

Inhalt

Datenbanken ?

2

IHK Aufgaben – IHK da cikmis sorular:

25-49

Was ist eine Datenbank?

Veritabanı Nedir?

Database Nedir?

Diğer bir ismi ile veri tabanı olarak bilinen database, adından da anlaşılacağı üzere verilerin belli bir alanda depolanması anlamına gelmektedir. Özellikle de günümüzde devlet kurumları bünyesinde bulunan birçok bilgi bu şekilde veritabanı sistemi ile saklanmaktadır. Belli bir düzen gerektiren bu sistemi aynı zamanda özel bir takım kuruluşlar da kullanmaktadır. Bu düzen sayesinde milyonlarca kişinin bilgilerine çok kısa bir süre içerisinde ulaşma imkânı sağlanabilmektedir. Okul kayıtları, nüfus müdürlükleri, banka ya da diğer birçok özel kurumların farklı insanlara verdikleri hizmetleri daha hızlı bir hale getirmesi için veri tabanı sistemini her daim kullanmaktadırlar.

Veritabanı genellikle bir bilgisayar sisteminde elektronik olarak depolanan yapılandırılmış bilgi veya veriden oluşan düzenli bir koleksiyondur. **Veritabanı genellikle bir veritabanı yönetim sistemi (DBMS) ile kontrol edilir.** Veri ve DBMS ve aynı zamanda bunlarla ilişkili uygulama yazılımları bir araya getirildiğinde sıklıkla yalnızca veritabanı olarak kısaltılan veritabanı sistemi olarak ifade edilir.

Günümüzde operasyonda kullanılan en yaygın veritabanı türlerindeki veri genellikle işlemeyi ve veri sorgulamayı verimli hale getirmek üzere bir dizi tablodaki satırlarda ve sütunlarda modellenir. Böylece veri kolayca erişilebilir, yönetilebilir, değiştirilebilir, güncellenebilir, kontrol edilebilir ve organize edilebilir hale getirilir. Çoğu veritabanında veri yazma ve sorgulama için yapılandırılmış sorgu dili (SQL) kullanılır.



Yapılandırılmış Sorgu Dili (SQL) nedir?

SQL verileri sorgulamak, değiştirmek ve tanımlamak ve aynı zamanda erişim kontrolü sağlamak üzere neredeyse tüm ilişkisel veritabanlarında kullanılan bir programlama

dilidir. SQL ilk olarak 1970'li yıllarda Oracle'ın büyük katkıları ile IBM'de geliştirilmiş ve daha sonra SQL ANSI standardı uygulanmış ve SQL IBM, Oracle ve Microsoft gibi şirketlerden pek çok uzantının temelini atmıştır. SQL günümüzde oldukça yaygın bir şekilde kullanılsa da yeni programlama dilleri geliştirilmeye başlandı.

Veritabanının evrimi

Veritabanları, 1960'lı yılların başlarında ortaya çıkmalarından bu yana radikal bir evrim sürecinden geçmiştir. Hiyerarşik veritabanı gibi navigasyonel veritabanları (ağaç benzeri bir modeli temel alır ve yalnızca bir kaynaktan çoklu alıcıya doğru ilişkiye olanak sağlar) ve ağ veritabanı (çoklu ilişkilere olanak sağlayan daha esnek bir model), verileri depolamak ve değiştirmek üzere kullanılan orijinal sistemlerdi. Bu ilk sistemler kolay olsa da esnekti. 1980'li yıllarda [ilişkisel veritabanları](#) popüler oldu ve ardından 1990'lı yıllarda [nesne odaklı veritabanları](#) ortaya çıktı. Daha yakın bir zamanda ise, internetin büyümesine ve yapılandırılmamış verilerin daha hızlı bir şekilde işlenmesine duyulan ihtiyaca yanıt olarak [NoSQL veritabanları](#) kullanılmaya başlandı. Günümüzde ise [bulut veritabanları](#) ve [kendi kendini yöneten veritabanları](#) veri toplama, depolama, yönetme ve kullanma konusunda çığır açıyor.

Veritabanı ve elektronik tablo arasındaki fark nedir?

Hem veritabanları hem elektronik tablolar (ör. Microsoft Excel) bilgi depolamak için uygun yöntemlerdir. Aralarındaki temel farklar şu şekilde sıralanabilir:

- **Veri toplama ve değiştirme yöntemi**
- **Verilere erişebilen kişiler**
- **Depolanabilecek veri miktarı**

Elektronik tablolar orijinal olarak tek kullanıcı için tasarlanmış olup özellikleri bu amacı yansıtmaktadır. Muazzam düzeyde karmaşık veri manipülasyonuna ihtiyaç duymayan tek veya az sayıda kullanıcı için harika bir seçenektir. Diğer yandan veritabanları ise çok daha yüksek miktarda, kimi zaman devasa miktarlarda bilgi koleksiyonlarını barındırmak üzere tasarlanmıştır. Veritabanları, oldukça karmaşık bir mantık ve dil kullanımıyla aynı anda birden fazla kullanıcının hızlı ve güvenli bir şekilde verilere erişmesine ve veri sorgulamasına olanak sağlar.

Veri Tabanı İçin Yönetim

- Türkçe adı ile veri tabanı yönetim sistemi olarak da bilinen, (Database Management system) şu an için birbirinden farklı programlar ile piyasa üzerinde bulunan ve elektronik ortamda tutulan verileri kolay erişebilir hale getiren bir sistemler dizisidir. Bu sistem sayesinde değişik birçok bilgi özel bir katalog haline getirilebilir.
- Bunlar sayesinde her bilgi adına tablo oluşturulabilir ve bu tablolar üzerinden yeni kayıt ekleme, silme ya da güncelleme yapmak mümkün hale gelebilir.
- Hangi verilere kimin ne kadar erişim izni olabilecek bunun kategorisini yapmak.
- Her ayrı veri için raporlama ve sorgulama imkanı yakalamak.
- İzinsiz girişlere karşı engel oluşturmak...

- Tüm verilerin yedek sistemlerini oluşturup saklayabilmek gibi daha birçok değişik analiz ve düzenleme yapılabilir.
- Database, yani veri tabanı programları her şirket ya da devlet kurumları adına farklı şekillerde kullanılabilir ve ona göre bir kayıt düzeni sağlanabilir. Bu sayede milyonlarca bilgi içerisinde hangisi isteniyor ise ona birkaç saniye içerisinde ulaşılabilir. Kademeli olarak konulan erişim izin seviyeleri sayesinde ise daha önemli bilgiler sağlam bir şekilde korunabilmektedir.

En Fazla Kullanılan Veritabanı (Database) Sistemleri



Kullanım durumlarına, kaliteleri ve sunduğu hizmetlere göre veri tabanı programları ücretli ya da ücretsiz olarak farklılıklar yaratmaktadır. Örneğin açık kaynak kodlu olarak hizmete açılmış database programları bulunurken, daha detaylı ve güvenli programlar ise profesyonel bir kullanım isteyen şirketler ile devlet kurumları adına tüm yıl boyunca online hizmet veren veri tabanı programları da bulunmaktadır. Şu an için piyasa üzerinde değişik veri tabanı programları mevcuttur. Bunlar arasında en çok bilinenlerinden bazıları, , gibi daha birbirinden farklı veri tabanı programları bulunabilmektedir. Burada önemli olan kullanım derecesidir. Yani oldukça özel bilgiler içeren ve erişimin zor olması istenen programlar ile akademik bir kayıt tutmak adına sunulan ücretsiz programlar da olabilir.

Database (Veri Tabanı) Türleri

İlişkisel veritabanı : İlişkisel veritabanları 1980'li yılların sonlarında piyasada hakimiyet kazandı. İlişkisel veritabanındaki öğeler, sütunlar ve satırlardan oluşan bir tablo kümesi şeklinde organize edilir. İlişkisel veritabanı teknolojisi, yapılandırılmış bilgilere en verimli ve esnek şekilde erişim sağlar.

NoSQL veritabanı : NoSQL veya ilişkisel olmayan bir veritabanı, yapılandırılmamış ve yarı yapılandırılmış verilerin depolanmasına olanak sağlar (veritabanına eklenen tüm verilerin nasıl oluşturulması gerektiğini tanımlayan ilişkisel veritabanının zıttıdır). NoSQL veritabanları, web uygulama yazılımlarının daha yaygın ve daha karmaşık hale gelmesi ile birlikte popülerlik kazandı.

Grafik veritabanı : Grafik veritabanı, verileri birimler ve birimler arasındaki ilişkiler açısından depolar.

Nesne odaklı veritabanı : Nesne odaklı bir veritabanındaki bilgiler, tıpkı nesne odaklı programlamada olduğu gibi nesneler biçiminde temsil edilir.

Dağıtılmış veritabanı : Dağıtılmış veritabanı, farklı yerlerde bulunan iki veya daha fazla dosyadan oluşur. Veritabanı, farklı ağlara yayılan ya da aynı fiziksel konumda yer alan birden fazla bilgisayarda depolanabilir.

Veri ambarları : Merkezi bir veri havuzu olan veri ambarı, özel olarak hızlı sorgulama ve analiz amaçlarıyla tasarlanmış bir veritabanı türüdür.

OLTP veritabanı : OLTP veritabanı, birden fazla kullanıcı tarafından çok sayıda işlemin gerçekleştirilmesi için tasarlanmış hızlı ve analitik bir veritabanıdır.

Açık kaynak veritabanları

- Açık kaynak veritabanı sistemi, kaynak kodu açık kaynak olan bir sistemdir. Bu tür veritabanları SQL veya NoSQL veritabanları olabilir.

Bulut veritabanları

- Bir [bulut veritabanı](#) özel, genel veya hibrit bulut bilgi işlem platformunda bulunan yapılandırılmış veya yapılandırılmamış bir veri koleksiyonudur. İki tür bulut veritabanı modeli vardır: geleneksel veritabanı ve veritabanı hizmeti (DBaaS). DBaaS sayesinde yönetim görevleri ve bakım işlemleri servis sağlayıcı tarafından gerçekleştirilir.

Çoklu model veritabanı

- Çoklu model veritabanları, farklı veritabanı türlerini tek bir entegre arka uçta buluşturur. Buna göre çeşitli veri türlerini içerebilirler.

Belge/JSON veritabanı

- Belge odaklı bilgilerin depolanması, alınması ve yönetilmesi için tasarlanan [belge veritabanları](#) verileri satırlar ve sütunlar yerine JSON biçiminde depolamak için modern bir yöntem sunar.

Kendi kendini yöneten veritabanları

- En yeni ve en ezber bozan veritabanı türü, kendi kendini yöneten veritabanları (aynı zamanda otonom veritabanları olarak bilinir), geleneksel olarak veritabanı yöneticileri tarafından gerçekleştirilen ince ayar, güvenlik, yedekleme, güncelleme ve diğer rutin yönetim görevlerini otomatikleştirmek üzere makine öğreniminden yararlanan bulut tabanlı çözümlerdir.

Veritabanı yazılımı nedir?

Veritabanı yazılımı dosya ve kayıt oluşturmayı, veri girişini, veri düzenlemeyi, güncellemeyi ve raporlamayı kolay hale getirerek veritabanı dosyalarını ve kayıtları oluşturmak, düzenlemek ve muhafaza etmek için kullanılır. Yazılım veri depolama, yedekleme ve

raporlama, çok erişimli denetleme ve güvenlik de sağlar. Hırsızlık olayları arttığından, güçlü veritabanı güvenliği günümüzde özellikle önem kazanmıştır. Veritabanı yazılımı bazen "veritabanı yönetim sistemi" (DBMS) olarak da adlandırılmaktadır.

Veritabanı yazılımı veri yönetimini daha kolay hale getirir ve kullanıcıların yapılandırılmış bir forma veri depolamalarına ve sonra buna erişmelerine olanak tanır. Genellikle veri oluşturmaya ve yönetmeye yardımcı olan bir grafiksel arayüzdür ve bazı durumlarda kullanıcılar veritabanı yazılımı kullanarak [kendi veritabanlarını oluşturabilir](#).

Veritabanı yönetim sistemi (DBMS) nedir?

Veritabanları tipik olarak veritabanı yönetim sistemi (DBMS) olarak bilinen kapsamlı bir veritabanı yazılım programını gerektirir. DBMS, veritabanı ve son kullanıcıları ya da programlar arasında bir arayüz işlevi görerek kullanıcıların bilgilerin nasıl organize ve optimize edildiğini yönetmesine, bilgileri almasına ve güncellemesine olanak sağlar. DBMS aynı zamanda veritabanlarına ilişkin gözetim ve kontrol faaliyetlerini kolaylaştırarak performans izleme, ince ayar ve yedekleme ve kurtarma gibi çeşitli yönetim operasyonlarının gerçekleştirilebilmesini sağlar.

Popüler veritabanı yazılımlarına ya da DBMS'lere MySQL, Microsoft Access, Microsoft SQL Server, FileMaker Pro, Oracle Database ve dBASE örnek olarak gösterilebilir.

3.1 Wissen:

Sie müssen u.a. folgende Begriffe verstehen, in Zusammenhänge einordnen können und mit eigenen Worten beschreiben können:



- Entity-Relationship-Modell, ERM, ERD
- Kardinalitäten
- Relationenmodell, Tabellenmodell
- Normalisierung
- SQL

Sie müssen mit Ihrem Wissen folgende Fragen beantworten können:

- Warum sind Datenbanken/ Datenbankmanagementsysteme wichtig, was tun die? Welche wichtigen Funktionen und Eigenschaften haben die? (Veritabanları/veritabanı yönetim sistemleri neden önemlidir, ne yaparlar? Hangi önemli işlevlere ve özelliklere sahipler?)
- Welche Arbeitsschritte müssen ausgeführt werden, um praktisch eine Datenbank zu konzipieren und zu erstellen? (Bir veritabanını pratik olarak tasarlama ve oluşturma ile ilgili adımlar nelerdir?)
- Welche Hilfsmittel stehen zur Verfügung, um mit Datenbanken/ Daten arbeiten zu können? (Veritabanlarıyla/verilerle çalışmak için hangi araçlar mevcuttur?)

Phasen des Datenbankentwurfs - Überblick

- **Ziel des Datenbank-Entwurfs:** Abbildung eines genau definierten Ausschnitts der realen Welt (Unternehmen, Organisation) in einem Modell, welches die Daten und ihre logischen Verbindungen darstellt
- es wird eine Datenbankstruktur konzipiert, in der eine Software-Anwendung ihre Daten abspeichern kann
- als Ergebnis liegt ein Datenbankentwurf vor, der anschließend in einer Datenbank-Software praktisch eingerichtet werden könnte



- **4 Phasen:**

- ① **Informations-Analyse** (des Ist-Zustands im Unternehmen), Analyse der konkreten Situation im Unternehmen → Schreiben des Pflichtenheftes
 - ② **konzeptioneller Entwurf** → Daten-Modell → konzeptionelles Modell = ERM
 - ③ **logischer Entwurf** → Datenbank-Schema → relationales Schema = Tabellen-Schema
 - ④ **physischer Entwurf** → Schema für technische Umsetzung
- Anwendung spezieller Methoden, Regeln und Software-Werkzeuge

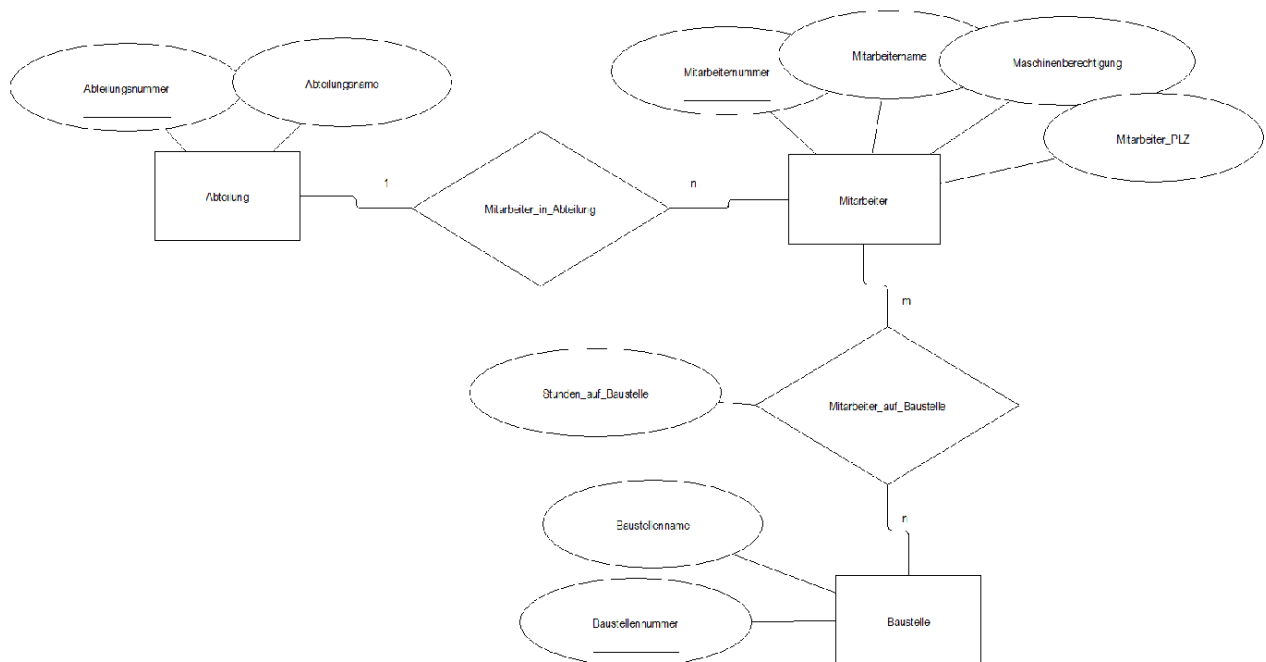
Anforderungen:

- **vollständig:** alle notwendigen Daten sind enthalten
- **korrekt:** sachlich richtig
- **konsistent:** widerspruchsfrei, keine Festlegung darf einer anderen widersprechen
- **minimal:** nur das absolut Notwendige
- **lesbar und anpassbar:** durch den Anwender

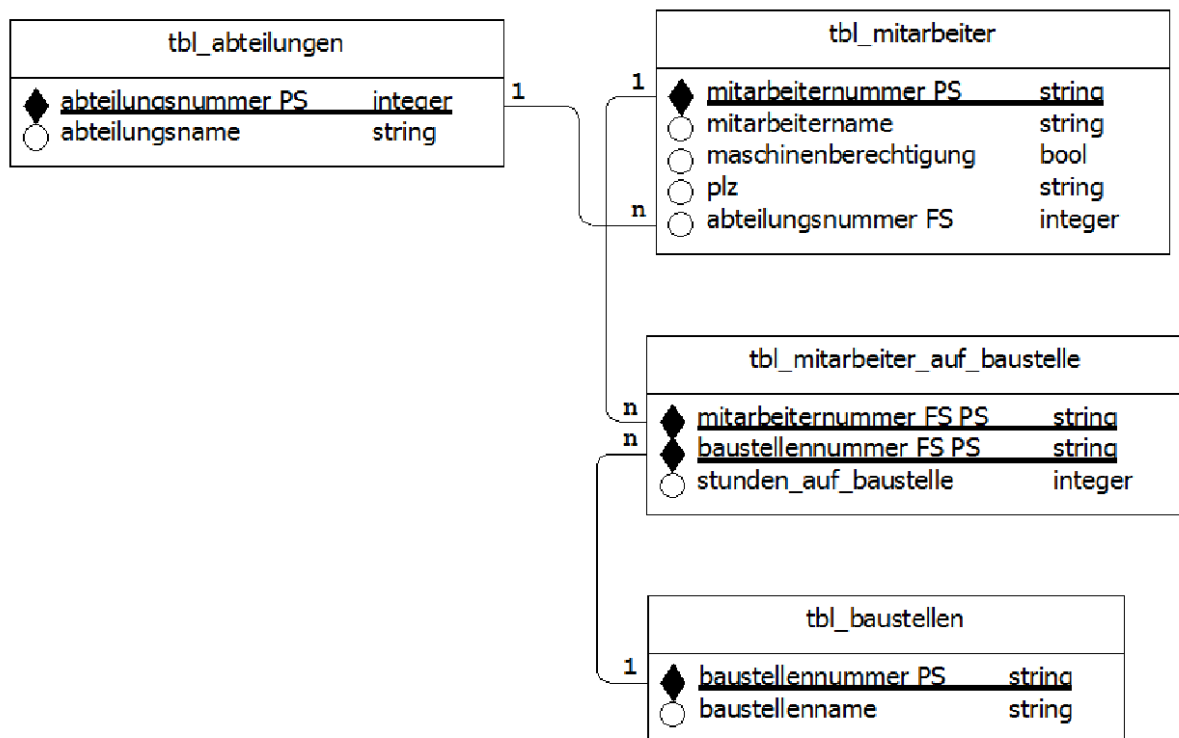
1.2.1 zu Phase 1: Informations-Analyse

- **Ziel = Endergebnis:** fertige Sammlung der vollständigen und konsistenten Informationen über den in der Datenbank abzubildenden Unternehmensbereich als Grundlage für den anschließenden konzeptionellen Entwurf → diese Sammlung heißt "**Pflichtenheft**"
- sollte auf einem **Lastenheft** des Auftraggebers basieren
- **(wesentlicher) Inhalt bei Datenbankentwicklung:**
 - Begriffe, Definitionen, Abgrenzungen finden und verbindlich vereinbaren
 - Identifizieren und Beschreiben der für die künftige Software-Anwendung benötigten Sachverhalte: Träger von Daten/ Informationen und deren Beziehungen/ Abhängigkeiten untereinander z.B. Personen, Firmen, Produkte, Ereignisse, Rechnungen, Zahlungsvorgänge, ...
 - ➔ also Objekte, deren Eigenschaften/ Attribute sowie deren Tätigkeiten/ Funktionen
- Methoden: Klassifikation, Generalisierung/ Spezialisierung, Aggregation

1.2.2 zu Phase 2: Konzeptioneller Entwurf: ERM



1.2.3 zu Phase 3: Relationen-Modell, Tabellen-Schema



1.2.4 zu Phase 4: Praktische Umsetzung des relationalen Entwurfs im Datenbanksystem

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like `tbl_abteilungen`, `tbl_baustellen`, and `tbl_mitarbeiter`. The main editor on the right shows the following SQL code:

```

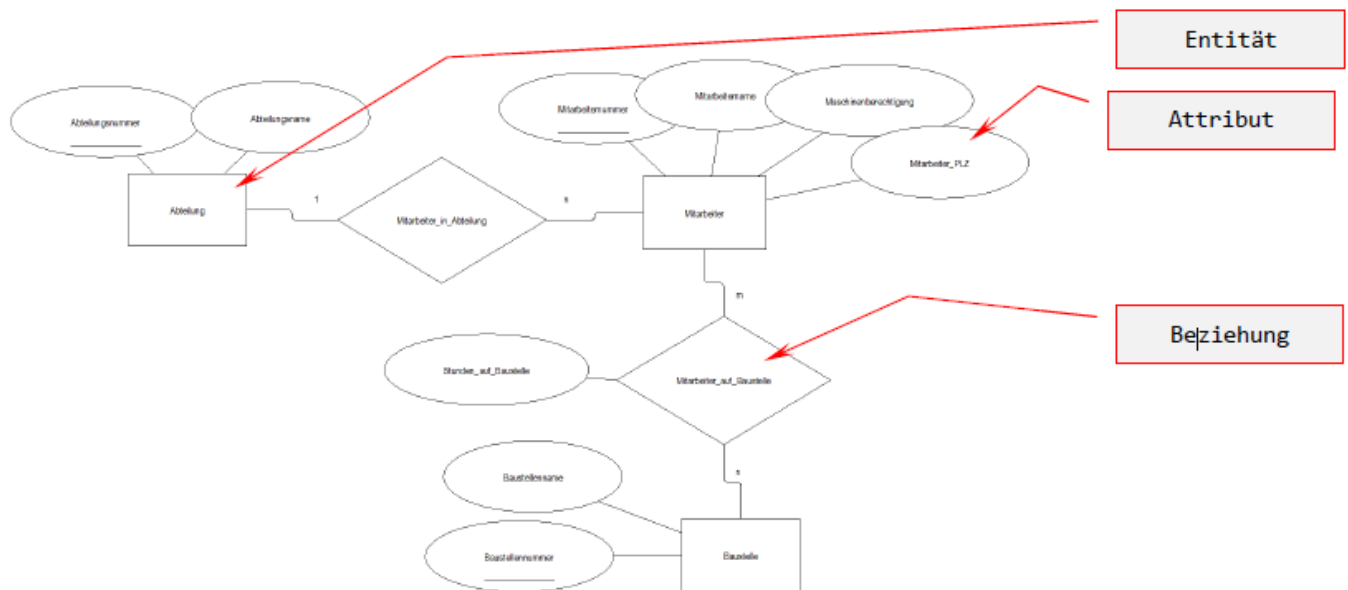
6 • create database hochbau;
7 • use hochbau;
8
9 • create table tbl_abteilungen
10   (abteilungsnummer int primary key,
11    abteilungsname char(50));
12
13 • create table tbl_baustellen
14   (baustellennummer char(4) primary key,
15    baustellensname char(150));
16
17 • create table tbl_mitarbeiter
18   (mitarbeiternummer char(4) PRIMARY KEY,
19    nachname char(50) NOT NULL,
20    vorname char(50) NOT NULL,
21    geburtsdatum date NOT NULL,
22    geschlecht char(1) NOT NULL,
23    matrikelnummer int NOT NULL);

```

2 Konzeptioneller Entwurf: ERM

2.1 Konzeptioneller Entwurf → Überblick

Ein ERM-Diagramm besteht aus folgenden wichtigen Elementen:



Infos zum Entity(Varlik) Relationship Model:

- Entity-Relationship-Model entwickelt von P.P. Chen 1976
- dient der graphischen Beschreibung und Strukturierung der Daten eines realen Anwendungsszenarios, fachliche Datenmodellierung → ER-Diagramm, als Basis für relationale Datenbankmanagementsysteme
- konzeptueller Charakter, hard- und software unabhängig
- Grundlagen:
 - ERM enthält Klassen von Datenobjekten = Entity-Klassen



- **Eigenschaften der Entities = Attribute und deren Wertebereiche/ Domänen**

- **Primärschlüssel kann bestimmt werden**
- **Klassen von Beziehungen der Objekte untereinander und deren Eigenschaften = Relationships**

ERM = Entity Relationship Model enthält folgende 2 Hauptelemente:

- **Entity, Entität, Entity Class, Entitätsklasse (alle Begriffe beziehen sich auf dasselbe)**

- **Entity-Klasse hat einen eindeutigen Namen**
- **Eigenschaften = Attribute der Entity-Klassen**
- **Attribute haben einen eindeutigen Namen und einen Wertebereich = Domain**

- **die Attribute, die die Objekte einer E.-Klasse eindeutig voneinander unterscheiden → Primärschlüssel → werden im ERM unterstrichen**

- **Relationship Class, Beziehungsklassen (alle Begriffe beziehen sich auf dasselbe)**

- **Eigenschaften = Attribute der Relationship-Klassen**

- **gekennzeichnet durch die Anzahl, wie oft ein Objekt maximal an der Beziehung beteiligt sein kann → Kardinalitäten = eine wichtige strukturelle Integritätsbedingung (bir nesnenin ilişkiye katılabileceği maksimum sayı ile karakterize edilir → önemlilikler = önemli bir yapısal bütünlük kısıtlaması)**

Überblick und Einordnung der Arbeitsschritte für die ER-Modellierung:



- (1) Entitäten (Objekte) und Relationships (Beziehungen) identifizieren
- (2) Festlegen der Entity-Klassen und Relationship-Klassen
- (3) für die Entity-Klassen und die Relationship-Klassen die Attribute und ihre Wertebereiche (=Domänen) ermitteln
- (4) Aus den Attributen den jeweiligen Primärschlüssel (=Identitätsschlüssel) für alle Entity- Klassen ermitteln
- (5) Festlegen/ Bestimmen der Beziehungskardinalitäten
- (6) ERD (Entity Relationship Diagram) zeichnen
- (7) Attribute und ihre Wertebereiche in Tabellen eintragen
- (8) Relationen-Modell erstellen
- (9) Datenbank auf Basis des Relationen-Modells im realen System anlegen

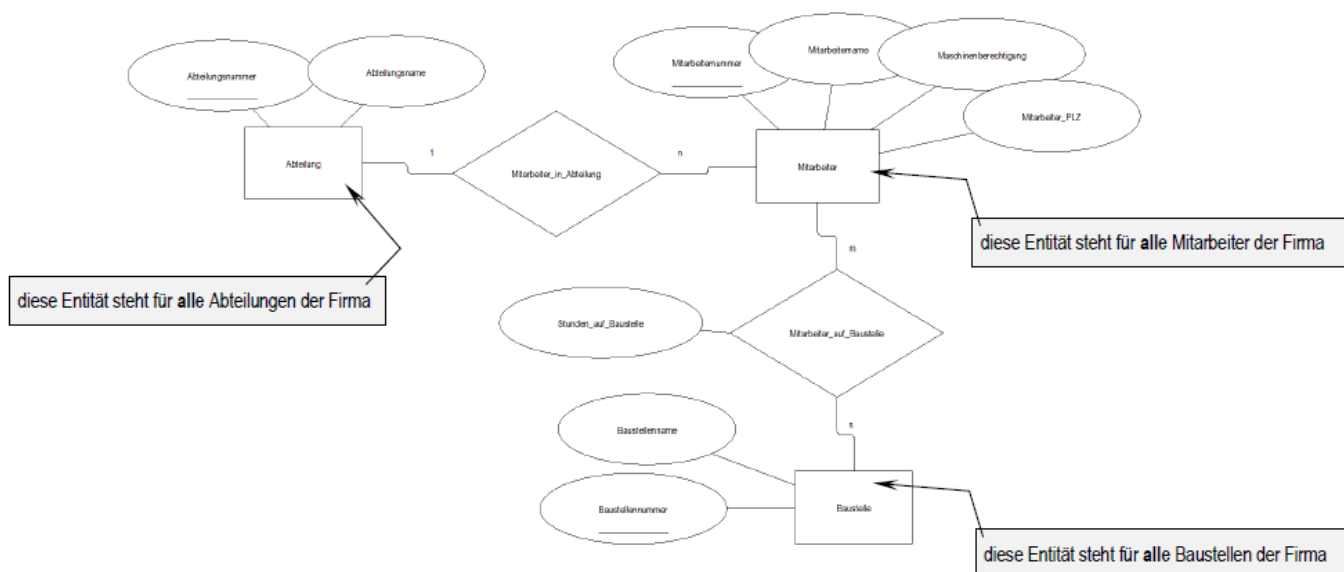
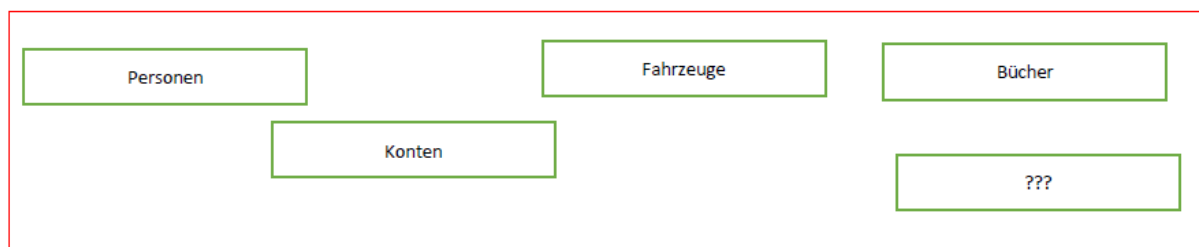
2.2 Entitäten ermitteln

- **Untersuchung der Situation im Unternehmen, wo die Datenbank verwendet werden soll**



- **beteiligte Objekte ermitteln**
 - **Wie erkennt man die Objekte?**

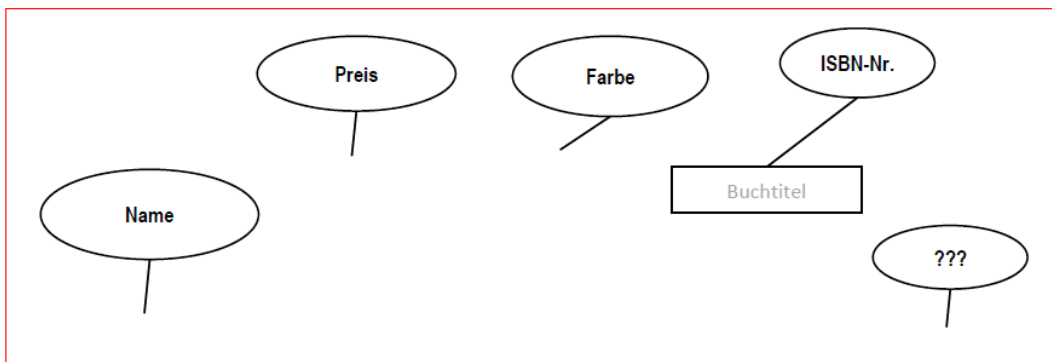
- **Entität (dt.)/ Entity/ Entities:** allgemein der Gegenstand, Objekt in der realen Welt
 - gleichartige Entities (mit gemeinsamen Eigenschaften) werden zu Entity-Mengen/ Objekttypen / Entity-Klassen zusammengefaßt
 - E. müssen unterscheidbar sein und sind unterscheidbar durch ihre Attributwerte bzw. durch Attributkombinationen
- **Identifizierung der Entities über Abstraktionsmethoden:**
 - Klassifikation
 - Generalisierung/ Spezialisierung
- **Beispiele:**



2.3 Attribute ermitteln

- die gemeinsamen Eigenschaften der Entity-Klassen werden Attribute genannt:
 - jedes Attribut hat einen eindeutigen Namen
 - jedes Attribut hat einen speziellen Wertebereich = wird auch Domäne genannt
- Wie werden die Attribute ausgewählt:
 - die Attribut-Werte beschreiben die Eigenschaften der einzelnen Entities in der Klasse

- über die Attribut-Werte müssen die einzelnen Entities unterscheidbar sein
- u.U. müssen künstliche Attribute eingeführt werden
- es müssen die Attribute gewählt werden, die für das Geschäftsmodell entscheidend sind
- spezieller Attribut-Wert ist NULL ($\neq 0$, 0 ist etwas anderes): bezeichnet nicht definierten Wert bzw. Zustand eines Attributs
- Darstellung im Diagramm
- Beispiele:



Attribute und Wertebereiche/ Domänen:

- Domänen/ Wertebereiche werden (vorrangig) über Datentypen definiert
- Beispiele für Domänen:
- CARDINAL = natürliche Zahlen (1, 2, 3, ...)
- INTEGER = ganze Zahlen (-2, -1, 0, 1, 2, ...)
- NUMERIC = Dezimalzahlen (2.5, 3.1415, ...)
- STRING = Zeichenketten, Text
- BOOLEAN = logische Werte, Wahrheitswerte (1|0, ja|nein)
- CHAR = einzelne Buchstaben bzw. Zeichen
- DATE/ DATUM/ UHRZEIT = Datum und Zeit
- Hinweis: in ausgewählten DBMS können Domänen auch zusätzlich durch die Konfiguration weiterer Eigenschaften und Constraints bestimmt bzw. eingeschränkt werden.

Beispiel:

Entity-Klasse „Lehrer“	
Attribut	Domäne
Name	STRING
PersNr	INTEGER > 0
Wohnort	STRING
Geschlecht	{w',m'}
Geburtsjahr	INTEGER > 1947

Besondere Attribute:

- für die Erstellung des ERM sind zunächst die sogenannten "Primärschlüssel" wichtig (weitere besonderen Attribute werden in den noch kommenden Arbeitsschritten ermittelt)
- aus den Attributen der Entitäten werden zunächst die Schlüsselkandidaten (key candidate) ermittelt: ~ ist das Attribut oder die Attribut-Kombination, die jedes Mitglied einer Entität eindeutig identifiziert (Anahtar adaylar ilk olarak varlıkların niteliklerinden belirlenir: ~, bir varlığın her bir üyesini benzersiz şekilde tanımlayan nitelik veya nitelik kombinasyonudur.)
- aus den Schlüsselkandidaten wird der Primärschlüssel = primary key gewählt
- ggf. künstlichen Schlüssel erzeugen: Beispiele für künstliche Schlüssel: Mitarbeiter-Nummer, Artikel-Nummer, Konto-Nummer, Bankleitzahlen, Bestellnummern, Ausweisnummer, ...
- Attribute, die als Primärschlüssel dienen, werden unterstrichen
- Beispiele für Schlüsselkandidaten: ????

2.6 Beziehungen ermitteln

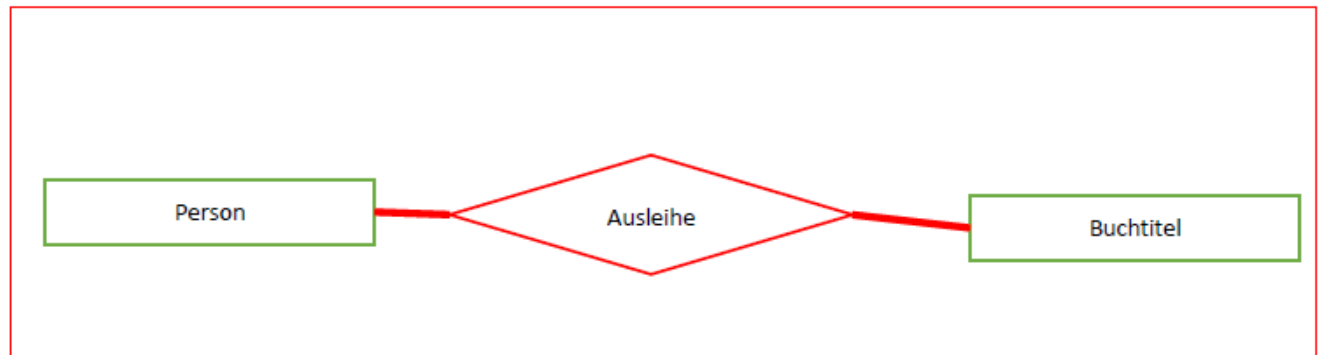
- Beziehungen müssen durch den Ersteller des ERM erkannt und beschrieben werden: wo stehen Objekte zueinander in Beziehungen durch Handlungen, Verwendungen, Funktionen
- Beziehungen mit eindeutigen Begriffen benennen
- Beziehungen aus beiden Richtungen beschreiben (bei zweistelligen Beziehungen)
- Relationships können Attribute haben
- R. besitzen aber keinen Primärschlüssel → sind eindeutig durch die Entitäten, die sie verbinden
- R. haben Kardinalitäten

Beispiele für Beziehungen:

Beschreibung:

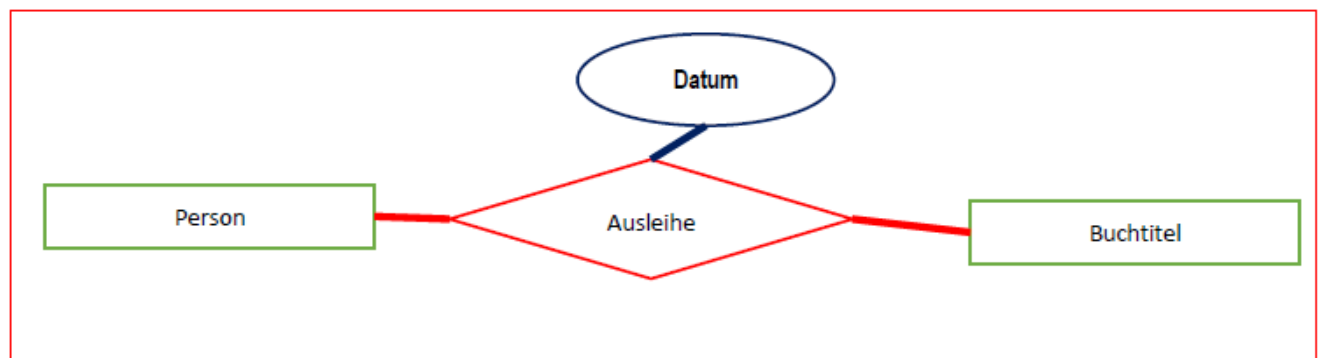
- Person leiht sich Buchtitel aus
- genauere Formulierung:

Mitglieder der Entitätsklasse 'Person' leihen sich Mitglieder der Entitätsklasse 'Buchtitel' aus



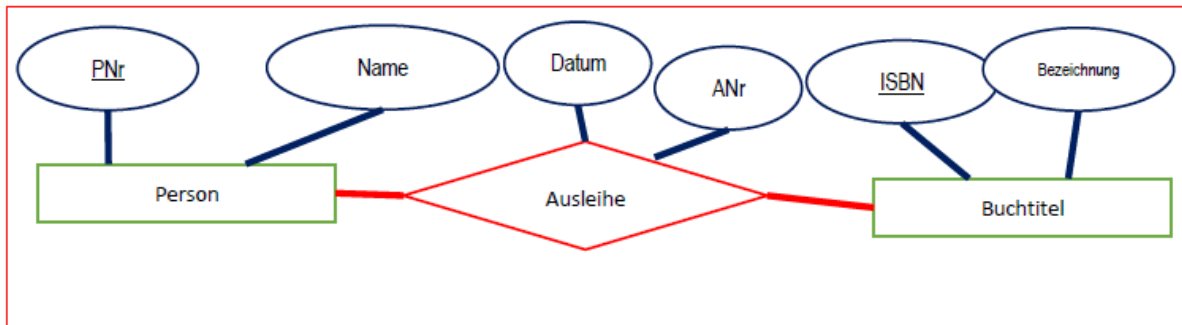
Beschreibung:

- Person leiht sich Buchtitel aus
- zur Ausleihe wird das Datum gespeichert
- genauere Formulierung:
Mitglieder der Entitätsklasse 'Person' leihen sich Mitglieder der Entitätsklasse 'Buchtitel' aus; die Beziehung hat das Attribut 'Datum'



Beschreibung:

- Person leiht sich Buchtitel aus
- zu Person, Ausleihe und Buchtitel existieren jeweils mehrere Attribute
- zu 'Person' und 'Buchtitel' existiert jeweils ein Primärschlüssel-Attribut
- Beziehung erhalten kein Primärschlüssel-Attribut



Kardinalitäten:

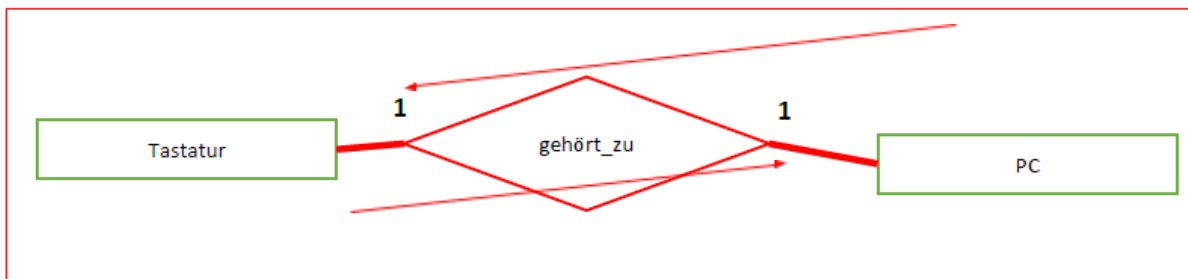
- Kardinalitäten = maximale Anzahl, Häufigkeit der an einer Beziehung beteiligten Entities (Önemlilikler = maksimum sayı, bir ilişkide yer alan varlıkların sıklığı)
- Beziehungen können durch die mögliche Anzahl = Kardinalität der Verbindungen klassifiziert werden
- wichtige strukturelle Integritätsbedingungen
- Wie werden K. dargestellt? Wie funktionieren K., wie werden sie bestimmt?
- Symbole: 1, m, n, c

Typen von Kardinalitäten:

- unterschiedliche Typen:
- eineindeutige Funktion → 1:1
- funktionale Abbildung → 1:n
- komplexe Abbildung → n:m
- mögliche Kardinalitäten:
- c: höchstens eine Verbindung, also 0 oder 1
- 1: genau eine Verbindung
- n: mindestens eine Verbindung, also 1 bis n (auch möglich mit m)
- cn: keine, eine oder mehr Verbindungen, also 0 bis n

Beziehungstyp: 1:1

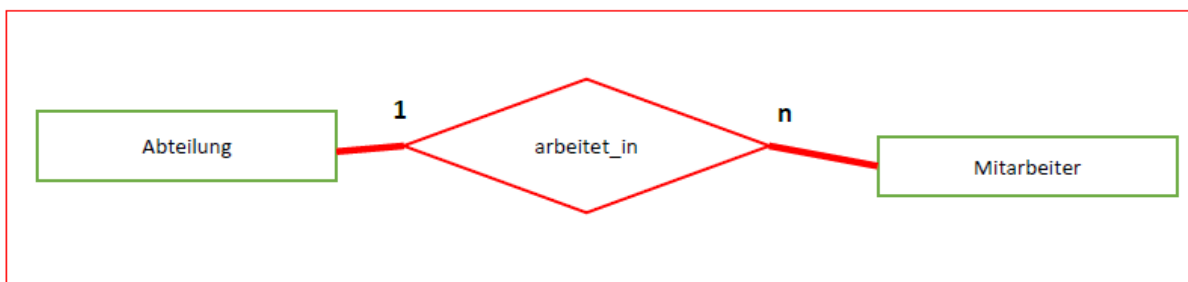
Beschreibung: jedes Mitglied der Entitätsklasse 'Tastatur' kann immer nur mit einem Mitglied der Entitätsklasse 'PC' eine Beziehung haben



Beziehungstyp: 1:n

Beschreibung:

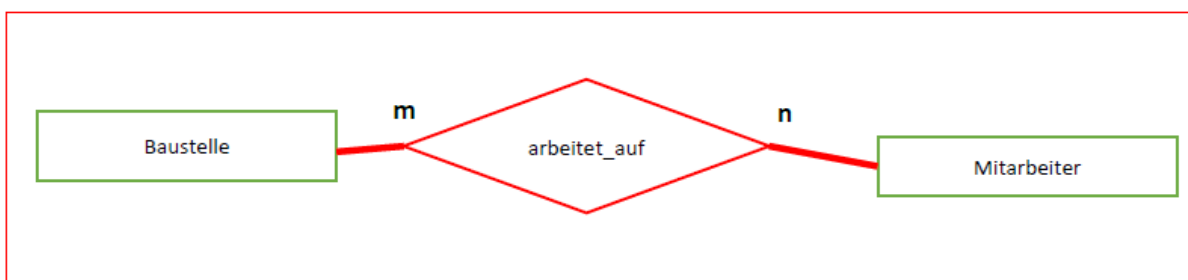
- jedes Mitglied der Entitätsklasse 'Abteilung' kann zu einem bis beliebig vielen Mitgliedern der Entitätsklasse 'Mitarbeiter' eine Beziehung haben
- jedes Mitglied der Entitätsklasse 'Mitarbeiter' kann nur zu einem Mitglied der Entitätsklasse 'Abteilung' eine Beziehung haben



Beziehungstyp: m:n

Beschreibung:

- jedes Mitglied der Entitätsklasse 'Baustelle' kann zu einem bis beliebig vielen Mitgliedern der Entitätsklasse 'Mitarbeiter' eine Beziehung haben
- jedes Mitglied der Entitätsklasse 'Mitarbeiter' kann zu einem bis beliebig vielen Mitgliedern der Entitätsklasse 'Baustelle' eine Beziehung haben



Önemlilikler = maksimum sayı, bir ilişkide yer alan varlıkların sıklığı

<https://app.diagrams.net>.

Verschaffen Sie sich einen Überblick über mögliche Schreibweisen von Kardinalitäten:

- [https://de.wikipedia.org/wiki/Kardinalit%C3%A4t_\(Datenbankmodellierung\)](https://de.wikipedia.org/wiki/Kardinalit%C3%A4t_(Datenbankmodellierung))
(abgerufen am 2022-01-21)

- <https://de.wikipedia.org/wiki/Entity-Relationship-Modell> (abgerufen am 2022-01-21)

- <https://de.wikipedia.org/wiki/Martin-Notation> (abgerufen am 2022-01-21)

1 Relationenmodell: Grundlagen

1.1 Das Relationen-Modell

- **Urheber E. F. Codd ca. 1970**
- **Basis bilden Relationen = Tabellen**
- **alle Daten werden als Werte in Relationen = zweidimensionalen Tabellen dargestellt**
 - **Spalten = Attribute**
 - **Zeilen = Tupel**
- **Benutzer-Zugriff auf die Daten erfolgt über deren Werte = Benutzer muß keine Kenntnisse über die Art und Weise der Speicherung haben, greift also nicht (direkt) auf Speicherort zu**
- **es existieren Operatoren für das Abrufen und Verarbeiten von Daten aus einzelnen und aus aufeinander verweisenden (referenzierenden) Tabellen**
- **Definition Datenbankschema**
- **Datenstrukturierung:**
 - **Beschreibung der logischen Aspekte der Daten, unabhängig von der anschließenden Verarbeitung in einer Software-Anwendung → Die Software-Anwendung bekommt ihre spezifische Sicht auf die Daten.**
 - **Definition von Integritätsbedingungen und Zugriffskontrollbedingungen**
- **formatierte, feste Datenstrukturen:**
 - **Beschreibung der zu speichernden Objekte durch Attribute und Attributwerte**
 - **Attribut besteht aus Attributbezeichner und einem Wert:**
 - **Beispiel: Objekt – Attribut – Attributwert**
 - **name = ‚Krause‘**
 - **kunden-nr = 101**

Prägen Sie sich die erweiterten Eigenschaften eines Relationalen Schemas ein:

- **alle Daten werden als Werte in Relationen = zweidimensionalen Tabellen dargestellt**
 - **logisch zusammengehörende Datensätze sind in derselben Tabelle z.B.**
- **...**
- **Spalten = Attribute (Attribut = Spaltenüberschrift) = Eigenschaft des Datenobjekts**
 - **Zeile = Datenobjekt = Datensatz = logisch zusammengehörende Daten = werden auch Tupel genannt**
- **zwischen den Datensätzen verschiedener Tabellen existieren logische Zusammenhänge/ Abhängigkeiten**

- eine Tabellen-Zeile = ein Datensatz = ein Tupel = ein Datenbankobjekt
 - Tupel werden durch ihre Attributwerte identifiziert.
 - Es sind nur atomare Attributwerte zugelassen.
 - Attribut bezieht seine Werte aus einem unstrukturierten Wertebereich → keine Mengen, Aufzählungstypen, Wiederholungsgruppen, ...
 - Die Reihenfolge der Tupel in der Relation ist nicht definiert.
 - Es erfolgt keine Addressierung der Datensätze z.B. durch Positionsangabe.
- eine Tabellen-Spalte entspricht einem Attribut
 - Die Reihenfolge der Attribute = Spalten ist nicht festgelegt.
 - Tupel werden eineindeutig durch den Attributwert identifiziert, der als Primärschlüssel festgelegt wurde.
 - Der Primärschlüssel wird im Rahmen des Datenbankentwurfs aus den Schlüsselkandidaten ausgewählt.
 - Für alle Schlüsselkandidaten gilt:
 - sie sind eindeutig
 - sie sind minimal

Ein relationales Schema enthält:

- Domänenregeln: mögliche Werte für Attribute werden eingeschränkt z.B. durch Festlegung von Datentypen (siehe weitere Infos im Anschluß)
- Primärschlüssel: identifizieren die einzelnen Tupel (= Tabellenzeile) eindeutig
- weitere Schlüssel: müssen eindeutig sein
- Fremdschlüssel: Attribut in einer Tabelle, das in einer anderen Tabelle als Primärschlüssel dient → Herstellen eindeutiger Beziehungen zwischen den Tabellen
- weitere Geschäftsregeln: Integritätsregeln, die sich durch logische Bezüge von Daten auf andere Daten ergeben

1.2.4 Das Relationen-Modell

- alle Daten werden in Relationen = zweidimensionalen Tabellen gespeichert: Spalten = Attribute, Zeilen = Tupel; die Tabellen können aufeinander verweisen = referenzieren



Kunden-Nr.	Name	(Bezeichner)	Ort
K011	Krause	(Wert)	Berlin
K099	Meier	10117	Berlin
K003	Müller	10120	Berlin

Spalte = Eigenschaft = Attribut

Kunden-Nr.	Name	PLZ	Ort
K001	Krause	10115	Berlin

Zeile = Tupel = zu speicherndes Objekt = Datensatz

Zeile = zu speicherndes Objekt = logisch zusammengehörende Daten

jedes Objekt:

- (1) hat Eigenschaften, die durch Werte bestimmt werden
- (2) kann über die Eigenschaften eindeutig bestimmt werden

Attribute

<u>Kunden-Nr</u>	Name	(Bezeichner)	PLZ
K001	Krause	(Wert)	04105

Primärschlüssel

Attributwerte

Spalte = Eigenschaft des zu speichernden Objekts = Attribut

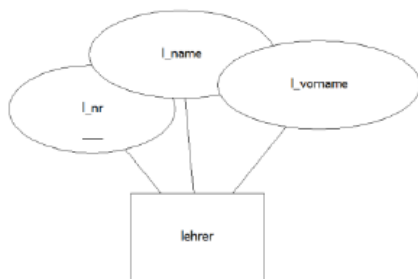
Spaltenüberschrift = Attributname



2.2 Schritt 1: alle Entitäten umwandeln in Tabellen

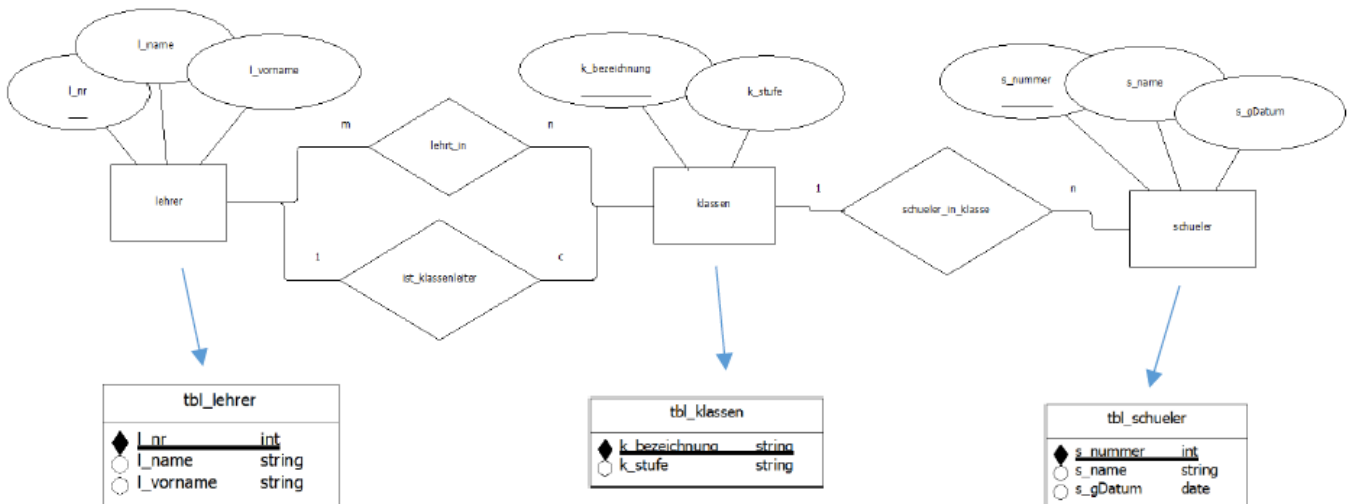
- jede einzelne Entität wird in jeweils eine Tabelle umgesetzt
- jedes Attribut wird zu einer Spalte in der jeweiligen Tabelle:

- die Spalten erhalten einen eindeutigen Namen
- jede Spalte erhält einen Wertebereich = Datentyp passend zu ihrer Verwendung
- der Primärschlüssel steht i.d.R. an erster Stelle und wird unterstrichen
- in den so entstehenden Tabellen werden später die zu speichernden Objekte mit je einer Tabellenzeile eingetragen, also je Objekt ein Datensatz/ Tupel
- Hinweise zur Darstellung des Symbols für die Tabelle:



tbl_lehrer	
<u>l_nr</u>	integer
l_name	string
l_vorname	string

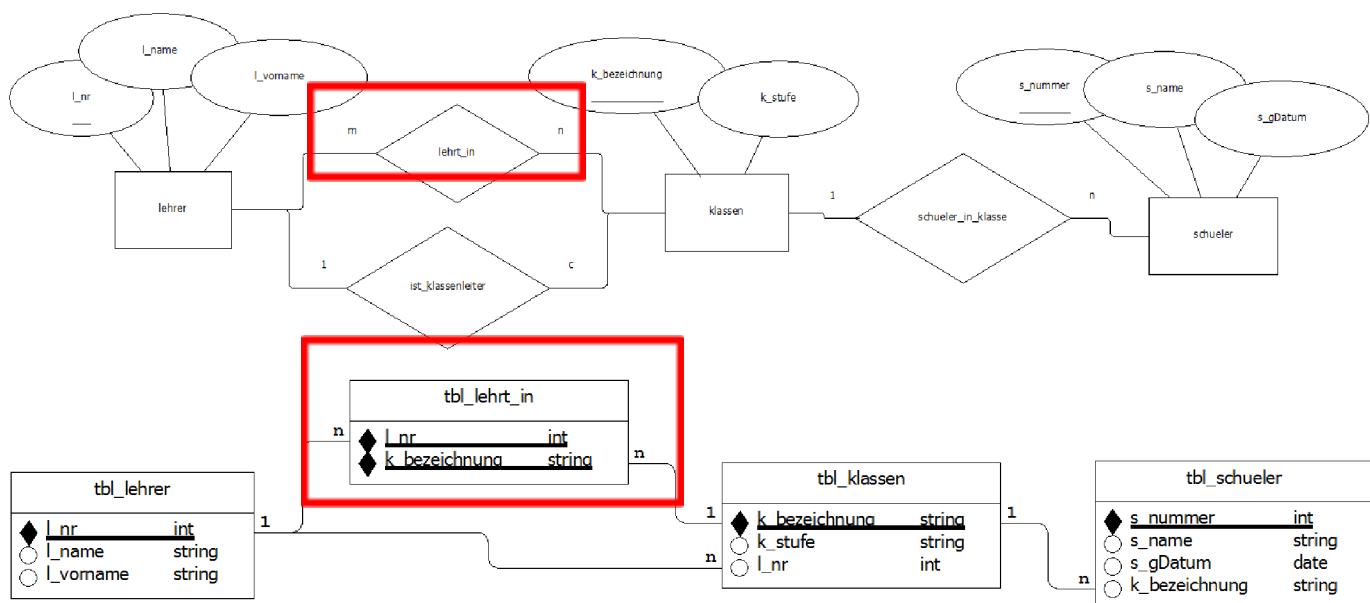
tbl_lehrer	
◆ <u>l_nr</u>	int
○ l_name	string
○ l_vorname	string



tbl_lehrer	
◆ <u>l_nr</u>	int
○ l_name	string
○ l_vorname	string

tbl_klassen	
◆ <u>k_bezeichnung</u>	string
○ k_stufe	string

tbl_schueler	
◆ <u>s_nummer</u>	int
○ s_name	string
○ s_gDatum	date

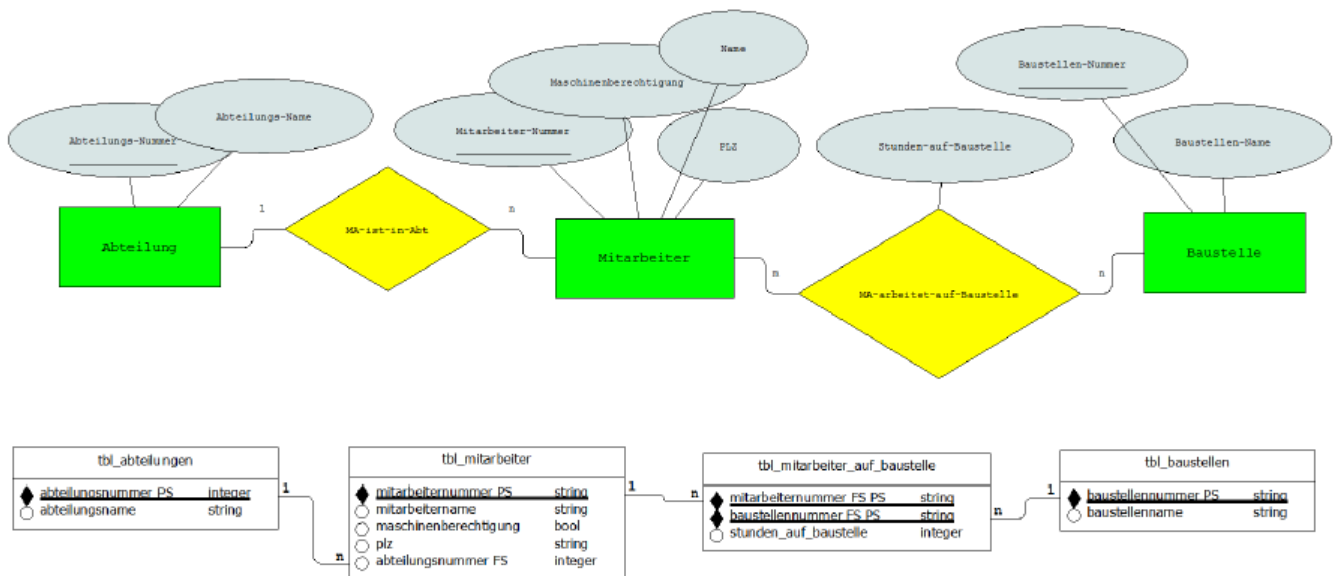


tbl_lehrer	
◆ <u>l_nr</u>	int
○ l_name	string
○ l_vorname	string

tbl_lehrt_in	
◆ <u>l_nr</u>	int
◆ k_bezeichnung	string

tbl_klassen	
◆ <u>k_bezeichnung</u>	string
○ k_stufe	string
○ l_nr	int

tbl_schueler	
◆ <u>s_nummer</u>	int
○ s_name	string
○ s_gDatum	date
○ k_bezeichnung	string



3 Zusammenfassung aller Handlungsschritte bis zur fertigen Datenbank

- (1) Aufgabenstellung analysieren: Entitäten (mit Eigenschaften = Attributen) und deren Beziehungen untereinander bestimmen
- (2) Entitäten mit Attributen in ERM eintragen, 1 Attribut muss Primärschlüssel werden
- (3) Beziehungen zwischen Entitäten eintragen, Kardinalitäten 1:n bzw m:n bestimmen und eintragen
- (4) Relationen-Modell aus dem ERM entwickeln
 - (1) jede Entität wird eine Tabelle
 - (2) jede 1:n bzw 1:m Beziehung in eine Fremdschlüsselbeziehung umwandeln
 - (3) jede m:n Beziehung in eine Beziehungstabelle mit den 2 entsprechenden Fremdschlüsselbeziehungen umwandeln
- (5) praktische Umsetzung im DBMS
 - (1) neue Datenbank erstellen
 - (2) die Tabellen nacheinander erstellen, zuerst die ohne Fremdschlüsselbeziehung (von außen nach innen) mit allen Attributen = Spalten, Primärschlüssel festlegen
 - (3) Datenbank-Diagramm erstellen lassen
 - (4) mit Drag&Drop die Beziehungen zwischen den Tabellen erstellen
 - (6) Daten in Datenbank eintragen ACHTUNG: Erst die Daten in die Tabellen OHNE Fremdschlüsselbeziehung eintragen (von außen nach innen)

(7) Datenbank ist fertig

Grundlagen

2.1 Was ist Normalisierung?



Arbeitsschritte der Datenbank-Modellierung:

1. Szenario/ Pflichtenheft/ Lastenheft
2. Informationsanalyse/ Anforderungsanalyse
3. konzeptioneller Entwurf → ERM
4. logischer Entwurf → Relationenmodell (Tabellen)
5. Normalisierung → **Überprüfung auf Einhaltung der Normalformen 1 ... 3**
6. Implementierung in einem realen Datenbanksystem

Was ist Normalisierung?

die sogenannte **Normalisierung** definierte Regeln und Maßnahmen für die (weitgehende) Sicherstellung einer

- a. **redundanzarmen** Speicherung → jeder Fakt ist nur einmal in der Datenbank gespeichert
- b. **Integritäts-/ Konsistenzsicherung** → es sind keine widersprüchliche Daten gespeichert
- c. höchstmögliche **Flexibilität** (beim Erfassen, Ändern und Löschen von Daten)
- d. **kurzen** Zugriffszeit

NF 1 – 3 → sind für die Praxis wichtig

► Normalform 1



- Eine Relation = Tabelle befindet sich in 1. Normalform (1NF), wenn alle Attributwerte elementar/ atomar (= unteilbar) sind.

1. Normalform

Eine Relation befindet sich in der ersten Normalform wenn alle Attribute nur atomare Werte beinhalten.

KdNr	Name	GebJahr	Alter	Bankverbindung
1	Huwaldt, Alexander	1965	36	430002345, Kreissparkasse XYZ, BLZ 850340200
2	Riedel, Toralf	1964	35	672009001, SpardaBank ABC, BLZ 850200310

KdNr	Vorname	Name	GebJahr	Alter	Kreditinstitut	BLZ	Konto
1	Alexander	Huwaldt	1965	36	Kreissparkasse XYZ	850340200	430002345
2	Toralf	Riedel	1964	35	SpardaBank ABC	850200310	672009001



► Normalform 2

► (1) in NF 1 befindet und

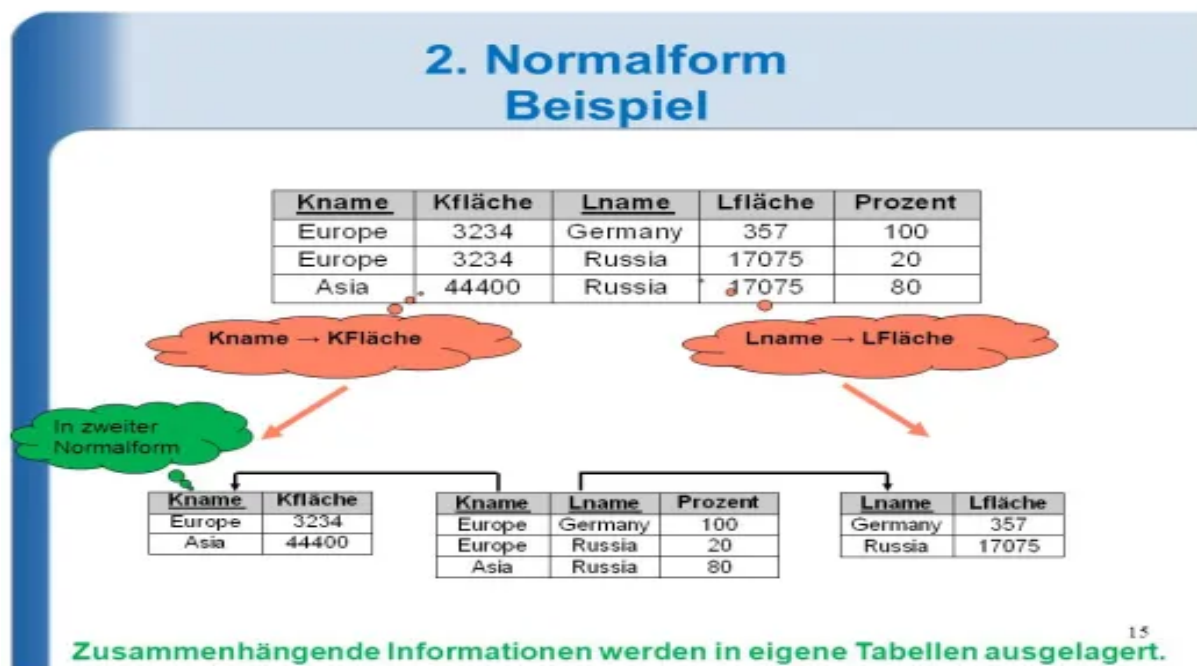
► (2) **jedes Attribut, das nicht Primärschlüsselattribut ist** (wird Nichtschlüsselattribut genannt), **vom gesamten Primärschlüssel** (und nicht nur von Teilen des Primärschlüssels (bei zusammengesetzten Primärschlüsseln)) **voll funktional abhängig** ist.

► Eine Tabelle enthält nur Daten eines Themen- bzw. Informationsbereiches.

► Aufteilung in mehrere Tabellen nach Themen/ Informationsgebieten.

► Verletzungen der NF2 werden dadurch gelöst, dass die Ausgangstabelle in mehrere Tabellen zerlegt wird, in der die zusammengehörigen Informationen gespeichert werden können

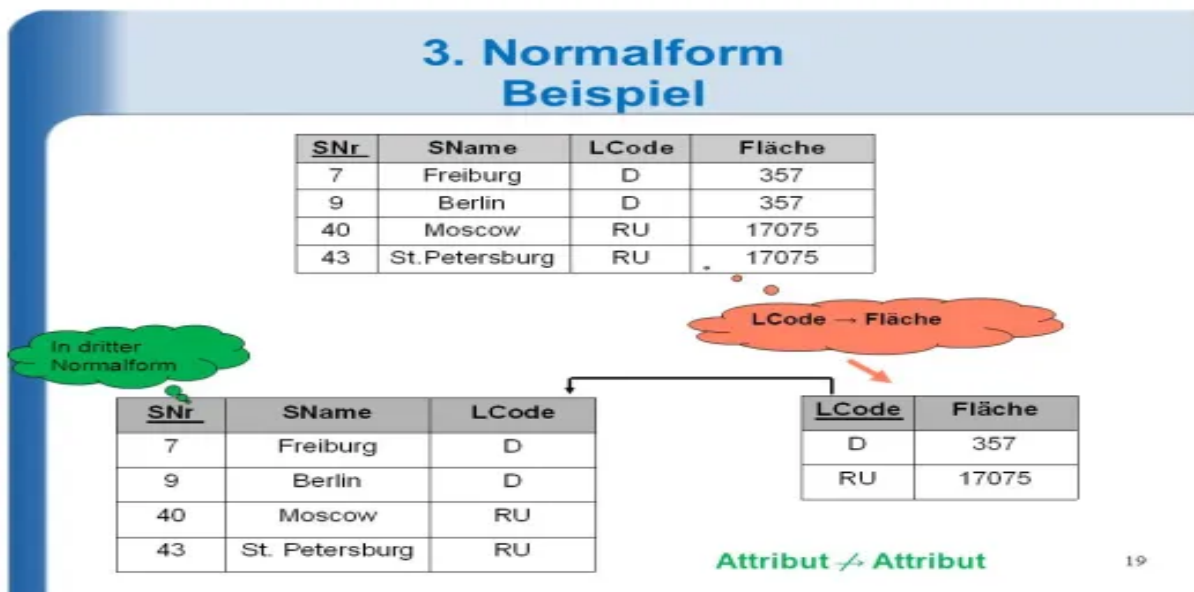
► <https://deristvollfettdertröttl.wordpress.com/erklaerung-1-3-normalform-2/>





► Normalform 3

- Eine Tabelle befindet sich in der dritten Normalform (3NF), wenn sie sich
 - (1) in 2NF befindet und
 - (2) **keine transitiven (indirekten) Abhängigkeiten aufweist** (Erläuterung unten bei Anforderungen)
- Es dürfen also auch keine berechneten Felder verwendet werden (z.B. Geburtsdatum, Alter → Alter berechnet sich aus Geburtsdatum).
- **Felder, die in keiner direkten Abhängigkeit vom Primärschlüssel stehen, müssen in einer eigenen Tabelle gespeichert werden.**
- Bedingung für NF3 nicht erfüllt? → Lösungsmöglichkeit? → vorhandene Tabellen teilen und neue Tabellen einfügen



IHK Aufgaben – IHK da cikmis sorular:

1. 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 Datenbankmodell, 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



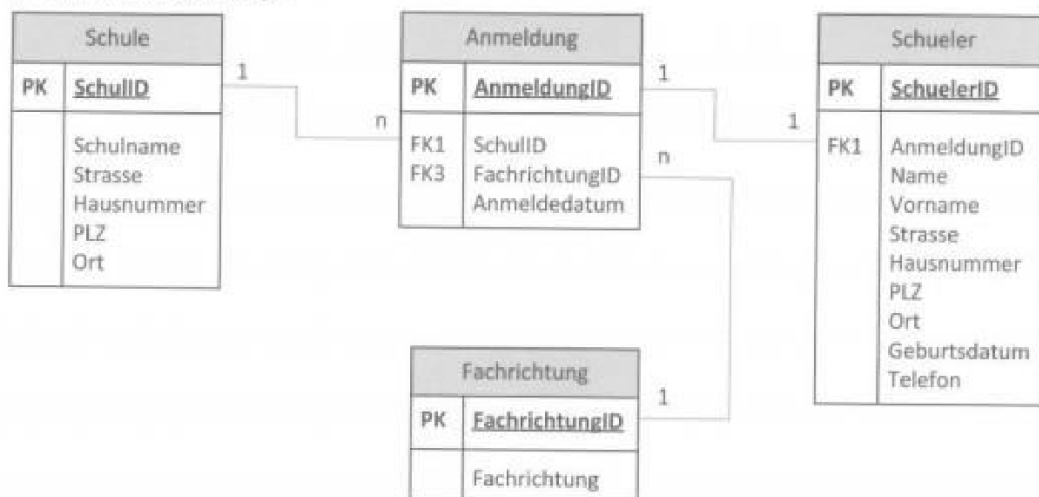
a) 10 Punkte

2 Punkte, 4 x 0,5 Punkte je Primärschlüssel

3 Punkte, 3 x 1 Punkt je Fremdschlüssel

3 Punkte, Kardinalitäten

2 Punkte, Entität Fachrichtungen



2. 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.

Mitarbeiter
Ma_Id
Vorname
Name

MitarbeiterWirtschaftsgut
Lfd_Nr
Wg_Id
Ma_Id
Nutzung_Datum_von
Nutzung_Datum_bis

WirtschaftsgutArt
WgArt_Id
Bezeichnung
Abschreibungs_Dauer_AfA

Lieferant
L_Id
Firma
Strasse
PLZ
Ort

Wirtschaftsgut
Wg_Id
Bezeichnung
WgArt_Id
Datum_Zugang
Anschaffungspreis
Nutzung_Ort
Menge
L_Id

aa) Vervollständigen Sie diesen Entwurf, indem Sie

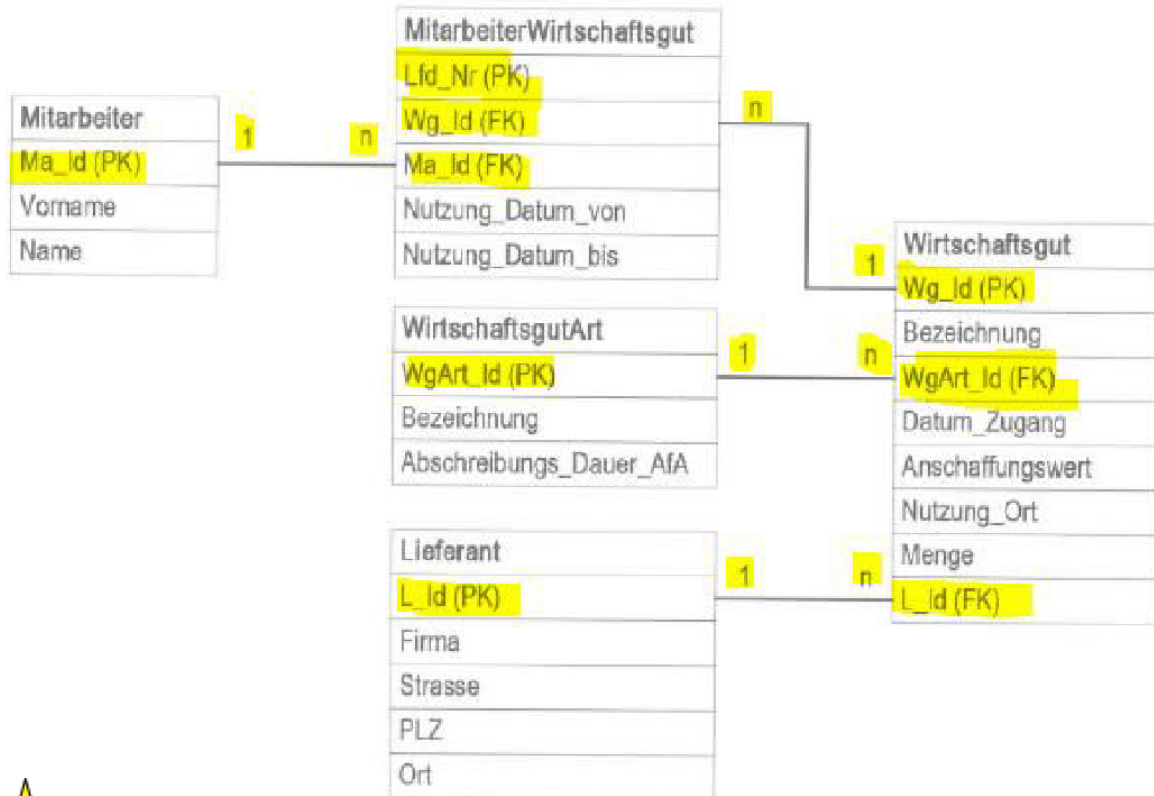
- die Primärschlüssel mit PK und die Fremdschlüssel mit FK kennzeichnen.
- die Beziehungen einzeichnen und deren Kardinalitäten angeben. 15 Punkte

aa) 15 Punkte

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

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

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

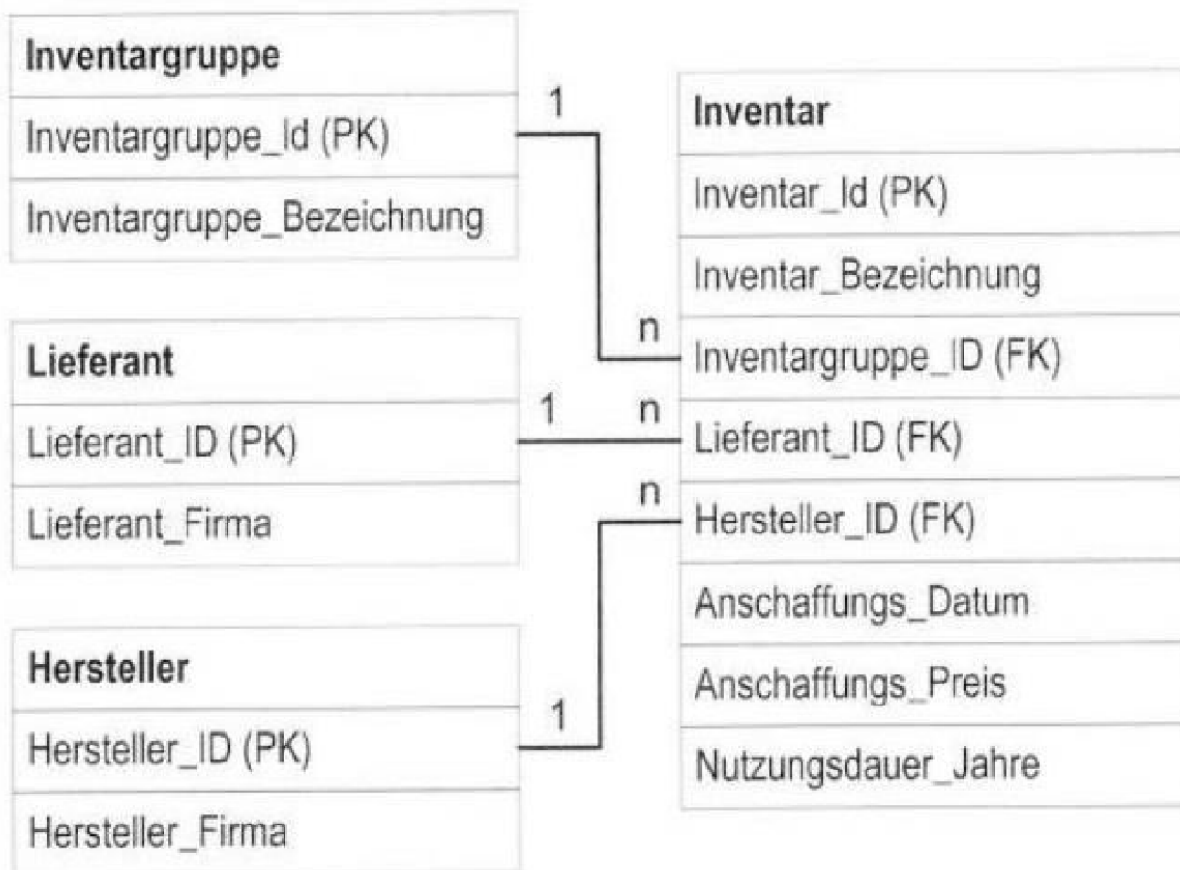


ab) Erläutern Sie die Notwendigkeit der Tabelle *Mitarbeiterwirtschaftsgut*. 1 Punkt
Zwischentabelle zur Auflösung einer m : n - Beziehung in zwei 1 : n - Beziehungen

3. 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

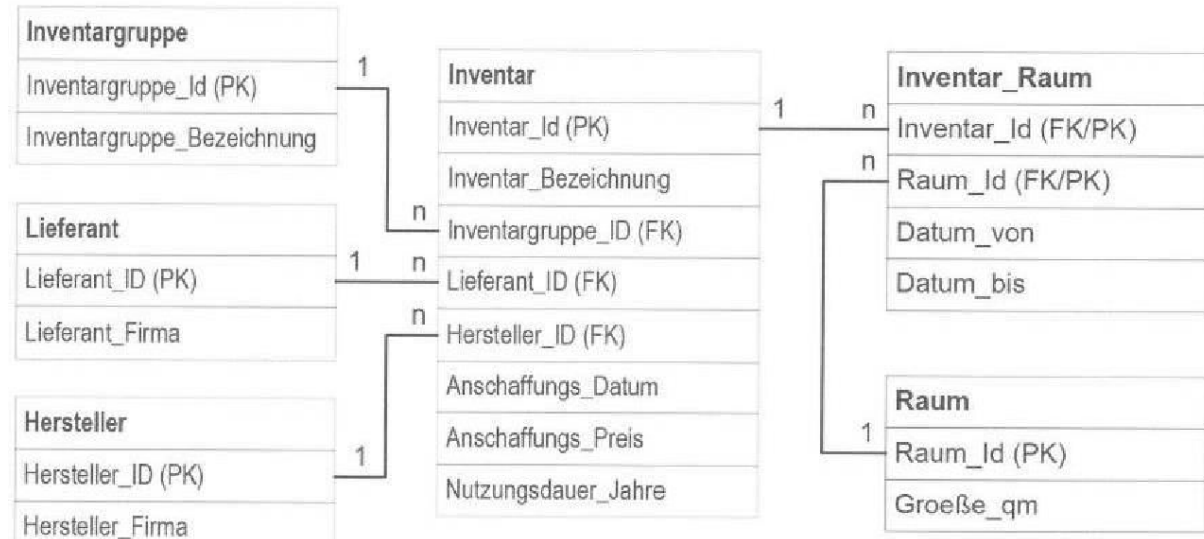
a) 8 Punkte

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



4. 2015SAP IT GA1 Lösungen



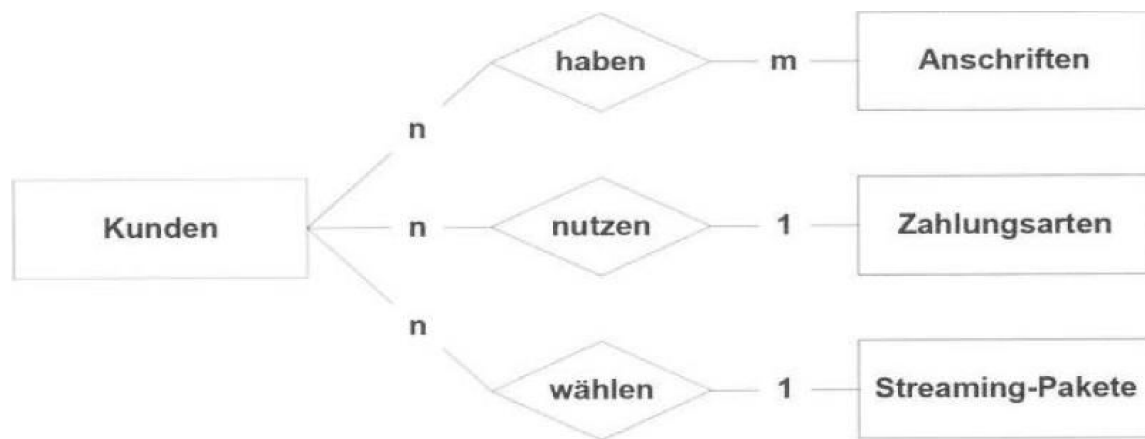
3. Handlungsschritt (25 Punkte)

Die TeNi GmbH bietet ihren Kunden verschiedene Streaming-Pakete 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 ein 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:

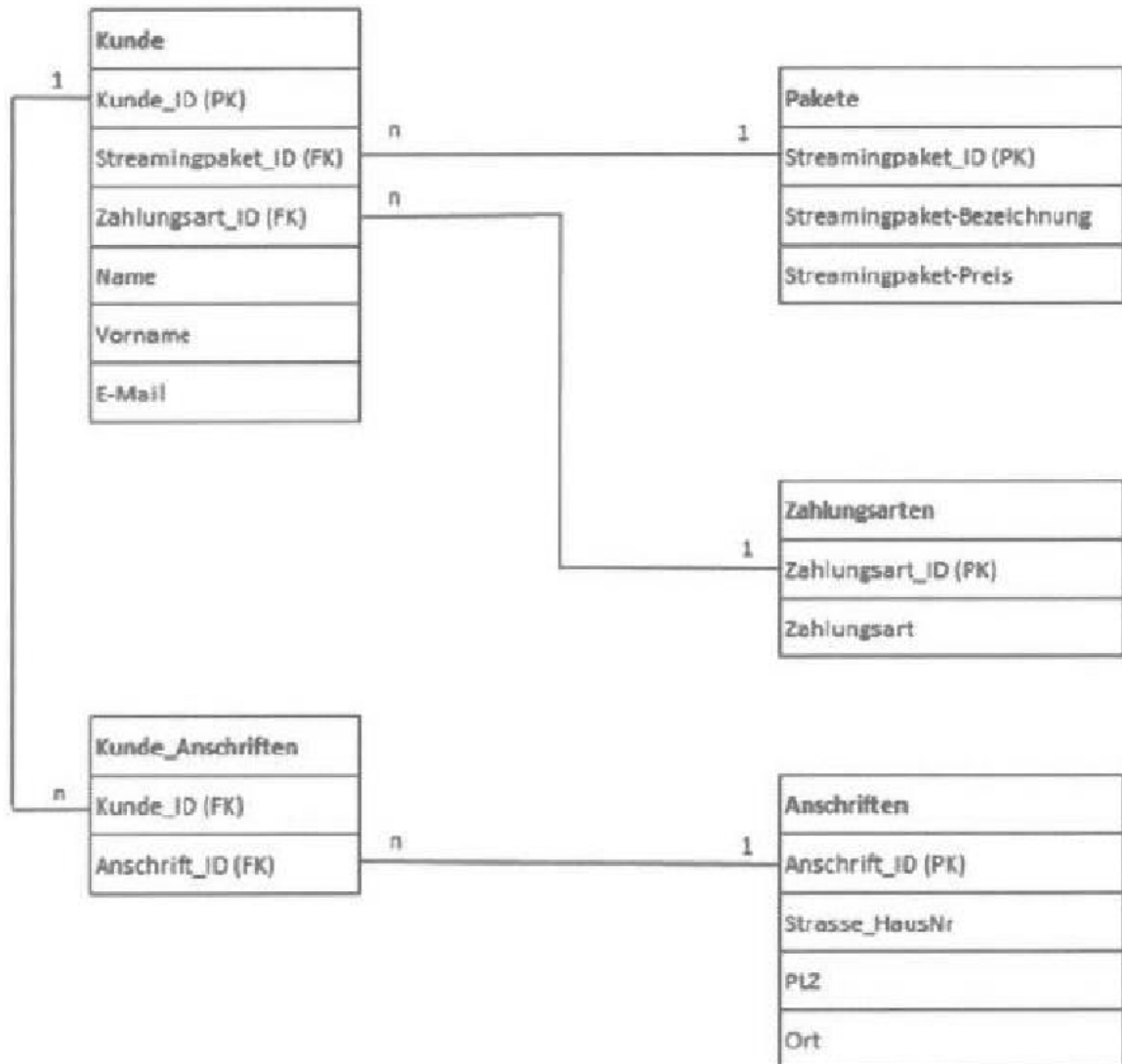
Erstellen Sie alle erforderlichen Tabellen.

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

<Nicht so wichtig!>

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.

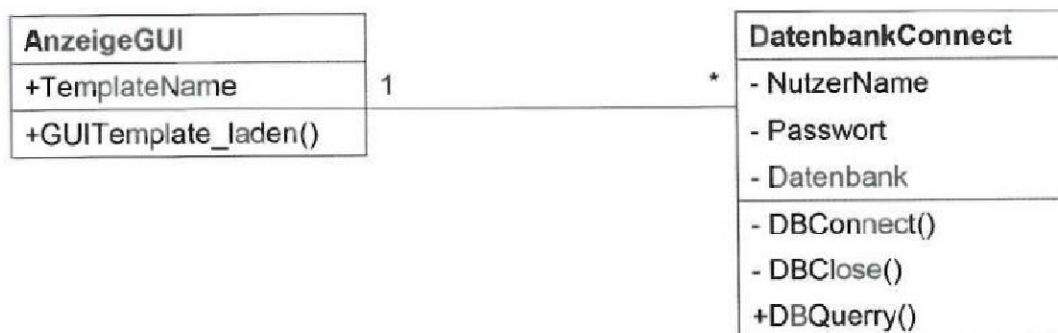
5. 2015W AP IT GA1 Lösungen

4. Handlungsschritt (25 Punkte)

c) Für das Backupsystem der RAIN GmbH soll eine Grafische Anzeige (GUI) entwickelt werden. Es sollen der Archivtyp und die Speicherkapazität sowie das Datum angezeigt werden. Die Daten sollen dann täglich in einer Datenbank gespeichert werden. Die Klassen AnzeigeGUI und DatenbankConnect sind bereits vorhanden.

- Für die Software soll ein Klassendiagramm erweitert werden.
- Erstellen Sie eine Klasse mit dem Namen AbfrageServer.
- Legen Sie eine private Methode mit dem Namen BackupDaten an.
- ★ ■ Legen Sie die privaten Attribute Archivtyp, Speicherkapazität und Datum an.
- Stellen Sie eine Assoziation zwischen den Klassen AbfrageServer und Datenbank her.

8 Punkte



8 Punkte

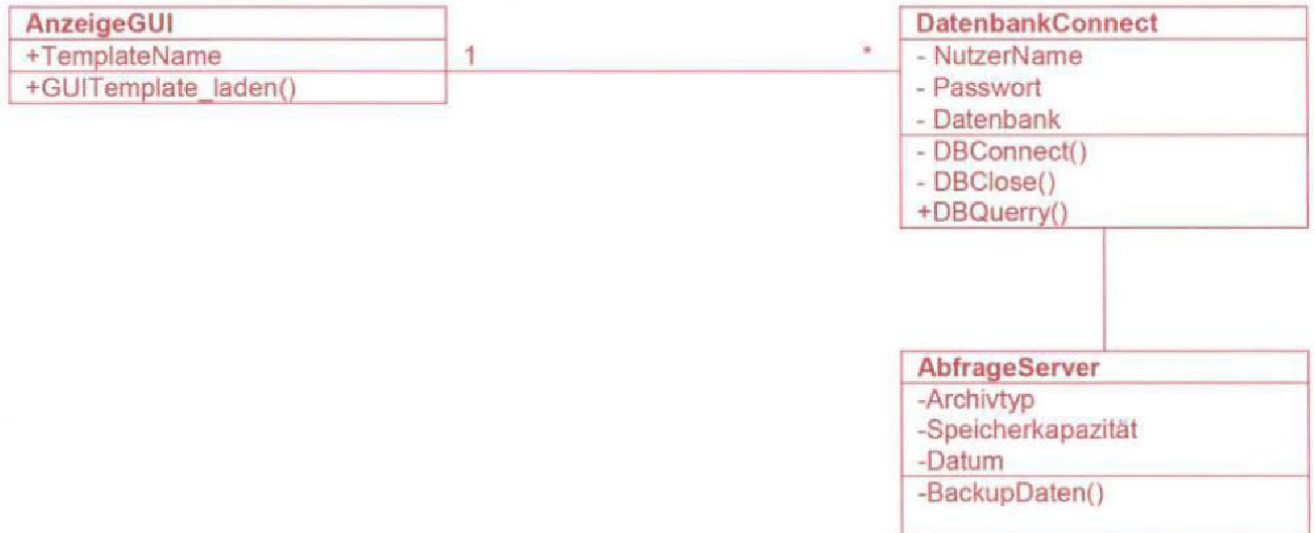
1 Punkt für die Klasse *AbfrageServer*

1 Punkt für die Methode

3 Punkte, 3 x 1 Punkt je Attribut *Archivtyp*, *Speicherkapazität* und *Datum*

1 Punkt für die Assoziation

2 Punkte, 4 x 0,5 Punkte für die Kennzeichnung der Attribute und Methode als „privat“



6. 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.

Artikel
Artikelnummer PK

a) 15 Punkte

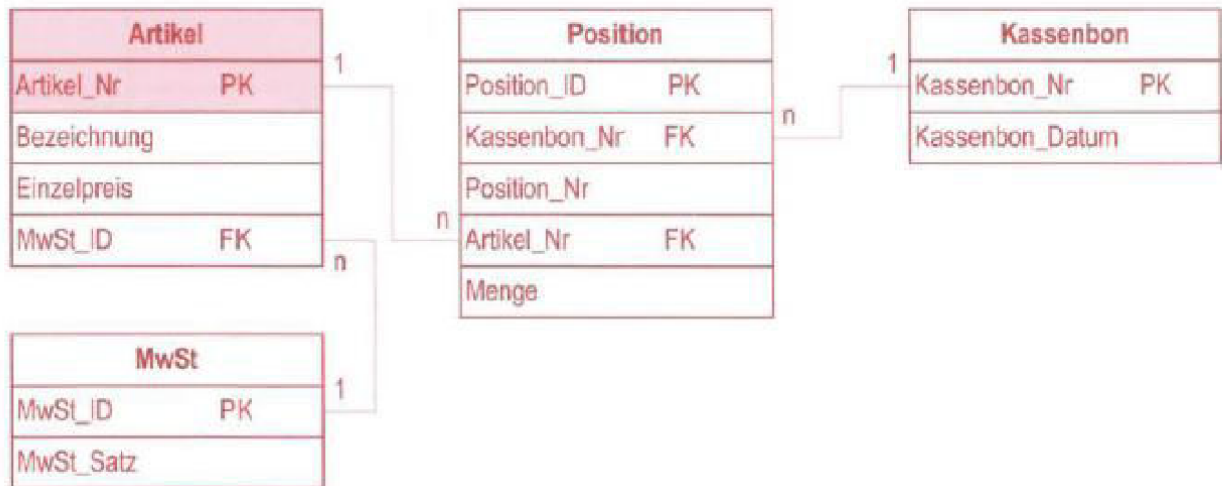
Es gibt verschiedene Lösungen, daher können die folgenden Punktwerte nur Richtwerte sein. Bepunktung für die folgende Lösung:

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.

7. 2016S AP IT GA2 Lösungen

2. Handlungsschritt (25 Punkte)

Die Klübero-IT GmbH soll für die Internet-Warenhaus GmbH eine Datenbank entwickeln.

a) Ein Teil dieser Datenbank ist folgende Tabelle.

Ordnen Sie den folgenden den Attributen sinnvolle Datentypen zu. 6 Punkte

Dokument

Attribut	Beispieldaten	Datentyp
Archivierungs-Nr	2015-270	String
Archivierungs_Datum	02.03.2015	Datum
Dokumentenart_ID	936632897	LongInteger
Aufbewahrungsfrist	10	Integer, Byte, LongInteger
Ablageort	d:\k1\Rechnungen	String
Geheim	true	Boolean



Datentypen

Boolean
Byte
Char
DateTime
Integer
LongInteger
String

Hinweis: Mehrfachnennungen sind möglich.

8. 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.

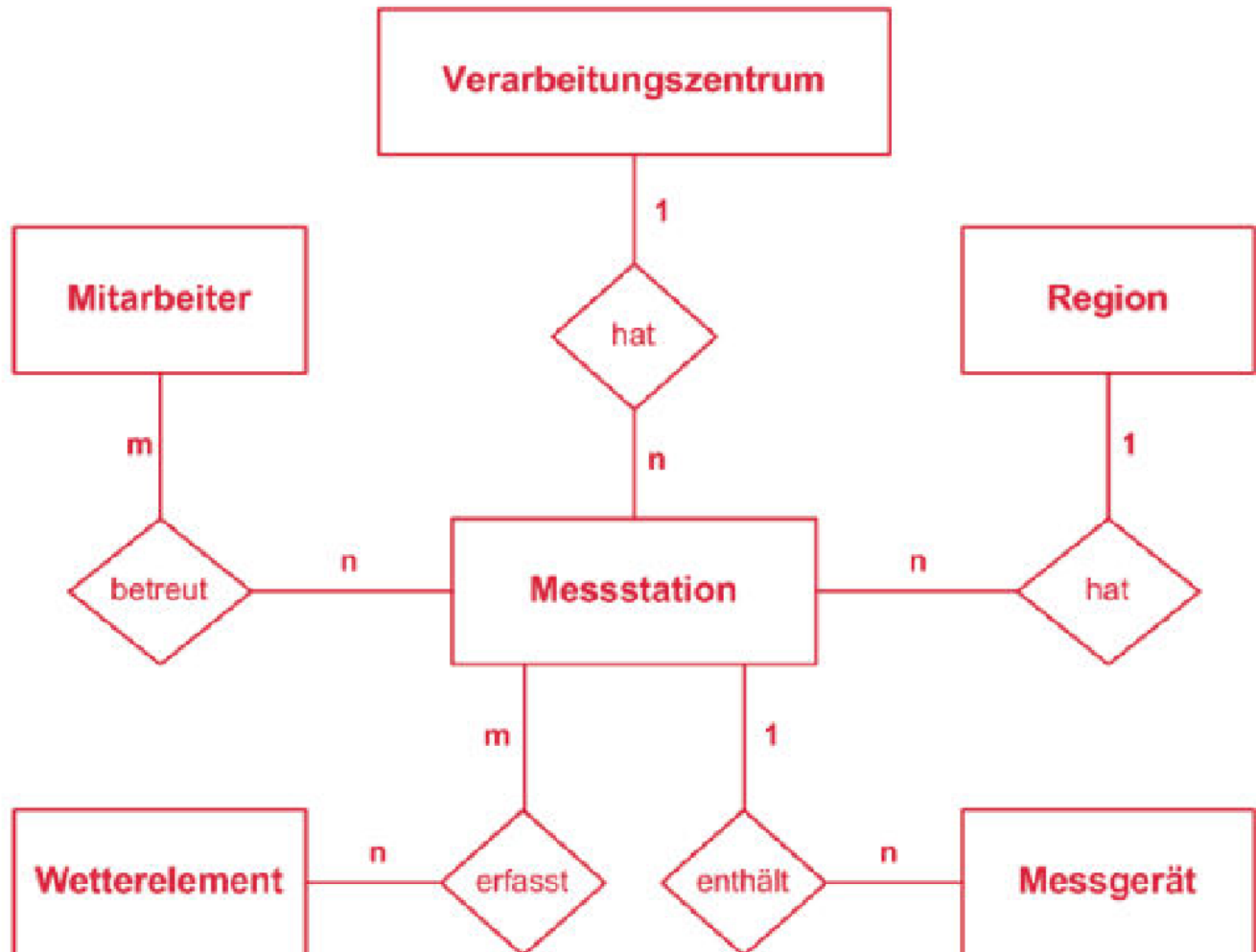
16 Punkte

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



Fortsetzung 4. Handlungsschritt

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

9. 2018W AP IT GA2 Lösungen

4. Handlungsschritt (25 Punkte)

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 Kardinalitäten eintragen.

12 Punkte

Datenmodell für die Datenbank Bestellung



10. 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.

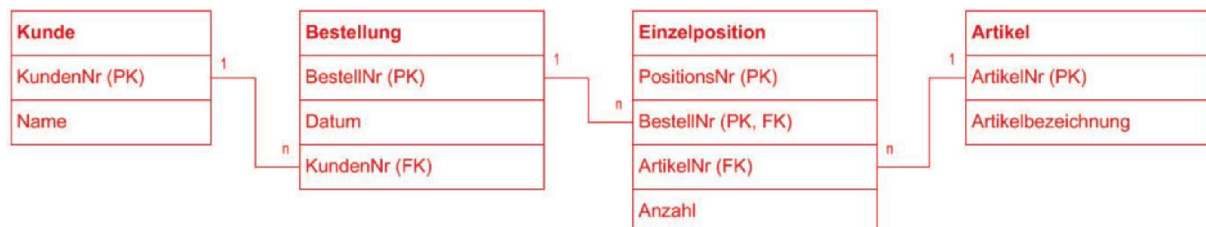
Entwickeln Sie daraus ein relationales Datenmodell, welches der 3. Normalform genügt und ergänzen Sie das folgende Datenbankmodell. 11 Punkte

2 Punkte Tabelle „Kunde“

2 Punkte Tabelle „Artikel“

4 Punkte Tabelle „Einzelpositionen“

3 Punkte (3 x 1 Punkt je Beziehung)



Bestellung
BestellNr (PK)
Datum
KundenNr (FK)

11. 2020W AP IT GA2 Lösungen

4. Handlungsschritt (25 Punkte)

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.

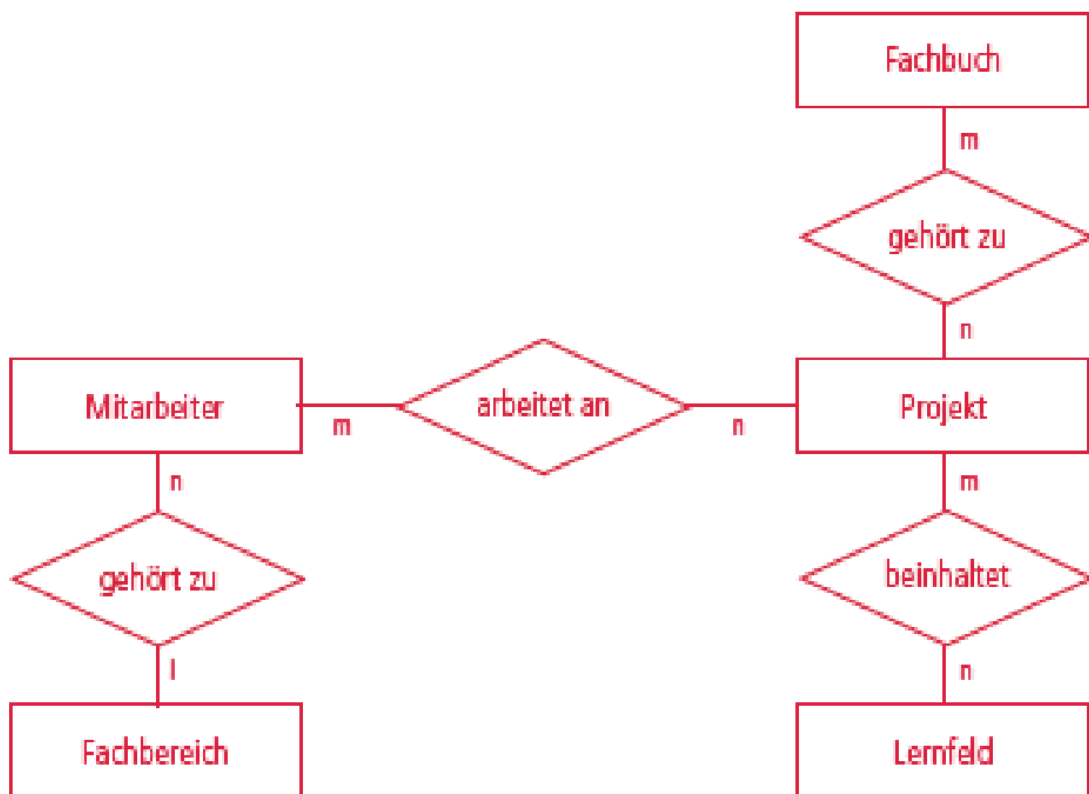
b) 13 Punkte

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.

12. 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.

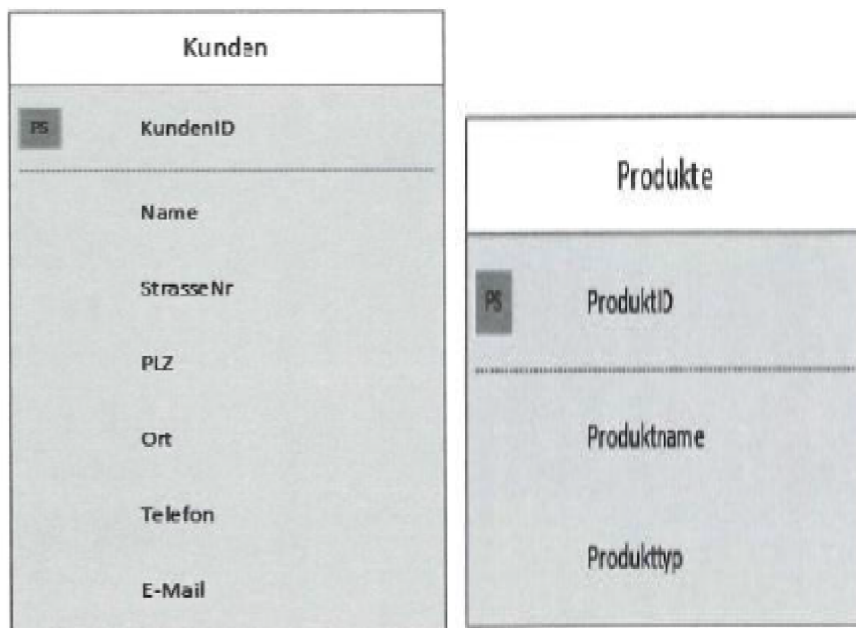
Fortsetzung 5. Handlungsschritt

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.

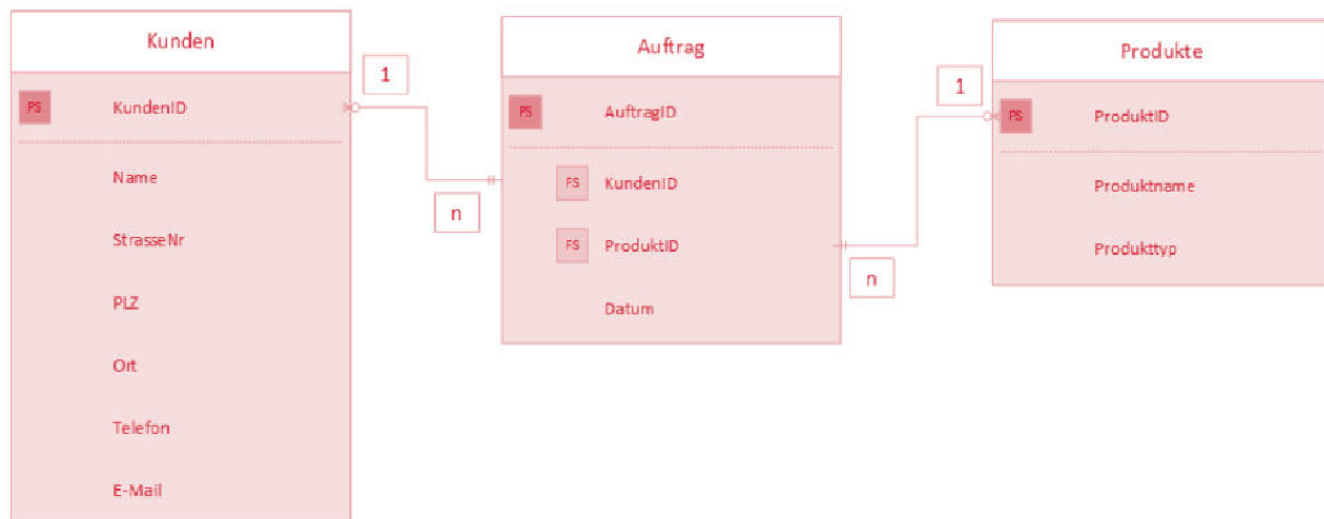
Verschiedene Kunden können die gleichen Produkte bestellen.

- 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



13. 2021W AP IT GA2 Lösungen

5. Handlungsschritt (25 Punkte)

Im Zuge der Umrüstung des mittelständischen Industriebetriebes soll auch die bisherige Datenbank Anwendung 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 protokolliert:

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

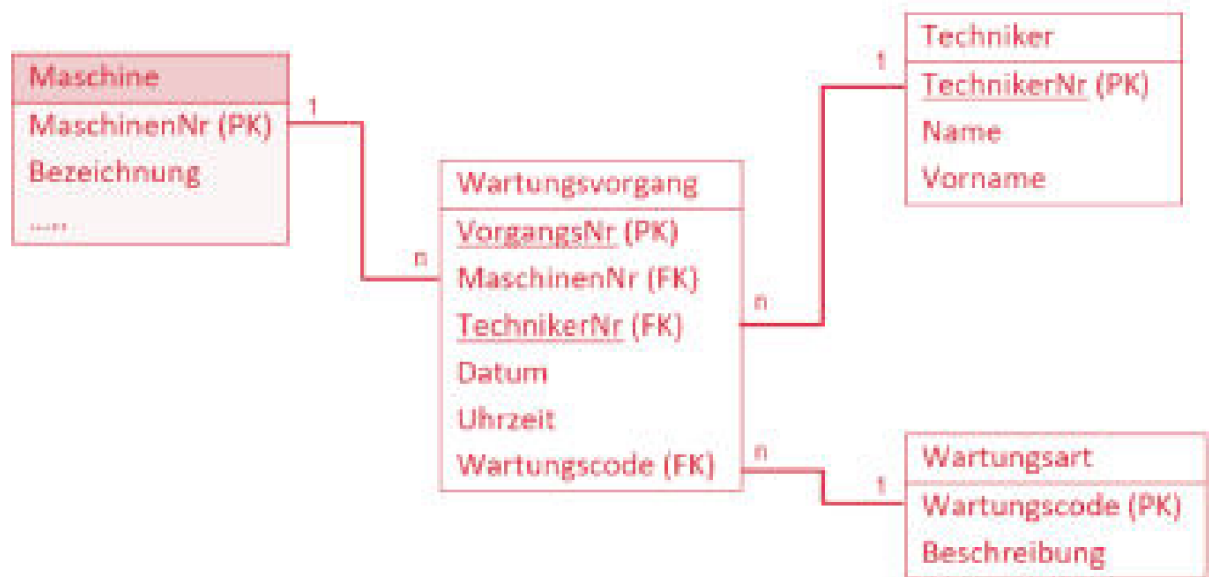
Hinweis: Eine Wartung wird immer von genau einem Techniker durchgeführt.

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

Maschine
MaschinenNr (PK)
Bezeichnung
....



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:

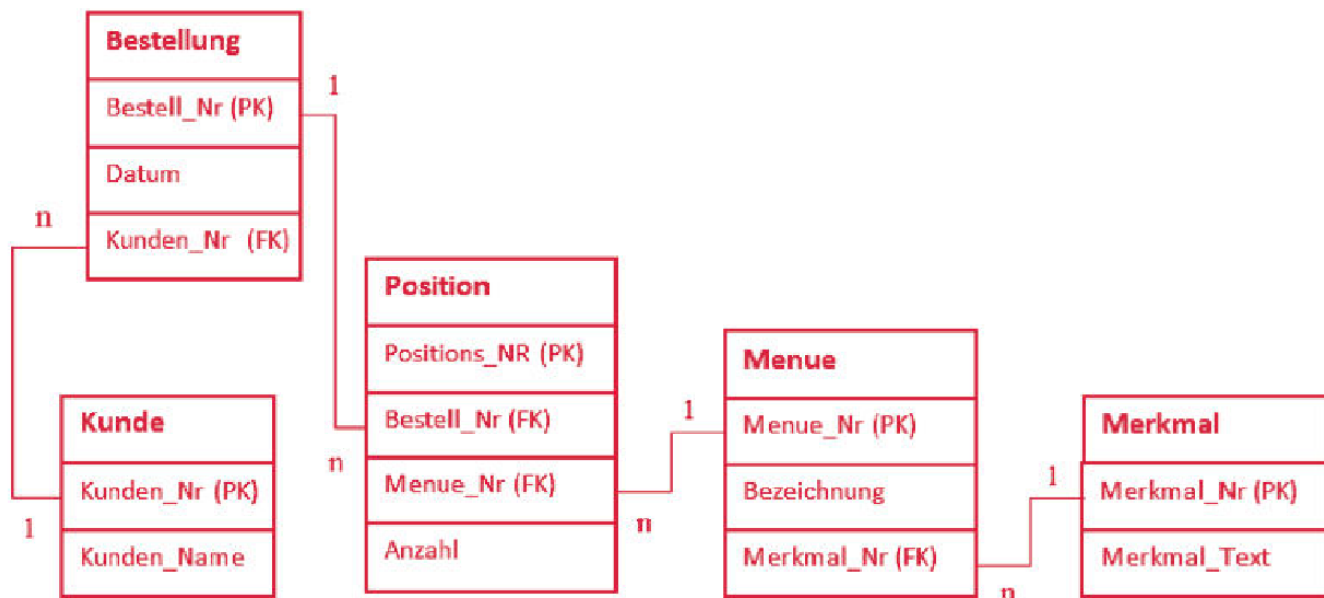
Bestellung	Kunde	Datum	Bestellte Menüs «Anzahl ist neu>>	Merkmal des Menüs
1	Müller - (KundenNr. 23)	23.04.2022	3 x Penne mit Ofengemüse	Vegetarisch
2	Mayer - (KundenNr. 71)	23.04.2022	1 x Tortellini mit Fisch	Glutenfrei
			1 x Steak mit Pfefferrahm auf Kartoffeln	Frei von Weizen
3	Müller - (KundenNr. 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

Bestellung
BestellNr (PK)
Datum
KundenNr (FK)

a) 11 Punkte

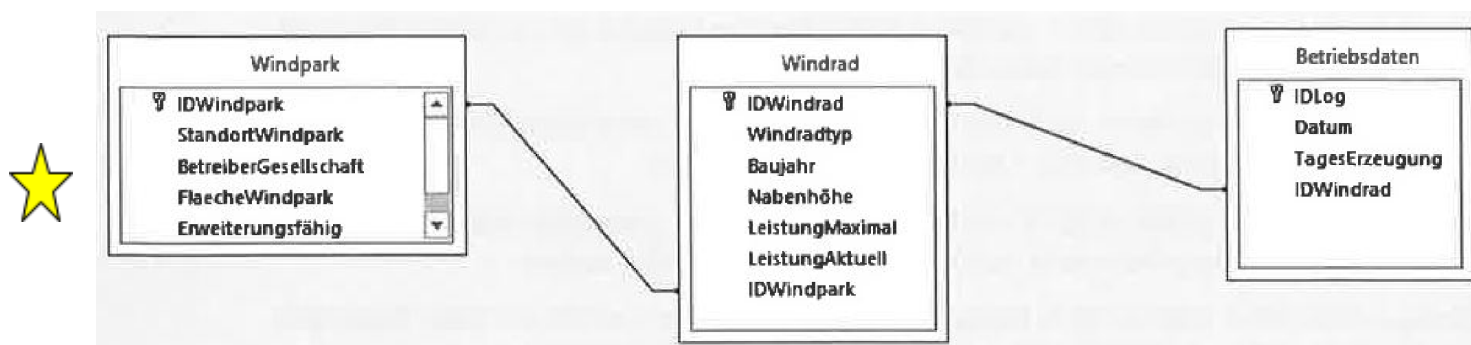
Tabellen Kunde, Merkmal je 1 Punkte (ein PK pro Tabelle wird vorausgesetzt),
Tabelle Position 3 Punkte,
Tabelle Menue 2 Punkte,
je Beziehung 1 Punkt



15. 2022S Konzeption und Administration von IT-Systemen

1. Aufgabe (26 Punkte)

b) Folgender vereinfachter Entwurf der neuen Datenbank liegt vor:



ba) Ergänzen Sie in dem Diagramm zu den beiden Beziehungen die entsprechenden Kardinalitäten. 4 Punkte



Beziehung Windpark zu Windrad 1 : n bzw. 1 : ∞

Beziehung Windrad zu Betriebsdaten 1 : n bzw. 1 : ∞

16. Abschlussprüfung Teil 1 Herbst 2022

5. Handlungsschritt (25 Punkte)

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.

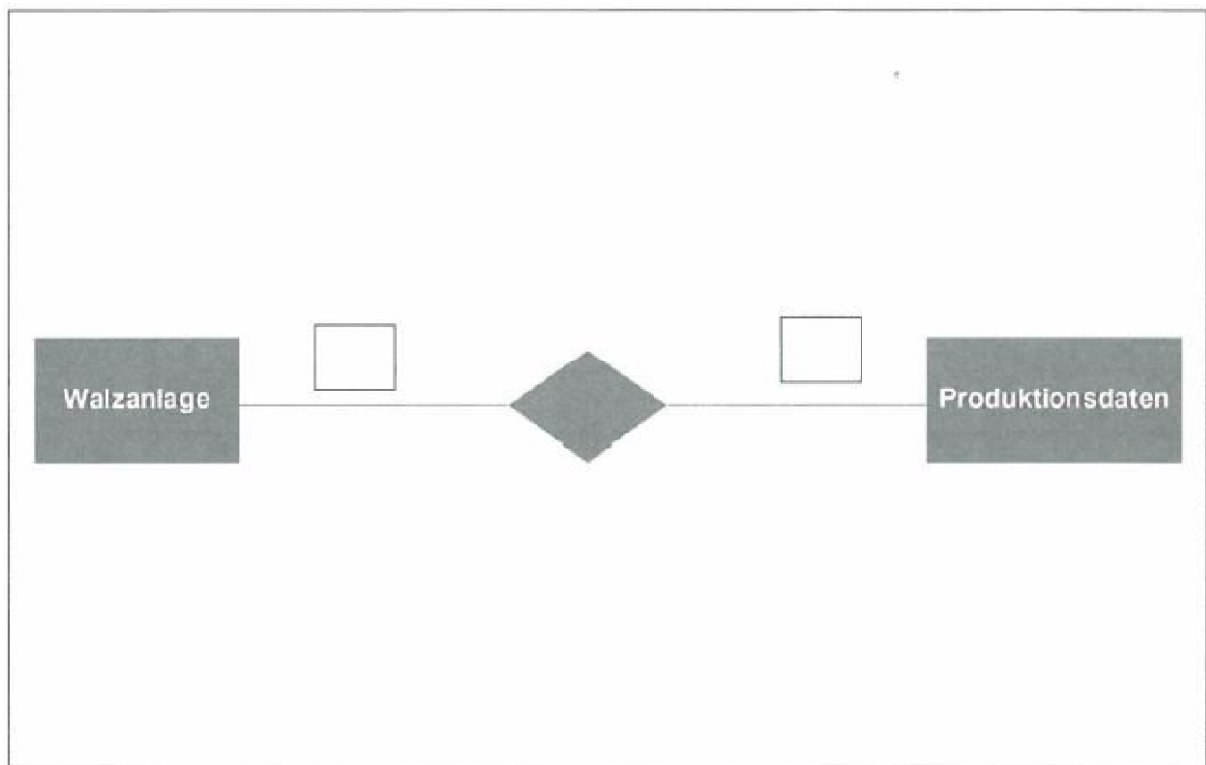
In der Produktionshalle sind mehrere Walzanlagen vorhanden. Diese jeweiligen Walzanlagen können Wellpappen mit unterschiedlichen Dicken (z. B. kleiner 4 mm, 4-8 mm, 8-12 mm) herstellen. In der Datenbank soll gespeichert werden, welche Walzanlage für welche Dicken (Spezifikation) verwendet werden kann. 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	
Attribut	
Primärschlüssel	
Beziehung (Relation, Relationship, Assoziation)	

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

Kardinalität:

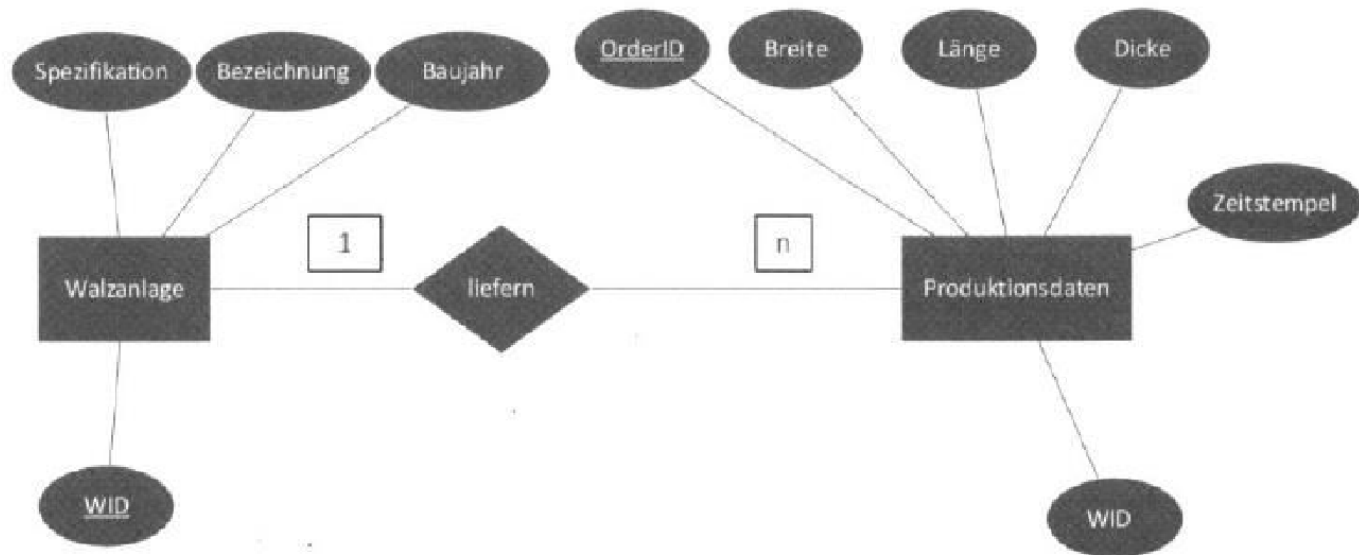
1 Punkt

Kennzeichnung Primärschlüssel in beiden Tabelle

2 Punkte

Attribute:

3 Punkte



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

Kaynaklar:

<https://www.oracle.com/tr/database/what-is-database/>

<https://www.mediaticlick.com.tr/tr/blog/database-nedir>

[WBS Handoute von Thomas Krause](#)

[IHK Abschluss-Prüfungen](#)