

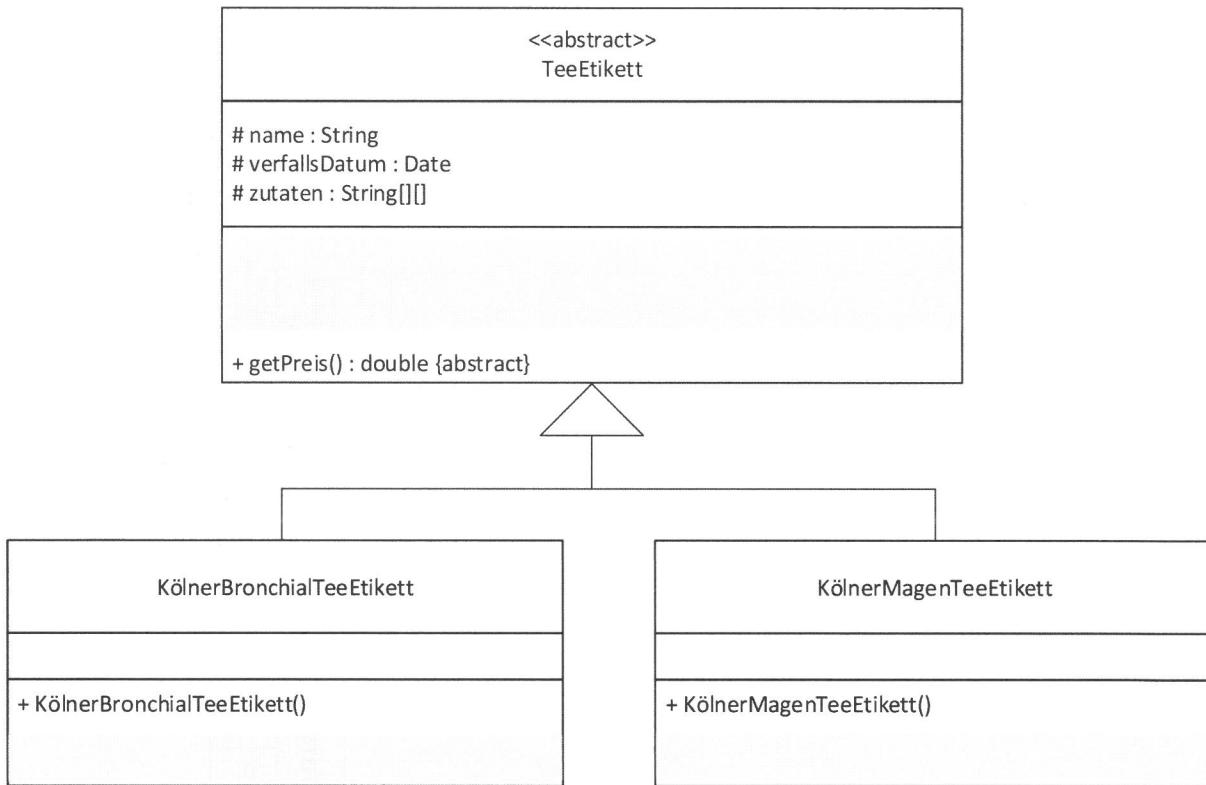
2. Handlungsschritt (25 Punkte)

Korrekturrand

Die eCo GmbH soll für die Pharmalog GmbH ein Programm erstellen, mit dem Etiketten für Tee-Verpackungen gedruckt werden können.

- a) Im Lagerbereich werden auch Teemischungen hergestellt, die etikettiert werden müssen.

Für die objektorientierte Drucksoftware liegt ein grober, noch unvollständiger Entwurf in Form eines Klassendiagramms bereits vor:



Die Instanzvariablen `name` und `zutaten` werden in den jeweiligen Konstruktoren initialisiert. Alle `TeeEtikett`-Klassen benötigen folgende öffentliche Methoden:

Funktion	Beschreibung
<code>berechneVerfallsDatum</code>	Legt für alle Teesorten in gleicher Weise die Instanzvariable <code>verfallsDatum</code> fest und gibt nichts zurück.
<code>druckeEtikett</code>	Erledigt für alle Teeetiketten in gleicher Weise den Ausdruck und gibt nichts zurück.
<code>getPreis</code>	Gibt den Preis der jeweiligen Teemischung als Dezimalzahl zurück.

aa) Ergänzen Sie im Klassendiagramm die fehlenden UML-Methodenbeschreibungen. 4 Punkte

ab) Erläutern Sie die Art der Beziehung zwischen den Klassen. 2 Punkte

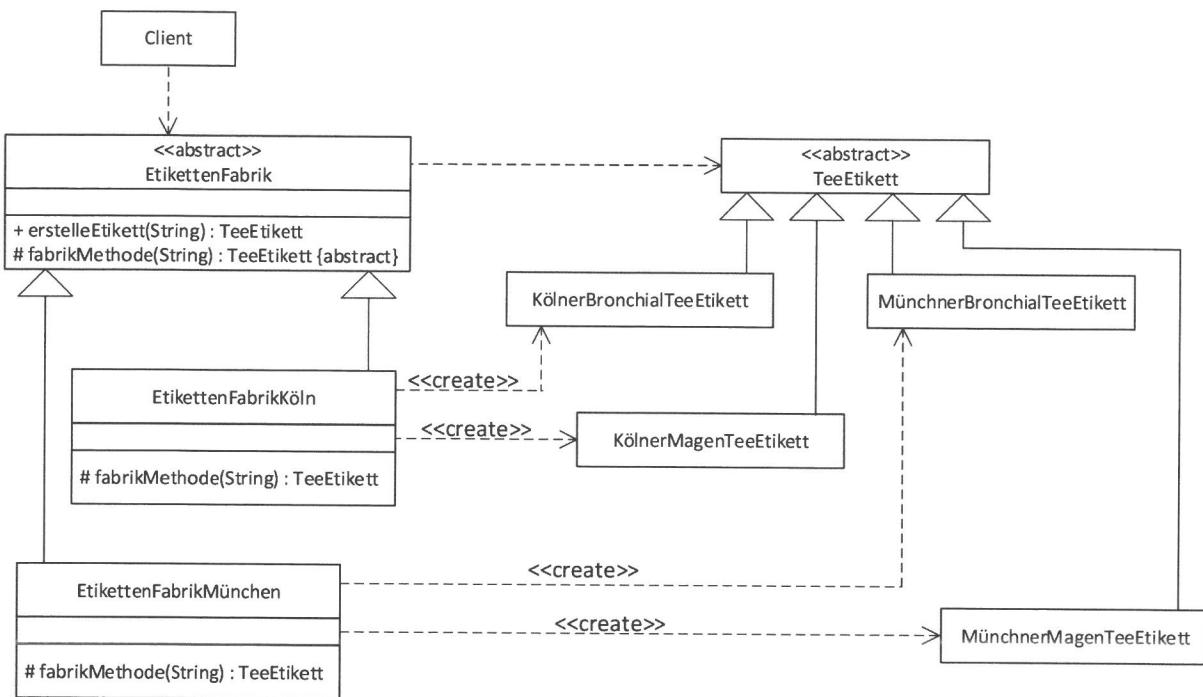
ac) Erläutern Sie die Bedeutung der `abstract`-Eigenschaft in Zusammenhang mit der Klasse `TeeEtikett` und

der Methode `getPreis` der `TeeEtikett`-Klasse. 2 Punkte

- b) Das Angebot an Teemischungen und auch deren Zusammensetzung wird laufend verändert. Deswegen soll die Verwendung von *TeeEtikett*-Objekten von ihrer Implementierung unabhängig sein.

Korrekturrand

Ein Kollege schlägt folgenden Entwurf vor:



Hinweis: Der Bezeichner für die String-Übergabeparameter soll *typ* sein.

String-Übergabeparameter = Bezeichnung für die Etiketten, z. B. „Bronchial“

- ba) Implementieren Sie beispielhaft in Pseudocode die Methode *fabrikMethode* der Klasse *EtikettenFabrikKöln* für die übergebene Zeichenkette „Bronchial“. 3 Punkte

-
-
-
-
-
- bb) Implementieren Sie in Pseudocode die Methode *erstelleEtikett*.

Die Methode *erstelleEtikett* soll mithilfe der Methode *fabrikMethode* ein Etikett erstellen und für dieses Etikett die Methode *berechneVerfallsDatum* der Klasse *TeeEtikett* aufrufen. 4 Punkte

Fortsetzung 2. Handlungsschritt

c) Die Fächer in den Regalen des Lagers der Pharmalog GmbH können eine bestimmte Anzahl (max) an Gegenständen aufnehmen. Daher kann jedes Fach einen der folgenden Zustände annehmen:

- leer: Das Fach enthält keine Gegenstände. Dies ist direkt nach der Einrichtung des Faches oder nach einer vollständigen Leerung der Fall.
- teilbelegt: Das Fach enthält Gegenstände, kann aber weitere (bis max Gegenstände) aufnehmen.
- voll Das Fach ist vollständig belegt.

Ein Fach wird beim Start eingerichtet und am Ende aufgelöst.

Der Übergang von einem Zustand zu einem anderen wird durch die Ereignisse *hineinlegen* bzw. *entnehmen* ausgelöst.

Dabei ist ein Zustandsübergang durch das Ereignis *hineinlegen* vom Zustand *leer* in den Zustand *teilbelegt* oder *voll* möglich. Durch das Ereignis *entnehmen* ist ein Zustandsübergang vom Zustand *voll* oder *teilbelegt* in den Zustand *teilbelegt* oder *leer* möglich.

Durch die Ereignisse *hineinlegen* oder *entnehmen* kann der Zustand *teilbelegt* erhalten bleiben.

Erstellen Sie für das beschriebene Szenario ein UML-Zustandsdiagramm für ein Fach des Lagers.

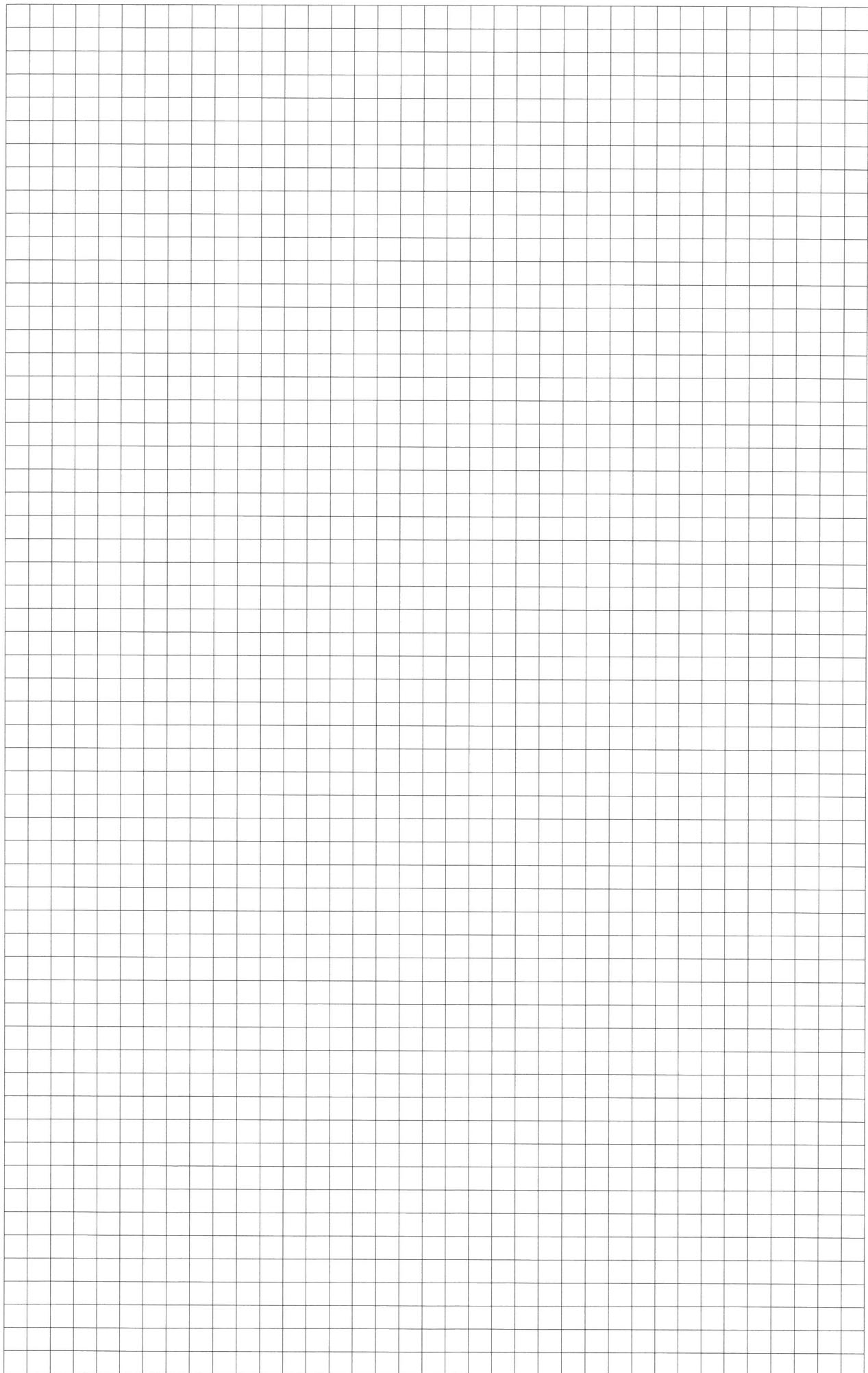
10 Punkte

	Anfangszustand
Zustand	Zustand, den ein Element einnehmen kann. z. B. Element = Fenster; Zustände: offen, geschlossen
Beispiel 	Zustandsübergang (Transition) von einem Quellzustand zu einem Zielzustand. In der Beschriftung kann Folgendes angegeben werden: - Ein Ereignis (trigger), welches den Zustandsübergang auslöst - Eine Bedingung (guard), welche beim Zustandsübergang erfüllt sein muss - Eine Handlung (action), welche den Zustandsübergang bewirkt
Zustand 	Selbstaufruf
	Endzustand

Fach

entnehmeWaren (liste : zweidimensionales Array vom Typ int)

Korrekturrand



4. Handlungsschritt (25 Punkte)

Korrekturrand

Die eCo GmbH soll für die Pharmalog GmbH eine Datenbank erstellen.

Der Sachverhalt wird wie folgt beschrieben:

Ein Einkäufer der Pharmalog GmbH führt mehrere Bestellungen aus.

Mit jeder Bestellung werden eine oder mehrere Waren zu unterschiedlichen Stückzahlen bestellt.

Eine Ware kann jeweils nur bei einem Hersteller bezogen werden.

Jede Ware wird einer Warengruppe zugeordnet.

An einem Lagerort können mehrere Waren gelagert werden.

- a) Erstellen Sie auf der gegenüberliegenden Seite zum dargestellten Sachverhalt das entsprechende Entity-Relationship-Modell (ER-Modell).

Hinweis: Das Attribut *Stückzahl* soll im ER-Modell angegeben werden.

18 Punkte

- b) Die Tabellen *Hersteller* und *Ware* der Datenbank wurden erstellt und mit Testdaten gefüllt.

Diese Testdaten sollen wieder gelöscht werden. Die Tabellen sind über das Attribut *Hersteller_ID* miteinander verknüpft.

Erläutern Sie, wie diese Löschung durchgeführt werden muss.

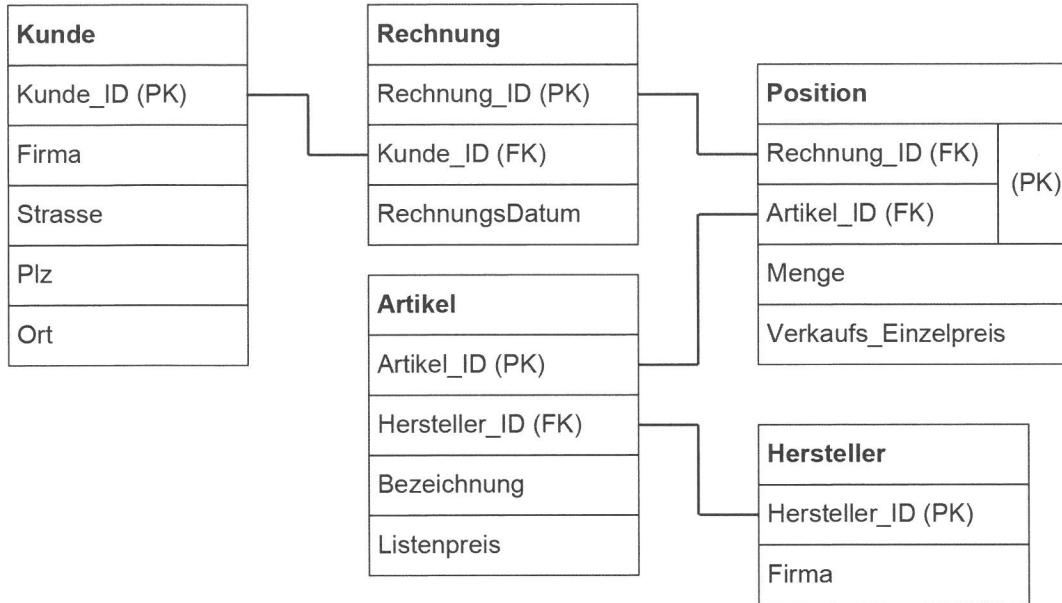
5 Punkte

- c) Erläutern Sie, wie die Beziehung zwischen *Ware* und *Bestellung* im relationalen Datenbankmodell abgebildet wird. 2 Punkte
-
-
-

5. Handlungsschritt (25 Punkte)

Korrekturrand

Die eCo GmbH hat für die Pharmalog GmbH folgende Datenbank entwickelt.



a) Zur Auswertung der Datenbank und zur Veränderung von Daten in der Datenbank sollen Sie die entsprechenden SQL-Anweisungen formulieren (siehe perforierte Anlage).

aa) Erstellen einer Liste aller Kunden mit Anzahl der Rechnungen im Jahr 2015.

Ausgegeben werden sollen: Kunde_ID, Firma und Anzahl.

7 Punkte

Beispiel

762762	ABC-Gesund AG	54
872346	Zeppelin-Pharma GmbH	54
138787	FeelGood-Partner OHG	49
...		
122636	Loser&Vaul GmbH	0

ab) Erhöhen der Listenpreise aller Artikel vom Hersteller BigPill AG um 4,5 %.

4 Punkte

Dieses Blatt kann an der Perforation aus dem Aufgabensatz herausgetrennt werden!

SQL-Syntax (Auszug)

Syntax	Beschreibung
Tabelle	
CREATE TABLE Tabellenname(Spaltenname < DATENTYP >, Primärschlüssel, Fremdschlüssel)	Erzeugt eine neue leere Tabelle mit der beschriebenen Struktur
ALTER TABLE Tabellenname ADD COLUMN Spaltenname Datentyp DROP COLUMN Spaltenname ADD FOREIGN KEY(Spaltenname) REFERENCES Tabellenname(Primärschlüsselpaltenname)	Änderungen in einer Tabelle: Hinzufügen einer Spalte Entfernen einer Spalte Definiert eine Spalte als Fremdschlüssel
CHARACTER	Textdatentyp
DECIMAL	Numerischer Datentyp (Festkommazahl)
DOUBLE	Numerischer Datentyp (Doppelte Präzision)
INTEGER	Numerischer Datentyp (Ganzzahl)
DATE	Datum (Format DD.MM.YYYY)
PRIMARY KEY (Spaltenname)	Erstellung eines Primärschlüssels
FOREIGN KEY (Spaltenname) REFERENCES Tabellenname(Primärschlüsselpaltenname)	Erstellung einer Fremdschlüssel-Beziehung
DROP TABLE Tabellenname	Löscht eine Tabelle
Befehle, Klauseln, Attribute	
SELECT * Spaltenname1 [, Spaltenname2, ...]	Wählt die Spalten einer oder mehrerer Tabellen, deren Inhalte in die Liste aufgenommen werden sollen; alle Spalten (*) oder die namentlich aufgeführten
FROM	Name der Tabelle oder Namen der Tabellen, aus denen die Daten der Ausgabe stammen sollen
SELECT ... (SELECT ... FROM ... WHERE ...) AS xyz FROM ... WHERE ...	Unterabfrage, die in eine äußere SELECT-Anweisung geschachtelt ist. Das Ergebnis der Unterabfrage wird im Spaltenausdruck (z. B. hier: xyz) ausgegeben.
INNER JOIN	Liefert nur die Datensätze zweier Tabellen, die gleiche Datenwerte enthalten
LEFT JOIN / Left OUTER JOIN	Liefert von der erstgenannten (linken) Tabelle alle Datensätze und von der zweiten Tabelle jene, deren Datenwerte mit denen der ersten Tabelle übereinstimmen
RIGHT JOIN / RIGHT OUTER JOIN	Liefert von der zweiten (rechten) Tabelle alle Datensätze und von der ersten Tabelle jene, deren Datenwerte mit denen der zweiten Tabelle übereinstimmen
FULL JOIN	Liefert aus beiden Tabellen jeweils alle Datensätze
WHERE	Bedingung, nach der Datensätze ausgewählt werden sollen
WHERE EXISTS (subquery)	Die Bedingungen EXISTS prüft, ob die Suchbedingung einer Unterabfrage mindestens eine Zeile zurückliefert. NOT EXIST negiert die Bedingung.
WHERE NOT EXISTS (subquery)	
GROUP BY Spaltenname1 [, Spaltenname2, ...]	Gruppierung (Aggregation) nach Inhalt des genannten Feldes
ORDER BY Spaltenname1 [, Spaltenname2, ...] ASC DESC	Sortierung nach Inhalt des genannten Feldes oder der genannten Felder ASC: aufsteigend; DESC: absteigend

Fortsetzung ->

SQL-Syntax (Auszug) – Fortsetzung

Syntax	Beschreibung
Datenmanipulation	
DELETE FROM Tabellenname	Löschen von Datensätzen in der genannten Tabelle
UPDATE Tabellenname SET	Aktualisiert Daten in Feldern einer Tabelle
INSERT INTO Tabellenname VALUES (Wert für Spalte 1 [, Wert für Spalte 2, ...]) oder SELECT ... FROM ... WHERE	Fügt Datensätze in die genannte Tabelle, die entweder mit festen Werten belegt oder Ergebnis eines SELECT-Befehls sind
Aggregatfunktionen	
AVG(Spaltenname)	Ermittelt das arithmetische Mittel aller Werte im angegebenen Feld
COUNT(Spaltenname *)	Ermittelt die Anzahl der Datensätze mit Nicht-NULL-Werten im angegebenen Feld oder alle Datensätze der Tabelle (dann mit Operator *)
SUM(Spaltenname Formel)	Ermittelt die Summe aller Werte im angegebenen Feld oder der Formelergebnisse
MIN(Spaltenname Formel)	Ermittelt den kleinsten aller Werte im angegebenen Feld
MAX (Spaltenname Formel)	Ermittelt den größten aller Werte im angegebenen Feld
Funktionen	
LEFT(Zeichenkette, Anzahlzeichen)	Liefert Anzahlzeichen der Zeichenkette von links.
RIGHT(Zeichenkette, Anzahlzeichen)	Liefert Anzahlzeichen der Zeichenkette von rechts.
CURRENT	Liefert das aktuelle Datum mit der aktuellen Uhrzeit
CONVERT(time,[DatumZeit])	Liefert die Uhrzeit aus einer DatumZeit-Angabe
DATE(Wert)	Wandelt einen Wert in ein Datum um
DAY(Datum)	Liefert den Tag des Monats aus dem angegebenen Datum
MONTH(Datum)	Liefert den Monat aus dem angegebenen Datum
TODAY	Liefert das aktuelle Datum
WEEKDAY(Datum)	Liefert den Tag der Woche aus dem angegebenen Datum
YEAR(Datum)	Liefert das Jahr aus dem angegebenen Datum
DATEADD(Datumsteil, Intervall, Datum)	Fügt einem Datum ein Intervall (ausgedrückt in den unter Datumsteil angegebenen Einheiten) hinzu
DATEDIFF(Datumsteil, Anfangsdatum, Enddatum) Datumsteile: DAY, MONTH, YEAR	Liefert Enddatum-Startdatum (ausgedrückt in den unter Datumsteil angegebenen Einheiten)
Operatoren	
AND	Logisches UND
LIKE	Überprüfung von Textattributen auf Gleichheit, Verwendung von Platzhaltern möglich.
NOT	Logische Negation
OR	Logisches ODER
=	Test auf Gleichheit
>, >=, <, <=, <>	Test auf Ungleichheit
*	Multiplikation
/	Division
+	Addition, positives Vorzeichen
-	Subtraktion, negatives Vorzeichen

Stand 2017-03-18

S

- ac) Erstellen einer Liste aller Kunden mit dem Umsatz je bezogenem Artikel.
Ausgegeben werden sollen: Kunde_ID und Firma des Kunden sowie Bezeichnung und Umsatz der Artikel.

8 Punkte

Korrekturrand

Beispiel

762762	ABC-Gesund AG	Aspiran forte	28.600,00
762762	ABC-Gesund AG	Cardioshock 2.0	17.300,00
...			
872346	Zeppelin-Pharma GmbH	Cardioshock 2.0	38.500,00
...			
138787	FeelGood-Partner OHG	Katerstop	19.800,00
...			

- b) Die Datenbank soll erweitert werden. Dazu sollen Sie für folgende Aufgaben SQL-Anweisungen erstellen.
- ba) Erstellen einer Tabelle Artikelgruppe mit einem ganzzahligen Primärschlüsselattribut und der Artikelgruppenbezeichnung.

3 Punkte

- bb) Einbinden der neuen Tabelle Artikelgruppe in die Datenbank.

3 Punkte

PRÜFUNGSZEIT – NICHT BESTANDTEIL DER PRÜFUNG!

Wie beurteilen Sie nach der Bearbeitung der Aufgaben die zur Verfügung stehende Prüfungszeit?

Sie hätte kürzer sein können.

Sie war angemessen.

Sie hätte länger sein müssen.



