

1 Vertraut werden und Probleme erkennen

In diesem Kapitel wirst du an Hand eines ersten, nur auf den ersten Blick einfach erscheinenden Beispiels mit den Ideen einer Datenbank vertraut machen. Ich werde so wenig wie möglich auf die Theorie der Datenmodellierung eingehen, so wenig wie möglich die korrekten Fachbegriffe verwenden. Du wirst intuitiv die Probleme selbst erkennen und somit die Einsicht gewinnen, dass ohne theoretische Kenntnisse eine solide Datenbank nicht zu erstellen ist. Wahrscheinlich wirst du in diesem Kapitel auch nicht alle Zusammenhänge verstehen. Versuche jedoch, die Ideen nachzuvollziehen. Die korrekten Fachbegriffe, Gesetzmäßigkeiten und die Theorie einer Datenbank wirst du in den nachfolgenden Kapiteln erarbeiten.

1.1 Die Datenbank PersonSprichtSprache 01

Als erste Aufgabe erstellst du eine Datenbank, in der die Schüler einer Schulklasse mit Nachnamen und Vornamen aufgelistet werden. Zusätzlich soll erfasst werden, welche Sprache diese Schüler lernen.

Um diese Aufgabe zu erledigen, legst du am besten eine Tabelle an:

Nachname	Vorname	Sprache
Drewnick	Robert	Englisch/Französisch
Kerpl	Denise	Indisch/Spanisch
Haag	Florian	Deutsch/Englisch
Schwaner	Nico	Englisch/Deutsch
Haag	Sebastian	Französisch/Deutsch
Hermann	Sandra	Deutsch/Englisch

Abbildung 1.1: PersonSprichtSprache01

An Hand dieser Tabelle lassen sich folgende Fragen leicht beantworten:

- Welche Sprachen spricht Haag?
- Welche Personen sprechen Deutsch/Englisch?
- Welche Personen sprechen Englisch/Deutsch?

Die letzten beiden Fragen lassen bereits erkennen, dass diese Tabelle nicht so ganz problemlos angelegt worden ist. Lass dich vorerst aber nicht beirren. Die Abbildung 1.2 zeigt eine erste, vereinfachte Informationsstruktur.

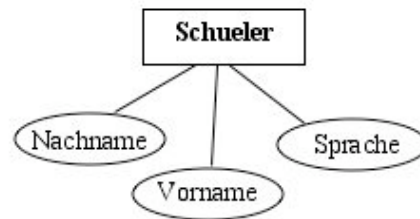


Abbildung 1.2: Erste Informationsstruktur der Klasse *Schueler*

Die zentrale Klasse der Datenbank *PerSpr01* ist die Klasse *Schueler*. Diese hat die Attribute *Nachname*, *Vorname* und *Sprache*. Die Attribute der Objekte dieser Klasse *Schueler* werden in der Datenbank in einer Tabelle gespeichert. Die Zellen dieser Tabelle geben die Attributwerte eines bestimmten Objekts wider. Die Spalten dieser Tabelle entsprechen den Attributen und sollten daher die gleiche Bezeichnung haben.

1.1.1 Erstellen der Datenbank

Deine erste Datenbank *PerSpr01* wird noch recht einfach gestaltet sein und besteht vorerst nur aus einer Tabelle *Schueler*.

Zuerst startest du das Programm DATENBANK und erstellst mit Hilfe des Punkts *Neue Datenbank erstellen* eine neue Datenbank und speicherst diese unter dem Namen *PerSpr01*. Anschließend öffnet sich ein Dialogfenster *PerSpr01 : NeoOffice Base* Hier wählst du in der linken Objektleiste *Tabellen* und anschließend *Tabelle in der Entwurfsansicht*....

Abbildung 1.3 zeigt die erforderlichen Einträge für die Attribute und Datentypen.

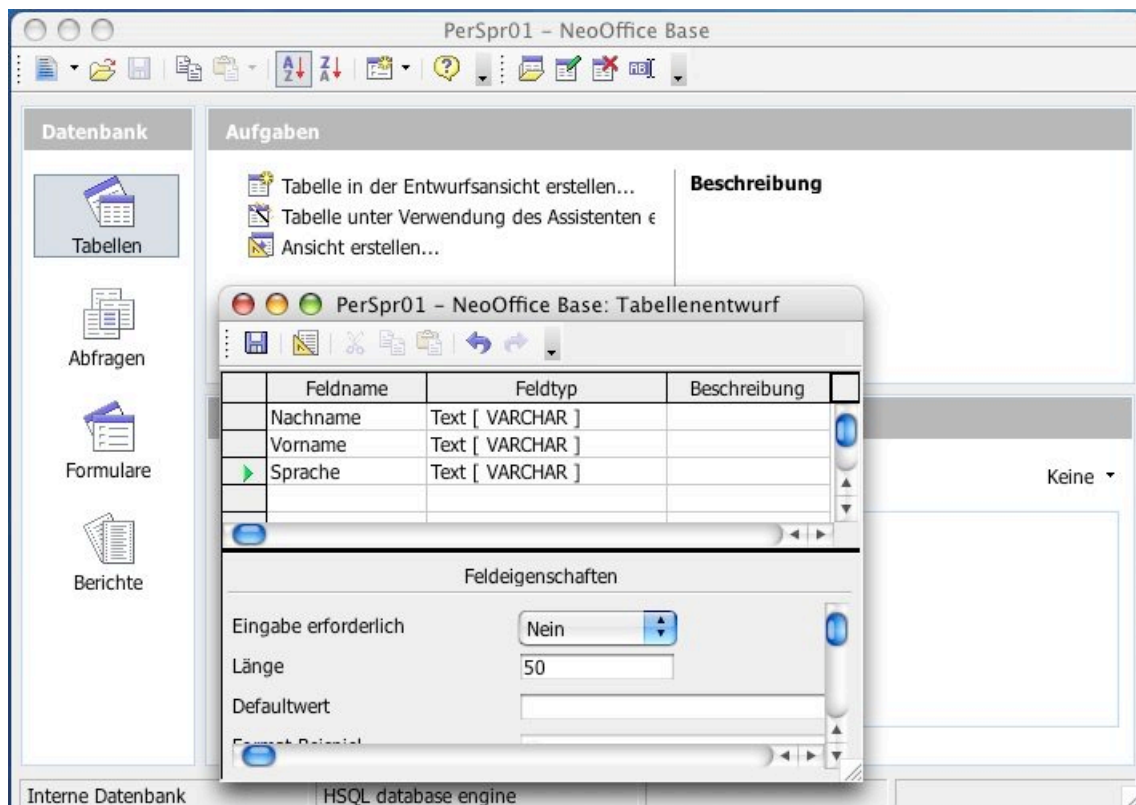


Abbildung 1.3: Entwurfsansicht der Klasse *Schueler*

Anschließend speicherst du die Klasse unter dem Namen *tblSchueler*. Den Hinweis *Soll ein Primärschlüssel erzeugt werden?* beantwortest du einfach mit *Ja* und stellst auch den Auto-Wert auf *Ja*. Ansonsten ignorierst aber für den Anfang diese neu eingefügte Zeile *ID*.

Schueler	
Nachname	
Vorname	
Sprache	

Feldname	Feldtyp
Nachname	Text [VARCHAR]
Vorname	Text [VARCHAR]
Sprache	Text [VARCHAR]

Abbildung 1.4: Die Klasse *Schueler*

Nun hast du die Vorbereitungen abgeschlossen und kannst anschließend Objekte der Klasse *Schueler* erzeugen. Alle diese Objekte haben die in Abbildung 1.4 dargestellten Attribute, unterscheiden sich jedoch in ihren Attributwerten.

Dazu wechselst du, wie in Abbildung 1.5 angedeutet, mit einem Klick in das Icon *Datenbankobjekt öffnen* oder auf das *Tabellensymbol* in die *Datenblattansicht* und gibst die in Abbildung 1.1 dargestellten Objekte in die Tabelle ein: Diese Objekte werden in relationalen Datenbanken als Zeilen einer Tabelle dargestellt.

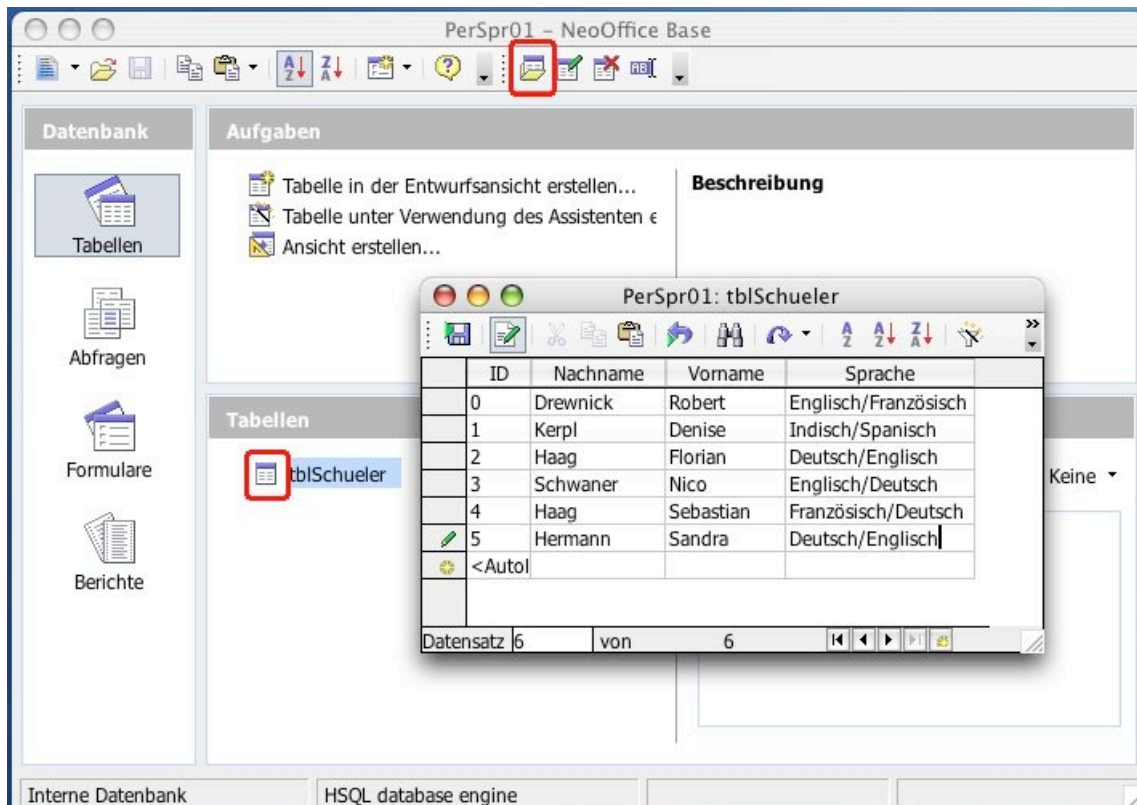


Abbildung 1.5: Die Zeilen stellen Objekte der Klasse *Schueler* dar

Du hast nun in *NeoOffice* eine kleine Datenbank *PerSpr01* erstellt, die vorläufig nur aus einer einzigen Tabelle *tblSchueler* besteht. Trotzdem kannst du nun die ersten Abfragen an diese Datenbank richten.

Bemerkung:

In objektorientierten Programmiersprachen wie Java haben diese Objekte eindeutige Bezeichner. In einer Tabelle sind aber diese Bezeichner nicht mehr vorhanden. Daher können die verschiedenen Objekte ausschließlich nach den Werten ihrer Attribute unterschieden werden. Es muss also mindestens ein Attribut existieren, in dem sich alle Objekte unterscheiden. Dieses Attribut (oder auch eine Gruppe von Attributen) bezeichnet man als **Schlüssel**.

Konvention:

Eine Standardisierung in Access sind die so genannten Reddick-VBA-Namenskonventionen (kurz: RVBA-Konventionen). Sie sind nach Greg Reddick, der einige Jahre im Access-Entwicklungsteam bei Microsoft gearbeitet hat, benannt worden. Um bereits an ihrem Namen beispielsweise Tabellen von Abfragen oder Formularen unterscheiden zu können, erhalten alle diese Datenbankobjekte eine Vorsilbe aus drei Buchstaben:

Datenbankobjekt	Vorsilbe	Englischer Name
Tabellen	tbl	tables

Abfragen	qry	queries
Formulare	frm	forms
Berichte	rpt	reports

1.1.2 Erste einfache Datenbankabfragen (Projektion)

Fragen an eine Datenbank werden in der »Programmiersprache« SQL (Structured Query Language) gerichtet. Diese ist derzeit die am weitesten verbreitete Sprache für relationale Datenbanken. Auch in Zukunft wird diese Sprache keine Konkurrenz bekommen, was sich aus der Tatsache ableiten lässt, dass mittlerweile alle Hersteller relationaler Datenbanken ihr System auf diese Sprache umgestellt haben.

Kleine SQL-Geschichte

Anfang der siebziger Jahre wurde in den IBM Forschungslaboratorien in San Jose, Kalifornien, ein Forschungsprojekt begonnen, das sich »System R« nannte. Es sollte die Praktikierbarkeit der relationalen Theorien untersuchen. Von den IBM-Mitarbeitern R.F. Boyce und D.D. Chamberlain wurde die Sprache SEQUEL (sprich: siequel) entwickelt, die später in SQL umbenannt wurde. Man lehnte hierbei die Syntax an Begriffe der englischen Umgangssprache wie z.B. SELECT, FROM, WHERE an. Seit dieser Zeit wurden von fast allen DB-Herstellern SQL-Schnittstellen zu ihren relationalen und nichtrelationalen Datenbanksystemen entwickelt.

Nun wirst du an die Datenbank folgende Abfrage stellen: „Wie lauten die Nachnamen aller Schüler?“

Für diese Abfrage musst du zuerst einige Vorbereitungen treffen:

1. Wähle in der Objekteiste *Abfragen*.
2. Klicke auf *Abfrage in der SQL-Ansicht erstellen....*
3. Gib die entsprechenden SQL-Befehle in das SQL-Fenster ein.
4. Klicke in der Symbolleiste auf *Ausführen*.
5. Sichere deine Abfrage mit einem sinnvollen Namen *qryXxxxx*.

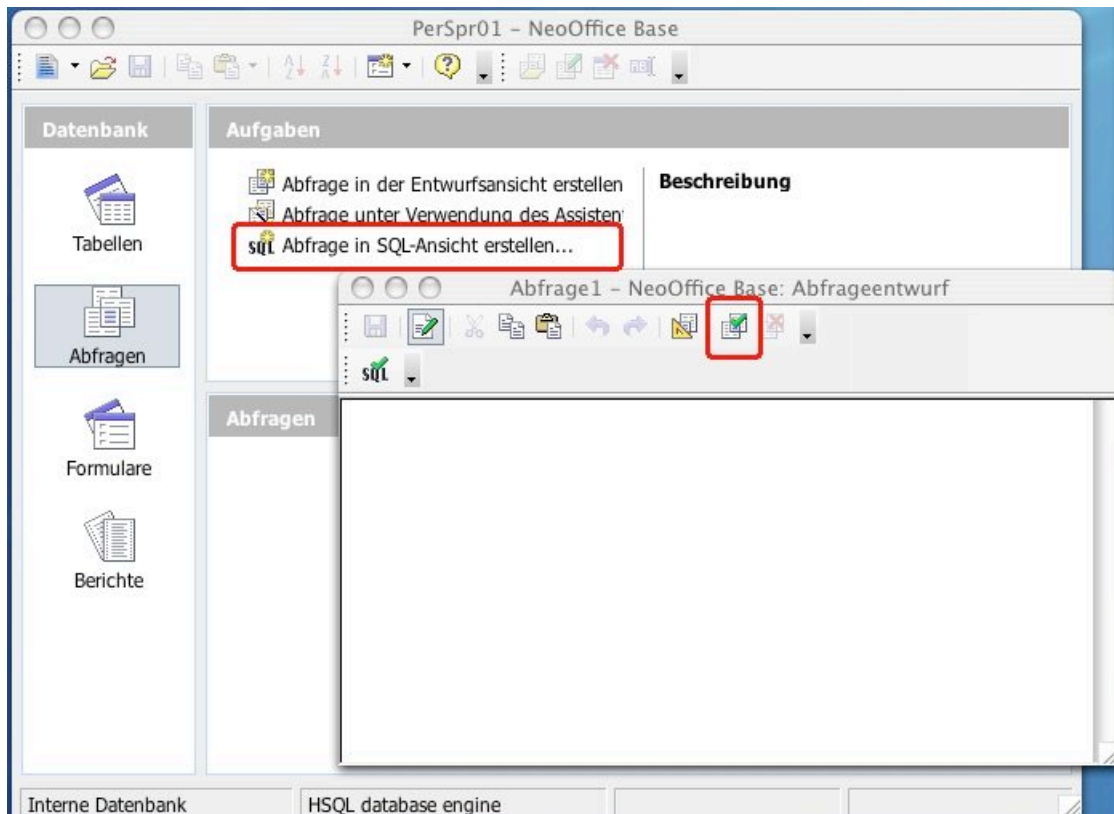


Abbildung 1.6: Eingabefenster für SQL-Abfragen

Wie Abbildung 1.7 zeigt, stehen die Nachnamen aller Schüler in der ersten Spalte *Nachname* der Tabelle *tblSchueler*. Somit ist die SQL-Anweisung recht einfach zu verstehen.

ID	Nachname	Vorname	Sprache
0	Drewnick	Robert	Englisch/Französisch
1	Kerpl	Denise	Indisch/Spanisch
2	Haag	Florian	Deutsch/Englisch
3	Schwaner	Nico	Englisch/Deutsch
4	Haag	Sebastian	Französisch/Deutsch
5	Hermann	Sandra	Deutsch/Englisch

Abbildung 1.7: Nachnamen aller Schüler

Auswahl einer Spalte einer Tabelle:

Syntax: **SELECT <Spalte>**
FROM <Tabelle>

Du gibst in das Fenster *Abfrage1 : Auswahlabfrage* die Anweisung

```
SELECT Nachname
FROM tblSchueler
```

ein. Abbildung 1.10 zeigt die SQL-Anweisung und das Ergebnis der Ausführung.

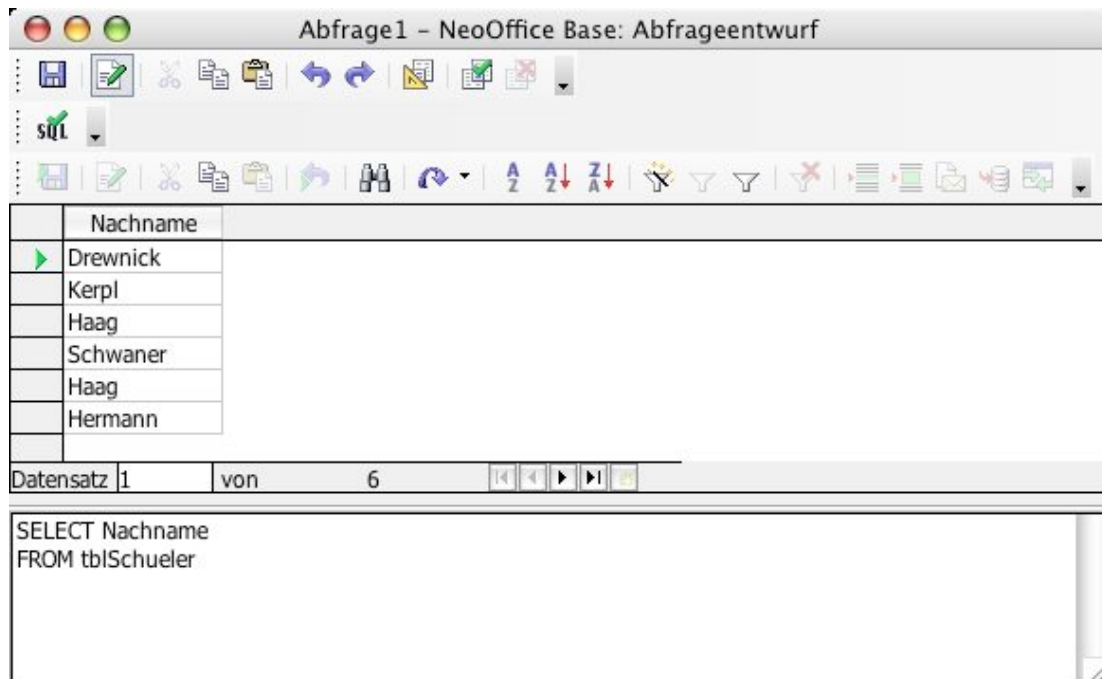


Abbildung 1.8: Eingabe der SQL-Anweisung und deren Ausführung

Speichere nun deine Abfrage unter dem Namen *qryNachnamen*. Mit Hilfe der unten angegebenen SQL-Anweisungen kannst du auch weitere Daten aus der Tabelle abfragen.

Auswahl mehrerer Spalten einer Tabelle:

Syntax: **SELECT <Spalte1> , <Spalte2> , ...**
 FROM <Tabelle>

Auswahl aller Spalten einer Tabelle:

Syntax: **SELECT ***
 FROM <Tabelle>

Übung 1.1:

Speichere die folgenden Abfragen immer unter einem geeigneten Namen beispielsweise in der Form *qry11aVorNachname*.

- Gesucht sind die Vornamen und Nachnamen aller Schüler.
- Gesucht sind die Nachnamen und Vornamen aller Schüler.
- Gesucht sind alle Daten aller Schüler.

Abfragen, die Spalten einer Tabelle ergeben, heißen *Projektion*.

1.1.3 Etwas komplexere Datenbankabfragen (Selektion)

Nun sollst du herausfinden, welche Sprache die Schülerin Kerpl spricht. Abbildung 1.11 zeigt, dass hier nun nach einer Zeile der Tabelle gefragt ist. Gesucht ist also der Attributwert *Sprache* eines Objekts der Klasse *Schueler*, dessen Attributwert *Nachname* gleich „Kerpl“ ist.



ID	Nachname	Vorname	Sprache
0	Drewnick	Robert	Englisch/Französisch
1	Kerpl	Denise	Indisch/Spanisch
2	Haag	Florian	Deutsch/Englisch
3	Schwamer	Nico	Englisch/Deutsch
4	Haag	Sebastian	Französisch/Deutsch
5	Hermann	Sandra	Deutsch/Englisch

Abbildung 1.9: Auswahl eines Objekts (Datensatzes)

Syntax: **SELECT** <Spalte>
FROM <Tabelle>
WHERE <Bedingung>

Ein Vergleich mit der Syntax dieser Abfrage liefert die SQL-Anweisung im Fenster *Abfrage1*: *Auswahlabfrage* die Anweisung

```
SELECT Sprache
FROM tblSchueler
WHERE Nachname = 'Kerpl'
```

Abbildung 1.12 zeigt das Ergebnis dieser Abfrage.

Sprache
Indisch/Spanisch

Datensatz 1 von 1

Abbildung 1.10: Ergebnis der Abfrage “Welche Sprache spricht Kerpl?”

Abfragen, die Zeilen einer Tabelle ergeben, heißen *Selektion*.

Übung 1.2:

- Füge zusätzlich in der Abfrage den Nachnamen.
- Welche Personen sprechen „Englisch/Deutsch“?
- Welche Personen sprechen „Deutsch/Englisch“?

Sprache	Nachname	Vorname
Deutsch/Englisch	Haag	Florian
Deutsch/Englisch	Hermann	Sandra

Datensatz 2 von 2

Abbildung 1.11: Ergebnis der Abfrage “Welche Personen sprechen Deutsch/Englisch?”

Übung 1.3:

Welche Person spricht Deutsch?

Die Abfrage „Welche Person spricht Deutsch?“ zeigt leider keine Person an, obwohl vier Personen die Sprache Deutsch beherrschen.

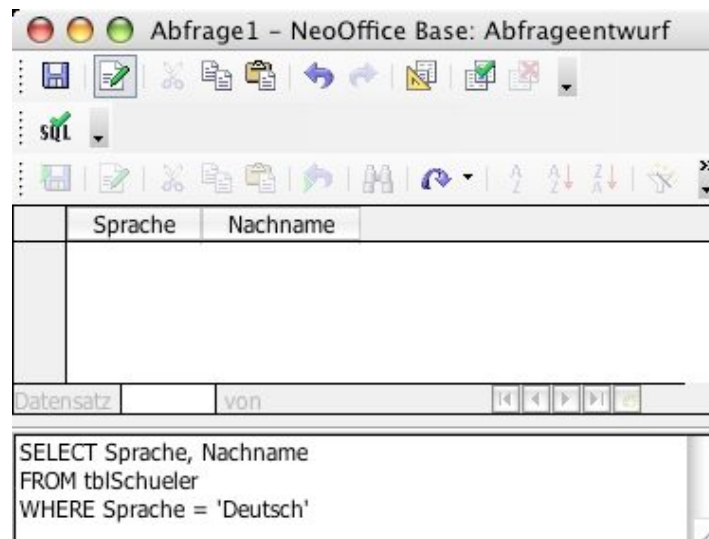


Abbildung 1.12: Ergebnis der Abfrage “Welche Person spricht Deutsch?”

Es wird nun offensichtlich, dass der erste Entwurf der Datenbank *PerSpr01* einen schwerwiegenden Fehler aufweist. Es war nicht sehr geschickt, die beiden Sprachen in einer Spalte zusammen zu fassen.

1.2 Die Datenbank *PersonSprichtSprache02*

Deswegen wirst du in der nächsten Datenbank *PerSpr02* neben *Nachname* und *Vorname* in der Klasse *Schueler* die beiden Attribute *Sprache1* und *Sprache2* einführen.

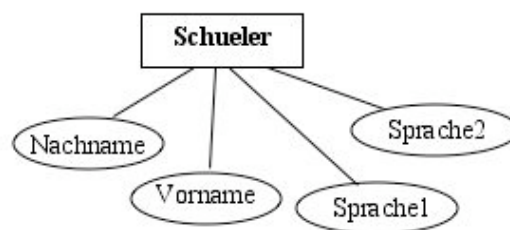


Abbildung 1.13: Informationsstruktur der Klasse *Schueler*

Schueler
Nachname
Vorname
Sprache1
Sprache2

Feldname	Feldtyp
Nachname	Text [VARCHAR]
Vorname	Text [VARCHAR]
Sprache1	Text [VARCHAR]
Sprache2	Text [VARCHAR]

Abbildung 1.14: Die Klasse *Schueler*

Übung 1.4:

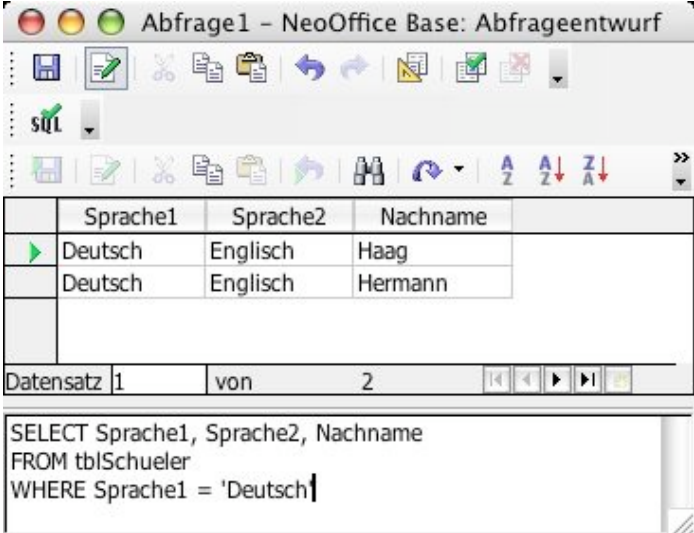
Erstelle eine neue Datenbank *PerSpr02* und in dieser eine Klasse *Schueler* mit den Attributen *Nachname*, *Vorname*, *Sprache1* und *Sprache2*. Den Hinweis *Soll ein Primärschlüssel erzeugt werden?* beantwortest du einfach mit *Ja* und stellst auch den Auto-Wert auf *Ja*. Ansonsten ignorierst aber für den Anfang diese neu eingefügte Zeile *ID*. Erzeuge anschließend die in Abbildung 1.15 dargestellten Objekte.



ID	Nachname	Vorname	Sprache1	Sprache2
0	Drewnick	Robert	Englisch	Französisch
1	Kerpl	Denise	Indisch	Spanisch
2	Haag	Florian	Deutsch	Englisch
3	Schwaner	Nico	Englisch	Deutsch
4	Haag	Sebastian	Französisch	Deutsch
5	Hermann	Sandra	Deutsch	Englisch
<AutoI				

Abbildung 1.15: Datenblattansicht von *tblSchueler*

Mit der unten dargestellten Abfrage *qryD1* erhalten wir die beiden Personen, die als erste Sprache Deutsch beherrschen.



Sprache1	Sprache2	Nachname
Deutsch	Englisch	Haag
Deutsch	Englisch	Hermann

```

SELECT Sprache1, Sprache2, Nachname
FROM tblSchueler
WHERE Sprache1 = 'Deutsch'

```

Abbildung 1.16: Abfrage “Wer spricht Deutsch (1.Sprache)?“

Übung 1.5:

Stelle eine Abfrage *qryD2*, die beiden anderen Personen mit Deutsch als zweite Sprache liefert.

Wie lautet nun eine Abfrage, die alle vier Schüler liefert, die Deutsch sprechen? Du könntest folgende SQL-Anweisung ausprobieren:

```
SELECT Sprache1, Sprache2, Nachname
FROM tblSchueler
WHERE Sprache1 = „Deutsch“ AND Sprache2 = „Deutsch“
```

Übung 1.6:

Stelle die obige Abfrage *qryD/D*.

Formuliere schriftlich, warum diese Abfrage keine Person liefert.

Versuche eine Abfrage zu formulieren, die das gewünschte Ergebnis liefert.

Eine unbefriedigende Lösung wäre zu fordern, Deutsch immer als erste Sprache einzugeben. Aber man erkennt sofort, dass dies zu Konflikten mit den anderen Sprachen führt. Weiterhin existieren bestimmt auch Personen, die mehr als zwei Sprachen beherrschen. Wie lassen diese sich in einer Datenbank verwalten?

1.3 Die Datenbank *PersonSprichtSprache03*

Nun siehst du hoffentlich ein, dass du noch tiefer in die Theorie der Datenbank eintauchen musst. Das Problem *Person spricht Sprache* wirst du nun doch genauer analysieren müssen (später wirst du von einer Normalform sprechen):

Nachname	Vorname	Sprache1	Sprache2
Drewnick	Robert	Englisch	Französisch
Kerpl	Denise	Indisch	Spanisch
Haag	Florian	Deutsch	Englisch
Schwaner	Nico	Englisch	Deutsch
Haag	Sebastian	Französisch	Deutsch
Hermann	Sandra	Deutsch	Englisch

Abbildung 1.17: Tabelle der Schüler mit ihren Sprachen

Die in Abbildung 1.17 gezeigte Tabelle enthält bei genauerer Betrachtung zwei Klassen. Es existiert eine Klasse *Schueler* mit den beiden Attributen *Nachname* und *Vorname* und eine Klasse *Fremdsprache* mit dem Attribut *Sprache*.

Dies führt zu einer Trennung der obigen Tabelle in zwei Tabellen, die jeweils die entsprechenden Schülerobjekte und Fremdspracheobjekte enthalten.

Nachname	Vorname	Sprache
Drewnick	Robert	Englisch
Kerpl	Denise	Deutsch
Haag	Florian	Indisch
Schwaner	Nico	Spanisch
Haag	Sebastian	Französisch
Hermann	Sandra	

Abbildung 1.18: Trennung in zwei Tabellen

Du hast also zunächst eine Spalte weniger. Dies macht das Ganze übersichtlich. Allerdings musst du nun eine Beziehung zwischen diesen beiden Tabellen erstellen mit Hilfe von Pfeilen:

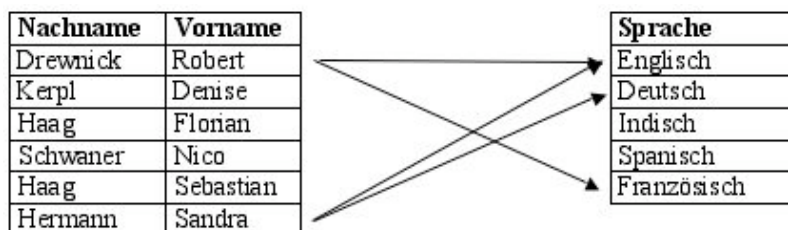


Abbildung 1.19: Beziehungen zwischen *Schüler* und *Fremdsprache*

Übung 1.7:

Vervollständige die obige Beziehung *Schueler* und *Fremdsprache*.

Diese auf den ersten Blick recht einfach erscheinende Beziehung *Person spricht Sprache* ist nun sehr kompliziert geworden. Von jedem Objekt der Klasse *Schueler* gehen zwei Pfeile aus und bei jedem Objekt der Klasse *Fremdsprache* können mehrere Pfeile enden. Und in deiner Klasse sollte eigentlich jeder Schüler drei Sprachen können, das Diagramm würde also noch komplizierter.

Das Problem ist folgendes, dass von jeder Zeile der linken Tabelle mehrere Pfeile ausgehen **und gleichzeitig** bei jeder Zeile der rechten Tabelle mehrere Pfeile enden. Eine saubere Lösung dieses Problems ist die Einführung einer dritten Tabelle, die die Zuordnung zwischen *Schueler* und *Fremdsprache* regelt. Somit schaffst du auch Ordnung in dieses Pfeilchaos.

Da jeder Computer leichter mit Zahlen rechnet, ist es sinnvoll, zuerst jedem Schüler und jeder Sprache eine Nummer geben, die diese eindeutig identifiziert, also eine *SchuelerID* und eine *SpracheID* (man bezeichnet eine solche eindeutig Identifizierung auch als **Primärschlüssel**). Mit Hilfe dieser beiden Identitätsnummern erstellst du eine dritte Klasse *Spricht*. Die zugehörigen Objekte werden in der Tabelle *tblSpricht* aufgelistet.

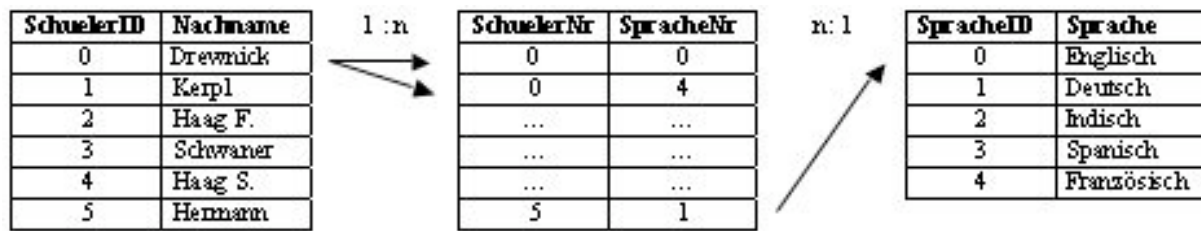


Abbildung 1.20: Eine Zuordnungstabelle schafft Ordnung im „Pfeilchaos“

Übung 1.8:

Vervollständige die obige Beziehung *Person spricht Sprache*, indem du alle Pfeile einzeichnest.

Überlege dir, was die Bezeichnung 1:n bzw. n:1 bedeuten könnte.

Du erstellst nun eine neue Datenbank *PerSpr03* mit den drei Klassen *Schueler*, *Spricht* und *Fremdsprache*. Das bedeutet, du hast drei Tabellen *tblSchueler*, *tblSpricht* und *tblFremdsprache*. Die *SchuelerID* und *SpracheID* sollten den Typ **Integer** mit **Auto-Wert Ja** besitzen und **Primärschlüssel** sein.

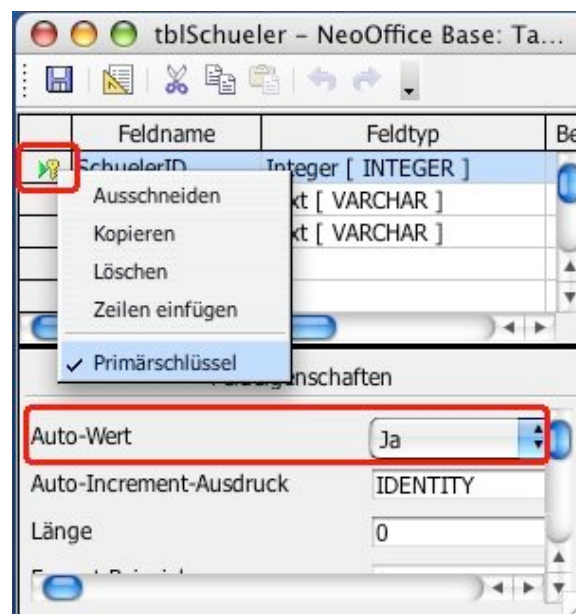


Abbildung 1.21: Primärschlüssel in der Tabelle *tblSchueler*

In der Zuordnungstabelle *tblSpricht* sollten *PersonNr* und *SpracheNr* vom Datentyp **Integer** mit **Auto-Wert Nein** sein und zusammen den **Primärschlüssel** bilden.

The image shows three database tables from the 'PerSpr03' database:

- tblSchueler:**

SchuelerID	Nachname	Vorname
0	Drewnick	Robert
1	Kerpl	Denise
2	Haag	Florian
3	Schwaner	Nico
4	Haag	Sebastian
5	Hermann	Sandra
- tblSpricht:**

SchuelerNr	SpracheNr
0	0
0	4
1	2
1	3
2	0
2	1
3	0
3	1
4	1
4	4
5	0
5	1
- tblFremdsprache:**

SpracheID	Sprache
0	Englisch
1	Deutsch
2	Indisch
3	Spanisch
4	Französisch

Abbildung 1.22: Tabellen der Datenbank *PerSpr03*

Frage nun nach den Sprachen, welche der Schüler Drewnick beherrscht.

The screenshot shows a query result window titled 'Abfrage1 - NeoOffice Base: Abfrageentwurf'. The query is:

```
SELECT Nachname, Vorname, Sprache
FROM tblSchueler, tblFremdsprache
WHERE Nachname = 'Drewnick'
```

The result table shows the following data:

Nachname	Vorname	Sprache
Drewnick	Robert	Englisch
Drewnick	Robert	Deutsch
Drewnick	Robert	Indisch
Drewnick	Robert	Spanisch
Drewnick	Robert	Französisch

Abbildung 1.23: Erster Versuch der Abfrage „Welche Sprache spricht Drewnick?“

Das richtige Ergebnis wäre jedoch Englisch und Französisch. Hier werden leider alle möglichen Sprachen aufgelistet.

Übung 1.9:

Versuche eine Abfrage zu erstellen, die nach der Sprache eines Schülers Haag sucht.

Überlege dir schriftlich, warum hier alle erfassten Sprachen aufgelistet werden.

Übung 1.10:

Versuche eine Abfrage zu erstellen, die diejenigen Schüler auflisten soll, die die Sprache Deutsch sprechen.

Überlege dir schriftlich, warum hier alle erfassten Schüler aufgelistet werden.

1.4 Die Datenbank *PersonSprichtSprache04*

Vielleicht erahnst du bereits, dass die obigen Beispiele deswegen unerwünschte Ergebnisse liefern, weil die Beziehungen zwischen den Tabellen noch nicht erstellt worden sind. Du musst alle drei Tabellen miteinander verknüpfen, ähnlich wie es bereits in den Übungen mit den Pfeilen durchgeführt wurde. Nur so gelangt man von einem Schüler über die Zuordnung zur Fremdsprache.

Erstelle nun eine neue Datenbank *PerSpr04* mit den Tabellen *tblSchueler*, *tblSpricht* und *tblFremdsprache*. Setze die korrekten Primärschlüssel und gib die Datensätze ein.

Nun klickst du auf *Extras > Beziehungen...*, es öffnen sich folgende beiden Fenster:

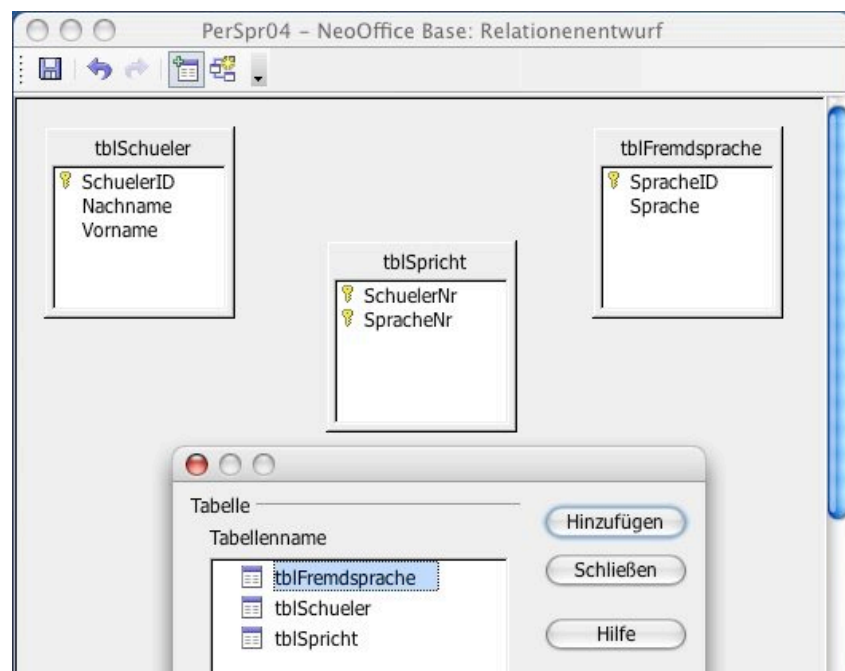


Abbildung 1.24: Das Fenster „Beziehungen“

Mit *Hinzufügen* werden alle notwendigen Tabellen in das Fenster *Relationenentwurf* geladen. Nun wird mit gedrückter Maustaste von *SchuelerNr* auf *SchueleID* gezogen.

Die Verknüpfung der Tabellen *tblSpricht* und *tblFremdsprache* erfolgt analog.

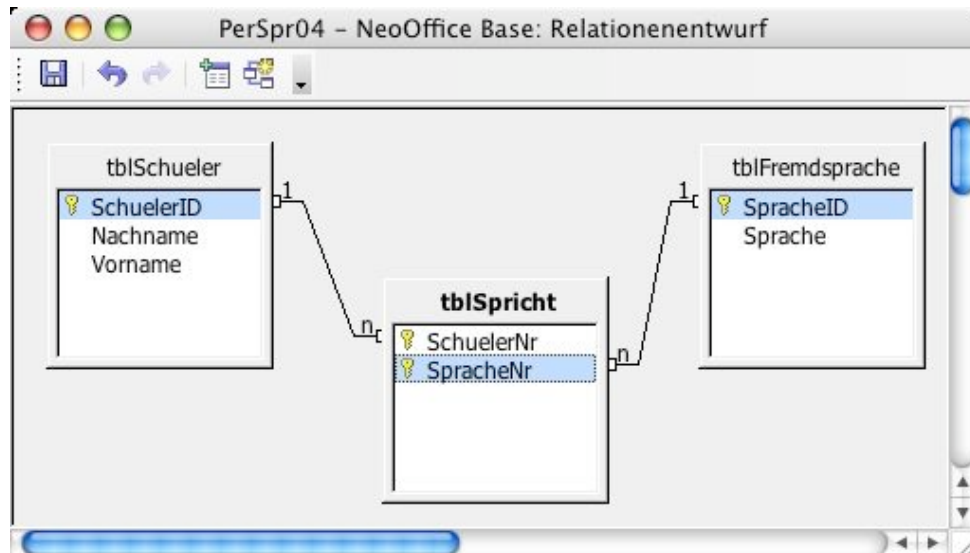


Abbildung 1.25: Verknüpfte Tabellen im Fenster „Beziehungen“

Das Bild dieser Beziehungen zwischen den drei Tabellen erinnert an die Übung 1.8.

Nun kannst du wieder die Abfrage „Welche Sprache spricht Drewnick?“ stellen. Allerdings ist die SQL-Anweisung nun komplizierter geworden.

```
SELECT Nachname, Sprache
FROM tblSchueler, tblSpricht, tblFremdsprache
WHERE (Nachname = „Drewnick“) AND (SchuelerID = SchuelerNr)
      AND (SpracheNr = SpracheID)
```

Du musst dich sozusagen durch alle Tabellen „hindurchhangeln“. Hierzu verwendest du die entsprechenden Primärschlüssel und Fremdschlüssel. Wie Abbildung 1.26 zeigt, wird das erwartete Ergebnis geliefert.

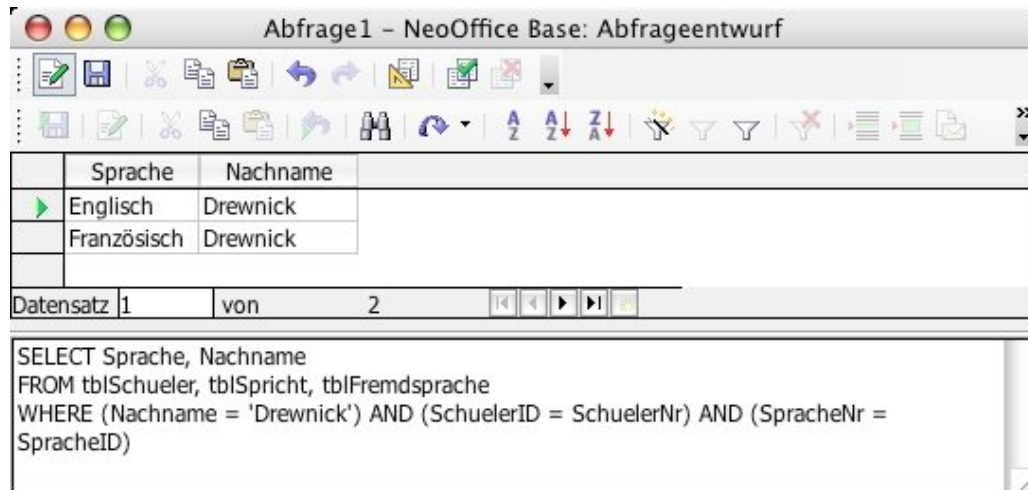


Abbildung 1.26: Zweiter Versuch der Abfrage „Welche Sprache spricht Drewnick?“

Übung 1.11:

Erstelle eine Abfrage, die nach der Sprache eines Schülers Haag sucht.

Übung 1.12:

Erstelle eine Abfrage, die diejenigen Schüler auflisten soll, die die Sprache Deutsch sprechen.

1.5 Die Datenbank PersonSprichtSprache05

Erstelle eine neue Datenbank *PerSpr05* wie bereits bei *PerSpr04* beschrieben. Füge nun noch weitere Personen und weitere Sprachen hinzu. Es sollen nun auch Personen existieren, die keine Sprache beherrschen und auch Personen, die mehr als zwei Sprachen beherrschen. Weiterhin sollen auch Sprachen aufgenommen werden, die von keiner Person gesprochen werden.

Übung 1.13:

- Welche Sprache spricht Drewnick?
- Welche Sprache spricht Haag?
- Welche Schüler sprechen Deutsch?

Die folgenden Übungen sind bereits recht kompliziert. Zu diesem Zeitpunkt sollen diese als Herausforderung gesehen werden. Jedoch wenn du im Laufe des Schuljahres das Kapitel **Abfragen mit SQL** bearbeitet hast, sollten diese Übungen lösen können.

Übung 1.14: (ausschließlich für Experten!)

- Gib an, wie viele Sprachen die jeweiligen Personen sprechen.

- b) Gib an, von wie vielen Personen die einzelnen Sprachen gesprochen werden.
- c) Gib an, welche Sprache von keiner Person gesprochen wird.
(Beachte die Verknüpfungseigenschaften)
- d) Gib an, wer keine Sprache spricht.
(Beachte die Verknüpfungseigenschaften)
- e) Gib an, wie viele Personen existieren.