



Datenbanken modellieren

Das Erstellen einer Datenbank erfolgt in mehreren Schritten, am Ende besteht eine relationale Datenbank, und nur die werden im Unterricht behandelt, lediglich aus Tabellen.

I. Erstellen des Semantischen Modells

1. Aufgaben der Datenbank analysieren und Objekte erkennen
2. Die Attribute des Objektes benennen
3. Den Datentyp jedes Attributes festlegen
4. Zeichnen des Semantischen Modells
5. Erkennen der Beziehungen und Festlegen der Kardinalität

Beispiel: Es soll der Deutsche Bundestag verwaltet werden. Von jedem Abgeordneten soll die Parteizugehörigkeit, der Name und das Geburtsdatum sowie das Geschlecht und der Wohnort gespeichert werden, außerdem seit wann er in diesem Ort wohnt. Beachte, dass ein Abgeordneter mehrere Wohnorte haben kann. Von den Parteien ist der Name und die Adresse der Parteizentrale zu speichern. Im Bundestag gibt es Fraktionen, die sich aus einer oder mehrerer Parteien zusammensetzen. Von den Fraktionen soll der Name gespeichert werden und welche Parteien ihr jeweils angehören. Um Probleme bei Tippfehlern vermeiden zu helfen, soll der Ort als eigene Klasse modelliert werden!

Zeichne das entsprechende Semantische Modell!

Lösungsvorschlag:

1. Es werden die Objekte Abgeordneter, Partei und Fraktion erkannt.
2. Folgende Attribute müssen in der DB gespeichert werden:
Abgeordneter(Name, Geburtsdatum, Geschlecht)
Partei(Name, Anschrift)
Fraktion(Name)
Ort(PLZ, Name)

Beachte: Beim Objekt Abgeordneter ist die Parteizugehörigkeit kein eigenes Objekt, da es ein Objekt „Partei“ gibt. Bei der Fraktion ist die Partei auch kein eigenes Objekt, da es das Objekt Partei gibt. Es wäre in diesem Fall sicherlich nicht möglich, die Partei wegzulassen und nur bei den Objekten „Abgeordneter“ und „Fraktion“ die Partei als Attribut einzufügen, da zum einen zu viel Redundanz entstehen würde, da bei jedem Abgeordneten die Anschrift der Parteizentrale abzuspeichern wäre, zum anderen im Vorfeld nicht eindeutig bestimmt werden kann, wie viele Parteien sich maximal zu einer Fraktion zusammenschließen.

3. Datentyp den Attributen zuordnen!

```
Abgeordneter{Name:varchar(100), Geburtsdatum(Date), Geschlecht:char}  
Partei{Name:varchar(100), Anschrift:varchar(100)}  
Fraktion{Name:varchar(100)}  
Ort{PLZ:varchar(5), Ort:varchar(100)}
```

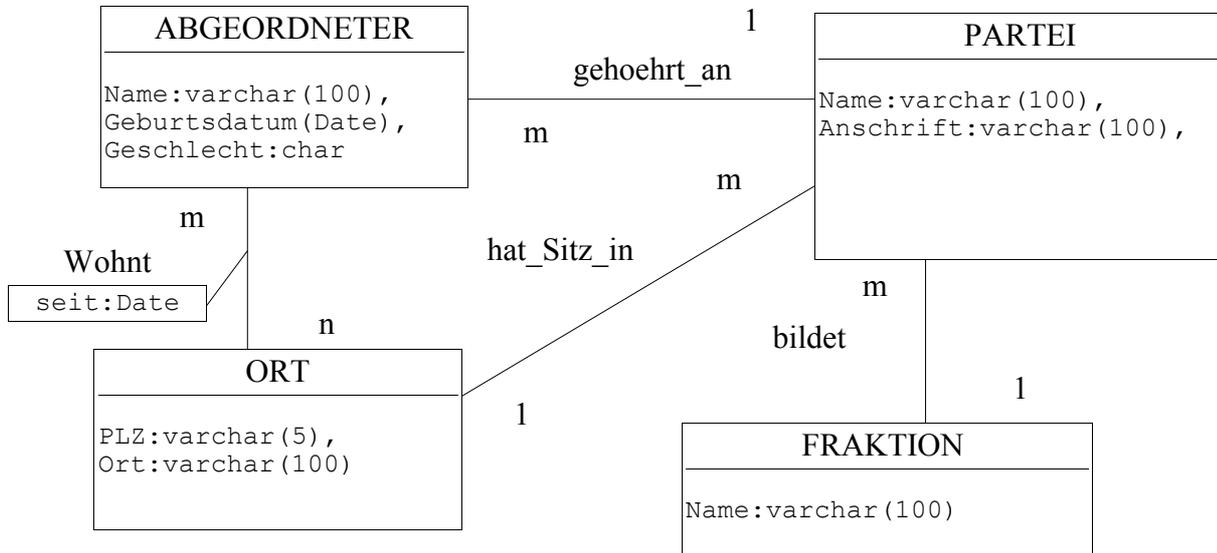
Es wird ausreichen, folgende Datentypen zu kennen:

varchar(100)	Zeichenfolge der Länge 100
int(11)	ganze Zahl (benötigter Speicherplatz 8 Byte)
Date	Datum im Format JJJJ-MM-TT
Decimal(10,2)	Dezimalzahl mit zwei Nachkommastellen
Double(16)	(Fließ-)Kommazahl



Anmerkung: Bei der Postleitzahl wird nicht „int“ verwendet, da es Postleitzahlen gibt, die mit 0 beginnen und diese führende 0 bei Verwendung einer Ganzzahl wegfallen würde.

4. / 5.



II. Erstellen des Logischen Modells

Es gilt nun, das Semantische Modell in ein Logisches Modell umzuwandeln!
Also eine Anzahl von Tabellen zu erstellen.

1. Wandle die Klassen zu Tabellen um, der Tabellename ergibt sich aus dem Klassennamen, jede Tabelle hat eine Spalte mit dem Namen „ID“, die den Primärschlüssel der Tabelle bildet und jedes Attribut bildet eine weitere Spalte.

<i>Abgeordneter</i>	<u><i>ID</i></u>	<i>Name</i>	<i>Geburtsdatum</i>	<i>Geschlecht</i>

<i>Partei</i>	<u><i>ID</i></u>	<i>Name</i>	<i>Anschrift</i>

<i>Ort</i>	<u><i>ID</i></u>	<i>PLZ</i>	<i>Name</i>



<i>Fraktion</i>	<i><u>ID</u></i>	<i>Name</i>

2. Wandle jede m-n-Beziehung in eine eigene Tabelle um. Die Tabelle hat den Namen der Beziehung und mindestens zwei Spalten, nämlich die beiden Primärschlüssel der verbundenen Klassen. Diese bilden in der Tabelle Fremdschlüssel und erhalten daher einen neuen Namen, nämlich <Klassenname>Nr. Außerdem können weitere Spalten dadurch entstehen, dass Attribute an der Beziehung angeheftet wurden.

<i>Wohnt</i>	<i><u>OrtNr</u></i>	<i><u>AbgeordneterNr</u></i>	<i>Seit</i>

3. Aus 1-m-Beziehungen müssen keine eigenen Tabellen erstellt werden, es genügt die Tabelle zu erweitern, an der die Kardinalität „m“ ermittelt wurde. Erweitert wird diese Tabelle um den Fremdschlüssel.

<i>Abgeordneter</i>	<i><u>ID</u></i>	<i>Name</i>	<i>Geburtsdatum</i>	<i>Geschlecht</i>	<i>ParteiNr</i>

<i>Partei</i>	<i><u>ID</u></i>	<i>Name</i>	<i>Anschrift</i>	<i>OrtNr</i>	<i>FraktionNr</i>

Nun sind alle Tabellen erstellt, was man auch als „**Logisches Modell**“ der Datenbank bezeichnet.

III. Abfragen

Mit Hilfe der Sprache SQL (Structured Query Language) können nun Informationen aus der Datenbank abgefragt werden.

Die Abfragen erfolgen stets nach folgendem Schema:

```
SELECT      Spaltenname FROM      Tabellename
```

und können aber noch erweitert werden um

```
WHERE      Auswahl bestimmter Zeilen
```

```
ORDER BY  Reihenfolge
```

```
AS        anderer Name
```

```
GROUP BY  in Zusammenhang mit count(*), MIN(*), MAX(*), AVG(*) und SUM(*)
```



Beispiel:

<i>Ort</i>	<i>ID</i>	<i>PLZ</i>	<i>Name</i>
	1	1000	Berlin
	2	95028	Hof

<i>Abgeordneter</i>	<i>ID</i>	<i>Name</i>	<i>Geburtsdatum</i>	<i>Geschlecht</i>	<i>ParteiNr</i>
	1	Paul Dampf	1950-10-12	M	1
	2	Monika Muster	1970-05-31	W	2

<i>Wohnt</i>	<i>OrtNr</i>	<i>AbgeordneterNr</i>	<i>Seit</i>
	2	2	1980-02-15
	1	2	2004-06-17
	2	1	1950-10-12

Beispiel 1 - Namen aller Abgeordneter: `SELECT Name FROM Abgeordneter;`

<i>Name</i>
Paul Dampf
Monika Muster

Beispiel 2 - Name und Geschlecht: `SELECT Name, Geschlecht FROM Abgeordneter;`

<i>Name</i>	<i>Geschlecht</i>
Paul Dampf	M
Monika Muster	W

Beispiel 3 - Name und Geschlecht sortiert nach Name:
`SELECT Name, Geschlecht FROM Abgeordneter ORDER BY Name;`

<i>Name</i>	<i>Geschlecht</i>
Monika Muster	M
Paul Dampf	W

Beispiel 4 - Wie viele Abgeordnete wohnen in jedem Ort:
`SELECT OrtNr, count(*) AS Anzahl FROM Wohnt GROUP BY OrtNr;`

<i>OrtNr</i>	<i>AbgeordneterNr</i>
1	1
2	2



Bei Abfragen über mehr als einer Tabelle sind alle betroffenen Tabellen (die auf dem Weg liegen, also die im Semantischen Modell zwischen den Klassen liegen) mit einzubeziehen. Die Fremdschlüssel der beteiligten Klassen/Beziehungen sind jeweils gleichzusetzen!

Beispiel 5 - Gib die Namen Aller Abgeordneten der Partei 2 aus und den Namen des Ortes in dem sie wohnen, sortiert nach Namen und bei Namensgleichheit auch nach Ortsname, der Ortsname soll die Überschrift Ort haben:

```
SELECT Abgeordneter.Name, Ort.Name AS Ort FROM Abgeordneter, Wohnt, Ort
WHERE ParteiNr=2 AND Abgeordneter.ID=wohnt.AbgeordneterNr AND
wohnt.OrtNr=Ort.ID ORDER BY Abgeordneter.Name, Ort.Name;
```

<i>Name</i>	<i>Geburtsdatum</i>
Monika Muster	Berlin
Monika Muster	Hof

Anmerkung: Wenn alle Spalten ausgewählt werden sollen kann das Wildcard „*“ verwendet werden.