

MODUL 153

DATENMODELLE ENTWICKELN

Reto Glarner

Vertiefung konzeptionelles und logisches Datenmodell



- › Sie kennen die methodische Vorgehensweise bei der Entwicklung eines konzeptionellen Datenmodells und können für die einzelnen Schritte erläutern, wie dabei vorgegangen wird.
- › Sie kennen die erweiterten Konstruktionselemente des ER Modells (Rekursive Assoziation) und der Datenmodellierung (Generalisierung/Spezialisierung) und können aufzeigen, bei welchen Sachverhalten diese in der Datenmodellierung eingesetzt werden.
- › Sie kennen die Regeln, welche beim Spezialisieren bzw. Generalisieren von Objekten in Entitätsmengen und Beziehungen zu beachten sind.
- › Sie kennen die spezifischen Merkmale, welche ein Identifikationsschlüssel in einem Datenmodell erfüllen muss und können erläutern, welche Verarbeitungsmöglichkeiten damit sichergestellt werden.
- › Sie kennen die verschiedenen Kardinalitäten (Assoziationen) und können erläutern, wie diese mittels der Definition von Primär- und Fremdschlüsseln in einem logischen Datenmodell umzusetzen sind.

Die eher einfache Übersicht des AB153-02 zeigt klassische Anforderungen an Datenbanken aus Handel und Industrie. Datenbanken werden aber auch für viel komplexere Gebiete verwendet, bei denen die Modellierung mehr gefordert ist. Denken Sie dabei an Simulationen, 3D Anwendungen, geografische Informationssysteme, Webdatenbanken, Big Data usw.

Soweit als möglich soll den Datenarchitekten Hand geboten werden bei der Entwicklung von Datenmodellen. Dazu gehören unter anderem die folgenden Modellierungsverfahren.

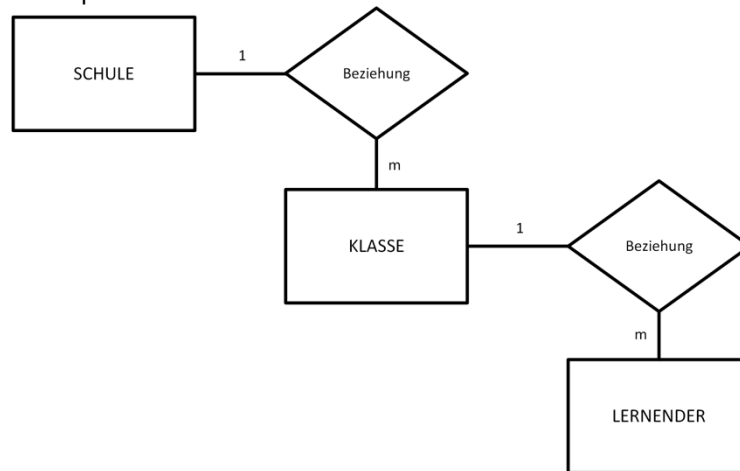
Vom konzeptionellen zum logischen Datenmodell: Hierarchische Beziehungen

In der realen Welt kommen häufig Hierarchien vor. Dabei handelt es sich entweder um unterschiedliche Entitäten oder um gleiche Entitäten.

Die 3 folgenden Fälle sollen vom konzeptionellen Datenmodell in ein logisches (in Tabellen) abgebildet werden. Dabei sollten keine Daten doppelt vorkommen (Redundanz), alle Angaben eindeutig und widerspruchsfrei sein sowie mittels relationalem Datenmodell realisierbar.

Die **Hierarchie bei unterschiedlichen Entitäten** können einfach mittels Beziehungen dargestellt werden. Das Beispiel einer Schule:

Konzeptuelles Datenmodell



Logisches Datenmodell

SCHULE			
<u>SchuleID</u>	Name	...	

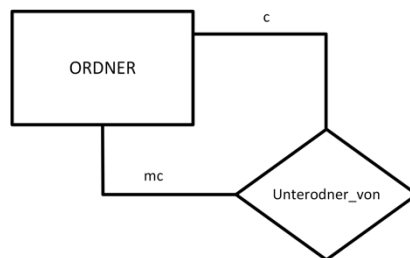
KLASSE			
<u>KlasseID</u>	SchuleID	Name	...

LERNENDER			
<u>LernenderID</u>	KlasseID	Vorname	...

Der Primärschlüssel ist nie leer und eindeutig, deshalb kann er in den unteren Hierarchieebenen als Fremdschlüssel verwendet werden. Der Beziehungstyp (Kardinalität ist 1-m bzw. 1-mc)

Wenn **gleiche Entitäten** hierarchisch vorkommen beziehen sich die Assoziationen auf die gleiche Entität. Dies nennt sich **Rekursion**. Dabei ist zu beachten in welche Richtung die Beziehung funktioniert. Das Beispiel einer Ordnerstruktur im Dateisystem:

Konzeptuelles DM



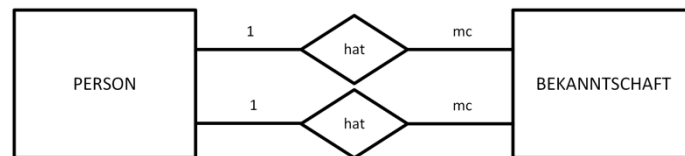
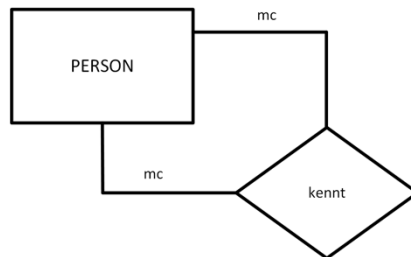
Logisches DM

ORDNER			
<u>OrdnerID</u>	UeberordnerID	Name	...

Ein Spezialfall bilden die **netzwerkartigen** Beziehungsformen, da diese nicht mit einer einzigen Verbindung pro Entität auskommen. Das Beispiel Personen kennen Personen:

Konzeptionelles DM

(Die zweite Darstellung ist zu bevorzugen)



Der Beziehungstyp (Kardinalität) ist mc-mc, deshalb wird bei der Abbildung ins logische Datenmodell eine Zwischentabelle benötigt.

Logisches DM

PERSON		
<u>PersonID</u>	Name	...

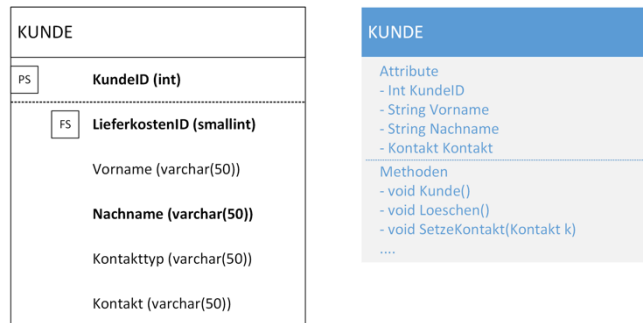
BEKANNTSCHAFT		
<u>BekanntschaftID</u>	PersonA	PersonB



Wenn Sie die Beispiele verstanden haben, sind Sie bereit die ersten Übungen auf diesem Arbeitsblatt durchzuführen. → Aufgabe 1 + 2

Subklassen, Superklassen und Vererbung

Vergleichen wir einen Datenbankentwurf mit der objektorientierten Programmierung fällt auf, dass es viele Parallelen zwischen den Entitäten bei Objekten bzw. Datensätzen und Entitätsmengen (Tabellen) und Klassen gibt. Sogar die Darstellung ähneln einander, wenn wir das logische Datenmodell und das Klassenmodell in UML vergleichen fällt auf, dass diese nur wenige Unterschiede haben. Der grösste Unterschied liegt darin, dass beim Klassenmodell noch Prozesse mittels Methoden angegeben werden können, dieser Ansatz ist beim Datenmodell nicht vorhanden:



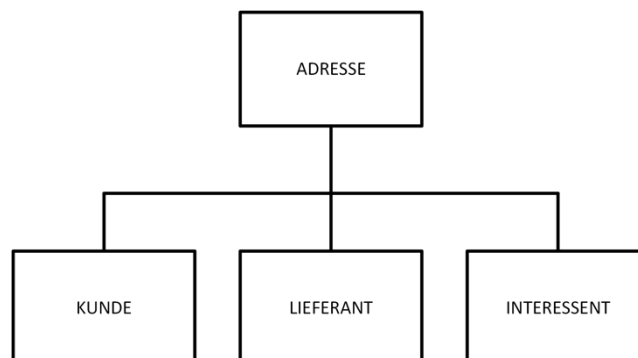
Diese Ähnlichkeit kann uns helfen die folgenden Konzepte besser zu verstehen, da diese in der OO-Programmierung zum täglichen Brot gehören.

Wir werden zuerst die Vererbung, Subklassen und Superklassen betrachten, welche unsere *Werkzeuge* sein sollen. Mit diesen Werkzeugen können wir dann die beiden *Prozesse* Generalisierung und Spezialisierung durchführen, welche sich sehr ähnlich sind.

Als Beispiel für Vererbung nehmen wir Adressen, diese kommen in der Praxis immer wieder vor.

Unsere Tante Emma aus AB153-02 ist sehr zufrieden mit dem Programm und möchte nun expandieren. Bisher hat sie lediglich Kundenadressen gespeichert. Neu möchte sie auch ihre Lieferanten erfassen, damit sie diese mit den Artikeln verknüpfen kann. Dann gibt es eine grosse Anzahl an Interessenten, diese sollen regelmässig einen Flyer erhalten. Daraus können wir folgende Klassen ableiten:

Fallbeispiel: Adresstypen von Tante Emma



Nun haben wir eine Superklasse „Adresse“ und 3 unterschiedliche Subklassen. Die Subklassen erben alle Attribute der Superklasse und haben zusätzlich noch eigene Attribute:

- › Kunde: Lieferkosten, Rabatt, Liefersperre, MwSt. Nummer, Ertragskonto usw.
- › Lieferant: MwSt. Nummer, Aufwandkonto usw.
- › Interessent: Kommt von, Flyerversand usw.

Spezialisieren und Generalisieren

Die beiden Begriffe stellen je einen Vorgang dar, welcher in der Modellierung von Datenbanken vorkommen kann. Wenn der Vorgang abgeschlossen ist, kommt es nicht mehr darauf an ob es sich ursprünglich um eine Spezialisierung oder eine Generalisierung gehandelt hat, das Resultat ist das gleiche! Je nach Ausgangslage wird der eine oder andere Prozess durchgeführt.

Spezialisierung

Tante Emmas Adressen sind bisher alle in der Tabelle ADRESSE gespeichert gewesen. Neu möchte Sie diese aufteilen können in die 3 oben erwähnten Gruppen.
→ Aufteilen, Top Down Sicht

Generalisierung

Tante Emma hat bis jetzt 3 unterschiedliche Tabellen für die Adressen geführt und möchte diese nun wie gleichwertige „Adressen“ behandeln können.
→ Zusammenfassen, Bottom Up Sicht

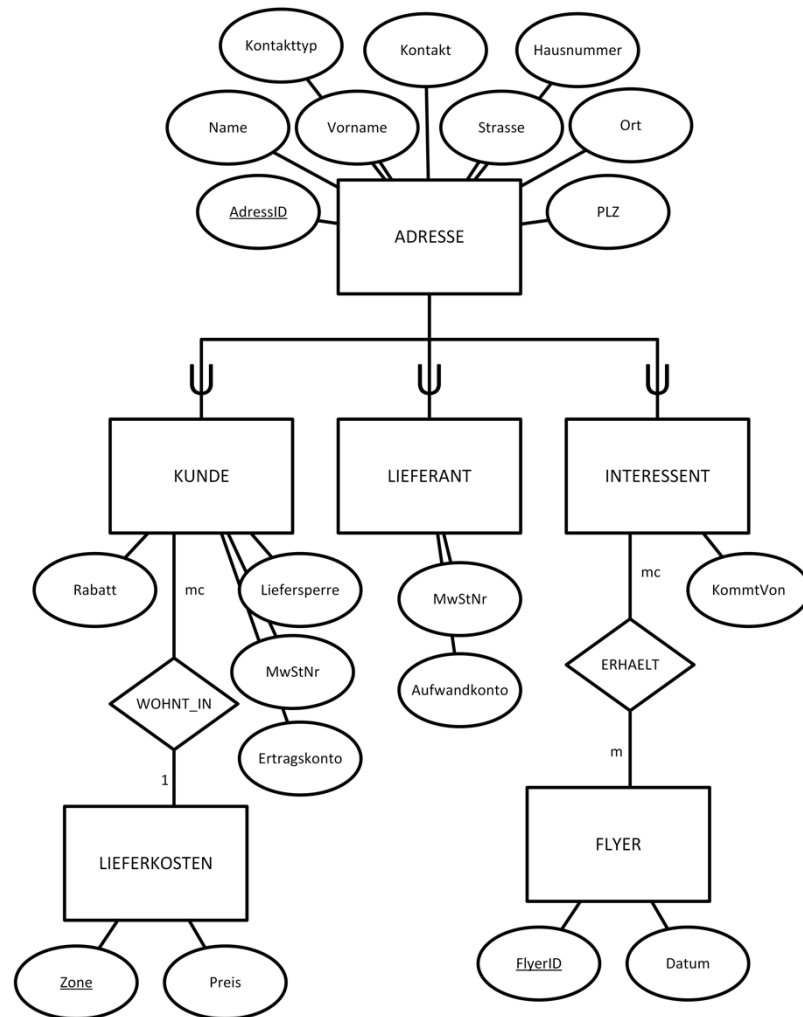
Wenn wir eine Datenbank von Grund auf entwerfen, spielt die Frage ob Spezialisierung oder Generalisierung keine grosse Rolle. Wichtig ist die korrekte Umsetzung der Super- bzw. Subklassen welche uns ermöglicht:

- › eine Menge an Subklassen für einen Entitätstyp zu definieren
- › zusätzliche, spezifische Attribute für jede Subklasse anzulegen
- › zusätzliche, spezifische Beziehungstypen für jede Subklasse anzulegen

Was bedeutet die dritte Aussage?

Je nach Subklasse kann es im Datenmodell unterschiedliche Verknüpfungen geben. Die Kunden haben eine Verknüpfung mit den Lieferkosten, diese besteht bei den Interessenten nicht, diese haben dann Verknüpfungen mit dem Flyerversand.

Bauen wir das Fallbeispiel in der **konzeptuellen Sicht** noch ein wenig aus. Die Schreibweise mit dem „U“ steht für die Vererbung.



Wie kann man nun feststellen, ob eine Adresse ein Kunde, ein Lieferant oder ein Interessent ist?
Das kommt ganz auf die Umsetzung in das **logische Datenmodell** an. Natürlich gibt es verschiedene Varianten, welche zielführend sind:

Benutzerdefinierte Spezialisierung / Generalisierung

ADRESSE					
<u>AdresseID</u>	Vorname	Nachname	Kontakttyp	Kontakt	

KUNDE					
<u>AdresseID</u>	LieferkostenID	Rabatt	Liefersperre	MwStNr	Ertragskonto

LIEFERANT					
<u>AdresseID</u>	MwStNr	Aufwandkonto			

INTERESSENT					
<u>AdresseID</u>	KommtVon				

Jeder Entitätstyp wird in einer eigenen Tabelle gespeichert. Die Primärschlüssel werden in den Subklassen wiederholt, wo sich auch die spezifischen Attribute befinden. In der Superklasse sind die Attribute, welche für alle Entitätstypen gelten bzw. vererbt werden.

Prädikatsdefinierte Spezialisierung / Generalisierung (Bedingungsdefiniert)

Bei dieser Variante wird anhand des Wertes eines oder mehreren Attributen entschieden, um welchen Entitätstyp es sich handelt.

Bsp.: Kunden = Adressnummer 1000 – 1999 Lieferant = 2000 – 2999 usw.

Attributdefinierte Spezialisierung / Generalisierung

ADRESSE					
<u>AdresseID</u>	Adresstyp	Vorname	Nachname	Kontakttyp	Kontakt
KUNDE					
<u>AdresseID</u>	LieferkostenID	Rabatt	Liefersperre	MwStNr	Ertragskonto
LIEFERANT					
<u>AdresseID</u>	MwStNr	Aufwandkonto			
INTERESSENT					
<u>AdresseID</u>	KommtVon				

In der Superklasse wird in einem Attribut festgehalten um welche Subklasse es sich handelt. Der Unterschied zur prädikatsdefinierten Spezialisierung besteht darin, dass dieses Attribut lediglich dem Zweck der Identifizierung gilt.

Bsp.: Kunden: Adresstyp = K Lieferant: Adresstyp = L Interessent: : Adresstyp = I

Einschränkungen für Spezialisierung / Generalisierung

Beim Gespräch mit Tante Emma stellen sich früher oder später die folgenden Fragen:

Frage: Kann ein Kunde auch ein Lieferant sein?

Tante Emma: Ja klar das gibt's!

Kann ein Interessent auch ein Kunde sein?

Tante Emma: Nein, beim ersten Kauf wird ein Interessent zum Kunden.

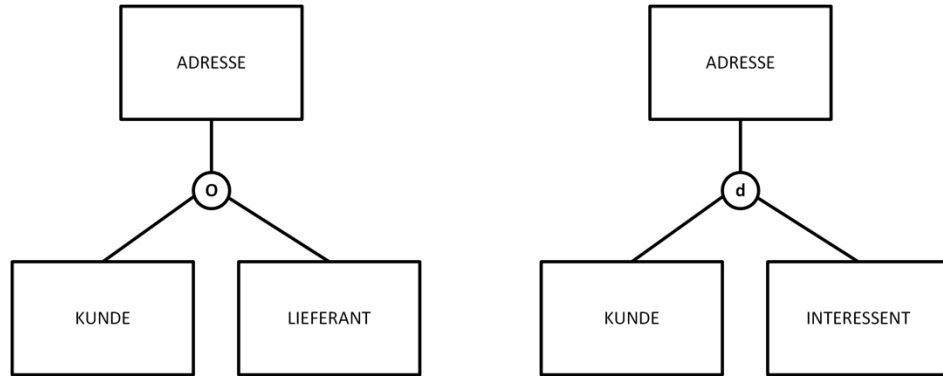
Was bedeutet dies für unser Datenmodell? **Disjunktheit - Einschränkung**

Die Subklassen Kunde und Lieferant sind **überlappend**, da ein Kunde auch Lieferant sein kann und umgekehrt.

Die Subklassen Kunde und Interessent sind **disjunkt**, da eine Adresse nicht beides gleichzeitig sein kann.

Darstellung im konzeptionellen Datenmodell
(o = overlap)

(d = disjoint)



Suchen Sie selber je 2 Beispiele, welche überlappend oder disjunkt sind und notieren Sie diese stichwortartig!

✍

Die letzte Frage, welche es durch Tante Emma zu beantworten gilt lautet: Wird jede Adresse einer Unterklasse zugeordnet oder gibt es auch „nur – Adressen“?

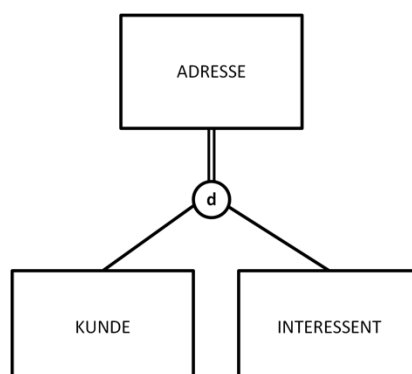
Antwort: Alle Adressen werden einer Unterklasse zugeordnet.

Diese Eigenschaft nennt sich in der Datenbanksprache **Vollständigkeits - Einschränkung**. Es gibt ähnlich wie oben zwei Möglichkeiten:

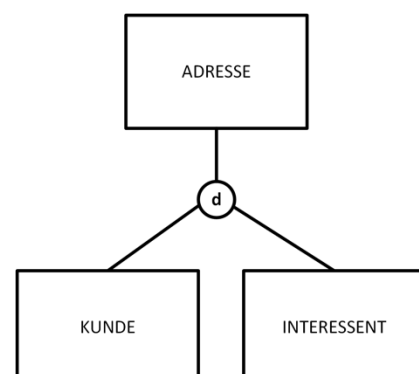
Es handelt sich um eine **totale Spezialisierung**, da jede Adresse einer Subklasse angehören muss.

Es handelt sich um eine **partielle Spezialisierung**, da es auch Adressen gibt, welche keiner Subklasse angehören. (nicht zutreffend in unserem Fallbeispiel)

Darstellung im konzeptionellen Datenmodell
(doppelte Linie)



(einfache Linie)



Durch die beiden Einschränkungen ergeben sich nun 4 mögliche Fälle bei der Spezialisierung / Generalisierung:

- 1) überlappend total
- 2) überlappend partiell
- 3) disjunkt total
- 4) disjunkt partiell

Nehmen Sie Ihre Beispiele von der vorigen Seite und überlegen Sie sich welchen Fall diese abbilden!



Zur Vertiefung der Theorie wird eine Aufgabe helfen. → Aufgabe 3



Repetition referentielle Integritätsbedingungen

Repetieren Sie das Arbeitsblatt 9 vom Modul 104. Es wird Ihnen helfen, um die weiteren Aufgaben zu lösen!

Abbilden des logischen Datenmodells für Spezialisierung / Generalisierung

Die Überführung vom konzeptionellen zum logischen Datenmodell wurde bereits unter dem Aspekt behandelt, wie die Zuweisung zur korrekten Subklasse erfolgen kann. Dabei wurde für die Superklasse eine Tabelle erstellt und pro Unterklasse auch. Dies ist nicht die einzige Möglichkeit!

1) Superklasse und Subklasse je eine Tabelle (bekannt)

ADRESSE						
AdresseID	Adresstyp	Vorname	Nachname	Kontakttyp	Kontakt	

KUNDE						
AdresseID	LieferkostenID	Rabatt	Liefersperre	MwStNr	Ertragskonto	

LIEFERANT						
AdresseID	MwStNr	Aufwandkonto				

INTERESSENT						
AdresseID	KommtVon					

2) Eine Klasse pro Subklasse (keine Superklasse)

KUNDE									
AdresseID	Vorname	Nachname	Kontakttyp	Kontakt	LieferkostenID	Rabatt	Liefersperre	MwStNr	Ertragskonto

LIEFERANT										
AdresseID	Vorname	Nachname	Kontakttyp	Kontakt	MwStNr	Aufwandkonto				

INTERESSENT										
AdresseID	Vorname	Nachname	Kontakttyp	Kontakt	KommtVon					

Bei dieser Variante fehlt die Superklasse, stattdessen werden alle (vererbten) Attribute in den Subklassen abgebildet. Wird eher bei Disjunktheit angewendet, da sonst viele redundante Daten geschaffen werden.

Vorteil: Es steht immer der komplette Datensatz ohne JOIN Operation zur Verfügung. Es gibt wenig leere Attribute.

Nachteil: Bei Abfragen über vererbte Attribute müssen die Abfragen über mehrere Tabellen gemacht werden. Der Primärschlüssel muss über eine zusätzliche Logik vergeben werden.

3) Alles in einer Tabelle 1 zusätzliches Attribut

ADRESSE													
AdresseID	Adresstyp	Vorname	Nachname	Kontakttyp	Kontakt	LieferkostenID	Rabatt	Liefersperre	KMwStNr	Ertragskonto	LMwStNr	Aufwandkonto	KommtVon

Diese Abbildungsart funktioniert für alle Varianten und ist ebenfalls attributdefiniert.

Vorteil: Es steht immer der komplette Datensatz ohne JOIN Operation zur Verfügung.

Nachteil: Das Attribut „Adresstyp“ ist nur bei disjunkten Daten einwertig und genau in diesem Fall erzeugt diese Darstellung viele NULL-Werte.

4) Alles in einer Tabelle mit 1 zusätzlichen Attribut pro Subklasse

ADRESSE															
AdresseID	Vorname	Nachname	Kontakttyp	Kontakt	Kflag	LieferkostenID	Rabatt	Liefersperre	KMwStNr	Ertragskonto	Lflag	LMwStNr	Aufwandkonto	Iflag	KommtVon

Dies ist eine Variante, welche für überlappende Klassen funktioniert. Die Flags sind Bitwerte, welche angeben ob die referenzierte Subklasse verwendet wird. So ergeben sich keine mehrwertigen Attribute wie in Variante 4.

Vorteil: Geht für alle 4 Möglichkeiten. Der Primärschlüssel ist einfach zu verwalten, da er nur in einer Tabelle vorkommt. Es werden keine JOIN Operationen benötigt.

Nachteil: Es entstehen Tabellen mit vielen Attributen. Es entstehen Tabellen mit vielen Datensätzen, da alle Daten in derselben Tabelle abgelegt sind.

Für welche Variante Sie sich entscheiden bleibt Ihnen überlassen, der Kunde wird diese Sicht nicht haben bzw. nicht verstehen. Entscheiden Sie nie vorschnell, jede Variante hat ihre Vor- und Nachteile. Für den Entscheid sollten die Operationen (CRUD, vor allem die Abfragen) in Betracht gezogen werden.



Übungen Theorie und Praxis → Aufgabe 4

Aufgabenteil

1) Hierarchische Beziehungen (Einzelarbeit)

15 Minuten

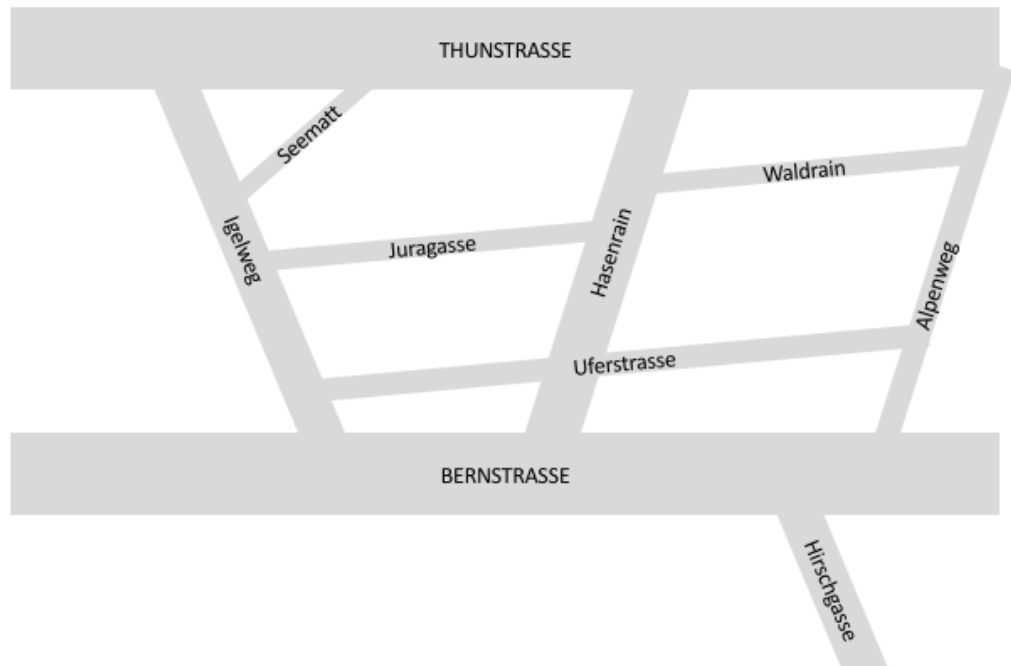
Denken Sie sich für die Übung 2 des AB153-02 (Kunstmuseum Bern) eine geeignete Lösung für die Archivierung aus. Stichworte: Gebäude, Räume, Stockwerke, Fächer, Regal, Gang usw.

Erstellen Sie das konzeptionelle und logische Datenmodell! Vergleichen Sie die Resultate untereinander, die Lehrperson steht bei Fragen zur Verfügung!

2) Hierarchische Beziehungen (2er Team)

45 Minuten

Denken Sie sich eine geeignete Lösung aus für die Speicherung von Strassen. Es gibt 3 Strassentypen (Quartierstrasse, Nebenstrasse, Hauptstrasse). Jeder Typ kann in jeden anderen Typ münden. Die Datenbank soll es erlauben mittels einer Abfrage auszulesen mit welchen anderen Strassen eine bestimmte Strasse verbunden ist. Beispiel:



Erstellen Sie in diesem Beispiel auch das physische Datenmodell und bilden Sie obiges Beispiel ab! Die Abfrage soll mit dem Strassennamen als Parameter funktionieren und alle verbundenen Strassen mit deren Typ ausgeben.

3) Einschränkungen (Einzelarbeit)

15 Minuten

Beantworten Sie folgende Fragen:

Wieso ist eine Generalisierung meistens total?




Was bedeutet die Generalisierung für die Modifikationen der Datensätze? Beachten Sie alle 4 Fälle (überlappend, disjunkt, partiell, total)


Einfügen in Superklasse



Löschen in Superklasse



Einfügen in Subklasse



Löschen in Subklasse



4) Abbilden des logischen Datenmodells (Theorie, Einzelarbeit)

20 Minuten

Überlegen Sie sich welche Variante (1-4) Sie wählen würden und begründen Sie Ihre Auswahl indem Sie die Vor- und Nachteile abwägen! Überlegen Sie sich zuerst die Eigenschaften (überlappend – disjunkt, partiell – total)

A) Wie bilden Sie die Ausstellungsstücke des Kunstmuseums von Bern ab wenn es sich dabei um Bilder (verwendete Technik, Länge, Breite,...), Plastiken (Gewicht, Material,...), Bücher (Anzahl Seiten, Sprache,...) und Multimediale Inhalte (Bild, Ton, Dauer, Technologie,...) handelt?



B) Wie bilden Sie die Adressen von Tante Emma ab? Kunden und Lieferanten sind überlappend, Interessenten disjunkt. Die Adressen werden total abgebildet.



C) Tante Emma ist auf den Biomarkt eingestiegen. Die Bioprodukte benötigen zusätzliche Angaben wie Transportweg CO2 Ausstoss, Wasserbedarf pro Kilo und Angaben zum Bio-Label. Die normalen Produkte benötigen diese Angaben nicht. Es gibt auch Bioprodukte mit mehreren Lieferanten, also unterschiedlichen Bio-Angaben fürs gleiche Produkt.



5) Abbilden des logischen Datenmodells (Praxis, 2er Team)

60 Minuten

Erstellen Sie eine der 4 Varianten für Tante Emmas Adressen als physisches Datenmodell auf dem SQL Server. Begründen Sie Ihre Auswahl!

Erstellen Sie 6 Datensätze so dass jede Möglichkeit mindestens einmal vorkommt.

Erstellen Sie eine Abfrage, welche alle Adressen mit allen Angaben ausgibt!

Erstellen Sie eine Abfrage, welche nur Kunden mit den spezifischen Attributen ausgibt.

Notieren Sie die Schritte in SQL, welche benötigt werden, um aus einem Interessenten einen Kunden zu machen!



Für Interessierte: Lesen Sie den kurzen Abschnitt „Abbildung von Vererbungshierarchien“ auf folgender Seite und ordnen Sie diese unseren 4 Varianten zu!

https://de.wikipedia.org/wiki/Objektrelationale_Abbildung