Inhalt

2013W AP IT GA1 Lösungen 2. Handlungsschritt (25 Punkte) 5

2013W AP IT GA2 Lösungen 4. Handlungsschritt (25 Punkte) 7

2014W AP IT GA2 Lösungen 5. Handlungsschritt (25 Punkte) 10

2015S AP IT GA1 Lösungen 3. Handlungsschritt (25 Punkte) 14

2015W AP IT GA2 Lösungen 4. Handlungsschritt (25 Punkte) 16

2017S AP IT GA2 Lösungen 4. Handlungsschritt (25 Punkte) 20

2017W AP IT GA2 Lösungen 3. Handlungsschritt (25 Punkte) 22

2018W AP IT GA2 Lösungen 4. Handlungsschritt (25 Punkte) 26

2019S AP IT GA2 Lösungen 4. Handlungsschritt (25 Punkte) 28

2019W AP IT GA2 Lösungen 4. Handlungsschritt (25 Punkte) 15 Punkte 30

2020S AP IT GA1 Lösungen 4. Handlungsschritt (25 Punkte) 32

2020W AP IT GA2 Lösungen 4. Handlungsschritt (25 Punkte) 34

2021W AP IT GA1 Lösungen 5. Handlungsschritt (25 Punkte) 37

2021W AP IT GA2 Lösungen 5. Handlungsschritt (25 Punkte) 39

2022S AP IT GA2 Lösungen 4. Handlungsschritt (25 Punkte) 43

Abschlussprüfung Teil 1 Herbst 2022 4. Aufgabe (24 Punkte) 46

ER- Modelle
Entity Relationship Modell

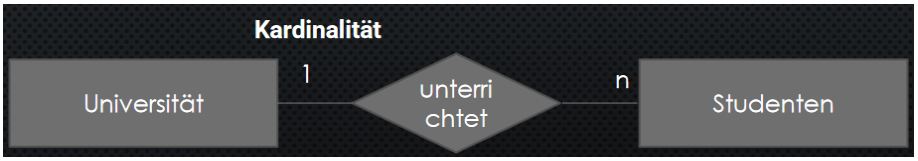
Es gibt drei Grundelemente in einem ER-Diagramm: ***Entität****,* ***Attribut****,* ***Beziehung***

Entität
Eine Entität kann eine Person, ein Ort, ein Ereignis oder ein Objekt sein, das für ein bestimmtes System relevant ist. Ein Schulsystem kann zum Beispiel Studenten, Lehrer, Hauptkurse, Fächer, Gebühren und andere Elemente umfassen. Entitäten werden in ER-Diagrammen durch ein Rechteck dargestellt.

Attribut
Ein Attribut ist eine Eigenschaft, ein Merkmal oder eine Charakteristik einer Entität, einer Beziehung oder eines anderen Attributs. Eine Entität kann so viele Attribute wie nötig haben. In der Zwischenzeit können Attribute auch ihre eigenen spezifischen Eigenschaften haben. Beispielsweise kann das Attribut “Kundenadresse” die Attribute Nummer, Straße, Ort und Bundesland haben. Diese werden als zusammengesetzte Attribute bezeichnet. Beachten Sie, dass einige Top-Level ER-Diagramme der Einfachheit halber keine Attribute zeigen. In denjenigen, die dies tun, werden die Attribute jedoch durch ovale Formen dargestellt.

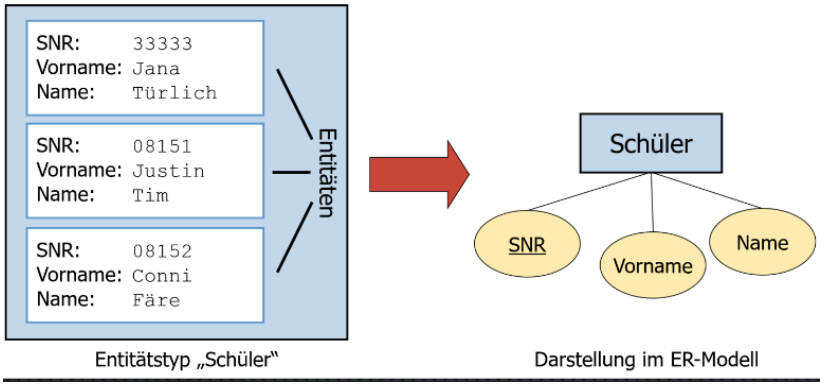
Beziehung
Eine Beziehung beschreibt, wie Entitäten interagieren. Zum Beispiel kann die Entität “Universität” mit der Entität “Studenten” durch die Beziehung “unterrichtet” verbunden sein. Beziehungen werden durch Rauten dargestellt und mit Verben beschriftet.

Kardinalität
Diese beiden definieren Beziehungen zwischen Entitäten weiter, indem sie die Beziehung in den Kontext von Zahlen stellen. Die Beziehung folgt in diesem Fall einem “one to many”-Modell. Es gibt eine Reihe von Notationen, die zur Darstellung der Kardinalität in ER-Diagrammen verwendet werden. Das folgende Beispiel verwendet CHEN zur Darstellung der Kardinalität. Dies benutzt auch die IHK.

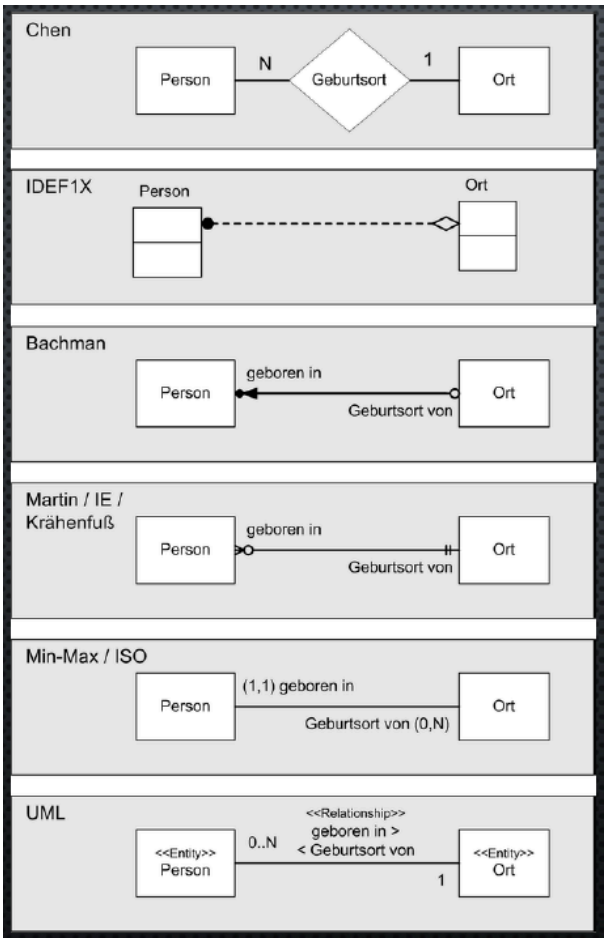


Entitätstyp, Attribute und Entitäten – inkl. Primärschlüssel

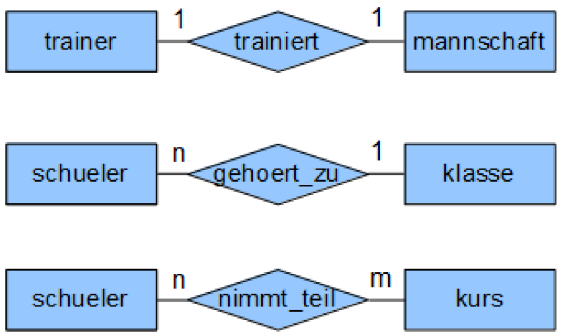
- Als *Entität* bezeichnet man ein Objekt, das eindeutig identifiziert werden kann.
- Alle Entitäten, die "gleichartig" in ihren Eigenschaften und ihrem Verhalten sind, werden zu *Entitätstypen* zusammengefasst.
- Die Eigenschaften (*Attribute*) werden mit Ellipsen an die Entitätstypen angehängt. Datentypen spielen keine Rolle. Attribute, die eine Entität eindeutig identifizieren können, werden unterstrichen.



IHK Darstellung



Kardinalitäten und Beziehungen



Mit Hilfe von Kardinalitäten kann man Beziehungstypen genauer beschreiben. Für jeden Beziehungstyp gibt man an, wie viele Entitäten mit jeweils einer Entität in Verbindung stehen.

BEST PRACTICE METHODE

1. Identifizieren Sie alle Entitäten im System. Eine Entität sollte nur einmal in einem bestimmten Diagramm erscheinen. Erstellen Sie Rechtecke für alle Entitäten und benennen Sie sie korrekt.
2. Identifizieren Sie Beziehungen zwischen Entitäten. Verbinden Sie sie mit einer Linie und fügen Sie in der Mitte eine Raute ein, die die Beziehung beschreibt.
3. Attribute für Entitäten hinzufügen: Geben Sie aussagekräftige Attributnamen an, damit sie leicht verständlich sind.
4. Geben sie die Kardinalitäten an: Benutzen Sie die Chen-Notation und filtern sie die Informationen aus der Aufgabenstellung

2013W AP IT GA1 Lösungen 2. Handlungsschritt (25 Punkte)

Die Anmeldung an den Beruflichen Schulen N-Stadt soll in Zukunft über das Internet möglich sein. Hierfür wurde ein Pilotprojekt bei einer Schule mit einem Webserver und einer Datenbank eingerichtet. Zukünftige Schüler sollen sich über eine verschlüsselte Webseite für eine Schule und eine gewünschte Fachrichtung anmelden.

a) Erweitern Sie das angegebene Datenbankmodel, sodass in den Schulen mehrere Fachrichtungen für die Anmeldung angeboten werden können.

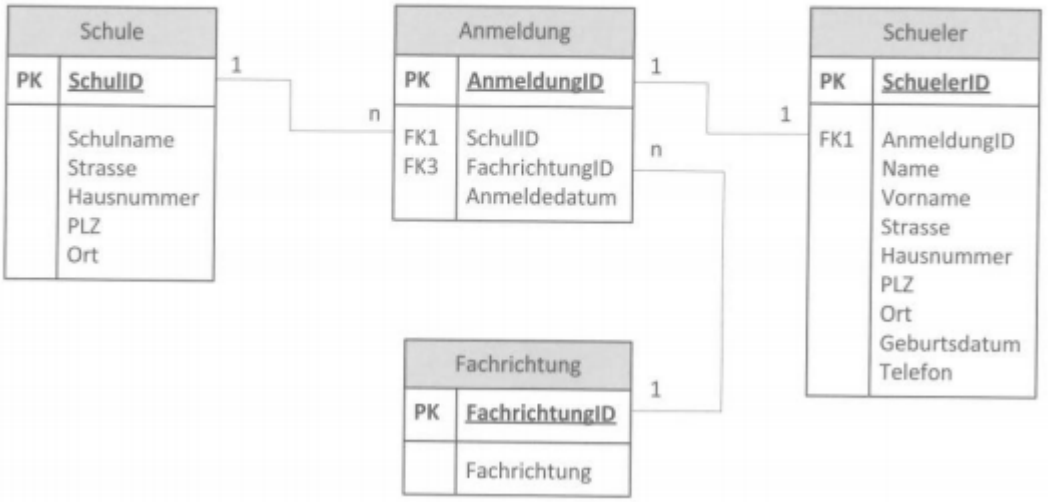
Kennzeichnen Sie die Primärschlüssel mit PK und die Fremdschlüssel mit FK und unterstreichen Sie diese.

Zeichnen Sie Kardinalitäten ein. 10 Punkte

Lösung

a) 10 Punkte

2 Punkte, 4 x 0,5 Punkte je Primärschlüssel
3 Punkte, 3 x 1 Punkt je Fremdschlüssel
3 Punkte, Kardinalitäten
2 Punkte, Entität Fachrichtungen



b) Die Daten sollen verschlüsselt übertragen werden. Es wird die symmetrische und asymmetrische Verschlüsselung diskutiert.
Erläutern Sie die beiden Verschlüsselungsverfahren und nennen Sie jeweils einen Vorteil gegenüber dem jeweiligen anderen Verfahren.
ba) Symmetrische Verschlüsselung 4 Punkte

Erläuterung:
Sender und Empfänger kennen und benutzen für die Ver- und Entschlüsselung einer Nachricht den gleichen geheimen Schlüssel.
Vorteil: Schnellere Ver- und Entschlüsselung
Andere Lösungen sind möglich.

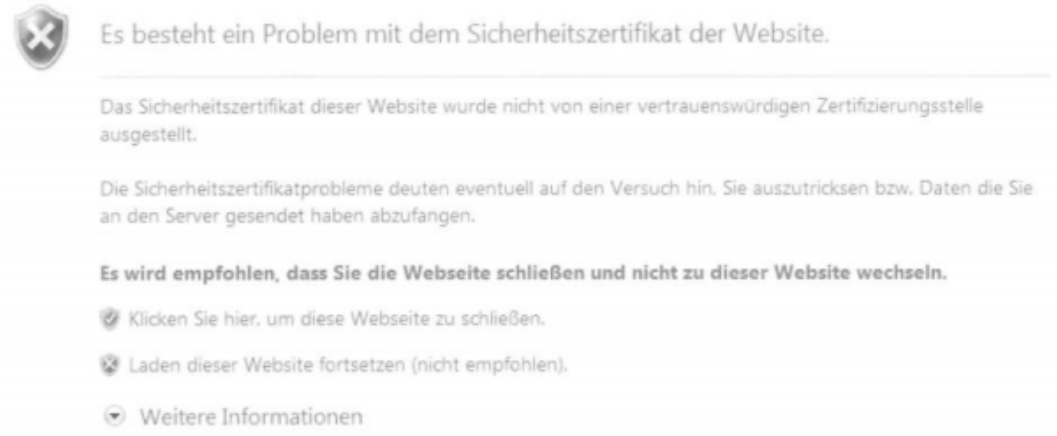
bb) Asymmetrische Verschlüsselung 4 Punkte

Erläuterung:
Der Empfänger besitzt einen öffentlichen und einen privaten Schlüssel (Public und Private Key). Der Empfänger schickt den öffentlichen Schlüssel an den Sender, der damit die zu übertragenden Daten verschlüsselt. Nur der Empfänger kann diese Daten mit seinem privaten Schlüssel wieder entschlüsseln.
Vorteil: Der Schlüsselaustausch über einen unsicheren Kanal ist möglich.
Andere Lösungen sind möglich

c) Der Zugriff auf die Webseite für die Anmeldung erfolgt über das Protokoll HTTPs.
ca) Erläutern Sie die Aufgabe von HTTPs und nennen Sie den Port, den dieses Protokoll standardmäßig verwendet. 3 Punkte

HTTPs verschlüsselt den http-Datenverkehr über den Port 443.

cb) Beim Aufruf der Schulwebseite https://anmeldung.schulen-nstadt.de erhalten Sie in Ihrem Browser folgende Meldung.



Erläutern Sie, warum dieser Sicherheitshinweis erscheint. 4 Punkte
Bei HTTPs authentifiziert sich der Server gegenüber dem Client mit seinem Zertifikat. Dieses enthält die digitale Signatur der CA. Diese Signatur prüft der Client mit dem Public Key der CA. Da der Client diesen Public Key nicht hat, kann er auch die Korrektheit des Zertifikats nicht bestätigen. — Sicherheitswarnung im Browser

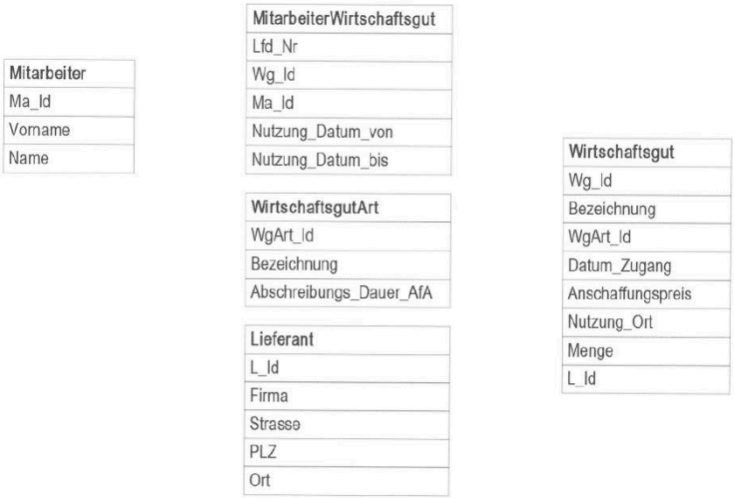
2013W AP IT GA2 Lösungen 4. Handlungsschritt (25 Punkte)

Die IT-Solution GmbH soll eine Anwendung zur Verwaltung des Hotelinventars (z. B. Computer, Stühle) erstellen. Die Daten zu diesen Wirtschaftsgütern sollen in einer relationalen Datenbank gespeichert werden.

a) Folgende Sachverhalte sollen durch das Datenmodell abgebildet werden:

- Ein Wirtschaftsgut kann nur einer Wirtschaftsgutart zugeordnet werden.
- Ein Wirtschaftsgut wird von einem Lieferanten geliefert.
- Ein Wirtschaftsgut kann von mehreren Mitarbeitern genutzt werden.
- Ein Lieferant kann mehrere verschiedene Wirtschaftsgüter liefern.
- Ein Mitarbeiter kann mehrere Wirtschaftsgüter nutzen.

Folgende Tabellen wurden für das relationale Datenmodell bereits angelegt.



aa) Vervollständigen Sie diesen Entwurf, indem Sie

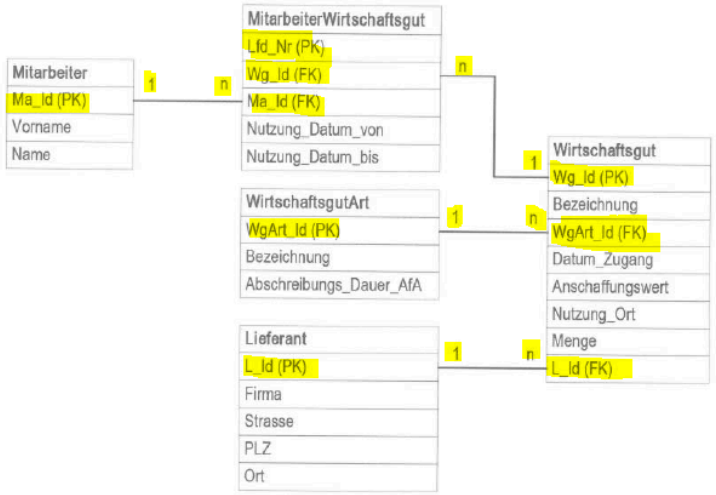
- die Primärschlüssel mit PK und die Fremdschlüssel mit FK kennzeichnen.
- die Beziehungen einzeichnen und deren Kardinalitäten angeben. 15 Punkte

aa) 15 Punkte

5 Punkte, 5 x 1 Punkt je Primärschlüssek (PK)

2 Punkte, 4 x 0,5 Punkte je Fremdschlüssel (FK)

8 Punkte, 4 x 2 Punkte je Beziehung und deren Kardinalität



ab) Erläutern Sie die Notwendigkeit der Tabelle *Mitarbeiterwirtschaftsgut*. 1 Punkt

Zwischentabelle zur Auflösung einer m : n - Beziehung in zwei 1 : n – Beziehungen

b) Viele Wirtschaftsgüter, die für das Hotel gekauft wurden, werden zunächst im Zentrallager der Rügen-Hotel AG angenommen, bevor sie nach der Inventarisierung von dort in das neue Hotel nach Badewiek ausgeliefert werden. - Der Vorgang der Inventarisierung wird wie folgt beschrieben:

„Nach Annahme eines Wirtschaftsgutes im Lager wird dieses in das Inventar (Bestandsverzeichnis aller Vermögensgegenstände) des Hotels aufgenommen. Dazu werden von der Abteilung „Lager" alle Daten des Wirtschaftsgutes bis auf den Anschaffungswert, der von der Abteilung „Rechnungswesen" (Rewe) ergänzt wird, erfasst.

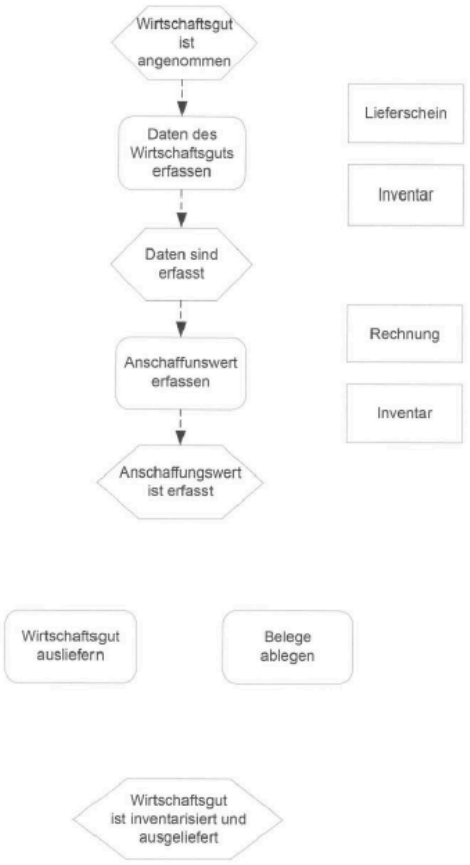
Das Wirtschaftsgut wird vom Lager ausgeliefert und die Belege werden im Rechnungswesen abgelegt. Danach ist die Inventarisierung abgeschlossen."

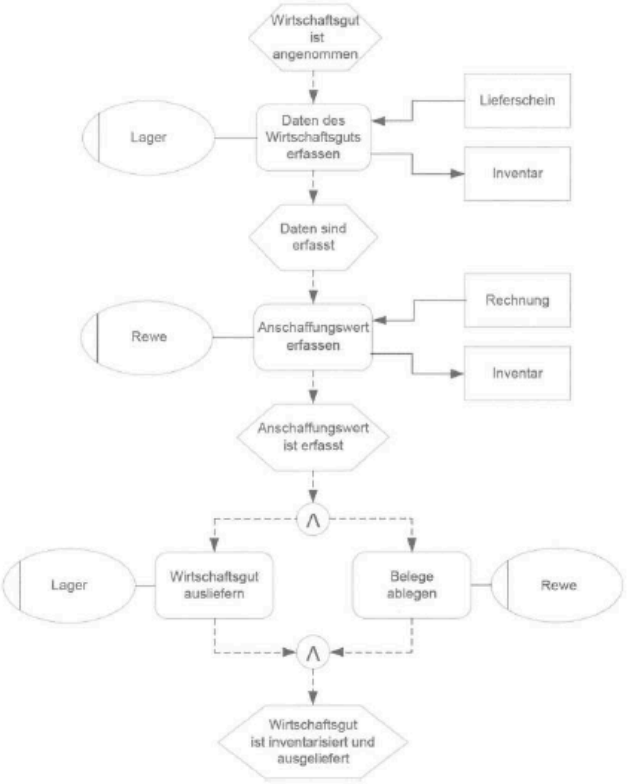
Die Inventarisierung soll in einer Ereignisgesteuerten Prozesskelte (EPK) dargestellt werden. Vervollständigen Sie den nebenstehenden Entwurf der EPK, indem Sie die beteiligten Organisationseinheiten sowie die Kontrollflüsse, Informationsflüsse und Konnektoren einzeichnen. 9 Punkte

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

Sinnbilder EPK-Technik (Auswahl)

⬡	**Ereignis:** Eingetretener Zustand, der den weiteren Ablauf festlegt
▢	**Funktion:** Betrieblicher Vorgang, der einen Eingangszustand in einen Zielzustand umwandelt
◯	**Organisationseinheit:** Benennt die Abteilung, der eine Funktion oder ein Ereignis zugeordnet ist
▭	**Objekt:** Ein Objekt kann ein Informationsobjekt, ein Material, eine Ressource oder ein Produkt sein.
- - - - →	**Kontrollfluss:** Gerichteter, zusammenhängender Graph, dessen Knoten Ereignisse, Funktionen und Verknüpfungs-operatoren sind
→ ←	**Informationsfluss:** Gibt an, dass anlässlich einer Funktion Informationen von einem Informationsobjekt gelesen oder auf ein Informationsobjekt geschrieben werden **Material-/Ressourcenfluss:** Gibt an, dass in einer Funktion Material bzw. Ressourcen verbraucht werden
——	**Zuordnung:** Gibt an, welche Organisationseinheit oder welche Objekte einer Funktion zugeordnet sind
∧	**Konnektor „Und":** Eine Funktion wird ausgeführt, wenn mehrere Ereignisse eingetreten sind. oder Nach einer Funktion treten mehrere Ereignisse ein. oder Ein Ereignis tritt ein, nachdem alle direkt vorangestellten Funktionen ausgeführt wurden.
XOR	**Konnektor „Exklusives Oder":** Eine Funktion wird ausgeführt, wenn genau ein Ereignis von mehreren eingetreten ist. oder Nach einer Funktion tritt genau eins von mehreren Ereignissen ein. oder Ein Ereignis tritt ein, nachdem eine von mehreren direkt vorangestellten Funktionen ausgeführt wurde.
V	**Konnektor „Offenes Oder":** Eine Funktion wird ausgeführt, wenn mindestens ein Ereignis von mehreren eingetreten ist. oder Nach einer Funktion tritt mindestens eins von mehreren Ereignissen ein. oder Ein Ereignis tritt ein, nachdem mindestens eine von mehreren direkt vorangestellten Funktionen ausgeführt wurde.





2 Punkte,	4 x 0,5 Punkte je Organisationseinheit und deren Zuordnung
2 Punkte,	2 x 1 Punkt je Konnektor
2 Punkte	für Informationsflüsse
3Punkte	für Kontrollflüsse

2014W AP IT GA2 Lösungen 5. Handlungsschritt (25 Punkte)
Die IT-System GmbH soll für die One AG eine Inventardatenbank zur Verwaltung aller Wirtschaftsgüter erstellen. Folgender Teilentwurf liegt bereits vor:

a) Die Datenbank soll so erweitert werden, dass Wirtschaftsgütern Räume zugeordnet werden können. Bedingungen:

- Die zeitliche Nutzung von Wirtschaftsgütern in verschiedenen Räumen soll dokumentiert werden, z. B.:

Inventar_ID	Datum_von	Datum_bis	Raum_ID
1234	02.01.2014	30.11.2014	R22
1234	01.12.2014	31.12.2014	R03

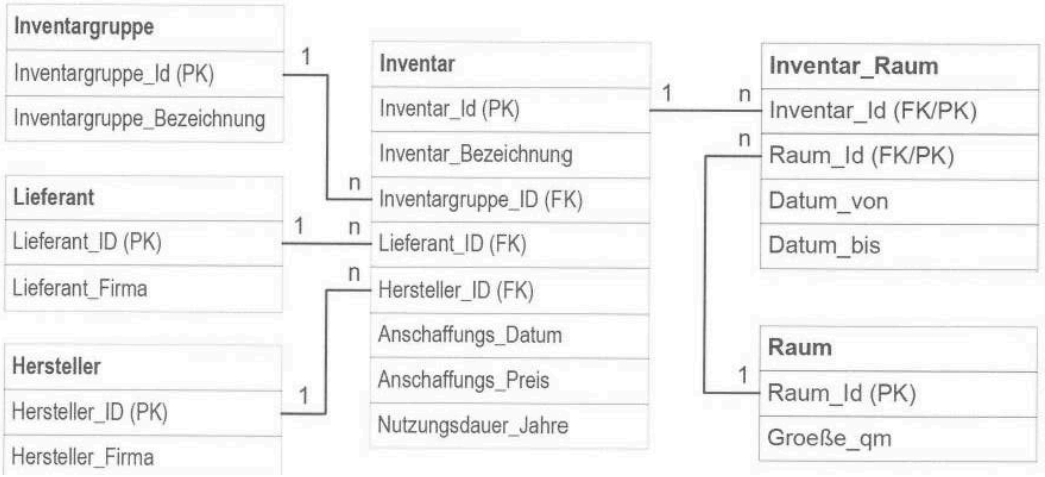
- Einem Raum können ein bis mehrere Wirtschaftsgüter oder kein Wirtschaftsgut zugeordnet werden.
- Zu jedem Raum soll die Flächenangabe in Quadratmetern gespeichert werden.

Erweitern Sie das obige Datenmodell der Anforderung entsprechend unter Beachtung der 3. Normalform, indem Sie

- alle erforderlichen Tabellen erstellen,
- in die Tabellen alle erforderlichen Attribute eintragen,
- die Primärschlüssel-Attribute mit PK, Fremdschlüssel-Attribute mit FK und die Attribute zusammengesetzter Primärschlüssel mit FK/PK kennzeichnen,
- die Beziehungen mit deren Kardinalitäten einzeichnen. 8 Punkte

a) 8 Punkte

2 Punkte	2 x 1 Punkt für Tabelle und Zwischentabelle
3 Punkte	6 x 0,5 Punkte je Attribut
1 Punkt	2 x 0,5 Punkte je Verbindung
2 Punkte	2 x 1 Punkt je Kardinalität



b) Sie sollen für folgende Aufgaben folgende SQL-Anweisungen erstellen.
Hinweis:
Die SQL-Aufgaben beziehen sich nur auf das vorgegebene Datenmodell, nicht auf die Erweiterung aus a).

ba) Für einen neuen Tisch sollen folgende Daten in die Datenbank eingetragen werden.

Inventar_ID:	2184
Inventar_Bezeichnung:	CutEdge
Inventargruppe_ID:	G4
Lieferant_ID:	L15
Hersteller_ID:	H178

Erstellen Sie die entsprechende SQL-Anweisung. 3 Punkte

INSERT INTO Inventar (Inventar_ID, Inventar_Bezeichnung, Inventargruppe_ID, Lieferant_ID, Hersteller_ID)
VALUES (2184, „CutEdge", ,,G4", ,,L15", „H178")

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

SQL-Syntax (Auszug)

Syntax	Beschreibung
Tabellen	
CREATE TABLE Tabellenname(Feldname1 Datentyp [,Feldname2 Datentyp]) Primärschlüssel, Fremdschlüssel)	Erzeugt eine neue leere Tabelle mit der beschriebenen Struktur
CHARACTER	Textdatentyp
DECIMAL	Numerischer Datentyp (Festkommazahl)
DOUBLE	Numerischer Datentyp (Doppelte Präzision)
INTEGER	Numerischer Datentyp (Ganzzahl)
DATE	Datum (Format DD.MM.YYYY)
PRIMARY KEY	
FOREIGN KEY (Feldname) **REFERENCES**	
DROP TABLE Tabellenname	Löscht eine Tabelle
Befehle, Klauseln, Attribute	
SELECT * | Feldname1 [, Feldname2, ...]	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
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 Beispiel: FROM Verkaeufer LEFT JOIN Kunde ON Verkaeufer.Ver_ID = Kunde.Ver_ID
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 Beispiel: FROM Verkaeufer RIGHT JOIN Kunde ON Verkaeufer.Ver_ID = Kunde.Ver_ID
FULL JOIN	Liefert aus beiden Tabellen jeweils alle Datensätze
WHERE	Bedingung, nach der Datensätze ausgewählt werden sollen Beispiel: WHERE name = 'Maier'
GROUP BY Feldname1 [,Feldname2, ...]	Gruppierung (Aggregation) nach Inhalt des genannten Feldes Beispiel: GROUP BY name, vorname
ORDER BY Feldname1 [,Feldname2, ...] **ASC | DESC**	Sortierung nach Inhalt des genannten Feldes oder der genannten Felder ASC: aufsteigend; DESC: absteigend Beispiel: ORDER BY name ASC
Datenmanipulation	
DELETE FROM Tabellenname	Löschen von Datensätzen in der genannten Tabelle
UPDATE Tabellenname **SET**	Aktualisiert Daten in Feldern einer Tabelle Beispiel: UPDATE Artikel SET(Preis=10.00)
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 Beispiele: INSERT INTO kunde VALUES 56532, 'Martina', 'Schmitz', '12345', 'Berlin' INSERT INTO kunde SELECT * FROM vertrag WHERE stadt='Berlin'
Funktionen	
AVG(Feldname)	Ermittelt das arithmetische Mittel aller Werte im angegebenen Feld
COUNT(Feldname | *)	Ermittelt die Anzahl der Datensätze mit Nicht-NULL-Werten im angegebenen Feld oder alle Datensätze der Tabelle (dann mit Operator *)
SUM(Feldname|Formel)	Ermittelt die Summe aller Werte im angegebenen Feld oder der Formelergebnisse Beispiel: SELECT SUM(preis)
MIN(Feldname|Formel)	Ermittelt den kleinsten aller Werte im angegebenen Feld Beispiel: SELECT MIN(preis)
MAX (Feldname|Formel)	Ermittelt den größten aller Werte im angegebenen Feld Beispiel: SELECT MAX(preis)

Datumsfunktionen	
CURRENT, CURDATE()	Liefert das aktuelle Datum mit der aktuellen Uhrzeit
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
Operatoren	
AND	Logisches UND
NOT	Logische Negation
OR	Logisches ODER
=	Test auf Gleichheit
>, >=, <, <=, < >	Test auf Ungleichheit
*	Multiplikation
/	Division
+	Addition, positives Vorzeichen
-	Subtraktion, negatives Vorzeichen

bb) Der Rechtsnachfolger des Lieferanten „Sitzgut GmbH" ist die „Tisch&Stuhl GmbH".
Die Firma soll in der Datenbank aktualisiert werden.
Erstellen Sie eine entsprechende SQL-Anweisung. 3 Punkte

```
UPDATE Lieferant
SET Firma = „Tisch&Stuhl GmbH"
WHERE Firma = „Sitzgut GmbH";
```

bc) Es soll eine Liste mit den Lieferanten und deren Umsätzen im Jahr 2014 erstellt werden, aufsteigend sortiert nach Umsatz (Umsatz = Summe Anschaffungspreise).
Erstellen Sie eine entsprechende SQL-Anweisung. 5 Punkte
Beispiel

L26	Scholz GmbH	27.281,62
L17	Kumpel AG	21.287,98
L36	Katz&Maus GmbH	19.750,66

```
SELECTL.Lieferant_ID, L.Firma, SUM(I.Anschaffungspreis) AS Umsatz
FROMLieferant L, Inventar I
WHEREL.Lieferant_ID = I.Lieferant_IDAND YEAR(I.Anschaffungs_Datum) = 2014
GROUP BYL.Lieferant_ID, L_Firma;
ORDER BYUmsatzASC
```

bd) Es soll das Datum „Erstkontakt“ zur Tabelle Lieferant hinzugefügt werden.
Erstellen Sie eine entsprechende SQL-Anweisung. 3 Punkte

```
ALTER TABLE Lieferant
ADD COLUMN Erstkontakt DATE;
```

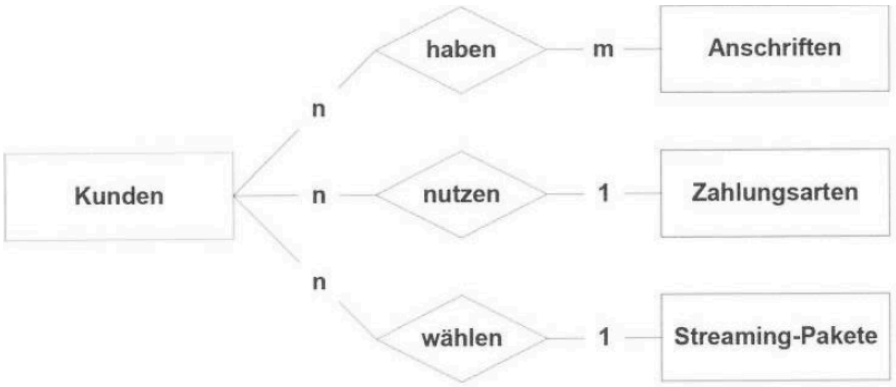
be) Es soll eine Liste aller inventarisierten Wirtschaftsguter erstellt werden, deren Nutzungsdauer im Jahr 2015 endet.
Erstellen Sie die entsprechende SQL-Anweisung. 3 Punkte

```
SELECTInventar_ID,Inventar_Bezeichnung
FROM Inventar
WHERE YEAR(Anschaffungs_Datum) + Nutzungsdauer_Jahre - 1 = 2015
```

2015S AP IT GA1 Lösungen 3. Handlungsschritt (25 Punkte)
Die TeNi GmbH bietet ihren Kunden verschiedene Streaming-Pakte für Filme an, z. B. „Komödien", „Action und Thriller", „Kinderfilme" oder „Alle Filme".
Die IT-Revolution AG soll für die TeNi GmbH einen Internetshop für Streaming-Angebote entwickeln. Im System sollen folgende Informationen gespeichert werden.

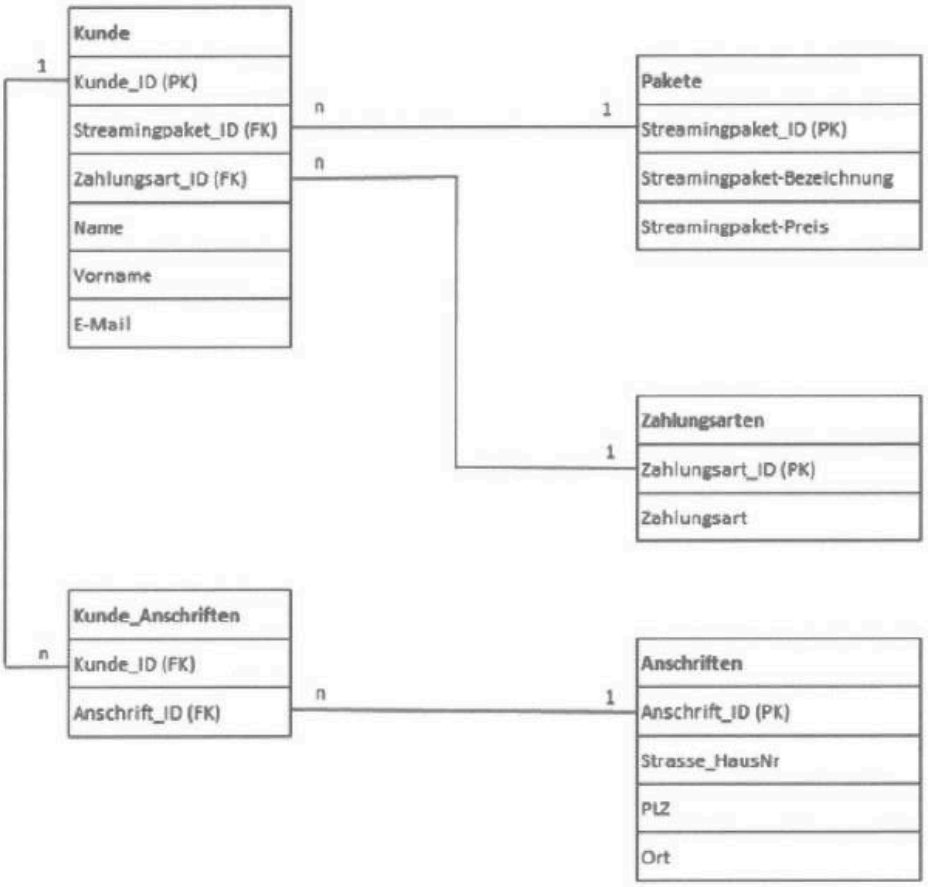
- Kunden-ID
- Name und Vorname des Kunden
- E-Mail des Kunden
- Anschriften des Kunden (Lieferadresse, Rechnungsadresse), jeweils mit Straße einschließlich Hausnummer, PLZ und Ort
- Zahlungsart (Kunde kann nur eine Zahlungsart aus mehreren auswählen)
- Vom Kunden bestelltes Streaming-Paket (Jeder Kunde kann sich nur für <u>ein</u> Streaming-Paket entscheiden.)
- Streaming-Paket-ID
- Streaming-Paket-Bezeichnung
- Streaming-Paket-Preis

Es wurde bereits folgendes ER-Diagramm entworfen:



a) Erstellen Sie anhand der Angaben ein relationales Datenmodell in der dritten Normalform.
18 Punkte
Hinweis:
Erstellen Sie alle erforderlichen Tabellen.
Tragen Sie alle erforderlichen Attribute ein.
Kennzeichnen Sie Primärschlüssel mit (PK) und Fremdschlüssel mit (FK).
Tragen Sie die Beziehungen zwischen den Tabellen ein und geben Sie die Kardinalitäten an.

a) 18 Punkte
4 Punkte, 4 x 1 Punkt je Beziehung
4 Punkte, 4 x 1 Punkt je Primärschlüssel
4 Punkte, 4 x 1 Punkt je Fremdschlüssel
6 Punkte, 6 x 1 Punkt je Nichtschlüsselattribut



b)Erläutern Sie den Zweck eines Primärschlüssels in einer relationalen Datenbank.3 Punkte

- Attribut zur eindeutigen Kennzeichnung eines Datensatzes, darf daher innerhalb einer Tabelle nur einmal Vorkommen
- Ermöglicht die Kombination von Tabellen über Primär-Fremdschlüssel-Beziehungen

c) Um den Datenschutz zu gewährleisten, soll die Datenbank verschlüsselt werden. Erläutern Sie, warum es nicht sinnvoll ist, alle Daten einer Datenbank zu verschlüsseln.
4 Punkte
Es ist sinnvoll, nur sicherheitsrelevante Daten zu verschlüsseln, da durch die Verschlüsselung die Rechenlast des Systems stark erhöht wird. Dadurch wird die Zeit für Abfragen und Einträge verlängert.

2015W AP IT GA2 Lösungen 4. Handlungsschritt (25 Punkte)

Die Sitec GmbH soll für die W-HausAG ein Kassensystem entwickeln.

a) Die auf dem Kassenzettel grau hinterlegten Daten sollen in einer Datenbank gespeichert werden. 15 Punkte

SB-Warenhaus
W-Haus AG
Warenhaus Allee 1
99999 Bstadt
USt-ID: DE123 456 789
Telefon 01234 – 567-0
E-Mail: sb-warenhaus.bstadt@warenhaus.de

Kassenbon-Nr.: **1227890098**
Kassenbon-Datum: **29.09.2015**

Position-Nr.	Menge	Artikel-Nr.	Bezeichnung	Einzelpreis	Gesamtpreis	MwSt-ID
1	2	786327	Bratpfanne	70,00	140,00	A
2	1	892348	Kochbuch	12,90	12,90	B
3	3	762734	Schürze	25,00	75,00	A

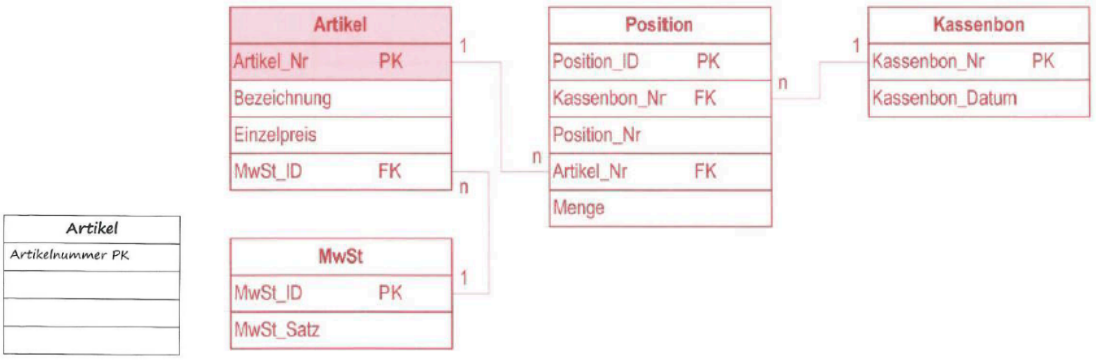
Endsumme 227,90

MwSt %	MwSt	+	Netto	=	Brutto
A 19 %	34,33		180,67		215,00
B 7 %	0,84		12,06		12,90
Summe	35,17		192,73		227,90

Vervollständigen Sie das nebenstehende Datenmodell in der 3. Normalform, indem Sie

- die noch erforderlichen Tabellen mit den erforderlichen Attributen entwerfen (siehe grau markierte Daten).
- die Primärschlüssel-Attribute mit PK und die Fremdschlüssel-Attribute mit FK kennzeichnen.
- die Beziehungen zwischen den Tabellen einzeichnen und deren Kardinalitäten angeben.

a) 15 Punkte
Es gibt verschiedene Lösungen, daher können die folgenden Punktwerte nur Richtwerte sein. Bepunktung für die folgende Lösung:
3 Punkte, 3 x 1 Punkt je Tabelle
3 Punkte, 6 x 0,5 Punkte je Attribut
3 Punkte, 6 x 0,5 Punkte je Schlüsselattribut
6 Punkte, 3 x 2 Punkte je Beziehung mit Kardinalität



Hinweis für Prüfer:

Es sind unterschiedliche Lösungen möglich, z. B.

- Einzelpreis auch Attribut der Tabelle R_Position; Diese Redundanz kann sinnvoll sein, wenn der Überlegung Rechnung getragen wird, dass sich der Einzelpreis über die Zeit ändert.
- Verwendung eines aus Rechnungsnummer und Positionsnummer zusammengesetzten Primärschlüssels statt einer eindeutigen Positions_ID
- Je nach Lösung kann sich eine andere Punkteverteilung ergeben.

b) Die Sitec GmbH soll die Methode Endsumme entwickeln, welche für einen Kassenbon die Endsumme und die in der Endsumme enthaltene Mehrwertsteuern ausgibt (siehe grau hinterlegte Beträge in folgendem Beispiel).

Position	Menge	Artikel-Nr.	Bezeichnung	Einzelpreis	Gesamtpreis	
1	2	78632746	Bratpfanne	70,00	140,00	A
2	1	89234890	Kochbuch	12,90	12,90	B
3	3	76273400	Schürze	25,00	75,00	A

Endsumme 227,90

MwSt %	MwSt	+	Netto	=	Brutto
A 19 %	34,33		180,67		215,00
B 7 %	0,84		12,06		12,90
Summe	35,17		192,73		227,90

Die Positionen eines Verkaufs liegen in einer Tabelle vor, Beispiel:

Position	Menge	Artikelnummer	Bezeichnung	Einzelpreis	Gesamtpreis	MwSt_ID
1	2	761236129	Bratpfanne	70,00	140,00	A
2	1	989932980	Kochbuch	12,90	12,90	B
3	3	878732498	Schürze	25,00	75,00	A

Hinweis:

- Einzelpreis und Gesamtpreis sind Bruttowerte und enthalten die Mehrwertsteuer
- Die MwSt_ID gibt den Mehrwertsteuersatz an: A = 19 % (z. B. Kleidung), B = 7 % (z. B. Bücher).
- Die übrigen auf dem Kassenbon abgebildeten berechneten Werte werden von der Methode noch nicht ermittelt und sind nicht Gegenstand der Aufgabe.

Für die Methode Endsumme wurde bereits nebenstehendes Struktogramm erstellt, was sich als fehlerhaft erwies.

Prüfen Sie das Struktogramm, markieren Sie fünf Fehler mit den Nummern 1 bis 5 und korrigieren Sie die Fehler durch handschriftlichen Eintrag im Struktogramm. 10 Punkte

Methode Endsumme

```
double : endsumme
integer:  mwst_A_Satz, mwst_B_Satz
integer : mwst_A_gesamt, mwst_B_gesamt, mwst_gesamt


endsumme := 0,00

mwst_A_Satz := 19,0
mwst_B_Satz :=  7

mwst_Betrag := 0,00
mwst_A_gesamt := 0,00
mwst_B_gesamt := 0,00
mwst_gesamt := 0,00
```

Öffne Tabelle

Lese 1. Zeile in Tabelle

Solange Ende der Tabelle nicht erreicht ist

endsumme := endsumme + gesamtpreis

wenn MwSt_ID = "B"

ja	nein
mwst_Betrag := gesamtpreis * mwst_A_Satz / (100 - mwst_A_Satz)	mwst_Betrag := gesamtpreis * mwst_B_Satz / (100 + mwst_B_Satz)
mwst_A_gesamt := mwst_A_gesamt + mwst_Betrag	mwst_B_gesamt := mwst_B_gesamt + mwst_Betrag

Lese nächste Zeile in Tabelle

Schliesse Tabelle

mwst_gesamt := mwst_gesamt + mwst_Betrag

Ausgabe: endsumme, mwst_A_gesamt, mwst_B_gesamt, mwst_gesamt

```
double : endsumme
integer:  mwst_A_Satz, mwst_B_Satz
integer : mwst_A_gesamt, mwst_B_gesamt, mwst_gesamt
double:  (1)

endsumme := 0,00

mwst_A_Satz := 19,0   19  (2)
mwst_B_Satz :=  7

mwst_Betrag := 0,00
mwst_A_gesamt := 0,00
mwst_B_gesamt := 0,00
mwst_gesamt := 0,00

Öffne Tabelle

Gehe in Tabelle zur 1. Zeile

Solange Ende der Tabelle nicht erreicht ist

    endsumme := endsumme + gesamtpreis

    wenn MwSt_ID = "B"   "A"  (3)

    ja                                                    nein

    mwst_Betrag := gesamtpreis * mwst_A_Satz / (100 - mwst_A_Satz)   (4)  +
    mwst_A_gesamt := mwst_A_gesamt + mwst_Betrag

    mwst_Betrag := gesamtpreis * mwst_B_Satz / (100 + mwst_B_Satz)
    mwst_B_gesamt := mwst_B_gesamt + mwst_Betrag

    Gehe in Tabelle zur nächsten Zeile

Schliesse Tabelle

mwst_gesamt := mwst_gesamt + mwst_Betrag
    (5)  mwst_A_gesamt + mwst_B_gesamt

Ausgabe: endsumme, mwst_A_gesamt, mwst_B_gesamt, mwst_gesamt
```

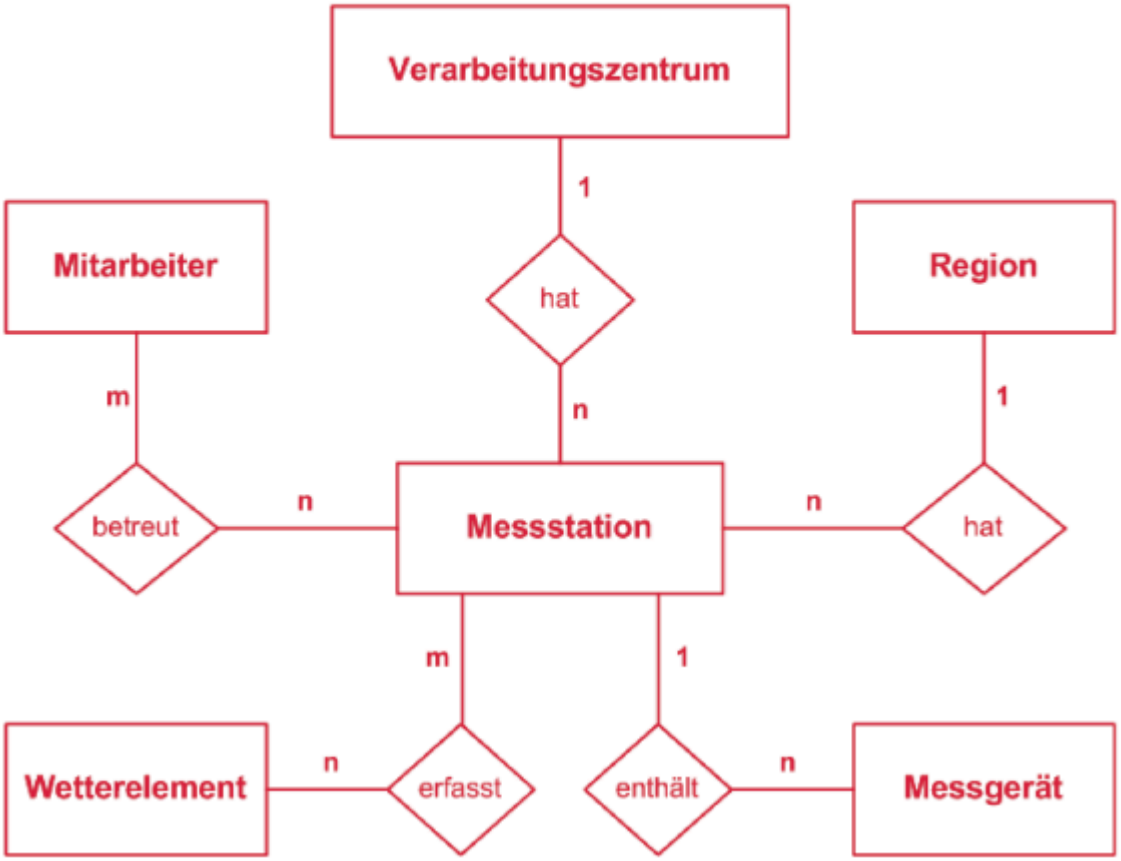
Hinweis für Prüfer:
1 falscher Variablentyp
2 falsche Variablendeklaration
3 falsche Verzweigung
4 Fehler in der Formel zur Berechnung der Variablen mwst_Betrag
5 falsche Berechnung der Variablen mwst_gesamt

2017S AP IT GA2 Lösungen 4. Handlungsschritt (25 Punkte)
Die GeoData AG möchte in einer Datenbank unter anderem ihre Messstationen, die Messgeräte und die verantwortlichen technischen Mitarbeiter erfassen.
Nach einem ersten Gespräch haben sich folgende Informationen ergeben:

- Jede Messstation erfasst die Werte mehrerer Wetterelemente (z. B. Temperatur, Luftdruck, Niederschlag).
- Jede Messstation liefert ihre Messdaten an jeweils ein Verarbeitungszentrum, welches die Daten von verschiedenen Messstationen auswertet.
- Jede Messstation wird einer bestimmten Region zugeordnet. In jeder Region sind mehrere Messstationen installiert.
- Mindestens zwei technische Mitarbeiter kümmern sich um alle Messstationen einer Region und sind diesen direkt zugeordnet.
- Eine Messstation enthält mehrere Messgeräte.

a) Erstellen Sie aus diesen Informationen ein ER-Modell, welches nur die Entitätstypen, die Beziehungen zwischen den Entitätstypen und die entsprechenden Kardinalitäten enthält.
16 Punkte
Hinweis: Es sollen keine Attribute in dieses Modell eingetragen werden.
ER-Modell

a) 16 Punkte
6 Punkte, 6 x 1 Punkt je Entitätstyp
5 Punkte, 5 x 1 Punkt je Verbindung
5 Punkte, 5 x 1 Punkt je Kardinalität



b) Aus dem ER-Modell soll später ein relationales Datenmodell entwickelt werden. In einem Gespräch mit einem Mitarbeiter der GeoData AG fallen die folgenden Begriffe.
Erläutern Sie diese Begriffe.

ba) Datenredundanz 3 Punkte

Mehrfaches Vorhandensein ein und derselben Information in einer Datenbank

bb) Normalisierung 3 Punkte

Ein Verfahren zur Verringerung von Datenredundanz in relationalen Datenmodellen mit dem Ziel, die Datenkonsistenz zu erhöhen

bc) Primärschlüssel 3 Punkte

Ein Attribut oder mehrere miteinander kombinierte Attribute, die jeden Datensatz einer Tabelle eindeutig kennzeichnen

2017W AP IT GA2 Lösungen 3. Handlungsschritt (25 Punkte)
Die HurryUp GmbH will die an die Mietwagen-Unternehmen vermittelten Fahrten in einer Datenbank verwalten. Die Klübero GmbH hat diese Datenbank bereits entwickelt und soll nun SQL-Anweisungen zur Auswertung und Veränderung der Daten erstellen.
a) Zunächst sollen Sie einem Mitarbeiter der HurryUp GmbH die SQL erläutern. Zur Vorbereitung des Informationgesprächs liegt folgender Text vor:
Structured Query Language is a standardizedlanguageusedformanaging relational databases. The language is divided into four types of primary language statements:
1. Data Manipulation Language
These statements are basic operations such as selecting records from a table, inserting new records, deleting records, and modifying existing records.
2. Data Definition Language
These statements are used to create a table or alter a table structure.
3. Data Control Language
These statements are used to provide or remove access rights to users in a database.
4. Transaction Control Language
These statements maintain the integrity of data.

Übersetzen Sie den Text sinngemäß ins Deutsche. 7 Punkte

Die strukturierte Abfragesprache ist eine standardisierte Sprache, die zum Verwalten relationaler Datenbanken verwendet wird. Die Sprache
ist in vier grundlegende Sprachbereiche gegliedert. (2 Punkte)
1. Datenbearbeitungs-Sprache
Diese Anweisungen sind grundlegende Operationen, wie das Auswählen von Datensätzen aus einer Tabelle, das Einfügen neuer Datensätze,
das Löschen von Datensätzen und die Veränderung von bestehenden Datensätzen. (2 Punkte)
2. Datendefinitions-Sprache
Diese Anweisungen werden benutzt, um eine Tabelle zu erzeugen oder eine Tabellenstruktur zu verändern. (1 Punkt)
3. Datenüberwachungs-Sprache
Diese Anweisungen werden benutzt, um Nutzern einer Datenbank Zugriffsrechte zu geben oder zu entziehen. (1 Punkt)
4. Transaktionsüberwachungs-Sprache
Diese Anweisungen erhalten die Datenintegrität. (1 Punkt)

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

SQL-Syntax (Auszug)

Syntax	Beschreibung
Tabelle	
CREATE TABLE Tabellenname(Feldname < DATENTYP >, Primärschlüssel, Fremdschlüssel)	Erzeugt eine neue leere Tabelle mit der beschriebenen Struktur
ALTER TABLE Tabellenname **ADD COLUMN** Spaltenname Datentyp **DROP COLUMN** Spaltenname	Änderungen in einer Tabelle: Hinzufügen einer Spalte Entfernen einer Spalte
CHARACTER	Textdatentyp
DECIMAL	Numerischer Datentyp (Festkommazahl)
DOUBLE	Numerischer Datentyp (Doppelte Präzision)
INTEGER	Numerischer Datentyp (Ganzzahl)
DATE	Datum (Format DD.MM.YYYY)
PRIMARY KEY (Feldname)	Erstellung eines Primärschlüssels
FOREIGN KEY (Feldname) **REFERENCES**	Erstellung von Fremdschlüssel-Beziehungen
DROP TABLE Tabellenname	Löscht eine Tabelle
Befehle, Klauseln, Attribute	
SELECT * | Feldname1 [, Feldname2, ...]	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) **WHERE NOT EXISTS** (subquery)	Die Bedingungen EXISTS prüft, ob die Suchbedingung einer Unterabfrage mindestens eine Zeile zurückliefert. NOT EXIST negiert die Bedingung.
GROUP BY Feldname1 [,Feldname2, ...]	Gruppierung (Aggregation) nach Inhalt des genannten Feldes
ORDER BY Feldname1 [,Feldname2, ...] **ASC | DESC**	Sortierung nach Inhalt des genannten Feldes oder der genannten Felder ASC: aufsteigend; DESC: absteigend
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

Fortsetzung ->

SQL-Syntax (Auszug) – Fortsetzung

Syntax	Beschreibung
Aggregatfunktionen	
AVG(Feldname)	Ermittelt das arithmetische Mittel aller Werte im angegebenen Feld
COUNT(Feldname | *)	Ermittelt die Anzahl der Datensätze mit Nicht-NULL-Werten im angegebenen Feld oder alle Datensätze der Tabelle (dann mit Operator *)
SUM(Feldname | Formel)	Ermittelt die Summe aller Werte im angegebenen Feld oder der Formelergebnisse
MIN(Feldname | Formel)	Ermittelt den kleinsten aller Werte im angegebenen Feld
MAX (Feldname | 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 2016-06-17

b) Eine Tabelle der Datenbank ist die folgende Tabelle Fahrt.

Fahrt
Fahrt_nr: Integer Datum: Date Fahrtstrecke_km : Double Ort: Varchar Anzahl_Fahrgaeste: Integer Preis_Fahrt: Double Preis_Zusatzleistung: Double

Sie sollen für die Tabelle Fahrt die SQL-Anweisungen für die folgenden vom Kunden gewünschten Aufgaben und Operationen erstellen.

ba) Ausgabe: Länge der längsten Fahrtstrecke in km, die bei einer Fahrt zurückgelegt wurde. Es soll der Alias km verwendet werden. 3 punkte

```
SELECT MAX(Fahrtstrecke_km) AS km
FROM Fahrt
```

bb) Ausgabe: Anzahl der Fahrgäste, die auf der Fahrt Nr. 2367 befördert wurden. 3 Punkte

```
SELECT Anzahl_Fahrgaeste
FROM Fahrt
WHERE Fahrt_Nr = 2367
```

bc) Ausgabe: Summe aller Preise pro Fahrt ohne Zusatzleistungen der am 10.11.2017 durchgeführten Fahrten. 4 Punkte

```
SELECT SUM(Preis_Fahrt)
FROM Fahrt
WHERE Datum = '10.11.2017‘
```

bd) Operation: Neuen Datensatz für die Fahrt Nr. 6789 einfugen, die am 10.11.2017 in Hamburg zum Fahrtpreis von 35,50 EUR durchgeführt wurde. Weitere Daten liegen noch nicht vor. 4 Punkte

```
INSERT INTO Fahrt(Fahrt_Nr, Datum, Ort, Preis_Fahrt)
VALUES (6789, ‘10.11.2017‘, ‘Hamburg‘, 35.50)
```

be) Operation: Für die Fahrt Nr. 3333 den Preis für Zusatzleistungen um 10,30 EUR erhöhen. 4 Punkte

```
UPDATE Fahrt SET
Preis_Zusatzleistung = Preis_Zusatzleistung + 10.30
WHERE Fahrt_nr = 3333
```

2018W AP IT GA2 Lösungen 4. Handlungsschritt (25 Punkte)
Die Klübero GmbH soll ein Kassensystem für das Restaurant in der Skihalle entwickeln.
a) Die Entwicklung des Programms soll objektorientiert erfolgen.
aa) In der objektorientierten Programmierung werden Klassen, Vererbung und Objekte verwendet. Erläutern Sie jeweils... 5 Punkte

Klasse:
Definiert eine Kategorie von Objekten mit gleichen Attributen und Methoden. Sie stellt einen Bauplan für Objekte dar.

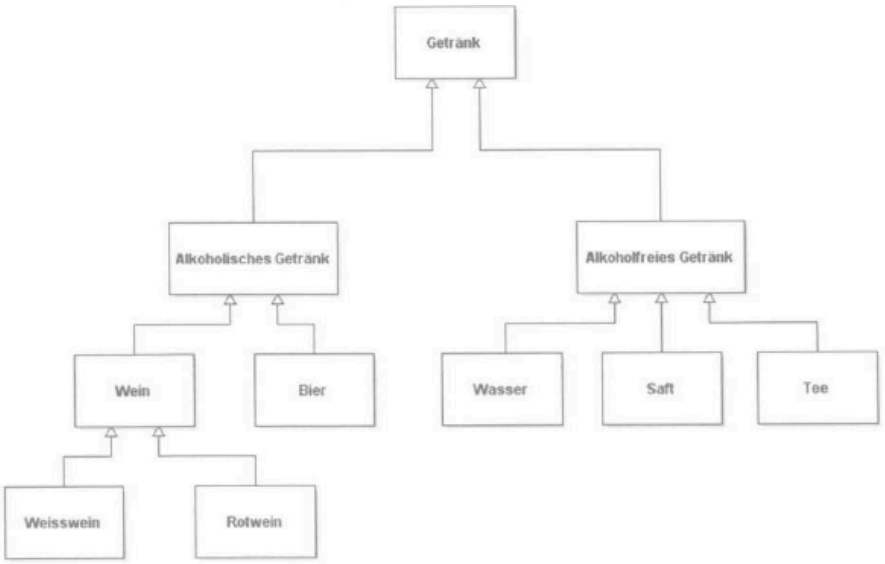
Vererbung:
Vorgang, durch den aus einer oder mehreren existierenden Klassen neue Klassen entwickelt werden können.

Objekt:
Ein Exemplar (Instanz) einer Klasse.

ab) Im Rahmen der Entwicklung des UML-Klassendiagramms für das Programm erhalten Sie die Aufgabe den Vererbungsbaum für die Klasse Getränk zu entwickeln. Ein erster Ansatz ist vorhanden.

Folgende Klassen sollen noch in den Vererbungsbaum integriert werden;
Wein, Wasser, Rotwein, Bier, Saft, Alkoholfreies Getränk, Tee und Weißwein.
Erweitern Sie den folgenden Vererbungsbaum um diese Klassen. 8 Punkte
Vererbungsbaum Klasse Getränk

Lösung



b) Die Klübero GmbH soll für das Kassensystem des Restaurants eine Datenbank entwickeln, in der die Bestellungen der Gäste erfasst werden.
Für eine Bestellung gelten folgende Sachverhalte:

- Eine Bestellung wird von genau einem Kellner bearbeitet.
- Pro Bestellung können mehrere Speisen bestellt werden.

Zunächst soll ein relationales Datenmodell in der dritten Normalform entwickelt werden.
Vervollständigen Sie dazu das folgende Datenmodell, indem Sie

- die entsprechenden Tabellen ergänzen.
- die Primärschlüssel- und Fremdschlüsselattribute eintragen (weitere Attribute nicht).
- Primärschlüssel mit PK und Fremdschlüssel mit FK kennzeichnen.
- die Beziehungen zwischen den Tabellen und deren Kardinalsten eintragen.

12 Punkte

Datenmodell für die Datenbank Bestellung

Lösung

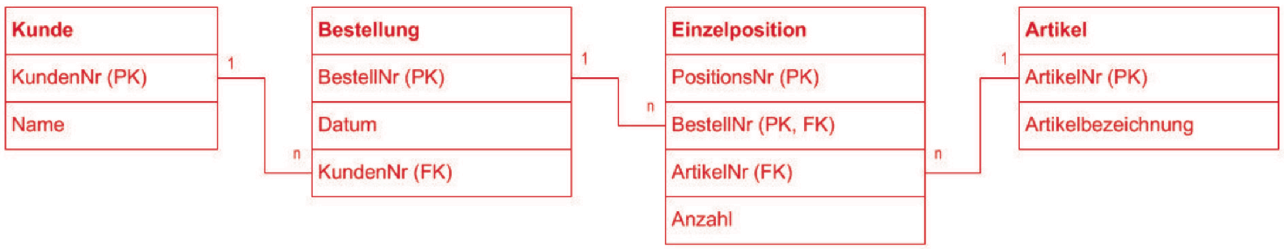
2019S AP IT GA2 Lösungen 4. Handlungsschritt (25 Punkte)
Die Fidule GmbH möchte eingehende Bestellungen zukünftig in einer Datenbank verwalten.
Dafür liegt folgende Tabelle „Bestellung" als erster Entwurf vor:

Best.-Nr.	Kunde	Datum	Einzelpositionen
1	Winter - (KundenNr. 23)	23.04.2019	1.20 Bälle (ArtikelNr. 134) 2.4 Schläger (ArtikelNr. 4) 3. 1 Netz (ArtikelNr. 77)
2	Sommer - (KundenNr. 71)	23.04.2019	1.10 Bälle (ArtikelNr. 134) 2. 30 Gymnastikbänder (ArtikelNr. 44)
3	Winter- (KundenNr. 23)	26.04.2019	1.30 Bälle (ArtikelNr. 134) 2. 6 Schläger (ArtikelNr. 4)

a) Die Tabelle ist noch nicht normalisiert.
Entwickeln Sie daraus ein relationales Datenmodell, welches der 3. Normalform genügt und ergänzen Sie das folgende Datenbankmodell. 11 Punkte

Bestellung
BestellNr (PK)
Datum
KundenNr (FK)

2 Punkte Tabelle „Kunde“
2 Punkte Tabelle „Artikel“
4 Punkte Tabelle „Einzelpositionen“
3 Punkte (3 x 1 Punkt je Beziehung)

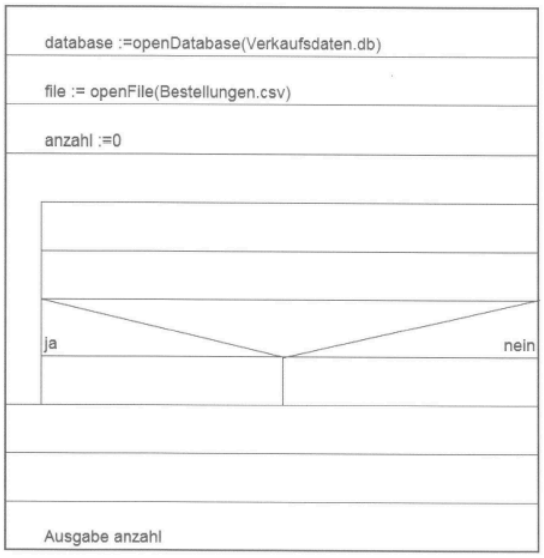


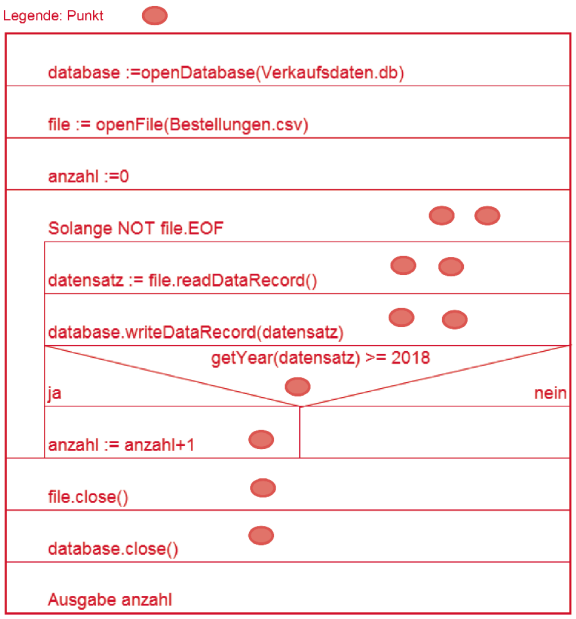
b) Die Daten wurden bisher in einer Datei „Bestellungen.csv" gespeichert. Diese sollen nun durch ein Programm ausgelesen und in die Datenbank „Verkaufsdaten.db" übertragen werden. Weiterhin sollen alle Bestellungen ab dem Jahr 2018 gezählt und die Anzahl am Ende ausgeben werden. Für diese Aufgabe stehen sieben Methoden zur Verfügung.

Methode	Beschreibung
openDatabase(stringdataBaseName)	Öffnet die entsprechende Datenbank und liefert eine Referenz auf die offene Datenbank zurück
openFile(stringfileName)	Öffnet die entsprechende Datei und liefert eine Referenz auf die offene Datei zurück
<<Referenz>>.close()	Schließt die entsprechende Referenz (Datenbank oder Datei)
<<Referenz>>.read Data Record()	Liest aus einer geöffneten CSV-Datei (Referenz) jeweils einen kompletten Datensatz einer Bestellung und gibt diesen als String zurück
<<Referenz>>.writeDataRecord(stringdataRecord)	Schreibt in eine geöffnete Datenbank (Referenz) die Daten einer Bestellung, welche der Funktion als Datensatz übergeben werden
getYear(stringdataRecord)	Ermittelt aus dem übergebenen Datensatz das Bestelljahr und gibt dieses als Integer zurück

Vervollständigen Sie folgendes Struktogramm: 10 Punkte

6 x 1 Punkt pro Anweisung
2 Punkte für die Verzweigung (Bedingung + zählen der Anzahl)
2 Punkte für die Schleife





EOF=> End of file

CSV=> Comma-separated values

2019W AP IT GA2 Lösungen 4. Handlungsschritt (25 Punkte) 15 Punkte

c) Für die Verwaltung der Mitarbeiter der Fidule GmbH liegt schon eine Datenbank vor. Diese enthält u. a. folgende Tabelle:

Mitarbeiter	
MitarbeiterNr	PK
Name	
Vorname	
Geburtsdatum	
TelefonPrivat	

Es soll nun mithilfe einer SQL-Anweisung die private Telefonnummer von Frank Müller mit der Mitarbeiter-Nr: 123 gelöscht werden. Dazu liegt folgende SQL-Anweisung vor:

DELETE FROM Mitarbeiter WHERE Name = "Müller" AND Vorname = 'Frank"
Diese liefert aber nicht das gewünschte Ergebnis.
ca) Beschreiben Sie das Ergebnis der SQL-Anweisung, wenn diese ausgeführt wird.
2 Punkte

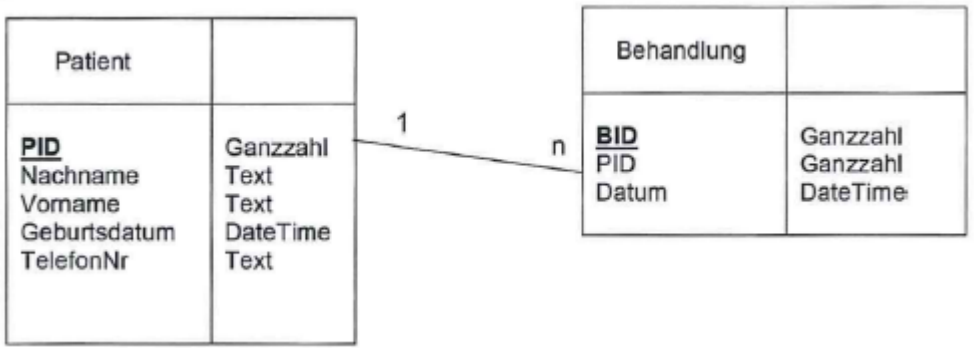
Löscht komplett alle Datensätze, wo Frank Müller als Name vorkommt.

cb) Formulieren Sie eine SQL-Anweisung, mit der das gewünschte Ergebnis erreicht wird. 2 Punkte

UPDATE Mitarbeiter SET TelefonPrivat = NULL WHERE MitarbeiterNr = 123

Hinweis für die Prüfer: namensgleiche Mitarbeiter, deswegen unbedingt die MitarbeiterNr verwenden

c) Das medizinische Versorgungszentrum verfügt über eine Datenbank. Ein Ausschnitt aus dieser Datenbank ist in der nachfolgenden Abbildung dargestellt.



Entwickeln Sie für die folgenden Aufgabenstellungen die notwendigen SQL-Anweisungen (siehe perforierte Anlage SQL-Anweisungen).

ca) Es sollen alle Patienten (Nachname, Vorname) abgefragt werden, deren Nachname mit einem, „M“ beginnt. Die Ergebnismenge soll nach Nachname aufsteigend sortiert sein. 4 Punkte

SELECT Patient.Nachname, Patient.Vorname
FROM Patient
WHERE Patient.Nachname LIKE ‘M%’
ORDER BY Patient.Nachname ASC
(Andere Lösungen sind möglich.)

cb) Es soll die Telefonnummer des Patienten mit der PID „734" auf „0162 -1234567“ geändert werden. 3 Punkte

UPDATE Patient SET Patient.TelefonNr = ‘0162 – 1234567’
WHERE PID = 734
(Andere Lösungen sind möglich.)

cc) Es soll die Anzahl der Behandlungen ermittelt werden, die im Januar 2019 durchgeführt wurden. 4 Punkte

SELECT COUNT(Behandlung.BID)
FROM Behandlung
WHERE Behandlung.Datum>= ‘01.01.2019’ AND
Behandlung.Datum<= ’31.01.2019‘
(Andere Lösungen sind möglich.)

2020S AP IT GA1 Lösungen 4. Handlungsschritt (25 Punkte)

a) Als IT-Administrator/-in der BauMit AG wollen Sie die Benutzer einer Datenbank automatisiert per E-Mail informieren, falls der Gültigkeitszeitraum ihres Kennwortes in den nächsten Tagen abläuft. Das Skript zum Prüfen und Benachrichtigen wird täglich ausgeführt. Ein entsprechendes Skript-Template liegt nebenstehend als Anlage bei und soll angepasst werden.

aa) Weisen Sie den folgenden Variablen die geeigneten Werte/Funktionen zu. 4 Punkte

$AktuellesDatum	Get-Date oder andere sinnvolle Schreibweisen
$smtpServer	192.168.99.1
$Firma	Baumit AG

ab) Geben Sie an, wie viele Tage vor Ablauf des Kennwortes die erste Benachrichtigung erfolgt und wie viele Benachrichtigungen ein Benutzer maximal erhält. 6 Punkte

Die erste E-Mail erfolgt sechs Tage vor Ablauf des Kennwortes.
Es erfolgen bis zu zwei Benachrichtigungen per E-Mail.

Anlage zu Handlungsschritt 4: Skript-Template zum automatisierten Versenden von E-Mails

1	// Variablen deklarieren, diese sollen Sie im Handlungsschritt 4 unter aa)initialisieren!
2	$AktuellesDatum
3	$smtpServer
4	$Firma
5	// Objekte erzeugen
6	$smtp = new-object Net.Mail.SmtpClient($smtpServer)
7	$Email = new-object Net.Mail.MailMessage
8	// E-Mail Benachrichtigungstext
9	$body = „Sehr geehrte/r Frau/Herr „ + $NachName
10	$body = $body + „Ihr Kennwort laeuft am + „$Datum + „ab. Bitte aendern Sie Ihr Kennwort.“
11	// Datum pruefen und Benutzer benachrichtigen
12	ForEach($Benutzer in $Benutzern) {
13	$KennwortAblaufdatum = $Benutzer.AblaufDatum
14	$VerbleibendeZeitAblauf = New-TimeSpan -Start $AktuellesDatum -End $KennwortAblaufdatum
15	$EmailSenden = $false
16	switch ($VerbleibendeZeitAblauf.Days) {
17	3{ $EmailSenden = $true }
18	6{ $EmailSenden = $true }
19	}
20	if ($EmailSenden -eq $true) {
21	$Email.To.Add($Benutzer.EmailAddresse)
22	$Email.Subject = „Wichtiger Hinweis zu Ihrem Datenbankzugang"
23	$Email.Body = $body -replace $NachName, $Benutzer.NachName
24	Den fehlenden Code dieser Zeile sollen Sie im Handlungsschritt 4 unter ac) erstellen!
25	$smtp.Send($Email)
26	}
27	}

Fortsetzung 4. Handlungsschritt
ac) Im E-Mail-Textfeld soll das jeweilige Ablaufdatum des Kennwortes enthalten sein.
Entwickeln Sie dazu den entsprechenden Code, der in Zeile 24 des Skripts einzufügen ist.
6 Punkte

$Email.Body = $Email.Body -replace $Datum, $Benutzer.AblaufDatum.

b) Nach der vollständigen Erstellung des Skripts wird von diesem ein Hashwert gebildet und hinterlegt.
Erläutern Sie, warum diese Maßnahme sinnvoll ist. 3 Punkte

Wird an dem Skript manipuliert, kann anhand des Hashwertes diese Manipulation erkannt werden.

c) E-Mail-Programme verwenden die Protokolle SMTP, POP3 und IMAP4.
Erläutern Sie jeweils die Aufgabe der Protokolle: 6 Punkte

SMTP: Versand von E-Mails (Client – Server, Server – Server)
POP3: Abholen von E-Mails (Client vom Server)
IMAP4: Lesen und Verwalten von E-Mails auf dem Server durch den Client

2020W AP IT GA2 Lösungen 4. Handlungsschritt (25 Punkte)

a) Der Verlag möchte ein Content-Management-System (CMS) einsetzen. Dazu sollen Sie sich im Vorfeld über Content-Management-Systeme informieren. Es liegtfolgender Text in EnglischerSprachevor:

A content management System (CMS) is a Software application or set of related programs that is used to create and manage digital content. Web content may include text and embedded graphics, photos, video, audio, maps, and program code that displays content or interacts with the user. Content management System (CMS) typically has two major components: a content management application (CMA), as the front-end user interface that allows a user, even with limited expertise, to add, modify, and remove content from a website without the Intervention of a webmaster and a content delivery application (CDA), that organizes and saves the content, and Updates the website. Based on market share statistics, the most popular content management System is WordPress, followed by Joomla and Drupal.

Beantworten Sie folgende Fragen.

aa) Beschreiben Sie die Aufgabe eines CMS. 3 Punkte

Ein CMS ist eine Softwareanwendung, mit der man digitale Inhalte erstellen und verwalten kann. Inhalte können Text, Grafiken, Fotos, Videos, Audios und Programmcode sein.

ab) Beschreiben Sie die Aufgabe der CMA. 3 Punkte

Mithilfe des CMA kann ein normaler Nutzer Inhalte hinzufügen, ändern und löschen, ohne dass die Hilfe eines Webmasters erforderlich ist.

ac) Nennen Sie zwei populäre CMS. 2 Punkte

WordPress, Joomla, Drupal

b) Die Inhalte für die Online-Lernprogramme sollen in einzelnen Projekten entwickelt werden. Für die Verwaltung der Projekte ist eine kleine Datenbank geplant. Folgende Informationen sind bekannt:

An einem Projekt arbeiten mehrere Mitarbeiter aus verschiedenen Fachbereichen und ein Mitarbeiter kann auch gleichzeitig in mehreren Projekten eingesetzt sein.

Jeder Mitarbeiter kann genau einem Fachbereich zugeordnet werden.

Jedes Projekt beinhaltet ein oder mehrere Lernfelder.

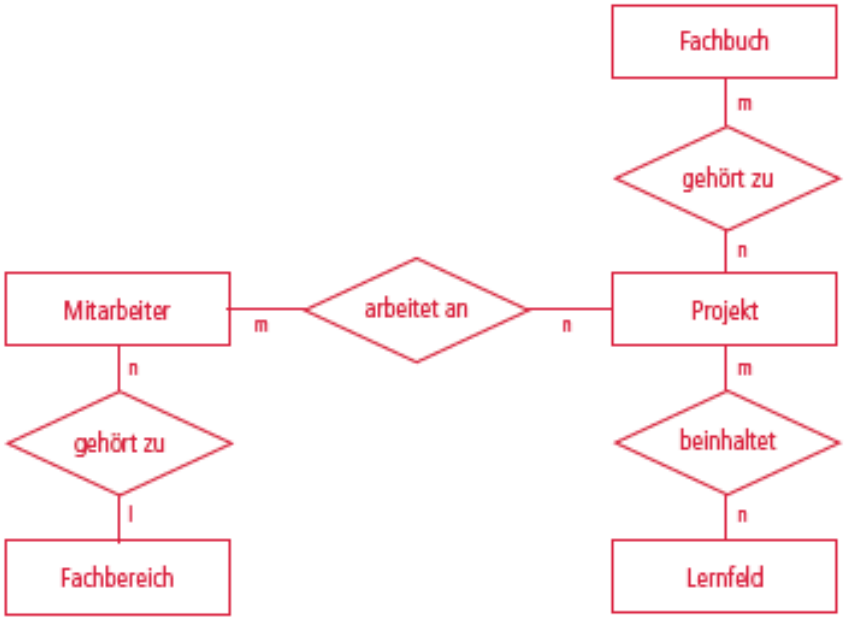
Die Inhalte der Lernfelder können auch durch mehrere Projekte erstellt werden.

Zu jedem Projekt werden mehrere Fachbücher zugeordnet. Dabei kann ein Fachbuch auch in verschiedenen Projekten Verwendung finden.

Erstellen Sie aus diesen Informationen ein ER-Modell, welches nur die Entitätstypen, die Beziehungen zwischen den Entitätstypen und die entsprechenden Kardinalitäten enthält. 13 Punkte

Hinweis: Es sollen keine Attribute in dieses Modell eingetragen werden. Eine Normalisierung wird nicht verlangt.

b) 13 Punkte
Je Entitätstyp 1 Punkt (5 x 1 Punkt)
Je Beziehung 1 Punkt (4 x 1 Punkt)
Je richtige Kardinalität 1 Punkt (4 x 1 Punkt)
Hinweis: Auch andere Lösungen sind möglich



c) Später soll ein relationales Datenbanksystem zur Verwaltung der Daten benutzt werden. Ein Datenbanksystem (DBS) setzt sich aus einem Datenbankmanagementsystem (DBMS) und der eigentlichen Datenbank zusammen.

Nennen Sie vier Funktionen eines DBMS. 4 Punkte

- Speichern, Überschreiben und Löschen von Daten
- Rollen-, Rechte- und Benutzerverwaltung
- Vorkehrungen zur Gewährleistung der Datenintegrität
- Umsetzung des Mehrbenutzerbetriebes
- Optimierung der Abfragen
- Realisierung von Triggern und StoredProcedures
- u. a.

2021W AP IT GA1 Lösungen 5. Handlungsschritt (25 Punkte)
Die Geschäftsführung der carbonom GmbH möchte zur besseren Auftragsabwicklung ihre Aufträge der Cloud-Produkte in einer separaten Datenbank verwalten.
a) Beim Entwurf einer relationalen Datenbank wird eine Normalisierung durchgeführt. Erläutern Sie, welcher Zweck mit der Normalisierung erreicht werden soll. 3 Punkte

Durch die Normalisierung soll eine korrekte, relationale Datenbank erstellt werden. Dabei sollen Redundanzen vermieden werden, da diese sonst schnell bei Änderungen von Inhalten zu Inkonsistenzen führen.

b) Die Tabelle Mitarbeiter soll in die 3. Normalform überführt werden.
Tabelle Mitarbeiter

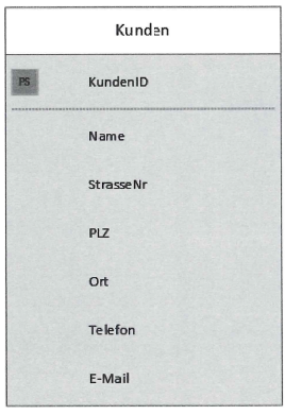
ID	Name	Vorname	Abteilung	Tel.	Kunden-Nr.
365	Müller	Tanja	A01	458	4582,4595.1258
366	Hansen	Hugo	A03	568	6845,5890
367	Lausen	Anette	A01	459	7358
368	Müller	Tanja	A01	458	5768

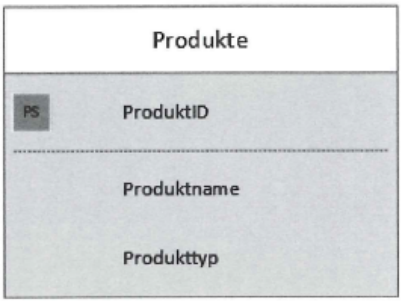
Erläutern Sie, an welchen Spalten eine Normalform verletzt wird. 5 Punkte

In der Spalte Name, Vorname, Abteilung und Telefonnummer befinden sich doppelte Einträge.
In der Spalte Kunden-Nr. sind Mehrfach-Einträge.

c) Die Tabellen für die Kunden und die Produkte liegen bereits vor. Es soll eine weitere Tabelle mit den Aufträgen angelegt werden.
Alle Aufträge erhalten eine „AuftragID" und ein Datum.
Verschiedene Kunden können die gleichen Produkte bestellen.

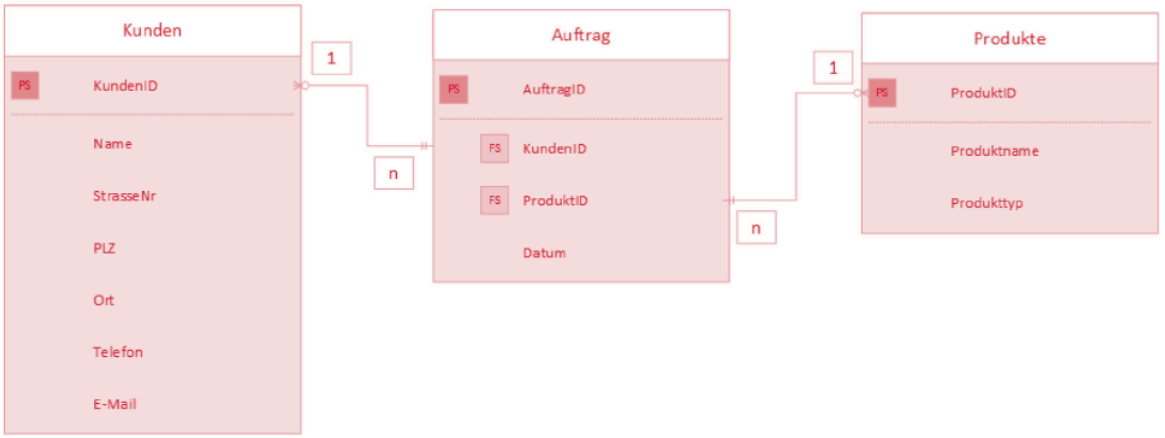
- Ein Kunde kann mehrere Aufträge erteilen.
- Ergänzen Sie den nebenstehenden Datenbankentwurf um die Tabelle Auftrag.
- Geben Sie die Primärschlüssel (PS) und Fremdschlüssel (FS) an.
- Zeichnen Sie die Beziehungen der Tabellen zueinander ein und geben Sie die Kardinalitäten an. 10 Punkte





c) 10 Punkte
Entität Auftrag 1 x 1 Punkt
Attribute 4 x 1 Punkt
Primärschlüssel Kennzeichnung 1 x 1 Punkt
Fremdschlüssel Kennzeichnung 2 x 1 Punkt
Beziehung mit Kardinalitäten 2 x 1 Punkt

Datenbank in der 3. Normalform



d) Die Datenbank soll abgesichert werden.
da) Dazu soll eine Zwei-Faktor-Authentifizierung eingesetzt werden.
Nennen Sie zwei Beispiele für eine Zwei-Faktor-Authentifizierung. 3 Punkte

PIN bzw. Passwort und App
PIN bzw. Passwort und Token/TAN
PIN bzw. Passwort und biometrisches Merkmal
u. a.

db) Da an der Datenbank täglich Veränderungen durchgeführt werden, soll diese gesichert werden. Geplant ist eine Vollsicherung am Sonntag. Für die Sicherung an den Werktagen (Montag - Samstag) stehen eine differentielle oder eine inkrementelle Sicherung zur Auswahl.
Erläutern Sie den Unterschied zwischen differentieller und inkrementeller Datensicherung.
4 Punkte
Die differentielle Sicherung speichert die Änderungen zu einem Basistag.
Die inkrementelle Sicherung speichert die Änderungen zum Vortag bzw. zur letzten Sicherung.

2021W AP IT GA2 Lösungen 5. Handlungsschritt (25 Punkte)
Im Zuge der Umrüstung des mittelständischen Industriebetriebes soll auch die bisherige Datenbankanwendung angepasst und erweitert werden.

a) Das Unternehmen besitzt eine Reihe von Maschinen, welche schon in der Datenbank erfasst wurden. Nun sollen auch die Wartungsvorgänge an diesen und die Techniker, welche die Wartung durchführen, in der Datenbank gespeichert werden. Bisher wurden die Wartungsdaten wie folgt in Papierform protokoliert:

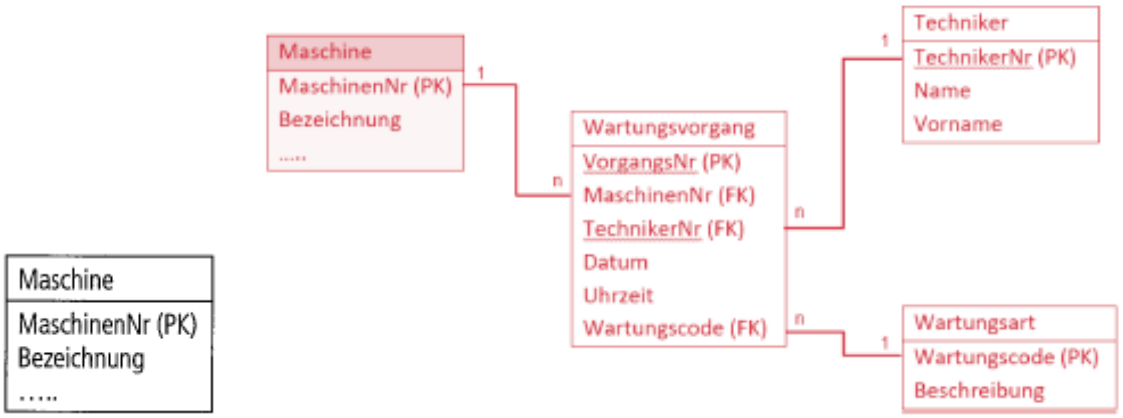
Wartungsvorgänge				
Vorgangs-Nr	Techniker	Datum/Uhrzeit	Maschinen-Nr	Wartungscode
1	Baum, Frank	20.03.2021 -08.00 Uhr	12	45
2	Baum, Rank	20.03.2021 -10.00 Uhr	14	45
3	Holz, Maria	21.03.2021 -08.00 Uhr	15	31
4	Holz, Maria	22.03.2021 -08.00 Uhr	21	22
5	Baum, Frank	22.03.2021 -13.30 Uhr	3	22

Hinweis: Eine Wartung wird immer von genau einem Techniker durchgeführt.

Alle Wartungscodes sind mit einer entsprechenden Beschreibung bisher in einem Ordner „Wartungsarten“ abgelegt und können dort eingesehen werden.

Erweitern Sie das bestehende relationale Datenmodell so, dass alle Daten der Wartungsvorgänge inklusive der Wartungscodes mit ihren Beschreibungen auch in der Datenbank erfasst werden können.

Erstellen Sie dazu die notwendigen Tabellen in der 3. Normalform, zeichnen Sie die Beziehungen zwischen den Tabellen mit den entsprechenden Kardinalitäten ein und kennzeichnen Sie alle Primär- und Fremdschlüssel mit PK bzw. FK. 12 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 Datentyp **ADD FOREIGN KEY**(Spaltenname) **REFERENCES** Tabellenname(Primärschlüsselspaltenname)	Änderungen an 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(Primärschlüsselspaltenname)	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.
SELECT **DISTINCT**	Eliminiert Redundanzen, die in einer Tabellen auftreten können, Werte werden jeweils nur einmal angezeigt.
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) **WHERE NOT EXISTS** (subquery)	Die Bedingungen EXISTS prüft, ob die Suchbedingung einer Unterabfrage mindestens eine Zeile zurückliefert. NOT EXIST negiert die Bedingung.
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
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, ...])	Fügt Datensätze in die genannte Tabelle, die entweder mit festen Werten belegt oder Ergebnis eines SELECT-Befehls sind

oder **SELECT ... FROM ... WHERE**	
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
IS	Überprüfung auf NULL
=	Test auf Gleichheit
>, >=, <, <=, < >	Test auf Ungleichheit
*****	Multiplikation
/	Division
+	Addition, positives Vorzeichen
-	Subtraktion, negatives Vorzeichen

b) Die bestehende Datenbankanwendung soll des Weiteren um einige SQL-Anweisungen erweitert werden. Diese beziehen sich auf die folgenden Tabellen.

Erstellen Sie für folgende Aufgabenstellungen die entsprechenden SQL-Anweisungen.
ba) Einfügen einer neuen Maschine (Bezeichnung: Schraubroboter 071X, HerstellerNr: 4) in die Datenbank, welche am 23.07.2021 erworben wurde. Die Maschine soll die Maschinen-Nr 555 erhalten. Der Standort steht noch nicht fest. 3 Punkte

INSERT INTO Maschine (MaschinenNr, Bezeichnung, Anschaffungsdatum, HerstellerNr)
VALUES (555, ‘Schraubroboter 071X‘ ‘23.07.2021‘, 4)

bb) Abfrage aller Maschinen (MaschinenNr, Bezeichnung und Datum), welche im Jahr 2020 angeschafft wurden. Die Maschinen sollen nach dem Datum aufsteigend sortiert werden.
4 Punkte

SELECT MaschinenNr, Bezeichnung, Anschaffungsdatum
FROM Maschine WHERE Anschaffungsdatum >= '01.01.2020‘ AND
Anschaffungsdatum <= ‘31.12.2020‘ ORDER BY Anschaffungsdatum;

bc) Abfrage der Anzahl der Maschinen vom Hersteller „XXXX", welche im Standort „Zuschnitt“ stehen. 6 Punkte

SELECT COUNT (M.MaschinenNr) FROM Maschine M, Standort S, Hersteller H
WHERE M.StandortNr = S.StandortNr AND M.HerstellerNr = H.HerstellerNr AND
S.Bezeichnung = ‘Zuschnitt‘ AND H.Name = ‘XXXX‘;

oder
SELECT COUNT (M.MaschinenNr) FROM Hersteller H
INNER JOIN Maschine M ON H.HerstellerNr = M.HerstellerNr
INNER JOIN Standort S ON S.StandortNr = M.StandortNr
WHERE S.Bezeichnung = ‘Zuschnitt‘ AND H.Name = ‘XXXX‘;
Andere Lösungen sind möglich.

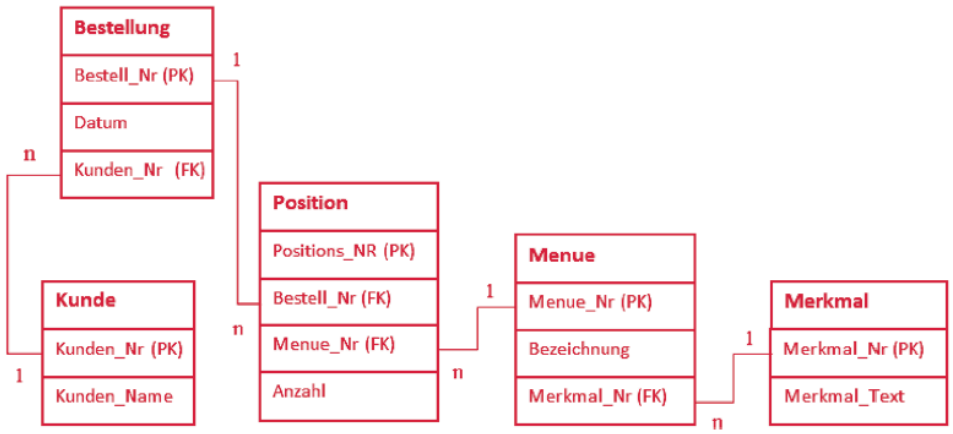
2022S AP IT GA2 Lösungen 4. Handlungsschritt (25 Punkte)

Die Bestellungen von Easy-Cooking-Boxen, welche die QuickMeal GmbH bekommt, sollen in einer Datenbank gespeichert und verwaltet werden.
Dafür liegt folgende Tabelle als erster Entwurf vor:

Bestellung	Kunde	Datum	Bestellte Menüs «Anzahl ist neu>>	Merkmal des Menüs
1	Müller - (KundenNr. 23)	23.04.2022	3 x Penne mit Ofengemüse	Vegetarisch
2	Mayer - (KundenNr. 71)	23.04.2022	1 x Tortellini mit Fisch	Glutenfrei
			1 x Steak mit Pfefferrahm auf Kartoffeln	Frei von Weizen
3	Müller - (KundenNr. 23)	26.04.2022	3 x Schweinefilet mit Gemüse	Glutenfrei
4	Krause - (KundenNr. 44)	30.04.2022	1 x Ziegenkäsetaler auf Salat	Vegetarisch
			1 x Kräuterschnitzel mit Kartoffeln	Vegetarisch
			1 x Tortellini mit Fisch	Glutenfrei
5	Mayer - (KundenNr. 71)	30.04.2022	2 x Steak mit Pfefferrahm auf Kartoffeln	Frei von Weizen

a) Bringen Sie die obige Tabelle in die 3. Normalform und ergänzen Sie das folgende Datenbankmodell. 11 Punkte

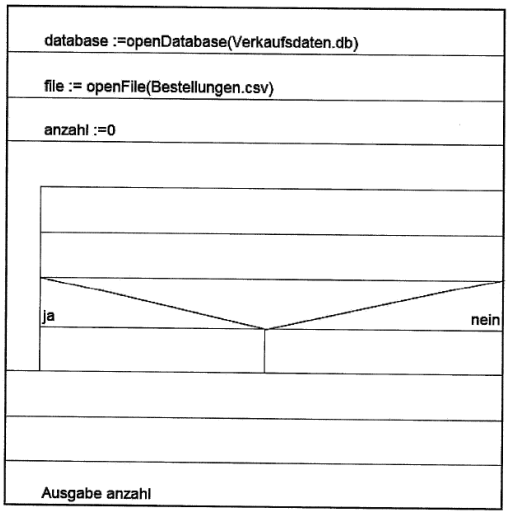
a) 11 Punkte
Tabellen Kunde, Merkmal je 1 Punkte (ein PK pro Tabelle wird vorausgesetzt),
Tabelle Position 3 Punkte,
Tabelle Menue 2 Punkte,
je Beziehung 1 Punkt



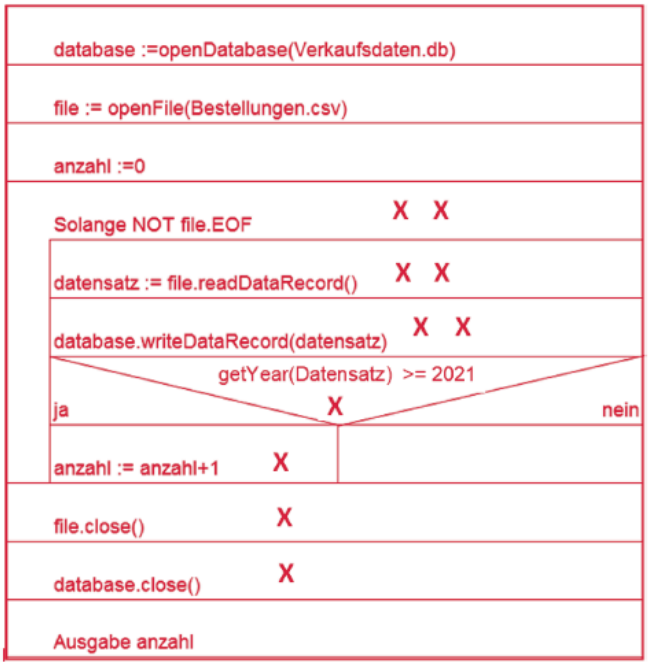
b) Die Daten wurden bisher in einer Datei „Bestellungen.csv" gespeichert. Diese sollen nun durch ein Programm ausgelesen und in die Datenbank „Verkaufsdaten.db" übertragen werden. Weiterhin sollen alle Bestellungen ab dem Jahr 2021 gezählt und die Anzahl am Ende ausgeben werden. Für diese Aufgabe stehen folgende Methoden zur Verfügung. (Hinweis: nicht jede der aufgelisteten Methoden wird zur Lösung benötigt.)

Methode	Beschreibung
openDatabase(stringdataBaseName)	Öffnet die entsprechende Datenbank und liefert eine Referenz auf die offene Datenbank zurück
openFile(stringfileName)	Öffnet die entsprechende Datei und liefert eine Referenz auf die offene Datei zurück
<<Referenz>>.close()	Schließt die entsprechende Referenz (Datenbank oder Datei)
<<Referenz>>.readDataRecord()	Liest aus einer geöffneten CSV-Datei (Referenz) jeweils einen kompletten Datensatz einer Bestellung und gibt diesen als String zurück.
<<Referenz>>.writeDataRecord(stringdataRecord)	Schreibt in eine geöffnete Datenbank (Referenz) die Daten einerBestellung, welche der Funktion als Datensatz übergeben werden
getyear(stringdataRecord)	Ermittelt aus dem übergebenen Datensatz das Bestelljahr und gibt dieses zurück

Vervollständigen Sie folgendes Struktogramm: 10 Punkte



b) 10 Punkte
6 x 1 Punkt pro Anweisung
2 Punkte für die Verzweigung (Bedingung + zählen der Anzahl)
2 Punkte für die Schleife
Legende: zu vergebender Punkt X



c) Für die Verwaltung der Mitarbeiter der QuickMeal GmbH liegt schon eine Datenbank vor. Diese enthält u. a. folgende Tabelle:

Mitarbeiter	
MitarbeiterNr	PK
Name	
Vorname	
Geburtsdatum	
TelefonPrivat	

Es soll nun mit Hilfe einer SQL-Anweisung die private Telefonnummer von Frank Müller mit der Mitarbeiter-Nr: 123 gelöscht werden. Dazu liegt folgende SQL-Anweisung vor:

DELETE FROM Mitarbeiter WHERE Name = "Müller" AND Vorname = "Frank"
Diese liefert aber nicht das gewünschte Ergebnis.
ca) Beschreiben Sie das Ergebnis der SQL-Anweisung, wenn diese ausgeführt wird.
2 Punkte

Löscht komplett alle Datensätze, in denen Frank Müller als Name vorkommt.

cb) Formulieren Sie eine SQL-Anweisung, mit der das gewünschte Ergebnis erreicht wird.
2 Punkte

UPDATE Mitarbeiter SET TelefonPrivat = NULL WHERE MitarbeiterNr = 123

Hinweis für die Prüfer: namensgleiche Mitarbeiter sind möglich, deswegen unbedingt die MitarbeiterNr verwenden

Abschlussprüfung Teil 1 Herbst 2022 4. Aufgabe (24 Punkte)
a) Sie erhalten den Auftrag, Produktionsdaten an die Steuerung der Walzanlage zu übergeben. Die Produktionsdaten werden in einer SQL-Datenbank gespeichert. Alle Datentypen sind Ganzzahlen. Die Breite, Länge und Dicke der Wellpappe wird in der Datenbank in Millimeter gespeichert.
Die Tabelle ProductionData hat den folgenden Aufbau:

OrderID (PK)
Width
Length
Thickness
Quantity

aa) Geben Sie den SQL-Befehl an, der die Breite, die Länge, die Dicke und die Anzahl der OrderID 736298 ausgibt. Die OrderID soll nicht in der Ergebnismenge enthalten sein.
3 Punkte

```
SELECT Width, Length, Thickness, Quantity FROM ProductionData
WHERE OrderID=736298;
```

ab) Wie viele Produktionsaufträge für Wellpappen mit einer Dicke von 2 mm wurden bisher in der Datenbank gespeichert.
Geben Sie dazu den entsprechenden SQL-Befehl an. 4 Punkte

```
SELECT Thickness, COUNT(*) As „Anzahl Wellpappen“
FROM ProductionData
GROUP BY Thickness
Having Thickness=2;
```

ac) Geben Sie die Gesamtanzahl gefertigter Wellpappen aus der Datenbank an, die mit einer Dicke von 2 mm, einer Breite von 200 mm und einer Länge von 300 mm gefertigt worden sind.
Geben Sie dazu den entsprechenden SQL-Befehl an. 4 Punkte

```
SELECT SUM (Quantity) As „Gesamtanzahl“
FROM ProductionData
WHERE width =200 AND length =300 AND Thickness=2
GROUP BY width;
```

Hinweis für den Prüfer:
GROUP BY width geht auch
GROUP BY length, oder
GROUP BY Thickness;
;

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

SQL-Syntax

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 Datentyp **ADD FOREIGN KEY**(Spaltenname) **REFERENCES** Tabellenname(Primärschlüsselspaltenname)	Änderungen an 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(Primärschlüsselspaltenname)	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 ... **FROM** ... **(SELECT** ... **FROM** ... **WHERE** ...) **AS** tbl **WHERE** ...	Unterabfrage (subquery), die in eine äußere Abfrage eingebettet ist. Das Ergebnis der Unterabfrage wird wie eine Tabelle – hier mit Namen "tbl" – behandelt.
SELECT **DISTINCT**	Eliminiert Redundanzen, die in einer Tabellen auftreten können, Werte werden jeweils nur einmal angezeigt.
JOIN / 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
WHERE	Bedingung, nach der Datensätze ausgewählt werden sollen
WHERE EXISTS (subquery) **WHERE NOT EXISTS** (subquery)	Die Bedingungen EXISTS prüft, ob die Suchbedingung einer Unterabfrage mindestens eine Zeile zurückliefert. NOT EXIST negiert die Bedingung.
WHERE ... IN (subquery) **WHERE NOT... IN** (subquery)	Der Wert des Datenfelds ist in der auswählten Menge vorhanden. Der Wert des Datenfelds ist in der auswählten Menge nicht vorhanden.
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

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[(spalte1, spalte2, ...)] **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
Berechtigungen kontrollieren	
CREATE Benutzer | Rolle **IDENTIFIED BY** 'Passwort'	Erzeugt einen neuen Benutzer oder eine neue Rolle mit einem Passwort
GRANT Recht | Rolle **ON** * * | Datenbank * | Datenbank.Objekt **TO** Benutzer | Rolle [WITH GRANT OPTION]	Weist einem Benutzer oder einer Rolle ein Recht auf ein bestimmtes Datenbank-Objekt zu Weist einem Benutzer eine Rolle zu
REVOKE Rechte | Rollen **ON** *.* | Datenbank.* | Datenbank.Objekt **FROM** Benutzer | Rolle	Entzieht einem Benutzer oder einer Rolle ein Recht auf ein bestimmtes Datenbank-Objekt Entzieht einem Benutzer eine Rolle
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 Text auf Gleichheit wenn Platzhalter ("regular expressions") eingesetzt werden.
NOT	Logische Negation
OR	Logisches ODER
IS NULL	Überprüfung auf NULL
=	Test auf Gleichheit
>, >=, <, <=, < >	Test auf Ungleichheit
*****	Multiplikation
/	Division
+	Addition, positives Vorzeichen
-	Subtraktion, negatives Vorzeichen

Stand 2021-09-30

b) Die abgefragten Produktionsdaten werden über eine entsprechende API an die Steuerung der Walzanlage übergeben. Die Auftragsdaten werden im Array result[] mit dem Index 0 bis 3 gespeichert. Sie sollen jetzt an die Steuerung der Walzanlage durch eine von Ihnen zu erstellende Funktion übergeben werden. Gehen Sie von einem Array result[] aus, bei dem im Index 0 die Breite, im Index 1 die Länge, im Index 2 die Dicke und im Index 3 die Anzahl der zu produzierenden Wellpappen stehen.
Erstellen Sie die Funktion **„launchTask(result[])".**
Zur Kommunikation mit der Steuerung der Walzanlage stehen Ihnen die folgenden API-Funktionen zur Verfügung:
setRollerDim(int,int,int) - Übergeben wird Breite, Länge und Dicke der Wellpappe.
rollerStart() - Startet einen Auftrag von einem Stück. Es wird eine Wellpappe mit den gesetzten Parametern erzeugt.

Die Walzanlage verfügt über einen Notausschalter. Sie darf nur laufen, wenn der Notaus nicht ausgelöst ist.

Der Status des Notausschalters kann mit der Funktion **bool getEmergencyStop()** abgefragt werden, der „true" liefert wenn der Notaus ausgelöst ist und „false" wenn der Notaus nicht ausgelöst ist.

Ergänzen Sie das gegebene Struktogramm durch die entsprechenden Befehle zur Produktion der geforderten Anzahl von Wellpappen (siehe Index 3) in den angegebenen Maßen (siehe Index 0,1 und 2). 7 Punkte

```
launchTask(result[])
int i = 0
bool emergencyStop = getEmergencyStop()
```

```
1  launchTask(result[])
2  int i = 0
3  bool emergencyStop = getEmergencyStop()
4  setRollerDim(result[0],result[1],result[2])
5  wiederhole solange i < result[3] und emergencyStop = false
6      rollerStart()
7      emergencyStop = getEmergencyStop()
8      i = i + 1
```

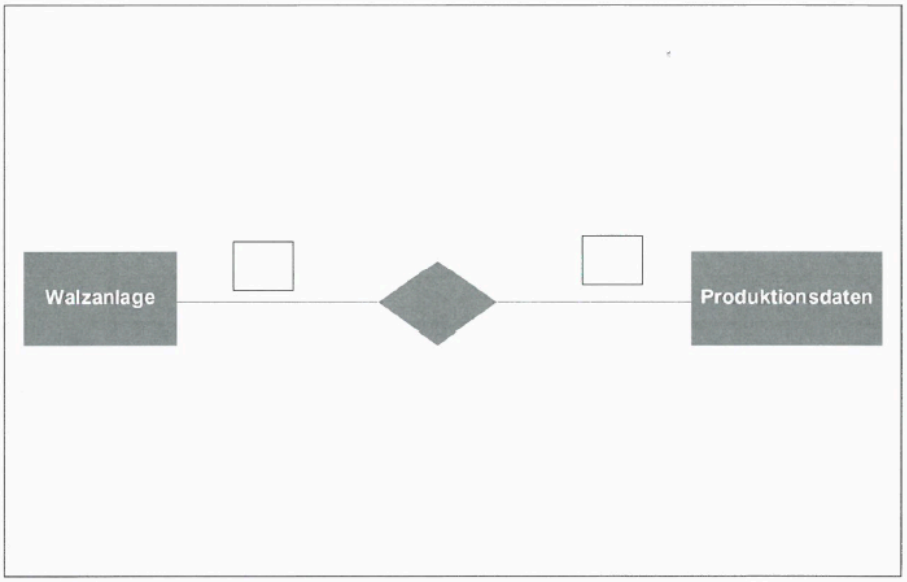
Zeile 4, 6 und 8 je 1 Punkt, Zeile 5 und 7 je 2 Punkte

Hinweis:
Die Zeilen 6, 7 und 8 dürfen in der Reihenfolge vertauscht werden.

c) Für die Produktion von Wellpappen ist die vorhandene Datenbank zu erweitern. Die Firma hat sich für ein SQL-fähiges relationales Datenbanksystem entschieden, in der die nachfolgenden Bedingungen berücksichtigt werden sollen. Die Speicherung der Datenbank wird auf dem Hostrechner „Steuerungs-PC" realisiert. In einer ersten Unterredung werden die zu speichernden Informationen definiert.
In dieser Datenbank sollen nur die Zusammenhänge zwischen den Walzanlagen, den Produktionsdaten abgebildet werden.

In der Produktionshalle sind mehrere Walzanlagen vorhanden. Diese jeweiligen Walzanlagen können Wellpappen mit unterschiedlichen Dicken (z. B. kleiner 4 mm, 4-8 mm, 8-12 mm) herstellen. In der Datenbank soll gespeichert werden, welche Walzanlage für welche Dicken (Spezifikation) verwendet werden kann. Außerdem soll das Baujahr, die Bezeichnung und eine eindeutige Maschinennummer gespeichert werden.
Für jede Walzanlage sollen die entsprechenden Produktionsdaten (Breite, Länge, Dicke und Anzahl) mit dem jeweiligen Zeit-Stempel abgespeichert werden.
Vervollständigen Sie das vorgegebene Entity-Relationship-Modell (kurz: ERM) für diese Datenbank mit allen erforderlichen Attributen und Kardinalitäten. 6 Punkte

Hinweis; Die eventuell benötigten Fremdschlüssel müssen nicht in diesem Entwurf eingetragen werden. Die Kardinalität zwischen den beiden Tabellen soll auf die entsprechenden Beziehungslinien eingetragen werden.

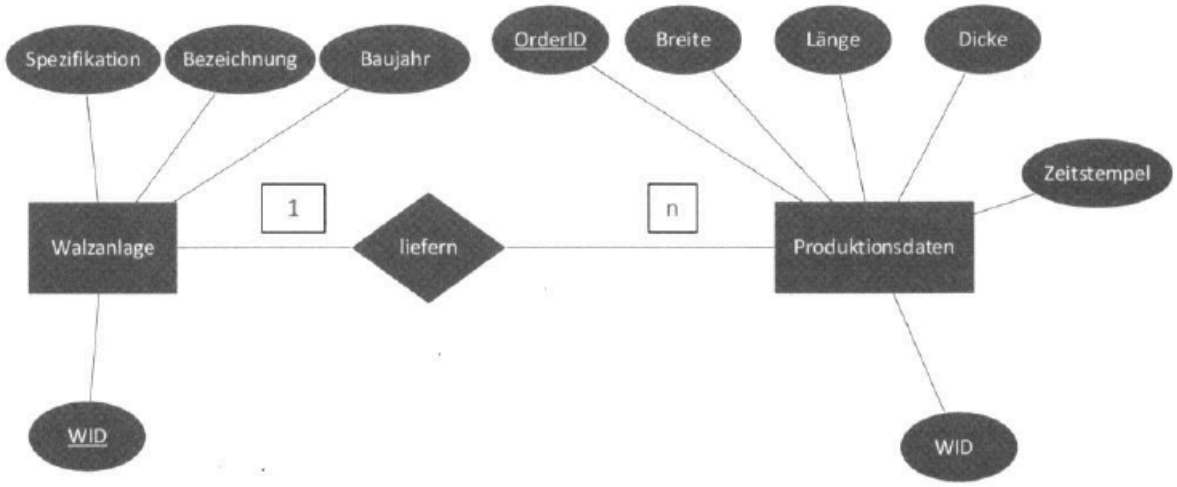


Hinweise: Tabelle (Chen-Notation)

Bezeichnung	Darstellung
Entity-Typ	Entity-Typ Name
Attribut	Attribut Name
Primärschlüssel	PK-Name
Beziehung (Relation, Relationship, Assoziation)	1 Beziehung 1 1 Beziehung n n Beziehung m

PK bezeichnet ein Primärschlüsselattribut, FK ein Fremdschlüsselattribut, Primärschlüsselattribute werden unterstrichen, Fremdschlüsselattribute werden durch ein nachgestelltes Hash-Zeichen (#) kenntlich gemacht.

Kardinaliät:	1 Punkt
Kennzeichnung Primärschlüssel in beiden Tabelle	2 Punkte
Attribute:	3 Punkte



Hinweise: **Fremdschlüssel#** optional, **nicht** in der Aufgabenstellung gefordert