

# 3 SQL - Grundlagen

Sonntag, 10. September 2023 15:59



3 SQL

### 3. Structured Query Language (SQL) – Grundlagen



SQL ist die Abkürzung für „Structured Query Language“ und heißt übersetzt „strukturierte Abfragesprache“. SQL dient in Datenbanksystemen zur Definition von Daten und zur Informationsgewinnung. Heutzutage nennt man SQL häufig in Verbindung mit relationalen Datenbanken. Dabei hat SQL zunächst nichts damit zu tun, wie die Datenbank technisch realisiert ist, sondern definiert die Sprache, die verwendet wird, um mit den Datenbankinhalten zu arbeiten. Das Konzept der relationalen Datenbanken basiert auf mathematischen Ansätzen aus den frühen 70er Jahren des vergangenen Jahrhunderts. Das Grundprinzip ist, dass die Daten in Form von Tabellen gespeichert werden, die logisch miteinander verknüpft sein können (siehe logisches Modell). In diesem Zusammenhang wurde auch die SQL-Sprache entwickelt, um auf diese Daten zugreifen zu können. Die Beherrschung der SQL-Befehlssyntax ist deshalb für einen effektiven Umgang mit SQL-Datenbanken unerlässlich.<sup>2</sup>

Die SQL-Syntax wird von verschiedenen Datenbanksystemen verwendet. Die folgende Auflistung zeigt eine kleine Auswahl:

MySQL	IBM DB2
MS Access	Oracle
PostgreSQL	Microsoft SQL Server
Firebird/InterBase	SQLite
OpenOffice.org	und weitere ...

Die Auflistung hat nicht den Anspruch vollständig und allumfassende zu sein. Vielmehr soll sie verdeutlichen, dass es viele verschiedene Systeme auf dem Markt gibt. Diese weisen zum Teil untereinander leichte Abweichungen bei der Syntax auf. Daher ist es immer hilfreich bei der Verwendung von bisher unbekannten Systemen, sich zunächst über mögliche Syntax-Abweichungen zu informieren. Wir werden im weiteren Verlauf mit MySQL arbeiten.

<sup>2</sup> Throll & Bartosch (2011): Einstieg in SQL – Verstehen, einsetzen, naschschlagen. 4. Aktualisierte und erweiterte Auflage, Galileo Computing, S. 23f.

SQL gliedert sich in verschiedene Teilbereiche.

- 1) Data Definition Language (DDL)
- 2) Data Manipulation Language (DML)
- 3) Transaction Control Language (TCL)
- 4) Data Control Language (DCL)



### Arbeitsauftrag

**Recherchieren** Sie in **Partnerarbeit** nach den einzelnen Teilbereichen und **erläutern** Sie diese kurz in der untenstehenden Tabelle. Finden Sie zudem **Beispiele** (Befehl) für das jeweilige Teilgebiet.



Teilgebiet	Erklärung	Beispiel
DDL (Data Definition Language)	DDL definiert die Schulter einer DB	CREATE, DROP, ALTER
DML (Data Manipulation Language)	DML beschäftigt sich mit dem Inhalt der DB	INSERT, UPDATE, DELETE, SELECT
DCL (Data Control Language)	Rechte f. Zugriff auf Objekte (Tabellen, Felder, Funktionen...)	GRANT, REVOKE
TCL (Transaction Control Language)	Für die Integrität d. Datum u. deren Sicherheit	COMMIT, ROLLBACK

## 3.1 DDL

Sonntag, 10. September 2023 16:00



### 3.1 DDL

### 3.1 Data Definition Language (DDL)

Um mit einer Datenbank zu arbeiten, wird logischerweise zunächst eine Datenbank benötigt. Wie Sie bereits im vorangegangenen Kapitel gesehen haben, befasst sich mit dieser Thematik die DDL (Data Definition Language). Im weiteren Verlauf werden daher im Bereich der DDL gängige Befehle vorgestellt und mit diesen gearbeitet.

#### 3.1.1 Datenbanken anlegen und löschen

**Ausgangssituation**

**KOSidas**

**FALL**

DIE KOSIDAS GMBH IST EIN REGIONAL ERFOLGREICHER PRODUZENT VON SPORTARTIKELN UND -BEKLEIDUNG MIT HAUPTSTADT IN WÜRZBURG. DIE PRODUKTPALETTE REICH VON SCHUHEN ÜBER BEKLEIDUNG BIS ZU ACCESSOIRES IM SPORTBEREICH. DER VERTRIEB DER PRODUKTE ERFOLGT ÜBER REGIONALE GROßHÄNDLER. AUFGRUND DER WACHSENDEN NACHFRAGE UND STEIGENDEN UMSÄTZE SOLL NUN EIN ONLINE-STORE ERÖFFNET WERDEN. GRUNDLAGE HIERFÜR IST EINE NEUE UMFASSENDE DATENBANK. SIE SIND DERZEIT AUSZUBILDENDER ALS FACHINFORMATIKER / -IN IN DER KOSIDAS GMBH UND HELFEN BEIM AUFBAU DER DATENBANK MIT.

**Artikelübersicht**

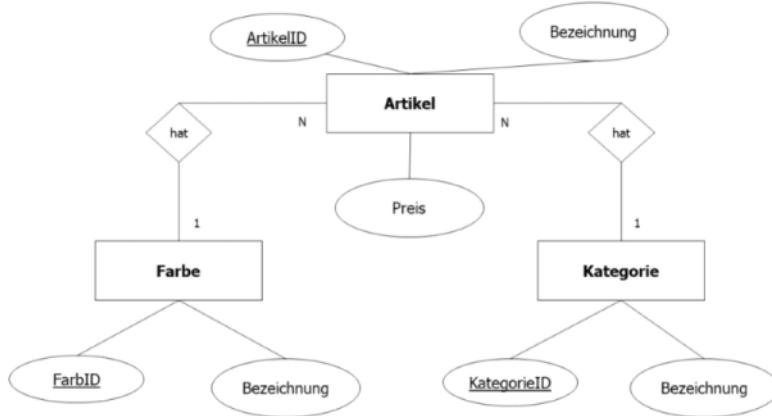


Kategorie	Produkte
Schuhe	* UEFA EURO 2016 Ball * FIFA WM 2014 Ball * KOSidas Team-Tasche
Bekleidung	* Poloshirt London * Tanktop O.L. * Longsleeve O.L. * Trainingshose CHECK
Accessoires	* River * Light * Natural



### Arbeitsauftrag

- a) Für den Aufbau der Datenbank liegt bisher ein kleiner Wirklichkeitsauschnitt (ER-Modell) vor, der von Ihrem Kollegen angefertigt wurde.  
Überführen Sie diesen in ein logisches Modell



Nachdem Sie das ER-Modell in ein logisches Modell überführt haben, folgt gemäß den Phasen der Datenbankentwicklung die Physische Phase und somit die Überführung des logischen Modells in ein Datenbankschema. Die KOSidas nutzt als relationales Datenbankverwaltungssystem MySQL mit der MySQL Workbench.

- b) **Informieren** Sie sich über den Befehl zum **Anlegen einer Datenbank** und notieren Sie die Syntax.

```
create database database_name;
```

- c) Immer wieder muss eine Datenbank gelöscht werden. **Recherchieren** Sie nach dem entsprechenden Befehl zum **Löschen von Datenbanken** und **notieren** Sie die **Syntax**.

drop database database\_name;

- d) Arbeiten Sie nun konkret in der MySQL Workbench und legen Sie die Datenbank Kosidas unter dem Namen „**kosidas\_[jahr]\_[klasse]**“ konkret an.



Ein interessanter Hinweis! Wussten Sie, dass eckige Klammern nur in der Syntax zu finden sind und nicht ins Statement übernommen werden? Allgemein stehen sie für optional. Im obigen Fall allerdings als Platzhalter.



- e) Legen Sie zunächst erneut die Datenbank an **OHNE** diese vorher zu löschen. Welche Meldung wird Ihnen ausgegeben? Finden Sie eine Möglichkeit durch Erweiterung der Befehle aus b) und c) dieses Problem zu umgehen.

### 3.1.2 Tabellen anlegen und löschen

Die Datenbank haben Sie erfolgreich angelegt. Im nächsten Schritt müssen nun die Tabellen angelegt werden.



#### Arbeitsauftrag

**Informieren** Sie sich über den entsprechenden Befehl zum Anlegen einer Datenbank (allgemeine Syntax). Berücksichtigen Sie bei Ihrer Recherche auch den **Primär-** und **Fremdschlüssel**, sowie die unten aufgeführten Besonderheiten



AUTO_INCREMENT	MySQL vergibt <b>automatisch</b> einen ganzzahligen Feldwert und <b>erhöht</b> diesen auch <b>automatisch</b> .
NOT NULL	Hierbei <b>muss</b> im Feld ein Wert stehen. Dies gilt besonders bei <b>Primärschlüsseln</b> .

### 3.1.3 Datentypen in MySQL

Beim Anlegen von Tabellen ist es auch wichtig für jede Spalte den richtigen Datentyp auszuwählen.



### Arbeitsauftrag

- a) Betrachten Sie das **unter Abschnitt 3.1.1** dargestellte **ER-Modell**. **Recherchieren** Sie nach **geeigneten und sinnvollen Datentypen** für die gezeigten **Attribute**.

Attribut	Datentyp	Bedeutung	Beispiel
ArtikelID, FarbID, KategorienID	Int	ganzzahl	ArtikelID = 2
Bezeichnung	VARCHAR(N) CHAR(N) TEXT BLOB	zeichenfolge	Bezeichnung = "Sportschuh x"
Preis	FLOAT(N,M) DECIMAL(N,M) DOUBLE(N,M)	gleitkommazahl N = Anzahl stellen M = Anzahl nachkommastellen	Preis = 199.99

### Beispieldaten

ArtikelID	Bezeichnung	Preis	KategorienID	Bezeichnung
134458	Sportschuh River	59.95	1	Schuhe
...	...	...	...	...

FarbID	Bezeichnung
#000000	schwarz
...	...

- b) Legen Sie nun die **Tabellen Farbe** und **Kategorie** in MySQL an.
- c) Legen Sie nun die **Tabelle Artikel** in MySQL an.
- d) Ihr Kollege ist **anders vorgegangen** und hat **erst** versucht die Tabelle **Artikel** anzulegen. **War das sinnvoll?**

Bitte beim Anlegen der Tabellen auch an ALLE Schlüssel denken!



### 3.1.4 Constraints

Mit Constraints ("Zwangsbedingungen") werden Bedingungen definiert, die zwingend vom Wert einer Variablen erfüllt werden müssen, damit der Wert ins System übernommen werden kann. Der Befehl Constraint gehört zum CreateTable-Befehl und erzwingt Integritätsbedingungen auf den Daten der Tabelle.



#### Arbeitsauftrag

- Recherchieren Sie, welche **Constraints** sich hinter den Erklärungen verbergen und schreiben Sie diese in die Spalte Keyword.
- Finden Sie je ein **Beispiel** aus den Tabellen der Datenbank **der KOSidas**.

Keyword	Erklärung (Bedeutung)	Beispiel
PRIMARY KEY	Legt ein Attribut als Primärschlüssel einer Tabelle fest	ArtikelId
FOREIGN KEY	Kennzeichnet ein Attribut als Fremdschlüssel und stellt Beziehung zu einer anderen Tabelle her	KategorieId
UNIQUE	Ein Wert darf in dieser Spalte nur jeweils einmal vorkommen	Tabelle Kunde → IBan
NOT NULL	Dieses Attribut muss für jeden Datensatz der Spalte gefüllt werden. Es darf nicht leer sein.	Tabelle Artikel → Bezeichnung
DEFAULT	Wird beim Einfügen eines Datensatzes kein Wert übergeben, wird der Default-Wert gesetzt	Tabelle Artikel → Preis
CHECK	Spezifiziert Bedingungen für die möglichen Werte einer Spalte.	Tabelle Artikel → Preis

### 3.1.5 Index

Indizes haben eine besondere Bedeutung für die Suche und die Performance von Datenbanken. Sie beschleunigen

- das Auffinden von Informationen bei Abfragen,
- die Sortierung von Tabellen,
- die Suche nach Maximal- und Minimalwerten innerhalb einer Datenreihe sowie
- die Abfrage über verschiedene Tabellen (wird später behandelt).

Ein Index ist dabei nichts anderes als eine interne Aufbereitung der Daten einer Form, die schnellere Suchen bzw. das Auffinden einzelner Datensätze erlaubt. Indizes werden vollständig von der Datenbank verwaltet und beim Löschen oder Hinzufügen von Datensätzen in einer Tabelle automatisch aktualisiert. Man muss für die Tabelle lediglich definieren, welche Datenfelder mit einem Suchindex versehen werden sollen. Dabei können Tabellen keinen, einen oder mehrere Indizes besitzen. Ein Index kann auch aus mehreren Attributen bestehen.

Index beim Tabellenanlegen definieren	Index nach Tabellenanlegen definieren
<pre>CREATE TABLE tabellenname (     Spaltenname1 datentyp(),     Spaltenname2 datentyp(),     ...     PRIMARY KEY (spaltenname1),     INDEX indexname(spaltenname2) );</pre>	<pre>CREATE INDEX indexname     ON tabellenname         (spaltenliste);</pre>
Index löschen	
<pre>DROP INDEX indexname;</pre>	



#### Arbeitsauftrag

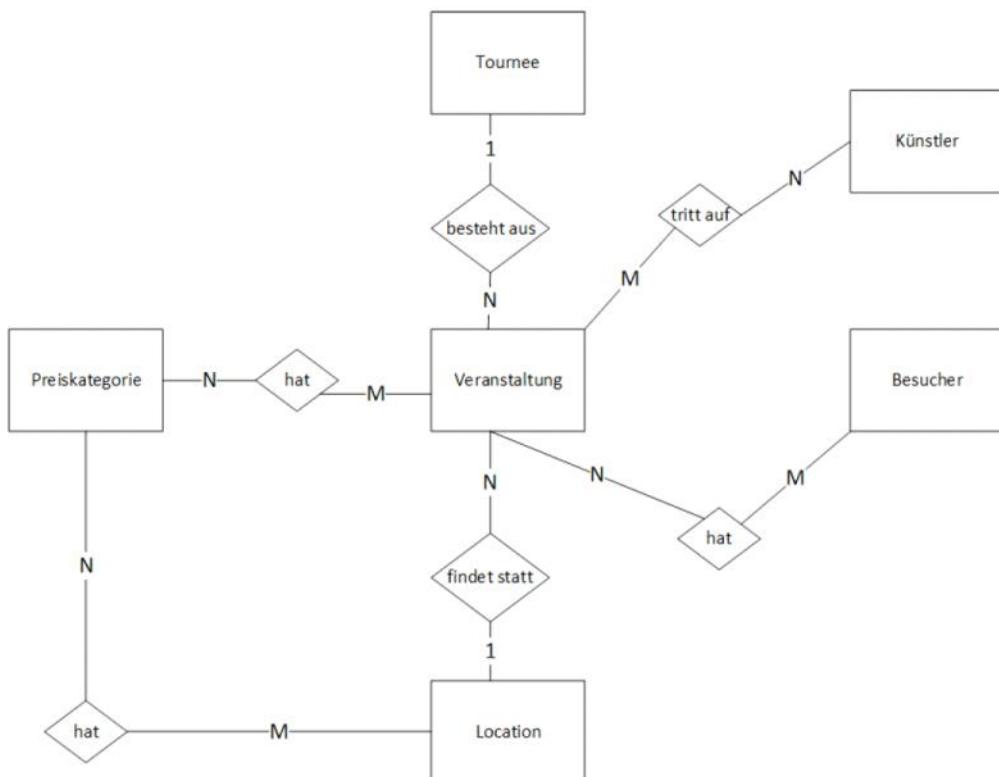
Überlegen Sie sich, auf welchen Spalten ein Index sinnvoll wäre und wieso?  
Was ist dabei das Problem bei einem Index?



### 3.1.6 Übungsaufgaben: Datenbanken und Tabellen anlegen

#### Aufgabe 1

Die Fun AG hat sich auf Event- und Marketing-Software spezialisiert. Die Fun AG wurde von der ADAM-Event GmbH, einem Online-Ticket-Anbieter, mit der Entwicklung eines Online-Buchungssystems beauftragt. Aus der Befragung der Mitarbeiter der ADAM-Event GmbH hat sich folgendes ER-Modell ergeben:



Anhand des ER-Modells soll ein relationales Datenbanksystem realisiert werden.

- a) Überführen Sie das ER-Modell in ein **logisches Modell** und geben Sie für jede Tabelle **zunächst nur Primär- und Fremdschlüssel** an.

- b) Ordnen Sie folgende Nicht-Schlüsselattribute der richtigen Tabelle zu:

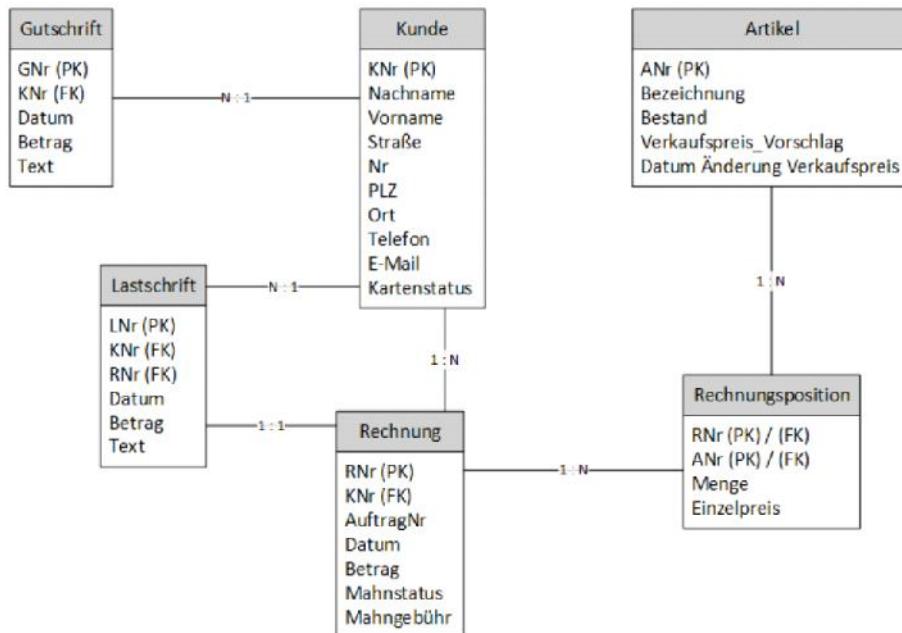
Sitzplatz	BesucherEmail	Uhrzeit Beginn	Durchführungsdatum
Gage des Künstlers	Tourneename	Künstleradresse mit PLZ und Ort	Künstlername
LocationAdresse	LocationName	Preis in EURO	Preisstufe

**Hinweis:** bei einigen Attributen kommen je nach Interpretation mehrere Tabellen in Frage. Entscheiden Sie sich für eine!

- c) **Setzen** Sie den von Ihnen erstellten Entwurf **mit allen Primär-, Fremdschlüssel- und Nicht-Schlüsselattributen** in **SQL** um.

### Aufgabe 2

Erzeugen Sie auf Basis des vorliegenden logischen Modells die Tabellen in SQL mit Schlüsseln und Attributen. Verwenden Sie passenden Datentypen.



Schon fertig?  
Bonusaufgabe!



Setzen Sie die Übungsaufgaben ER-Modell Abschnitt 2.1.6 in MySQL um. Finden Sie dabei passende Attribute mit Datentypen.

### 3.1.7 Tabellenstruktur ändern

Sie haben bei der Erstellung einer Tabelle einen Fehler gemacht oder erkennen, dass der gewählte Datentyp in der Tabelle doch nicht vorteilhaft ist. Oder Sie möchten die Schlüssel neu anpassen. In all diesen Fällen kommen Sie um die Änderung der Tabelle nicht umher.



#### Arbeitsauftrag

- a) Informieren Sich mittels Internetrecherche über den **ALTER**-Befehl beim Hinzufügen neuer Spalten. Notieren Sie sich die grundlegende Syntax und füllen Sie die Tabelle aus

ALTER TABLE tabellename  
ADD spaltenname datentyp;

#### Besonderheiten

Befehl	Bedeutung
AFTER	Es wird eine Spalte nach einer anderen Spalte eingefügt.

- b) Neben dem Hinzufügen neuer Spalten können über den **ALTER**-Befehl auch Spalten gelöscht werden. Notieren Sie sich die Syntax für diesen Fall.

ALTER TABLE tabellename  
DROP spaltenname;

- c) Auch können Tabellenspalten modifiziert und umbenannt, sowie ganze Tabellen umbenannt werden. Füllen Sie die nachstehende Tabelle aus.

Bedeutung	Syntax
Tabellenspalte modifizieren	ALTER TABLE tabellenname MODIFY spaltenname datentyp;
Tabellenspalte umbenennen	ALTER TABLE tabellenname CHANGE spaltenname_alt Spaltenname_neu datentyp;
Tabelle umbenennen	RENAME TABLE tabellenname_alt TO tabellenname_neu;

- d) Gehen Sie davon aus, dass Sie eine Spalte löschen wollen, die als Fremdschlüssel gekennzeichnet ist. Was müssen Sie hierbei beachten?

1. Schritt FK löschen

```
ALTER TABLE tabellenname  
DROP FOREIGN KEY fk_name;
```

2. Schritt: Spalte löschen

```
ALTER TABLE tabellenname  
DROP spaltenname;
```

### 3.1.8 Übungsaufgaben: Tabellenstruktur ändern

#### Aufgabe 1

**Hinweis:** Arbeiten Sie in der von Ihnen erstellen Datenbank „kosisadas\_[jahr]\_[klasse]“ weiter.

kunde
KundeID INT(11)
Nachname VARCHAR(45)
Vorname VARCHAR(45)
Straße_NR VARCHAR(45)
PLZID INT(11) (FK)
IBAN VARCHAR(34)
BIC VARCHAR(11) (FK)

- a) Legen Sie die obenstehende Tabelle „Kunde“ an.
- b) Ändern Sie die Spalte „Straße\_NR“ in „Strasse“ mit VARCHAR (45).
- c) Fügen Sie die Spalte „Nummer“ (VARCHAR (3)) direkt nach „Strasse“ hinzu.
- d) Legen Sie nun folgende Tabellen neu an:
  - a. Bank mit den Spalten „BIC“ (VARCHAR (11)) als PRIMARY KEY, der Spalte „Name“ (VARCHAR(45)) und „PLZID“ als INT.
  - b. Ort mit den Spalten „PLZID“ als INT (PRIMARY KEY), Spalte „PLZ“ als VARCHAR(5) und „Stadt“ als VARCHAR (45).
- e) Ändern Sie alle Tabellen insoweit, dass die Fremdschlüssel zu den bestehenden Tabellen (Bank und Ort) berücksichtigt werden.

### Aufgabe 2

Für die weiteren Aufgaben ändern Sie die von Ihnen erstelle Datenbank **Tournee**.

- a) Ändern Sie das Attribut Tourneename auf einen VARCHAR mit 50 Zeichen ab.
- b) Ändern sie den Namen der Tabelle Veranstaltung\_Besucher auf „Veranstaltungsbesucher“, Preiskategorie\_Location auf "Locationpreise", Preiskategorie\_Veranstaltung auf "Veranstaltungspreise" und Veranstaltung\_Kuenstler auf "Teilnehmende\_Kuenstler" ab.
- c) Löschen Sie das Attribut Anzahl\_Plätze\_gesamt in location.
- d) Fügen Sie beim Künstler ein Attribut „Besonderheiten“ vom Datentyp VARCHAR (100 Zeichen) hinzu.
- e) Fügen Sie die Tabelle Ort in die Datenbank ein. In dieser soll der Ort und die PLZ erscheinen.

OrtID	PLZ	Ort
1	99010	Würzburg
2	99014	Würzburg
...	...	...

- f) Das Einfügen der Tabelle Ort genügt den Regeln der Normalisierung. In welchen Tabellen macht es Sinn, die Tabelle Ort als Fremdschlüssel aufzunehmen? Setzen dies konkret um.

## 3.2 DML (ab 3.2.2 Abfragen auf eine Tabelle)

Sonntag, 10. September 2023 16:01



3.2 DML

### 3.2 Data Manipulation Language (DML)

Nachdem wir Datenbanken und Tabellen anlegen können, sowie deren Strukturen ändern können, wollen wir in Datenbanken arbeiten. Im Bereich der Data Manipulation Language fallen sämtliche Befehle, um Datensätze zu schreiben, aktualisieren, löschen und abzufragen.

#### 3.2.1 Datensätze einfügen und löschen



##### Arbeitsauftrag

- a) **Informieren** Sich mittels Internetrecherche über die grundlegende Syntax zum **Einfügen von Datensätzen** in MySQL. **Notieren** Sie sich die **Syntax**.

INSERT INTO tabellenname (spalte1,spalte2,...,spalteN)  
VALUES(wert1,wert2,wert3,...,wertN)

- b) Gehen Sie davon aus, dass für ein Attribut **kein Wert** übergeben werden soll. Wie ist dies realisierbar?

Man übergibt NULL bei den VALUES.  
setzt voraus dass die Spalte NULL erlaubt.

- c) Informieren Sich mittels Internetrecherche über die grundlegende Syntax zur Änderung und Löschung von Datensätzen in MySQL. Notieren Sie sich die Syntax.

UPDATE tabellenname  
SET spaltenname = wert  
WHERE auswahlbedingung;

DELETE FROM tabellenname  
WHERE auswahlbedingung;

UPDATE Schueler  
SET nachname = 'Schmidt'  
WHERE ID\_S = 3;

### 3.2.2 Übungsaufgaben: Datensätze einfügen, löschen und aktualisieren

#### Aufgabe 1

Fügen Sie folgende Datensätze in die bereits von Ihnen angelegten Tabellen ein.

Farbcode	Bezeichnung
#FFFFFF	Weiß
#000000	schwarz
#c0c0c0	gruen
#FFFF00	gelb

PLZID	PLZ	Stadt
1	90402	Nuernbrg
2	80271	Muenchen
3	97070	Wuerzburg
4	97816	Lohr a. Main

Kategorie	Bezeichnung
1	Schuhe
2	Kleidung
3	Accessoires

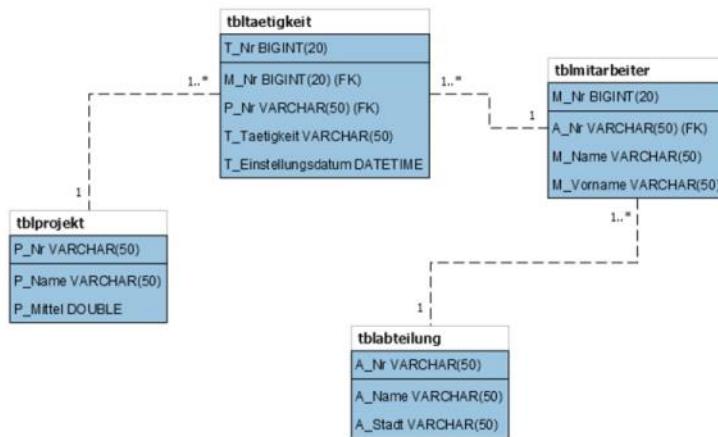
BIC	Bankname	PLZID
GENODEF1GEM	Raiffeisenbank Main-Spessart	4
BYLADEM1SWU	Sparkasse Main- franken Wuerz- burg	3
DEUTDEMMXXX	Deutsche Bank Muenchen	2

ArtikelID	Bezeichnung	KategorieID	Farbcode	Preis
134456	Sportschuh Ri- ver	1	#000000	59.95
134457	Sportschuh Ri- ver	1	#c0c0c0	79.95
134465	Poloshirt Lon- don	2	#FFFFFF	19.95

**Achtung:** Beachten Sie mögliche Fremdschlüssel und berücksichtigen Sie diese beim Anlegen und hinzufügen von Datensätzen!

### Aufgabe 2

**Nutzen** Sie für die Bearbeitung der Aufgaben die **Datenbank lunna2000\_in-nodb**.



- Fügen Sie der Tabelle **tblabteilung** die Abteilung IT mit der **A\_Nr „a4“** hinzu. Diese befindet sich in Würzburg.
- Fügen Sie außerdem eine Abteilung Marketing mit der **A\_Nr „a5“** hinzu. Diese soll im Laufe dieses Monats neu etabliert werden. Der Standort steht noch nicht fest.
- Die Abteilung Marketing wird ihren Sitz in Schweinfurt haben. Ändern Sie den Datensatz entsprechend.
- Die Abteilung IT wird ausgelagert. Löschen Sie den Datensatz aus der Tabelle!
- Jede Abteilung soll zukünftig mit einem "Motivationsspruch" versehen werden. Ändern Sie die Tabelle entsprechend ab, dass in einem zusätzlichen Datenfeld zunächst bei jeder Abteilung der Spruch "You are the best!" erscheint.
- Die Motivationssprüche scheinen die Mitarbeiter eher zu belustigen, deshalb wurde beschlossen, zukünftig auf Motivationssprüche zu verzichten. Löschen Sie die entsprechende Spalte aus der Tabelle.

### 3.2.3 Abfragen auf eine Tabelle

#### Ausgangssituation



Zurzeit sind Sie allein in der Abteilung und Sie bekommen von Ihrem Chef den Auftrag mit Hilfe der neuen Datenbank eine Übersicht aller Kunden, die mit „B“ anfangen (Nachname) zu erstellen. Zudem sollen Sie alle Artikel mit Artikelnummer und Bezeichnung auflisten, die „Sportschuh“ oder „Trainingshose“ in ihrer Bezeichnung haben.



#### Arbeitsauftrag

- a) **Informieren** Sie sich mittels Internetrecherche über den entsprechenden Befehl für eine **einfach Abfrage**. **Notieren** Sie sich die grundlegende **Syntax** und **füllen** Sie die **Tabelle** aus.

SELECT [spalte, spalte, -] FROM tabelle [WHERE bedingung]

Schlüsselwort	Frage	Beschreibung
SELECT [spalte, spalte, -]	Was?	Gibt an, was (welche Merkmale) ausgegeben werden sollen.
FROM tabelle	Woher?	Bestimmt die Datenquelle, aus der die Daten stammen.
WHERE spalte = 'asd'	Wobei? Womit?	Legt Bedingungen fest, die erfüllt werden müssen.

- b) Häufig ist es zunächst sinnvoll, sich die **gesamte Tabelle** (alle Merkmale der Datensätze) anzeigen zu lassen. **Notieren** Sie sich die **Syntax** für diesen Fall.

### Besonderheiten bei Bedingungen

Zur Formulierung von Bedingungen können die gängigen mathematischen Operatoren gleich (=), kleiner (<), größer (>), ungleich (<>), kleiner oder gleich (<=) und größer oder gleich (>=) in der WHERE-Bedingung verwendet werden. Zudem können im WHERE-Teil auch logische Operatoren AND, OR und NOT kombiniert werden, um mehrere Bedingungen miteinander zu verbinden. Dabei wird erst das NOT dann das AND und zum Schluss das OR ausgewertet. Wird eine andere Auswertungsreihenfolge benötigt, müssen Klammern gesetzt werden. Neben Vergleichsoperatoren stehen in MySQL aber auch Vergleichsfunktionen zur Verfügung. Eine Vergleichsfunktion ist der LIKE-Befehl. Mit LIKE und NOT LIKE kann ein Wert mit einer Vorgabe verglichen werden, indem für die Zeichen Platzhalter verwendet werden. Somit können Daten selektiert werden, deren genauer Wert nicht bekannt ist. In MySQL wird als Platzhalter das Prozentzeichen (%) und der Unterstrich ( \_ ) verwendet.



#### Arbeitsauftrag

Füllen Sie die Tabelle aus.

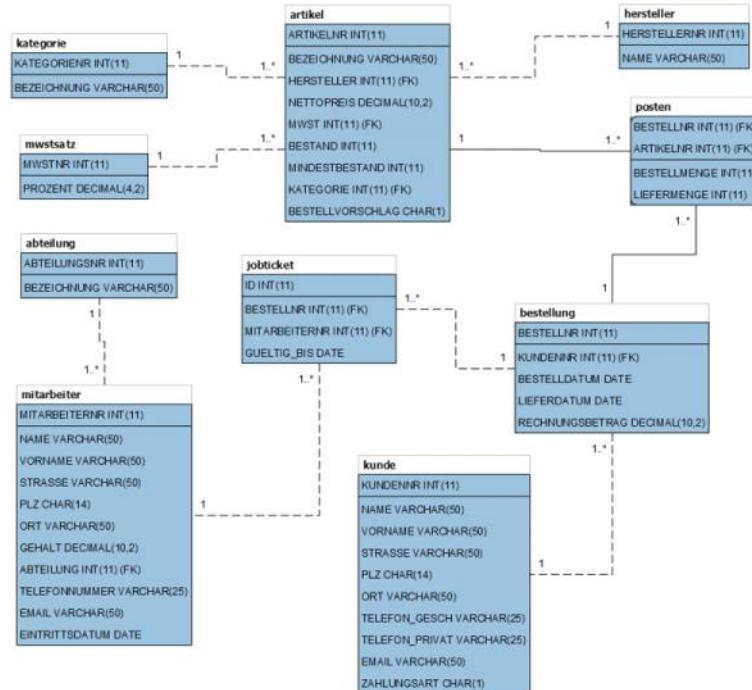
Platzhalter			
% (Unterstrich)			
Steht für beliebig viele beliebige Zeichen			Steht für genau ein beliebiges Zeichen
	<u>Beispiel</u> LIKE '%B'		<u>Beispiel</u> LIKE '_ DE '
Nach dem Buchstaben A kommen beliebig viele Zeichen.		Vor und nach dem Buchstaben C kommen beliebig viele Zeichen.	



#### Lösen Sie nun die Ausgangssituation!



### 3.2.4 Übungsaufgabe: Einfacher SELECT – Teil 1



Lösen Sie die folgenden Aufgaben **schriftlich auf Papier**. Prüfen Sie **anschließend** in der **MySQL Workbench** mit der **UnternehmensDB** Ihre Statements.

- Fragen Sie aus der Tabelle *kunde* den name, kundenr und die Zahlungsart ab.
- Listen Sie aus der Tabelle *artikel* die Bezeichnung und den Nettopreis aus.
- Suchen Sie aus der Tabelle *mitarbeiter* die Namen und die jeweiligen Abteilungsnummern heraus.
- Lassen Sie sich die Hersteller aus der Tabelle *artikel* ausgeben.
- Listen Sie alle Artikel der Tabelle *artikel* auf, deren Nettopreis höher als 100 Euro sind.
- Listen Sie alle Mitarbeiter auf, die in der Abteilung 2 beschäftigt sind.
- Geben Sie alle Kunden aus, deren Kundennummer größer als 50 ist und die nicht in Köln wohnen.

### 3.2.5 Vergleichsfunktionen

Im vorherigen Unterabschnitt haben Sie bereits „like“ als String-Vergleichsfunktion kennen gelernt. Neben „LIKE“ können mit den Anweisungen „IN“ und „BETWEEN“ Mengen- bzw. Suchbereiche definiert werden, die das Formulieren von WHERE-Anweisungen deutlich vereinfachen.



#### Arbeitsauftrag

Vervollständigen Sie die Beispiele.

- a) Welcher Mitarbeiternachname beginnt mit einem Buchstaben zwischen „A“ und „D“?

SELECT Nachname FROM Mitarbeiter

WHERE

- b) Welche Mitarbeiter wurden in den Jahren 1967, 1988 und 1990 geboren?

SELECT Vorname, Nachname, Geburtsdatum FROM Mitarbeiter

WHERE

- c) Welche Mitarbeiter beziehen ein Gehalt zwischen 1.500,00 Euro und 2.500,00 Euro?

SELECT Vorname, Nachname, Gehalt FROM Mitarbeiter

WHERE

- d) Welche Mitarbeiter arbeiten in den Abteilungen mit ID „3“ oder ID „1“?

SELECT Vorname, Nachname, ID\_Abteilung FROM Mitarbeiter

WHERE

### 3.2.6 Übungsaufgabe: Einfacher SELECT – Teil 2

Nutzen Sie die Datenbank luna.

**Hinweis:** Durch „DISTINCT“ können Doppelungen zusammengefasst werden → „SELECT DISTINCT ...;“



- a) Verschaffen Sie sich einen Überblick über die Tabelle tblAbteilung.
- b) Lassen Sie sich alle Mitarbeiter mit Vornamen und Nachnamen anzeigen.
- c) Finden Sie den Namen und Nummer aller Abteilungen, die ihren Sitz in München haben.
- d) Nennen Sie die Namen und Vornamen aller Mitarbeiter, deren Mitarbeiternummer mindestens 15000 ist.
- e) Gesucht werden Mitarbeiternummer, Projektnummer und Aufgabe der Mitarbeiter, die im Projekt p2 Sachbearbeiter sind.
- f) Gesucht wird die Mitarbeiternummer der Mitarbeiter, die entweder im Projekt p1 oder p2 oder in beiden tätig sind.
- g) Gesucht wird die Mitarbeiternummer der Mitarbeiter, die entweder im Projekt p1 oder p2 oder in beiden tätig sind. Jeder Treffer soll nun nur noch einmal erscheinen.
- h) Nennen Sie Mitarbeiternummer und Nachnamen der Mitarbeiter, die nicht in der Abteilung a2 arbeiten.
- i) Finden Sie alle Mitarbeiter, deren Mitarbeiternummer entweder 29346, 28559 oder 25384 ist.
- j) Nennen Sie alle Mitarbeiter, deren Mitarbeiternummer weder 10102 noch 9031 ist
- k) Nennen Sie Namen und Mittel aller Projekte, deren Etat zwischen 95000,00 € und 120000,00 € liegt.
- l) Nennen Sie die Mitarbeiternummer aller Mitarbeiter, die Projektleiter sind und vor oder nach 1988 eingestellt worden sind.
- m) Finden Sie die Mitarbeiter- und Projektnummer aller Mitarbeiter, die im Projekt p1 arbeiten und deren Aufgabe noch nicht festgelegt ist.
- n) Finden Sie Namen und Mitarbeiternummer aller Mitarbeiter, deren Name mit dem Buchstaben „K“ beginnt.
- o) Finden Sie Namen, Vornamen und Mitarbeiternummer aller Mitarbeiter, deren Vorname als zweiten Buchstaben ein „a“ hat.
- p) Finden Sie Abteilungsnummer und Standorte aller Abteilungen, die sich in den Orten, die mit einem der Buchstaben zwischen E und N beginnen, befinden.
- q) Finden Sie Namen, Vornamen und Mitarbeiternummer aller Mitarbeiter, deren Namen nicht mit den Buchstaben M, N, O und P, und deren Vornamen nicht mit H beginnt.
- r) Nennen Sie alle Mitarbeiter, deren Name nicht mit „mann“ endet.

### 3.2.7 Aggregatsfunktionen & Zusatzfunktionen

#### Ausgangssituation



Die neue Datenbank findet große Akzeptanz in der KOSidas GmbH. Jedoch bestehen Schwierigkeiten, wenn es darum geht Informationen aus den Abfragen zu erhalten. So lassen die Mitarbeiter immer noch alle Datensätze raus und suchen händisch nach den Informationen aus den Datensätzen. Vor allem bei Informationen über Maximal- und Minimalwerte oder Durchschnitts- und Summenwerte, welche noch händisch berechnet werden. Hier scheinen Datenbanken doch nicht so effektiv zu sein, wie behauptet und erhofft.



#### Arbeitsauftrag

**Informieren** Sie sich **mit Ihrem Sitznachbarn** über bestimmte **Aggregatsfunktionen** und deren Umsetzung in MySQL. **Füllen** Sie dabei die untenstehende Tabelle aus.

Syntax	Erklärung	Beispiel in der KO-Sidasdb
MIN(spalte)	Minimalwert	SELECT MIN(preis) FROM Artikel;
MAX(spalte)	Maximalwert	SELECT MAX(preis) FROM Artikel;

COUNT(*)	Anzahl der Zeilen in der Ergebnismenge	SELECT COUNT(*) FROM Personen;
COUNT(spalte)	Anzahl der Zeilen, die bei dem Merkmal nicht den Wert NULL haben	SELECT COUNT(Stand) FROM Personen;
COUNT(DISTINCT Spalte)	Anzahl der Zeilen mit unterschiedlichen Werten	SELECT COUNT(DISTINCT Stand) FROM Personen;
SUM(Spalte)	Summenwert	SELECT SUM(Preis) FROM Artikel;
AVG(Spalte)	Durchschnittswert	SELECT AVG(Preis) FROM Artikel;

**Fortsetzung Ausgangssituation**



Ihr Chef ist begeistert von Ihren Bemühungen die Effizienz bei der Nutzung der Datenbank zu steigern. Er fragt sich, ob es noch weitere Funktionen, z.B. im Bereich Datum und Rechnen, in SQL gibt, die man produktiv einsetzen kann. Er beauftragt Sie, Handouts zu erstellen, die die Informationen sachgerecht aufbereiten und auch auf Anwendungsszenarien verweisen.



**Arbeitsauftrag**



- a) Finden Sie sich in **4er-Gruppen (Stammgruppen)** zusammen.
- b) Entsenden Sie einen **Experten** zu den **jeweils folgenden Themen**.
  - a. Aktuelles Datum und Zeit in verschiedenen Varianten
  - b. Teileinformationen extrahieren (nur das Jahr, den Monat, den Tag, die Stunde, die Minute, die Sekunde aus Datum / Zeit bekommen)
  - c. Datumzusatzinformationen (Name des Monats, Name des Tages, Tag der Woche als Zahl, Quartalsausgabe und Kalenderwoche)
  - d. Rechenfunktionen beim Datum (Tage zwischen zwei Daten, Zeit zwischen zwei Daten)
- c) **Erstellen** Sie in den Expertengruppen das jeweilige von Ihrem Chef geforderte **Handout** mit sachgerechten Informationen und Anwendungsszenarien (Beispielen) bezogen auf die Datenbank KOSidasdb.
- d) **Gehen** Sie anschließend mit Ihren **Handouts** in die ursprünglich **Gruppen (Stammgruppen)** und verteilen Sie Ihr **Handout** und tauschen sich darüber aus.



### 3.2.8 INSERT-SELECT-Abfragen

#### Ausgangssituation



Die KOSidas GmbH gilt als ein arbeitnehmerfreundliches Unternehmen. So können auch die Mitarbeiter des Unternehmens bei der KOSidas GmbH Bestellungen aufgeben und profitieren von Aktionen und speziellen Rabatten. Derweil ist es allerdings problematisch, da die Mitarbeiter nur in der Tabelle „Mitarbeiter“ geführt werden und nicht in der Tabelle „Kunde“. Ihr Kollege hat den Vorschlag gemacht alle Mitarbeiter in die Tabelle „Kunde“ händisch durch Abtippen zu überführen. Das scheint die beste Möglichkeit zu sein, das Problem zu lösen.

Auch ein Insert-Befehl kann auf einer Abfrage basieren. Damit ist es möglich z.B. Daten aus verschiedenen Tabellen in einer anderen Tabelle neu zusammenzustellen.



#### Arbeitsauftrag

- Informieren Sie sich über die allgemeine Syntax zum Insert-Select-Befehl und notieren Sie sich diese.

```
INSERT INTO tabellenname1 (spalte1,spalte2,...,spalteN)
SELECT spalet1,spalte2,...,spalteN+FROM tabellenname 2
[WHERE auswahlbedingung];
```

- b) Ihr Kollege ist auf das Problem gestoßen, dass evtl. bereits Mitarbeiter in der Tabelle Kunde aufgeführt sind. Wie kann sichergestellt werden, dass keine doppelten Einträge in der Tabelle Kunde vorhanden sind?

INSERT IGNORE INTO ....

Achtung !!: Jedoch muss vorher der Constraint unique gesetzt sein!

Alternative: subselect, um Datensätze zu filtern u. zwar vorher



- c) Helfen Sie bei der Lösung der Ausgangssituation.

### 3.2.9 Sortieren und Gruppieren

Der SELECT-Befehl ist um eine Sortierreihenfolge erweiterbar und es besteht zusätzlich die Möglichkeit die Ergebnismenge der Abfrage in einzelne Gruppen aufzuteilen, für die wiederum separate Gruppen-Bedingungen formuliert werden können.



#### Sortieren mit ORDER BY

Mit dem Zusatz ORDER BY kann die Ergebnismenge nach bestimmten Merkmalen sortiert werden. Die Sortierung kann auch nach mehreren Merkmalen erfolgen. Dafür werden alle Sortierungsmerkmale durch Komma getrennt und hinter der ORDER BY Anweisung aufgelistet. Die Sortierung erfolgt dann zunächst nach dem ersten Merkmal. Nur bei völliger Übereinstimmung in diesem Merkmal ist das zweite Merkmal für die feinere Sortierung maßgeblich usw.



#### Auftrag

Die ORDER BY-Anweisung muss die letzte Anweisung in einer SELECT-Abfrage sein.



- Ergänzen Sie folgendes Beispiel um die Sortierung. Es sollen alle Mitarbeiter mit Nachnamen und Vornamen ausgegeben werden. Sortiert alphabetisch nach Nachnamen.



SELECT Nachname, Vorname FROM Mitarbeiter

ORDER BY Nachname ASC



- Informieren Sie sich mittels Recherche, wie die Sortierung in aufsteigend und absteigend geändert werden kann. Halten Sie die Syntax in der Tabelle fest und ergänzen Sie das Beispiel aus a)



ASC	aufsteigend (Standard)
DESC	absteigend

SELECT Nachname, Vorname FROM Mitarbeiter

ORDER BY Nachname DESC



### Gruppierung mit GROUP BY



Durch die Anweisung GROUP BY wird die Ergebnismenge der Abfrage nach dem angegebenen Merkmal in Gruppen aufgeteilt. Da für jede Gruppe nur ein Wert je Merkmal angezeigt wird, müssen für alle Merkmale, nach denen nicht gruppiert worden ist, Aggregationsfunktionen im SELECT-Teil der Abfrage dafür sorgen, dass genau ein eindeutiger Gruppenwert bestimmt werden kann. Es können auch mehrere Merkmale zur Gruppenbildung verwendet werden. In dem Fall werden alle Datensätze zu einer Gruppe zusammengefasst, deren Datenwerte bei allen angegebenen Gruppierungsmerkmalen übereinstimmen.



### Arbeitsauftrag

- c) Betrachten Sie folgende Tabelle und das dazugehörige noch unvollständige SQL-Statement. Ergänzen Sie das Statement um die Gruppierung.

Region	Durchschnittstemperatur
Mittelfranken	10.7
Oberfranken	10.4
Unterfranken	10.6
...	...

SELECT Region, AVG(Temperatur)  
FROM Wetter  
GROUP BY Region

i

Die GROUP BY-Anweisung einer SELECT-Abfrage folgt nach der WHERE-Anweisung (wenn vorhanden, sonst direkt nach der FROM-Anweisung) und vor der ORDER BY-Anweisung.



- d) Die Tabelle soll nun um das Jahr erweitert werden, um eine Entwicklung abzulesen. Ergänzen Sie das Statement.

Region	Jahr	Durchschnittstemperatur
Mittelfranken	2020	10.3
Mittelfranken	2021	10.4
Oberfranken	2020	10.2
Oberfranken	2021	10.5
Unterfranken	2020	10.6
Unterfranken	2021	10.7
...	...	...

SELECT Region, YEAR(Datum), AVG(Temperatur)

FROM Wetter

GROUP BY Region, YEAR(Datum)  
ORDER BY Region, YEAR(Datum)



i



#### Gruppenbedingung im GROUP BY

Mit Hilfe der HAVING-Anweisung kann man Bedingungen definieren, die von der gesamten Gruppe erfüllt werden müssen. Die HAVING-Anweisung ist vergleichbar mit der WHERE-Anweisung, die ebenfalls Bedingungen definiert, allerdings für jeden einzelnen Datensatz und nicht für Gruppen.



#### Arbeitsauftrag

- e) Ergänzen Sie das Statement, um die entsprechende Tabelle zu generieren. Es sollen dabei nur die Regionen mit einer Durchschnittstemperatur von über 10.5 Grad Celsius ausgegeben werden.

Region	Jahr	Durchschnittstemperatur
Unterfranken	2020	10.6
Unterfranken	2021	10.7
...	...	...

SELECT Region, YEAR(Datum), AVG(Temperatur)

FROM Wetter

GROUP BY Region, YEAR(Datum)  
HAVING AVG(Temperatur) > 10.5



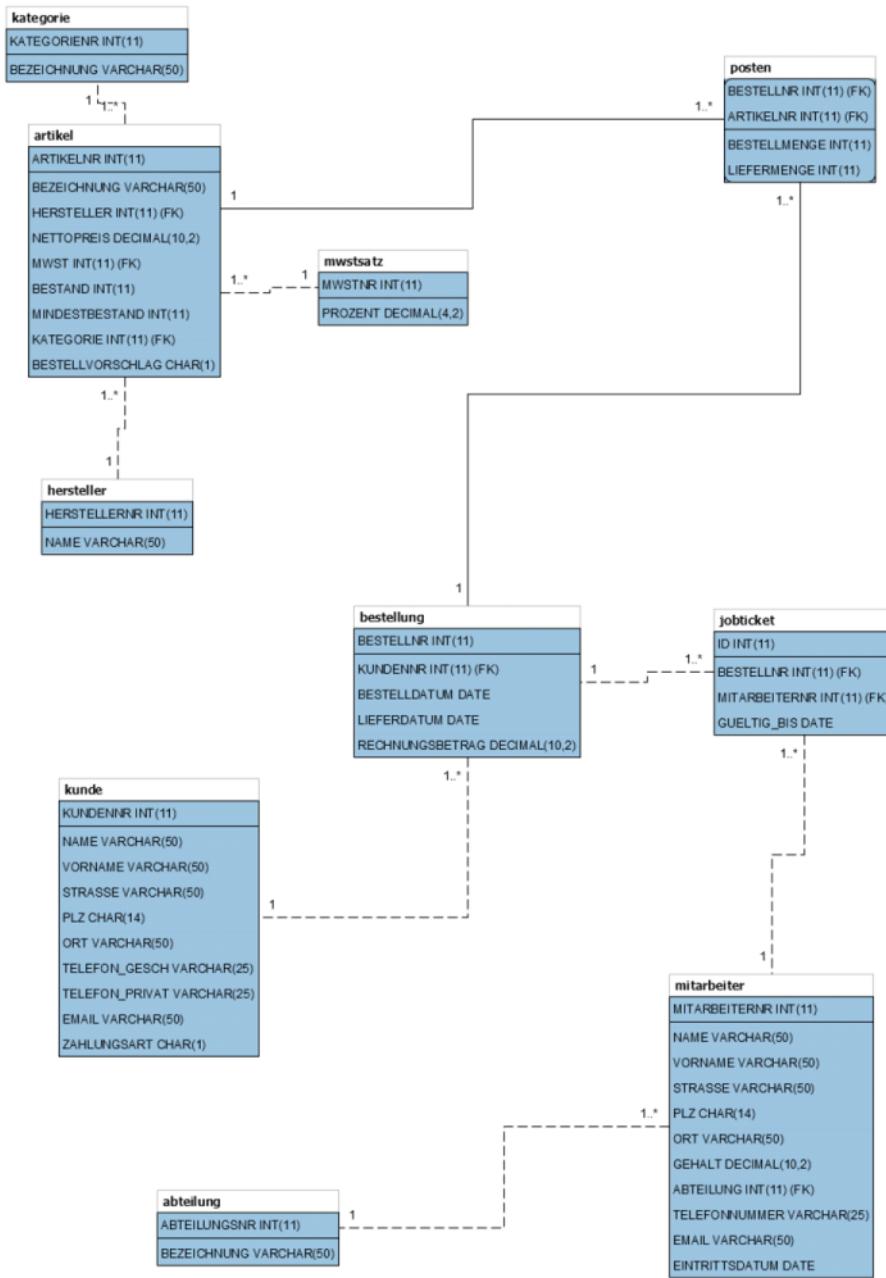
i

### 3.2.10 Übungsaufgaben: Aggregatsfunktionen, Gruppierung und Sortierung

**Aufgabe 1** - Nutzen Sie die Datenbank „kosidasdb“.

- a) Wie hoch ist das Durchschnittsgehalt der Mitarbeiter in PLZ-Region 6?
- b) Wie hoch ist das Durchschnittsgehalt der Mitarbeiter in den einzelnen Regionen (PLZID)?
- c) Wie viele männliche und weibliche Mitarbeiter arbeiten in den einzelnen Regionen (PLZID)?
- d) Wie hoch ist der durchschnittliche, höchste und niedrigste Preis der einzelnen Kategorien?
- e) Wie viele Artikel gibt es in den einzelnen Kategorien jeweils? Listen Sie nur die Kategorien auf, die mehr als 5 Artikel haben
- f) Wie viele Artikel gibt es in den einzelnen Kategorien jeweils mit einem Preis über 50,00 Euro.
- g) Wie viele Artikel gibt es in den einzelnen Kategorien jeweils mit einem Preis über 50,00 Euro? Listen Sie nur die Kategorien auf, die mehr als 2 solcher Artikel haben.
- h) Wie hoch ist das durchschnittliche Gehalt der Mitarbeiter in den einzelnen Regionen (PLZID) mit mindestens 2 Mitarbeitern.
- i) In welchen Kategorien liegt der Durchschnittspreis über 30,00 Euro?
- j) Aus wie vielen verschiedenen Artikeln bestanden die Bestellungen jeweils?
- k) Welche Kategorie hat Produkte mit den meisten unterschiedlichen Farben?
- l) Geben Sie an, wie viele Artikel sich in den einzelnen Kategorien befinden deren Gesamtwert kleiner als 600,00 Euro ist und deren Anzahl unter 20 liegt.

Lösen Sie die folgenden Aufgaben mit der Datenbank „UnternehmensDB“!



**Aufgabe 2**

- a) Sortieren Sie alle Artikel (Bezeichnung) in der Reihenfolge der Kategorie und dann alphabetisch auf.
- b) Listen Sie alle Mitarbeiter nach ihrem Gehalt und dann nach der Abteilung (ID) auf.  
Das Gehalt soll absteigend, die Abteilung aufsteigend sortiert werden.
- c) Listen Sie alle Artikel der Kategorie 4 (Festplatten) absteigend nach dem Preis auf.
- d) Listen Sie alle Kunden, die per Nachnahme (N) bezahlen, nach Postleitzahlbezirken auf.

**Aufgabe 3**

- a) Wie hoch ist der Durchschnittsverdienst aller Angestellten?
- b) Wie groß ist die höchste Bestellmenge in der Tabelle posten?
- c) Welcher Kunde steht alphabetisch am Anfang der Liste?
- d) Geben Sie IDs und Anzahl der Produkte von den zwei Herstellern aus, von denen die meisten Produkte im Angebot sind.

**Aufgabe 4**

- a) Wie hoch ist der Durchschnittsverdienst der Angestellten je nach Abteilung?
- b) Wie viel wird im Durchschnitt pro Artikel bestellt? Geben Sie auch die Artikelnummer an.
- c) Es soll eine Abfrage erstellt werden, die die Städte sortiert nach der Anzahl der dort lebenden Kunden ausgibt. Die Stadt mit den meisten Kunden soll dabei zuerst ausgegeben werden.
- d) Geben Sie die Städte ebenfalls nach der Anzahl der dort lebenden Kunden aus. Bei einer gleichen Anzahl an Kunden sollen die Städte alphabetisch ausgeben werden.
- e) Geben Sie die Anzahl der Artikel je Kategorie aus, die teuer als 50 Euro sind.
- f) Geben Sie die Bestellnummern von allen Bestellungen aus der Tabelle posten aus, bei der mehr als vier Artikel bestellt wurden. Sortieren Sie die Ausgabe nach der Anzahl der bestellten Artikel absteigend
- g) Wie viele Produkte von den einzelnen Herstellern (Hersteller- ID) sind im Angebot?