



Bedeutung von Daten für Prozesse im Unternehmen.

Aufgaben

- 1 Sammeln Sie Informationen, die im Hinblick auf Prozesse in einem Unternehmen benötigt werden.
- 2 Beurteilen Sie die Einführung eines Systems, mit dem Daten digital bereitgestellt und verarbeitet werden können, hinsichtlich der Prozessziele.

Datenbank- systeme

2

2.1 Konzeptioneller Entwurf einer Datenbank

Zur Abwicklung ihrer Geschäftsprozesse benötigen Unternehmen zahlreiche Informationen. Diese setzen sich aus gespeicherten und abrufbaren Daten über Produkte und Dienstleistungen sowie über Lieferanten, Mitarbeiter und Kunden aber auch über Bestellungen, Produktionspläne, Werkstoffe, Rechnungen, Reklamationen, Lohnnebenkosten und vielem mehr zusammen. Um die Daten nicht nur effektiv, sondern auch effizient zu verwalten, werden üblicherweise Datenbanksysteme verwendet. Die Voraussetzung aller Datenbankanwendungen ist ein konzeptionell einwandfreier Entwurf: dieser muss vollständig und systematisch durchgeführt werden, da mögliche Fehler in späteren Phasen des Datenbankeinsatzes nur mit hohen Kosten oder gar nicht zu korrigieren sind.

Bearbeiter

Das können Sie nach diesem Kapitel:

- Problemstellungen im Hinblick auf die Anforderungen an das Datenbanksystem analysieren.
- für Problemstellungen objektorientierte semantische Modelle entwickeln.

2.1.1 Wo und warum werden Datenbanken eingesetzt?

Ob Sie eine App auf dem Smartphone verwenden, eine Software auf dem Laptop oder eine Anwendung im Internet nutzen: die angezeigten Ergebnisse sehen Sie nur deshalb, weil der jeweiligen Darstellung Daten zugrunde liegen, die häufig in Datenbanken gespeichert sind. Hinter welchen Anwendungen im Alltag findet sich eine Datenbank?

M1 Online-Bestellung im Internet

Mazon.de Alle

Techno Wireless Pro Kopfhörer
 ★★★★★ 73.042 Bewertungen

Unv. Preisempf.: ~~90,00€~~
 Preis: **79,82€**
 & **KOSTENLOSE Rücksendungen**
 Sie sparen: **10,18€ (11%)**
 Preisangaben inkl. USt. Abhängig von der Lieferadresse kann die USt. an der Kasse variieren.
[Weitere Informationen.](#)

Mein Konto

79,82€
 & **KOSTENLOSE Rücksendungen**

GRATIS Lieferung:
Donnerstag, 2. Dez.
 Siehe Details.

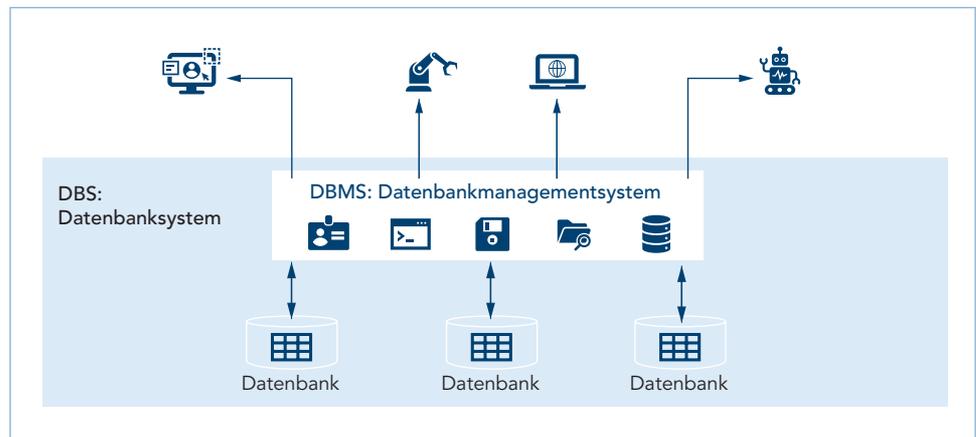
Verfügbar

Menge:

Marke:	Techno	Formfaktor:	Im Ohr
Farbe:	Weiß	Besonderes Merkmal:	Vor Schweiß und Wasser geschützt
Konnektivitätstechnologie:	Kabellos	Enthaltene Komponenten:	Wireless Charging Case
Modellname:	Wireless Pro		

Bearbeiter

M2 Datenbanksystem und Anwendungsumgebung



Bestandteile eines Datenbanksystems

Bearbeiter

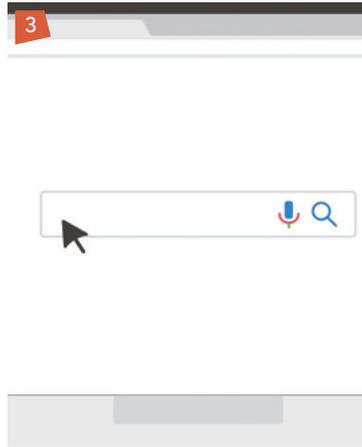
M3 Einsatz von Datenbanken im Alltag



Fahrkartenautomat



Social-Media-Plattformen



Suchmaschinen



Blitzer



Bibliothek



Smartwatch

M4 Möglichkeiten der analogen Datenspeicherung



Datenverwaltung



M5 Digitaler Datenzugriff ohne Datenbankmanagementsystem

Geben Sie einen Administratornutzernamen und ein zugehöriges Kennwort ein, um den Vorgang fortzusetzen.

E-Mail-Adresse

Kennwort

Weitere Optionen

Ja Nein

Zugriffskontrolle

Dokument wird verwendet ? x

Übungsblatt_01_docx ist zur Bearbeitung gesperrt von 'User 1'.

Möchten Sie:

Schreibgeschützte Kopie öffnen

Lokale Kopie erstellen und Änderungen später zusammenführen

Benachrichtigungen, wenn das Original zur Verfügung steht

OK Abbrechen

Mehrbenutzerkoordination

File Explorer interface showing a folder named "Wirtsch... > LP Wln 8.0 Einführu...". The selected file is "Wln_8_AB_Schnittstgellen.docx". The context menu is open, showing options: Öffnen, Bearbeiten, Neu, Drucken, CRC SHA, Teilen..., Mit Transfer senden..., Zum Unterzeichnen versenden, and Versionsverlauf.

Datensicherheit bei der digitalen, umstrukturierten Datenspeicherung
 Bearbeiter

M6 Das passiert im Internet in einer Minute



Quelle: Carl Lewis, JDR.com

Aufgaben

- 1 Beschreiben Sie am Beispiel eines Online-Angebots (M1), wo Verknüpfungen mit Datenbanken erforderlich sind.
- 2 Vollziehen Sie die Bestandteile eines Datenbanksystems (M2) bei einem Online-Bestellvorgang (M1) nach.
- 3 Zeigen Sie mögliche Zugriffe auf Datenbanken anhand der Bilder in M3 auf.
- 4 Recherchieren Sie verschiedene Webseiten und zeigen Sie mögliche Datenbankverknüpfungen auf.
- 5 Leiten Sie aus M4 und M5 Probleme im Hinblick auf die Verwaltung, Speicherung und Nutzung von Daten ab.
- 6 Erstellen Sie unter Berücksichtigung der Probleme (M4, M5) eine Auflistung mit Anforderungen an eine Datenverwaltung.
- 7 Begründen Sie die Notwendigkeit des Einsatzes von Datenbanksystemen bei den Internetanwendungen in M6.

2.1.2 Wie werden Probleme identifiziert, die mithilfe einer Datenbank gelöst werden können?

Nicht alle privaten und unternehmerischen Herausforderungen können mithilfe von Datenbanksystemen bewältigt werden. Doch wie kann man herausfinden, welche Probleme mit entsprechenden datenbankgestützten Anwendungen gelöst werden können?

M1 Schwarzes Brett für Nachhilfeangebote und -gesuche in der Schule



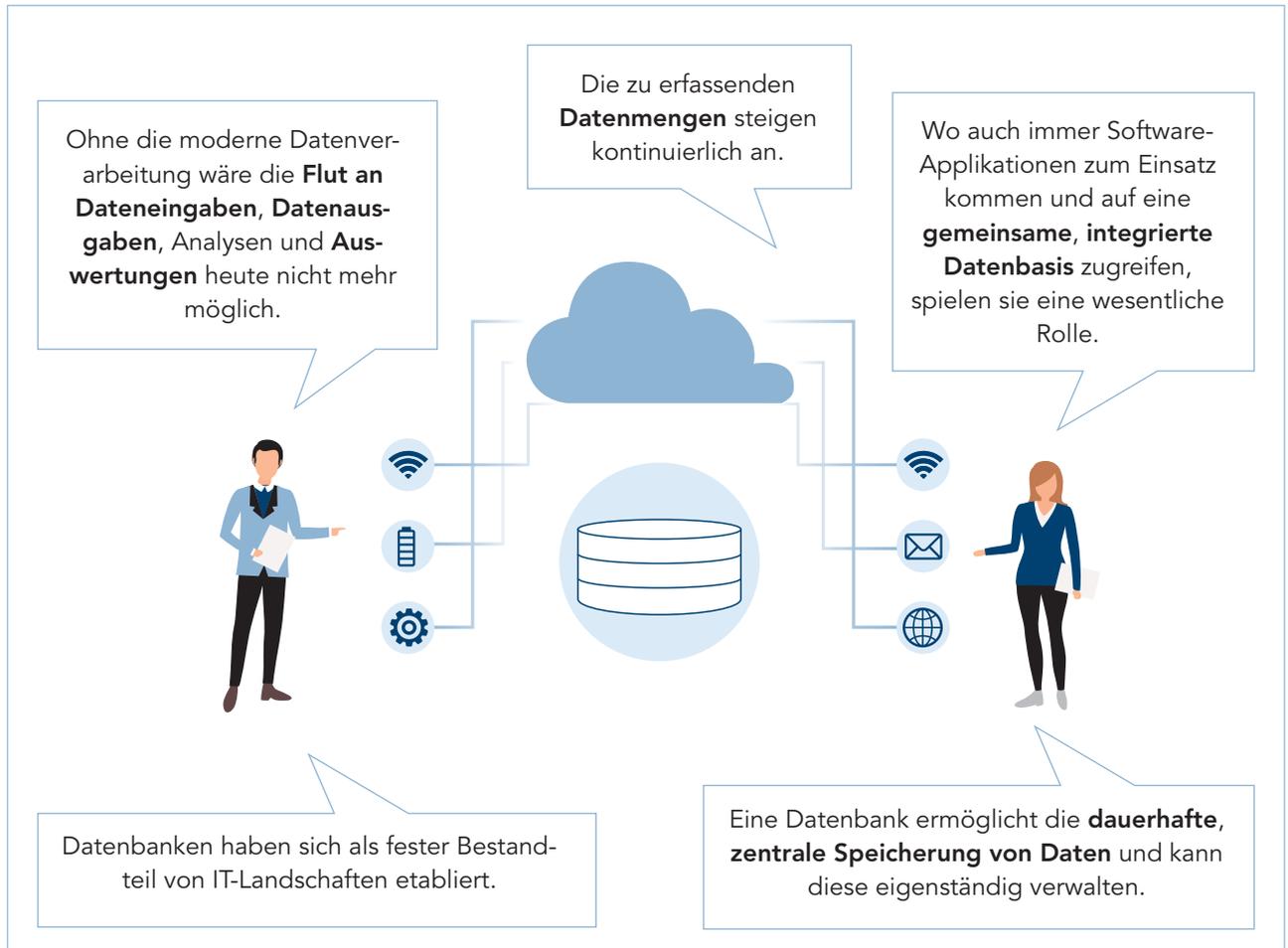
M2 Ein Nachhilfeprogramm in der Schule

„Schüler helfen Schülern“ – unter diesem Motto führt eine Lehrkraft an einem bayerischen Gymnasium Listen zur Vermittlung von Nachhilfe. Gleichzeitig hängen die Schülerinnen und Schüler ihre Nachhilfeangebote und -gesuche an einem Schwarzen Brett in der Pausenhalle aus. Obwohl die gesamte Schulfamilie das Projekt unterstützt, sind die Beteiligten oft von der Zettelwirtschaft genervt, da die Pinnwand sowohl der Schülerschaft mit Förderungsbedarf als auch den Lerntutorinnen und Lerntutoren zugänglich ist und die Zettel kreuz und quer hängen. Unter der gesamten Schüler-

schaft kann es natürlich auch Schülerinnen und Schüler geben, die aufgrund ihrer Fähigkeiten und Fertigkeiten Nachhilfe in einem Fach geben können, während sie aber gleichzeitig Lernunterstützung in einem anderen Fach benötigen. Dies wird von der betreuenden Lehrkraft mit zwei verschiedenen Listen erfasst. Außerdem müssen das Datum einer gehaltenen Nachhilfestunde und die jeweilige Schüleranzahl sowie die Namen registriert werden. Grundsätzlich können auch Lerngruppen gebildet werden.

Bearbeiter

M3 Aussagen zum Datenbankeneinsatz



Begerow Beratungsgesellschaft: Datenbanken im Alltag. In: www.datenbanken-verstehen.de, Zugriff am 20.01.2023 und Begerow Beratungsgesellschaft: Einsatz von Datenbanken. In: www.datenbanken-verstehen.de, Abruf am 20.01.2023

Aufgaben

- 1 Zeigen Sie anhand von M1, dass diese Form zur Nachhilfesuche weder effektiv noch effizient ist.
- 2 Beschreiben Sie die Probleme des Nachhilfeprogramms der Schule (M2).
- 3 Arbeiten Sie aus den Aussagen in M3 zentrale Kriterien für einen Data-bank-einsatz heraus, wodurch die in M2 beschriebenen Probleme verringert werden könnten.
- 4 Sammeln Sie anhand Ihrer bisherigen Erkenntnisse – auch aus dem vorherigen Kapitel – weitere Alltagssituationen, in denen Datenbanken helfen könnten.

Wie wird eine Datenbank entworfen?

Datenbanken dienen als Informationssysteme zur Überwachung der unternehmensweiten Performance sowie zur Verbesserung und Beschleunigung der Entscheidungsfindung. Zum Aufbau einer computergestützten Datenbank ist die Modellierung von Sachverhalten der realen (Unternehmens-) Welt mithilfe weniger Grundkonzepte notwendig. Die konzeptionelle Ebene hilft also dabei, den Ausschnitt aus der Realität zu strukturieren. Auf dieser Ebene sollte nur die Anwendersicht modelliert werden.



Phase 1: Anwendungsbereich abgrenzen (Kap. 2.1.3)

Im ersten Schritt geht es darum, den zu modellierenden Realitätsausschnitt abzugrenzen und zu klären, welche Aufgaben die Datenbank erfüllen soll. Dazu wird ein Problem oder Sachverhalt in Teilbereiche zerlegt (Partitionierung):

- Identifikation relevanter Organisationseinheiten
- Identifikation der durch die Datenbank zu unterstützenden Aufgaben
- Festhalten von Anwendungsfällen (Use Case) und Benutzerschnittstellen



Phase 2: Anforderungen an Informationen strukturieren (Kap. 2.1.4 bis 2.1.6)

Die Anforderungsanalyse berücksichtigt die Informationsanforderungen der zu modellierenden Welt – hier wird das „was“ festgelegt (Projektion), nicht das „wie“:

- Sammlung relevanter einzelner Dinge, Personen oder Begriffe (Objekte mit Attributen und Ausprägungen) aus Anwendersicht
- Beschreibung der Beziehungen zwischen einzelnen Objekten
- Prüfung auf Verständlichkeit und Eindeutigkeit der Spezifikationen



Phase 3: Semantisches Modell entwerfen (Kap. 2.1.7 bis 2.1.8)

Nach der Fertigstellung der Informationsanforderungen erfolgt der konzeptionelle Entwurf, in dem die Informationsstruktur auf einer anwenderorientierten Ebene festgelegt wird (Abstraktion):

- Modellierung des Realitätsausschnitts (Klassen und Kardinalitäten) aus fachlicher Sicht der Anwender
- von der technischen Umsetzung mit einem DBMS noch völlig unabhängig



Phase 4: Modellierten Entwurf verbessern (Kap. 2.1.9)

Ein konzeptionell einwandfreier Entwurf ist die Voraussetzung aller Datenbankanwendungen, weshalb Fehler identifiziert und behoben werden müssen:

- Beseitigung von Redundanzen
- Vermeidung von Inkonsistenzen

Bearbeiter

Aufgaben

Modellieren Sie projektbasiert einen Datenbankentwurf. Sie haben die Wahl zwischen der Entwicklung einer Datenbank für das Nachhilfeprojekt „Schüler helfen Schülern“ (vgl. Kapitel 2.1.2, M1 und M2) oder einer eigenen geeigneten Projektidee. Die Phasen der Methode Datenbankentwurf werden in den Kapiteln 2.1.3 bis 2.1.9 konkretisiert, indem Ihnen Materialien und dazu passende Aufgaben bereitgestellt werden. Die allgemeinen Aufgaben und die Projektaufgaben sind für jedes Kapitel in etwa im Verhältnis 1:2 zu bearbeiten, d. h. es ist mehr Zeit für den Datenbankentwurf vorgesehen.

- 1 Lesen Sie die Methodenseiten und bilden Sie Gruppen sinnvoller Größe.
- 2 Wählen Sie eine geeignete Projektidee aus.
- 3 Definieren Sie Ziele für ihr Datenbankprojekt:
 - a) Bis zum [Datum] soll ein Datenbankentwurf zur Unterstützung des Projekts [Name] fachlich fehlerfrei konzipiert worden sein.
 - b) Der Datenbankentwurf sowie eine kritische Reflexion werden am [Datum] in einer Präsentation (ca. fünf bis zehn Minuten) der eigenen Klasse vorgestellt.
- 4 Bearbeiten Sie die folgenden Kapitel schrittweise:
 - a)  Vollziehen Sie die Projektzieldefinition nach.
 - b)  Bearbeiten Sie die allgemeinen materialbezogenen Aufgaben.
 - c)  Wenden Sie Ihr so neuerworbenes Wissen und ihre Kompetenzen auf Ihr Datenbankprojekt an.

2.1.3 Welche Aufgaben soll die Datenbank für wen erfüllen?

Nachdem für eine Alltagssituation die Notwendigkeit einer Datenbank erkannt wurde, gilt es, die relevanten Akteure sowie die durch die Datenbank zu unterstützenden Aufgaben zu identifizieren. Von wem soll also die Datenbank zu welchem Zweck verwendet werden?

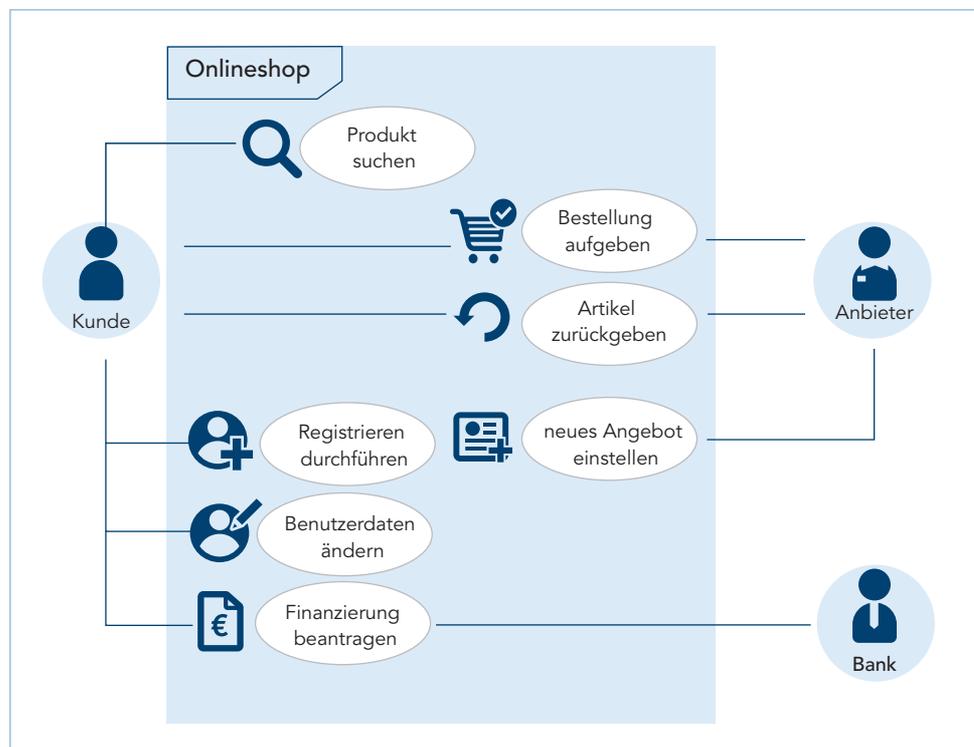
M1 Wie fängt man an?

Es empfiehlt sich, den Zweck der Datenbank auf Papier zu schreiben: Zweck, erwartete Verwendung und geplante Benutzer. Für eine kleine Datenbank für ein Heimbüro könnten Sie etwas Einfaches schreiben wie „Die Kundendatenbank enthält eine Liste der Kundeninformationen zum Erstellen von Postsendungen und Berichten.“ Wenn die Datenbank komplexer ist oder von vielen Personen

verwendet wird, wie dies oft in Unternehmen der Fall ist, kann der Zweck schnell [eine Seite] [...] oder mehr umfassen, und er sollte enthalten, wann und wie jede Person die Datenbank verwenden wird. Die Idee ist, eine gut formulierte Aufgabenbeschreibung zu haben [...] [.]

©Microsoft: Grundlagen des Datenbankentwurfs.
In: www.support-microsoft.com, Abruf am 20.01.2023

M2 Mögliche Anwendungsfälle von Akteuren eines Onlineshops



Use-Case-Diagramm eines Onlineshops

Bearbeiter

M3 Use-Case

In einem Use Case – auch Anwendungsfall genannt – wird das [...] Verhalten eines Systems aus Sicht der Nutzer beschrieben. Ein Nutzer kann hierbei eine Person, eine Rolle oder ein anderes Sys-

tem sein. Dieser Nutzer tritt als Akteur mit dem System in Interaktion, um ein bestimmtes Ziel zu erreichen.

*microTOOL: Use Cases. Das System im Überblick.
In: www.microtool.de, Abruf am 20.01.2023*

M4 Von der Notwendigkeit gegenseitigen Verständnisses



Wie die Anforderungen vom Anwender beschrieben wurden.



Wie es der Entwickler verstanden hat.



Wie die Anforderungsanalyse erfolgte.



Was modelliert wurde.



Was der Anwender eigentlich brauchte.

Anforderungsanalyse

©Paragon Innovations, Bearbeiter



Projektzieldefinition

Sie grenzen den zu modellierenden Realitätsausschnitt ab, indem Sie klären, welche Aufgaben die Datenbank erfüllen soll. Dazu entwerfen Sie ein Use-Case-Diagramm.



Aufgaben zum Material

- 1 Beschreiben Sie für das Beispiel eines Onlineshops, welchen Zweck die Datenbank erfüllen soll (M1).
- 2 Gleichen Sie ihre Ergebnisse aus Aufgabe 1 mit M2 und M3 ab und diskutieren Sie Unterschiede.
- 3 Erläutern Sie Herausforderungen beim Entwurf von Datenbanken anhand von M4.



Aufgaben zum Projekt

- 4 Führen Sie ein Brainstorming zu Anforderungen ihres Projekts durch (M1).
- 5 Entwerfen Sie zu ihrem Projekt ein Use-Case-Diagramm (M2, M3).

2.1.4 Wie kann die komplexe Realität vereinfacht werden?

Anhand der erhobenen Anwendungsfälle kann mit der Planung des Datenbanksystems begonnen werden. Im nächsten Schritt müssen die dafür notwendigen Daten identifiziert werden. Welche Daten sollen also in der Datenbank gespeichert werden?

M1 Objektorientierte Modellierung

Aufbau einer Objektkarte:

Jedes Objekt erhält einen *Objektbezeichner*, also einen eindeutigen Namen.

Außerdem werden alle relevanten Eigenschaften/Attribute aufgelistet. Dazu werden jeweils der *Attributbezeichner* und der *Attributwert* notiert. Jede Objektkarte folgt somit einem Bauplan, der für alle gleichartigen Objekte identisch ist. Den Namen dieses Bauplans nennt man *Klasse*.



Autoquartett und Objektkarte

Bearbeiter, nach: www.alternativ-mobil.info

M2 Eine Person – unterschiedliche Perspektiven



Eine Person aus unterschiedlichen Perspektiven

Projektzieldefinition

Sie vereinfachen das Anwendungsszenario Ihres Datenbank-projekts, indem Sie mithilfe der objektorientierten Modellierung Daten erfassen, die in Ihrer Datenbank gespeichert werden sollen.

Aufgaben zum Material

- 1 Vollziehen Sie den Aufbau einer Objektkarte nach, indem Sie der Objektkarte in M1 die Fachbegriffe (vgl. Randspalte) begründet zuordnen.
- 2 Erstellen Sie vier Objektkarten ein und derselben Person. Berücksichtigen Sie bei der Auswahl der Attribute die jeweils verschiedenen Perspektiven als Kunde, Patient, Mitglied eines Sozialen Netzwerks und Kfz-Versicherungsnehmer (M2).

Aufgaben zum Projekt

- 3 Identifizieren Sie die für Ihr Datenbankprojekt benötigten Baupläne für Objekte.
- 4 Erstellen Sie aus den Ergebnissen aus Aufgabe 3 beispielhaft Objektkarten mit konkreten Attributwerten. Nutzen Sie ggf. ein digitales Diagramm- und Zeichenprogramm.

2.1.5 Wie werden Zusammenhänge zwischen Objekten dargestellt?

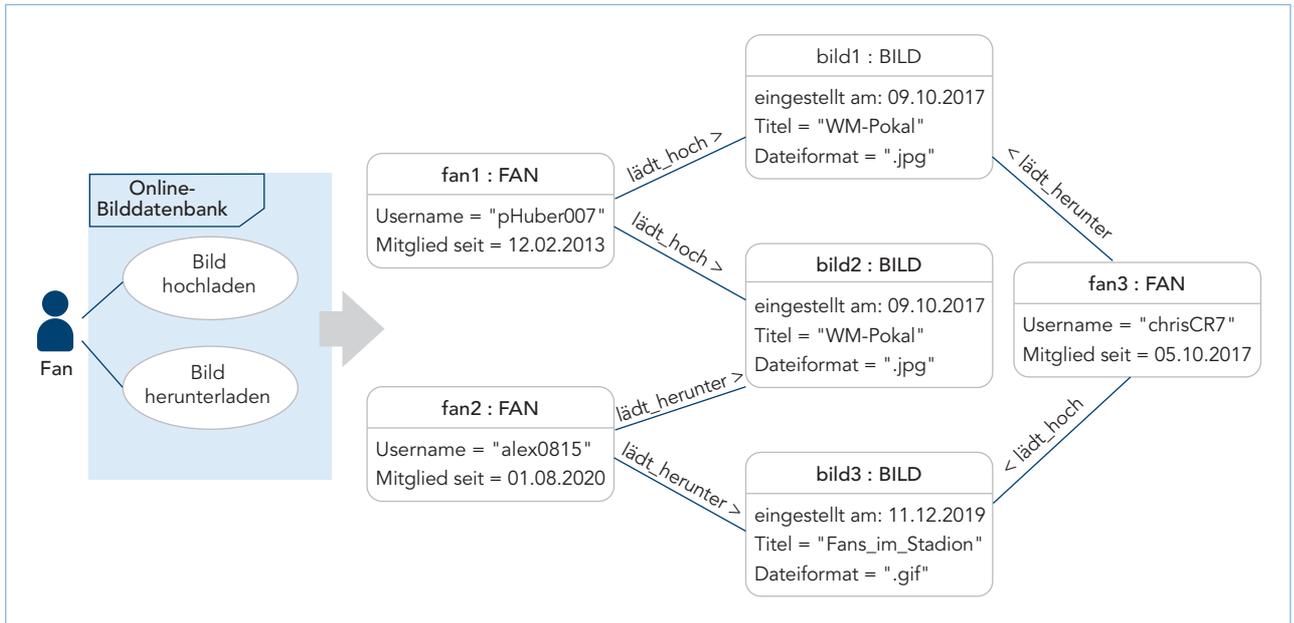
Objekte allein reichen zur Planung eines Datenbanksystems nicht aus. Manche Daten, die gespeichert werden sollen, ergeben sich erst aus dem Zusammenspiel verschiedener Objekte. Wie kann man also unterschiedliche Zusammenhänge zwischen Objekten in der Planung berücksichtigen?

M1 Akteure rund um ein Fußballspiel



Beispiele für unterschiedliche Objekte

M2 Vom Use-Case-Diagramm zum Objektdiagramm



Use-Case-Diagramm und Objektdiagramm
 Bearbeiter

In umfangreichen Objektdiagrammen wird die Beschriftung der Beziehungen aus Gründen der Übersichtlichkeit häufig weggelassen.



Projektzieldefinition

Sie stellen die Beziehungen zwischen Objekten ihres Datenbanksystems mithilfe eines Objektdiagramms dar und erhalten somit weitere Informationen für die Planung Ihrer Datenbank.



Aufgaben zum Material

- 1 Beschreiben Sie die Beziehungen zwischen den in M1 relevanten Objekten im sportlichen Alltag.
- 2 Erstellen Sie anhand Ihrer Ergebnisse aus Aufgabe 1 ein Objektdiagramm mit folgenden Objekten: eine Fußballmannschaft, drei Spieler, zwei Trainer, ein Mannschaftsarzt. Beschränken Sie sich aus Gründen der Übersichtlichkeit auf wenige Attribute (M1, M2).



Aufgaben zum Projekt

- 3 Erstellen Sie ein Objektdiagramm mit mehreren Beispielobjekten für Ihr Datenbankprojekt und verschaffen Sie sich einen Überblick über die relevanten Beziehungen zwischen den Objekten (M2).

2.1.6 Wie unterscheiden sich Beziehungen zwischen Objekten?

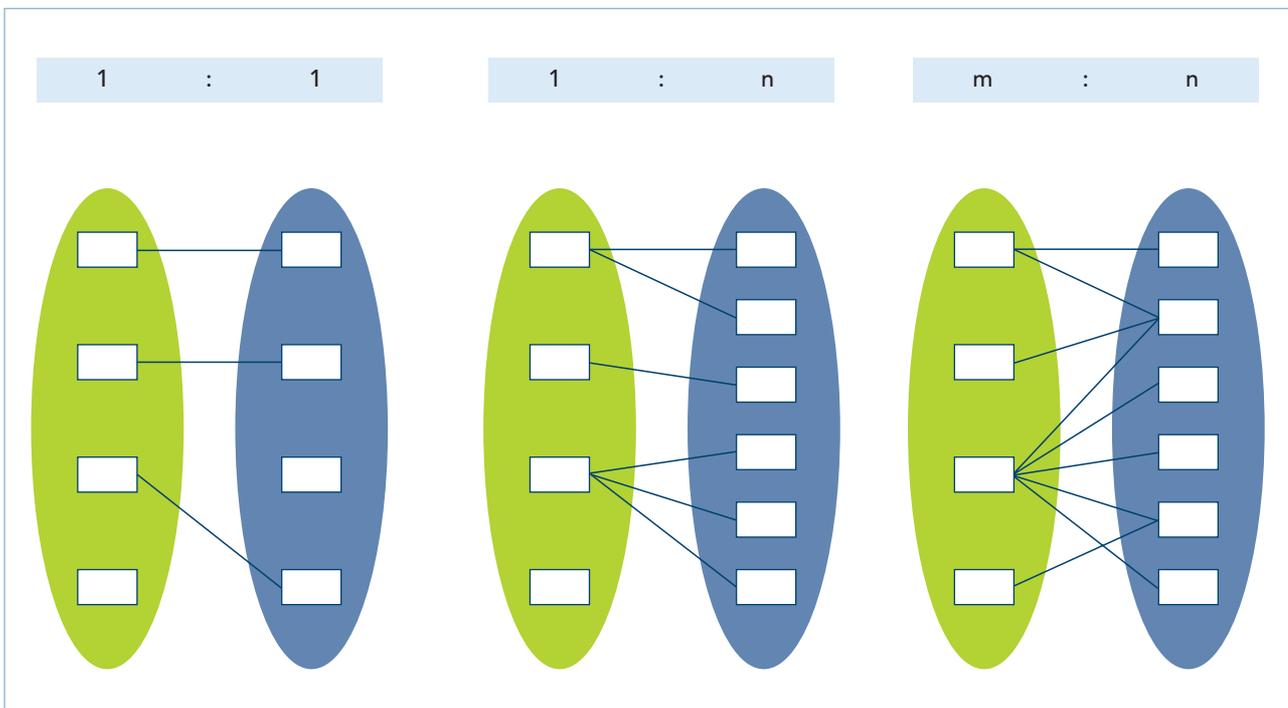
In Objektdiagrammen können Beziehungen zwischen Objekten übersichtlich dargestellt werden. Je nach Anwendungsszenario gibt es jedoch Einschränkungen, da unterschiedlich viele Objekte daran beteiligt sind. Wie kann man dies im Objektdiagramm berücksichtigen?

M1 Beziehungen zwischen Objekten



Beispiele für unterschiedliche Objektbeziehungen

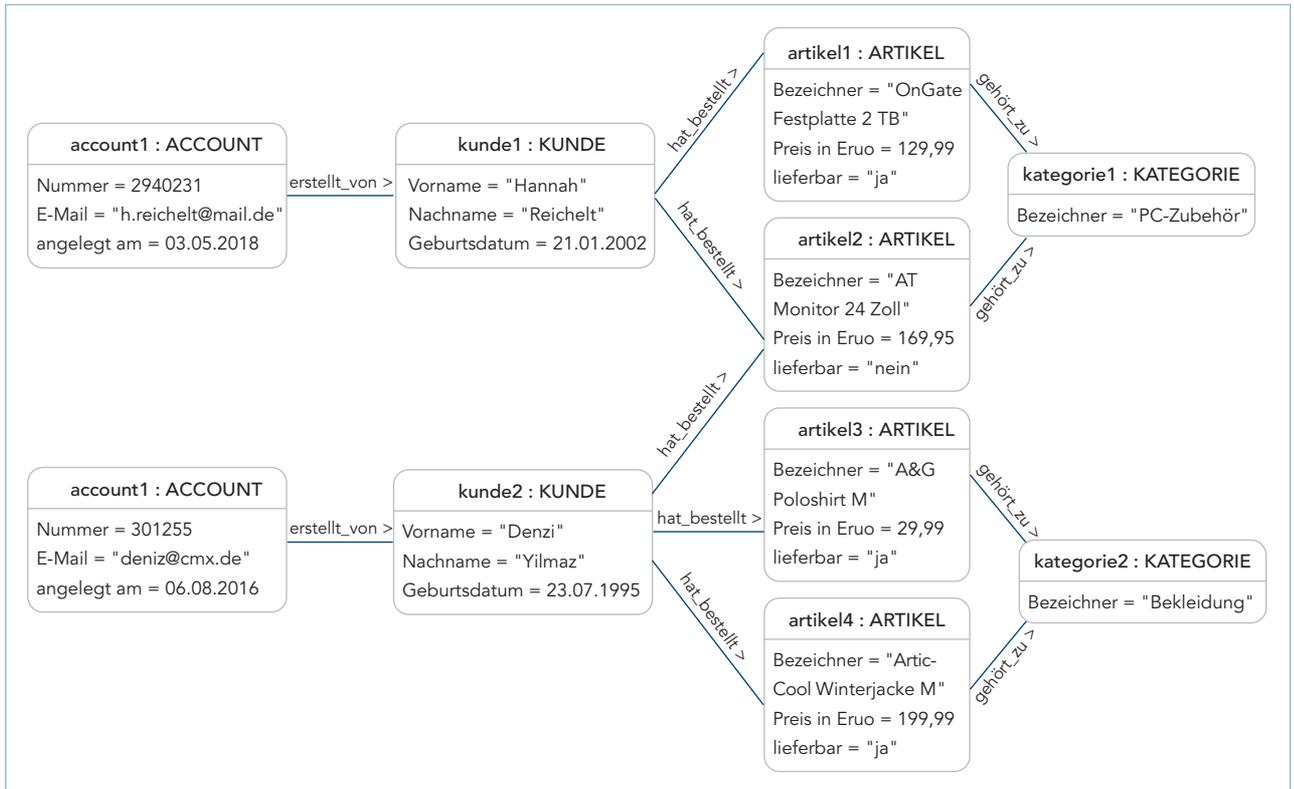
M2 Beziehungstypen



Grafische Veranschaulichung verschiedener Beziehungstypen

Nach: Kemper/Eickler: Datenbanksysteme. Eine Einführung, 2015, S. 43.

M3 Beziehungen im Objektdiagramm



Beziehungen im
Objektdiagramm
Bearbeiter



Projektzieldefinition

Sie entscheiden, welche Beziehungstypen zwischen Objekten Ihrer Datenbank geeignet sind. Dabei berücksichtigen Sie die konkreten Anforderungen Ihres Szenarios.



Aufgaben zum Material

- 1 Beschreiben Sie Einschränkungen bei den Beziehungen der in den Abbildungen M1 beteiligten Objekte.
- 2 Ordnen Sie die Abbildungen aus M1, sowie die Beziehungen des Objektdiagramms M3 den Beziehungstypen aus M2 begründet zu.
- 3 Zeigen Sie anhand von M1 und M2, dass eine eindeutige Zuordnung einer Beziehung zu einem Beziehungstypen ohne Zusatzinformationen schwierig sein kann.



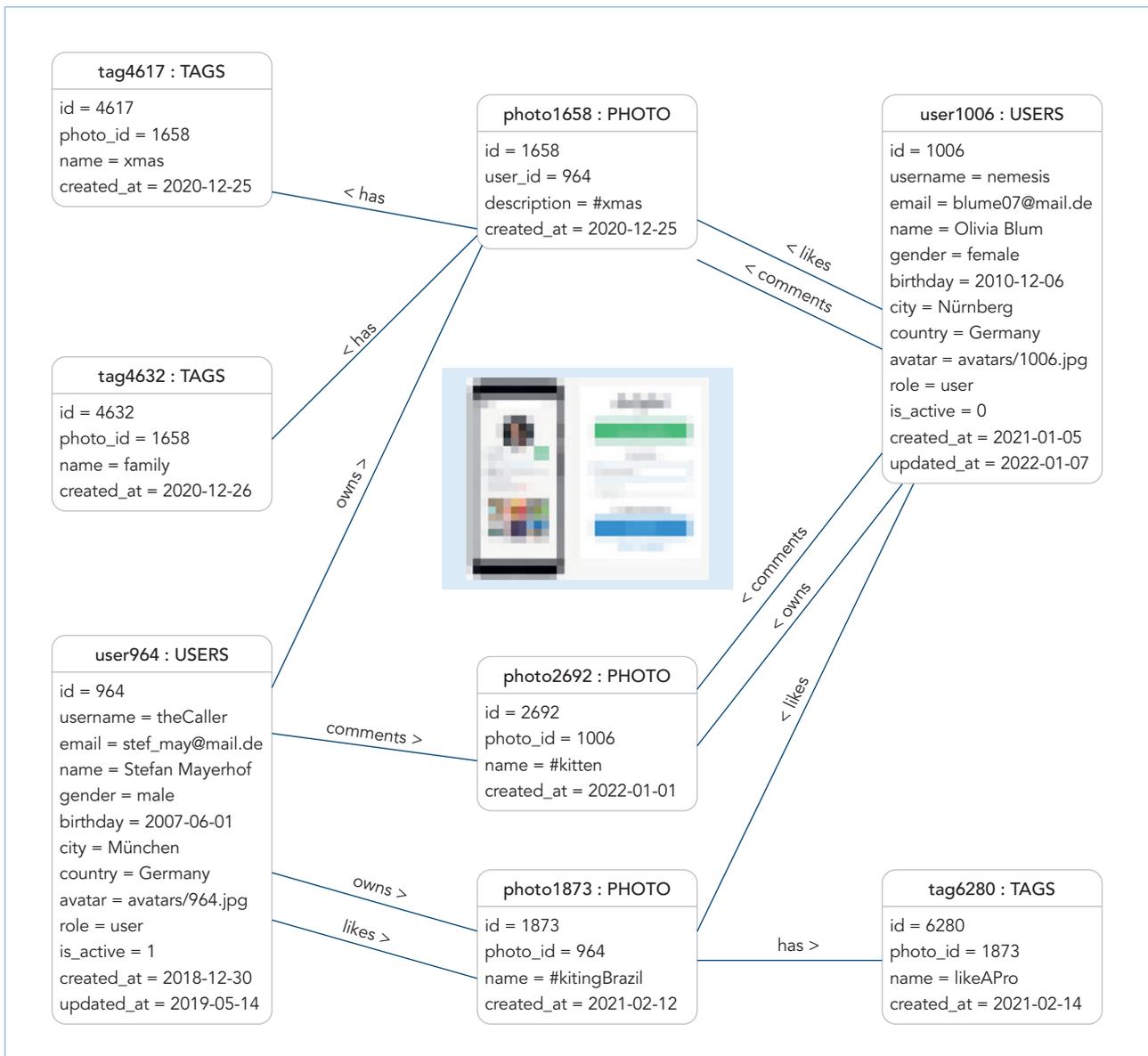
Aufgaben zum Projekt

- 4 Ordnen Sie die Objektbeziehungen, die Sie bisher im Rahmen Ihres Datenbankprojekts definiert haben, einem der Beziehungstypen aus M2 zu.
- 5 Diskutieren Sie, welche Beziehungstypen mit Blick auf die Anforderungen Ihrer Datenbank am geeignetsten erscheinen.

2.1.7 Wie stellt man den Entwurf einer Datenbank übersichtlich dar?

Objektdiagramme werden schon bei kleinen Datenbankprojekten recht unübersichtlich. Wie kann nun diese „Unordnung“ in eine strukturierte Übersicht gebracht werden, damit später daraus die Datenbank wird?

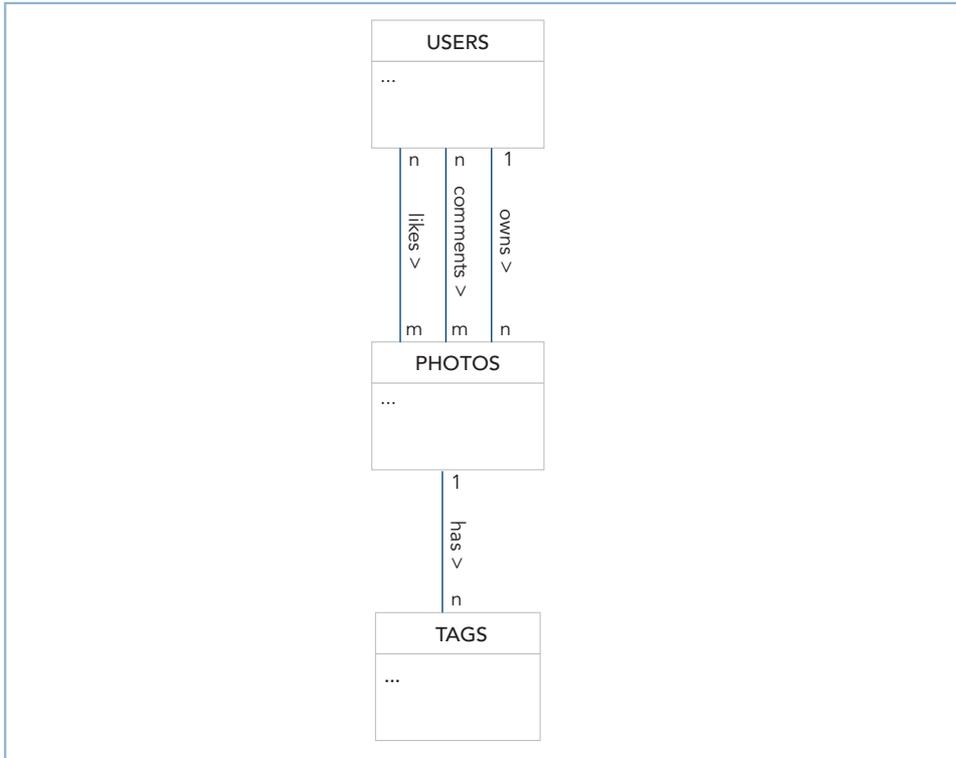
M1 InstaHub – ein soziales Netzwerk



Ausschnitt eines denkbaren Objektdiagramms für die Datenbank eines sozialen Netzwerks

Bearbeiter

M2 Klassendiagramm eines sozialen Netzwerks



Einfaches Klassendiagramm (ohne Attribute) mit den Klassen USERS, PHOTOS und TAGS

Bearbeiter



Projektzieldefinition

Sie modellieren mithilfe eines Klassendiagramms den von Ihnen gewählten Ausschnitt aus der realen Welt und erkennen so schon die Struktur der späteren Datenbank.



Aufgaben zum Material

- 1 Bestimmen Sie gleichartige Objekte (M1) und beschreiben Sie für jede Gruppierung jeweils einen „allgemeingültigen Bauplan“.
- 2 Erstellen Sie für gleichartige Objekte (M1) die jeweils passenden Klassen, indem Sie bei den rechteckigen Klassenkarten auf Objektbezeichner und Attributwerte verzichten.
- 3 Begründen und erklären Sie unter Berücksichtigung von M1 die Beziehungsarten (Kardinalitäten) im Klassendiagramm in M2.
- 4 Erweitern Sie das Klassendiagramm M2 gegebenenfalls um eine Klasse, sodass eine 1:1-Beziehung entsteht.



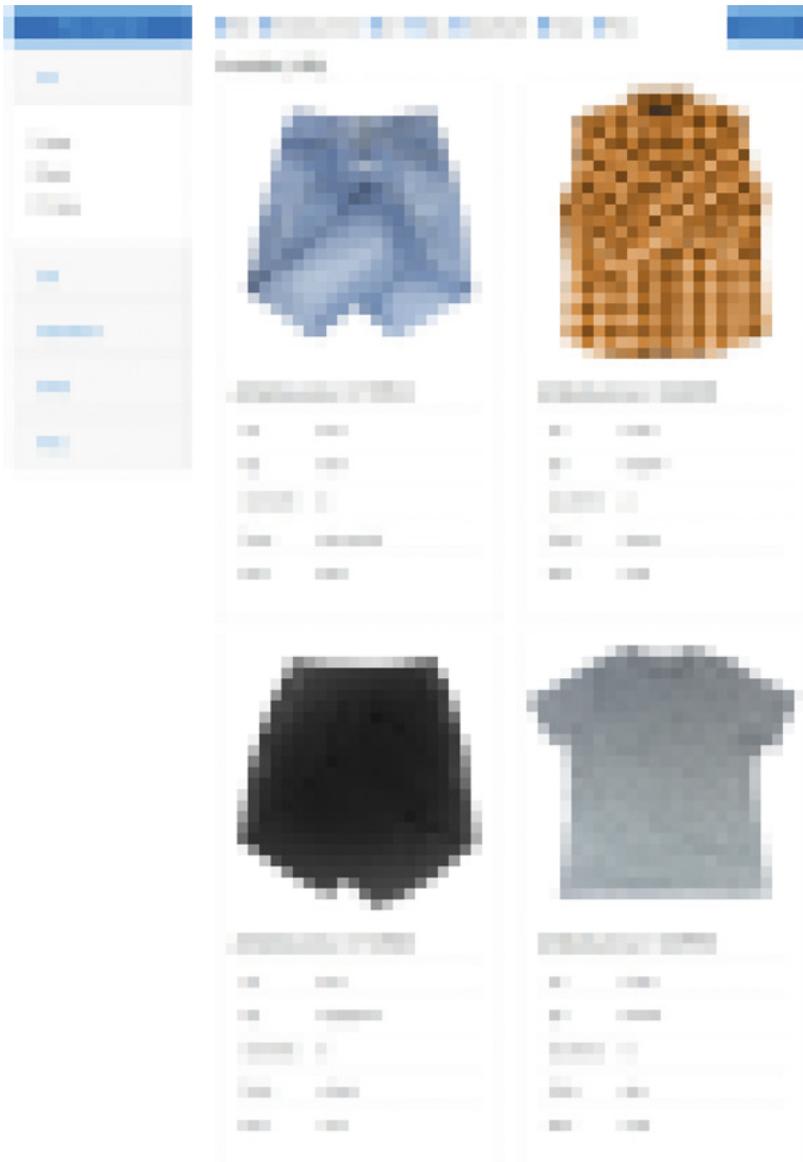
Aufgaben zum Projekt

- 5 Entwerfen Sie ein Klassendiagramm (mit Attributen) zu Ihrem Datenbankprojekt.

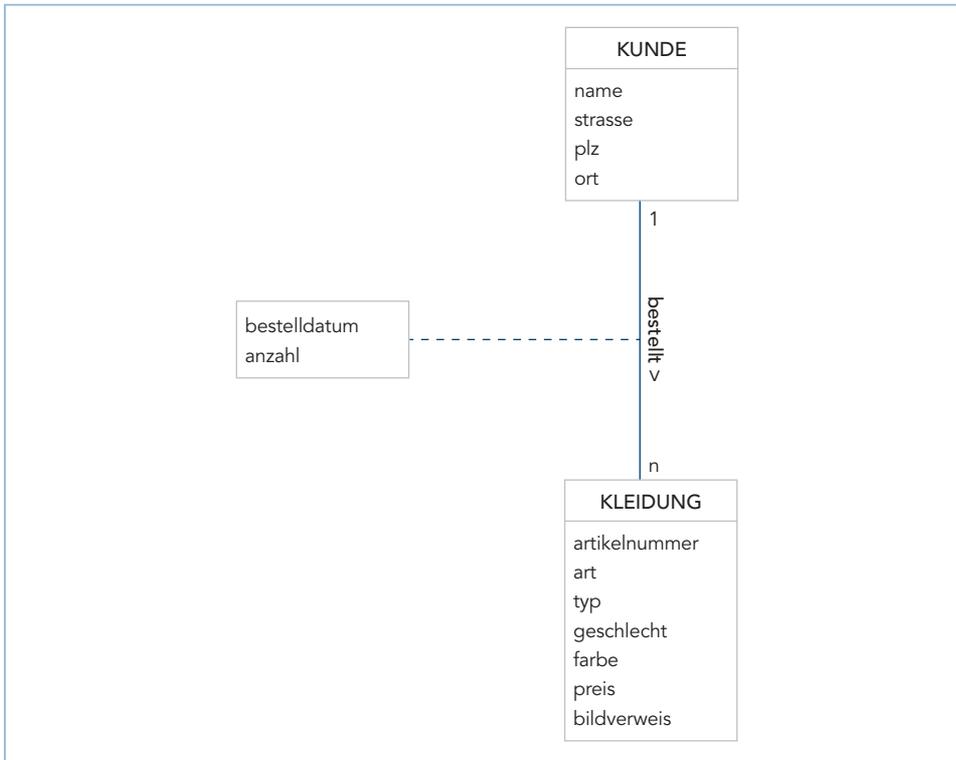
2.1.8 Welche Sonderfälle für Beziehungstypen gilt es in der Planung der Datenbank zu berücksichtigen?

Nicht nur zwischen Objekten unterschiedlicher Klassen, sondern auch zwischen gleichartigen Objekten können Beziehungen bestehen. Außerdem gibt es Attribute, die erst im Zusammenspiel zweier Klassen Sinn machen. Wie können diese Besonderheiten für die Datenbank umgesetzt werden?

M1 Onlineshop für Modeartikel



M2 Informationen, die zu einer Bestellung gehören



Beispiel für eine Assoziationsklasse
Bearbeiter



Projektzieldefinition

Sie erweitern die Modellierungstechnik, indem Sie besondere Beziehungstypen in einem semantischen Modell bzw. Klassendiagramm berücksichtigen.



Aufgaben zum Material

- 1 Ermitteln Sie verschiedene Klassen, die mit der Klasse KLEIDUNG gemäß den Modeartikeln des Onlineshops (M1) in Beziehung stehen können. Beschreiben Sie die jeweiligen Beziehungen und bestimmen Sie die Kardinalitäten.
- 2 Begründen Sie, warum die Attribute „bestelldatum“ und „anzahl“ nicht zu einer der beiden Klassen KUNDE und KLEIDUNG, sondern zur Beziehung „bestellt“ gehören und damit eine Assoziationsklasse bilden (M1, M2).
- 3 Stellen Sie folgende Beziehung im Klassendiagramm (M2) dar: „Kunde wirbt Kunde“.



Aufgaben zum Projekt

- 4 Erweitern Sie das Klassendiagramm Ihres Datenbankprojekts um potenzielle Assoziationsklassen und berücksichtigen Sie mögliche rekursive Beziehungen.

2.1.9 Wie kann ein Datenbankentwurf verbessert werden?

Ein Datenbanksystem muss nicht nur seinen Anforderungen entsprechen, es sollte auch fehlerfrei funktionieren. Immer wieder schleichen sich jedoch Fehlerquellen in Datenbankentwürfen ein. Dies fällt beispielsweise dann auf, wenn gespeicherte Daten im Laufe der Zeit verändert werden und an manchen Stellen immer noch die alten Daten zu finden sind. Doch wie kann man einen Datenbankentwurf auf Fehlerquellen untersuchen?

M1 Redundanz und Inkonsistenz im Alltag



Stopschild



Fallschirme

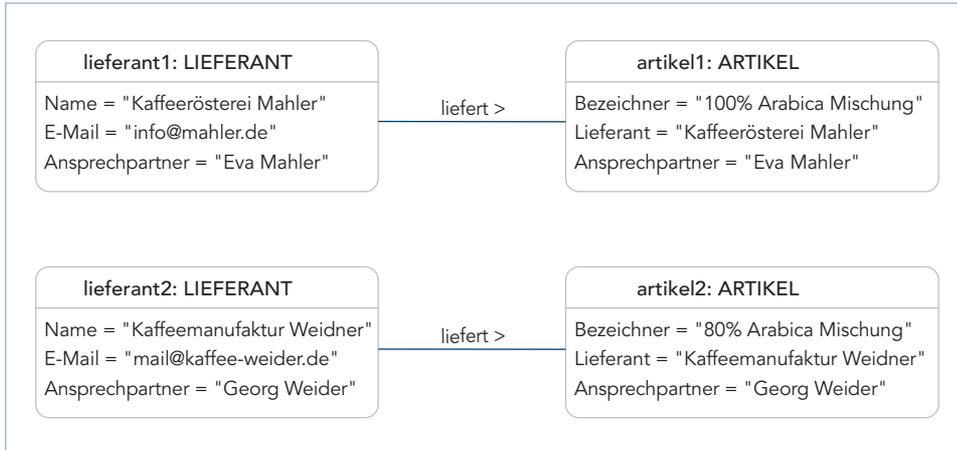


Flugzeugcockpit



Verkehrsschilder

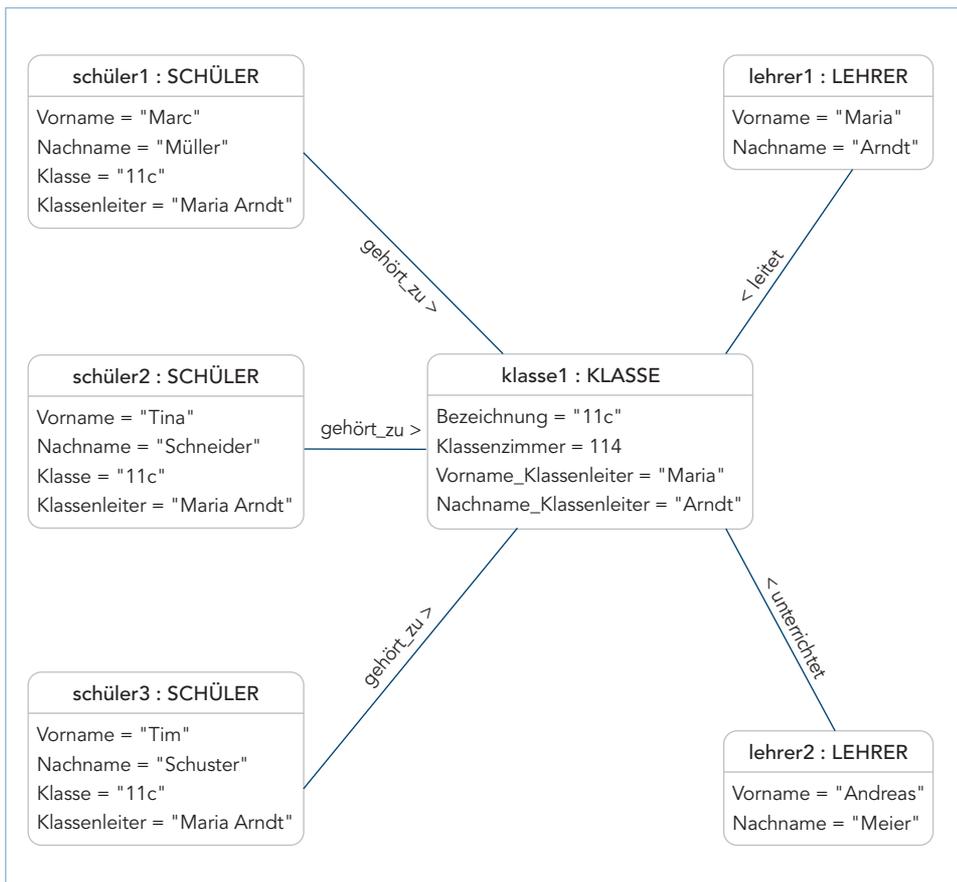
M2 Redundanz im Objektdiagramm eines Cafés



Ausschnitt eines Objektdiagramms eines Cafés

Bearbeiter

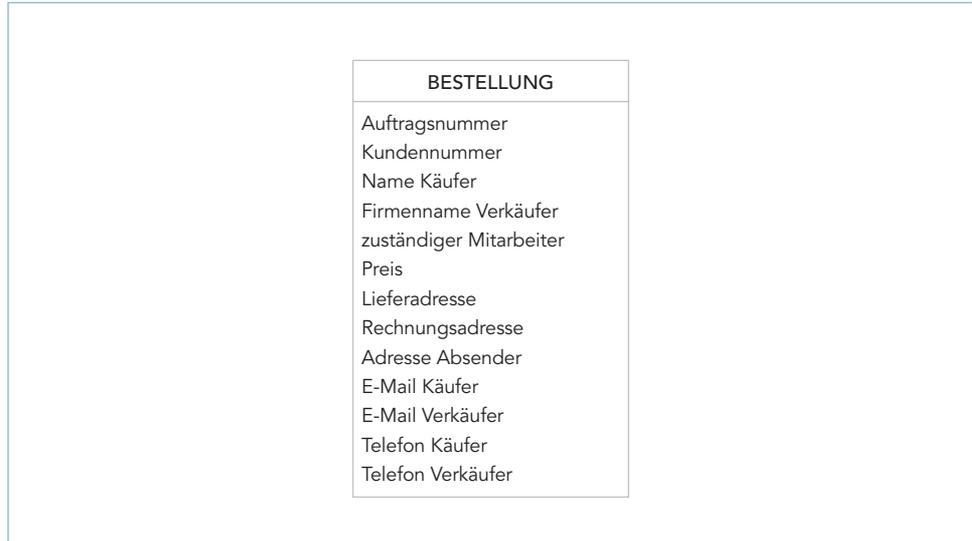
M3 Redundanz im Objektdiagramm einer Schulklasse



Ausschnitt eines Objektdiagramms einer Schulklasse

Bearbeiter

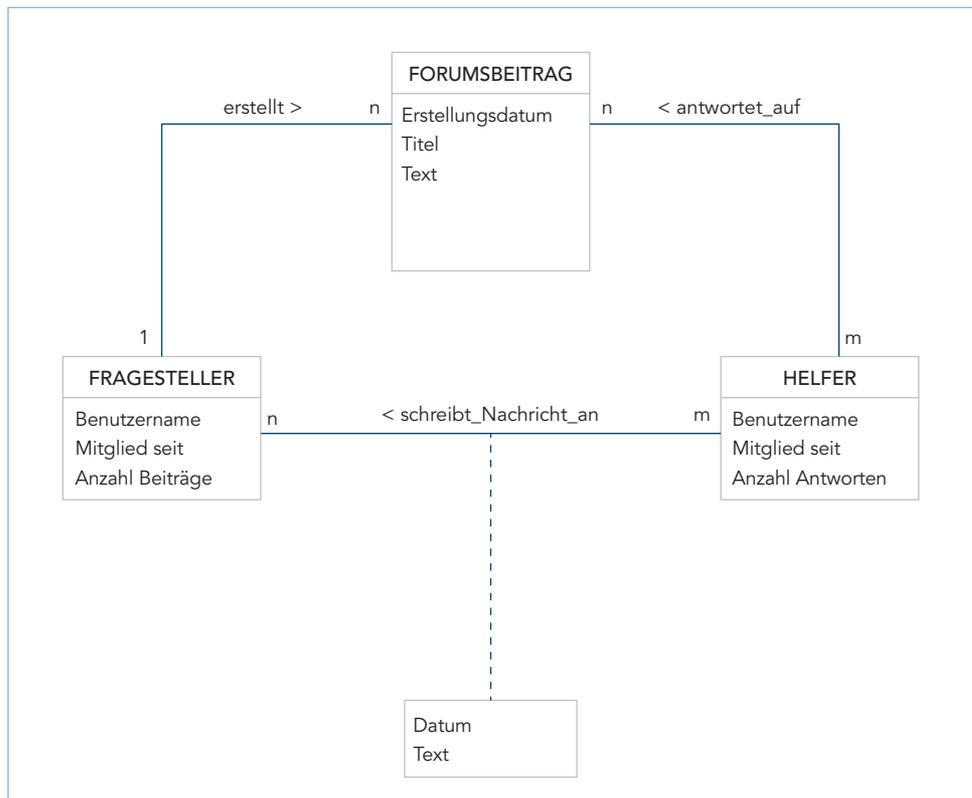
M4 Beispiel 1 – Wozu Beziehungen? Eine Klasse genügt!



Klassendiagramm eines Onlineshops

Bearbeiter

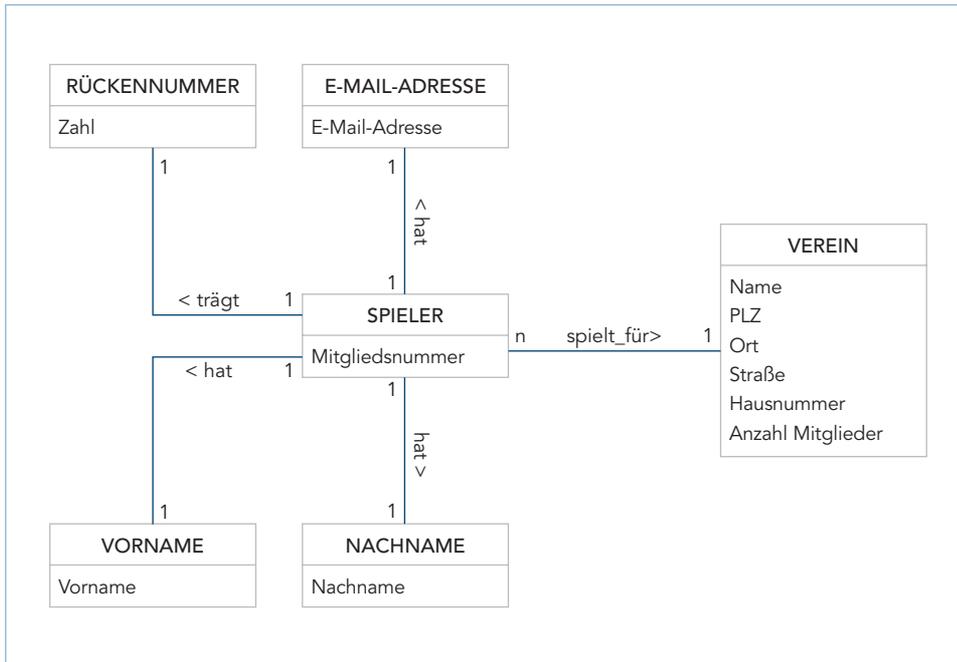
M5 Beispiel 2 – Wer macht was?



Klassendiagramm eines Online-Forums

Bearbeiter

M6 Beispiel 3 – Klassen ohne Ende



Klassendiagramm eines Fußballvereins
 Bearbeiter



Projektzieldefinition

Sie analysieren Ihren Datenbankentwurf auf Redundanzen und mögliche Inkonsistenzen und verbessern ihn.



Aufgaben zum Material

- 1 Recherchieren Sie die Begriffe „Redundanz“ und „Inkonsistenz“ und erläutern Sie deren Bedeutung in den abgebildeten Situationen (M1).
- 2 Analysieren Sie M2 und M3 im Hinblick auf Redundanzen und möglicherweise auftretende Inkonsistenzen.
- 3 Zeigen Sie anhand konkreter Beispiele, dass die Redundanzen in M2 zu Inkonsistenz führen können.
- 4 Verbessern Sie arbeitsteilig in einer Partner- oder Gruppenarbeit die Datenbankentwürfe (M4 bis M6).
 - a) Erstellen Sie ein Objektdiagramm mit wenigen Beispielobjekten.
 - b) Führen Sie Änderungen am Datenbestand durch und überprüfen Sie das Objektdiagramm auf Inkonsistenz.
 - c) Diskutieren Sie in der Gruppe Verbesserungen des Datenbankentwurfs, die die festgestellten Probleme beheben.



Aufgaben zum Projekt

- 5 Verbessern Sie den Entwurf Ihres Datenbankprojekts.

Fachwissen im Zusammenhang – Lernbereich 2.1

Einsatz von Datenbanken

→ Kapitel 2.1.1

Heutzutage gibt es so gut wie kein größeres Unternehmen, welches im Zeitalter der Digitalisierung auf den **Einsatz von Datenbanken** verzichtet. Über alle Branchen hinweg finden sich Bereiche, in denen Datenbanken verwendet werden, auch um den Kundinnen und Kunden einen Mehrwert zu bieten.

Content Management Systeme (CMS)

CMS dienen der einfachen Verwaltung digitaler Inhalte zumeist von Internetseiten. Dabei benötigen die Anwenderinnen und Anwender keine oder kaum Programmierkenntnisse, da das System sowohl für die Kundinnen und Kunden als auch für denjenigen, der die Seiten pflegt (Admin), eine virtuelle Oberfläche bietet.

Customer Relationship Management (CRM)

CRM-Systeme kommen immer dann zum Einsatz, wenn die Beziehung eines Unternehmens zu seinen Kundinnen und Kunden oder die Beziehung einer Organisation zu seinen Mitgliederinnen und Mitgliedern verwaltet und gestaltet werden muss. Dazu werden Kundendaten erfasst und für unternehmensinterne Auswertungen zur Verfügung gestellt.

Enterprise-Resource-Planning (ERP)

ERP-Systeme unterstützen die Planung eines möglichst effizienten betrieblichen Ablaufs und decken als integrierte Unternehmenssoftware die gesamte Bandbreite betriebswirtschaftlicher Aufgaben ab. Dadurch können sämtliche Geschäftsprozesse im Unternehmen unterstützt werden: z. B. im Bereich des Marketings, der Personalverwaltung, des Rechnungswesens, der Beschaffung.

Bestandteile eines Datenbanksystems

→ Kapitel 2.1.1

Um die Datenbanken zu verwalten, kommen **Datenbankmanagementsysteme** zum Einsatz, welche es in Abhängigkeit der Rechteverwaltung den Benutzerinnen und Benutzern erlauben, auf die Datenbasis zuzugreifen, diese zu verändern oder neue Daten abzuspeichern. Es handelt sich dabei also um Softwaresysteme, welche die zur Datenverarbeitung notwendigen Programme bereitstellen.

Ein **Datenbanksystem** (DBS) unterstützt die computerbasierte Datenverarbeitung von Informationen. Es besteht aus einer oder mehreren Datenbanken (DB) sowie einem Datenbankverwaltungssystem (DBMS für engl.: database management system).

Die **Datenbank** ist eine Datei, in der eine logisch zusammenhängende Sammlung von Daten gespeichert wird. Diese Datensätze repräsentieren einen Ausschnitt der realen Welt, der zur Kontrolle und Steuerung eines Aufgabenbereichs oder eines ganzen Unternehmens notwendig ist. In einer relationalen Datenbank werden die Daten in zweidimensionalen Tabellen (Relationen) verwaltet.

Motivation zur Verwendung von Datenbanken

→ Kapitel 2.1.1,
Kapitel 2.1.2

Die Bedeutung von Datenbanken lässt sich an den Problemen erkennen, die ohne den Einsatz eines Datenbanksystems bei der Verwaltung von Informationen auftreten. Die meist miteinander in Beziehung stehenden Daten müssten in Papierform abgelegt oder in einzelnen Dateien abgespeichert werden. Dies würde dazu führen, dass Daten mehrfach an verschiedenen Orten für unterschiedliche Nutzerinnen und Nutzer gespeichert werden müssten (**Mehrfachspeicherung von Daten**). Eine Änderung der Daten müsste dann an allen Speicherorten durchgeführt werden, was zu unterschiedlichen Dateiversionen führen kann (**Widersprüchlichkeit von Daten**). Auftretende **Datenverluste** (z. B. ein Brand im Archiv) wären im Allgemeinen nicht auszuschließen. Heutige Dateisysteme erlauben **keinen sinnvollen Mehrbenutzerbetrieb**; Daten werden aber von sehr vielen Anwenderinnen und Anwendern innerhalb und außerhalb von Unternehmen und Organisationen genutzt (z. B. Platzreservierungen), was ohne DBS unerwünschte Anomalien beim Datenbestand hervorrufen würde. Schwierig wäre ebenfalls die **Koordinierung der Zugriffsrechte**. Datenbanksysteme ermöglichen eine zentrale und dauerhafte Speicherung der Daten und verringern oder vermeiden gar die Probleme.

Anforderungs- analyse

→ Kapitel 2.1.3

Bevor mit der Planung einer Datenbank begonnen werden kann, muss ihr Einsatzzweck bekannt sein. Um nachträgliche Anpassungen zu vermeiden, sollten die Anforderungen an das System möglichst vollständig und strukturiert erfasst werden. Die Identifikation von konkreten **Anwendungsfällen bzw. Use-Cases** der Akteure, die die Datenbank verwenden sollen, kann dabei helfen.

Notation	Erklärung
	Systemrahmen mit dem Namen des zu erstellenden Systems.
 Akteur	Akteure können beteiligte Personen oder auch andere Systeme sein. Bei Personen wird hier die Bezeichnung ihrer Rolle im System zur Beschriftung verwendet.
	Verbindungslinien zwischen Akteur und Use-Case, die Antwort auf die Frage „Wer tut oder darf was?“ geben.
	Funktion bzw. Funktionalität des zu erstellenden Systems. Ein Use-Case wird mindestens mit einem Nomen („Wer oder was tut etwas?“) und einem Verb („Was geschieht/passiert?“) beschrieben.

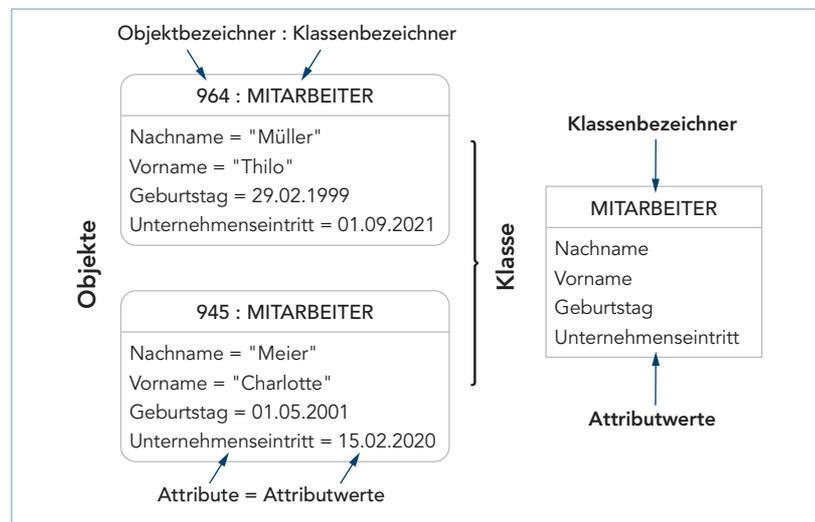
Nach: Metzner, Anja: *Software Engineering kompakt*, 2020, S. 73 f.

Objekte, Klassen und semantisches Modell

→ Kapitel 2.1.4-
2.1.7

Das semantische Modell, auch als **Klassendiagramm** bezeichnet, stellt eine formale Beschreibung des abzubildenden Ausschnitts aus der realen Welt dar. Dieses wird unabhängig vom Datenbanksystem und nur aus der Sicht des Anwenders modelliert.

Dabei werden gleichartige **Objekte**, d. h. physisch oder gedanklich existierende Gegenstände/Konzepte, sogenannte Entitäten, mit gleichen Attributen (z. B. Mitarbeiterinnen und Mitarbeiter eines Unternehmens mit Namen, Geburtsdatum, Eintritt), zu einer **Klasse** (z. B. MITARBEITER) abstrahiert. Die Objektbezeichner und die Attributwerte werden nicht übernommen.



Gleichartige Beziehungen zwischen den Objekten werden ebenfalls abstrahiert, welche die **Beziehung** zwischen zwei Klassen charakterisieren; dargestellt werden diese als ungerichtete, aussagekräftig beschriftete Kanten, welche zwei Klassen miteinander verbinden.

Kardinalitäten beschreiben quantitativ die Art der Beziehung zwischen zwei Klassen; d. h. wie viele Beziehungen von einem Objekt einer Klasse zu Objekten einer anderen Klasse existieren können. Für die Angabe der Kardinalität wird die (1, M, N)-Notation verwendet.

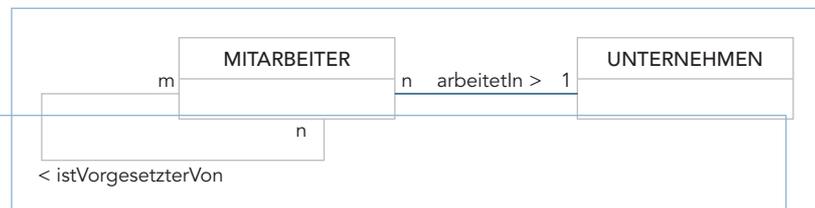
Kardinalität	Bedeutung
1:1-Beziehung	Jedes Objekt einer Klasse hat genau eine Beziehung zu einem Objekt einer anderen Klasse. Dies gilt auch umgekehrt.
1:n-Beziehung	Jedes Objekt einer Klasse hat besitzt Beziehungen zu mehreren Objekten einer anderen Klasse. Umgekehrt haben mehrere Objekte der anderen Klasse eine Beziehung zu genau einem Objekt der einen Klasse.
m:n-Beziehung	Jedes Objekt einer Klasse kann mehrere Beziehungen zu den Objekten der anderen Klasse haben.

Assoziationsklassen und rekursive Beziehungen

→ Kapitel 2.1.8

Es gibt Attribute, welche die Beziehung zwischen zwei Klassen beschreiben. Diese gehören somit nicht zu einer einzelnen Klasse, da diese Attribute nur im Zusammenspiel der beiden Klassen einen Sinn ergeben. Deshalb wird eine **Assoziationsklasse** mit diesen Attributen – diese Klasse hat keinen eigenen Namen – über eine gestrichelte Linie mit der Beziehung verbunden.

Nicht nur zwischen Objekten unterschiedlicher Klassen können Beziehungen existieren, auch zwischen Objekten derselben Klasse können Beziehungen bestehen. Diese heißen **rekursive Beziehungen**. Sind in einem Klassendiagramm sämtliche Mitarbeiter eines Unternehmens (z. B. Sachbearbeiterinnen und Sachbearbeiter, Teamleitung, Abteilungsleiterinnen und Abteilungsleiter, Geschäftsführung) als Objekte innerhalb einer Klasse MITARBEITER enthalten, so wäre es möglich das hierarchische Verhältnis zwischen Führungskräften und Ausführenden durch eine reflexive Beziehung „istVorgesetzterVon“ zu beschreiben, indem sich die Beziehung auf die Klasse MITARBEITER selbst bezieht.



Beseitigung von Redundanzen und Vermeidung von Inkonsistenzen

➔ Kapitel 2.1.9

Eine Datenbank kann ihren Zweck nicht erfüllen, wenn sie widersprüchliche Daten enthält. Eine solche **Inkonsistenz** kann durch **Redundanzen** entstehen. Man bezeichnet Daten als **redundant**, wenn sie ohne Informationsverlust weggelassen werden können, da sie mehrfach gespeichert wurden. Zur Vermeidung von Inkonsistenz muss das semantische Modell somit auf Redundanz untersucht werden. Folgende Aspekte sollten dabei berücksichtigt werden:

- Eine Klasse darf nur die Attribute beinhalten, die sachlich bzw. logisch einer anderen Klasse zuzuordnen sind. Fremde Attribute müssen in andere Klassen ausgelagert werden.
- Attribute, die sachlich bzw. logisch einer bestimmten Klasse zuzuordnen sind, sollten nicht in anderen Klassen enthalten sein. Ggf. müssen Klassen zusammengefasst werden.
- Attribute sollten atomar sein, d. h. sie sollten nicht mehrere Informationen in sich vereinen.

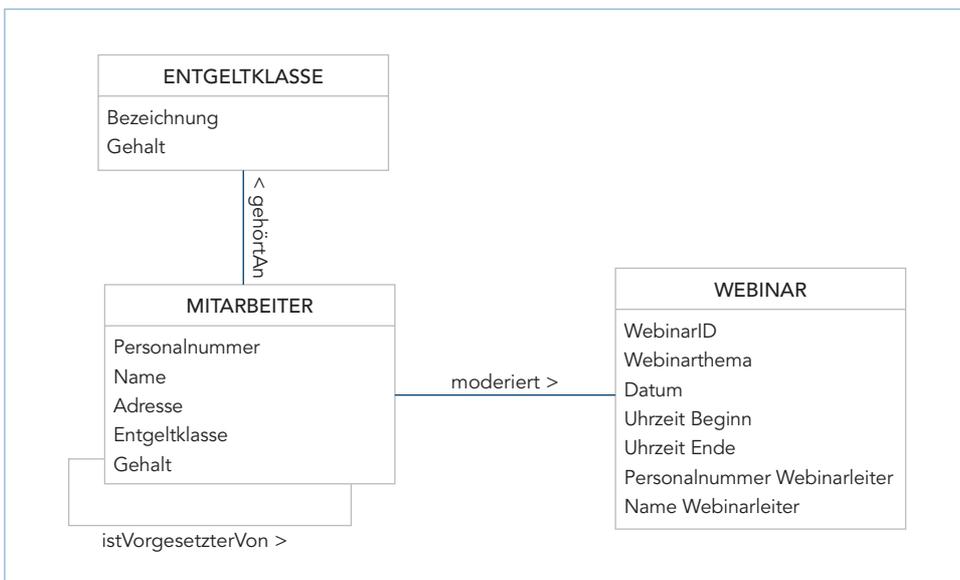
Kompetenzcheck

M1 Virtuelles Lernen mit einem Webinar



Webinar

M2 Semantisches Modell für die Datenbank des Webinar-Anbieters



Semantisches Modell

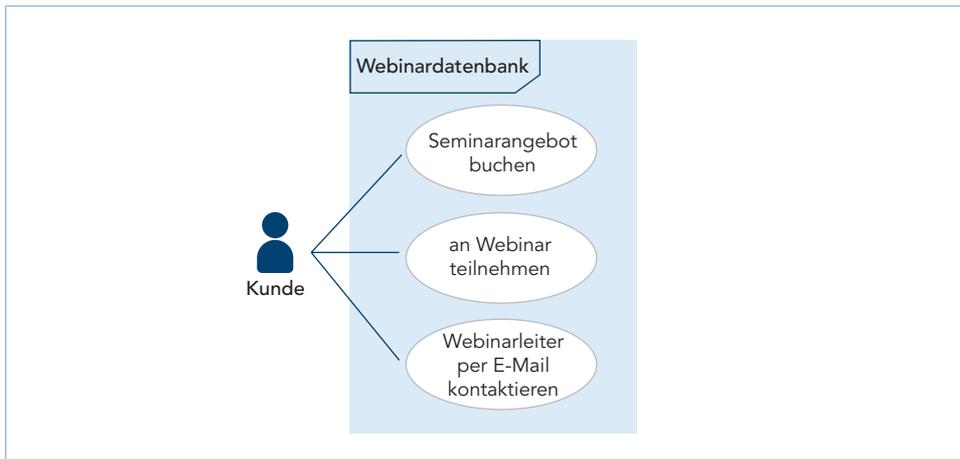
Bearbeiter

Draw.io
Vorlage zum
Download



Mediencode
82111-14

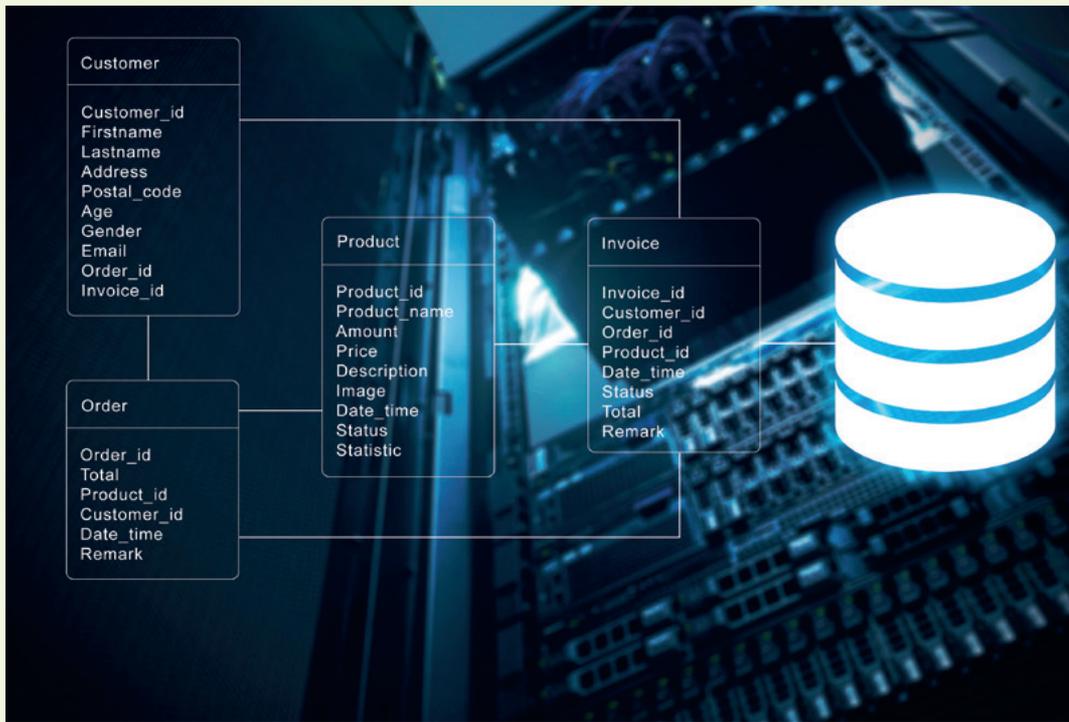
M3 Anforderungen der Kunden



Use-Case-Diagramm
Bearbeiter

Aufgaben

- 1 Identifizieren Sie verschiedene Akteure, die an der Nutzung der Datenbank (M1, M2) interessiert sind und formulieren Sie jeweils geeignete Anwendungsfälle.
- 2 Ergänzen Sie das semantische Modell (M2) anhand der Ergebnisse aus Aufgabe 1.
- 3 Bestimmen Sie die fehlenden Kardinalitäten im erweiterten semantischen Modell und begründen Sie Ihre Entscheidung.
- 4 Untersuchen Sie M2 auf mögliche Inkonsistenz und verbessern Sie das semantische Modell entsprechend.
- 5 Ergänzen Sie M2 um die Klasse KUNDE unter Berücksichtigung von M3. Ergänzen Sie ggf. bereits vorhandene Klassen.



Relationale Datenbank

Aufgaben

- 1 Formulieren Sie Szenarien, in denen Unternehmen Informationen aus ihren Datenbanksystemen herausfiltern müssen.
- 2 Beurteilen Sie, welche Herausforderungen entstehen können, wenn Datenbestände aus umfangreichen Datenbanksystemen ausgelesen werden.

Datenbank- systeme

2

2.2 Relationale Datenbanksysteme

Eine relationale Datenbank wird für die Speicherung und Bearbeitung von Daten durch Computersysteme verwendet. Sie beruht auf dem relationalen Datenbankmodell und speichert Daten in verschiedenen Tabellen, die untereinander in Beziehung stehen. Das relationale Datenbankmodell wurde erstmals im Jahr 1970 von Edgar F. Codd [...] vorgestellt. Bis heute sind relationale Datenbanken ein etablierter Standard und sehr weit verbreitet.

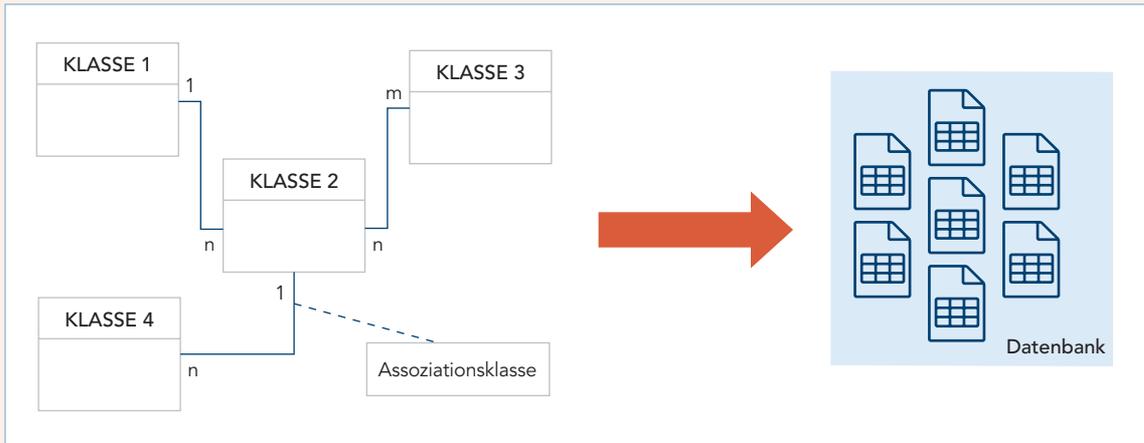
Litzel/Luber: Was ist eine relationale Datenbank?. In: www.bigdata-insider.de, 13.09.2017

Das können Sie nach diesem Kapitel:

- ausgehend vom semantischen Modell das logische Modell für eine relationale Datenbank erstellen.
- das logische Modell mithilfe eines Datenbankmanagementsystems umsetzen.
- unter Beachtung der referenziellen Integrität die Tabellen der Datenbank durch Eintragung oder Import der Daten füllen.
- mithilfe von SQL-Abfragen die Datenbestände einer Datenbank im Hinblick auf vorgegebene Fragestellungen auswerten.
- sich vorausschauend im Umgang mit persönlichen Daten vor dem Hintergrund von Speicherung, Kombination und Auswertung von Daten in Datenbanksystemen verhalten.

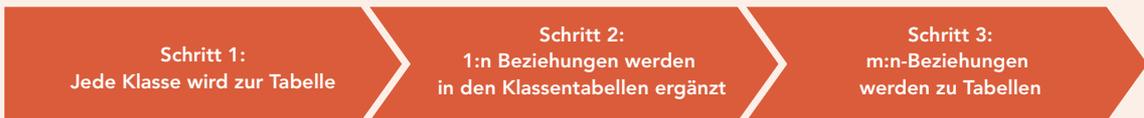
Vom semantischen in das logische Modell

Relationale Datenbanken speichern Daten in Tabellenform. Das von Menschen entworfene semantische Modell muss daher nun informationsverlustfrei in ein Tabellenmodell, das sogenannte logische Modell, überführt werden (vgl. Abb.). Die Bestandteile des semantischen Modells (Klassen, Attribute, Beziehungen) werden auch im logischen Modell abgebildet, hier jedoch in Tabellen. Über das Datenbankmanagementsystem kann dann auf die Tabellen zugegriffen werden und daraus Informationen generiert werden.



Das semantische Modell muss in Tabellen überführt werden
Bearbeiter

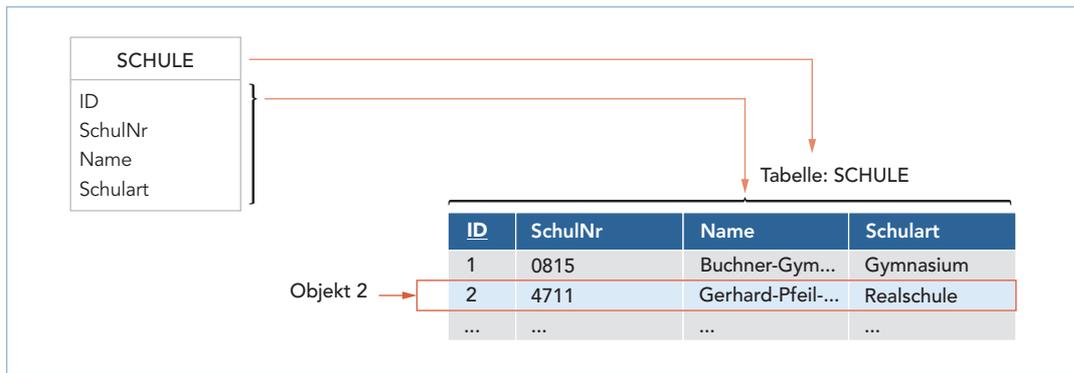
Um das semantische Modell in ein logisches zu überführen, hat sich ein dreistufiges Vorgehen etabliert:



Die Attributwerte, die im Folgenden in den Abbildungen zu sehen sind, dienen demnach nur der Veranschaulichung. Beim Erstellen des logischen Modells müssen keine Attributwerte übertragen werden, das Ziel sind nur die Tabellen und deren Spaltenbezeichner.

Schritt 1: Jede Klasse wird zur Tabelle

Im ersten Schritt wird aus jeder Klasse eine Tabelle. Dabei wird der Klassenbezeichner zum Namen der Tabelle, die Attribute der Klasse werden zu den Spaltenbezeichnern. Ein einzelnes Objekt einer Klasse findet sich später dann in einer Zeile der Tabelle wieder, dort steht dann in der passende Spalte (Attribut) der zum jeweiligen Objekt gehörend Attributwert. Falls bei der Modellierung noch nicht geschehen, wird jedem Objekt ein Attribut als Primärschlüssel zugeordnet. Mithilfe dieses Attributs lassen sich alle Objekte einer Klasse eindeutig identifizieren. Beispiele dafür können fortlaufende Nummern sein, aber auch Attribute, die in der Realität eindeutig vergeben werden, wie Telefonnummer, Mailadresse oder Autokennzeichen. Der Primärschlüssel wird im logischen Modell durch eine Unterstreichung gekennzeichnet.

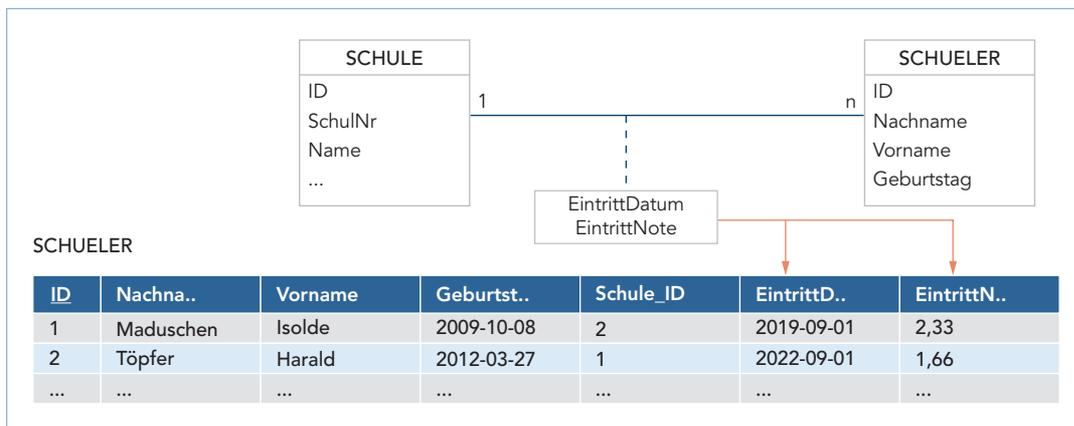


Aus jeder Klasse wird eine Tabelle
 Bearbeiter

Schritt 2: 1:n-Beziehungen in den passenden Tabellen ergänzen

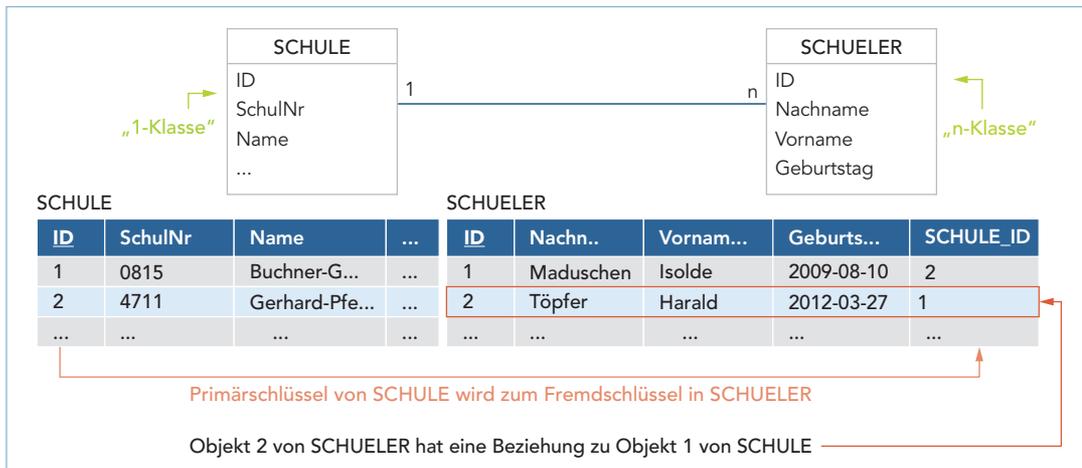
Im zweiten Schritt werden die 1:n-Beziehungen in die Tabellen überführt. Im folgenden wird die Klasse, an deren Anknüpfungspunkt der Beziehung das „n“ steht, als „n-Klasse“ (und die zugehörige Tabelle als „n-Tabelle“) und die andere als „1-Klasse“ bzw. „1-Tabelle“ bezeichnet. Da bei einer 1:n-Beziehung ein Objekt der n-Klasse eine Beziehung zu genau einem Objekt der 1-Klasse hat, wird die Information über diese Beziehung beim Datensatz in der n-Tabelle abgebildet. Man ergänzt also zunächst in der n-Tabelle eine zusätzliche Spalte. In diese Spalte wird nun zu jedem Datensatz (also zu jedem Objekt) der Primärschlüssel des Datensatzes (Objekts) der „1-Tabelle“ eingetragen, zu dem das Objekt der n-Klasse die Beziehung hat.

Dieses Attribut, das auf den Primärschlüssel einer anderen Tabelle verweist, nennt man Fremdschlüssel. Jeder Fremdschlüssel in einer Tabelle stellt also die Verbindung zu einem Primärschlüssel einer anderen Tabelle her und bildet so die 1:n-Beziehung ab.



Überführen einer 1:n-Beziehung in die Tabellenstruktur
 Bearbeiter

Für die Attribute einer Assoziationsklasse einer 1:n-Beziehung werden ebenfalls in der n-Tabelle zusätzliche Spalten angelegt.



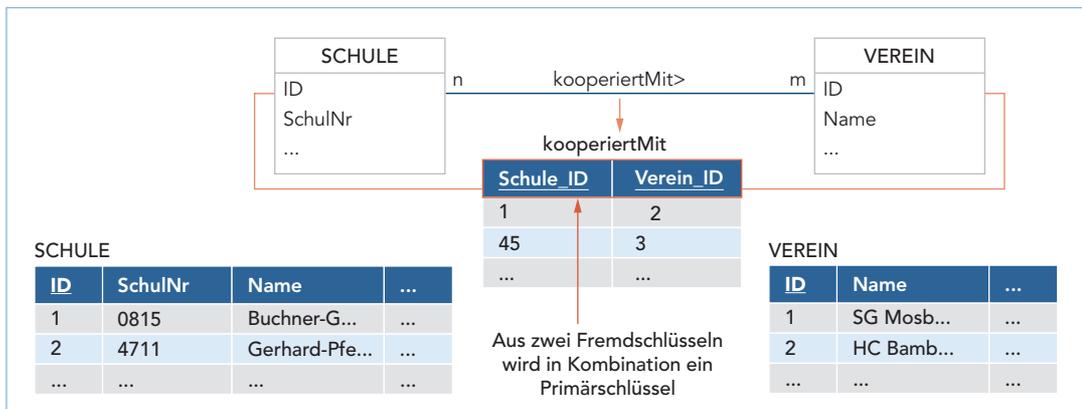
Überführen einer 1:n-Beziehung in die Tabellenstruktur

Bearbeiter

Schritt 3: m:n-Beziehungen werden zu Tabellen

Für eine m:n-Beziehung muss eine neue Tabelle angelegt werden, da eine Ergänzung in einer bereits bestehenden Tabelle (wie bei der 1:n-Beziehung) aufgrund der mehrdeutigen Zuordnung in beide Richtungen nicht funktioniert.

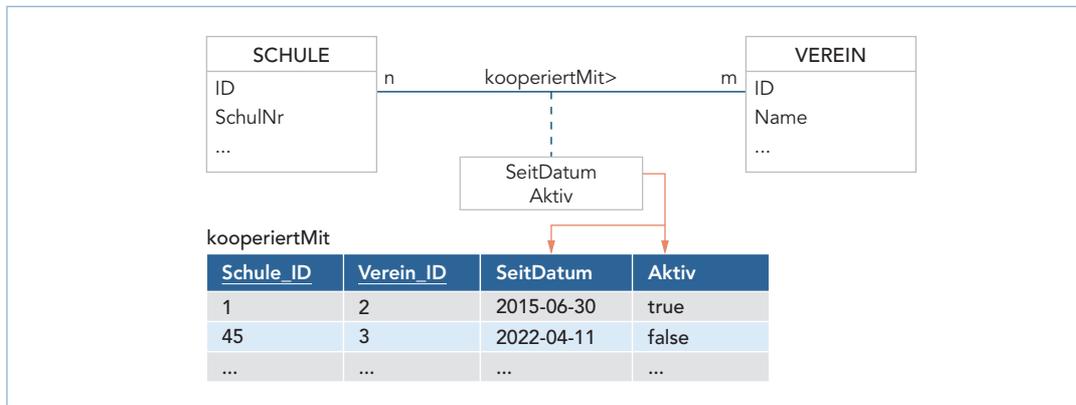
Die Tabelle erhält den Namen der Beziehung und wird aus den Fremdschlüsseln gebildet, die auf die Primärschlüssel der beteiligten Tabellen verweisen. Da die gleiche Beziehung zwischen zwei Objekten nicht mehrfach vorkommen kann, verwendet man die Kombination der Fremdschlüssel als Primärschlüssel der m:n-„Beziehungstabelle“.



Aus einer m:n-Beziehung wird eine eigene Tabelle

Bearbeiter

Für die Attribute einer Assoziationsklasse einer m:n-Beziehung werden in der „Beziehungstabelle“ zusätzliche Spalten angelegt:



Bearbeiter

Kurzschreibweise

Da man die einzelnen Tabellen nicht grundsätzlich in einem Tabellenkalkulationsprogramm anlegen möchte, da man diese danach sowieso in ein Datenbanksystem überführt, bietet sich folgende Kurzschreibweise der Tabellen an:

Tabellenname (Spaltenbezeichner-1, Spaltenbezeichner-2,..., Spaltenbezeichner-x)

Hinter dem Tabellennamen folgen in einer Klammer durch Kommata getrennt die einzelnen Spaltenbezeichner, der Primärschlüssel wird unterstrichen, Fremdschlüssel werden überstrichen. Die Schreibweise der unter Schritt 3 erzeugten Tabelle wäre also:

hatBeziehung (Schule_ID, Verein_ID , SeitDatum, Aktiv)



Projektzieldefintion

Sie überführen Ihr semantisches Modell in ein logisches Modell.



Aufgaben zum Projekt

- 1 Überführen Sie alle Klassen Ihres semantischen Modells in Tabellen.
- 2 Ergänzen Sie in den Tabellen die 1:n-Beziehungen sowie deren Assoziationsklassen.
- 3 Legen Sie die Tabellen der m:n-Beziehungen und der zugehörigen Assoziationsklassen an.
- 4 Stellen Sie Ihr logisches Modell mithilfe der Kurzschreibweise dar.

Wie wird mit einem onlinebasierten Datenbanksystem gearbeitet?

Als Datenbanksystem (DBS) wird die Gesamtheit der Programme bezeichnet, mit denen auf die Datensätze zugegriffen wird und mit denen Daten verändert werden können. Dementsprechend ist es als fortgeschrittener Benutzer oder Datenbankadministrator wichtig zu wissen, wie mit einem onlinebasierten Datenbankmanagementsystem (DBMS) umgegangen wird.

Schritt 1: Beim DBMS anmelden

Onlinebasierte DBMS haben den Vorteil, dass auf sie von vielen internetfähigen Devices zugegriffen werden kann. Durch die integrierte Nutzer- und Rechteverwaltung kann sichergestellt werden, dass sich nur registrierte Verwender des Systems mit den ihnen zugewiesenen Rechten anmelden können.

Schritt 2: Eine Tabelle erstellen oder löschen

- 1 Tabellennamen eintragen
- 2 Attribute angeben
- 3 Datentypen der Attribute wählen
- 4 Primärschlüssel angeben
- 5 Ggf. Primärschlüssel automatisch erzeugen lassen (Hinweis: Häkchen bei Auto Increment, d. h. bei Nummerierungen wird automatisch hochgezählt)

Schritt 3: Eine Tabelle bearbeiten

#	Name	Typ	Kollation	Attribute	Null	Standard	Kommentare	Extra	Aktion
1	attribut1	varchar(256)	latin1_swedish_ci		Nein	kein(e)			Bearbeiten, Löschen, Mehr

- 1 Struktur der Tabelle aufrufen
- 2 Attribute bearbeiten

Schritt 4: Beziehungen zwischen Tabellen anlegen

- 1 Designer der Datenbank auswählen
- 2 Verknüpfungen erzeugen, wenn Attribute bei mehr als einer Tabelle relevant

Table	Attribute
ARTIKEL	AID : int(11), Bezeichnung : varchar(150), Verfügbarkeit : varchar(1), Beschreibung : varchar(500), Verpackungseinheit : varchar(150), Einkaufspreis : decimal(10,2), Listenpreis : decimal(10,2), gehörtZuKategorie : int(11)
KATEGORIE	KOID : int(11), Bezeichnung : varchar(150)

- 3 Den zu referenzierenden Schlüssel wählen (hier: KOID der Tabelle KATEGORIE)
- 4 Fremdschlüssel wählen (hier: gehörtZuKategorie der Tabelle ARTIKEL)

5 Beim Erzeugen der Verknüpfung referentielle Integrität definieren (Ändern oder Löschen eines Attributwertes, der Primärschlüssel ist, nicht möglich bei abhängigem Fremdschlüssel)

6 Angelegte Beziehung (Hinweis: kann die Beziehung nicht angelegt werden, zurück in die Tabellenansicht und beim Fremdschlüssel „Index“ setzen)

Schritt 5: Datensätze in eine Tabelle eintragen

- 1** Nach Auswahl der Tabelle über „Einfügen“ Datensätze eintragen (Hinweis: alternativ können Datensätze auch mithilfe von SQL-Befehlen, z. B. INSERT, eingegeben oder bestehende Datenbestände importiert werden)
- 2** Ggf. bei Fremdschlüssel zugehörigen Primärschlüssel-Wert aus der Liste wählen (Hinweis: Reihenfolge der Eintragung beachten, damit die referentielle Integrität durch Auswahl vorhandener Primärschlüssel-Werte sichergestellt ist)



Schritt 6: Datensätze ändern oder löschen

- 1 Ggf. einzelne Attributwerte direkt ändern (Hinweis: z. B. durch Doppelklick)
- 2 Datensatz ändern (Hinweis: auch hier können Primär- und Fremdschlüssel-Werte eingeschränkt werden)
- 3 Datensatz löschen

Anzeigen Struktur SQL Suche Einfügen

Zeige Datensätze 0 - 6 (7 insgesamt, Die Abfrage dauerte 0.0010 Seku

SELECT * FROM `KATEGORIE`

Alles anzeigen | Anzahl der Datensätze: 25 Zeilen:

+ Optionen

	KAID	Bezeichnung
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	1	Baumarkt
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	2	Haushaltsgeräte

Bearbeiter

Aufgaben

- 1 Melden Sie sich bei einem onlinebasierten Datenbankverwaltungssystem an.
- 2 Vollziehen Sie die Schritte 2, 3, 5 und 6 der Methode praktisch mit einem onlinebasierten Datenbankverwaltungssystem nach, indem Sie z. B. eine Tabelle SCHULBÜCHER mit den Attributen Fach, Autor, Jahr, Titel, Verlag, Seitenzahl erstellen und diese exemplarisch mit Datensätzen füllen.

2.2.1 Wie werden Daten in einem Datenbankmanagementsystem verwaltet?

Nach der Umsetzung des semantischen in das logische Modell muss letzteres in einem Datenbankmanagementsystem implementiert werden. Das Ergebnis sind zunächst leere Tabellen. Wie erfolgt die Umsetzung und wie werden die Datensätze in einer bestehenden Datenbank eingefügt? Und wie können sie später wieder geändert oder gelöscht werden?

M1 „Schüler helfen Schülern“ digital organisiert und durchgeführt



Nachhilfeprogramm



Projektzieldefintion

Sie setzen das logische Modell (z. B. ihr eigenes Datenbankprojekt) in einem DBMS um und füllen unter Beachtung der referenziellen Integrität die Tabellen ihrer Datenbank durch Einfügen, Ändern und ggfs. Löschen von Daten bzw. Datensätzen.



Aufgaben zum Projekt

- 1 Erstellen Sie ausgehend vom logischen Modell ihres Datenbankprojekts Tabellen mit den entsprechenden Attributen, passenden Datentypen sowie Primärschlüssel.
- 2 Legen Sie ggf. Beziehungen zwischen Tabellen an, wenn das Attribut einer Tabelle (= Fremdschlüssel) auf das Attribut einer anderen Tabelle referenziert.
- 3 Füllen Sie die Relationen ihres Datenbankprojekts, in dem Sie die Funktion „Einfügen“ im DBMS aufrufen und die Attributwerte eingeben.
- 4 Testen Sie ggfs. das Ändern und Löschen von Datensätzen im DBMS, indem Sie eine Relation ihres Projekts aufrufen und bei einzelnen Datensätzen „Bearbeiten“ und/oder „Löschen“ auswählen.

2.2.2 Wie werden Datenbankabfragen verfasst?

Datenbanken kommen zum Einsatz, wenn große Datenmengen strukturiert abgelegt und effizient ausgewertet werden sollen. Dabei werden im Rahmen der Datenintegration, also die Zusammenführung von Daten aus unterschiedlichen Abteilungen, in einem zentralen Datenbestand zusammengeführt, um umfassende Analysen durchführen zu können. Aber wie können diese umfassenden Datenbestände ausgewertet werden?

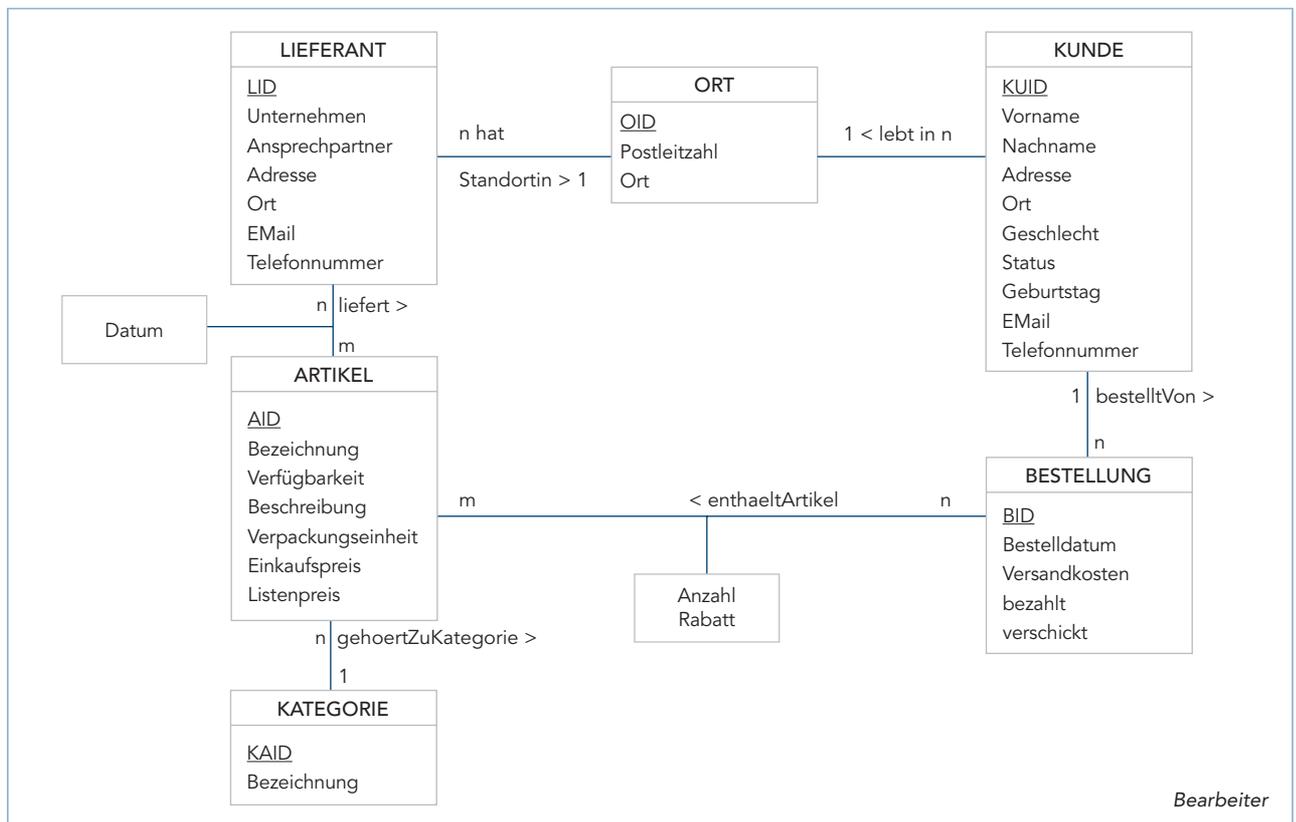
M1 OnlineVersand24

Der Versandhandel OnlineVersand24 vertreibt über den eigenen Onlineshop Produkte aller Art. Elektronikartikel wie Leuchtmittel, Batterien und Mehrfachstecker finden sich im Sortiment genauso wie Schreibwaren, Möbel oder auch Hygieneartikel. Dabei werden die Artikel von unterschiedlichen Lieferanten bezogen

und im eigenen Onlineshop angeboten. Kunden können hier bequem Artikel suchen oder über die entsprechenden Kategorien finden, diese bestellen und erhalten dann alle Artikel einer Bestellung gesammelt zugeschickt.

Bearbeiter

M2 Semantisches Modell des OnlineVersand24



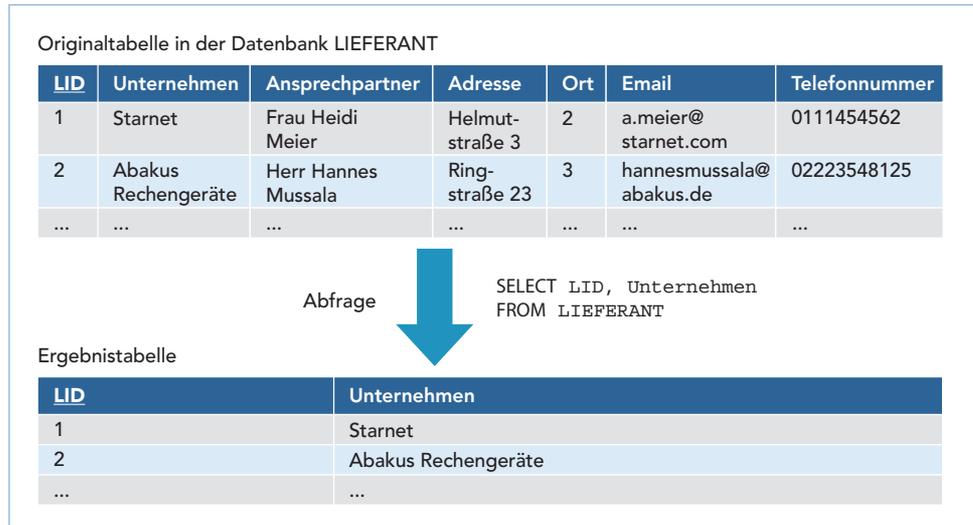
M3 Abfragen in SQL-Syntax erstellen

Um Informationen aus einer Datenbank abzurufen, werden in einer relationalen Datenbank meist Abfragen in der Datenbanksprache SQL (Structured Query Language) verfasst. Damit werden Daten aus einer oder mehrerer Originaltabellen (Tabellen in einer Datenbank) ausgelesen oder Berechnungen ausgeführt. Diese

Ergebnisse werden dann in einer Ergebnistabelle angezeigt.

Jede SQL-Abfrage braucht mindestens zwei Bestandteile: Welche Informationen (Attribute) werden aus welchen Tabellen gesucht? Die Ergebnistabelle liefert dann die ausgewählten Spalten aus dieser Tabelle zurück.

Schlüsselwörter der SQL-Syntax werden aus Gründen der Übersichtlichkeit in Großbuchstaben angegeben, auch wenn nicht alle Systeme dies zwingend verlangen.



Bearbeiter

Die SQL-Abfragen können um weitere Bestandteile, welche die Datensätze ein-

schränken und verändern, ergänzt werden.

Aufgaben

- 1 Sammeln Sie Anforderungen, die an ein Datenbanksystem für den Versandhandel (M1) gestellt werden.
- 2 Beschreiben Sie das semantische Modell der Datenbank in M2.
- 3 Lassen Sie sich im Datenbankmanagementsystem Spalten unterschiedlicher Tabellen anzeigen.
- 4 Finden Sie Gründe, warum Nutzern meist eine grafische Oberfläche angeboten wird und sie nicht SQL eingeben müssen.

SQL-Syntax

Alle Spalten in Ergebnistabelle übernehmen

Sollen alle Spalten der Originaltabelle in die Ergebnistabelle übernommen werden, kann anstatt einer Aufzählung aller Spaltennamen auch das Symbol * verwendet werden.

<pre>SELECT * FROM ORT</pre>	Die Ergebnistabelle enthält alle Spalten der Originaltabelle ORT.
------------------------------	---

Mehrfachnennungen vermeiden

Um die mehrfache Nennung des gleichen Attributwerts zu vermeiden, kann das Schlüsselwort DISTINCT vor dem entsprechenden Attribut (Spalte) verwendet werden.

SQL-Abfrage	Ergebnis
<pre>SELECT DISTINCT Ort FROM ORT</pre>	Ein Ort kann mehrere Postleitzahlen besitzen. Durch die Verwendung von DISTINCT wird jeder Ortsname nur einmal aufgeführt.

Datensätze auswählen

Um nur bestimmte Datensätze in die Ergebnistabelle zu übernehmen, können die Abfragen mit dem Signalwort WHERE um sogenannte Bedingungen ergänzt werden. Es werden dabei nur Datensätze in die Ergebnistabelle übernommen, welche diese Bedingung erfüllen.

<pre>SELECT * FROM ARTIKEL WHERE Listenpreis < 100</pre>	Die Ergebnistabelle enthält alle Informationen der Artikel, die einen Listenpreis besitzen, der unter 100 liegt.
<pre>SELECT Postleitzahl FROM ORT WHERE Ort = "Geiselhöring "</pre>	Die Ergebnistabelle beinhaltet die Postleitzahlen der Orte, deren Name Geiselhöring ist.

Die Vergleichswerte (hier 100 oder Geiselhöring) werden nach Vorgaben des Datenbanksystems eingegeben. So werden Zahlen oft ohne, Text mit Anführungszeichen und Datumswerte im Format "Jahr-Monat-Tag" (2023-08-15) angegeben. Bei den Bedingungen können folgende Operatoren verwendet werden:

Bei den Bedingungen können folgende Operatoren verwendet werden:

Operator	Erklärung	Datentypen
=	ist gleich	Alle Datentypen
<>	ist ungleich	Alle Datentypen
>	ist größer	Zahl, Datum
<	ist kleiner	Zahl, Datum
>=	ist größer oder gleich	Zahl, Datum
<=	ist kleiner oder gleich	Zahl, Datum

Verknüpfung von Bedingungen

Hängt ein Ergebnis von mehreren Bedingungen ab, können auch mehrere Bedingungen verknüpft werden:

AND – Alle Bedingungen müssen erfüllt sein

OR – Eine der Bedingungen muss erfüllt sein

Dabei wird das Schlüsselwort WHERE aber nur **einmal** angegeben!

Um die Abarbeitungsreihenfolge bei der Verwendung mehrerer WHERE-Bedingungen festzulegen müssen diese oft geklammert werden.

<pre>SELECT * FROM ARTIKEL WHERE Listenpreis < 10 AND Kategorie = „Baumarkt“ OR Kategorie = „Elektro“</pre>	Gibt die Datensätze der Tabelle ARTIKEL aus, die sowohl billiger als 10 € als auch in der Kategorie Baumarkt gelistet sind oder die Kategorie Elektro (ohne Preiseinschränkung) aufweisen.
<pre>SELECT * FROM ARTIKEL WHERE Listenpreis < 10 AND (Kategorie = „Baumarkt“ OR Kategorie = „Elektro“)</pre>	Gibt die Datensätze der Tabelle ARTIKEL aus, die billiger als 10 € und zusätzlich in der Kategorie Baumarkt oder Elektro gelistet sind.

Ähnliche Werte finden

Ist beispielsweise die genaue Schreibweise einer Artikelbezeichnung nicht bekannt, kann durch die Verwendung des Schlüsselworts LIKE auch nach Datensätzen gesucht werden, die eine bestimmte Zeichenfolge enthalten:

<pre>SELECT Bezeichnung FROM ARTIKEL WHERE Bezeichnung LIKE "%bohrer"</pre>	Die Ergebnistabelle enthält alle Artikelbezeichnungen, die mit "bohrer" enden, z. B. Holzbohrer, Eisenbohrer
---	--

Der Platzhalter % steht dabei für kein, eines oder mehrere Zeichen. Zusätzlich zu oben kann er folgendermaßen platziert werden: "boh%rer", "bohrer%", "%bohrer%"

Ergebnisse berechnen

Werte aus der Datenbank können auch direkt bei der Abfrage berechnet werden. So können die Artikel gleich mit den Bruttopreisen ausgegeben werden, obwohl die Listenpreise in der Tabelle ARTIKEL netto gespeichert werden:

```
SELECT ID, Bezeichnung,  
Listenpreis * 1.19 AS  
Bruttopreis  
FROM ARTIKEL
```

Die Ergebnistabelle weist die Spalten ID, Bezeichnung und Bruttopreis auf.

Mögliche mathematische Operationen sind hier + , - , / und *. Bei Dezimalzahlen wird anstelle des "," ein "." angegeben. Für diese berechneten Werte bietet es sich an, der Spalte in der Ergebnistabelle einen neuen Namen zu geben. Dies geschieht hier mit `AS Bruttopreis`. Dieser Bezeichner der neuen Spalte wird direkt hinter die Berechnung geschrieben und darf keine Leerzeichen enthalten.

Ergebnistabelle sortieren

Um die Ergebnisse in alphabetischer Reihenfolge, nach einer Nummerierung oder einem Datum zu ordnen kann die Ergebnistabelle nach einem Kriterium oder mehreren Kriterien sortiert werden. Standardmäßig wird aufsteigend sortiert, was auch mit der Angabe von `ASC` hinter den Attributen, nach denen sortiert werden soll, in der Zeile des Schlüsselworts `ORDER BY` erreicht wird. `DESC` sortiert absteigend. Dabei stehen die Abkürzungen für `ascending` (deutsch aufsteigend) und `descending` (deutsch absteigend).

Ein alphabetische Kundenliste kann mit folgender Abfrage erstellt werden:

```
SELECT Nachname, Vorname  
FROM KUNDE  
ORDER BY Nachname, Vorname
```

Die Ergebnistabelle enthält eine alphabetisch zuerst nach Nachname und dann nach Vorname sortierte Kundenliste.

Bearbeiter

2.2.3 Wie werden Datenbanken verwendet?

Unternehmen verwenden Datenbanken um gezielt Informationen abzurufen. Sind die Tabellen angelegt und die Daten in die Datenbank eingepflegt, können mithilfe von SQL-Abfragen gewünschte Ergebnisse erstellt werden. Aber wie können aus einem Datenbanksystem die passenden Datensätze ausgelesen oder sogar Berechnungen durchgeführt werden?

M1 Listen, Listen, Listen



M2 Meine Bestellungen

Mazon.de Alle ▾ Suche Einkaufswagen ¹

Mein Konto > Meine Bestellungen

Meine Bestellungen Suche nach allen Bestellungen Los

55 Bestellungen aufgegeben Alle Bestellungen anzeigen Offene Bestellungen anzeigen

BESTELLUNG AUFGEGEBEN	SUMME			
1. Januar 2022	EUR 63,02	BESTELLN.	<input type="text" value="3"/>	Bestelldetails anzeigen
26. Dezember 2021	EUR 4,95	BESTELLN.	<input type="text" value="5"/>	Bestelldetails anzeigen

Bearbeiter

M3 Artikelsuche

Hauptmenü ONLINE SHOP

in Kategorien

- Büro
- Lebensmittel
- Werkstatt
- ...

bis Preis

- Bis 10 €
- Bis 20 €
- Bis 50 €

Bearbeiter

Siehe **Methode S. 97** und **SQL-Datei**



Mediencode 82111-15

Aufgaben

- 1** Verfassen Sie Datenbankabfragen, die Listen (**M1**) für die unterschiedlichen Interessensgruppen (Abteilungen im Unternehmen und Kunden) zurückliefern. Geben Sie dabei nur relevante Spalten an, verwenden Sie Berechnungen wo nötig und sortieren Sie Ihre Ergebnistabelle passend:
 - a. Artikelliste mit Informationen über die Produkte
 - b. Auflistung aller Ansprechpartner mit zugehörigen Unternehmen
 - c. Kundenadressliste
 - d. Liste aller unbezahlten Bestellungen
 - e. Auflistung aller nicht verfügbaren Produkte
 - f. Liste mit Gewinnspanne aller Produkte
 - g. Preisliste der Artikel für Geschäftskunden (netto)
 - h. Preisliste der Artikel für Privatkunden (brutto)
- 2** Verfassen Sie Abfragen, die beim Klick auf die vier Buttons in **M2** ausgeführt werden.
- 3** Entwickeln Sie eine Datenbankabfrage, die sich hinter der Suche in **M3** verbirgt. Erweitern Sie dazu die WHERE-Bedingung schrittweise.
 - a. Suche mithilfe des Stichworts auch ähnlicher Begriffe
 - b. Anzeige alle Produkte der gewählten Kategorien
 - c. Anzeige aller Produkte unter dem Preislimit

2.2.4 Wie erstellt man Datenbankabfragen über mehrere Tabellen?

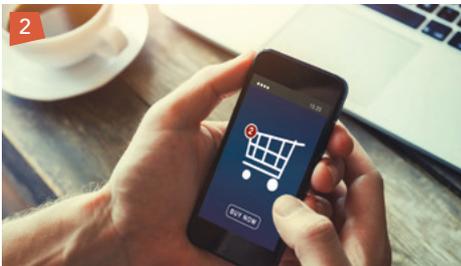
Aufgrund der Komplexität von unternehmerischen Anforderungen beinhalten Datenbanken in der Regel mehrere Tabellen. Will man umfassende Analysen des Datenbestands durchführen, so ist häufig der Zugriff auf verschiedene Tabellen nötig. Wie erstellt man also Datenbankabfragen, die sich auf mehrere Tabellen beziehen?

M1 Kunden- und Lieferantenbetreuung des Unternehmens Online-Versand²⁴

Lieferengpässe, verzögerte Bestellungen, offene Rechnungen oder einfach Kontaktpflege, immer wieder müssen die Mitarbeiter des Versandhandels in Kontakt mit ihren Kunden und Lieferanten treten. Meist sind sie dabei auf Informationen angewiesen, die auf mehrere Tabellen verteilt sind.



Kunden- und Lieferantenbetreuung



Online-Einkauf



Warenentladung

M2 Personalisierte Werbung

Das beliebteste Haustier der Deutschen ist die Katze - Aber warum ist das so?

Galerie

Kaum Zeit zum Kochen? So zaubern Sie aus nur **5 Zutaten** tolle Gerichte.

Wohnen

Heizkosten sparen: Wann lohnt sich ein Fensteraustausch?

Sichere dir **20 %** Rabatt bei deinem Einkauf ab 99,- € - **Jetzt bestellen!**

**NUR HEUTE
SONDER-
ANGEBOT!**

Kennst du deinen Partner wirklich?
Testet euch jetzt in nur 16 Fragen!



Die **7 besten Zimmerpflanzen** für Anfänger - Hol dir die Natur in dein Zuhause!

Anzeige

Handytarif zu teuer? Wir helfen dir! Jetzt auf bestertarif21.de!

Anzeige

Anzeige • <https://www.mode-onlineshop.de> 



32,99 €
22,99 Euro

MILE
Kapuzenpullover

 kostenloser Versand

Anzeige • <https://www.mode-onlineshop.de> 



21,00 €
12,99 Euro

MILE
Wollmütze

 kostenloser Versand

Bearbeiter

Die notwendigen SQL-Befehle werden auf der nachfolgenden Methodenseite erklärt.

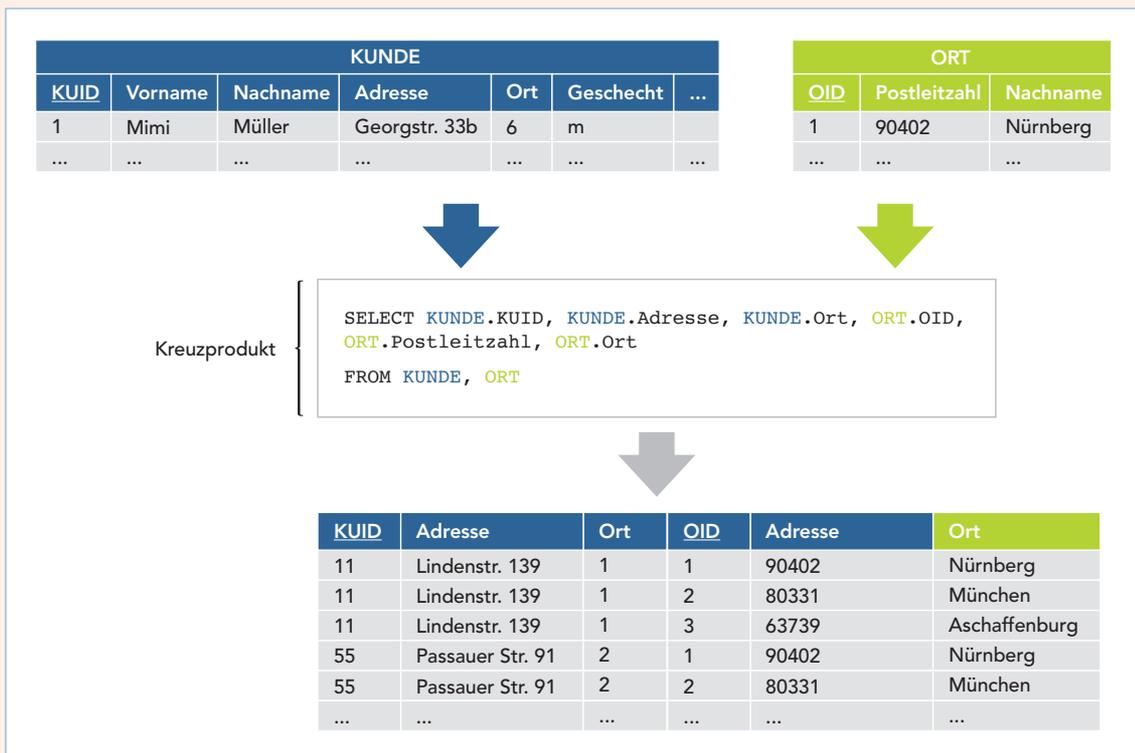
Aufgaben

- 1 Nennen Sie Attribute und die jeweils zugehörige Tabelle (Kap. 2.2.2 M2), die die Mitarbeiter zur Kunden- und Lieferantenbetreuung benötigen könnten (M1).
- 2 Erstellen Sie Datenbankabfragen zu den folgenden wiederkehrenden Aufgaben, die die Mitarbeiter bei der Betreuung von Kunden und Lieferanten immer wieder unterstützen können (M1).
 - a) Allen Lieferanten müssen auf dem Postweg neue Vertragsunterlagen zugestellt werden. Erstellen Sie eine Liste mit den Namen und den Adressen aller Lieferanten.
 - b) Ein Kundendienstmitarbeiter soll Kunden, die ihre Bestellung noch nicht bezahlt haben, per E-Mail anmahnen. Stellen Sie ihm die entsprechenden Informationen zusammen.
 - c) Zur Betreuung der Lieferanten erhalten Außendienstmitarbeiter bestimmte Regionen, in denen Lieferanten ansässig sind, als Zuständigkeitsbereich. Erstellen Sie eine Liste mit den Adressen, Ansprechpartnern und Kontaktdaten der Lieferanten für den Außendienstmitarbeiter in der Region Nürnberg/Erlangen/Fürth.
 - d) Der Artikel mit der ID 25 muss nachbestellt werden. Verschaffen Sie sich einen Überblick über die Namen und die Kontaktdaten der entsprechenden Lieferanten.
 - e) Die Firma „Winter ProX GmbH“ kann wegen Materialengpässen nicht liefern. Erstellen Sie eine Liste der Artikel, die nun anderweitig besorgt werden müssen.
- 3 Entwickeln Sie passende Abfragen, die dem Unternehmen Ergebnistabellen ausgeben, die für konkrete personalisierte Werbemaßnahmen verwendet werden können (M2). Achten Sie dabei darauf, dass die Ergebnistabelle nur Spalten und Datensätze enthält, die für den speziellen Werbezweck geeignet sind.

Mehrere Tabellen verknüpfen

SQL-Abfragen aus verschiedenen Tabellen

SQL-Abfragen können Attribute aus verschiedenen Tabellen enthalten. Die verwendeten Tabellen müssen dabei im FROM-Teil durch Komma getrennt genannt werden. Da es bei mehrfach verwendeten Attributbezeichnern zu Verwechslungen kommen kann, sollte jedem Attribut im SELECT-Teil der Tabellename vorangestellt werden.



SQL-Abfrage mit Kreuzprodukt
 Bearbeiter

Zur Vereinfachung können für die einzelnen Tabellen Abkürzungen verwendet werden, z. B.

```
SELECT k.KID, ...
FROM KUNDE k, ...
```

Das Kreuzprodukt zweier Tabellen entsteht, indem jeder Datensatz der ersten Tabelle mit jedem Datensatz der zweiten Tabelle kombiniert wird. Die Ergebnistabelle enthält dabei auch Datensätze, die keinen Sinn ergeben.

Auswahl der sinnvollen Datensätze

Aus der Ergebnistabelle des Kreuzprodukts müssen nun die sinnvollen Datensätze mit Hilfe einer **Joinbedingung** herausgefiltert werden. Man sucht die Datensätze aus, deren Werte in Primär- und Fremdschlüsselattribut übereinstimmen. Dazu setzt man im WHERE-Teil den **Fremdschlüssel** der einen Tabelle mit dem zugehörigen **Primärschlüssel** der anderen Tabelle gleich.

KUID	Adresse	Ort	OID	Postleitzahl	Ort
11	Lindenstr. 139	1	1	90402	Nürnberg
11	Lindenstr. 139	1	2	80331	München
11	Lindenstr. 139	1	3	63739	Aschaffenburg
55	Passauer Str. 91	2	1	90402	Nürnberg
55	Passauer Str. 91	2	2	80331	München
...



```
SELECT KUNDE.KUID, KUNDE.Adresse, KUNDE.Ort, ORT.OID
      ORT.Postleitzahl, ORT.Ort
FROM KUNDE, BESTELLUNG
WHERE KUNDE.Ort = ORT.OID } Joinbedingung
AND ...
```



KUID	Adresse	Ort	OID	Postleitzahl	Ort
11	Lindenstr. 139	1	1	90402	Nürnberg
55	Passauer Str. 91	2	2	80331	München
...

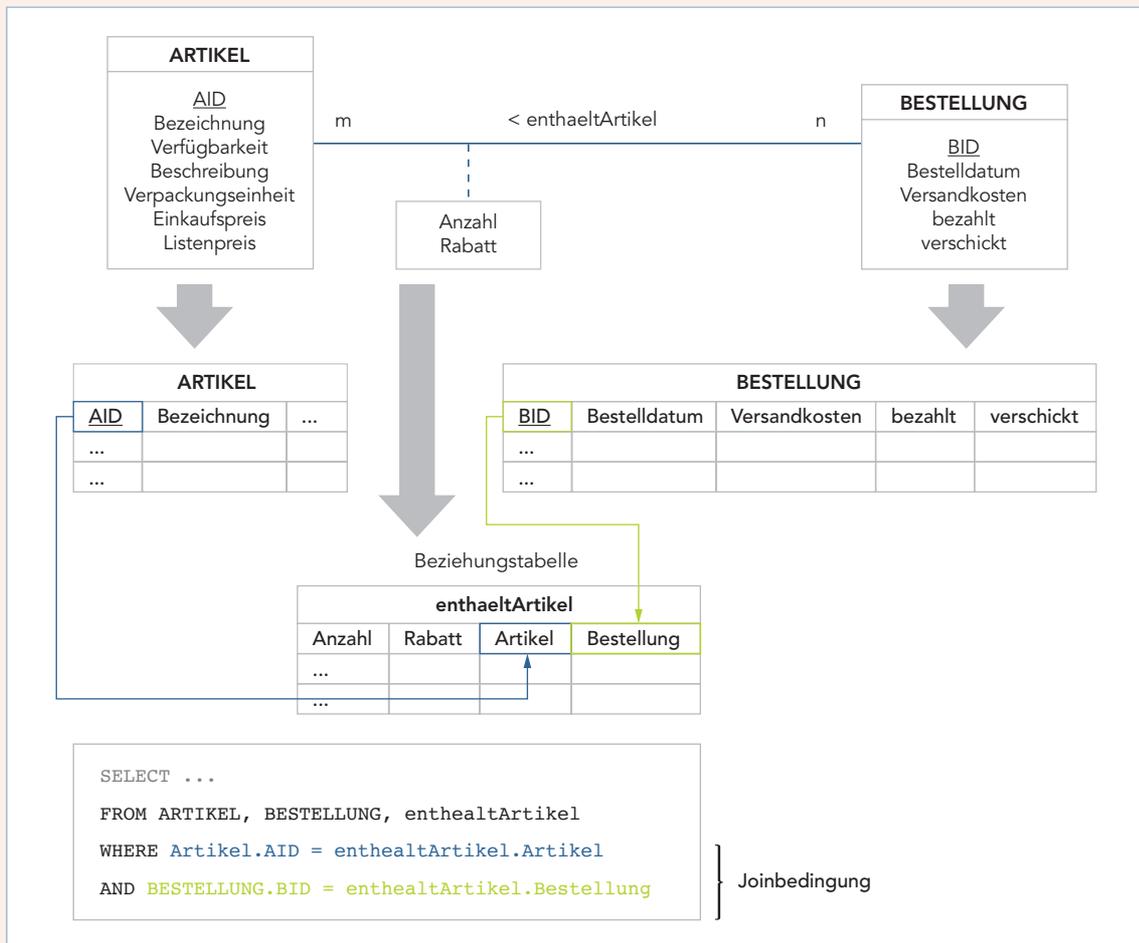
SQL-Abfrage mit Joinbedingung

Bearbeiter

Die Ergebnistabelle enthält nun lediglich zusammengehörige Datensätze. Im WHERE-Teil können weitere Bedingungen zum Filtern der Datensätze mit Hilfe von AND bzw. OR hinzugefügt werden.

Verknüpfung bei mehr als zwei Tabellen

Möchte ein Mitarbeiter wissen, welche Artikel in einer Bestellung enthalten sind, müssen drei Tabellen miteinander verknüpft werden. Die Joinbedingung besteht dabei aus zwei Teilen.



Joinbedingung bei m:n-Beziehung

Bearbeiter

Durch die Verknüpfung mehrerer WHERE-Statements mit Hilfe von AND können beliebig viele Tabellen miteinander verknüpft werden. Zur Verknüpfung von **n** Tabellen werden dabei **(n-1)** WHERE-Statements in der Joinbedingung benötigt.

2.2.5 Exkurs: Wie kann man mit Hilfe von Aggregatfunktionen und der Gruppierung von Datensätzen Datenbankabfragen verfeinern?

Um den Erfordernissen eines Unternehmens gerecht zu werden, genügt es häufig nicht, in einer Datenbank gespeicherte Daten nach geeigneten Kriterien abzufragen. Immer wieder besteht die Notwendigkeit, Daten zu vergleichen, Mittelwerte zu bilden, Ausreißer zu ermitteln und anhand von Kriterien Gruppen zu bilden. Wie kann man Datenbankabfragen dahingehend präzisieren?

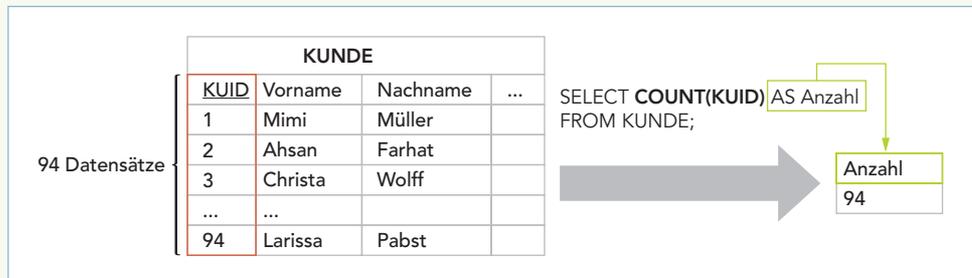
M1 Aggregatfunktionen

Befehl	Erklärung
MAX(Attribut)	Gibt den größten Wert der Spalte aus.
MIN(Attribut)	Gibt den kleinsten Wert der Spalte aus.
SUM(Attribut)	Gibt die Summe aller Werte der Spalte aus.
AVG(Attribut)	Gibt das arithmetische Mittel aller Werte der Spalte aus.
COUNT(Attribut)	Gibt die Anzahl an Zeilen in der Spalte aus .

Um die Zeilen einer Tabelle zu zählen, kann anstelle von COUNT(<Spalte>) auch COUNT(*) verwendet werden.

Um die Spaltenüberschriften der Ergebnistabelle sinnvoll zu beschriften, kann im SELECT-Teil mit Hilfe des Schlüsselworts **AS** ein eigener Spaltenname vergeben werden.

Beispiel:



SQL-Abfrage mit Joinbedingung
Bearbeiter

M2 Sollte das Sortiment ergänzt werden?



Bezeichnung	AID
Hygieneartikel	1
Spielwaren	2
Spielwaren	3
Bekleidung	4
Bekleidung	5
Spielwaren	6
Bekleidung	7
Bekleidung	8

Bezeichnung	Anzahl
Hygieneartikel	1
Spielwaren	3
Bekleidung	4
...	...

Gruppierung von Datensätzen
 Bearbeiter

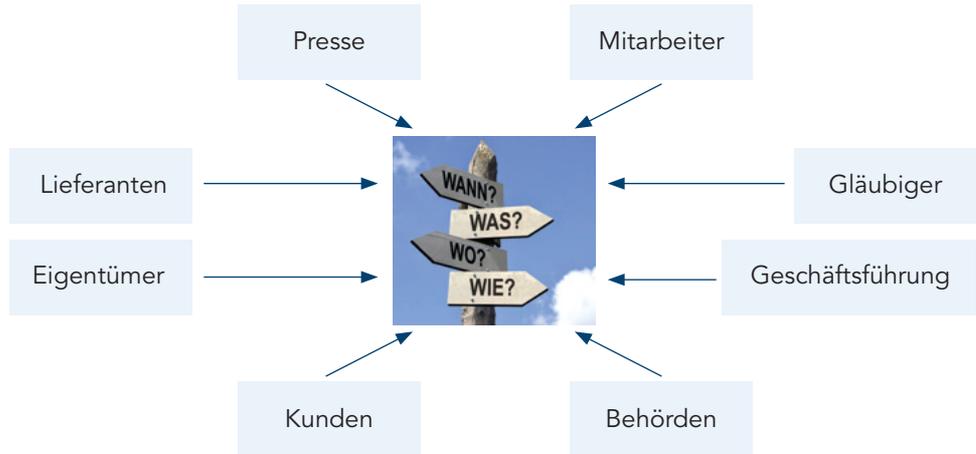
Aufgaben

- 1 Erstellen Sie Datenbankabfragen unter Verwendung von Aggregatfunktionen (M1). Verwenden Sie dazu die Datenbank des OnlineVersand24 (vgl. S. 95, M2)
 - a) Zeigen Sie an, wie viele Artikel der OnlineVersand24 aktuell im Sortiment hat.
 - b) Bestimmen Sie den Preis des teuersten aktuell verfügbaren Artikels.
 - c) Bestimmen Sie die durchschnittlichen Versandkosten aller Bestellungen des Unternehmens OnlineVersand24.
 - d) Zeigen Sie an, welchen Rabatt der OnlineVersand24 im Jahr 2020 durchschnittlich vergeben hat.
- 2 Erklären Sie die Funktionsweise von GROUP BY (M2).
- 3 Erstellen Sie Datenbankabfragen, welche die Mitarbeiter der Kundenbetreuung bei der Identifikation der wichtigsten Kunden unterstützen. Geben Sie dabei nur relevante Spalten an (M1, M2).
 - a) Zeigen Sie für jede KundenID (KUID) an, wie viele Bestellungen der Kunde bereits getätigt hat.
 - b) Erstellen Sie einen Überblick über die IDs aller Bestellungen (BID) sowie deren Gesamtpreis. Ordnen Sie die Tabelle absteigend nach der Höhe des Gesamtpreises.
 - c) Geben Sie für jeden Kunden (KUID anzeigen) den Gesamtwert seiner bisher getätigten Bestellungen an.
 - d) Zeigen Sie für jeden Kunden (ID anzeigen), wie viel Gewinn der OnlineVersand24 durch dessen Bestellungen erzielt hat. Ordnen Sie die Liste absteigend nach dem Gewinn.
- 4 Entwickeln Sie Datenbankabfragen, die einen Überblick über die wichtigsten Lieferanten verschaffen (M1, M2).

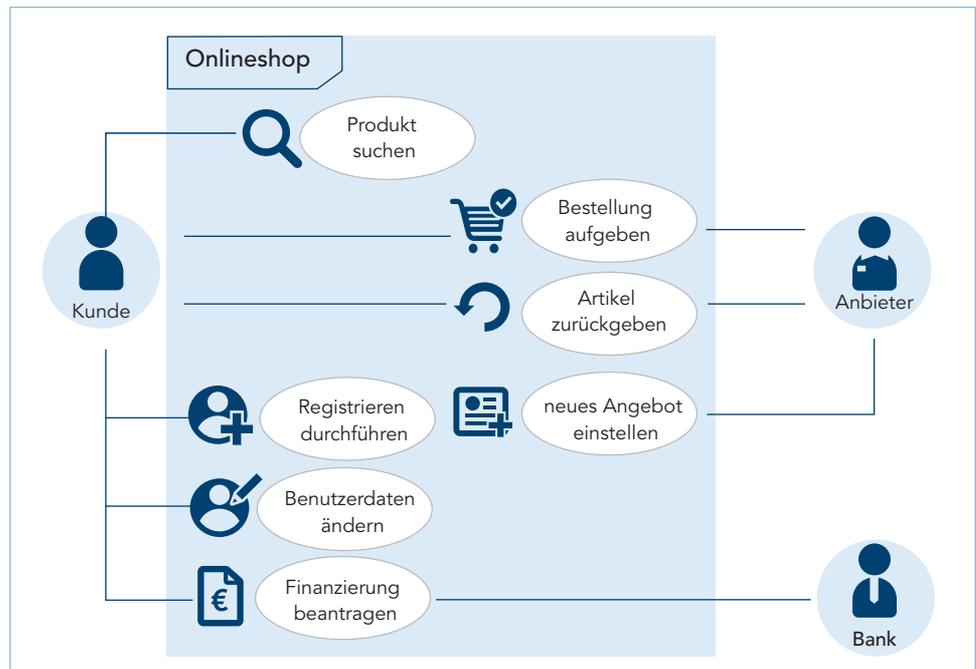
2.2.6 Wie werden Datenbanken verwendet?

Datenbanken werden für bestimmte Einsatzzwecke entworfen und umgesetzt. Um die benötigten Informationen anschließend zu erhalten, müssen die richtigen Abfragen formuliert werden. Aber wie gehe ich dabei vor?

M1 Wer benötigt Informationen?



M2 Welche Anwendungsfälle sollen abgedeckt werden?



Use-Case-Diagramm eines Onlineshops

Bearbeiter

M3 Welche Daten darf ich weitergeben?

Die „informationelle Selbstbestimmung“ ist im Grundgesetz nicht ausdrücklich erwähnt, wird aber als eigenes, aus dem Bereich der Persönlichkeitsrechte hervorgegangenes Grundrecht behandelt. [...] In die Persönlichkeitssphäre des Einzelnen darf nur dann eingegriffen werden, wenn bestimmte Voraussetzungen erfüllt sind. Das Recht auf informationelle Selbstbestimmung wird verletzt, wenn ohne gesetzliche Ermächtigung oder persönliche Zustimmung des Betroffenen Daten erhoben, bearbeitet oder weitergegeben werden. [...]

15 Datenschutz in der Wirtschaft

[...] Firmen und IT-Dienstleister kommen auf unterschiedlichen Ebenen ihrer Tätigkeit mit persönlichen Daten und damit auch mit den Datenschutzregelungen in Berührung. Die Vorschriften des Datenschutzes müssen beim Umgang mit sensiblen Kundendaten, beispielsweise Kontendaten oder Adressdaten, ebenso wie

beim Umgang mit Daten von Geschäftspartnern und Mitarbeitern beachtet werden. [...]

Datenschutz Beispiel

[...] Zur Abwicklung eines Versandgeschäfts muss die Adresse des Bestellers an den Zustelldienst weitergegeben werden. Dort muss allerdings klargestellt werden, dass die Adressdaten nur für diesen einen Vorgang benutzt werden dürfen.

Nach Zustellung sind die Daten sofort zu löschen. Zur Vereinfachung können sich Unternehmer vom Betroffenen die Zustimmung zur Datenerhebung und -verarbeitung einholen. Die Zustimmung gilt, wie eine gesetzliche Bestimmung, als Ermächtigung, die allerdings auch zurückgenommen werden kann.

Datenbanken-verstehen.de: Datenschutz. In: www.datenbanken-verstehen.de, Abruf am 22.03.2023



Projektzieldefinition

Sie erstellen Datenbankabfragen zur Ihrem Datenbankprojekt.



Aufgaben zum Projekt

- 1 Beschreiben Sie, welche Informationen Interessengruppen aus Ihrer Datenbank benötigen (M1) und in welchen Tabellen diese zu finden sind.
- 2 Formulieren Sie zu jedem Anwendungsfall aus dem Use-Case-Diagramm Ihres Datenbankprojekts (vgl. Kapitel 2.1.3, Aufgabe 5) konkrete Fragen, deren Beantwortung der beteiligte Akteur erwartet (M2).
- 3 Entwerfen Sie Datenbankabfragen, die die in Aufgabe 2 gestellten Fragen beantworten.
- 4 Beurteilen Sie, ob bei der Beantwortung der in Aufgabe 2 gestellten Fragen die Zustimmung von Betroffenen eingeholt werden muss (M3).

2.2.7 Wie verhalte ich mich vorausschauend im Umgang mit meinen Daten?

„Herzlichen Glückwunsch, Sie haben gewonnen!“, „Sie wurden ausgewählt!“ Solche oder ähnlich lautende E-Mails hat wohl schon jeder einmal in seinem Posteingang gefunden und weiß, dass er mit derartigen Nachrichten mit Vorsicht umgehen sollte, da sich dahinter meist eine Betrugsmasche verbirgt, um unsere persönlichen Daten abzugreifen. Aber auch seriöse Unternehmen sind daran interessiert, möglichst viele Daten über ihre Kunden zu gewinnen. Daher stellt sich die Frage, wie derartige Kundenprofile erstellt werden und wie man sich dahingehend vorausschauend verhalten kann?

M1 Möglichkeiten der Datenerhebung

1 **3 einfache Schritte:**

- 1 Melden Sie sich **kostenlos** an.
- 2 Nehmen Sie an **Umfragen** teil.
- 3 Verdienen Sie **Prämien**.

Online-Umfrage



Versicherungsabschluss



Kontaktlos Bezahlen

4

Hallo, User
Konto und Listen ▾

- Mein Konto
- Meine Bestellungen
- Wunschliste

Dropdown-Menü

5

Jobportal

6

Kundenrezensionen
★★★★☆ 4,6 von 5 ▾
32 Sternebewertungen

5 Sterne	<div style="width: 75%;"></div>	75%
4 Sterne	<div style="width: 13%;"></div>	13%
3 Sterne	<div style="width: 6%;"></div>	6%
2 Sterne	<div style="width: 6%;"></div>	6%
1 Sterne	<div style="width: 0%;"></div>	0%

Kundenrezension

M2 Aus Kartendaten Profile schmieden

Forscher der Universität von Massachusetts haben es ans Licht gebracht: Nichts ist leichter, als aus Kreditkartendaten Kundenprofile zu ziehen. [...]

5 Die Wissenschaftler hatten dazu einen Satz Kreditkartendaten aus 10.000 Geschäften von 1,1 Millionen Menschen im Zeitraum von drei Monaten analysiert. Die Informationen enthielten weder Namen noch Karten- oder Kontonummern, sondern nur Tag, Ort und die Höhe der Zahlungen, die von einem bestimmten Konto getätigt wurden. [...] Erkenntnisse über Bezahlvorgänge konnten beispielsweise aus Social-Media-Daten wie [...] [Kurznachrichten] oder einer Bewertung auf Internet-Plattformen ausgelesen werden. Dann glichen sie vier verschiedene Bezahlvorgänge, die ihnen durch das Nutzungsverhalten der Kunden bekannt waren, mit dem anonymen Datensatz ab. [...]

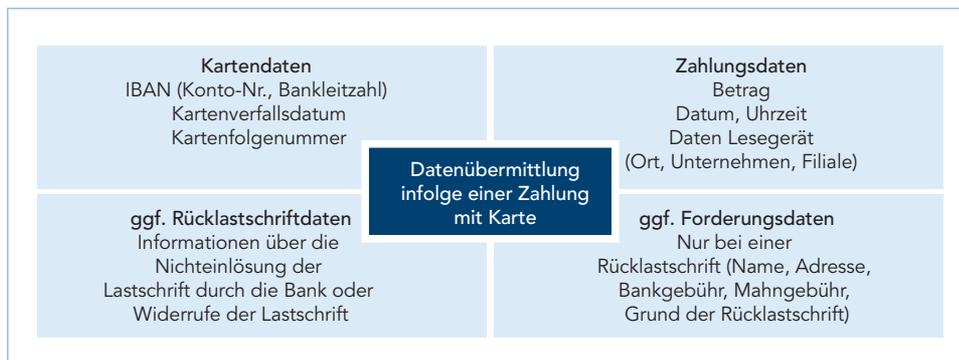
An einem Beispiel machten Forscher [...] deutlich, wie einfach Kreditkartendaten zu Profilen von konkreten Personen hinführen können: Wussten die Forscher beispielsweise, dass ein bestimmter Kunde immer montags seinen Kaffee mit Kreditkarte zahlt, dienstags im Supermarkt eingekauft hat sowie freitags in einem Bekleidungsgeschäft war, konnten sie diese Person in rund 90 Prozent der

Fälle in dem getesteten Datensatz wiederfinden. Um die Nutzer identifizieren zu können, griffen die Forscher auf die Zeitstempel der Transaktionen zurück und führten diese mit weiteren Daten wie etwa dem Standort der Kunden zusammen. [...]

Für Banken und Kreditkartenemittenten ist dies ebenso wie für die Bankkunden eine wertvolle Erkenntnis. Denn mit dem Experiment belegen die Datenwissenschaftler zum einen, dass Anonymisierungsverfahren, wie sie beispielsweise Kreditinstitute oder auch Krankenkassen derzeit einsetzen, trotz Datenschutz keine absolute Anonymität bieten. Zum anderen bieten solche Meta-Finanzdaten, die digitale Zahlungen beinhalten, den Kreditinstituten ein enormes Potenzial. Denn sie enthalten wertvolle Metadaten zum Individualverhalten der Kartennutzer. Auf der anderen Seite werden sie längst zur Betrugsprävention, im Scoring und zur Analyse des Einkaufsverhaltens der Kunden ausgewertet. Der bisherige Datenschutz, so bemängeln die Forscher, schütze zwar personenbezogene Daten, das gelte aber nicht für anonymisierte Daten.

Krah, Eva-Susanne: Aus Kartendaten Profile schmieden. In: www.springerprofessional.de, 05.02.2015

M3 Datenübermittlung infolge einer Zahlung mit Karte



Bearbeiter, basierend auf: C&A: Zahlverfahren SEPA Lastschrift. In: www.c-and-a.com, Abruf am 15.03.2023

Customer Journey

Die „Reise des Kunden“ umfasst den ersten bis letzten Kontakt des Verbrauchers mit dem Unternehmen. Sie dient der Marketingabteilung zur besseren Abschätzung des Kundenverhaltens und seiner Bedürfnisse, um damit eine kundenzentrierte Marketingstrategie aufzubauen.

M4 Mit Data Driven Marketing auf Nutzerbedürfnisse reagieren

Durch Social-Media-Kanäle und die hohe Verbreitung verschiedener Endgeräte stehen heute mehr Daten denn je zu Verfügung. Im Zeitalter von Big Data gilt es für Unternehmen, die Informationen, die User online hinterlassen, zu sammeln und gezielt in ihrer Kommunikation zu nutzen. [...] Beim Data Driven Marketing sollen aus Daten Erkenntnisse gewonnen werden, mit denen Marketingaktionen besser an die Bedürfnisse und Wünsche der Kunden angepasst werden können. Es geht darum, das Kundenverhalten besser zu verstehen und die Marketing-Strate-

gie darauf aufzusetzen. Dazu werden Daten an sämtlichen Touchpoints der Customer-Journey (z. B. Social Media, Online-Shop, Unternehmenswebsite) gezielt gesammelt, um daraus Maßnahmen abzuleiten und diese zu optimieren. Die Herausforderung besteht darin, die vielen Kundeninformationen, die online und offline gesammelt werden, zu verknüpfen. [...]

Textbroker – Sario Marketing: Data Driven Marketing. In: www.textbroker.de, Abruf am 14.02.2023

M5 Berührungspunkte des Kunden mit dem Unternehmen



awareness
engl.: Aufmerksamkeit, Wahrnehmung

Nach: Wiederer, Christian A.: Die wichtigsten Bestandteile der Customer Journey. In: www.five8.de, 12.02.2021

Aufgaben

- 1 Analysieren Sie, welche Daten bei dem jeweiligen Angebot erhoben werden und wer an diesen Daten Interesse haben könnte (M1).
- 2 Finden Sie Beispiele, bei denen durch die Kombination unterschiedlicher Datensätze, ggf. auch aus verschiedenen Unternehmen, neue Erkenntnisse über Kunden gewonnen werden können (M1).
- 3 Stellen Sie dar, wie auch aus vermeintlich anonymisierte Daten Kundenprofile erstellt werden können (M2).
- 4 Erklären Sie, welche Erkenntnisse der Kartenanbieter aus den übermittelten Daten bei einer Kartenzahlung über den jeweiligen Kunden gewinnen kann, die für andere Unternehmen von Interesse sein könnten (M3).
- 5 Erläutern Sie, welche verschiedenen Informationen aus sozialen Netzwerken kombiniert für Marketingzwecke von Interesse sind (M4).
- 6 Zeigen Sie an Beispielen auf, welche Informationen das Unternehmen über einem Kunden an den verschiedenen Touchpoints sammeln kann und wie diese in den verschiedenen Phasen des Kaufprozesses hilfreich sein könnten (M5).
- 7 Sammeln Sie aus Ihrer persönlichen Social-Media-Nutzung, der Verwendung von Bezahlendiensten, der Suche in Online-Shops und abgeschlossenen Bestellungen Informationen, die für Marketing-Zwecke verwendet werden können (M4).
- 8
 - a) Analysieren Sie die Datenbank des Musterunternehmens Onlineversand24 und erläutern Sie, aus welchen Angaben bereits sinnvolle Kundenprofile für Marketingzwecke erstellt werden könnten.
 - b) Entwerfen Sie für das Musterunternehmens Onlineversand24 eine Kundenbefragung, die in Kombination mit anderen Informationen aus der Datenbank dabei hilft ein detailliertes Kundenprofil zu erstellen.
- 9 Diskutieren Sie, inwiefern Maßnahmen zum Schutz persönlicher Daten im Alltag noch möglich sind.

Fachwissen im Zusammenhang – Lernbereich 2.2

<p>Logisches Modell ↳ Methode S. 86</p>	<p>Da relationale Datenbanken Daten in Tabellenform speichern, muss ein entworfenen semantisches Beziehungsmodell daher informationsverlustfrei in ein Tabellenmodell, das sogenannte logische Modell, überführt werden. Das logische Modell bildet die gleichen Strukturen, Beziehungen, Attribute, etc. ab wie das semantische Modell – allerdings dargestellt in einzelnen Tabellen.</p> <p>Das logische Modell bildet systemunabhängig die Grundlage für eine Implementierung der Datenbank und ist daher von großer Bedeutung auf dem Weg zu einem funktionierenden Datenbanksystem.</p>
<p>Spickzettel: Vom semantischen zum logischen Modell ↳ Methode S. 86</p>	<div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="border: 1px solid gray; padding: 5px; width: 30%; background-color: #f0f0f0;"> <p>Schritt 1: Jede Klasse wird zu einer Tabelle.</p> </div> <div style="border: 1px solid gray; padding: 5px; width: 30%; background-color: #f0f0f0;"> <p>Schritt 2: Ergänze in den passenden Tabellen die 1:n-Beziehungen.</p> </div> <div style="border: 1px solid gray; padding: 5px; width: 30%; background-color: #f0f0f0;"> <p>Schritt 3: Jede m:n-Beziehung wird zu einer eigenen Tabelle.</p> </div> </div>
<p>Primärschlüssel ↳ Methode S. 86</p>	<p>Als Primärschlüssel bezeichnet man im logischen Modell ein Attribut einer Klasse, durch dessen Werte sich alle Objekte dieser Klasse eindeutig identifizieren und damit voneinander unterscheiden lassen. Meistens verwendet man im logischen Modell dazu eine automatisierte fortlaufende Nummerierung als Objekt-ID. Den Primärschlüssel kennzeichnet man in der Tabelle durch eine Unterstreichung.</p>
<p>Fremdschlüssel ↳ Methode S. 87</p>	<p>Wird ein Primärschlüssel einer Tabelle als Verweis in einer anderen Tabelle verwendet (z. B. aufgrund einer 1:n- oder m:n-Beziehung), so bezeichnet man diesen in der „fremden“ Tabelle als Fremdschlüssel. Jeder Fremdschlüssel in einer Tabelle stellt also die Verbindung zu einem Primärschlüssel einer anderen Tabelle her.</p>
<p>Tabellenschreibweise ↳ Methode S. 89</p>	<p>Um ein logisches Modell knapp und präzise darzustellen, bietet sich folgende Kurzschreibweise der einzelnen Tabellen an: Tabellenname(<u>Spaltenbezeichner-1</u>, <u>Spaltenbezeichner-2</u>, ..., <u>Spaltenbezeichner-x</u>) Hinter dem Tabellennamen folgen in einer Klammer durch Kommata getrennt die einzelnen Spaltenbezeichner, der Primärschlüssel wird unterstrichen, Fremdschlüssel werden überstrichen. Schreibt man alle Tabellen des logischen Modells untereinander, so hat man eine übersichtliche und klare Darstellung, die im Folgenden beim Implementieren der Datenbank hilfreich ist.</p>

Merkmale eines Datenbankmanagement-systems

➔ Methode
S. 90

Das Datenbankmanagementsystem (DBMS) macht eine **einheitliche Datenverwaltung** aller von den Anwendungen benötigten Daten überhaupt erst möglich. Somit wird jedes Datenelement in der Datenbank an nur einer Stelle gespeichert. Darüber hinaus wird meist eine entsprechende Benutzeroberfläche und eine **Datenbanksprache** für unterschiedliche Zwecke zur Verfügung gestellt, damit die verschiedenen Benutzergruppen Zugriffs- und Modifikationsmöglichkeiten auf die Datenbasis haben. Die Datenintegrität, also die Korrektheit der Dateninhalte sowie die korrekte Ausführung von Änderungen, wird durch Setzen der Beziehungen im Designer automatisch gewährleistet. Auch für die **Datensicherung** zeichnet sich das DBMS verantwortlich: Nach Hardware- oder Softwarefehlern muss der korrekte Datenbestand wiederhergestellt werden können. Durch festgelegte Regeln können nicht autorisierte Zugriffe auf Daten vermieden werden, d. h. die Datenzugriffskontrolle wird vom DBMS verwaltet.

Architektur eines Datenbank-systems

➔ Methode
S. 90

Es gibt unterschiedliche Anwendergruppen, die mit einem Datenbankverwaltungssystem über **Benutzerschnittstellen** in Berührung kommen. Die Definition der Schnittstelle hängt von der Erfahrung und der Verantwortlichkeit der jeweiligen Benutzer ab. „Einfache“ Endnutzer können meist über menügesteuerte Anwendungen mit grafischer Benutzeroberfläche (z. B. Sitzplatzreservierungen, Einkäufe über Onlineshops, Anmeldung bei einem sozialen Netzwerk) mit der Datenbank kommunizieren. „Fortgeschrittene“ Anwender, die häufig wechselnde Aufgaben im Hinblick auf die Datenbasis haben, können sogenannte interaktive Anfragesprachen (z. B. SQL) verwenden. Diese können beispielsweise für die Entwicklung neuer Anwendungsprogramme verwendet werden. Auch hier sind die Rechte in der Benutzer- und Rechteverwaltung des Datenbankmanagementsystems hinterlegt. Ein Datenbankadministrator besitzt eine eigene Schnittstelle zur Veränderung der Datenbanktabellen und zur Administration von Benutzerkonten.

Definition des Datenbankschemas mit einem Datenbankmanagementsystem

➔ Methode
S. 90

Je nach DBMS kann das Datenbankschema mithilfe einer grafischen Benutzeroberfläche (siehe Methodenschritte) oder mit einer Datenbanksprache festgelegt werden:

Operation	Befehl
Erzeugen von Tabellen	CREATE TABLE Tabellename (Attribut Datentyp, ..., PRIMARY KEY (Spaltenname/n))

Ändern von Tabellen (Attribute hinzufügen, ändern, löschen)	ALTER TABLE Tabellenname ADD (Attribut Datentyp)
	ALTER TABLE Tabellenname MODIFY (Attribut Datentyp)
	ALTER TABLE Tabellenname DROP (Attribut Datentyp)
Löschen von Tabellen	DROP TABLE Tabellenname

Beim Erzeugen einer Tabelle wird zu jedem Attribut ein Wertebereich angegeben, der durch einen Datentyp definiert ist. Damit wird sichergestellt, dass beim Attribut „Kaufpreis in €“ tatsächlich nur Dezimalzahlen mit der angegebenen Anzahl an Nachkommastellen eingegeben werden können; andere Eintragungen führen zu einer Fehlermeldung. Auch die Länge kann für den Wertebereich festgelegt werden. Häufige Datentypen sind z. B.:

Datentypen	Beschreibung
INT	Integer sind ganze (negative und positive) Zahlen, z. B. 0, -14 oder 394
DECIMAL	Dezimalzahlen mit bestimmter Anzahl an Stellen vor und nach dem Komma, z. B. DECIMAL(4,2) für 3202,99
BOOLEAN	Wahrheitswerte können „ja“ oder „nein“ annehmen
VARCHAR	beliebige Zeichen, z. B. „Dr. Max Münstermann“
TEXT	Zeichenkette aus Buchstaben, z. B. „München“
DATE	Datumsangabe, z. B. 2023-02-18 für den 18. Februar 2023

Verwaltung von Daten mit einem Datenbankmanagement-system

→ Kapitel 2.2.1

Auch um Tabelleninhalte, also einzelne Datensätze, einzufügen, zu ändern oder zu löschen, wird eine grafische Benutzeroberfläche oder eine Datenbanksprache verwendet. Folgende Befehle ermöglichen die Umsetzung:

Operation	Befehl
Einfügen von Zeilen	INSERT INTO Tabellename [(Spaltennamensliste)] VALUES(Werteliste)
Ändern von Zeilen	UPDATE Tabellename SET Spaltenname = Wertausdruck, ... [Bedingung]
Löschen von Zeilen	DELETE FROM Tabellename [Bedingung]

Mittels INSERT können einzelne Zeilen eingefügt werden (VALUES-Klausel). Die Befehle UPDATE und DELETE wirken in allen Zeilen der ausgewählten Tabelle.

Zu beachten ist dabei die **referentielle Integrität (RI)**, nach der Daten über ihre Fremdschlüssel nur auf existierende Daten verweisen dürfen. D.h. ein neuer Datensatz (mit Fremdschlüssel) kann nur eingefügt werden, wenn in der referenzierten Relation ein Datensatz mit einem entsprechenden Wert im Primärschlüssel vorliegt. Das Ändern und Löschen eines Attributwertes, welcher Primärschlüssel ist, ist nur möglich, wenn zu diesem Datensatz keine abhängigen Fremdschlüssel-Datensätze in Beziehung stehen. Viele DBMS stellen die referentielle Integrität automatisch durch das vorherige Anlegen sogenannter Beziehungen sicher.

SQL Abfragen

→ Kapitel 2.2.2

Mithilfe von Abfragen, die in der Structured Query Language (SQL) verfasst werden, können Informationen aus einer Datenbank gewonnen oder Berechnungen mit diesen Daten durchgeführt werden. Dazu werden aus den Originaltabellen in der Datenbank mithilfe der Abfragen Ergebnistabellen erstellt. Dies liefert nur richtige Ergebnisse zurück, wenn die korrekte SQL-Syntax verwendet wird.

Jede SQL-Abfrage muss dabei mindestens Angaben darüber machen, aus welchen Tabellen Daten verwendet werden und welche Spalten der Originaltabelle die Ergebnistabelle beinhalten soll:

```
SELECT *
FROM ORT
```

Gibt die gesamte Tabelle ORT als Ergebnistabelle aus.

	<pre>SELECT LID FROM LIEFERANT</pre>	Liefert nur die Spalte LID der Tabelle LIEFERANT zurück.
	<pre>SELECT DISTINCT Bezeichnung FROM ARTIKEL</pre>	Gibt Bezeichnungen, die in der Tabelle ARTIKEL mehrfach vorkommen, in der Ergebnistabelle jeweils nur einmal an.
<p>Bedingungen  Methode S. 97</p>	<p>Soll die Ergebnistabelle nur ausgewählte Datensätze beinhalten, müssen Bedingungen formuliert werden. In die Ergebnistabelle werden damit nur die Datensätze übernommen, die diese Bedingungen erfüllen. Diese können unter anderem mit <, >, <=, =, >= und = angegeben werden.</p>	
	<pre>SELECT * FROM ARTIKEL WHERE Listenpreis < 100</pre>	Gibt als Ergebnistabelle alle Informationen der Artikel aus, die einen Listenpreis besitzen, der kleiner als 100 ist.
	<p>Werden mehrere Bedingungen kombiniert müssen diese oft geklammert werden, um logische Zusammenhänge festzulegen:</p>	
	<pre>SELECT * FROM ARTIKEL WHERE Listenpreis < 10 AND Kategorie = „Baumarkt“ OR Kategorie = „Elektro“</pre>	Gibt die Datensätze der Tabelle ARTIKEL aus, die sowohl billiger als 10 € als auch in der Kategorie Baumarkt gelistet sind oder die Kategorie Elektro (ohne Preiseinschränkung) aufweisen.
	<pre>SELECT * FROM ARTIKEL WHERE Listenpreis < 10 AND (Kategorie = „Baumarkt“ OR Kategorie = „Elektro“)</pre>	Gibt die Datensätze der Tabelle ARTIKEL aus, die billiger als 10 € und zusätzlich in der Kategorie Baumarkt oder Elektro gelistet sind.
	<p>Mit dem Schlüsselwort LIKE können auch Datensätze identifiziert werden, eine bestimmte Zeichenfolge enthalten.</p>	

	<p>Dabei wird der gesamte Suchbegriff in Anführungszeichen gesetzt und % verwendet um Platzhalter für mögliche weitere Zeichen anzugeben:</p> <table border="1" data-bbox="434 384 1248 603"> <tr> <td data-bbox="434 384 839 603"> <pre>SELECT Bezeichnung FROM ARTIKEL WHERE Bezeichnung LIKE "%bohrer"</pre> </td> <td data-bbox="839 384 1248 603"> <p>Gibt als Ergebnistabelle die Bezeichnungen aller Artikel aus, deren Bezeichnung mit „bohrer“ endet. Dabei spielt es keine Rolle ob sich davor keine, eines oder mehrere Zeichen befinden.</p> </td> </tr> </table> <p>Der Platzhalte % kann dabei vor, nach und zwischen Suchbegriffen verwendet werden.</p>	<pre>SELECT Bezeichnung FROM ARTIKEL WHERE Bezeichnung LIKE "%bohrer"</pre>	<p>Gibt als Ergebnistabelle die Bezeichnungen aller Artikel aus, deren Bezeichnung mit „bohrer“ endet. Dabei spielt es keine Rolle ob sich davor keine, eines oder mehrere Zeichen befinden.</p>		
<pre>SELECT Bezeichnung FROM ARTIKEL WHERE Bezeichnung LIKE "%bohrer"</pre>	<p>Gibt als Ergebnistabelle die Bezeichnungen aller Artikel aus, deren Bezeichnung mit „bohrer“ endet. Dabei spielt es keine Rolle ob sich davor keine, eines oder mehrere Zeichen befinden.</p>				
<p>Sortierung ↳ Methode S. 99</p>	<p>Die Ergebnistabelle kann anhand einer oder mehrerer Spalten sortiert werden:</p> <table border="1" data-bbox="434 811 1248 1152"> <tr> <td data-bbox="434 811 839 997"> <pre>SELECT Nachname, Vorname FROM KUNDE ORDER BY Geburtstag DESC</pre> </td> <td data-bbox="839 811 1248 997"> <p>Gibt als Ergebnistabelle alle Vor- und Nachnamen absteigend sortiert nach dem Geburtstag aus. Die Liste startet dabei bei dem jüngsten Kunden.</p> </td> </tr> <tr> <td data-bbox="434 997 839 1152"> <pre>SELECT Nachname, Vorname FROM KUNDE ORDER BY Nachname, Vorname</pre> </td> <td data-bbox="839 997 1248 1152"> <p>Gibt eine nach Nach- und Vorname geordnete Kundenliste aus.</p> </td> </tr> </table> <p>Sortiert werden kann alphabetisch, numerisch oder auch nach Datumswerten. Dabei wird die Ergebnistabelle bei einer Angabe von ORDER BY immer automatisch aufsteigend sortiert. Eine Ergänzung um DEC lässt die Ergebnistabelle absteigend sortiert ausgeben.</p>	<pre>SELECT Nachname, Vorname FROM KUNDE ORDER BY Geburtstag DESC</pre>	<p>Gibt als Ergebnistabelle alle Vor- und Nachnamen absteigend sortiert nach dem Geburtstag aus. Die Liste startet dabei bei dem jüngsten Kunden.</p>	<pre>SELECT Nachname, Vorname FROM KUNDE ORDER BY Nachname, Vorname</pre>	<p>Gibt eine nach Nach- und Vorname geordnete Kundenliste aus.</p>
<pre>SELECT Nachname, Vorname FROM KUNDE ORDER BY Geburtstag DESC</pre>	<p>Gibt als Ergebnistabelle alle Vor- und Nachnamen absteigend sortiert nach dem Geburtstag aus. Die Liste startet dabei bei dem jüngsten Kunden.</p>				
<pre>SELECT Nachname, Vorname FROM KUNDE ORDER BY Nachname, Vorname</pre>	<p>Gibt eine nach Nach- und Vorname geordnete Kundenliste aus.</p>				
<p>Datenbank-abfragen aus mehreren Tabellen ↳ Kapitel 2.2.4</p>	<p>Soll eine Datenbankabfrage Attribute aus verschiedenen Tabellen enthalten, müssen die beteiligten Tabellen im FROM-Teil durch Kommata getrennt aufgelistet werden. Um Verwechslungen bei identischen Spaltenbezeichnern in unterschiedlichen Tabellen vorzubeugen, sollte den Attributen im SELECT-Teil der Tabellenname vorangestellt werden.</p> <p>Ohne weitere Angaben im WHERE-Teil wird hierbei das Kreuzprodukt der Tabellen gebildet. Dabei werden alle Datensätze der beteiligten Tabellen jeweils miteinander kombiniert. Die Ergebnistabelle beinhaltet somit Datensätze, die nicht sinnvoll sind.</p>				

	<pre>SELECT KUNDE.KUID, BESTELLUNG.BID FROM KUNDE, BESTELLUNG</pre>	<p>Jede KUID aus der Tabelle KUNDE wird mit jeder BID der Tabelle BESTELLUNG kombiniert.</p> <p>Wenn die Tabelle Kunde 94 und die Tabelle Bestellung 153 Datensätze haben, hat die Ergebnistabelle $94 * 153 = 14.382$ Datensätze.</p>
<p>Joinbedingung bei 1:n-Beziehungen ↳ Methode S. 106</p>	<p>Um nach der Bildung des Kreuzprodukts die sinnvollen Datensätze herauszufiltern, muss im WHERE-Teil der Datenbankabfrage eine Joinbedingung formuliert werden. Bei 1:n-Beziehungen setzt man hierzu den Fremdschlüssel mit dem zugehörigen Primärschlüssel gleich.</p> <pre>SELECT KUNDE.KUID, BESTELLUNG.BID FROM KUNDE, BESTELLUNG WHERE KUNDE.KUID = BESTELLUNG.bestelltVon</pre>	<p>In der Ergebnistabelle werden lediglich die Datensätze angezeigt, bei denen der Fremdschlüssel der Tabelle BESTELLUNG (bestelltVon) und der Primärschlüssel der Tabelle KUNDE (KUID) übereinstimmen.</p> <p>Von den 14.382 Datensätzen des Kreuzprodukts bleiben nun noch 153 Datensätze übrig.</p>

Joinbedingung bei m:n- Beziehungen

➔ Methode
S. 107

Beinhaltet eine Datenbankabfrage Attribute aus Tabellen, die in einer **m:n-Beziehung** stehen, so wird für die **Joinbedingung** die **Beziehungstabelle** benötigt. Im WHERE-Teil muss dann der jeweilige Primärschlüssel der beteiligten Tabellen mit dem zugehörigen Fremdschlüssel in der Beziehungstabelle gleichgesetzt werden. Die beiden Statements werden durch ein AND verknüpft.

```
SELECT ARTIKEL.AID,
BESTELLUNG.BID

FROM ARTIKEL,
BESTELLUNG,
enthaeltArtikel

WHERE ARTIKEL.AID =
enthaeltArtikel.AID
AND BESTELLUNG.BID =
enthaeltArtikel.BID
```

Um die AIDs der Tabelle ARTIKEL und die BIDs der Tabelle BESTELLUNG miteinander zu verknüpfen, muss im FROM-Teil die Beziehungstabelle **enthaeltArtikel** aufgenommen werden.

Im WHERE-Teil werden die beiden Primärschlüssel der Tabellen ARTIKEL (AID) und BESTELLUNG (BID) mit den zugehörigen Fremdschlüsseln in der Beziehungstabelle **enthaeltArtikel** gleichgesetzt.

Aggregat- funktionen

➔ Kapitel 2.2.5

Neben den Grundrechenarten können in Datenbankabfragen weitere Berechnungen mit Hilfe der **Aggregatfunktionen** durchgeführt werden.

Befehl	Erklärung
MAX(<Spalte>)	Gibt den größten Wert der Spalte aus.
MIN(<Spalte>)	Gibt den kleinsten Wert der Spalte aus.
SUM(<Spalte>)	Gibt die Summe aller Werte der Spalte aus.
AVG(<Spalte>)	Gibt das arithmetische Mittel aller Werte der Spalte aus.
COUNT(<Spalte>)	Gibt die Anzahl an Zeilen in der Spalte aus.

Um in der Ergebnistabelle sinnvolle Spaltenbezeichner zu gewährleisten, sollte bei der Verwendung einer Aggregatfunktion im SELECT-Teil eine sinnvolle Benennung nach dem Schlüsselwort AS erfolgen.

	<pre>SELECT COUNT(KUID) AS Anzahl FROM KUNDE</pre>	<p>Die Ergebnistabelle zeigt an, wie viele Datensätze in der Tabelle KUNDE hinterlegt sind. Der Spaltenbezeichner lautet „Anzahl“.</p>						
<p>Gruppierung von Datensätzen ↳ Kapitel 2.2.5</p>	<p>Möchte man Berechnungen über mehrere Attributwerte einer Spalte durchführen, beispielsweise durch Aggregatfunktionen, muss man Datensätze häufig zunächst zusammenfassen. Durch die Gruppierung nach einem bestimmten Attribut werden alle Datensätze, die in Bezug auf dieses Attribut identische Attributwerte haben, zusammengefasst und durch einen einzigen Attributwert repräsentiert. Mit Hilfe von Aggregatfunktionen können nun innerhalb einer Gruppe Berechnungen vorgenommen werden.</p> <p>Die Gruppierung mit Hilfe von GROUP BY erfolgt dabei nach dem WHERE-Statement. Im SELECT-Teil dürfen neben dem Attribut, nach dem gruppiert wird und verwendeten Aggregatfunktionen keine weiteren Attribute aus Tabellen vorkommen.</p> <table border="1" data-bbox="501 1001 1317 1801"> <tr> <td data-bbox="501 1001 911 1223"> <pre>SELECT Bestelldatum FROM BESTELLUNG GROUP BY Bestelldatum</pre> </td> <td data-bbox="911 1001 1317 1223"> <p>Die Ergebnistabelle fasst alle Bestellungen, die zum gleichen Datum eingegangen sind, zusammen. Jedes Bestelldatum wird in der Ergebnistabelle genau einmal angezeigt.</p> </td> </tr> <tr> <td data-bbox="501 1223 911 1413"> <pre>SELECT Bestelldatum, COUNT(*) AS Anzahl FROM BESTELLUNG GROUP BY Bestelldatum</pre> </td> <td data-bbox="911 1223 1317 1413"> <p>Die Ergebnistabelle zeigt zu jedem Bestelldatum an, wie viele Bestellungen an diesem Tag eingegangen sind.</p> </td> </tr> <tr> <td data-bbox="501 1413 911 1801"> <pre>SELECT KATEGORIE. Bezeichnung, MAX(ARTIKEL.Listenpreis) AS teuersterArtikel FROM KATEGORIE, ARTIKEL WHERE KATEGORIE.KAID = ARTIKEL.gehoertZuKategorie GROUP BY KATEGORIE. Bezeichnung</pre> </td> <td data-bbox="911 1413 1317 1801"> <p>Die Ergebnistabelle zeigt die Bezeichnung jeder Kategorie genau einmal an. Außerdem wird der höchste Listenpreis jeder Kategorie ausgegeben.</p> <p>Vor der Gruppierung mit GROUP BY wird die Joinbedingung im WHERE-Teil angegeben.</p> </td> </tr> </table>		<pre>SELECT Bestelldatum FROM BESTELLUNG GROUP BY Bestelldatum</pre>	<p>Die Ergebnistabelle fasst alle Bestellungen, die zum gleichen Datum eingegangen sind, zusammen. Jedes Bestelldatum wird in der Ergebnistabelle genau einmal angezeigt.</p>	<pre>SELECT Bestelldatum, COUNT(*) AS Anzahl FROM BESTELLUNG GROUP BY Bestelldatum</pre>	<p>Die Ergebnistabelle zeigt zu jedem Bestelldatum an, wie viele Bestellungen an diesem Tag eingegangen sind.</p>	<pre>SELECT KATEGORIE. Bezeichnung, MAX(ARTIKEL.Listenpreis) AS teuersterArtikel FROM KATEGORIE, ARTIKEL WHERE KATEGORIE.KAID = ARTIKEL.gehoertZuKategorie GROUP BY KATEGORIE. Bezeichnung</pre>	<p>Die Ergebnistabelle zeigt die Bezeichnung jeder Kategorie genau einmal an. Außerdem wird der höchste Listenpreis jeder Kategorie ausgegeben.</p> <p>Vor der Gruppierung mit GROUP BY wird die Joinbedingung im WHERE-Teil angegeben.</p>
<pre>SELECT Bestelldatum FROM BESTELLUNG GROUP BY Bestelldatum</pre>	<p>Die Ergebnistabelle fasst alle Bestellungen, die zum gleichen Datum eingegangen sind, zusammen. Jedes Bestelldatum wird in der Ergebnistabelle genau einmal angezeigt.</p>							
<pre>SELECT Bestelldatum, COUNT(*) AS Anzahl FROM BESTELLUNG GROUP BY Bestelldatum</pre>	<p>Die Ergebnistabelle zeigt zu jedem Bestelldatum an, wie viele Bestellungen an diesem Tag eingegangen sind.</p>							
<pre>SELECT KATEGORIE. Bezeichnung, MAX(ARTIKEL.Listenpreis) AS teuersterArtikel FROM KATEGORIE, ARTIKEL WHERE KATEGORIE.KAID = ARTIKEL.gehoertZuKategorie GROUP BY KATEGORIE. Bezeichnung</pre>	<p>Die Ergebnistabelle zeigt die Bezeichnung jeder Kategorie genau einmal an. Außerdem wird der höchste Listenpreis jeder Kategorie ausgegeben.</p> <p>Vor der Gruppierung mit GROUP BY wird die Joinbedingung im WHERE-Teil angegeben.</p>							

Aufbau einer Datenbankabfrage in SQL

→ Kapitel 2.2.6

Datenbankabfragen in SQL können je nach Fragestellung diverse SQL-Schlüsselworte enthalten. Dabei gilt es die folgende Reihenfolge einzuhalten.

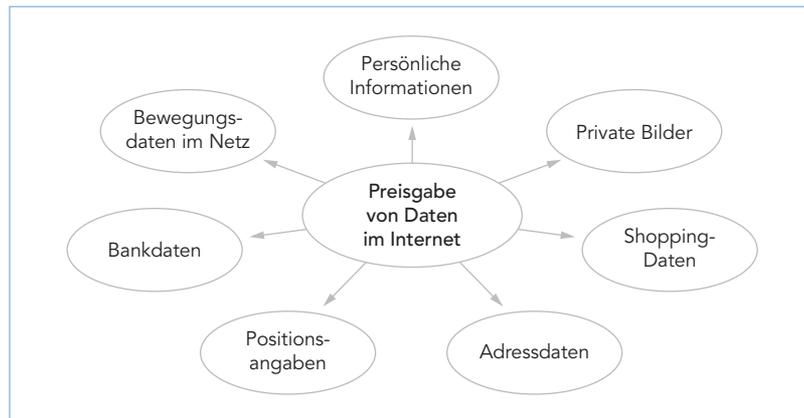
Die Verwendung von WHERE, GROUP BY und ORDER BY ist nicht notwendig, sofern es die Fragestellung nicht erfordert.

```
SELECT      <anzuzeigende Spalten>
FROM        <betroffene Tabellen>
WHERE       <Joinbedingungen und Filterkriterien>
GROUP BY   <Attribute nach denen gruppiert wird>
ORDER BY   <Sortierreihenfolge>
```

Preisgabe von Daten im Internet und sozialen Medien

→ Kapitel 2.2.7

Neben den Grundrechenarten können in Datenbankabfragen weitere Berechnungen mit Hilfe der **Aggregatfunktionen** durchgeführt werden.



Vorausschauender Umgang mit den eigenen Daten

→ Kapitel 2.2.7

Durch das gezielte und umfassende Sammeln von personenbezogenen Daten können einzelne Unternehmen in Verbindung mit modernen Datenbankmanagementsystemen schnell detaillierte Profile ihrer Kunden erstellen, um anschließend deren Wünsche und Bedürfnisse noch besser zu befriedigen und so ihren Umsatz zu erhöhen (= Data Driven Marketing). Aber auch aus vermeintlich anonymen Daten lassen sich durch Kombination unterschiedlicher Datenquellen Rückschlüsse über eine bestimmte Person ziehen. Daher gilt es bei der Nutzung des Internets sowie sozialer Medien sparsam und vorsichtig mit der Preisgabe von Daten über die eigene Person zu sein.