



**Hochschule  
Bonn-Rhein-Sieg**  
University of Applied Sciences

**Datenbanken**

*- Übung 2 -  
Semantisch ausdrucksstarkes  
konzeptionelles Schema*

**Harm Knolle  
Markus Schneider**

DBS\_Übung\_2.doc vom 22.10.2023 12:30:00

Druck vom 21.04.2024 04:11:00

**Prof. Dr. H. Knolle  
Hochschule Bonn-Rhein-Sieg  
Fachbereich Informatik  
Grantham-Allee 20  
53757 Sankt Augustin**

---

**Inhaltsverzeichnis**

<b>1</b>	<b>Das Werkzeug DB-Main .....</b>	<b>3</b>
1.1	Installation unter Windows und Linux .....	3
1.2	Installation unter Mac OS .....	3
<b>2</b>	<b>Erstellung eines semantischen (konzeptionellen) Schemas .....</b>	<b>4</b>
2.1	Transformation der Entitätstypen .....	4
2.2	Transformation der Beziehungstypen .....	5
2.3	Datentypen für Attribute .....	6
2.4	Erforderliche Modellierungsfälle .....	6
2.5	Benötigte Beziehungsketten als spätere Übungsbereiche .....	7
<b>3</b>	<b>Einreichung Ihrer Ergebnisse .....</b>	<b>9</b>

## 1 Das Werkzeug DB-Main

Im Rahmen der folgenden Übungen werden Sie mit dem computer-aided software engineering Werkzeug (CASE Tool) DB-Main arbeiten. Das Werkzeug steht in den Übungslaboren unter Windows und Linux zur Verfügung. Sie können das System zur Vertiefung der Übungen auch auf Ihrem privaten Rechner nutzen.

### 1.1 Installation unter Windows und Linux

- a) Die aktuelle freie und nicht begrenzte DB-Main-Version 11.0.2 finden Sie unter <https://www.db-main.eu/getit/>. Es werden Windows und Linux unterstützt. **Bitte beachten Sie, dass das Sicherheitszertifikat der DB-Main Webseite offenbar am 23.9.2023 abgelaufen ist. Dadurch wird Ihr Browser beim Zugriff auf die Webseite eine Warnung ausgeben.**
- b) Machen Sie sich mit dem Werkzeug vertraut.
  - a. Im Installationsordner finden Sie das DB-Main-„Reference Manual“. Im Programm DB-Main finden Sie im Menü-Punkt „Help“ den Unterpunkt „First Steps“.
  - b. Versuchen Sie erste Schemata zu erstellen, abzuspeichern und wieder zu laden.
- c) **WICHTIG: speichern Sie Ihre Arbeit in regelmäßigen Abständen ab!**

### 1.2 Installation unter Mac OS

Leider gibt es die aktuelle Version von DB-Main nicht nativ für Mac OS. Sie können DBMain auf dem Mac aber mit Hilfe einer Virtualisierungslösung betreiben, z.B.

- UTM (kostenfrei): <https://mac.getutm.app/>
- Parallels Desktop (kommerziell): <https://www.parallels.com/>

Sie benötigen dann noch ein geeignetes Gast-Betriebssystem (Windows oder Linux). Bei der Auswahl des Gast-Betriebssystems sollten Sie die Prozessorarchitektur Ihres Mac berücksichtigen:

- Intel (Core i5/i7/...): Jede (halbwegs aktuelle) Version von Windows bzw. Linux.
- ARM (M1/M2/...): Hier bietet sich Windows 11 als Gast-Betriebssystem an, da dieses auch auf einem ARM-Prozessor x86-Software wie DB-Main ausführen kann

Nach der Installation eines geeigneten Gast-Betriebssystems können Sie DB-Main in der passenden Version für Ihr Gast-Betriebssystem herunterladen und installieren.

## 2 Erstellung eines semantischen (konzeptionellen) Schemas

Die von Ihnen in der letzten Übung erstellte Skizze soll mit Hilfe des CASE-Tools DB-Main in ein semantisch ausdrucksstarkes konzeptionelles Schema transformiert werden. Grundlage hierfür sind die Basis-Konzepte der Entity Relationship Modellierung (oft auch konzeptuelles Modell genannt). Gehen Sie bitte wie folgt vor:

- a) Erstellen Sie in DB-Main ein neues Projekt mit dem Namen Ihres zu realisierenden Informationssystems (`File / New project`).
- b) Fügen Sie dem Projekt ein neues Schema hinzu (`Product / New schema`). Nennen Sie Ihr Schema „Semantisches Schema“.
- c) Überführen Sie jeden Entitäts- und Beziehungstypen gemäß der Abschnitte 2.1 und 2.2.
  - a. Die Schreibweise der Entitäts- und Beziehungstypen muss mit den Anwendungsfällen der „EigeneDB.ods“-Datei übereinstimmen.
  - b. Sollten gegenüber der Übung 1 Entitäts- oder Beziehungstypen hinzukommen, wegfallen oder sich ändern, so passen Sie auch die Anwendungsfälle an. Die Konsistenz zwischen der DB-Main-Datei und den Anwendungsfällen wird vom Praktomaten geprüft.

Damit Sie die folgenden Übungsblätter problemlos bearbeiten können, ist es notwendig, dass Ihr semantisches Schema alle benötigten Konstruktionen enthält. Bitte beachten Sie dazu die Abschnitte 2.4 und 2.5.

### 2.1 Transformation der Entitätstypen

Erstellen Sie für jeden Entitätstypen Ihrer Skizze einen gleichnamigen Entitätstypen:

- a) Im Schema (`New / Entity type`), beenden mit der `<Esc>`-Taste.
- b) Setzen Sie den Namen des Entitätstypen (links, in der `Property box`).
- c) Legen Sie pro Entitätstyp die Namen der jeweiligen Attribute fest (ganz unten in der `Property box`, das `Icon First att`, die nächsten Attribute dann mit `Next att`).
- d) Legen Sie pro Attribut in der `Property box` des Attributes einen sinnvollen Datentyp fest und geben Sie unter `cardinality` an, ob es sich um ein optionales Attribut handelt. Eine Übersicht über mögliche Datentypen finden Sie in Abschnitt 2.3.
- e) Ausgewählte Attribute können komplex sein:

- a. Es sind sowohl zusammengesetzte Attribute mit Unterattributen (z.B. „Adresse“, das sich aus den Unterattributen „Land“, „Postleitzahl“, „Ort“, „Straße“ und „Hausnummer“ zusammensetzen kann). Das geschieht, in dem bei einem Attribut erneut das Icon `First att`, ausgewählt wird (und nicht `Next att`),
  - b. als auch mengenwertige Attribute erwünscht (z.B. „Hobbies“, das sich aus einer Menge von typgleichen Unterattributen „Hobby“ zusammensetzen kann). Das erfolgt ebenfalls über `cardinality`, wo die Obergrenze der Mengenwertigkeit festgelegt werden kann.
- f) Schlüssel:
- a. Auf Ebene des ER-Diagramms sollte auf künstliche Schlüssel verzichtet werden. Künstliche Schlüssel werden bei der Überführung in das logische Schema behandelt.
  - b. Sofern natürliche (auch zusammengesetzte) Schlüssel vorhanden sind, kennzeichnen Sie die entsprechenden Attribute (ein oder mehrere Attribute im Schema markieren und dann auf das Icon `ID` klicken).
  - c. Sollte ein natürlicher Schlüssel „große“ Werte enthalten, dann legen Sie den Schlüssel als alternativen (sekundären) Schlüssel an. Später wird dann ein kompakter künstlicher Primärschlüssel hinzugefügt (Frage: warum sollte das so gemacht werden?).
  - d. Falls mehrere unabhängige Schlüssel für einen Entitätstyp vorhanden sind, unterscheiden Sie primäre und alternative (sekundäre) Schlüssel (unter `identifizier` in der `Property box` des Schlüssels). Es können auch alle Schlüssel als alternative Schlüssel angelegt werden. Dieses stellt sicher, dass später ein möglichst kompakter Primärschlüssel generiert wird.

## 2.2 Transformation der Beziehungstypen

Überführen Sie jeden Beziehungstypen Ihrer Skizze nach DB-Main:

- a) Im Schema das „+“-Symbol anklicken und dann zwei Entity-Typen verbinden. Alternativ können Sie auch das Raute-Symbol anklicken und einen Beziehungstyp platzieren. Danach das „+“-Symbol anklicken und Verbindungen zwischen dem Beziehungstyp und Entitätstypen herstellen.
- b) Übernehmen Sie den Namen des Beziehungstypen aus dem Diskursbereich (links, in der `Property box`).
- c) Legen Sie für die Beziehungstypen, sofern semantisch erforderlich, eigene beschreibende Attribute fest.
- d) Legen Sie pro Beziehungstyp-Attribut einen sinnvollen Datentyp fest. Eine Übersicht finden Sie in Abschnitt 2.3.

- e) Bestimmen Sie den Typ der Beziehung („1:1“, „1:N“ oder „N:M“) mit Hilfe entsprechender Kardinalitäten nach **der Min/Max-Notation** (in der Property box der Kardinalität).
- f) Versetzen Sie die Kardinalitäten mit sinnvollen Namen (ebenfalls in der Property box der Kardinalität).

## 2.3 Datentypen für Attribute

Wir empfehlen die Verwendung der folgenden Datentypen für Attribute:

- **char (n)**: Zeichenketten einer maximalen Länge n. Ein Speicherbereich der Länge n wird für jeden Datensatz fix reserviert, auch bei der Speicherung kürzerer Werte. Vorteilhaft ist, dass sich bei Änderungen der Werte kein Umorganisationsaufwand auf dem Sekundärspeicher ergibt.
- **varchar (n)**: Zeichenketten einer maximalen Länge n. Im Datensatz wird nur der Speicher belegt, der von der konkreten Zeichenkette benötigt wird. Nachteilig ist, dass sich bei Änderungen der Länge ein Umorganisationsaufwand auf dem Sekundärspeicher ergibt.
- **numeric (n, d)**: Ein Zahl mit insgesamt n Ziffern und d Nachkommastellen. Beispiele:
  - numeric (3, 0): Für die Ganzzahlen im Bereich von 0 bis 999.
  - numeric (5, 4): Für Dezimalbrüche mit einer Ziffer vor dem Komma und vier Nachkommastellen (,3.1415‘)
- **date (n)**: Ein Datums- oder Zeitwert. Die Länge n entspricht der Anzahl der Zeichen bei einer kompakten Kodierung ohne Trennzeichen. Beispiele:
  - date(4) für Uhrzeiten (,1443‘, also ,14:43‘)
  - date(8) für Datumsangaben (,20220418‘, also: ,2022-04-18‘)
  - date(12) für Datumsangaben mit Uhrzeit (,202204181443‘, also: ,2022-04-18 14:43‘)
  - date(16) für Datumsangaben mit Uhrzeit (,20220418144359‘, also: ,2022-04-18 14:43:59‘)
- **boolean (1)**: Ein Wahrheitswert, die Länge ist hier fix 1.
- **compound (n)**: Ein komplexes strukturiertes Attribut, das Unterattribute besitzt. Die Länge n ergibt sich automatisch aus der summierten Länge der Unterattribute.

Bitte verzichten Sie bei der Modellierung auf komplex strukturierte Attribute mit mehr als einer Unterenebene. Das ist zwar prinzipiell möglich, wird zur Zeit in den Praktomat-Prüfungen aber nicht unterstützt.

## 2.4 Erforderliche Modellierungsfälle

Für den weiteren Verlauf der Übungen ist es wichtig, dass Ihr semantisches Schema alle relevanten Modellierungsfälle abdeckt. Prüfen Sie bitte, ob Sie jeden der folgenden Fälle mindestens einmal in Ihrem semantischen Schema wiederfinden:

- a) Komplexes strukturiertes Attribut
- b) Komplexes mengenwertiges Attribut

- c) Semantisch ausdrückstarker Schlüssel
- d) Aus mehreren Attributen zusammengesetzter alternativer Schlüssel
- e) Beziehungstyp mit mindestens einem eigenen Attribut
- f) „1:1“-Beziehungstyp
- g) „1:N“-Beziehungstyp
- h) „N:M“-Beziehungstyp
- i) „Muss“-Kardinalität auf einer „mehrfachen“ Seite
- j) „Muss“-Kardinalität auf einer „einfachen“ Seite

Falls ein Modellierungsfall fehlt, erweitern Sie bitte Ihr Schema zweckdienlich und passen ggf. die Anwendungsfälle entsprechend an. Falls Sie Schwierigkeiten haben, eine geeignete fachliche Änderung durchzuführen, diskutieren Sie bitte in Ihrer Arbeitsgruppe, im LEA-Forum oder in der Übung.

## 2.5 Benötigte Beziehungsketten als spätere Übungsbereiche

Auf der physischen Ebene (ab Übung 5) wird jedes Teammitglied einen kleinen Ausschnitt Ihres gemeinsamen Schemas mit Daten befüllen und SQL-Übungen durchführen. Ein solcher Übungsbereich muss aus einer geeigneten Kette von zwei Beziehungstypen bestehen:

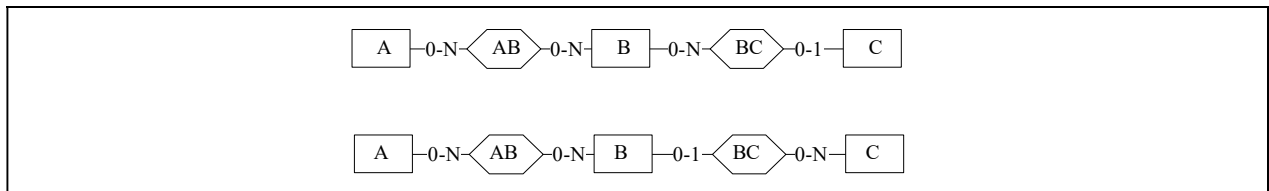


Abbildung 1: Kette aus zwei Beziehungstypen (in zwei Varianten)

1. Einem „N:M“-Beziehungstyp (in der Abbildung abstrakt als „AB“ bezeichnet)
2. Einem angeschlossenen „1:N“- oder „N:1“-Beziehungstyp („BC“)

Die (min)-Werte der Kardinalitäten sind hier nicht relevant: In Abbildung 1 wurden bedingt einfache und bedingt mehrfache Kardinalitäten verwendet, es können aber genauso Beziehungstypen mit nicht-bedingten Kardinalitäten verwendet werden.

Jedes Teammitglied braucht eine eigene Kette zum Üben, wobei Sie einzelne Beziehungen gemeinsam nutzen dürfen, die Ketten dürfen sich also überschneiden. Falls Ihr gemeinsames Schema trotzdem nicht für jedes Teammitglied eine eigene Kette enthält, so erweitern Sie das Schema bitte pragmatisch. Falls Sie Schwierigkeiten haben, eine geeignete fachliche Änderung an Ihrem Diskursbereich durchzuführen, diskutieren Sie bitte in Ihrer Arbeitsgruppe, im LEA-Forum oder im Übungsbetrieb.

Erfassen Sie die Ketten der Teammitglieder in der Tabelle „Thema“ der Datei „EigeneDB.ods“:

	A	B	C
7	Teammitglieder	mschne2m,rhartm2m,mbalg2m	Accountnamen Ihres Teams (komma-separiert). Maximal 3 Personen.
8	Beziehungskette Teammitglied 1	beinhaltet, umfasst	Beziehungskette, die Teammitglied 1 benutzen möchte
9	Beziehungskette Teammitglied 2	beinhaltet, verantwortet	Beziehungskette, die Teammitglied 2 benutzen möchte
10	Beziehungskette Teammitglied 3	beinhaltet, deckt ab	Beziehungskette, die Teammitglied 3 benutzen möchte

**Abbildung 2: Erfassung der Beziehungskette pro Teammitglied**

Eine Kette wird durch die Namen der beiden Beziehungstypen beschrieben, z.B. „beinhaltet, umfasst“. Damit der Praktomat Ihre Zuordnung findet, tragen Sie die Ketten in die Zellen B8-B10 ein (siehe Abbildung 2). Unsere ursprüngliche Vorlage dieser Datei enthielt die entsprechenden Zeilen noch nicht, das ist aber kein Problem: Es genügt wenn Sie die Werte in die leeren Zellen B8-B10 schreiben, damit der Praktomat Ihre Angaben findet.



### 3 *Einreichung Ihrer Ergebnisse*

Sie entscheiden, ob Sie Ihre Ergebnisse im Rahmen der Übung aktiv präsentieren und somit zur Diskussion stellen und Feedback erhalten oder „nur“ im Rahmen der Klausurzulassung einreichen wollen.

Die Einreichung Ihrer Ergebnisse erfolgt über den Praktomaten (Link im LEA-Kurs), der während des Uploads eine formale Umfangs- und Konsistenzprüfung durchführt und Ihnen zu dieser Ebene ein unmittelbares Feedback gibt. Es handelt sich dabei allerdings nicht um eine semantisch-inhaltliche Prüfung Ihrer Lösung. Für die inhaltliche Diskussion und für Feedback zu dieser Ebene steht Ihnen der Übungsbetrieb zur Verfügung.

Eine Diskussion Ihrer Ergebnisse ist im Rahmen des Übungstermins möglich, wenn Sie:

1. Eine formal gültige Einreichung über den Praktomaten durchführen.
2. In der enthaltenen „EigeneDB.ods“-Datei bestätigen, dass Sie bereit sind, Ihre Ergebnisse im Rahmen des Übungstermins zu präsentieren und Sie Feedback und eine Diskussion wünschen. Für diese Angabe existiert in der Tabelle „Thema“ ein entsprechendes Feld.

Bitte beachten Sie, dass die tatsächliche Präsentation Ihrer Ergebnisse insbesondere von der Menge der zur Verfügung gestellten Ergebnisse abhängig ist und leider nicht garantiert werden kann.

Eine Einreichung umfasst folgende Bestandteile:

- „EigeneDB.pdf“: Skizze Ihrer Entitäts- und Beziehungstypen (aus Übung 1)
- „EigeneDB.ods“: Spezifikation Ihres Diskursbereiches (aus Übung 1, ggf. mit Anpassungen)
- „EigeneDB.lun“: Ein Entity-Relationship-Diagramm, das Sie mit Hilfe von DB-Main aus Ihrer Skizze aus Übung 1 abgeleitet haben (entsprechend Kapitel 2 dieser Übung).