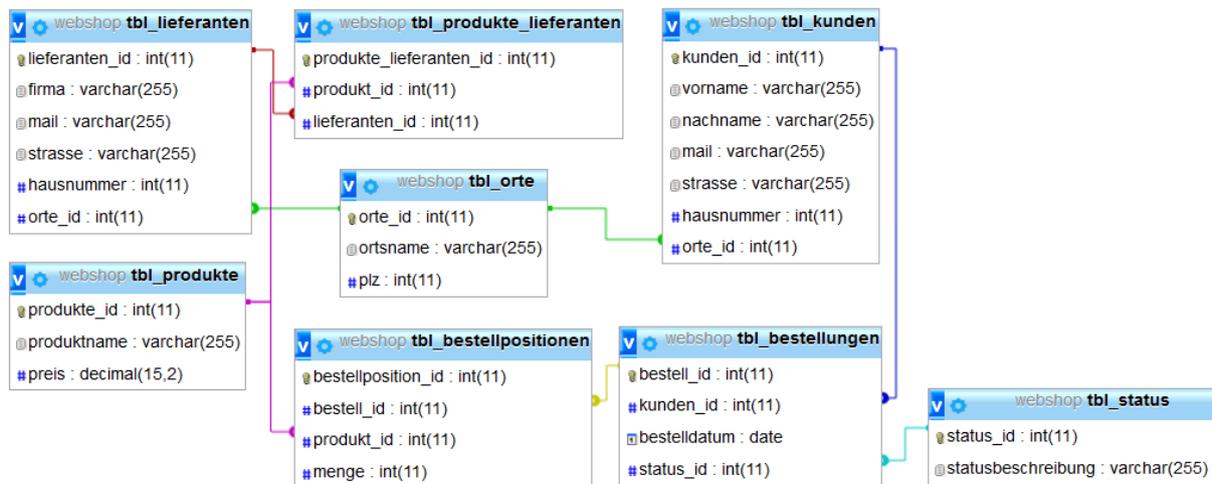


Abb. 45: Tabellen verknüpfen in der Webshop-Datenbank

5.10 phpMyAdmin Designer

Der phpMyAdmin Designer ist eine Funktion von phpMyAdmin (Abb. 46). Mit dem phpMyAdmin können die erstellten Tabellenbeziehungen graphisch dargestellt werden. Über das Hauptmenü des phpMyAdmin Designers ist es möglich, die graphische Darstellung zu verändern und zu speichern.

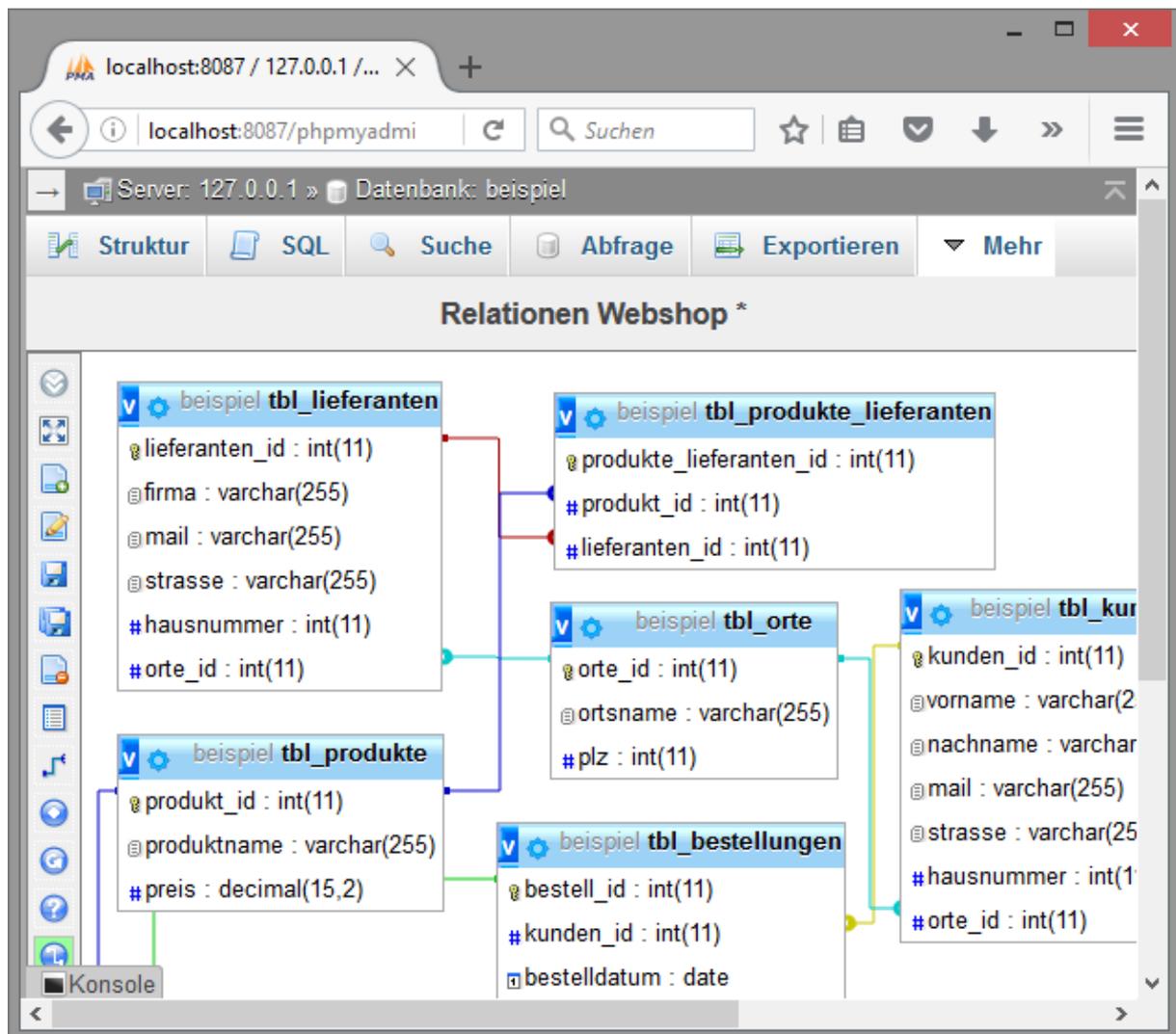


Abb. 46: phpMyAdmin Designer

Nachdem eine Datenbank in phpMyAdmin ausgewählt wurde, kann der phpMyAdmin Designer über einen eigenen Tab im Navigationsmenü aufgerufen werden (Abb. 47).

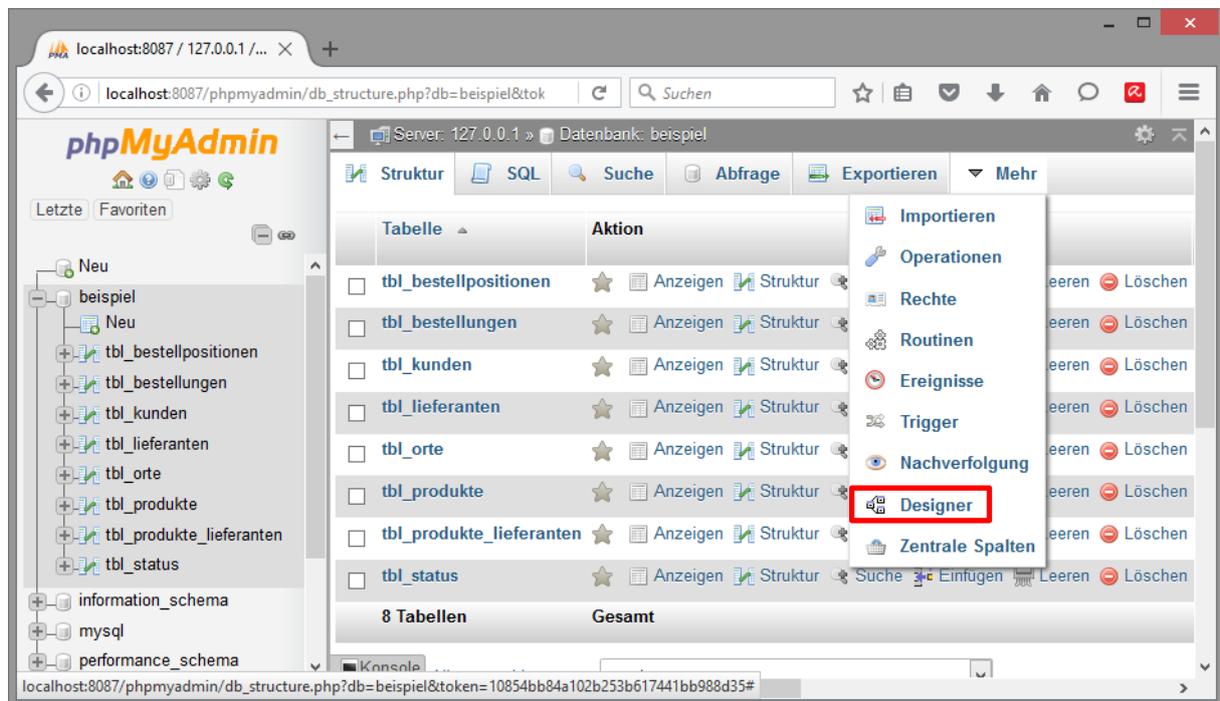


Abb. 47: phpMyAdmin Designer öffnen

5.11 Übung: phpMyAdmin Designer

Sortieren Sie die Datenbank-Tabellen im phpMyAdmin Designer wie unten dargestellt (Abb. 48). Überprüfen Sie, ob Sie die korrekten Verknüpfungen zwischen den Tabellen des Webshops erstellt haben, indem Sie die unten gezeigte Darstellung der Tabellenbeziehungen mit ihrer Darstellung der Tabellenbeziehungen im phpMyAdmin Designer vergleichen.

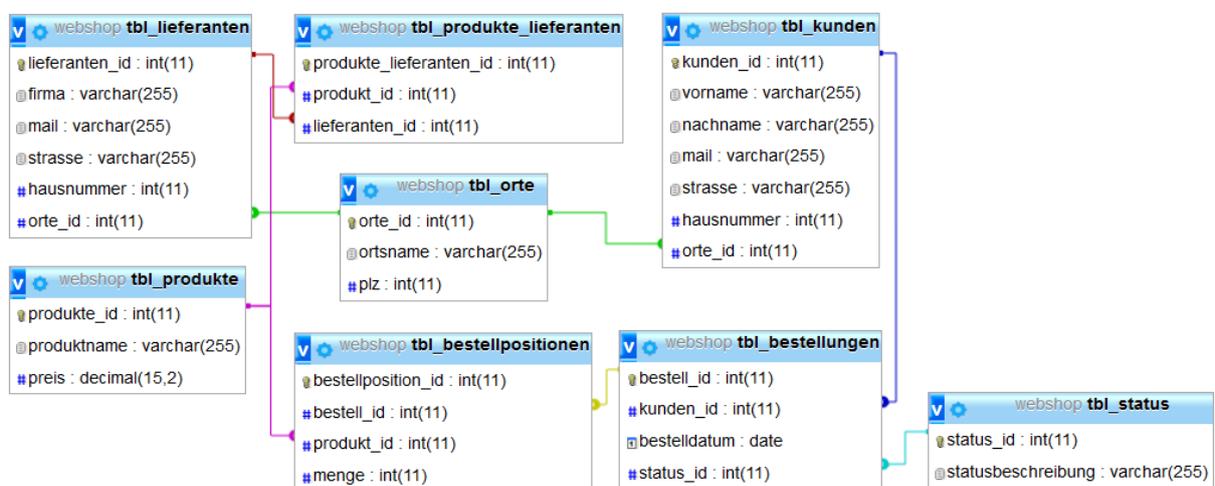


Abb. 48: Datenbank-Tabellen sortieren mit dem phpMyAdmin Designer

5.12 Zusammenfassung und Ausblick

In diesem WBT haben wir die Datenbank für den Webshop der Lemonline AG erstellt. Damit ist der letzte Schritt der Datenmodellierung abgeschlossen.

Im nächsten WBT werden wir die bereits vorhandenen Produkt- und Kundendaten in die Webshop-Datenbank einpflegen. Zudem werde ich Ihnen die Datenbanksprache "SQL" vorstellen.

6 Einpflegen der Produkt- und Kundendaten

6.1 Einleitung

Im letzten WBT haben wir gemeinsam die Webshop-Datenbank für die Lemonline AG erstellt.

In diesem WBT werden wir die Webshop-Datenbank mit Produkt- und Kundendaten befüllen. Dazu werden wir mittels phpMyAdmin und SQL die Produkt- und Kundendaten in die Webshop-Datenbank einpflegen, über welche die Lemonline AG bereits durch ihr Ladengeschäft verfügt.

6.2 Produktdaten einpflegen mit phpMyAdmin

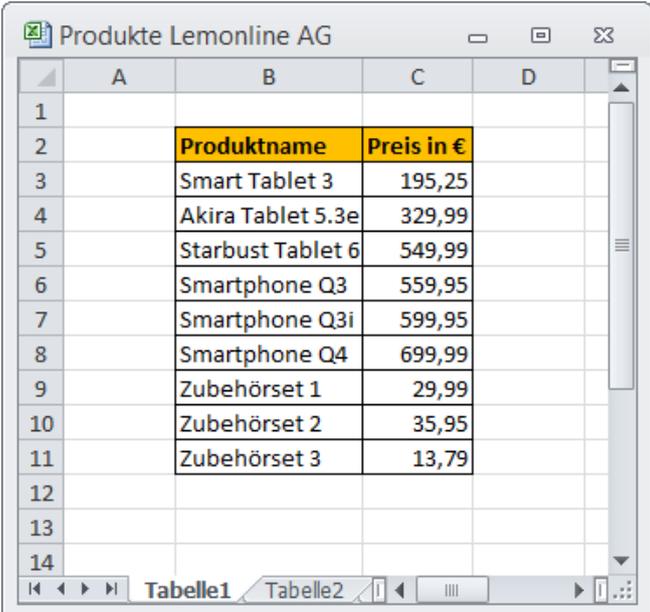
Damit die Kunden der Lemonline AG künftig ihre Wunschprodukte über den Webshop kaufen können, müssen diese zunächst in die Tabelle "tbl_produkte" eingepflegt werden.

Produkte können mit Hilfe von phpMyAdmin in die Webshop-Datenbank eingepflegt werden. Im Video sehen Sie, wie Sie dabei vorgehen müssen.

Im WBT können Sie sich an dieser Stelle ein Video anschauen, was Ihnen die Vorgehensweise zum Einpflegen von Produktdaten mit phpMyAdmin verdeutlicht.

6.3 Übung: Produktdaten einpflegen mit phpMyAdmin

Pflegen Sie die folgenden Produkte mit phpMyAdmin in die Tabelle "tbl_produkte" ein (Abb. 49).



	A	B	C	D
1				
2		Produktname	Preis in €	
3		Smart Tablet 3	195,25	
4		Akira Tablet 5.3e	329,99	
5		Starbust Tablet 6	549,99	
6		Smartphone Q3	559,95	
7		Smartphone Q3i	599,95	
8		Smartphone Q4	699,99	
9		Zubehörset 1	29,99	
10		Zubehörset 2	35,95	
11		Zubehörset 3	13,79	
12				
13				
14				

Abb. 49: Produkte der Lemonline AG

Bitte beachten Sie: Damit phpMyAdmin Preise richtig speichert, müssen Euro- und Cent-Beträge durch einen Punkt getrennt werden. Allgemein gilt, dass in MySQL als Dezimaltrennzeichen immer ein Punkt statt eines Kommas verwendet wird.

6.4 Datenübertragung I

Derzeit werden Eingaben in die MySQL-Datenbank des Webshops über die phpMyAdmin-Benutzeroberfläche vorgenommen.

Die MySQL-Datenbank gibt die Ergebnisse anschließend zurück an die phpMyAdmin-Benutzeroberfläche. Doch wie genau funktioniert diese Datenübertragung?

Die Datenbanksprache SQL ist eine Programmiersprache, welche zwischen einer Benutzeroberfläche wie z. B. phpMyAdmin und einer Datenbank wie z. B. MySQL vermittelt (Abb. 50).

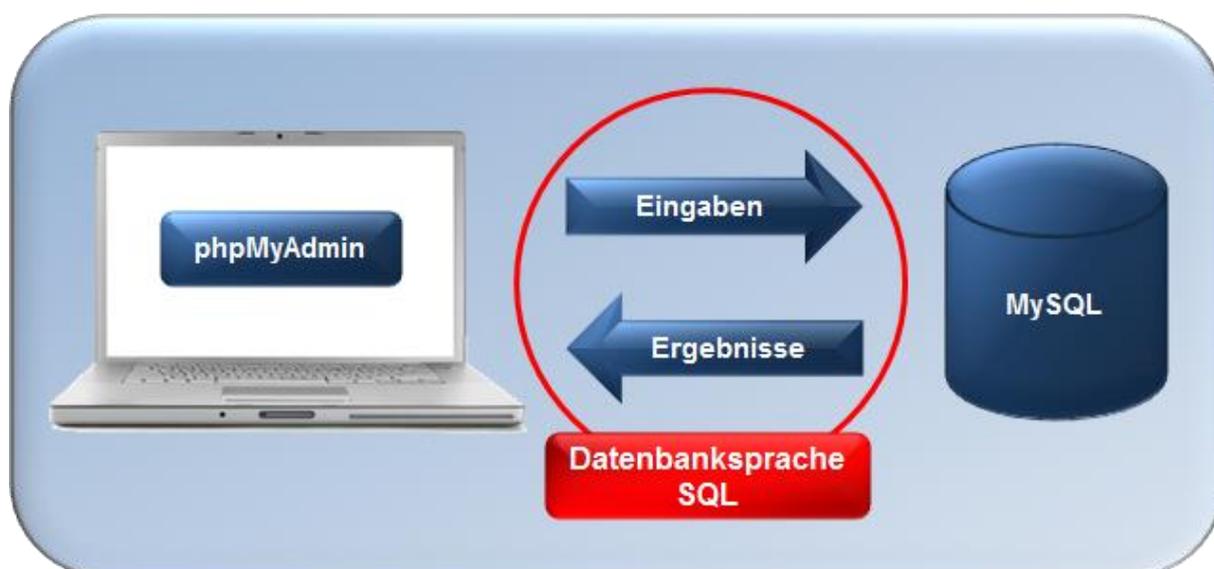


Abb. 50: Datenbanksprache SQL

6.5 Datenübertragung II

SQL ist auch in der Lage Daten von anderen Benutzeroberflächen, wie z. B. der Benutzeroberfläche eines Webshops, an die MySQL-Datenbank zu übertragen.

Zudem ist es mit SQL möglich, Daten nicht nur an die MySQL-Datenbank zu übertragen, sondern auch an andere relationale Datenbanken wie z. B. MS Access (Abb. 51).

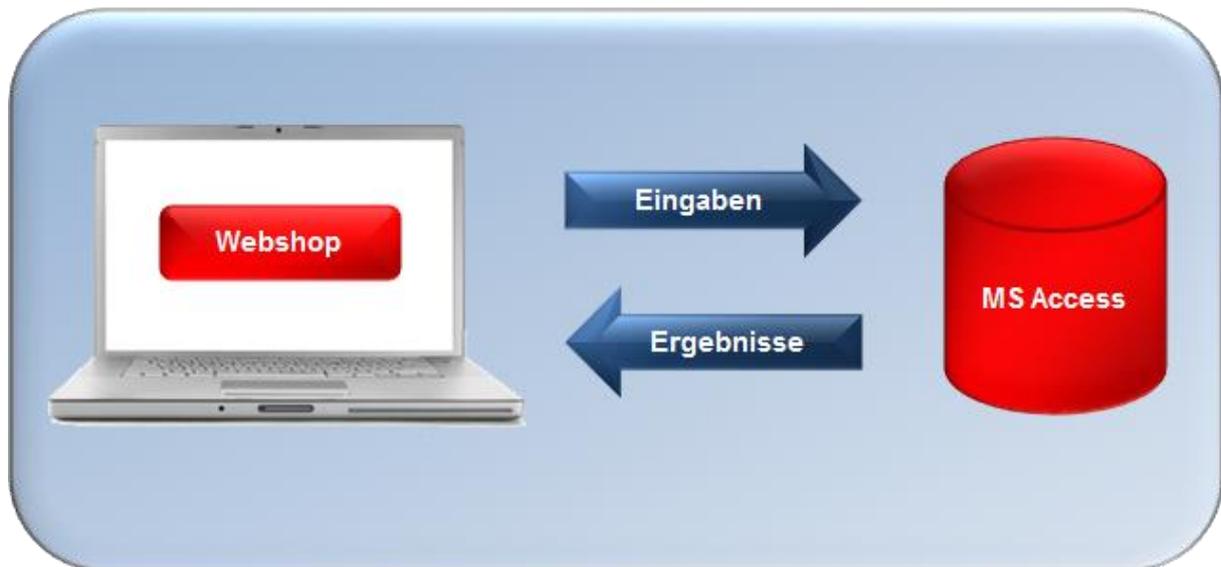


Abb. 51: Datenbanksprache SQL

6.6 Was ist SQL?

Die Structured Query Language (SQL) ist eine weit verbreitete Datenbanksprache für relationale Datenbanken. Für den Webshop der Lemonline AG muss SQL verwendet werden, um eine Datenübertragung mit der Webshop-Datenbank zu ermöglichen.

Mit Hilfe von SQL können Tabellen in der Webshop-Datenbank angelegt, befüllt, bearbeitet und abgefragt werden (Abb. 52).

Das Screenshot zeigt die Struktur der Tabelle 'tbl_produkte' in der Datenbank 'webshop'. Die Spalten sind:

Spaltenname	Datentyp
produkte_id	int(11)
produktname	varchar(255)
preis	decimal(15,2)

Abb. 52: Tabellenstruktur der Produkte-Tabelle

- Mit dem SQL-Befehl CREATE TABLE kann die Produkte-Tabelle in der Webshop-Datenbank angelegt werden:

```
CREATE TABLE tbl_produkte (  
    produkte_id INT(11),  
    produktname VARCHAR(255),  
    preis DECIMAL(15,2)  
)
```

- Mit dem SQL-Befehl INSERT INTO kann ein neues Produkt in die Webshop-Datenbank eingepflegt werden:

```
INSERT INTO `tbl_produkte`  
SET `produktname` = 'Smart Tablet 5', `preis` = '295.25';
```

- Mit dem SQL-Befehl UPDATE kann der Preis des Tablet 5 aktualisiert werden:

```
UPDATE `tbl_produkte`  
SET `preis` = '290'  
WHERE `produktname` = 'Tablet 5';
```

- Mit dem SQL-Befehl SELECT können alle Produktnamen aus der Produkte-Tabelle abgefragt werden (Abb. 53):

```
SELECT `produktname`  
FROM `tbl_produkte`;
```

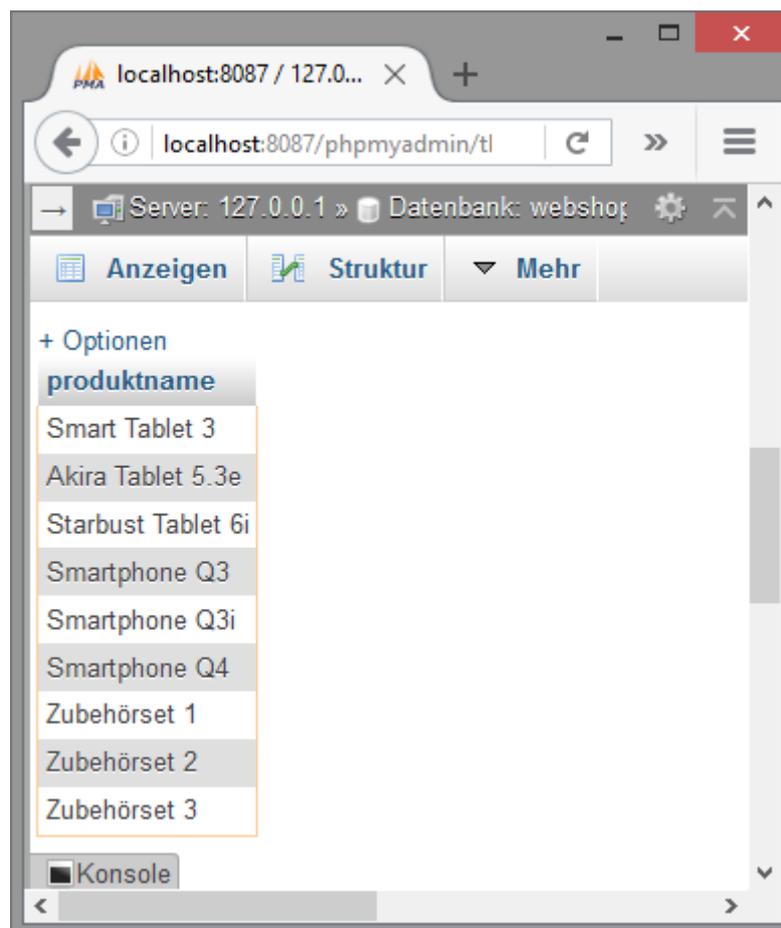
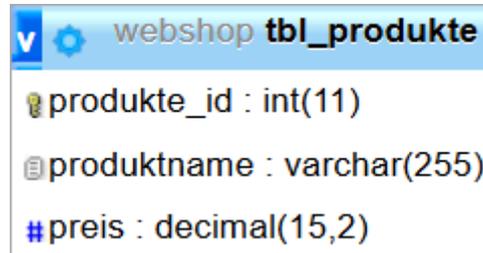


Abb. 53: Abfrageergebnis des SQL-Befehls

6.7 Produktdaten einpflegen mit SQL I

Um ein Produkt der Lemonline AG in die Produkte-Tabelle (tbl_produkte) einzupflegen, wird der SQL-Befehl "INSERT INTO" verwendet (Abb. 54).



Column Name	Data Type
produkte_id	int(11)
produktname	varchar(255)
preis	decimal(15,2)

Abb. 54: Tabellenstruktur der Produkte-Tabelle

Dieser Befehl muss einer fest vorgegebenen Struktur folgen.

```
INSERT INTO `tbl_produkte` (`produktname`,`preis`)
VALUES ('Smart Tablet 5', '295.25');
```

Nach dem SQL-Befehl wird festgelegt, dass das Produkt in die Tabelle tbl_produkte eingepflegt werden soll. In der Produkte-Tabelle sollen dabei alle Tabellenfelder befüllt werden.

Mit dem SQL-Befehl "VALUES" werden die Werte der Tabellenfelder festgelegt. Da die Produktnummer (produkte_id) automatisch durch das System vergeben wird, wird sie nicht im SQL-Befehl berücksichtigt.

Im WBT können Sie sich an dieser Stelle ein Video anschauen, was Ihnen die Vorgehensweise zum Einpflegen von Produktdaten mit SQL verdeutlicht.

6.8 Produktdaten einpflegen mit SQL II

Alternativ kann ein Produkt der Lemonline AG auch über die SQL-Befehle "INSERT INTO" und "SET" in die Produkte-Tabelle (tbl_produkte) eingepflegt werden. Auch dieser SQL-Befehl muss einer fest vorgegebenen Struktur folgen.

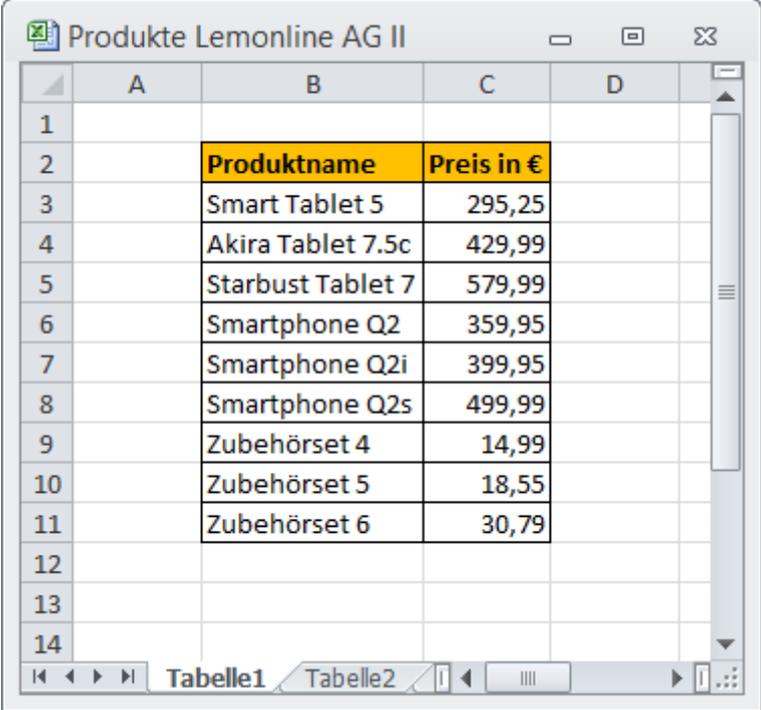
```
INSERT INTO `tbl_produkte`
SET `produktname` = 'Smart Tablet 5',
    `preis` = '295.25';
```

Nach dem SQL-Befehl "INSERT INTO" wird festgelegt, dass das Produkt in die Tabelle tbl_produkte eingepflegt werden soll.

Mit dem SQL-Befehl "SET" werden den Tabellenfeldern die jeweiligen Werte durch den Operator "=" zugeordnet. Da die Produktnummer (produkte_id) automatisch durch das System vergeben wird, wird sie nicht im SQL-Befehl berücksichtigt.

6.9 Übung: Produktdaten einpflegen mit SQL

Pflegen Sie zusätzlich zu den Produkten aus Übung 1 die folgenden Produkte mit SQL in die Tabelle "tbl_produkte" ein (Abb. 55). Nutzen Sie dazu die SQL-Befehle "INSERT INTO" und "VALUES" bzw. "SET".



	A	B	C	D
1				
2		Produktname	Preis in €	
3		Smart Tablet 5	295,25	
4		Akira Tablet 7.5c	429,99	
5		Starbust Tablet 7	579,99	
6		Smartphone Q2	359,95	
7		Smartphone Q2i	399,95	
8		Smartphone Q2s	499,99	
9		Zubehörset 4	14,99	
10		Zubehörset 5	18,55	
11		Zubehörset 6	30,79	
12				
13				
14				

Abb. 55: Produkte der Lemonline AG

6.10 Kundendaten einpflegen

Auch zum Einpflegen der Kundendaten wird der SQL-Befehl "INSERT INTO" verwendet. Durch mehr Attribute wird dieser Befehl hier natürlich etwas länger. Für die Kundin Frau Müller sieht das dann z. B. so aus:

```
INSERT INTO `tbl_kunden`  
SET `vorname` = 'Rebecca',  
    `nachname` = 'Müller',  
    `mail` = 'rebecca.mueller@example.com',  
    `strasse` = 'Birkenweg',
```

```
`hausnummer` = '3',
```

```
`orte_id` = '?.?.?';
```

Oh, hier gibt es ein Problem. Beim Eintragen der Kundendaten muss an einer Stelle die "orte_id" eingetragen werden, da sie ein Fremdschlüssel in der Kunden-Tabelle (tbl_kunden) ist (Abb. 56). Schauen wir uns das genauer an...

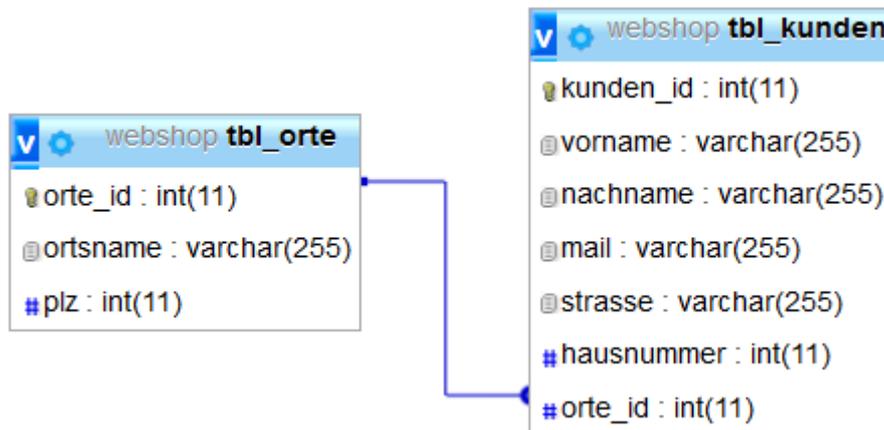


Abb. 56: Orte- und Kunden-Tabelle

6.11 Kundendaten einpflegen II

Beim Einpflegen von Daten sind die Beziehungen zwischen den Tabellen der Webshop-Datenbank entscheidend für die Reihenfolge, in der die Daten einpflegt werden können.

So ist es notwendig, zunächst die Daten der Tabellen zu füllen, die keine Fremdschlüssel besitzen (z. B. "tbl_orte") (Abb. 57).

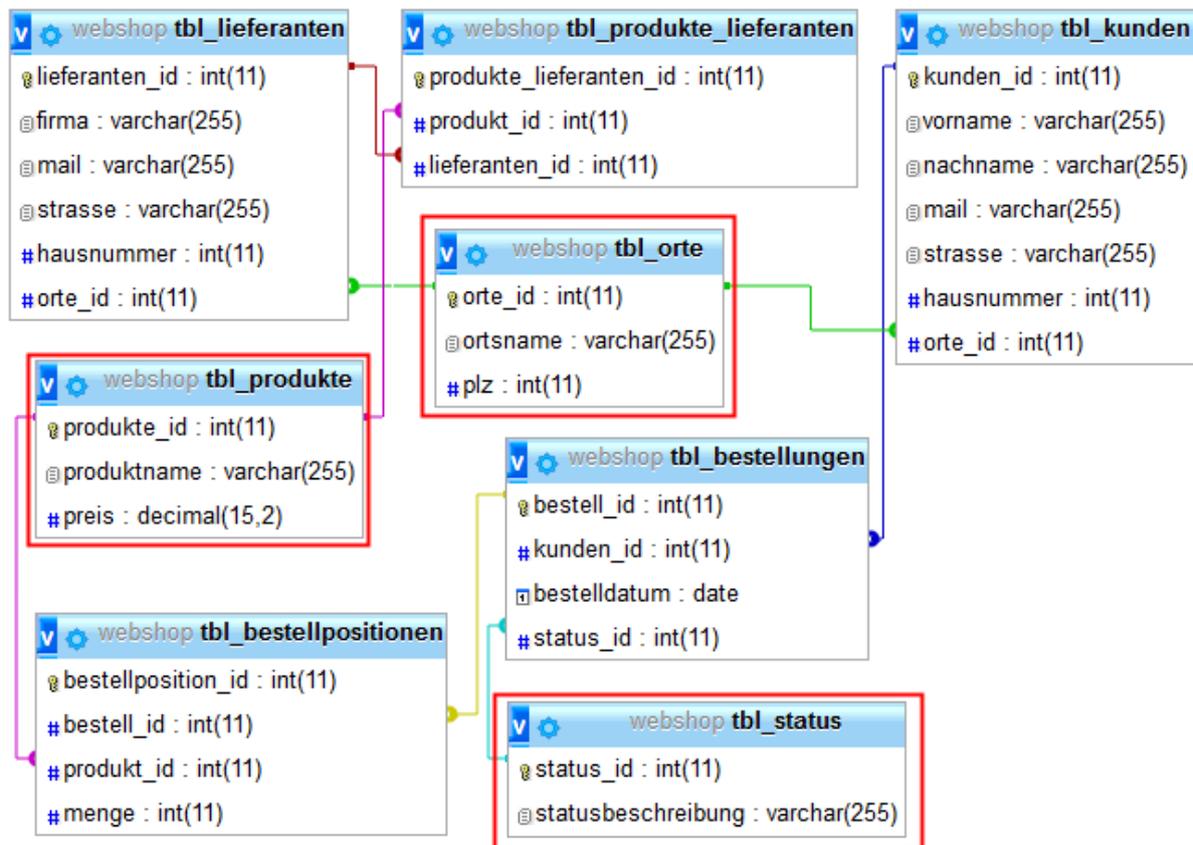


Abb. 57: Tabellen ohne Fremdschlüssel

Nachdem diese Tabellen gefüllt sind, ist es erst möglich die Tabellen mit Fremdschlüsseln zu füllen (z. B. "tbl_kunden") (Abb. 58).

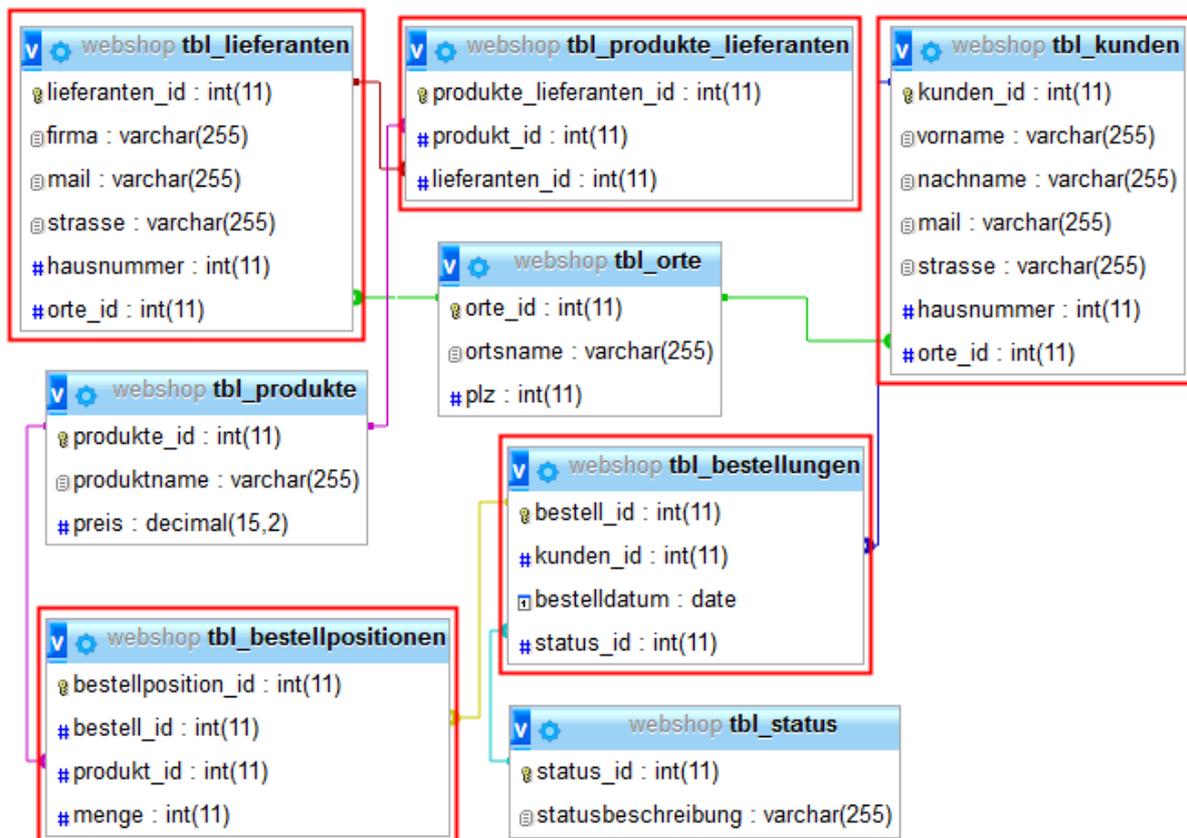


Abb. 58: Tabellen mit Fremdschlüsseln

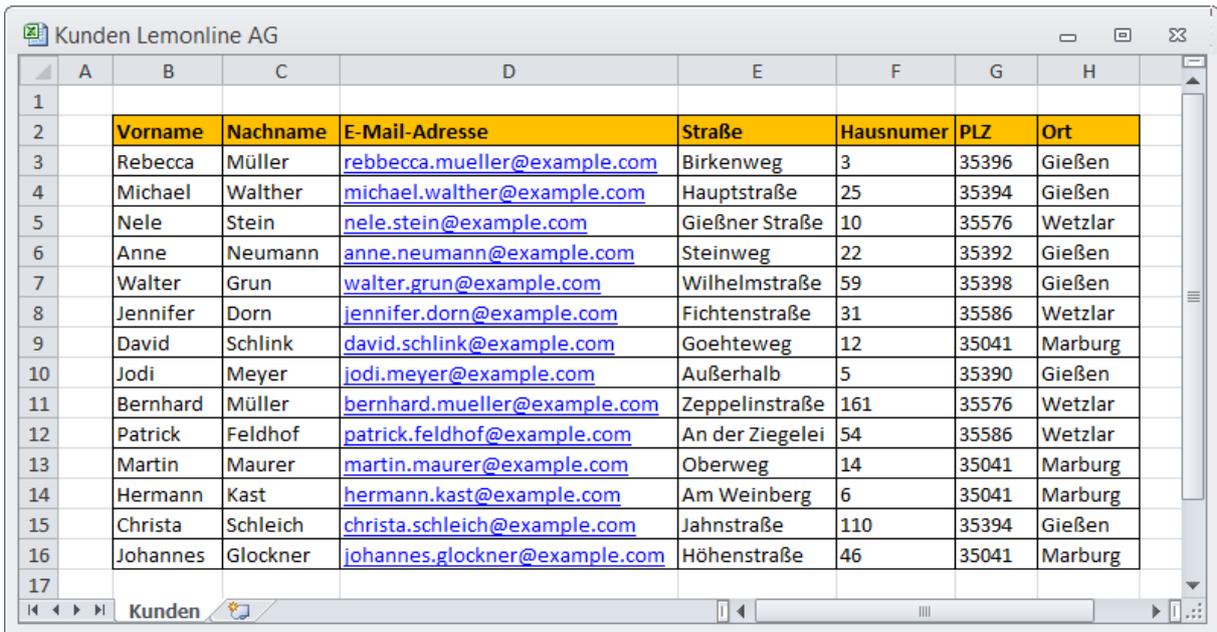
Im Video sehen Sie, wie bei dem stufenweisen Einpflegen der Kundendaten vorgegangen wird.

Im WBT können Sie sich an dieser Stelle ein Video anschauen, was Ihnen die Vorgehensweise zum stufenweisen Einpflegen von Kundendaten mit SQL verdeutlicht.

6.12 Übung: Kunden- und Ortsdaten einpflegen

Pflegen Sie die folgenden Kunden- und dazu gehörigen Ortsdaten in die beiden Tabellen "tbl_kunden" und "tbl_orte" ein. Gehen Sie dabei wie im Video auf der vorherigen WBT-Seite vor.

Durch Klick auf das Diskettensymbol können Sie sich die Excel-Datei mit den Kunden- und Ortsdaten herunterladen (Abb. 59).



	A	B	C	D	E	F	G	H
1								
2		Vorname	Nachname	E-Mail-Adresse	Straße	Hausnummer	PLZ	Ort
3		Rebecca	Müller	rebecca.mueller@example.com	Birkenweg	3	35396	Gießen
4		Michael	Walther	michael.walther@example.com	Hauptstraße	25	35394	Gießen
5		Nele	Stein	nele.stein@example.com	Gießner Straße	10	35576	Wetzlar
6		Anne	Neumann	anne.neumann@example.com	Steinweg	22	35392	Gießen
7		Walter	Grun	walter.grun@example.com	Wilhelmstraße	59	35398	Gießen
8		Jennifer	Dorn	jennifer.dorn@example.com	Fichtenstraße	31	35586	Wetzlar
9		David	Schlink	david.schlink@example.com	Gohteweg	12	35041	Marburg
10		Jodi	Meyer	jodi.meyer@example.com	Außerhalb	5	35390	Gießen
11		Bernhard	Müller	bernhard.mueller@example.com	Zeppelinstraße	161	35576	Wetzlar
12		Patrick	Feldhof	patrick.feldhof@example.com	An der Ziegelei	54	35586	Wetzlar
13		Martin	Maurer	martin.maurer@example.com	Oberweg	14	35041	Marburg
14		Hermann	Kast	hermann.kast@example.com	Am Weinberg	6	35041	Marburg
15		Christa	Schleich	christa.schleich@example.com	Jahnstraße	110	35394	Gießen
16		Johannes	Glockner	johannes.glockner@example.com	Höhenstraße	46	35041	Marburg
17								

Abb. 59: Kunden der Lemonline AG

6.13 Zusammenfassung und Ausblick

In diesem WBT haben wir die bereits vorhandenen Produkt- und Kundendaten der Lemonline AG in die Webshop-Datenbank eingepflegt.

Im nächsten WBT werden wir uns anschauen, wie wir mit SQL Daten aus der Webshop-Datenbank abfragen können. Dazu werden Sie den SQL-Befehl "SELECT" kennenlernen.

7 Abfragen der Produkt- und Kundendaten

7.1 Einleitung

Im letzten WBT haben wir gemeinsam die Produkt- und Kundendaten in die Webshop-Datenbank der Lemonline AG eingepflegt. Außerdem haben Sie gesehen, wozu SQL verwendet wird.

In diesem WBT werden wir uns ansehen, wie das Kundenmanagement der Lemonline AG funktioniert. Zur Eröffnung des Webshops plant die Lemonline AG Rabatt- und Werbeaktionen. Am Besten schauen wir uns anhand dieser Aktionen das Kundenmanagement der Lemonline AG an.

7.2 Kunden selektieren I

Zur Eröffnung des Webshops möchte die Lemonline AG ein Eröffnungsangebot erstellen. Dieses Angebot soll an alle Bestandskunden der Ladengeschäfte per E-Mail versendet werden. Um die E-Mail-Adressen der Bestandskunden aus der Webshop-Datenbank zu selektieren, wird der SQL-Befehl "SELECT " verwendet (Abb. 60).

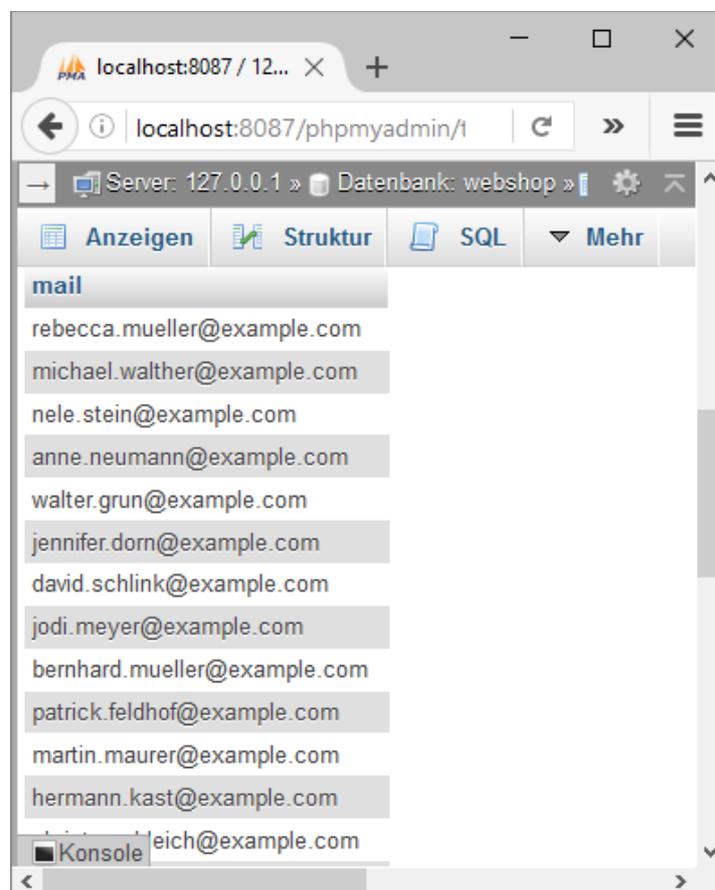


Abb. 60: Ergebnis der Abfrage

Dieser Befehl folgt einer fest vorgegebenen Struktur.

```
SELECT `mail` FROM `tbl_kunden`;
```

Der SQL-Befehl "SELECT `mail`" legt fest, dass nur die E-Mail-Adressen selektiert werden. Anschließend wird mit dem SQL-Befehl "FROM `tbl_kunden`" festgelegt, dass die E-Mail-Adressen aus der Kunden-Tabelle selektiert werden (Abb. 61).



Column Name	Data Type
kunden_id	int(11)
vorname	varchar(255)
nachname	varchar(255)
mail	varchar(255)
strasse	varchar(255)
hausnummer	int(11)
orte_id	int(11)

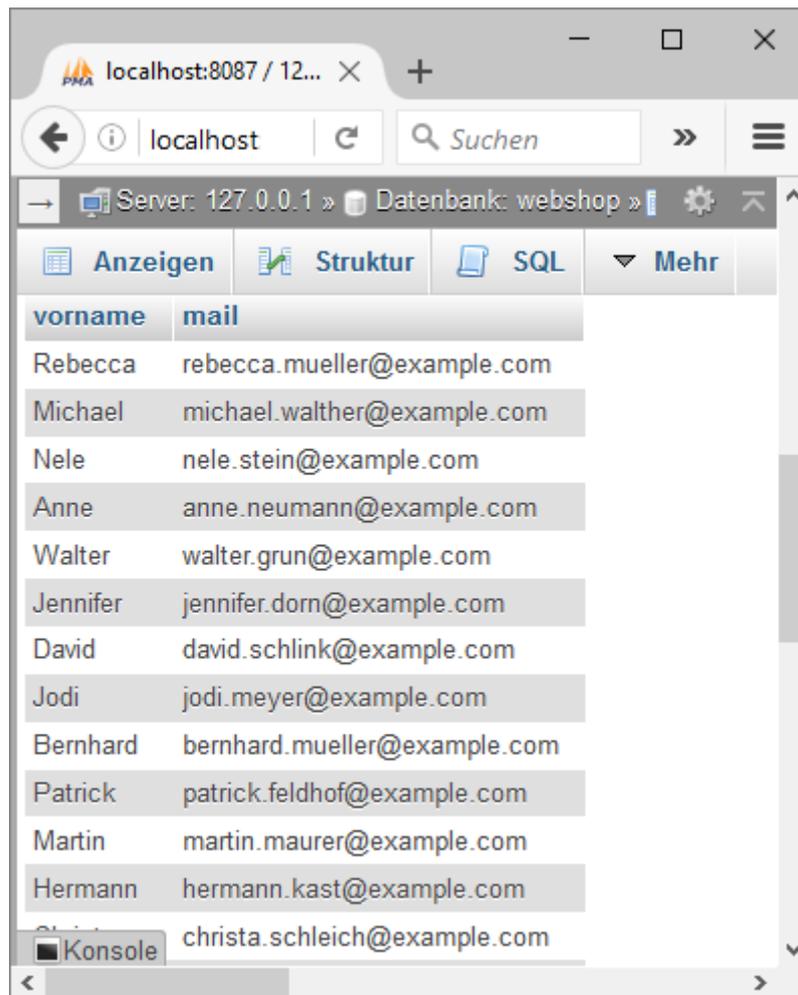
Abb. 61: Tabellenstruktur der Kunden-Tabelle

7.3 Kunden selektieren II

Der Geschäftsführer der Lemonline AG, Leo Lemon, möchte seine Bestandskunden in der E-Mail gerne persönlich ansprechen. Daher sollen zusätzlich zu den E-Mail-Adressen die Vornamen der Bestandskunden abgefragt werden (Abb. 62).

```
SELECT `vorname`, `mail` FROM `tbl_kunden`;
```

Mit dem SQL-Befehl "SELECT `vorname`, `mail`" wird daher festgelegt, dass die Vornamen und die E-Mail-Adressen selektiert werden. Anschließend wird mit dem SQL-Befehl "FROM `tbl_kunden`" festgelegt, dass diese Daten aus der Kunden-Tabelle selektiert werden.



vorname	mail
Rebecca	rebecca.mueller@example.com
Michael	michael.walther@example.com
Nele	nele.stein@example.com
Anne	anne.neumann@example.com
Walter	walter.grun@example.com
Jennifer	jennifer.dorn@example.com
David	david.schlink@example.com
Jodi	jodi.meyer@example.com
Bernhard	bernhard.mueller@example.com
Patrick	patrick.feldhof@example.com
Martin	martin.maurer@example.com
Hermann	hermann.kast@example.com
Christa	christa.schleich@example.com

Abb. 62: Ergebnis der Abfrage

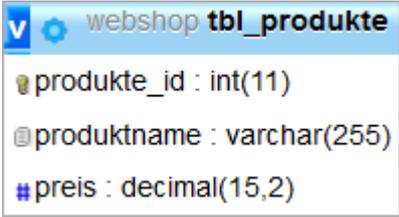
7.4 Angebot erstellen

Nachdem klar ist, welche Kunden über die Eröffnung des Webshops informiert werden, muss nun ein Angebot ausgearbeitet werden. Um eine Übersicht über die Produktpreise der Lemonline AG zu erhalten, ist es hilfreich, die Produkte nach Preisen zu sortieren.

```
SELECT * FROM `tbl_produkte`  
ORDER BY `preis` ASC;
```

Durch die SQL-Befehle "SELECT *" und "FROM `tbl_produkte`" wird festgelegt, dass alle Informationen aus der Produkte-Tabelle selektiert werden (Abb. 63).

- Der Stern im SQL-Befehl "SELECT *" dient als Platzhalter. Durch den Stern werden die Informationen aus allen vorhandenen Tabellenspalten ausgegeben.

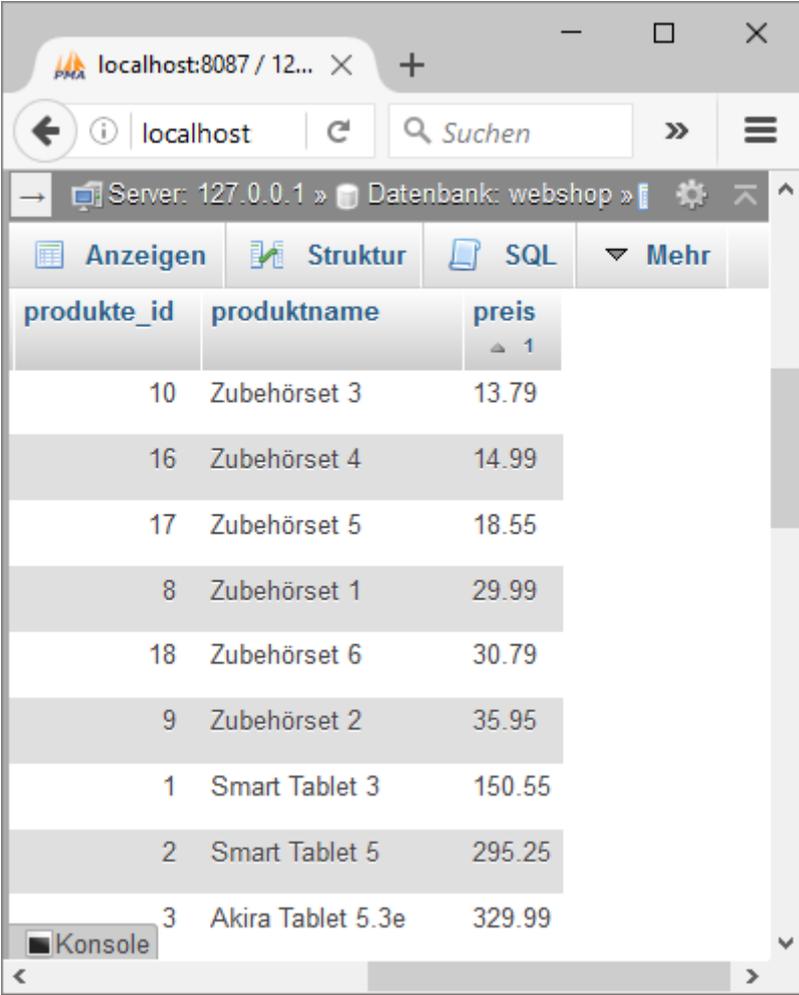


Column	DataType
produkte_id	int(11)
produktname	varchar(255)
preis	decimal(15,2)

Abb. 63: Tabellenstruktur der Produkte-Tabelle

Der SQL-Befehl "ORDER BY `preis`" legt fest, dass die Produkte nach der Spalte "preis" sortiert werden.

Mit dem SQL-Befehl "ASC" werden die Produkte aufsteigend sortiert (Abb. 64).

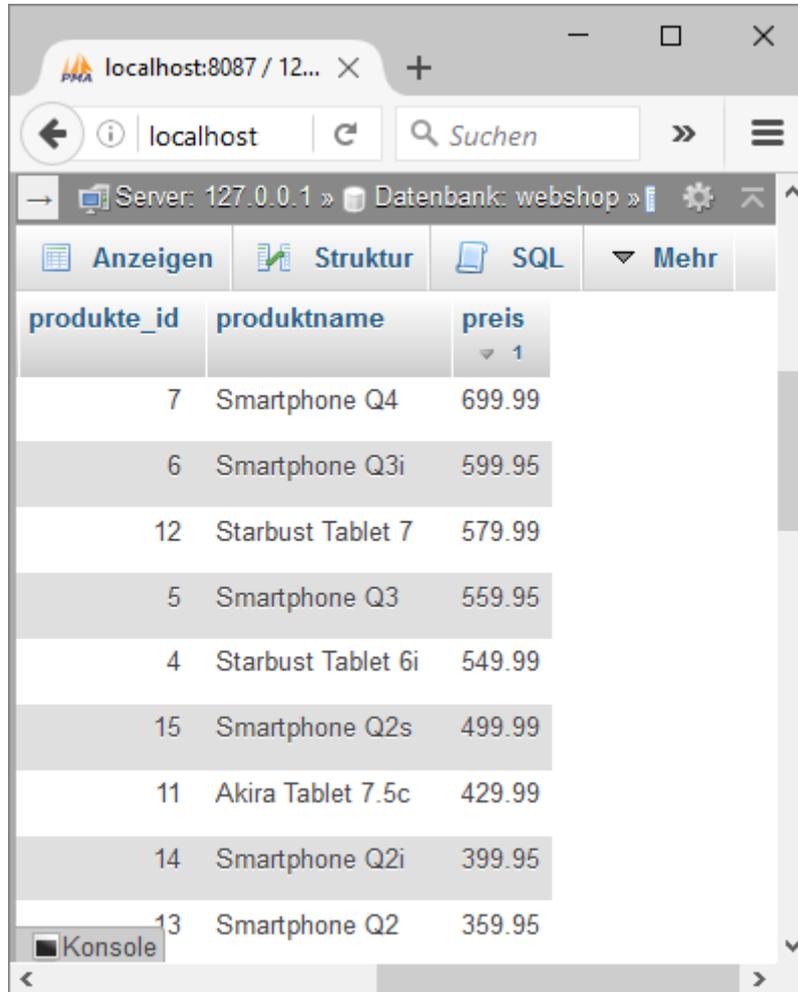


produkte_id	produktname	preis
10	Zubehörset 3	13.79
16	Zubehörset 4	14.99
17	Zubehörset 5	18.55
8	Zubehörset 1	29.99
18	Zubehörset 6	30.79
9	Zubehörset 2	35.95
1	Smart Tablet 3	150.55
2	Smart Tablet 5	295.25
3	Akira Tablet 5.3e	329.99

Abb. 64: Ergebnis der Abfrage aufsteigend sortiert

Alternativ können die Produkte absteigend sortiert werden mit dem SQL-Befehl "DESC" (Abb. 65).

```
SELECT * FROM `tbl_produkte`  
ORDER BY `preis` DESC;
```



produkte_id	produktname	preis
7	Smartphone Q4	699.99
6	Smartphone Q3i	599.95
12	Starbust Tablet 7	579.99
5	Smartphone Q3	559.95
4	Starbust Tablet 6i	549.99
15	Smartphone Q2s	499.99
11	Akira Tablet 7.5c	429.99
14	Smartphone Q2i	399.95
13	Smartphone Q2	359.95

Abb. 65: Ergebnis der Abfrage absteigend sortiert

7.5 Angebot eingrenzen

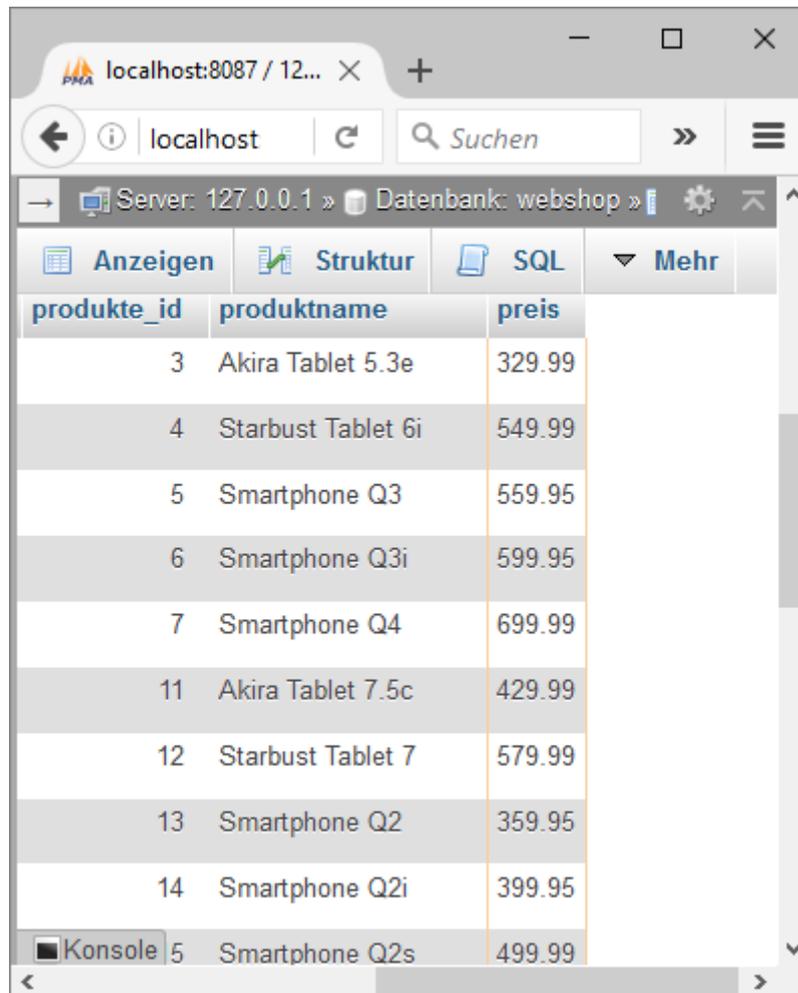
Der Geschäftsführer Leo Lemon hat entschieden, dass die Bestandskunden zur Eröffnung des Webshops 15% Rabatt auf alle Produkte erhalten, die teurer als 300€ sind (Abb. 66).

```
SELECT * FROM `tbl_produkte`  
WHERE `preis` > 300;
```

Durch die SQL-Befehle "SELECT *" und "FROM `tbl_produkte`" wird festgelegt, dass alle Informationen aus der Produkte-Tabelle abgefragt werden sollen.

Mit dem SQL-Befehl "WHERE `preis`>300" wird festgelegt, dass nur Informationen zu den Produkten ausgegeben werden, deren Preis größer ist als 300.

Mit "WHERE" können Bedingungen angegeben werden, um das Ergebnis der Abfrage einzuzugrenzen. Bedingungen werden mithilfe von Operatoren wie z. B. "<", ">", "=", ">=", "<=", LIKE, AND, OR, NOT, IN oder BETWEEN formuliert.



produkte_id	produktname	preis
3	Akira Tablet 5.3e	329.99
4	Starbust Tablet 6i	549.99
5	Smartphone Q3	559.95
6	Smartphone Q3i	599.95
7	Smartphone Q4	699.99
11	Akira Tablet 7.5c	429.99
12	Starbust Tablet 7	579.99
13	Smartphone Q2	359.95
14	Smartphone Q2i	399.95
5	Smartphone Q2s	499.99

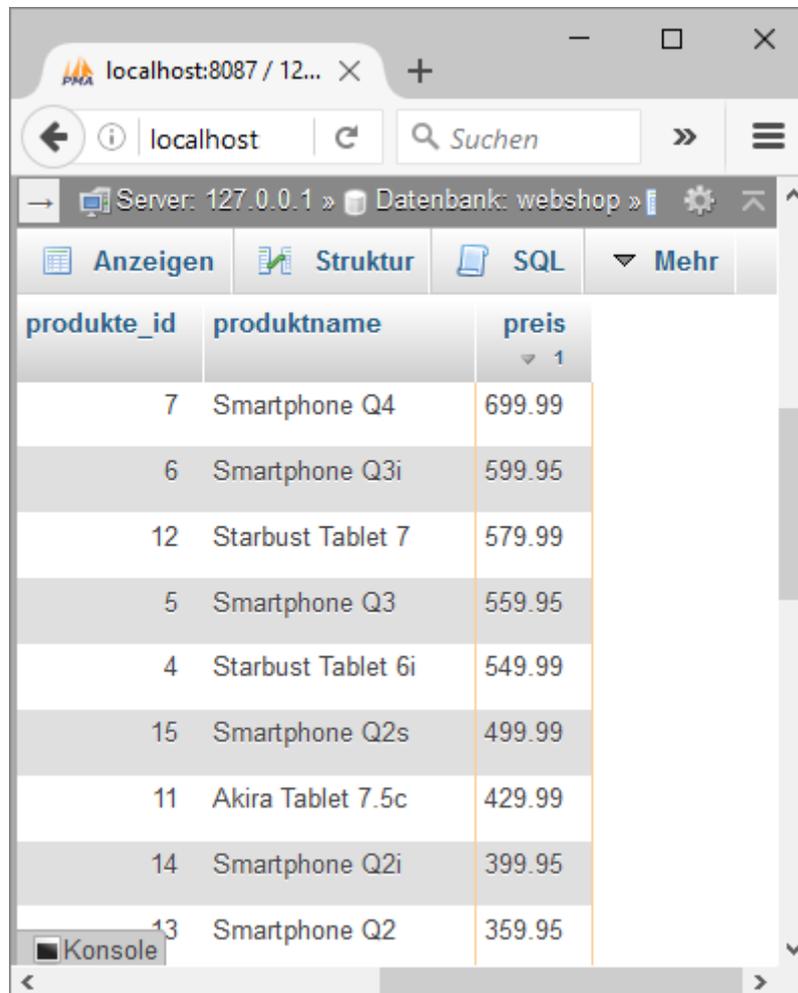
Abb. 66: Ergebnis der Abfrage

7.6 Angebot eingrenzen II

Um direkt sehen zu können, welche Produktpreise vom Rabatt betroffen sein werden, kann das eingegrenzte Angebot zusätzlich nach Produktpreisen sortiert werden (Abb. 67).

```
SELECT * FROM `tbl_produkte`  
WHERE `preis` > 300  
ORDER BY `preis` DESC;
```

Dazu wird zuerst festgelegt, dass alle Daten aus der Produkte-Tabelle (tbl_produkte) abgefragt werden sollen. Erst danach wird festgelegt, in welcher Reihenfolge die abgefragten Daten sortiert werden sollen.



produkte_id	produktname	preis
7	Smartphone Q4	699.99
6	Smartphone Q3i	599.95
12	Starbust Tablet 7	579.99
5	Smartphone Q3	559.95
4	Starbust Tablet 6i	549.99
15	Smartphone Q2s	499.99
11	Akira Tablet 7.5c	429.99
14	Smartphone Q2i	399.95
13	Smartphone Q2	359.95

Abb. 67: Ergebnis der Abfrage

7.7 Werbeaktion

Zusätzlich zum E-Mail-Angebot möchte die Lemonline AG Werbeaktionen in einigen Städten veranstalten. Diese Aktionen werden in den Wohnorten der Bestandskunden stattfinden. Um eine Liste mit den Wohnorten zu erstellen, wird der SQL-Befehl "SELECT DISTINCT" verwendet (Abb. 68).

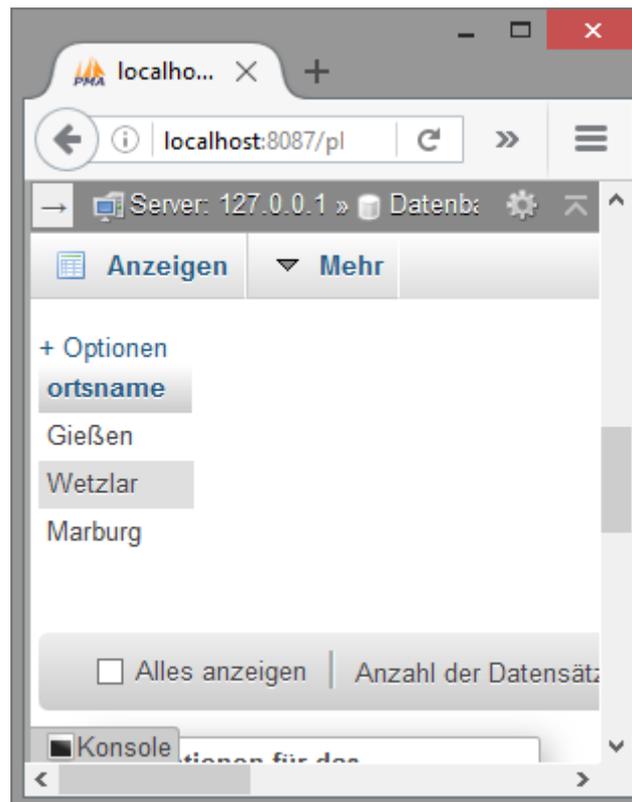


Abb. 68: Ergebnis der Abfrage mit dem SQL-Befehl „SELECT DISTINCT“

```
SELECT DISTINCT `ortsname` FROM `tbl_orte`;
```

Mit dem SQL-Befehl "SELECT DISTINCT `ortsname`" wird festgelegt, dass die Ortsnamen lediglich einmal ausgegeben werden, unabhängig davon wie oft sie in der Tabelle tatsächlich vorkommen.

Durch den SQL-Befehl "FROM `tbl_orte`" wird festgelegt, dass die Ortsnamen aus der Orte-Tabelle (tbl_orte) abgefragt werden (Abb. 69).

Column Name	Data Type
orte_id	int(11)
ortsname	varchar(255)
plz	int(11)

Abb. 69: Tabellenstruktur der Orte-Tabelle

Im Gegensatz zum SQL-Befehl "SELECT DISTINCT" werden beim SQL-Befehl "SELECT" mehrere Tabelleneinträge doppelt angezeigt (Abb. 70).

```
SELECT `ortsname` FROM `tbl_orte`;
```

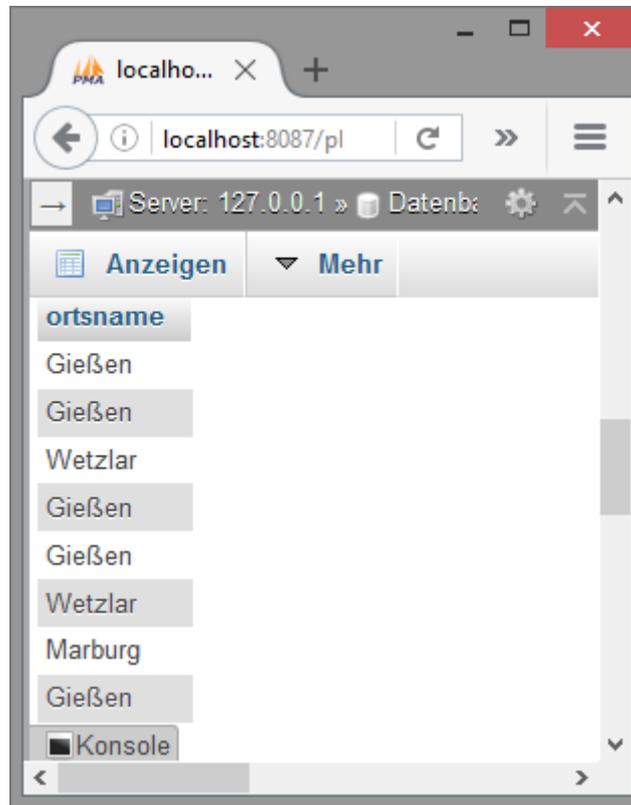


Abb. 70: Ergebnis der Abfrage mit dem SQL-Befehl „SELECT“

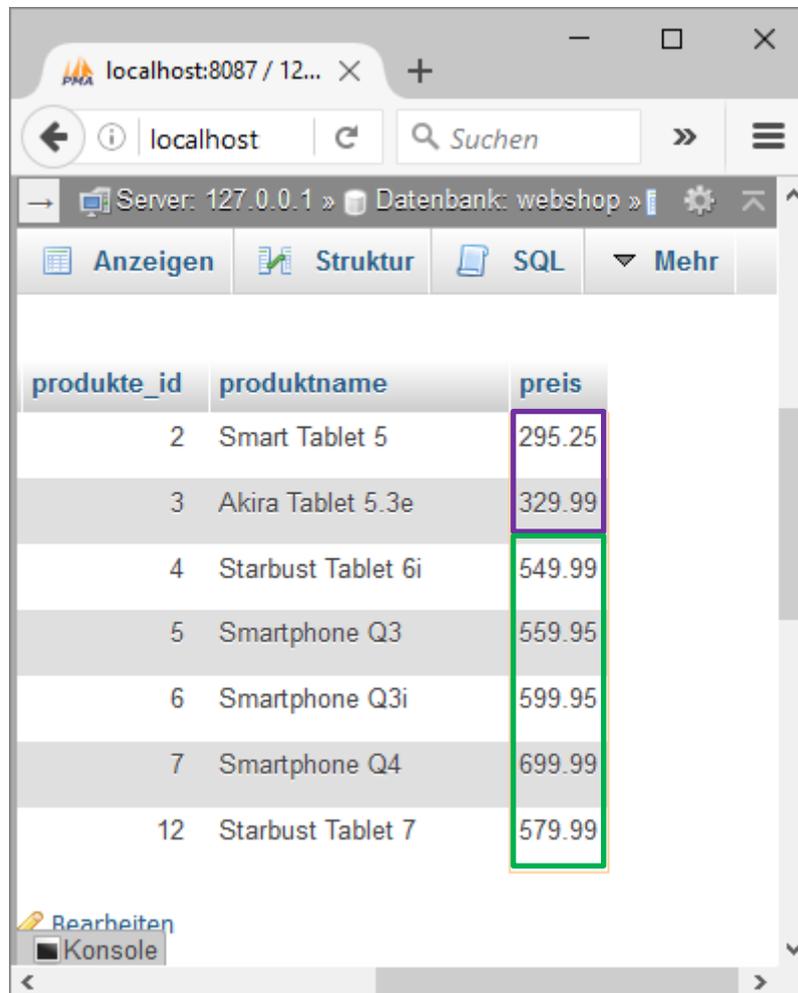
7.8 Werbeaktion II

Die Lemonline AG plant bei der Werbeaktion Flyer in den ausgewählten Städten zu verteilen, um neue Kunden auf den Webshop aufmerksam zu machen. Leo Lemon plant verschiedene Rabatte zu gewähren. Die ersten Neukunden erhalten 10% Rabatt auf alle Produkte, die teurer als 500€ sind und 15% Rabatt auf alle Produkte, die zwischen 250€ und 350€ kosten.

```
SELECT * FROM `tbl_produkte`  
WHERE `preis` > 500 OR  
`preis` BETWEEN 250 AND 350;
```

Mit dem SQL-Befehl "WHERE" wird festgelegt, dass nur Informationen zu den Produkten ausgegeben werden, die mindestens eine der beiden Bedingungen erfüllen (Abb. 71).

Mit den Operatoren AND und OR können mehrere Bedingungen miteinander verknüpft werden. Bei OR ist es ausreichend, wenn eine der Bedingungen erfüllt wird. Bei AND müssen alle verknüpften Bedingungen erfüllt sein.



produkte_id	produktname	preis
2	Smart Tablet 5	295.25
3	Akira Tablet 5.3e	329.99
4	Starbust Tablet 6i	549.99
5	Smartphone Q3	559.95
6	Smartphone Q3i	599.95
7	Smartphone Q4	699.99
12	Starbust Tablet 7	579.99

Abb. 71: Ergebnis der Abfrage

7.9 Grundlage zur Übung

Die Webshop-Datenbank der Lemonline AG dient als Grundlage für die Übung auf der nächsten WBT-Seite (Abb. 72).

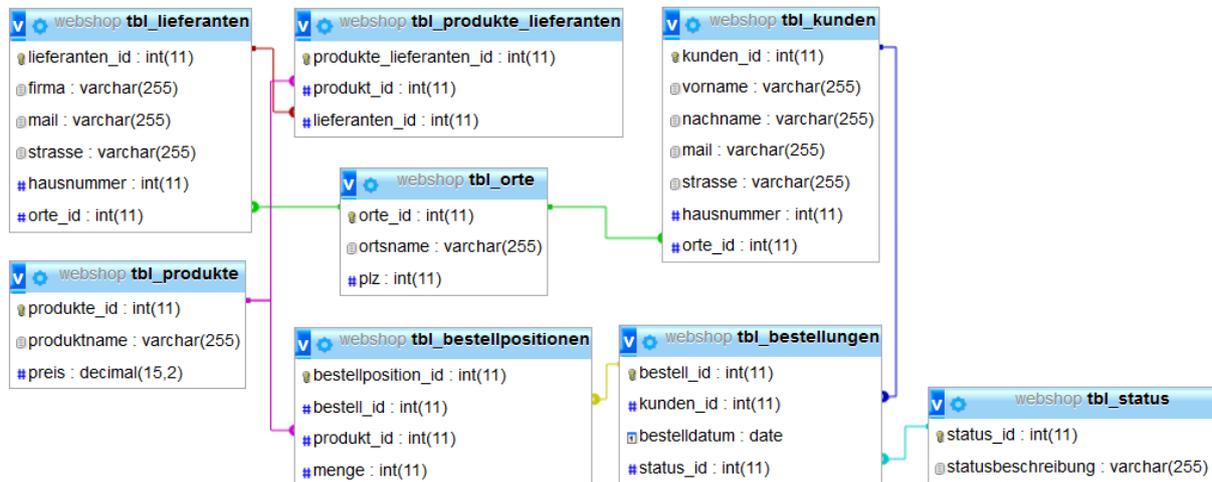


Abb. 72: Webshop-Datenbank der Lemonline AG

7.10 Übung: Abfragen der Produkt- und Kundendaten

Bitte beantworten Sie die folgenden Fragen durch Ankreuzen der korrekten Antworten. Bei einigen Fragen können auch mehrere Antworten richtig sein. Als Grundlage für die Testfragen dient die Webshop-Datenbank der Lemonline AG.

Nr.	Frage	Richtig	Falsch
1	Sie möchten herausfinden, welche Lieferanten ihren Firmensitz in Grünberg (orte_id: 9) oder Bad Nauheim (orte_id: 10) haben. Welche Antwort liefert das richtige Ergebnis?		
	SELECT `firma` FROM `tbl_lieferanten` WHERE `orte_id`=9 OR `orte_id`=10;		
	SELECT `firma` FROM `tbl_lieferanten` WHERE `orte_id`=9 AND `orte_id`=10;		
2	Sie möchten alle Daten der Tabelle "tbl_orte" nach den Postleitzahlen aufsteigend ordnen. Welche Antwort liefert das richtige Ergebnis?		
	SELECT * FROM `tbl_orte` ORDER BY `plz` ASC;		
	SELECT `orte_id`, `ortsname`, `plz` FROM `tbl_orte` ORDER BY `plz` ASC;		
	SELECT * FROM `tbl_orte` ORDER BY `plz` DESC;		

3	Sie möchten die E-Mailadressen der Kundin Rebecca Müller wissen. Welche Antwort liefert das richtige Ergebnis?		
	SELECT `mail` FROM `tbl_kunden` WHERE `nachname` LIKE 'Müller';		
	SELECT `mail` FROM `tbl_kunden` WHERE `nachname` LIKE 'Müller' AND `vorname` LIKE 'Rebecca';		
4	Sie möchten wissen, in welchem Status sich Bestellungen der Lemonline momentan befinden. Dabei möchten Sie jede Statusbeschreibung nur einmal angezeigt bekommen. Welche Antwort liefert das richtige Ergebnis?		
	SELECT DISTINCT * FROM `tbl_status`;		
	SELECT DISTINCT `statusbeschreibung` FROM `tbl_status`;		
5	Die Lemonline AG plant eine weitere Rabattaktion. Alle Produkte, die billiger als 100€ sind, erhalten 20% Rabatt. Sie sollen eine Liste erstellen mit allen betroffenen Produkten. Die Liste soll aufsteigend nach dem Preis sortiert sein. Welche Antworten liefern das richtige Ergebnis?		
	SELECT * FROM `tbl_produkte` WHERE `preis` < 100 ORDER BY ASC;		
	SELECT * FROM `tbl_produkte` WHERE `preis` < 100 ORDER BY `preis` ASC;		
	SELECT * FROM `tbl_produkte` WHERE `preis` LIKE 100 ORDER BY `preis` ASC;		
6	Sie möchten eine Liste aller Lieferanten aus Gießen (orte_id: 1, 2, 4, 5, 8) erstellen. Welche Antwort liefert das richtige Ergebnis?		
	SELECT * FROM `tbl_lieferanten` WHERE `orte_id`=1 OR `orte_id`=2 OR `orte_id`=4 OR `orte_id`=5 OR `orte_id`=8;		
	SELECT * FROM `tbl_lieferanten` WHERE `orte_id`=1 AND `orte_id`=2 AND `orte_id`=4 AND `orte_id`=5 AND `orte_id`=8;		

Tab. 2: Übungsfragen WBT 06

7.11 Zusammenfassung und Ausblick

In diesem WBT haben wir mit Hilfe von SQL-Befehlen verschiedene Informationen aus der Webshop-Datenbank der Lemonline AG abgefragt.

Im nächsten WBT werden wir uns anschauen, wie wir mit SQL Daten aus der Webshop-Datenbank aktualisieren und löschen können. Dazu werden Sie die SQL-Befehle "UPDATE" und "DELETE" kennenlernen.

8 Verwalten der Produkt- und Kundendaten

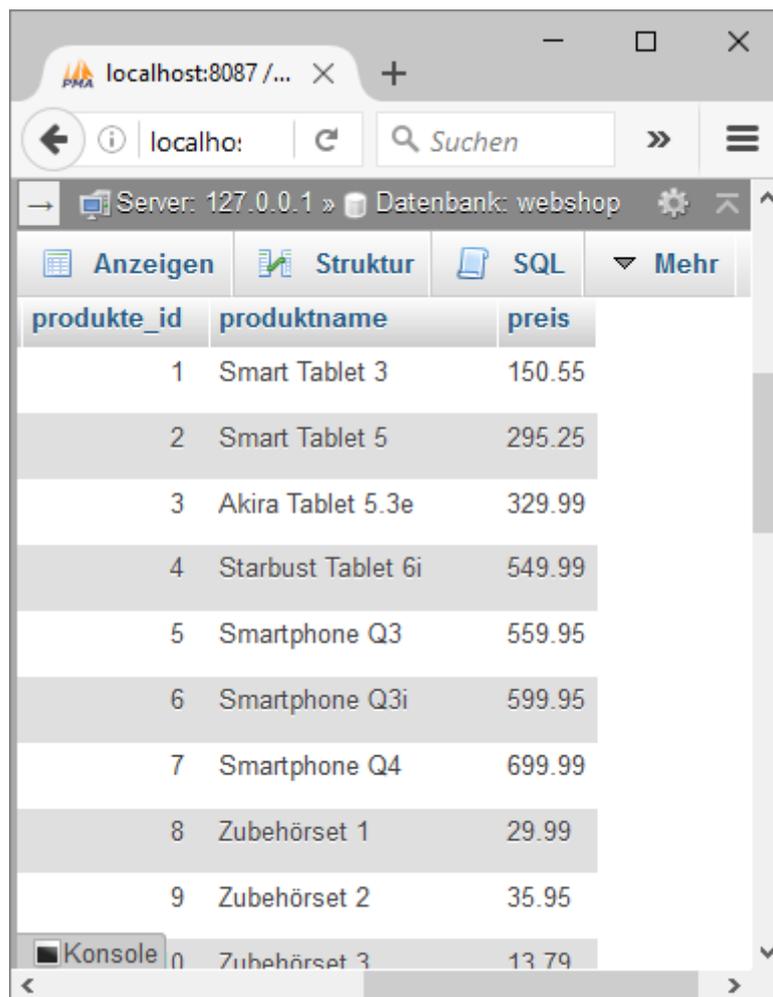
8.1 Einleitung

Im letzten WBT haben wir verschiedene Informationen aus der Webshop-Datenbank der Lemonline AG abgefragt. Dazu haben Sie den SQL-Befehl "SELECT" kennengelernt.

Produkt- und Kundendaten ändern sich häufig im Laufe der Zeit. In diesem WBT werden wir uns daher ansehen, wie Produkt- und Kundendaten der Webshop-Datenbank aktualisiert oder gelöscht werden können.

8.2 Preisänderung eines Produkts I

Der Preis des "Smart Tablet 3" ist von 195,25€ auf 150,55€ gesunken. Um den Preis des Tablets in der Webshop-Datenbank der Lemonline AG zu aktualisieren, wird der SQL-Befehl "UPDATE" verwendet (Abb. 73).



produkte_id	produktname	preis
1	Smart Tablet 3	150.55
2	Smart Tablet 5	295.25
3	Akira Tablet 5.3e	329.99
4	Starbust Tablet 6i	549.99
5	Smartphone Q3	559.95
6	Smartphone Q3i	599.95
7	Smartphone Q4	699.99
8	Zubehörset 1	29.99
9	Zubehörset 2	35.95
0	Zubehörset 3	13.79

Abb. 73: Ergebnis der Abfrage

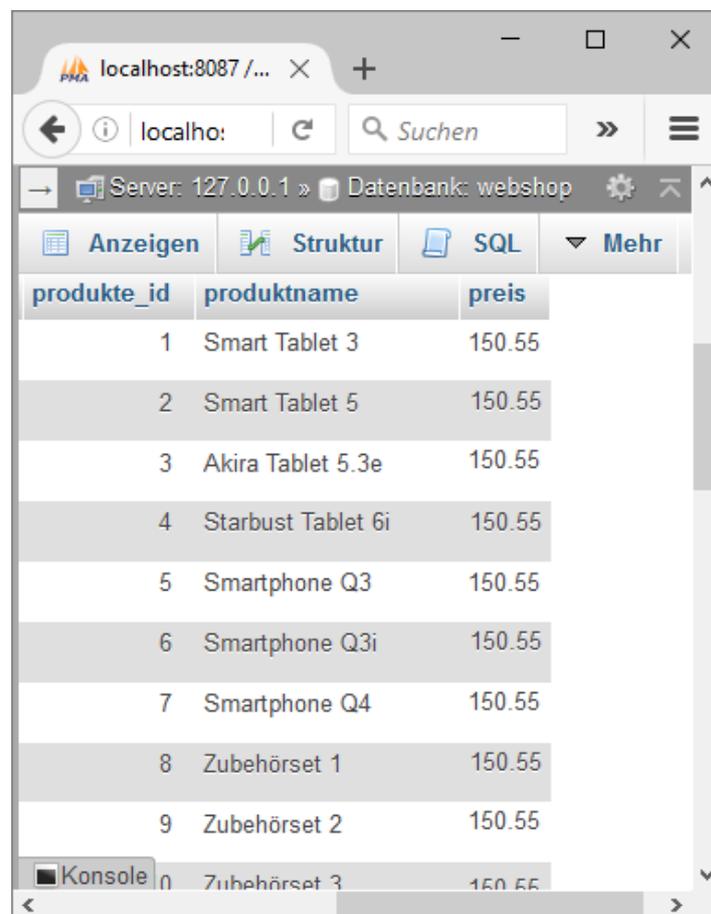
```
UPDATE `tbl_produkte`  
SET `preis` = 150.55  
WHERE `produkte_id` = 1;
```

Mit dem SQL-Befehl "UPDATE `tbl_produkte`" wird festgelegt, dass die Änderungen in der Produkte-Tabelle vorgenommen werden. Mit dem SQL-Befehl "SET `preis`= 150.55" wird anschließend der neue Preis des Tablets festgelegt. Abschließend wird mit dem SQL-Befehl "WHERE `produkte_id`= 1" festgelegt, dass der Preis nur bei dem Produkt mit der Produkte-ID "1" verändert wird.

Als WHERE-Bedingung wird die Produktnummer des Smart Tablets gewählt, da diese im Gegensatz zum Produktnamen das Produkt eindeutig identifiziert.

8.3 Preisänderung eines Produkts II

Vorsicht! Wird der UPDATE-Befehl nicht durch eine WHERE-Bedingung eingegrenzt, werden die Preise von allen Produkten in der Produkte-Tabelle (tbl_produkte) auf den neuen Preis geändert (Abb. 74).



The screenshot shows a web browser window displaying a database management interface. The browser address bar shows "localhost:8087/...". The database connection is "Server: 127.0.0.1" and "Datenbank: webshop". The interface has tabs for "Anzeigen", "Struktur", "SQL", and "Mehr". The "Anzeigen" tab is active, showing a table with the following data:

produkte_id	produktname	preis
1	Smart Tablet 3	150.55
2	Smart Tablet 5	150.55
3	Akira Tablet 5.3e	150.55
4	Starbust Tablet 6i	150.55
5	Smartphone Q3	150.55
6	Smartphone Q3i	150.55
7	Smartphone Q4	150.55
8	Zubehörset 1	150.55
9	Zubehörset 2	150.55
10	Zubehörset 3	150.55

Abb. 74: Ergebnis der Abfrage

```
UPDATE `tbl_produkte`
SET `preis` = 150.55;
```

8.4 Umzug eines Kunden

Frau Neumann hat der Lemonline AG mitgeteilt, dass sie von Wetzlar nach Marburg umgezogen ist. Die Adresse von Frau Neumann muss in der Webshop-Datenbank der Lemonline AG aktualisiert werden (Abb. 75).

kunden_id	vorname	nachname	strasse	hausnummer	orte_id
1	Rebecca	Müller	Birkenweg	3	1
2	Michael	Walther	Hauptstraße	25	2
3	Nele	Stein	Gießner Straße	10	3
4	Anne	Neumann	Turmweg	5	7
5	Walter	Grun	Wilhelmstraße	59	5
6	Jennifer	Dorn	Fichtenstraße	31	6
7	David	Schlink	Gohteweg	12	7
8	Jodi	Meyer	Außerhalb	5	8
9	Bernhard	Müller	Zeppelinstraße	161	3
	Patrick	Feldhof	An der Ziegerei	54	6

Abb. 75: Ergebnis der Abfrage

```
UPDATE `tbl_kunden`
SET `strasse` = 'Turmweg', `hausnummer` = 5, `orte_id` = 7
WHERE `kunden_id` = 4;
```

Mit dem SQL-Befehl "UPDATE `tbl_kunden`" wird daher festgelegt, dass die Änderungen in der Kunden-Tabelle vorgenommen werden (Abb. 76).



Abb. 76: Tabellenstruktur der Kunden-Tabelle

Mit dem SQL-Befehl "SET..." werden anschließend die neue Straße, Hausnummer und orte_id für Frau Neumann festgelegt.

- **Hinweis:** Die Orte und deren zugehörige Ortsnummern (orte_id) sind in der Tabelle tbl_orte festgelegt. In dieser Tabelle hat Marburg die orte_id "7" erhalten.

Abschließend wird mit dem SQL-Befehl "WHERE `kunden_id` = 4" festgelegt, dass die Angaben nur bei der Kundin mit der Kunden-ID "4" verändert werden.

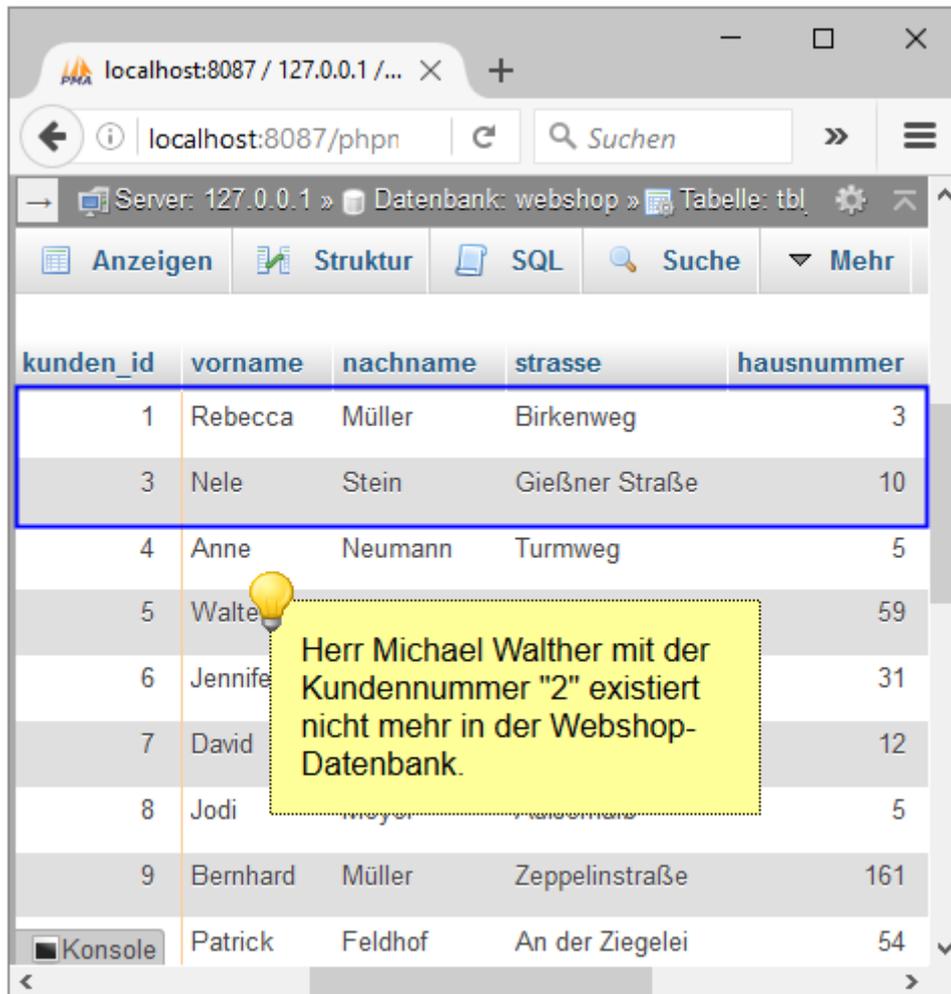
8.5 Kundenkonto löschen

Herr Michael Walther hat der Lemonline AG mitgeteilt, dass er nicht mehr als Kunde geführt werden möchte. Um alle Daten von ihm aus der Webshop-Datenbank der Lemonline AG zu löschen wird der SQL-Befehl "DELETE FROM" verwendet (Abb. 77).

```
DELETE FROM `tbl_kunden` WHERE `kunden_id` = 2;
```

Mit dem SQL-Befehl "DELETE FROM `tbl_kunden`" wird festgelegt, dass die Löschung in der Kunden-Tabelle vorgenommen wird. Abschließend wird mit dem SQL-Befehl "WHERE `kunden_id` = 2" festgelegt, dass die Daten nur bei dem Kunden mit der Kunden-ID "2" gelöscht werden.

Vorsicht! Wird der DELETE-Befehl nicht durch eine WHERE-Bedingung eingegrenzt, werden alle Kunden in der Kunden-Tabelle gelöscht. Außerdem müssen ggf. noch Kundendaten außerhalb der Kunden-Tabelle gelöscht werden wie z. B. Bestelldaten.



kunden_id	vorname	nachname	strasse	hausnummer
1	Rebecca	Müller	Birkenweg	3
3	Nele	Stein	Gießner Straße	10
4	Anne	Neumann	Turmweg	5
5	Walte			59
6	Jennife			31
7	David			12
8	Jodi			5
9	Bernhard	Müller	Zepelinstraße	161
	Patrick	Feldhof	An der Ziegelei	54

Abb. 77: Ergebnis der Abfrage

8.6 Sortimentsänderung I

Ein Produkthersteller hat mehrere Produkte aus seinem Sortiment genommen. Alle Daten dieser Produkte müssen nun aus der Webshop-Datenbank der Lemonline AG gelöscht werden (Abb. 78).

```
DELETE FROM `tbl_produkte`
WHERE `produkte_id` = 13 OR
      `produkte_id` = 14 OR
      `produkte_id` = 15;
```

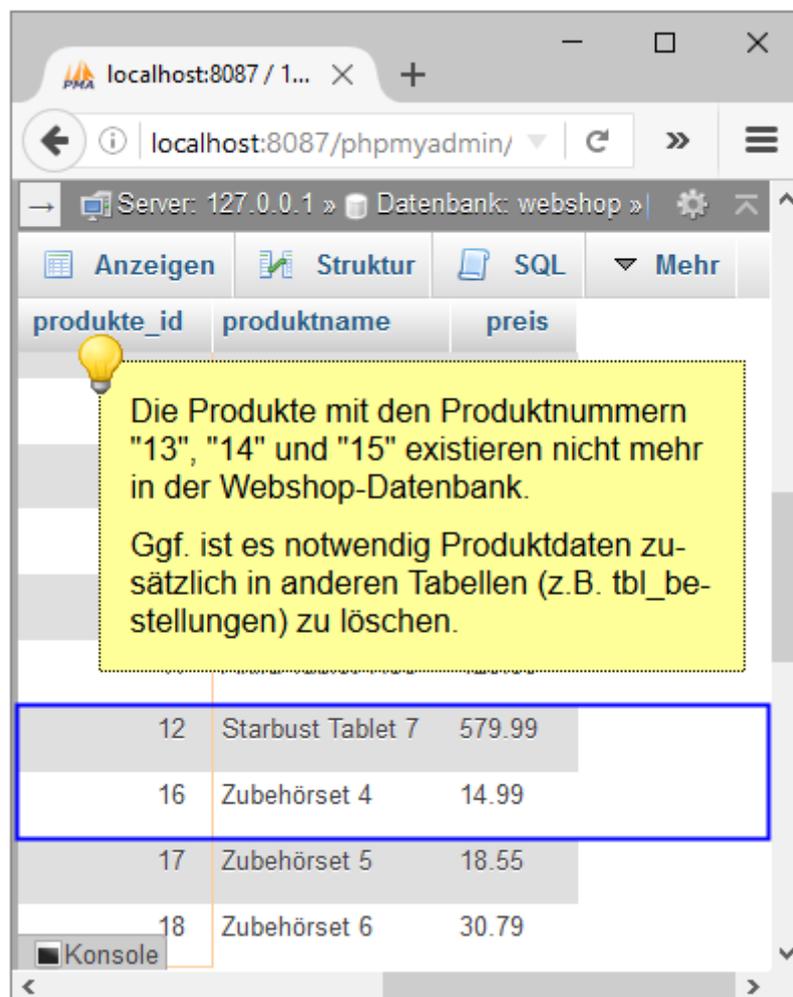
Mit dem SQL-Befehl "DELETE FROM `tbl_produkte`" wird festgelegt, dass die Löschung in der Produkte-Tabelle vorgenommen wird. Abschließend wird mit dem SQL-Befehl "WHERE..." festgelegt, dass die Daten nur bei den Produkten mit der Produkte-ID "13", "14" oder "15" gelöscht werden.

Alternativ können die Produkte mit drei einzelnen SQL-Befehlen nacheinander gelöscht werden.

```
DELETE FROM `tbl_produkte`  
WHERE `produkte_id` = 13;
```

```
DELETE FROM `tbl_produkte`  
WHERE `produkte_id` = 14;
```

```
DELETE FROM `tbl_produkte`  
WHERE `produkte_id` = 15;
```



Die Produkte mit den Produktnummern "13", "14" und "15" existieren nicht mehr in der Webshop-Datenbank.
Ggf. ist es notwendig Produktdaten zusätzlich in anderen Tabellen (z.B. tbl_bestellungen) zu löschen.

produkte_id	produktname	preis
12	Starbust Tablet 7	579.99
16	Zubehörset 4	14.99
17	Zubehörset 5	18.55
18	Zubehörset 6	30.79

Abb. 78: Ergebnis der Abfrage

8.7 Sortimentsänderung II

Die WHERE-Bedingung zur Löschung der Produkte kann auch mit dem Operator "IN" formuliert werden. Dieser Operator ersetzt die Verknüpfung von mehreren Bedingungen mit "OR" und vereinfacht so die WHERE-Bedingung.

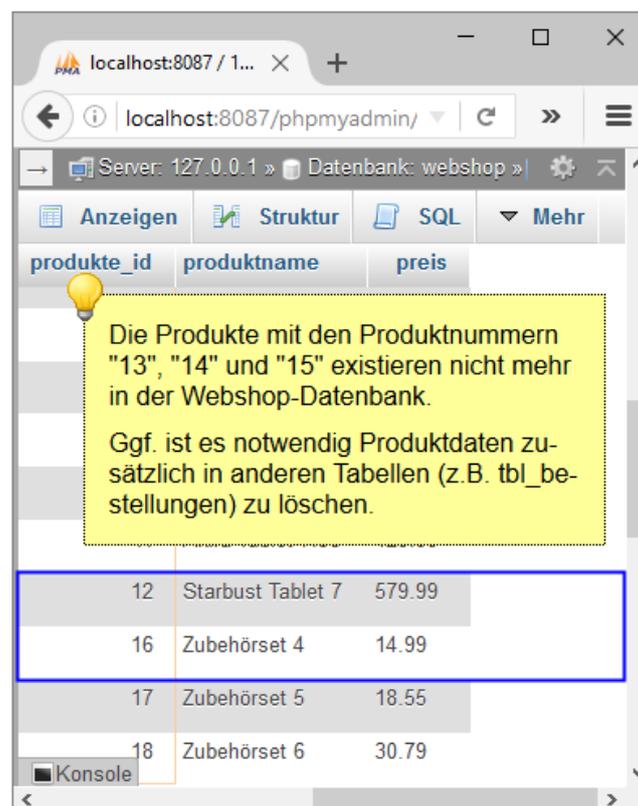
WHERE-Bedingung mit dem Operator „IN“:

```
DELETE FROM `tbl_produkte`  
WHERE `produkte_id`  
IN(13, 14, 15);
```

WHERE-Bedingung mit dem Operator „OR“:

```
DELETE FROM `tbl_produkte`  
WHERE  
`produkte_id` = 13 OR  
`produkte_id` = 14 OR  
`produkte_id` = 15;
```

Die beiden SQL-Befehle führen dabei zum gleichen Ergebnis und löschen die festgelegten Produkte (Abb. 79).



produkte_id	produktname	preis
12	Starbust Tablet 7	579.99
16	Zubehörset 4	14.99
17	Zubehörset 5	18.55
18	Zubehörset 6	30.79

Abb. 79: Ergebnis der Abfrage

8.8 Übung: Verwalten der Produkt- und Kundendaten I

Einige Produkt- und Kundendaten haben sich im Laufe der Zeit geändert. Nehmen Sie folgende Änderungen bzw. Löschungen mit Hilfe eines geeigneten SQL-Befehls in der Webshop-Datenbank der Lemonline AG vor:

1. Der Preis des "Smart Tablet 3" ist von 195,25€ auf 150,55€ gesunken. Aktualisieren Sie den Preis des Tablets in der Webshop-Datenbank.
2. Frau Anne Neumann ist in den Turmweg 5 nach Marburg umgezogen. Aktualisieren Sie die Adresse der Kundin in der Webshop-Datenbank.
3. Herr Michael Walther möchte nicht mehr als Kunde der Lemonline AG geführt werden. Löschen Sie alle Daten des Kunden aus der Webshop-Datenbank.
4. Ein Produkthersteller hat die drei Produkte "Smartphone Q2", "Smartphone Q2i" und "Smartphone Q2s" aus seinem Sortiment genommen. Löschen Sie alle Daten der Produkte aus der Webshop-Datenbank.

8.9 Grundlage zur Übung

Die Webshop-Datenbank der Lemonline AG dient als Grundlage für die Übung auf der nächsten WBT-Seite (Abb.80).

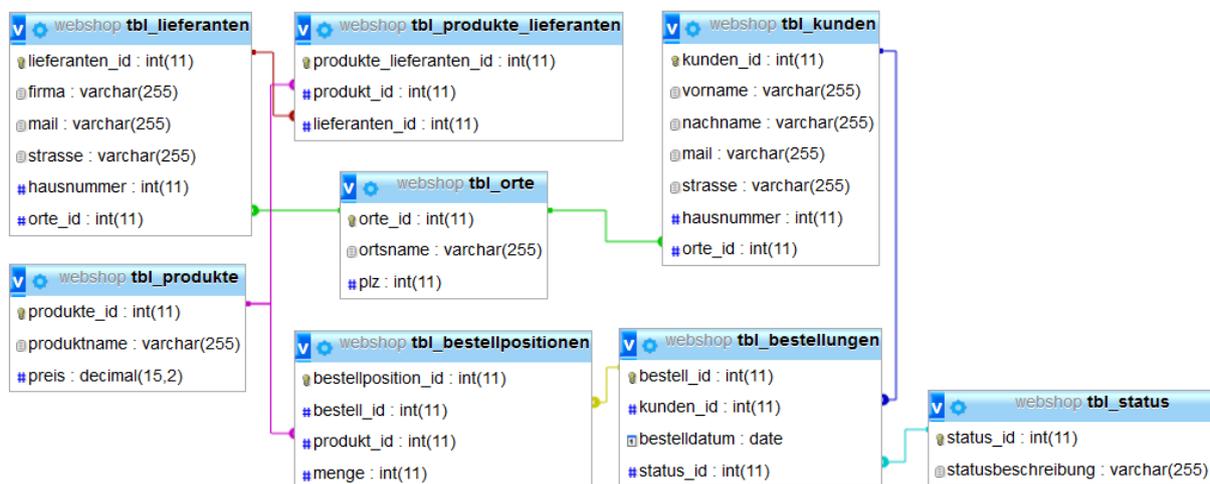


Abb. 80: Webshop-Datenbank der Lemonline AG

8.10 Übung: Verwalten der Produkt- und Kundendaten II

Bitte beantworten Sie die folgenden Fragen durch Ankreuzen der korrekten Antworten. Bei einigen Fragen können auch mehrere Antworten richtig sein. Als Grundlage für die Testfragen dient die Webshop-Datenbank der Lemonline AG.

Nr.	Frage	Richtig	Falsch
1	Sie möchten die E-Mail-Adresse eines Lieferanten (lieferanten_id: 9) ändern. Welche Antwort liefert das richtige Ergebnis?		
	UPDATE `tbl_lieferanten` SET `mail` = 'xy-ag@info.de';		
	UPDATE `tbl_lieferanten` SET `mail` = 'xy-ag@info.de' WHERE `lieferanten_id` = 9;		
2	Sie aktualisieren die Liste der Lieferanten. Dabei sollen alle Lieferanten, welche die Lemonline AG seit 3 Jahren nicht mehr beliefert haben (lieferanten_id: 5,15, 18,21) gelöscht werden. Welche Antwort liefert das richtige Ergebnis?		
	DELETE FROM `tbl_lieferanten` WHERE `lieferanten_id` = 5 OR `lieferanten_id` = 15 OR `lieferanten_id` = 18 OR `lieferanten_id` = 21;		
	DELETE FROM `tbl_lieferanten` WHERE `lieferanten_id` = 5 AND `lieferanten_id` = 15 AND `lieferanten_id` = 18 AND `lieferanten_id` = 21;		
	DELETE FROM `tbl_lieferanten` WHERE `lieferanten_id` IN (5, 15, 18, 21);		
3	Alle Bestellvorgänge wurden abgeschlossen. Sie möchten den Bestellstatus aller Bestellungen auf "geschlossen" (status_id: 2) ändern. Welche Antwort liefert das richtige Ergebnis?		
	UPDATE `tbl_status` SET `status_id` = 2;		
	UPDATE `tbl_status` SET `status_id` = '2';		
4	Sie möchten die Adresse von Frau Jennifer Dorn (kunden_id: 6) aktualisieren. Frau Dorn ist innerhalb ihres Wohnortes in den Tannenweg 9 umgezogen. Welche Antwort liefert das richtige Ergebnis?		
	UPDATE SET `strasse` = 'Tannenweg', `hausnummer` = 9 WHERE `kunden_id` = 6;		
	UPDATE `tbl_kunden` SET `strasse` = 'Tannenweg', `hausnummer` = 9 WHERE `kunden_id` = 6;		
	Keine der Antworten ist richtig		
5	Sie möchten den Kunden Bernhard Müller (kunden_id: 8) aus der Webshop-Datenbank der Lemonline AG löschen. Welche Antwort liefert das richtige Ergebnis?		
	DELETE `kunden_id` = 8 FROM `tbl_kunden`;		

	DELETE FROM `tbl_kunden` WHERE `kunden_id` = 8;		
6	Sie möchten den Preis des Produkts "Smart Tablet 5" (produkte_id: 2) aktualisieren. Welche Antwort liefert das richtige Ergebnis?		
	UPDATE `tbl_produkte` SET `preis` = 180,25 WHERE `produkte_id` = 2;		
	UPDATE `tbl_produkte` SET preis = 180.25 WHERE `produkte_id` = 2;		
	Keine der Antworten ist richtig		

Tab. 3: Übungsfragen WBT 07

8.11 Zusammenfassung und Ausblick

In diesem WBT haben wir mit Hilfe von SQL-Befehlen die Produkt- und Kundendaten der Webshop-Datenbank der Lemonline AG aktualisiert.

Im nächsten WBT werden wir uns anschauen, wie wir mit SQL-Befehlen die Struktur der Webshop-Datenbank anpassen können. Dazu werden Sie die SQL-Befehle "CREATE" und "ALTER" kennenlernen.

9 Anpassen der Webshop-Datenbank

9.1 Lieferanten- und Kunden-Tabelle anpassen

9.1.1 Einleitung

Im letzten WBT haben wir einige Produkt- und Kundendaten in der Webshop-Datenbank der Lemonline AG aktualisiert und gelöscht. Dazu haben Sie die SQL-Befehle "UPDATE" und "DELETE" kennengelernt.

Derzeit macht die Lemonline AG einen Probelauf mit der Webshop-Datenbank. Dabei ist aufgefallen, dass noch einige Änderungen in der Struktur der Webshop-Datenbank vorgenommen werden müssen. In diesem WBT werden wir uns daher ansehen, wie die Struktur der Webshop-Datenbank mit SQL angepasst werden kann.

9.1.2 Anpassen der Lieferanten-Tabelle

Vor der Eingabe der Lieferantendaten ist aufgefallen, dass einige Lieferanten Hausnummern wie z. B. 10a haben. Diese können derzeit nicht in die Webshop-Datenbank eingetragen werden, da das Feld "hausnummer" den Datentyp INT hat (Abb. 81).

- VORSICHT! Durch die Änderung des Datentyps werden oft alle Informationen gelöscht, die in diesem Feld hinterlegt wurden. Da die Lieferanten-Tabelle bisher noch nicht befüllt wurde, ist eine Änderung des Datentyps problemlos möglich.



Abb. 81: Lieferanten-Tabelle vor der Anpassung

Zudem soll der Name des Feldes in "hs-nr." geändert werden. Um den Namen und den Datentyp des Feldes in der Webshop-Datenbank der Lemonline AG zu ändern, wird der SQL-Befehl "ALTER" verwendet (Abb. 82).

```
ALTER TABLE `tbl_lieferanten`  
CHANGE `hausnummer` `hs-nr.` VARCHAR(255);
```



Abb. 82: Lieferanten-Tabelle nach der Anpassung

Mit dem SQL-Befehl "ALTER TABLE `tbl_lieferanten`" wird festgelegt, dass die Änderungen in der Lieferanten-Tabelle vorgenommen werden. Mit dem SQL-Befehl "CHANGE `hausnummer` `hs-nr.` VARCHAR (255)" wird festgelegt, dass die Änderungen das Feld bestehende "hausnummer" betreffen. Zudem wird festgelegt, dass das Feld in "hs-nr." umbenannt und den Datentyp VARCHAR(255) erhalten soll.

Der Datentyp INT legt fest, dass nur ganze Zahlen in diesem Feld gespeichert werden. Da im Feld "hs-nr." auch Buchstaben gespeichert werden müssen, muss das Feld den Datentyp VARCHAR erhalten. VARCHAR legt fest, dass Buchstaben, Ziffern und Zeichen in diesem Feld gespeichert werden können.

9.1.3 Anpassen der Kunden-Tabelle

In der Kunden-Tabelle soll es möglich sein, Zusatzdaten zu den Kundenadressen zu speichern, wie z. B. drittes Stockwerk. Dies erleichtert den Versanddiensten die Lieferung der Ware. Diese Zusatzinformationen sollen in einem neuen Feld gespeichert werden. Dazu wird der SQL-Befehl "ALTER" verwendet (Abb. 83).

```
ALTER TABLE `tbl_kunden` ADD `adresszusatz` VARCHAR(255);
```



Abb. 83: Kunden-Tabelle vor der Anpassung

Mit dem SQL-Befehl "ALTER TABLE `tbl_kunden`" wird festgelegt, dass die Änderungen in der Kunden-Tabelle vorgenommen werden. Mit dem SQL-Befehl "ADD `adresszusatz` VARCHAR (255)" wird festgelegt, dass das Feld den Namen "adresszusatz" und den Datentyp VARCHAR(255) erhalten soll (Abb. 84).



Abb. 84: Kunden-Tabelle nach der Anpassung

9.1.4 Übung: Lieferanten- und Kunden-Tabelle anpassen

Einige Änderungen müssen in der Struktur der Webshop-Datenbank vorgenommen werden. Nehmen Sie folgende Änderungen mit Hilfe geeigneter SQL-Befehle in der Webshop-Datenbank vor:

1. In der Lieferanten-Tabelle soll der Name des Feldes "hausnummer" auf "hs-nr." geändert werden. Das Feld soll zudem den Datentyp VARCHAR (255) erhalten. Nehmen Sie diese Änderungen in der Webshop-Datenbank vor.
2. In der Kunden-Tabelle soll das Feld "adresszusatz" erstellt werden. Das Feld soll den Datentyp VARCHAR (255) erhalten. Nehmen Sie diese Änderung in der Webshop-Datenbank vor.

9.2 Produkte kategorisieren

9.2.1 Produkte kategorisieren

Leo Lemon möchte Produkte in Produktkategorien wie z. B. Tablets einteilen. Um dies in der Webshop-Datenbank umzusetzen, müssen mehrere Änderungen vorgenommen werden:

1. Eine Produktkategorie-Tabelle muss angelegt werden, in der die Produktkategorien beschrieben werden können.

2. Ein neues Feld muss in der Produkte-Tabelle erstellt werden, welches als Fremdschlüssel zur Produktkategorien-Tabelle fungiert.
3. Die beiden Tabellen müssen über das Feld "produktkategorien_id" verbunden werden.

Unten sehen Sie die neue Struktur, die wir nun auf den nächsten WBT-Seiten in der Webshop-Datenbank umsetzen (Abb. 85).

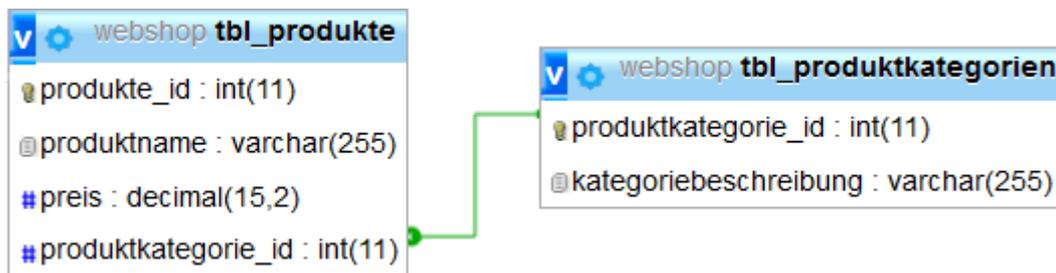


Abb. 85: Produkte kategorisieren

9.2.2 Produktkategorien anlegen

Leo Lemon möchte Produkte in der Webshop-Datenbank in Produktkategorien wie z. B. Tablets einteilen. Dazu muss zunächst eine neue Tabelle in der Webshop-Datenbank mit dem SQL-Befehl "CREATE" angelegt werden (Abb. 86).

```
CREATE TABLE tbl_produktkategorien
(
produktkategorie_id INT(11) AUTO_INCREMENT
NOT NULL PRIMARY KEY,

kategoriebeschreibung VARCHAR(255) NOT NULL
);
```

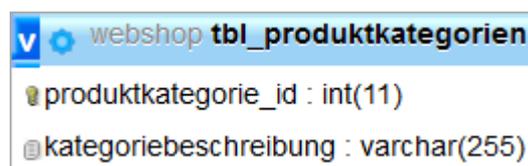


Abb. 86: Tabellenstruktur der Produktkategorien-Tabelle

Mit dem SQL-Befehl "CREATE TABLE tbl_produktkategorien" wird festgelegt, dass die neue Produktkategorien-Tabelle erstellt wird. Anschließend werden die beiden Felder "produktkategorie_id" und "kategoriebeschreibung" definiert.

Folgende Einstellungen werden für das Feld "produktkategorie_id" festgelegt:

- Das Feld erhält den Namen "produktkategorie_id".
- Das Feld erhält den Datentyp INT(11), sodass nur ganze Zahlen in diesem Feld gespeichert werden können.
- Die produktkategorie_id wird automatisch durch das System vergeben.
- Das Feld darf bei der Erstellung eines Datensatzes nicht leer bleiben.
- Das Feld ist der Primärschlüssel der Tabelle.

Folgende Einstellungen werden für das Feld "kategoriebeschreibung" festgelegt:

- Das Feld erhält den Namen "kategoriebeschreibung".
- Das Feld erhält den Datentyp VARCHAR(255), sodass nur Buchstaben und Zeichen in diesem Feld gespeichert werden können.
- Das Feld darf bei der Erstellung eines Datensatzes nicht leer bleiben.

9.2.3 Anpassen der Produkt-Tabelle

Die neue Tabelle "tbl_produktkategorien" soll mit der Produkte-Tabelle verbunden werden. Eine Verbindung der beiden Tabellen kann über das Feld "produktkategorie_id" erfolgen. Dazu muss nun zunächst in der Produkte-Tabelle mit dem SQL-Befehl "ALTER" das Feld "produktkategorie_id" erstellt werden (Abb. 87).

```
ALTER TABLE `tbl_produkte` ADD `produktkategorie_id` INT(11);
```

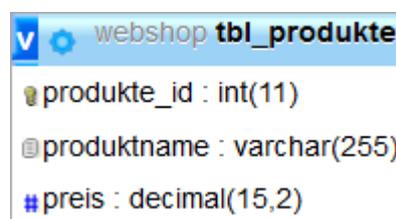


Abb. 87: Produkte-Tabelle vor der Anpassung

Mit dem SQL-Befehl "ALTER TABLE `tbl_produkte`" wird festgelegt, dass die Änderungen in der Produkte-Tabelle vorgenommen werden. Mit dem SQL-Befehl "ADD `produktkategorie_id` INT(11)" wird festgelegt, dass das Feld den Namen "produktkategorie_id" und den Datentyp INT(11) erhalten soll (Abb. 88).



Abb. 88: Produkte-Tabelle nach der Anpassung

9.2.4 Übung: Produkte kategorisieren

Lassen Sie mich kurz wiederholen, welche Änderungen Sie für die Lemonline AG durchführen sollen:

Leo Lemon möchte Produkte in der Webshop-Datenbank in Produktkategorien wie z. B. Tablets einteilen. Dazu muss zunächst eine Tabelle mit den Produktkategorien erstellt werden. Außerdem muss in der Produkte-Tabelle das Feld "produktkategorie_id" erstellt werden, über das die Produkte- und die Produktkategorien-Tabelle miteinander verbunden werden können.

Weil bereits Daten in der Produkte-Tabelle enthalten sind, müssen die Felder, über welche die beiden Tabellen miteinander verbunden werden, mit Daten gefüllt sein. Daher müssen wir sowohl in der Produkt- als auch in der Produktkategorien-Tabelle das Feld "produktkategorie_id" befüllen, bevor wir die beiden Tabellen verbinden können.

Aufgabenstellung der Übung:

1. Erstellen Sie eine neue Tabelle mit dem Namen „tbl_produktkategorien“. Diese soll zwei Felder mit den folgenden Eigenschaften enthalten:

produktkategorie_id:

- Datentyp INT(11),
- automatisch durch das System vergeben,
- Primärschlüssel der Tabelle,
- soll immer befüllt werden bei der Eingabe von Datensätzen

kategoriebeschreibung:

- Datentyp VARCHAR (255),
- soll immer befüllt werden bei der Eingabe von Datensätzen

2. Befüllen Sie die Produktkategorien-Tabelle mit folgenden Beschreibungen:
 - Smartphone
 - Tablet
 - Zubehörset
3. In der Produkte-Tabelle soll das Feld "produktkategorie_id" erstellt werden. Das Feld soll den Datentyp INT(11) erhalten. Nehmen Sie diese Änderung in der Webshop-Datenbank vor.
4. Ergänzen Sie in der Produkte-Tabelle zu jedem Produkt die geeignete produktkategorie_id. Nutzen Sie zur Einteilung in die Kategorien die folgende Tabelle als Hilfsmittel:

Produktname	Produktkategorie
Smart Tablet 3	Tablet
Akira Tablet 5.3e	Tablet
Starbust Tablet 6i	Tablet
Smartphone Q3	Smartphone
Smartphone Q3i	Smartphone
Smartphone Q4	Smartphone
Zubehörset 1	Zubehörset
Zubehörset 2	Zubehörset
Zubehörset 3	Zubehörset
Smart Tablet 5	Tablet
Akira Tablet 7.5c	Tablet
Starbust Tablet 7	Tablet
Zubehörset 4	Zubehörset
Zubehörset 5	Zubehörset
Zubehörset 6	Zubehörset

Tab. 4: Produkte und deren Produktkategorien

9.2.5 Produkte und Produktkategorien verknüpfen I

Jedes Produkt kann einer Produktkategorie zugeordnet werden. Diese Beziehung wird technisch über die beiden Felder "produktkategorie_id" hergestellt. Das DBMS weiß jedoch nicht automatisch, welches Produkt welcher Produktkategorie zugeordnet werden soll. Daher muss vor der Verknüpfung der Tabellen jedem Produkt eine der Produktkategorien zugeordnet werden. Erst dann kann das DBMS die beiden Tabellen miteinander verknüpfen.

Im WBT können Sie sich an dieser Stelle eine Animation anschauen, die Ihnen die Verknüpfung der Produkte und der Produktkategorien verdeutlicht.

9.2.6 Produkte und Produktkategorien verknüpfen II

Es soll eine Beziehung zwischen der Produkt- und der Produktkategorien-Tabelle in der Webshop-Datenbank bestehen. Dazu müssen die Tabellen über das Feld "produktkategorie_id" verknüpft werden.

- Vorsicht! Das Befüllen des Feldes "produktkategorie_id" ist in beiden Tabellen zwingend erforderlich, damit sie miteinander verknüpft werden können.

In der Produktkategorien-Tabelle wurde das Feld bereits als Primärschlüssel eingestellt. Um die Beziehung nun herzustellen, muss das Feld in der Produkte-Tabelle als Fremdschlüssel eingestellt werden (Abb. 89).

```
ALTER TABLE `tbl_produkte`  
ADD FOREIGN KEY (`produktkategorie_id`)  
REFERENCES `tbl_produktkategorien` (`produktkategorie_id`);
```



Abb. 89: Produkte und Produktkategorien verknüpfen

Mit dem SQL-Befehl "ALTER TABLE `tbl_produkte`" wird festgelegt, dass die Änderungen in der Produkte-Tabelle vorgenommen werden. Der SQL-Befehl "ADD FOREIGN KEY `produktkategorie_id`" stellt das Feld "produktkategorie_id" als Fremdschlüssel ein. Mit dem SQL-Befehl "REFERENCES..." wird festgelegt, dass der Fremdschlüssel auf die Produktkategorien-Tabelle verweist. Außerdem wird festgelegt, dass innerhalb der Produktkategorien-Tabelle auf das Feld "produktkategorie_id" verwiesen wird.

9.2.7 Übung: Produkte und Produktkategorien verknüpfen

Verknüpfen Sie die Produkte- und die Produktkategorien-Tabelle über ein geeignetes Feld miteinander. Verwenden Sie einen SQL-Befehl zum Verknüpfen der beiden Tabellen. Wenn Sie Probleme haben, die Übungsaufgabe zu lösen, können Sie im WBT zurückblättern und sich dort die Lösungen anschauen.

9.2.8 Grundlage zur Übung

Die Webshop-Datenbank der Lemonline AG dient als Grundlage für die Übung auf der nächsten WBT-Seite (Abb. 90).

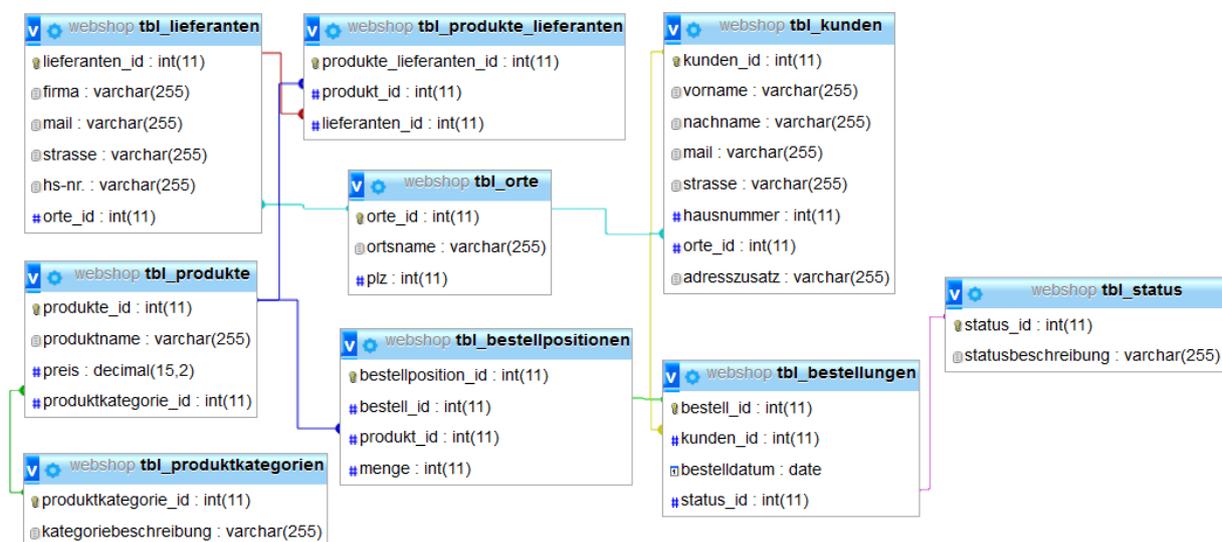


Abb. 90: Webshop-Datenbank der Lemonline AG

9.2.9 Übung: Anpassen der Webshop-Datenbank

Bitte beantworten Sie die folgenden Fragen durch Ankreuzen der korrekten Antworten. Bei einigen Fragen können auch mehrere Antworten richtig sein. Als Grundlage für die Testfragen dient die Webshop-Datenbank der Lemonline AG.

Nr.	Frage	Richtig	Falsch
1	Sie möchten den Datentyp des Feldes "hausnummer" in der Kunden-Tabelle auf VARCHAR (255) ändern. Welche Antwort ist zutreffend?		
	ALTER TABLE `tbl_kunden` CHANGE `hausnummer` `hausnummer` VARCHAR(255);		
	ALTER TABLE `tbl_kunden` CHANGE `hausnummer` VARCHAR(255);		

	Bei dieser Änderung des Datentyps gehen alle Daten des betroffenen Feldes verloren.		
2	Sie möchten ein zusätzliches Feld für die gewünschte Zahlungsart in der Kunden-Tabelle erstellen. Das Feld soll den Datentyp VARCHAR (255) erhalten. Welche Antwort liefert das richtige Ergebnis?		
	ALTER TABLE `tbl_kunden` ADD `zahlungsart` VARCHAR (255);		
	ALTER `tbl_kunden` ADD `zahlungsart` VARCHAR (255);		
3	Sie möchten den Namen des Feldes "mail" in der Lieferanten-Tabelle auf "email" ändern. Welche Antwort liefert das richtige Ergebnis?		
	ALTER TABLE `tbl_lieferanten` CHANGE `mail` `email`;		
	ALTER TABLE `tbl_lieferanten` CHANGE `mail` `email` VARCHAR(255);		
4	Sie möchten eine Tabelle für die verschiedenen Zahlungsarten anlegen. Diese soll die Felder "zahlungsart_id" und "zahlungsartbeschreibung" erhalten. Welche Antwort liefert ein sinnvolles Ergebnis?		
	CREATE TABLE tbl_zahlungsart (zahlungsart_id INT (11) AUTO_INCREMENT NULL PRIMARY KEY, zahlungsartbeschreibung VARCHAR (255) NOT NULL);		
	CREATE TABLE tbl_zahlungsart (zahlungsart_id INT (11) AUTO_INCREMENT NOT NULL PRIMARY KEY, zahlungsartbeschreibung VARCHAR (255) NOT NULL);		
	CREATE tbl_zahlungsart (zahlungsart_id INT (11) AUTO_INCREMENT NOT NULL PRIMARY KEY, zahlungsartbeschreibung VARCHAR (255) NOT NULL);		

Tab. 5: Übungsfragen WBT 08

9.2.10 Zusammenfassung und Ausblick

In diesem WBT haben wir mit Hilfe von SQL-Befehlen die Struktur der Webshop-Datenbank der Lemonline AG angepasst.

Im nächsten WBT werden wir uns anschauen, wie Webshop-Daten aus mehreren Tabellen gleichzeitig abgefragt werden können. Dazu lernen Sie die SQL-Befehle "INNER JOIN" und "LEFT JOIN" kennen.

10 Abfragen komplexer Webshop-Daten

10.1 Einleitung

Im letzten WBT haben wir einige Änderungen in der Struktur der Webshop-Datenbank der Lemonline AG vorgenommen. Dazu haben Sie die SQL-Befehle "CREATE" und "ALTER" kennengelernt.

Die Lemonline AG hat die Webshop-Datenbank mittlerweile in Betrieb genommen und vertreibt ihre Produkte nun auch über ihren Webshop. Nun benötigt die Lemonline AG einige Webshop-Daten, die aus mehreren Tabellen gleichzeitig abgefragt werden müssen. Bevor ich Ihnen zeige wie das möglich ist, müssen Sie sich jedoch zunächst den aktuellen Stand der Datenbank mit phpMyAdmin in MySQL importieren. Los gehts!

10.2 Übung: Datenbank importieren

Die Lemonline AG nutzt die Webshop-Datenbank seit einiger Zeit und vertreibt einige ihrer Produkte über den Webshop. Dadurch sind nun neue Daten in der Webshop-Datenbank enthalten. Wir müssen uns daher den aktuellen Stand der Webshop-Datenbank in MySQL importieren.

Mit Klick auf das Diskettensymbol können Sie sich die aktuelle Webshop-Datenbank herunterladen. Bitte speichern Sie die Datei auf ihrem PC und importieren Sie diese Datei anschließend mit phpMyAdmin in MySQL. Im Video sehen Sie, wie Sie dabei vorgehen müssen.

Im WBT können Sie sich an dieser Stelle ein Video anschauen, was Ihnen die Vorgehensweise zum Importieren einer Datenbank mit phpMyAdmin in MySQL verdeutlicht.

10.3 Sommerangebot

Leo Lemon möchte ein Sommerangebot per Post an alle Kunden des Webshops versenden. Daher wird eine Liste mit den Anschriften aller Kunden benötigt. Die Daten der Kunden sind auf zwei Tabellen in der Webshop-Datenbank aufgeteilt: die Kunden-Tabelle und die Orte-Tabelle. Um Daten aus der Kunden- und der Orte-Tabelle gleichzeitig abzufragen wird der SQL-Befehl "INNER JOIN" verwendet.

Mit dem SQL-Befehl "INNER JOIN" wird eine Schnittmenge der Kunden- und der Orte-Tabelle ausgegeben (Abb. 91). Nicht ausgegeben werden:

- Kunden, die bisher keinen Wohnort angegeben haben (gelb)
- Orte, in denen kein Kunde wohnt (magenta)

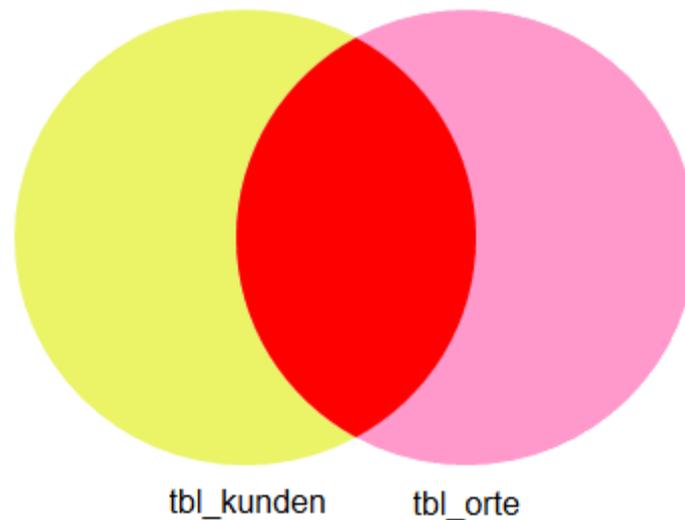


Abb. 91: Schnittmenge der Kunden- und Orte-Tabelle

10.4 Sommerangebot: Kunden- und Ortsdaten abfragen

Der SQL-Befehl "INNER JOIN" wird verwendet, um eine Schnittmenge aus zwei Tabellen abzufragen (Abb. 92).



Abb. 92: Orte- und Kunden-Tabelle

```
SELECT *
FROM `tbl_kunden`
INNER JOIN `tbl_orte`
ON tbl_kunden.orte_id = tbl_orte.orte_id;
```

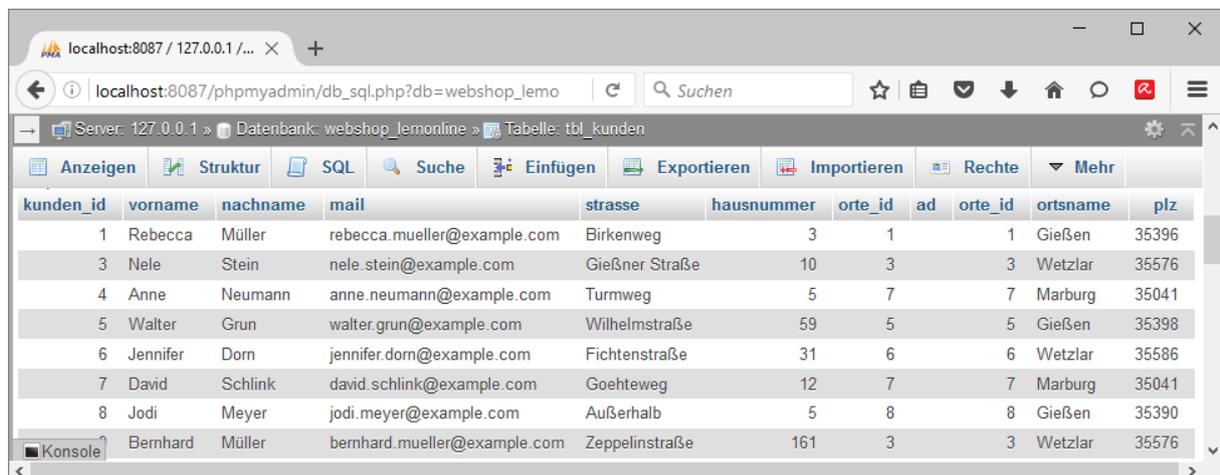
Mit den SQL-Befehlen "SELECT *" und "FROM `tbl_kunden`" wird festgelegt, dass alle Informationen aus der Kunden-Tabelle selektiert werden. Durch den SQL-Befehl "INNER

JOIN `tbl_orte`" wird festgelegt, dass die Abfrage um Informationen aus der Orte-Tabelle ergänzt wird.

Mit dem SQL-Befehl "ON tbl_kunden.orte_id = tbl_orte.orte_id" wird festgelegt, dass alle Datensätze aus der Kunden-Tabelle und der Orte-Tabelle ausgegeben werden, die den gleichen Wert im Feld "orte_id" haben. Die Abfrage ergibt ein Abfrageergebnis mit allen Daten aus der Kunden- und der Orte-Tabelle.

10.5 Sommerangebot: Abfrageergebnis

Der SQL-Befehl der vorherigen WBT-Seite generiert das folgende Abfrageergebnis (Abb. 93). Durch den SQL-Befehl "SELECT *" werden alle Felder der Kunden-Tabelle und der Orte-Tabelle ausgegeben. Das Feld "orte_id" ist in beiden Tabellen vorhanden, daher wird dieses Feld doppelt ausgegeben.



kunden_id	vorname	nachname	mail	strasse	hausnummer	orte_id	ad	orte_id	ortsname	plz
1	Rebecca	Müller	rebecca.mueller@example.com	Birkenweg	3	1		1	Gießen	35396
3	Nele	Stein	nele.stein@example.com	Gießner Straße	10	3		3	Wetzlar	35576
4	Anne	Neumann	anne.neumann@example.com	Turmweg	5	7		7	Marburg	35041
5	Walter	Grun	walter.grun@example.com	Wilhelmstraße	59	5		5	Gießen	35398
6	Jennifer	Dorn	jennifer.dorn@example.com	Fichtenstraße	31	6		6	Wetzlar	35586
7	David	Schlink	david.schlink@example.com	Goeteweg	12	7		7	Marburg	35041
8	Jodi	Meyer	jodi.meyer@example.com	Außerhalb	5	8		8	Gießen	35390
	Bernhard	Müller	bernhard.mueller@example.com	Zeppelinstraße	161	3		3	Wetzlar	35576

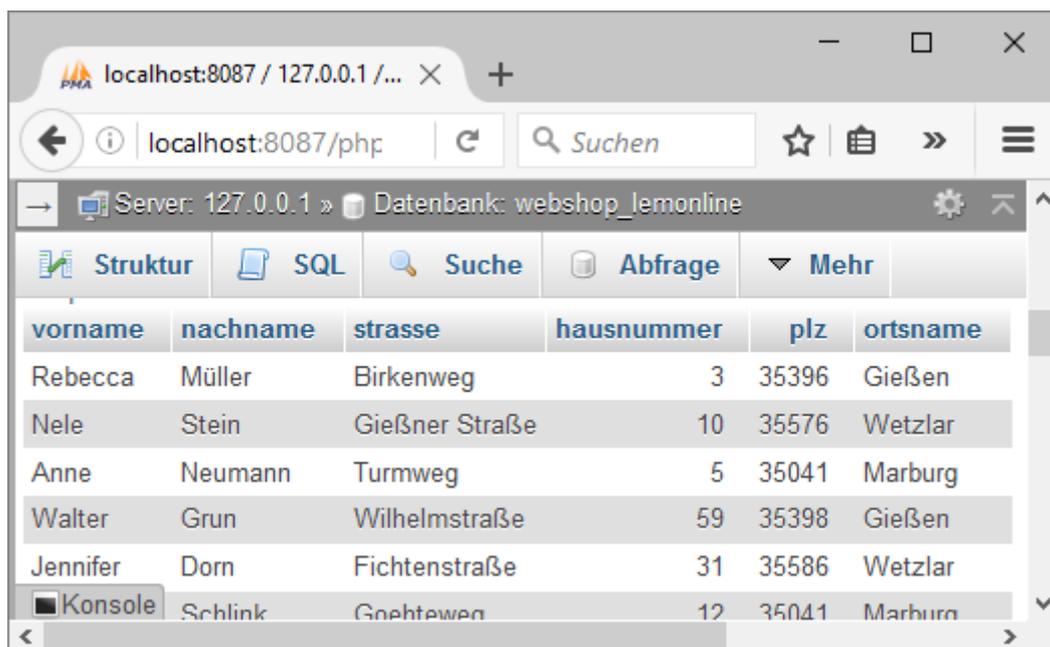
Abb. 93: Ergebnis der Abfrage

10.6 Sommerangebot: Adressliste erstellen

Um das Sommerangebot zu versenden, benötigt die Lemonline AG nur die Adressdaten und nicht das gesamte Abfrageergebnis der vorherigen WBT-Seite.

```
SELECT
tbl_kunden.vorname,
tbl_kunden.nachname,
tbl_kunden.strasse,
tbl_kunden.hausnummer,
tbl_orte.plz,
tbl_orte.ortsname
FROM `tbl_kunden`
INNER JOIN `tbl_orte`
ON tbl_kunden.orte_id = tbl_orte.orte_id;
```

Durch den SQL-Befehl "SELECT tbl_kunden.vorname, tbl_kunden.nachname, tbl_kunden.strasse, tbl_kunden.hausnummer, tbl_orte.plz, tbl_orte.ortsname" wird festgelegt, das nur bestimmte Daten aus der Kunden-Tabelle und aus der Orte-Tabelle abgefragt werden (Abb. 94). Damit das DBMS eindeutig zuordnen kann, aus welcher Tabelle die Daten abgefragt werden sollen, wird dabei immer der Tabellename vor dem Feld angegeben.



vorname	nachname	strasse	hausnummer	plz	ortsname
Rebecca	Müller	Birkenweg	3	35396	Gießen
Nele	Stein	Gießner Straße	10	35576	Wetzlar
Anne	Neumann	Turmweg	5	35041	Marburg
Walter	Grun	Wilhelmstraße	59	35398	Gießen
Jennifer	Dorn	Fichtenstraße	31	35586	Wetzlar
Konsole	Schlink	Goeteweg	12	35041	Marburg

Abb. 94: Ergebnis der Abfrage

10.7 Bestellübersicht erstellen I

Leo Lemon möchte eine Übersicht über die Bestellungen seiner Webshop-Kunden bekommen. Die Übersicht soll auch Kunden umfassen, die sich bisher nur registriert haben, aber noch keine Bestellungen getätigt haben. Daher muss hier der SQL-Befehl "LEFT JOIN" verwendet werden. Die benötigten Daten sind dabei auf zwei Tabellen in der Webshop-Datenbank aufgeteilt: die Kunden-Tabelle und die Bestellungen-Tabelle.

Mit dem SQL-Befehl "LEFT JOIN" wird zum einen die Schnittmenge der Kunden- und der Bestellungen-Tabelle ausgegeben (Abb. 95).

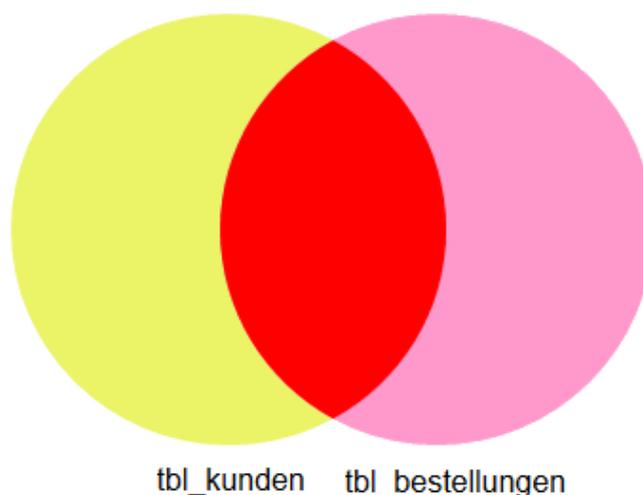


Abb. 95: Schnittmenge der Kunden- und Bestellungen-Tabelle

Im Gegensatz zum SQL-Befehl "INNER JOIN" werden jedoch zusätzlich zur Schnittmenge noch alle Datensätze aus der Kunden-Tabelle ausgegeben (Abb. 96).

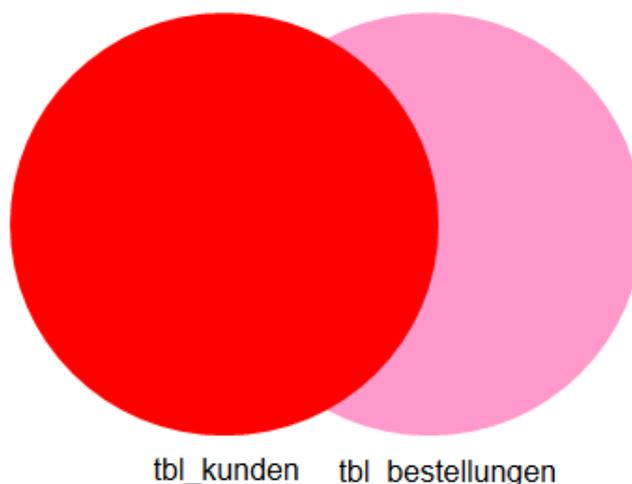
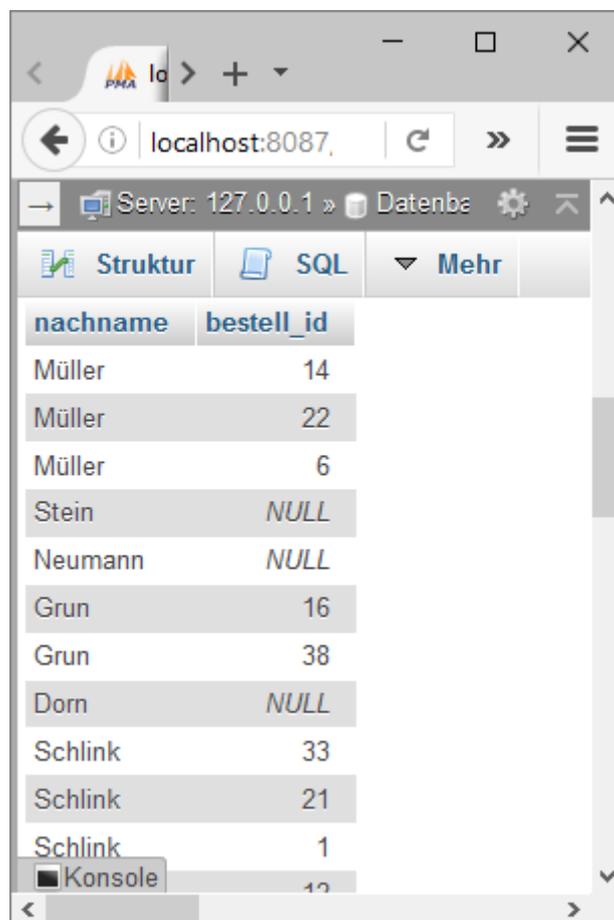


Abb. 96: Ergebnisausgabe bei der LEFT JOIN-Abfrage

VORSICHT! Bei der LEFT JOIN-Abfrage werden nicht nur Daten aus der Kunden-Tabelle ausgegeben, sondern zusätzlich Daten aus der Bestellungen-Tabelle. Die reine Abfrage von Kundendaten würde ein anderes Abfrageergebnis erzeugen, in dem keine Bestelldaten vorhanden wären.

10.8 Bestellübersicht erstellen II

Der SQL-Befehl "LEFT JOIN" wird verwendet, um eine Übersicht aller Kunden und deren Bestelldaten zu erstellen. Es werden jedoch nicht alle Daten aus den beiden Tabellen benötigt (Abb. 97).



The screenshot shows a web browser window displaying a SQL query result. The browser address bar shows 'localhost:8087'. The page title is 'Server: 127.0.0.1 » Datenba'. The query result is displayed in a table with two columns: 'nachname' and 'bestell_id'. The table contains the following data:

nachname	bestell_id
Müller	14
Müller	22
Müller	6
Stein	NULL
Neumann	NULL
Grun	16
Grun	38
Dorn	NULL
Schlink	33
Schlink	21
Schlink	1
Konsole	12

Abb. 97: Ergebnis der Abfrage

```
SELECT
tbl_kunden.nachname,
tbl_bestellungen.bestell_id

FROM `tbl_kunden`

LEFT JOIN `tbl_bestellungen`

ON tbl_kunden.kunden_id =
    tbl_bestellungen.kunden_id;
```

Mit den SQL-Befehlen "SELECT...", "FROM `tbl_kunden`" wird festgelegt, dass der Kundenname aus der Kunden-Tabelle selektiert wird. Durch den SQL-Befehl "LEFT JOIN `tbl_bestellungen`" wird festgelegt, dass die Abfrage um Informationen aus der Bestellungen-Tabelle erweitert wird. Wenn den Kunden keine Bestellungen zugeordnet sind, wird der Wert "NULL" nach dem Kundennamen ausgegeben.

- Der Wert "NULL" wird angezeigt, wenn ein Feld in einem Datensatz leer ist bzw. noch nie ein Wert eingetragen wurde.

Mit dem SQL-Befehl "ON tbl_kunden.kunden_id = tbl_bestellungen.kunden_id" wird festgelegt, dass alle Datensätze aus der Kunden-Tabelle und der Bestellungen-Tabelle ausgegeben werden, die den gleichen Wert im Feld "kunden_id" haben.

10.9 Marketingmaßnahme planen

Leo Lemon ist aufgefallen, dass sich einige Personen zwar im Webshop registriert haben, aber noch keine Bestellung aufgegeben haben. Diesen Personen möchte er ein Angebot per Mail senden. Daher benötigt er eine Liste mit den E-Mail-Adressen aller Personen, die noch keine Bestellung aufgegeben haben (Abb. 98).

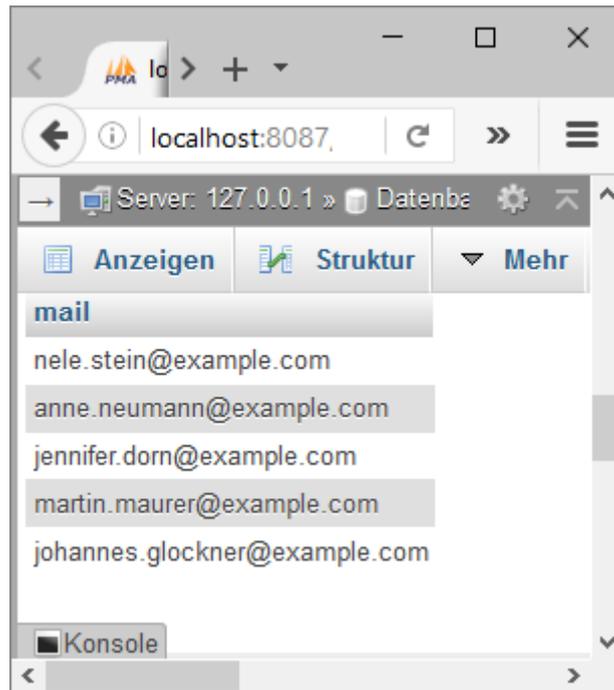


Abb. 98: Ergebnis der Abfrage

```
SELECT tbl_kunden.mail
FROM `tbl_kunden`
LEFT JOIN `tbl_bestellungen`
ON tbl_kunden.kunden_id = tbl_bestellungen.kunden_id

WHERE bestell_id IS NULL;
```

Um diese Liste zu erstellen, wird der "LEFT JOIN"-Befehl um eine WHERE-Bedingung ergänzt. Es sollen nur E-Mail-Adressen von Kunden ausgegeben werden, bei denen keine Bestellungen hinterlegt wurden.

10.10 Grundlage zur Übung

Die Webshop-Datenbank der Lemonline AG dient als Grundlage für die Übung auf der nächsten WBT-Seite (Abb. 99).

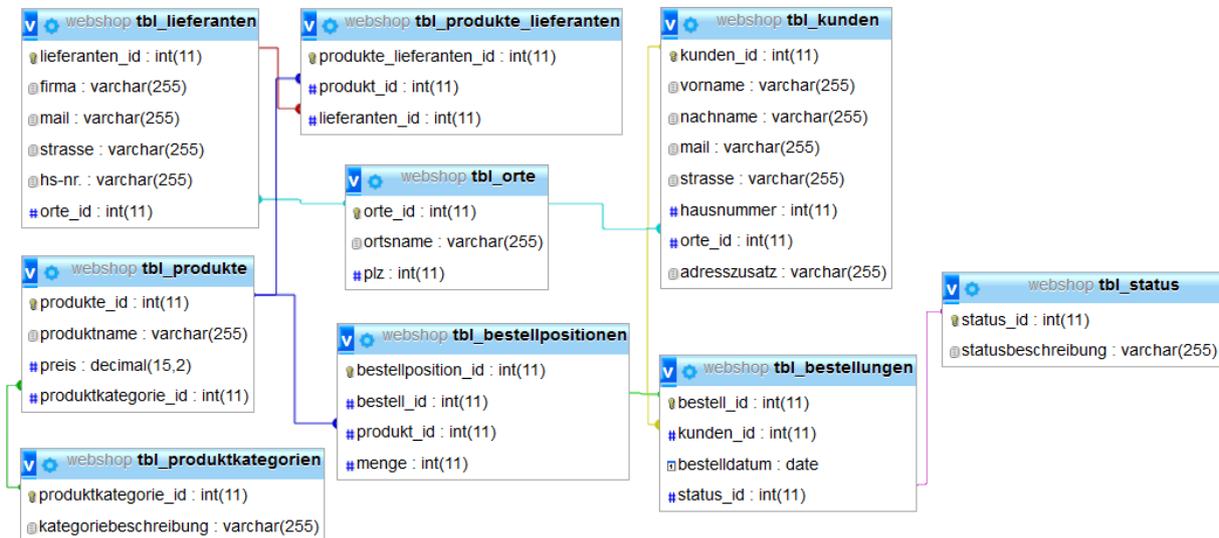


Abb. 99: Webshop-Datenbank der Lemonline AG

10.11 Übung: Komplexe Abfragen

Bitte beantworten Sie die folgenden Fragen durch Ankreuzen der korrekten Antworten. Bei einigen Fragen können auch mehrere Antworten richtig sein. Als Grundlage für die Testfragen dient die Webshop-Datenbank der Lemonline AG.

Nr.	Frage	Richtig	Falsch
1	Sie möchten eine Liste mit allen Bestell-IDs und deren Statusbeschreibungen erstellen. Dabei sollen auch Bestellungen ohne Status aufgelistet werden. Welche Antwort ist zutreffend?		
	<pre> SELECT tbl_bestellungen.bestell_id, tbl_status.statusbeschreibung FROM `tbl_bestellungen` INNER JOIN `tbl_status` ON tbl_bestellungen.status_id = tbl_status.status_id; </pre>		
	<pre> SELECT tbl_bestellungen.bestell_id, tbl_status.statusbeschreibung FROM `tbl_bestellungen` LEFT JOIN `tbl_status` ON tbl_bestellungen.status_id = tbl_status.status_id; </pre>		

	keine der Antworten ist richtig		
2	Sie möchten ein Angebot an alle Kunden versenden, die bereits über den Webshop bestellt haben. Daher benötigen Sie alle E-Mail-Adressen dieser Kunden. Welche Antwort ist korrekt?		
	SELECT tbl_kunden.mail FROM `tbl_kunden` LEFT JOIN `tbl_bestellungen` ON tbl_kunden.kunden_id = tbl_bestellungen.kunden_id;		
	SELECT tbl_kunden.mail FROM `tbl_kunden` INNER JOIN `tbl_bestellungen` ON tbl.kunden_kunden_id = tbl.bestellungen_kunden_id;		
	keine der Antworten ist richtig		
3	Sie möchten kontrollieren, ob alle Produkte einer Produktkategorie zugeordnet wurden und ob diese Kategorie korrekt ist. Daher benötigen Sie eine Liste mit allen Produktnamen und deren zugeordneter Produktkategoriebeschreibung. Welche Antwort ist korrekt?		
	SELECT tbl_produkte.produktname, tbl_produktkategorien.kategoriebeschreibung FROM `tbl_produktkategorien` LEFT JOIN `tbl_produkte` ON tbl_produkte.produktkategorie_id = tbl_produktkategorien.produktkategorie_id;		
	SELECT tbl_produkte.produktname, tbl_produktkategorien.kategoriebeschreibung FROM `tbl_produkte` LEFT JOIN `tbl_produktkategorien` ON tbl_produkte.produktkategorie_id = tbl_produktkategorien.produktkategorie_id;		
	keine der Antworten ist richtig		
4	Sie möchten ihre Lieferanten per Post über den neuen Webshop informieren. Dazu benötigen Sie die Adressen ihrer Lieferanten. Welche Antwort ist korrekt?		
	SELECT tbl_lieferanten.firma, tbl_lieferanten.strasse, tbl_lieferanten.hs-nr., tbl_orte.ortsname, tbl_orte.plz FROM `tbl_orte` INNER JOIN `tbl_lieferanten` ON tbl_orte.orte_id = tbl_lieferanten.orte_id;		
	SELECT tbl_lieferanten.firma, tbl_lieferanten.strasse, tbl_lieferanten.hs-nr., tbl_orte.ortsname, tbl_orte.plz FROM `tbl_lieferanten` INNER JOIN `tbl_orte` ON tbl_lieferanten.orte_id = tbl_orte.orte_id;		
	keine der Antworten ist richtig		
5	Sie möchten eine Übersicht erhalten, wie schnell die Bestellungen abgewickelt werden. Dazu benötigen Sie eine		

	Liste mit den Bestell-IDs, dem Bestelldatum und der Statusbeschreibung. Die Liste soll nur Bestellungen enthalten, bei denen ein Bestellstatus hinterlegt wurde. Welche Antwort ist korrekt?		
	<pre>SELECT tbl_bestellungen.bestell_id, tbl_bestellungen.bestelldatum, tbl_status.statusbeschreibung FROM `tbl_bestellungen` LEFT JOIN `tbl_status` ON tbl_bestellungen.status_id = tbl_status.status_id;</pre>		
	<pre>SELECT tbl_bestellungen.bestell_id, tbl_bestellungen.bestelldatum, tbl_status.statusbeschreibung FROM `tbl_status` LEFT JOIN `tbl_bestellungen` ON tbl_bestellungen.status_id = tbl_status.status_id;</pre>		
	keine der Antworten ist richtig		

Tab. 6: Übungsfragen WBT 09

10.12 Zusammenfassung und Ausblick

In diesem WBT haben wir Webshop-Daten aus mehreren Tabellen der Webshop-Datenbank gleichzeitig abgefragt. Dazu haben Sie die SQL-Befehle "INNER JOIN" und "LEFT JOIN" kennengelernt.

Im nächsten WBT werden wir uns anschauen, wie eine Webshop-Datenbank mit Hilfe von Indizes optimiert werden kann.

11 Optimieren der Webshop-Datenbank

11.1 Einleitung

Im letzten WBT haben wir Daten aus mehreren Tabellen der Webshop-Datenbank gleichzeitig abgefragt. Dazu haben Sie die SQL-Befehle "INNER JOIN" und "LEFT JOIN" kennengelernt.

Die Lemonline AG hat mittlerweile 100.000 Kunden. Dadurch benötigen Abfragen in der Webshop-Datenbank viel Zeit. In diesem WBT werden wir uns ansehen, wie wir dieses Problem beheben und somit die Datenbank optimieren können. Bevor ich Ihnen zeige, wie das möglich ist, müssen Sie sich jedoch zunächst den aktuellen Stand der Datenbank mit phpMyAdmin in MySQL importieren. Los gehts!

11.2 Übung: Datenbank importieren

Die Lemonline AG hat mittlerweile 100.000 Kunden. Dadurch sind nun neue Daten in der Webshop-Datenbank enthalten. Wir müssen uns daher den aktuellen Stand der Webshop-Datenbank mit phpMyAdmin in MySQL importieren. Mit Klick auf das Diskettensymbol können Sie sich die aktuelle Webshop-Datenbank herunterladen. Bitte speichern Sie die Datei auf Ihrem PC.

Da mittlerweile 100.000 Kunden in der aktuellen Webshop-Datenbank enthalten sind, ist die SQL-Datei der Datenbank ziemlich groß. Um eine so große SQL-Datei mit phpMyAdmin in MySQL importieren zu können, muss daher zunächst eine kleine Anpassung an phpMyAdmin vorgenommen werden. Bitte stoppen Sie den Apache- und den MySQL-Server, bevor Sie die Anpassung vornehmen. Im Video sehen Sie, wie Sie bei der Anpassung und dem Import der Datenbank vorgehen müssen.

Im WBT können Sie sich an dieser Stelle ein Video anschauen, was Ihnen die Vorgehensweise zum Importieren einer Datenbank mit phpMyAdmin in MySQL verdeutlicht.

11.3 Problem: Kundenanfragen

Die Lemonline AG hat mittlerweile 100.000 Kunden. Diese können sich bei Problemen mit dem Webshop telefonisch an den Kundenservice der Lemonline AG wenden. Viele Kunden haben jedoch ihre Kundennummer nicht parat. Daher müssen sie von den Mitarbeitern der Lemonline AG über den Namen in der Webshop-Datenbank gesucht werden. Durch die hohe Kundenanzahl benötigen diese Abfragen in der Webshop-Datenbank jedoch viel Zeit.

Aus der Kunden-Tabelle soll der Eintrag für einen Kunden mit dem Nachnamen "Peters" über das Feld "nachname" herausgefiltert werden. Dies bedeutet für das DBMS, dass es den ge-

suchten Nachnamen "Peters" mit allen Nachnamen in der Kunden-Tabelle vergleichen muss (Abb. 100). Der gesuchte Nachname kann mehrmals in der Kunden-Tabelle vorkommen.



nachname
Murray
Müller
Peters
Meyer
Feldhof
Kast
Schmidt
Dixon
Wilson
Hill
...
Peters

Abb. 100: Vorgehensweise des DBMS

11.4 Optimieren der Kunden-Tabelle

Um Kunden in der Kunden-Tabelle schneller über ihren Nachnamen zu finden, kann ein Index für das Feld "nachname" in der Kunden-Tabelle angelegt werden.

- Ein Index ist vergleichbar mit dem Schlagwortverzeichnis eines Buchs. Anhand der Seitenzahl, die zu jedem Schlagwort vermerkt ist, kann das gesuchte Thema schnell im Buch gefunden werden.

Dies bedeutet, dass MySQL alle Datensätze der Kunden-Tabelle in alphabetischer Reihenfolge der Nachnamen abspeichert. Dabei wird eine neue Datei angelegt, in der alle Nachnamen und ein Verweis auf den zugehörigen Datensatz in der Kunden-Tabelle gespeichert werden (Abb. 101).

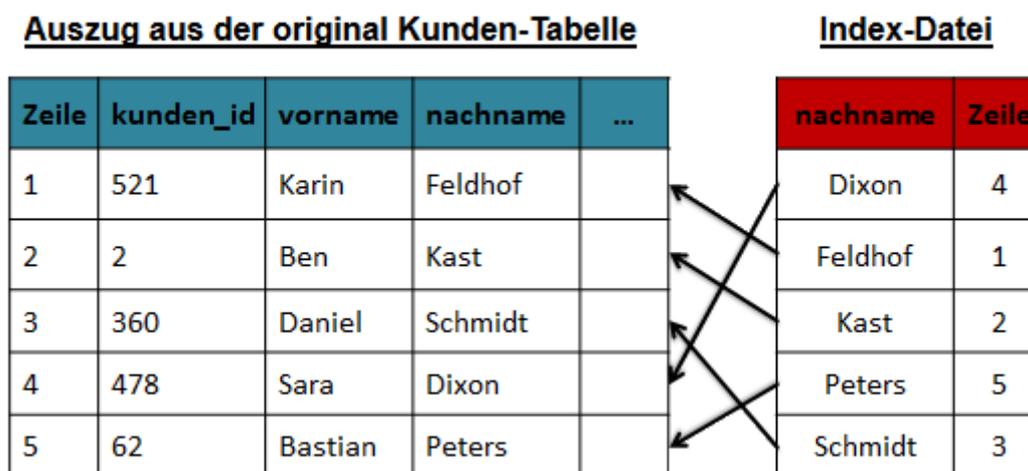


Abb. 101: Zusammenhang Kunden-Tabelle und Index

11.5 Vor- und Nachteile eines Indexes

Vor dem Anlegen eines Indexes sollten dessen Vor- und Nachteile gut überdacht werden. Ein Index führt bei Feldern, die regelmäßig in WHERE- oder ORDER BY-Befehlen verwendet werden, zu schnelleren Suchergebnissen. In der Kunden-Tabelle ist daher ein Index für das Feld "nachname" sinnvoll.

Der Index benötigt jedoch zusätzlichen Speicherplatz und verlangsamt zudem Schreibbefehle (z. B. INSERT). Da in der Kunden-Tabelle jedoch mehr Lesebefehle (z. B. SELECT) als Schreibbefehle ausgeführt werden, nimmt die Lemonline AG diese Nachteile in Kauf.

Im Video sehen Sie, wie Sie bei der Erstellung eines Indexes mit phpMyAdmin vorgehen müssen.

Im WBT können Sie sich an dieser Stelle ein Video anschauen, was Ihnen die Vorgehensweise zur Erstellung eines Indexes mit phpMyAdmin verdeutlicht.

11.6 Übung: Optimieren der Webshop-Datenbank

Führen Sie die Aufgabenstellung bitte in der vorgegebenen Reihenfolge durch und achten Sie dabei auf die Ausführungszeiten der beiden SELECT-Befehle.

- Suchen Sie in der Kunden-Tabelle nach der Kundin "Anne Krause" und notieren Sie sich die Ausführungszeit des SQL-Befehls. Verwenden Sie dazu den folgenden SQL-Befehl: `SELECT * FROM `tbl_kunden` WHERE `nachname`='Krause';`
- Legen Sie einen Index für das Feld "nachname" in der Tabelle "tbl_kunden" an.
- Suchen Sie erneut in der Kunden-Tabelle nach der Kundin "Anne Krause" und notieren Sie sich die Ausführungszeit des SQL-Befehls. Verwenden Sie dazu den gleichen SQL-Befehl wie in Teilaufgabe 1.

Obwohl zweimal der gleiche SELECT-Befehl ausgeführt wird, liefert das DBMS auf einem Beispielrechner das Abfrageergebnis in einer unterschiedlichen Geschwindigkeit:

- ohne Index: ca. 0,1551 Sekunden
- mit Index: ca. 0,0115 Sekunden

Der zeitliche Unterschied erscheint zwar marginal, ist jedoch gravierend: Mit dem Index wird das Abfrageergebnis mehr als zehn Mal schneller geliefert als ohne den Index.

11.7 Zusammenfassung und Ausblick

In diesem WBT haben wir die Webshop-Datenbank der Lemonline AG optimiert. Dazu haben Sie Indizes kennengelernt.

Im Rahmen der WBT-Serie "Relationale Datenbanken und MySQL" haben Sie einen umfassenden Einblick in die Thematik erhalten.

Anhang

Lösung zur Übung – WBT 06

Nr.	Frage	Richtig	Falsch
1	Sie möchten herausfinden, welche Lieferanten ihren Firmensitz in Grünberg (orte_id: 9) oder Bad Nauheim (orte_id: 10) haben. Welche Antwort liefert das richtige Ergebnis?		
	SELECT `firma` FROM `tbl_lieferanten` WHERE `orte_id`=9 OR `orte_id`=10;	x	
	SELECT `firma` FROM `tbl_lieferanten` WHERE `orte_id`=9 AND `orte_id`=10;		x
2	Sie möchten alle Daten der Tabelle "tbl_orte" nach den Postleitzahlen aufsteigend ordnen. Welche Antwort liefert das richtige Ergebnis?		
	SELECT * FROM `tbl_orte` ORDER BY `plz` ASC;	x	
	SELECT `orte_id`, `ortsname`, `plz` FROM `tbl_orte` ORDER BY `plz` ASC;	x	
	SELECT * FROM `tbl_orte` ORDER BY `plz` DESC;		x
3	Sie möchten die E-Mailadressen der Kundin Rebecca Müller wissen. Welche Antwort liefert das richtige Ergebnis?		
	SELECT `mail` FROM `tbl_kunden` WHERE `nachname` LIKE 'Müller';		x
	SELECT `mail` FROM `tbl_kunden` WHERE `nachname` LIKE 'Müller' AND `vorname` LIKE 'Rebecca';	x	
4	Sie möchten wissen, in welchem Status sich Bestellungen der Lemonline momentan befinden. Dabei möchten Sie jede Statusbeschreibung nur einmal angezeigt bekommen. Welche Antwort liefert das richtige Ergebnis?		
	SELECT DISTINCT * FROM `tbl_status`;		x
	SELECT DISTINCT `statusbeschreibung` FROM `tbl_status`;	x	
5	Die Lemonline AG plant eine weitere Rabattaktion. Alle Produkte, die billiger als 100€ sind, erhalten 20% Rabatt. Sie sollen eine Liste erstellen mit allen betroffenen Produkten. Die Liste soll aufsteigend nach dem Preis sortiert sein. Welche Antworten liefern das richtige Ergebnis?		
	SELECT * FROM `tbl_produkte` WHERE `preis` < 100 ORDER BY ASC;		x
	SELECT * FROM `tbl_produkte` WHERE `preis` < 100 ORDER BY `preis` ASC;	x	

	SELECT * FROM `tbl_produkte` WHERE `preis` LIKE 100 ORDER BY `preis` ASC;		x
6	Sie möchten eine Liste aller Lieferanten aus Gießen (orte_id: 1, 2, 4, 5, 8) erstellen. Welche Antwort liefert das richtige Ergebnis?		
	SELECT * FROM `tbl_lieferanten` WHERE `orte_id`=1 OR `orte_id`=2 OR `orte_id`=4 OR `orte_id`=5 OR `orte_id`=8;	x	
	SELECT * FROM `tbl_lieferanten` WHERE `orte_id`=1 AND `orte_id`=2 AND `orte_id`=4 AND `orte_id`=5 AND `orte_id`=8;		x

Tab. 7: Lösung zur Übung - WBT 06

Lösung zur Übung – WBT 07

Nr.	Frage	Richtig	Falsch
1	Sie möchten die E-Mail-Adresse eines Lieferanten (lieferanten_id: 9) ändern. Welche Antwort liefert das richtige Ergebnis?		
	UPDATE `tbl_lieferanten` SET `mail` = 'xy-ag@info.de';		x
	UPDATE `tbl_lieferanten` SET `mail` = 'xy-ag@info.de' WHERE `lieferanten_id` = 9;	x	
2	Sie aktualisieren die Liste der Lieferanten. Dabei sollen alle Lieferanten, welche die Lemonline AG seit 3 Jahren nicht mehr beliefert haben (lieferanten_id: 5,15, 18,21) gelöscht werden. Welche Antwort liefert das richtige Ergebnis?		
	DELETE FROM `tbl_lieferanten` WHERE `lieferanten_id` = 5 OR `lieferanten_id` = 15 OR `lieferanten_id` = 18 OR `lieferanten_id` = 21;	x	
	DELETE FROM `tbl_lieferanten` WHERE `lieferanten_id` = 5 AND `lieferanten_id` = 15 AND `lieferanten_id` = 18 AND `lieferanten_id` = 21;		x
	DELETE FROM `tbl_lieferanten` WHERE `lieferanten_id` IN (5, 15, 18, 21);	x	
3	Alle Bestellvorgänge wurden abgeschlossen. Sie möchten den Bestellstatus aller Bestellungen auf "geschlossen" (status_id: 2) ändern. Welche Antwort liefert das richtige Ergebnis?		
	UPDATE `tbl_status` SET `status_id` = 2;	x	
	UPDATE `tbl_status` SET `status_id` = '2';		x
4	Sie möchten die Adresse von Frau Jennifer Dorn (kunden_id: 6) aktualisieren. Frau Dorn ist innerhalb ihres Wohnortes in den Tannenweg 9 umgezogen. Welche		

	Antwort liefert das richtige Ergebnis?		
	UPDATE SET `strasse` = 'Tannenweg', `hausnummer` = 9 WHERE `kunden_id` = 6;		x
	UPDATE `tbl_kunden` SET `strasse` = 'Tannenweg', `hausnummer` = 9 WHERE `kunden_id` = 6;	x	
	Keine der Antworten ist richtig		x
5	Sie möchten den Kunden Bernhard Müller (kunden_id: 8) aus der Webshop-Datenbank der Lemonline AG löschen. Welche Antwort liefert das richtige Ergebnis?		
	DELETE `kunden_id` = 8 FROM `tbl_kunden`;		x
	DELETE FROM `tbl_kunden` WHERE `kunden_id` = 8;	x	
6	Sie möchten den Preis des Produkts "Smart Tablet 5" (produkte_id: 2) aktualisieren. Welche Antwort liefert das richtige Ergebnis?		
	UPDATE `tbl_produkte` SET `preis` = 180,25 WHERE `produkte_id` = 2;		x
	UPDATE `tbl_produkte` SET preis = 180.25 WHERE `produkte_id` = 2;		x
	Keine der Antworten ist richtig	x	

Tab. 8: Lösung zur Übung - WBT 07

Lösung zur Übung – WBT 08

1. Erstellen Sie eine neue Tabelle mit dem Namen „tbl_produktkategorien“.

Nutzen Sie zur Lösung der Aufgabe den folgenden SQL-Befehl:

```
CREATE TABLE tbl_produktkategorien
(
  produktkategorie_id INT(11) AUTO_INCREMENT
  NOT NULL PRIMARY KEY,
  kategoriebeschreibung VARCHAR(255) NOT NULL
);
```

2. Befüllen Sie die Produktkategorien-Tabelle mit folgenden Beschreibungen:

- Smartphone
- Tablet
- Zubehörset

Nutzen Sie zur Lösung der Aufgabe die folgenden SQL-Befehle:

- a.

```
INSERT INTO `tbl_produktkategorien`  
SET `kategoriebeschreibung`= `Smartphone`;
```
- b.

```
INSERT INTO `tbl_produktkategorien`  
SET `kategoriebeschreibung`= `Tablet`;
```
- c.

```
INSERT INTO `tbl_produktkategorien`  
SET `kategoriebeschreibung`= `Zubehörset`;
```

- 3. In der Produkte-Tabelle soll das Feld "produktkategorie_id" erstellt werden. Das Feld soll den Datentyp INT(11) erhalten. Nehmen Sie diese Änderung in der Webshop-Datenbank vor.**

Nutzen Sie zur Lösung der Aufgabe den folgenden SQL-Befehl:

```
ALTER TABLE `tbl_produkte`  
ADD `produktkategorie_id` INT(11);
```

- 4. Ergänzen Sie in der Produkte-Tabelle zu jedem Produkt die geeignete produktkategorie_id.**

Nutzen Sie zur Lösung der Aufgabe den folgenden SQL-Befehl. Anstelle der Fragezeichen müssen die Produkte_id des entsprechenden Produkts bzw. die dazugehörige Produktkategorie_id eingesetzt werden.

```
UPDATE `tbl_produkte`  
SET `produktkategorie_id` = ?  
WHERE `produkte_id`= ?;
```

Nr.	Frage	Richtig	Falsch
1	Sie möchten den Datentyp des Feldes "hausnummer" in der Kunden-Tabelle auf VARCHAR (255) ändern. Welche Antwort ist zutreffend?		
	<pre>ALTER TABLE `tbl_kunden` CHANGE `hausnummer` `hausnummer` VARCHAR(255);</pre>	x	
	<pre>ALTER TABLE `tbl_kunden` CHANGE `hausnummer`</pre>		x

	VARCHAR(255);		
	Bei dieser Änderung des Datentyps gehen alle Daten des betroffenen Feldes verloren.	x	
2	Sie möchten ein zusätzliches Feld für die gewünschte Zahlungsart in der Kunden-Tabelle erstellen. Das Feld soll den Datentyp VARCHAR (255) erhalten. Welche Antwort liefert das richtige Ergebnis?		
	ALTER TABLE `tbl_kunden` ADD `zahlungsart` VARCHAR (255);	x	
	ALTER `tbl_kunden` ADD `zahlungsart` VARCHAR (255);		x
3	Sie möchten den Namen des Feldes "mail" in der Lieferanten-Tabelle auf "email" ändern. Welche Antwort liefert das richtige Ergebnis?		
	ALTER TABLE `tbl_lieferanten` CHANGE `mail` `email` ;		x
	ALTER TABLE `tbl_lieferanten` CHANGE `mail` `email` VARCHAR(255);	x	
4	Sie möchten eine Tabelle für die verschiedenen Zahlungsarten anlegen. Diese soll die Felder "zahlungsart_id" und "zahlungsartbeschreibung" erhalten. Welche Antwort liefert ein sinnvolles Ergebnis?		
	CREATE TABLE tbl_zahlungsart (zahlungsart_id INT (11) AUTO_INCREMENT NULL PRIMARY KEY, zahlungsartbeschreibung VARCHAR (255) NOT NULL);		x
	CREATE TABLE tbl_zahlungsart (zahlungsart_id INT (11) AUTO_INCREMENT NOT NULL PRIMARY KEY, zahlungsartbeschreibung VARCHAR (255) NOT NULL);	x	
	CREATE tbl_zahlungsart (zahlungsart_id INT (11) AUTO_INCREMENT NOT NULL PRIMARY KEY, zahlungsartbeschreibung VARCHAR (255) NOT NULL);		x

Tab. 9: Lösung zur Übung - WBT 08

Lösung zur Übung – WBT 09

Nr.	Frage	Richtig	Falsch
1	Sie möchten eine Liste mit allen Bestell-IDs und deren Statusbeschreibungen erstellen. Dabei sollen auch Bestellungen ohne Status aufgelistet werden. Welche Antwort ist zutreffend?		
	SELECT tbl_bestellungen.bestell_id, tbl_status.statusbeschreibung FROM `tbl_bestellungen`		x

	INNER JOIN `tbl_status` ON tbl_bestellungen.status_id = tbl_status.status_id;		
	SELECT tbl_bestellungen.bestell_id, tbl_status.statusbeschreibung FROM `tbl_bestellungen` LEFT JOIN `tbl_status` ON tbl_bestellungen.status_id = tbl_status.status_id;	x	
	keine der Antworten ist richtig		x
2	Sie möchten ein Angebot an alle Kunden versenden, die bereits über den Webshop bestellt haben. Daher benötigen Sie alle E-Mail-Adressen dieser Kunden. Welche Antwort ist korrekt?		
	SELECT tbl_kunden.mail FROM `tbl_kunden` LEFT JOIN `tbl_bestellungen` ON tbl_kunden.kunden_id = tbl_bestellungen.kunden_id;		x
	SELECT tbl_kunden.mail FROM `tbl_kunden` INNER JOIN `tbl_bestellungen` ON tbl.kunden_kunden_id = tbl.bestellungen_kunden_id;		x
	keine der Antworten ist richtig	x	
3	Sie möchten kontrollieren, ob alle Produkte einer Produktkategorie zugeordnet wurden und ob diese Kategorie korrekt ist. Daher benötigen Sie eine Liste mit allen Produktnamen und deren zugeordneter Produktkategoriebeschreibung. Welche Antwort ist korrekt?		
	SELECT tbl_produkte.produktname, tbl_produktkategorien.kategoriebeschreibung FROM `tbl_produktkategorien` LEFT JOIN `tbl_produkte` ON tbl_produkte.produktkategorie_id = tbl_produktkategorien.produktkategorie_id;		x
	SELECT tbl_produkte.produktname, tbl_produktkategorien.kategoriebeschreibung FROM `tbl_produkte` LEFT JOIN `tbl_produktkategorien` ON tbl_produkte.produktkategorie_id = tbl_produktkategorien.produktkategorie_id;	x	
	keine der Antworten ist richtig		x
4	Sie möchten ihre Lieferanten per Post über den neuen Webshop informieren. Dazu benötigen Sie die Adressen ihrer Lieferanten. Welche Antwort ist korrekt?		
	SELECT tbl_lieferanten.firma, tbl_lieferanten.strasse, tbl_lieferanten.hs-nr., tbl_orte.ortsname, tbl_orte.plz FROM `tbl_orte`	x	

	INNER JOIN `tbl_lieferanten` ON tbl_orte.orte_id = tbl_lieferanten.orte_id;		
	SELECT tbl_lieferanten.firma, tbl_lieferanten.strasse, tbl_lieferanten.hs-nr., tbl_orte.ortsname, tbl_orte.plz FROM `tbl_lieferanten` INNER JOIN `tbl_orte` ON tbl_lieferanten.orte_id = tbl_orte.orte_id;	x	
	keine der Antworten ist richtig		x
5	Sie möchten eine Übersicht erhalten, wie schnell die Bestellungen abgewickelt werden. Dazu benötigen Sie eine Liste mit den Bestell-IDs, dem Bestelldatum und der Statusbeschreibung. Die Liste soll nur Bestellungen enthalten, bei denen ein Bestellstatus hinterlegt wurde. Welche Antwort ist korrekt?		
	SELECT tbl_bestellungen.bestell_id, tbl_bestellungen.bestelldatum, tbl_status.statusbeschreibung FROM `tbl_bestellungen` LEFT JOIN `tbl_status` ON tbl_bestellungen.status_id = tbl_status.status_id;		x
	SELECT tbl_bestellungen.bestell_id, tbl_bestellungen.bestelldatum, tbl_status.statusbeschreibung FROM `tbl_status` LEFT JOIN `tbl_bestellungen` ON tbl_bestellungen.status_id = tbl_status.status_id;		x
	keine der Antworten ist richtig	x	

Tab. 10: Lösung zur Übung - WBT 09

Literaturverzeichnis

1. **Apache Friends (Hrsg.):** Stichwort: XAMPP, Online im Internet: <https://www.apachefriends.org/de/index.html>, 30.03.2016.
2. **Appsolute GmbH (Hrsg.):** Stichwort MAMP, Online im Internet: <https://www.mamp.info/de/>, 30.03.2016.
3. **Beighley, Lynn; Morrison, Michael:** PHP & MySQL von Kopf bis Fuß, Köln: O'Reilly Verlag 2009.
4. **Creationes Virtuales CV GmbH (Hrsg.):** Stichwort: MySQL lernen, Online im Internet: <http://lernenhoch2.de>, April 2016.
5. **Däßler, Rolf:** MySQL, Bonn: verlag moderne industrie Buch AG & Co. KG 2001.
6. **Delisle, Marc:** phpMyAdmin – MySQL-Datenbanken effizient über das Web verwalten, München: Addison-Wesley Verlag 2005.
7. **Fontecchio, Mark:** Oracle the clear leader in \$24 billion RDBMS market, Online im Internet: <http://itknowledgeexchange.techtarget.com/eye-on-oracle/oracle-the-clear-leader-in-24-billion-rdbms-market/>, 12.04.2012.
8. **Koch, Daniel:** Das große Buch: MySQL, Düsseldorf: Data Becker GmbH & Co. KG 2010.
9. **Kofler, Michael:** MySQL – Einführung, Programmierung, Referenz, 3. Aufl., München: Addison-Wesley Verlag 2001.
10. **Mockaroo, LLC (Hrsg.):** Stichwort: data generator, Online im Internet: <https://www.mockaroo.com/>, 26.07.2016.
11. **Pollakowski, Martin:** Grundkurs MySQL und PHP, 2., verb. und erw. Auflage, Braunschweig, Wiesbaden 2005.
12. **Reimers, Nils:** Stichwort: MySQL Tutorial, Online im Internet: <http://www.php-einfach.de>, 2003 – 2016.
13. **Rüttger, Michael:** MySQL für Dummies, Bonn: mitp-Verlag 2002.
14. **Widenius, Michael; Axmark, David:** MySQL – Das offizielle Handbuch, Bonn: mitp-Verlag 2002.
15. **Wikimedia Foundation Inc. (Hrsg.):** Stichwort: Logo MySQL, Online im Internet: https://de.wikipedia.org/wiki/MySQL#/media/File:Logo_MySQL.svg, 01.04.2016.
16. **Wikimedia Foundation Inc. (Hrsg.):** Stichwort: Logo XAMPP, Online im Internet: https://upload.wikimedia.org/wikipedia/commons/thumb/0/03/Xampp_logo.svg/2000px-Xampp_logo.svg.png, 02.04.2016.

Impressum



- Reihe:** **Arbeitspapiere Wirtschaftsinformatik** (ISSN 1613-6667)
- Bezug:** <https://wi.uni-giessen.de>
- Herausgeber:** Prof. Dr. Axel Schwickert
Prof. Dr. Bernhard Ostheimer
- c/o Professur BWL – Wirtschaftsinformatik
Justus-Liebig-Universität Gießen
Fachbereich Wirtschaftswissenschaften
Licher Straße 70
D – 35394 Gießen
Telefon (0 64 1) 99-22611
Telefax (0 64 1) 99-22619
eMail: Axel.Schwickert@wirtschaft.uni-giessen.de
<https://wi.uni-giessen.de>
- Ziele:** Die Arbeitspapiere dieser Reihe sollen konsistente Überblicke zu den Grundlagen der Wirtschaftsinformatik geben und sich mit speziellen Themenbereichen tiefergehend befassen. Ziel ist die verständliche Vermittlung theoretischer Grundlagen und deren Transfer in praxisorientiertes Wissen.
- Zielgruppen:** Als Zielgruppen sehen wir Forschende, Lehrende und Lernende in der Disziplin Wirtschaftsinformatik sowie das IT-Management und Praktiker in Unternehmen.
- Quellen:** Die Arbeitspapiere entstehen aus Forschungs-, Abschluss-, Studien- und Projektarbeiten sowie Begleitmaterialien zu Lehr-, Vortrags- und Kolloquiumsveranstaltungen der Professur BWL – Wirtschaftsinformatik, Prof. Dr. Axel Schwickert, Justus-Liebig-Universität Gießen sowie der Professur für Wirtschaftsinformatik, insbes. medienorientierte Wirtschaftsinformatik, Prof. Dr. Bernhard Ostheimer, Fachbereich Wirtschaft, Hochschule Mainz.
- Hinweise:** Wir nehmen Ihre Anregungen zu den Arbeitspapieren aufmerksam zur Kenntnis und werden uns auf Wunsch mit Ihnen in Verbindung setzen.
- Falls Sie selbst ein Arbeitspapier in der Reihe veröffentlichen möchten, nehmen Sie bitte mit einem der Herausgeber unter obiger Adresse Kontakt auf.
- Informationen über die bisher erschienenen Arbeitspapiere dieser Reihe erhalten Sie unter der Web-Adresse <https://wi.uni-giessen.de/>