Datenmodellierung und Datenbankentwurf – eine lange Übung

Christoph Draxler
draxler@phonetik.uni-muenchen.de

29. Juni 2018

Einleitung

In dieser freiwilligen Übung, die über mehrere Stunden gehen wird, entwickeln wir ein Datenmodell für die Sprachaufnahmen am Institut für Phonetik. Bitte bearbeiten Sie diese Übung gemeinsam mit anderen – das macht einfach mehr Spaß!

1 Ausgangspunkt: Beschreibung

Das Institut für Phonetik ist Partner in wissenschaftlichen und industriellen Projekten. Ein Projekt kann nur einen, aber auch mehrere Partner haben. Im Rahmen von Projekten werden am Institut für Phonetik Sprachaufnahmen erstellt. Sprachaufnahmen erfolgen in Aufnahmesitzungen; bei einer Aufnahmesitzung werden Sprecher aufgenommen. Ein Sprecher ist mit geografischen Orten verknüpft, z.B. wohnt er oder sie im Wohnort, wurde irgendwo eingeschult, irgendwo geboren, usw.

Eine Aufnahme besteht aus Signaldateien, und eine Signaldatei ist einer Sitzung zuzuordnen. Die Signaldateien werden annotiert; jede Signaldatei kann mehrere Annotationen bekommen, und jede Annotation gehört zu genau einer Signaldatei. Für jede Signaldatei können automatisch akustische Merkmale berechnet werden, z.B. Formanten, Grundfrequenz oder spektrale Momente; für eine Signaldatei kann es mehrere akustische Merkmale parallel geben.

Ein geografischer Ort kann in einem anderen geografischen Ort liegen, z.B. eine Stadt in einem Bundesland, das Bundesland im Staat, usw. Eine Annotation kann sich auf eine andere Annotation beziehen, z.B. ein Allophon

eine Realisierung eines Phonems, ein Phonem Teil einer Silbe, eine Silbe Teil eines Wortes, usw.

Aufgabe Erstellen Sie ein erstes ER-Diagramm, noch ohne Attribute und Kardinalitäten. Ermitteln Sie dazu im Text die Substantive und Verben; gruppieren Sie die Substantive und hierarchisieren Sie sie, und zeichnen Sie dann anhand der Verben Beziehungen zwischen den Substantiven.

2 Entwurf ER-Diagramm

2.1 Extraktion von Substantiven und Verben

Die Substantive der obigen Beschreibung sind: Institut für Phonetik, Projekt, Partner, Rahmen, Sprachaufnahme, Aufnahmesitzung, Sprecher, Ort, Wohnort, Aufnahme, Signaldatei, Sitzung, Annotation, Merkmal, Formant, Grundfrequenz, Momente, Stadt, Bundesland, Staat, Allophon, Realisierung, Phonem, Silbe, Wort.

Verben sind: sein, können, haben, erstellen, erfolgen, aufnehmen, verknüpfen, einschulen, geboren sein, bestehen, zuordnen, annotieren, bekommen, berechnen, liegen, beziehen.

2.2 Gruppieren

Die Substantive kann man recht natürlich gruppieren:

Tabelle 1: Gruppierung der Substantive

Nr.	Gruppe	Wörter
I	Projekt	Institut für Phonetik, Partner
II	Aufnahme	Sitzung, Signaldatei
III	Sprecher	
IV	Merkmal	Formant, Grundfrequenz, Moment
V	Ort	Stadt, Bundesland, Staat
VI	Annotation	Allophon, Phonem, Sible, Wort

In jeder Gruppe wird das best-passende Wort zum Titel der Gruppe, z.B. bekommt Gruppe I den Titel 'Projekt'.

Die Verben werden im übernächsten Schritt benötigt.

2.3 Entwurf der Entitäten

Jede Gruppe wird zu einer Entität, d.h. einem benannten Rechteck im ER-Diagramm. Es kann hilfreich sein, näher verwandte Gruppen auch benachbart zu zeichnen – vielleicht gibt es dann später weniger sich kreuzende Verbindungen.

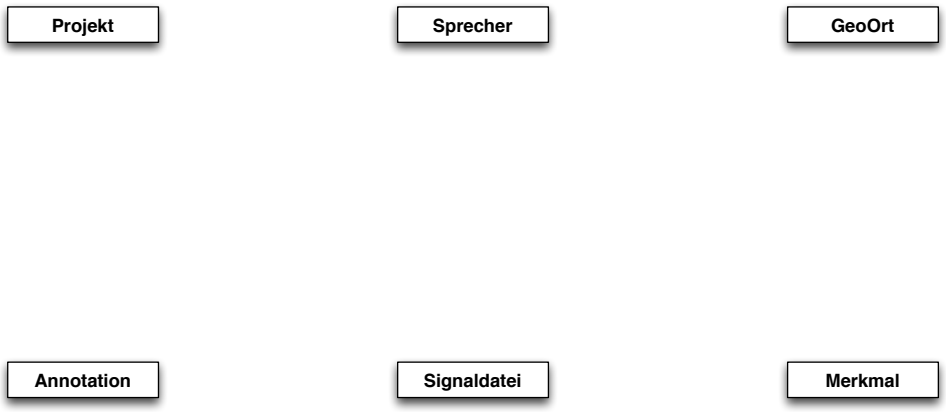


Abbildung 1: erste Entitäten im ER-Diagramm

Evtl. kann es sinnvoll sein, bestimmte Daten, z.B. die akustischen Merkmale, in je einer eigenen Entität zu erfassen, z.B. eine Entität `Formant`, eine `f0`, usw.

2.4 Entwurf der Beziehungen

Alle Entität sollen nun so miteinander verbunden werden, dass

- möglichst jede Entität mindestens eine Beziehung hat
- jede Beziehung nur eine Entität mit sich oder zwei Entitäten miteinander verbindet
- es möglichst wenig parallele Beziehungen gibt

Die Beziehungen werden mit den Verben – oder nah verwandten Verben – bezeichnet.

Die Pfeile geben einen ersten Hinweis auf die Kardinalitäten: die Pfeilspitze bezeichnet die `1-`, das stumpfe Pfeilende die `1`-Seite einer $1 : n$ Beziehung. So werden z.B. in *einem* Projekt meist *mehrere* Sprachaufnahmen erstellt.

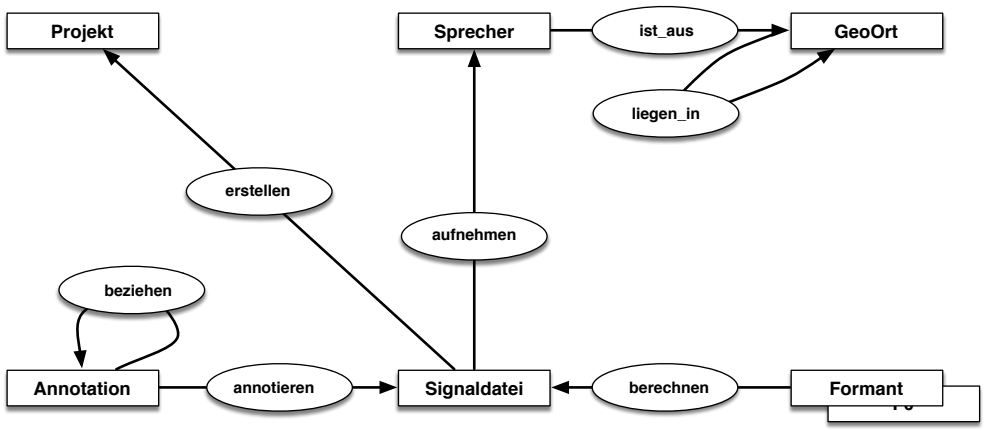


Abbildung 2: ER-Diagramm mit Entitäten und Beziehungen

2.5 Attribute hinzufügen

Nun kann man die Entitäten um Attribute ergänzen. Sin Sprecher hat einen Code, ein Alter (oder Geburtsdatum?), ein Geschlecht, und evtl. möchte man in Form eines Kommentars zusätzliche Bemerkungen zu ihm oder ihr festhalten. So geht es weiter – bis alle Entitäten mit Attributen versorgt sind.

3 Was sie hier zeichnen, gilt später in der Welt!

Daher kritisch prüfen, am besten gemeinsam mit dem Auftraggeber oder einem Fachexpterten:

- Ist es wahr, dass eine Aufnahme immer in einem Projekt entsteht, und auch nur in einem Projekt?
- Wieviele Sprecher dürfen für eine Aufnahme aufgenommen werden?
- Wieviele Aufnahmen darf ein Sprecher machen?
- Welche Annotationen beziehen sich wie aufeinander – gilt es immer, dass z.B. zwischen Wort und Phonem eine $1 : n$-Beziehung besteht?

Spielen Sie 'Guter Polizist – böser Polizist', verteilen Sie die Rollen des Prüfers und des Schülers und versuchen Sie, die Fehler im Diagramm zu finden.

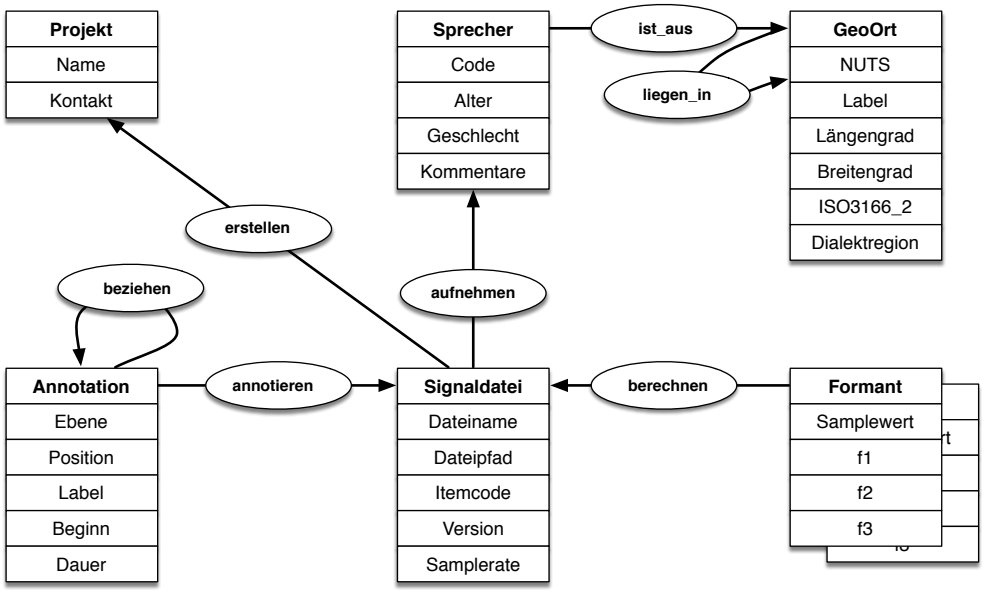


Abbildung 3: ER-Diagramm der Sprecherdatenbank mit Attributen

Gehen Sie Ihr ER-Diagramm sorgfältig durch und prüfen Sie jede Entität und Beziehung. Dann unterschreiben Ihr Auftraggeber und Sie bitte unten rechts, mit Datum. Jetzt dürfen Sie Ihrem Auftraggeber die erste Rate von ca. 30% in Rechnung stellen.

4 Verwenden der Datenbank

Am Institut für Phonetik ist die phonetische Datenbank als relationale Datenbank mit der Software PostgreSQL installiert. PostgreSQL ist eine frei verfügbare relationale Datenbanksoftware, die sich im Betrieb am Institut als außerordentlich stabil und sehr schnell erwiesen hat. Es gibt sie als fertiges Anwendungspaket für die gängigen Betriebssysteme, oder auch als Quellcode.

Der Zugriff auf die phonetische Sprachdatenbank erfolgt über ein Terminal oder z.B. die Software dbVisualizer. Diese ist auf Rechnern des Instituts installiert, man kann sie aber auch auf den eigenen Rechner installieren.

> **Achtung!** Die PostgreSQL-Datenbank ist so eingerichtet, dass sie nur innerhalb des Instituts-Netzwerks erreicht werden kann. Das bedeutet, dass Sie sich von außen zunächst mit einem Terminal über eine verschlüsselte Verbindung anmelden oder ihren Rechner mit einem Ethernetkabel am Institut verbinden müssen.

4.1 Anmelden bei der Datenbank

Von außerhalb des Instituts öffnen Sie ein Terminal, das die Verschlüsselung per sog. `ssh`-Protokoll unterstützt (`ssh` steht für *secure shell*). Geben Sie als Verbindungsziel den Rechnernamen `USERNAME@ssh.phonetik.uni-muenchen.de` ein, wobei `USERNAME` Ihre Kennung auf den Institutsrechnern (üblicherweise Ihre LMU-Kennung) ist, und loggen Sie sich mit Ihrem Passwort ein.[1]

```
ssh USERNAME@ssh.phonetik.uni-muenchen.de
```

Wenn Sie bereits auf einem Rechner des Instituts eingeloggt sind, dann geben Sie den folgenden Befehl ein:

```
psql -h postgres -U db_kurs speechdb
```

`psql` ist der Befehl zum Starten der PostgreSQL-Shell, `-h postgres` ist der Name des Rechners, auf dem der PostgreSQL-Server läuft, `-U db_kurs` gibt den Nutzernamen für die Datenbank an, und `speechdb` ist der Datenbankname.

Der Datenbankserver fragt nach dem Passwort. Geben Sie *Phonetik!* in exakt dieser Schreibweise ein. Nach Eingabe des Passworts erscheint eine Status- und Willkommensmeldung des Datenbankservers, und Sie können hinter der Aufforderung `speechdb=>` Befehle eingeben:

```
psql -h postgres -U db_kurs speechdb
Password for user db_kurs:
psql (9.3.4, server 9.1.15)
SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
Type "help" for help.

speechdb=>
```

4.2 PostgreSQL-Befehle

Nach der Anmeldung sind Sie automatisch mit der Datenbank `speechdb` verbunden. Die folgenden PostgreSQL-Befehle geben Auskunft über die Datenbank bzw. beenden die Sitzung.

`\q` Verlassen der PostgreSQL-Shell

`\d` Anzeigen der Namen aller relationalen Tabellen, Sequenzen und Views

[1] Eine Institutskennung bekommen Sie beim Systemadministrator des Instituts, Klaus Jänsch.

`\d NAME` Anzeigen der Struktur der Tabelle NAME

`\?` Anzeige aller PostgreSQL-Befehle

`\help SQL` Hilfetext zu SQL-Befehlen

Die Eingabe von `\d` z.B. gibt die folgende Liste zurück:

```
 Schema |         Name          |   Type   |  Owner
--------+-----------------------+----------+---------
 public | formant               | table    | draxler
 public | formants_view         | view     | draxler
 public | geolocation           | table    | draxler
 public | keys                  | sequence | draxler
 public | pitch                 | table    | draxler
 public | project               | table    | draxler
 public | segment               | table    | draxler
 public | signalfile            | table    | draxler
 public | speaker               | table    | draxler
 public | speaker_file_segments | view     | draxler
(10 rows)
```

Für die Aufgaben in Kap. 5 benötigen Sie nur die Tabellen, bei denen in der Spalte `Type` der Wert `table` steht.

4.3 SQL Befehle

In der PostgreSQL-Shell können Sie jeden SQL-Befehl eingeben. Beenden Sie den Befehl mit dem Semikolon (';') und drücken sie die Eingabetaste.

```
select * from speaker order by code limit 5;

    id    | age | sex | code | comment | geolocation_id | weight | height | accent
----------+-----+-----+------+---------+----------------+--------+--------+--------
 48789947 |  20 | f   | AAC1 |         |        1044768 |        |        |
 48789948 |  20 | m   | AAC2 |         |        1044768 |        |        |
 48789949 |  19 | f   | AAC3 |         |        1044768 |        |        |
 48789950 |  19 | m   | AAC4 |         |        1044768 |        |        |
 48789951 |  59 | m   | AAC5 |         |        1044768 |        |        |
(5 rows)
```

5 Aufgaben

Bitte bearbeiten Sie die folgenden Aufgaben. Wie üblich empfehle ich Ihnen, das gemeinsam mit anderen zu machen. Entwickeln Sie die Abfragen schrittweise.

Für diese Aufgaben benötigen Sie nur die Tabellen, die die in der obigen Auflistung aller Relationen als 'table' bezeichnet werden.

5.1 Fünf ganz einfache Aufgaben

Formulieren Sie die folgenden Abfragen in SQL und führen Sie sie in der Datenbank aus:

1. Suche alle männlichen Sprecher
2. Suche alle weiblichen Sprecherinnen zwischen 18 und 21
3. Suche Alter, Geschlecht und Code von Sprechern jünger als 15 und sortiere sie nach Code
4. Suche nur das Alter aller männlichen Sprecher, ohne Duplikate, und sortiere nach Alter
5. Suche alle Sprecher, für die es Gewichts- oder Größenangaben gibt

Für diese Abfragen benötigen Sie nur die Tabelle `speaker`.

5.2 Fünf einfache Aufgaben

Diese Abfragen benötigen einen Join oder mehrere Joins von Tabellen. Verwenden Sie daher in Ihren Abfragen sprechende Aliasnamen für die Tabellen.

1. Suche alle männlichen Sprecher aus Bayern
2. Suche alle weiblichen Sprecherinnen aus Thüringen zwischen 18 und 21 Jahren
3. Suche alle Orte in Bayern, für die es Sprecher gibt
4. Suche alle Wörter der Äußerung in der Datei 'AAA4640X2_0' und sortiere sie nach Position
5. Suche die Werte für den ersten und zweiten Formanten für den Vokal /i/ aus der Datei mit der id 17034510 und sortiere sie nach Zeitpunkt

Schauen Sie sich die Verknüpfungen der Relationen in Abb. 3 an und entwickeln Sie Ihre Abfrage dann ganz systematisch.

5.3 Fünf schwierigere Aufgaben

1. Suche die Phonem-Segmente für das Wort 'Computerspiele' und sortiere sie nach der Reihenfolge im Signal

2. Suche alle Wörter, die mit dem Phonem /S/ beginnen

3. Suche alle Wörter, die die Zeichenkette 'st' enthalten, und von Sprechern aus Baden-Württemberg gesprochen werden

4. Suche die tatsächlich realisierten Phoneme in den Wörtern 'ist' und 'fast'

5. Gib Alter, Geschlecht, Ort des Sprechers, die Wörter und realisierten Phonem-Segmente sowie die f0 Werte für die Aufnahme in der Signaldatei 'AAA4784B2_0' zurück und ordne das Ergebnis nach Samplewert

Überlegen Sie für jede Aufgabe, welche Relationen hierfür miteinander verknüpft werden müssen und entwickeln Sie die Abfragen systematisch und schrittweise.

Anhang

6 Einrichten einer Verbindung mit dbVisualizer

Laden Sie dbVisualizer auf Ihren Rechner herunter und installieren Sie die Software, oder öffnen Sie die Anwendung `dbvis` in einem Linux-Terminal.

Achtung! Sie können nur innerhalb des Institutsnetzes auf die Datenbank zugreifen, also nicht von zuhause aus.

1. Wählen Sie im Menü `Tools` die Option `Connection Wizard` aus. Es öffnet sich ein Dialogfenster. Geben Sie dort den Namen, unter dem Sie die Datenbankverbindung später finden möchten, ein, z.B. `SpeechDB` (Abb. 4).

a) 

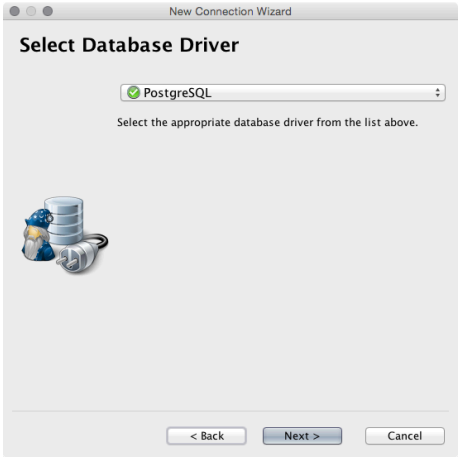
b) 

Abbildung 4: Connection Wizard in dbVisualizer: a) Bezeichnung der DB-Verbindung eingeben und b) DB-Treiber auswählen

2. Wählen Sie aus dem Popup-Menü den passenden Datenbanktreiber aus, hier: PostgreSQL. Es gibt für alle gängigen Datenbanksysteme Treiber, und Sie müssen für den für die verwendete Datenbank passenden auswählen.

3. Geben Sie nun die Zugriffsdaten für die Datenbank ein:

   Nutzername `db_kurs`

   Passwort `Phonetik!`

Datenbankserver `postgres.phonetik.uni-muenchen.de`

Datenbankport `5432`

Datenbankname `speechdb`

Mit einem Klick auf `Ping Server` können Sie testen, ob der angegebene Servername korrekt ist (Abb. 5).

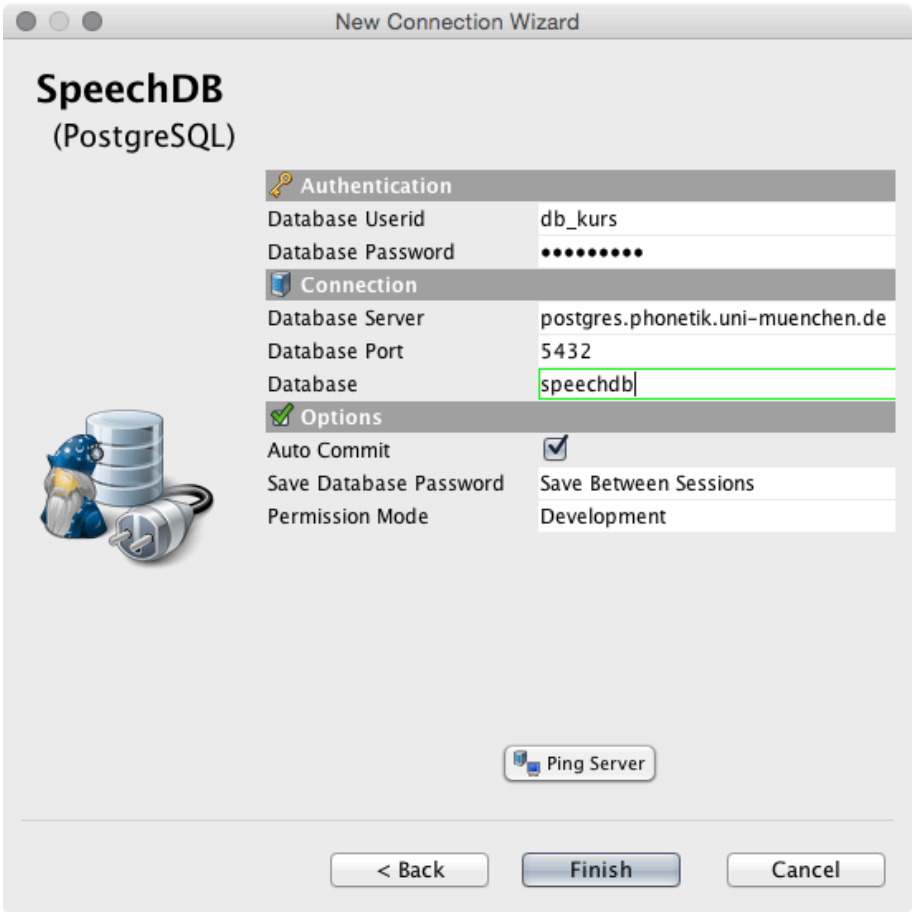


Abbildung 5: Zugriffsdaten für die SpeechDB-Datenbank

4. Nach Abschluss der Konfiguration wird die neu angelegte Datenbankverbindung im Hauptfenster von dbVisualizer angezeigt. Sie können durch einen Klick auf den Verbindungsnamen die Verbindung zur Datenbank aufbauen. Wenn die Verbindung hergestellt ist, sehen Sie unten im Fenster eine Statusmeldung des PostgreSQL Datenbankservers (Abb. 6).

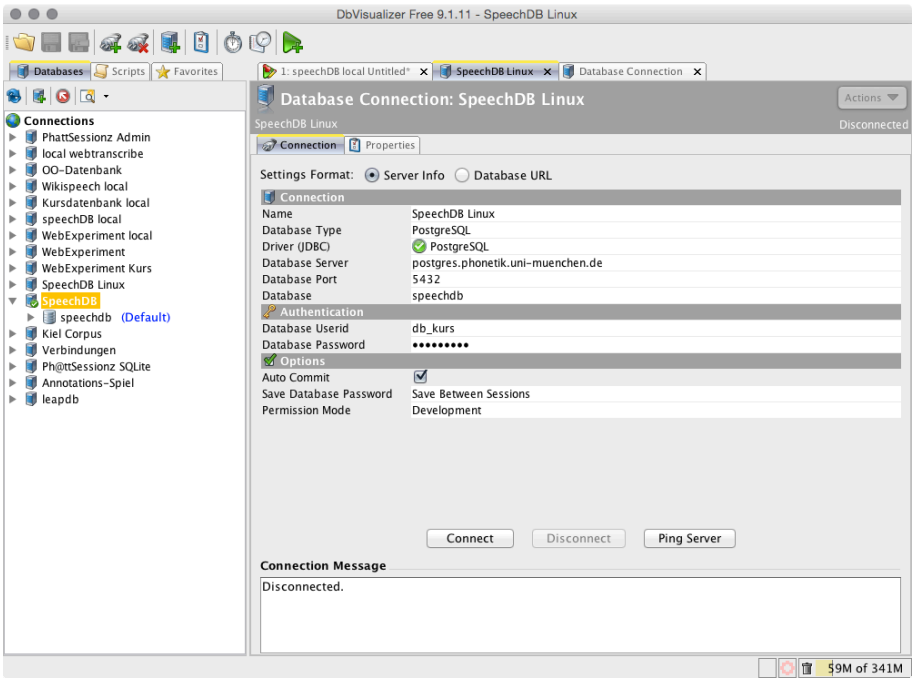


Abbildung 6: Hauptfenster von dbVisualizer mit Datenbankverbindungen. Unten rechts wird die Statusmeldung der aktuellen Datenbankverbindung angezeigt, hier die Willkommensmeldung des PostgreSQL-Servers.

5. Klicken Sie nun auf den Datenbanknamen unterhalb des Datenbankverbindungsnamens. Am oberen Rand des Fensters erscheint der Reiter `References`. Klicken Sie auf diesen Reiter. Damit startet dbVisualizer die Suche nach allen benutzerdefinierten Tabellen; wenn diese gefunden sind, werden Sie grafisch dargestellt (Abb. 7). Sie können verschiedene Anordnungs- und Darstellungsoptionen wählen – probieren Sie es aus!

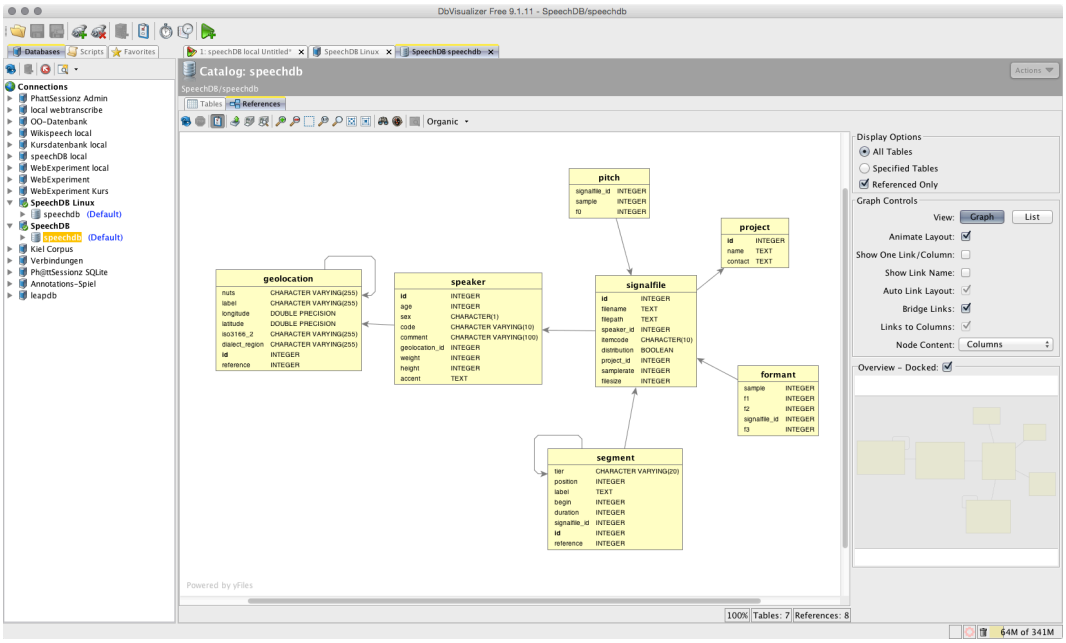


Abbildung 7: Relationen-Diagramm der Datenbank speechdb mit Primärschlüsseln und Attributen sowie den Verknüpfungen der Tabellen

6. Wenn es noch keinen SQL Commander gibt, dann legen Sie ihn über das Menü `SQL Commander` und die Option `New SQL Commander` an. Wenn es bereits einen solchen Commander gibt, dann gibt dbVisualizer eine Fehlermeldung zurück. Der SQL Commander erlaubt die komfortable Eingabe von SQL Befehlen. Dabei werden SQL Schlüsselwörter, Zeichenketten usw. farblich hervorgehoben. Achtung: ein SQL Commander kann für mehrere Datenbankverbindungen genutzt werden. Stellen Sie sicher, dass sie im Popup-Menü über dem Commanderfenster die richtige Verbindung eingestellt haben. Zum Abschicken einer Abfrage klicken Sie auf den grünen Pfeil (Abb. 8).

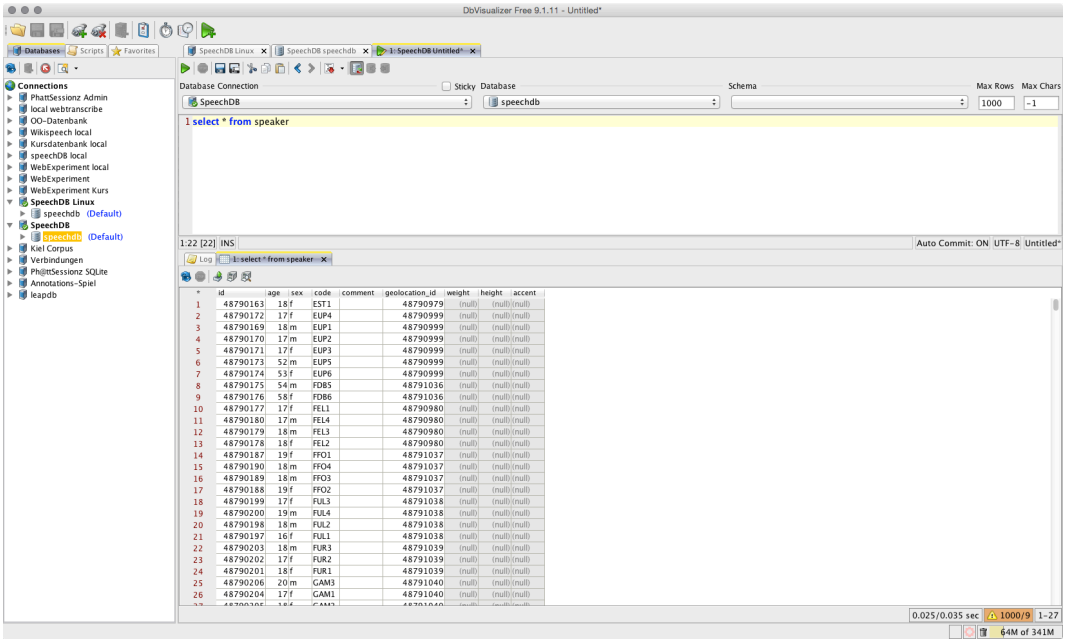


Abbildung 8: SQL Commander zur Eingabe von SQL Befehlen. Zum Ausführen des Befehls klicken Sie auf den grünen Pfeil. dbVisualizer beschränkt die Anzahl zurückgegebener Datensätze auf 1000. Diese Beschränkung kann man aber aufheben oder verändern.

7 Musterlösung Datenbankabfragen

7.1 Fünf ganz einfache Aufgaben

Formulieren Sie die folgenden Abfragen in SQL und führen Sie sie in der Datenbank aus:

1. Suche alle männlichen Sprecher

   ```
   select * from speaker where sex = 'm';
   ```

2. Suche alle weiblichen Sprecherinnen zwischen 18 und 21

   ```
   select * from speaker
   where sex = 'f'and age between 18 and 21;
   ```

3. Suche Alter, Geschlecht und Code von Sprechern jünger als 15 und sortiere sie nach Code

   ```
   select age, sex, code from speaker
   where age < 15 order by code;
   ```

4. Suche nur das Alter aller männlichen Sprecher, ohne Duplikate, und sortiere nach Alter

   ```
   select distinct age from speaker
   where sex = 'm'order by age;
   ```

5. Suche alle Sprecher, für die es Gewichts- oder Größenangaben gibt

   ```
   select * from speaker
   where not (weight is null or height is null);
   ```

Für diese Abfragen benötigen Sie nur die Tabelle `speaker`.

7.2 Fünf einfache Aufgaben

Diese Abfragen benötigen einen Join oder mehrere Joins von Tabellen. Verwenden Sie daher in Ihren Abfragen sprechende Aliasnamen für die Tabellen.

1. Suche alle männlichen Sprecher aus Bayern

```
select spk.*
from speaker spk
 join geolocation geo on spk.geolocation_id = geo.id
where spk.sex = 'm'
 and geo.iso3166_2 = 'DE-BY';
```

2. Suche alle weiblichen Sprecherinnen aus Thüringen zwischen 18 und 21 Jahren

```
select spk.*
from speaker spk
 join geolocation geo on spk.geolocation_id = geo.id
where spk.sex = 'f'
 and geo.iso3166_2 = 'DE-TH'
 and spk.age between 18 and 21;
```

3. Suche alle Orte in Bayern, für die es Sprecher gibt

```
select distinct geo.label, geo.iso3166_2
from geolocation geo
 join speaker spk on spk.geolocation_id = geo.id
where geo.iso3166_2 = 'DE-BY';
```

4. Suche alle Wörter der Äußerung in der Datei 'AAA4640X2_0' und sortiere sie nach Position

```
select ort.label, ort.position
from segment ort
 join signalfile sig on ort.signalfile_id = sig.id
where sig.filename = 'AAA4640X2_0'
 and ort.tier = 'ORT'
order by ort.position;
```

5. Suche die Werte für den ersten und zweiten Formanten für den Vokal /i/ aus der Datei mit der id 17034510 und sortiere sie nach Zeitpunkt

```
select f.f1, f.f2
from formant f
 join segment mau on mau.signalfile_id = f.signalfile_id
where mau.tier = 'MAU'
 and mau.label = 'i'
 and mau.signalfile_id = 17034510
order by mau.begin;
```

Schauen Sie sich die Verknüpfungen der Relationen in Abb. 3 an und entwickeln Sie Ihre Abfrage dann ganz systematisch.

7.3 Fünf schwierigere Aufgaben

1. Suche die Phonem-Segmente für das Wort 'Computerspiele' und sortiere sie nach der Reihenfolge im Signal

```
select mau.label, mau.begin, mau.duration
from segment mau
 join links l on mau.id = l.lfrom
 join segment ort on ort.id = l.lto
where ort.label = 'Computerspiele'
and mau.tier = 'MAU'
order by mau.signalfile_id, mau.begin;
```

2. Suche alle Wörter, die mit dem Phonem /S/ beginnen

```
select distinct ort.label
from segment mau
 join links l on mau.id = l.lfrom
 join segment ort on ort.id = l.lto
where mau.label = 'S'
and mau.tier = 'MAU'
and mau.position = 0
and ort.tier = 'ORT'
order by ort.label;
```

3. Suche alle Wörter, die die Zeichenkette 'st' enthalten, und von Sprechern aus Baden-Württemberg gesprochen werden

```
select distinct ort.label
from segment ort
 join signalfile sig on ort.signalfile_id = sig.id
 join speaker spk on sig.speaker_id = spk.id
 join geolocation geo on spk.geolocation_id = geo.id
where ort.tier = 'ORT'
 and geo.iso3166_2 = 'DE-BW'
 and ort.label like '%st%'
order by ort.label;
```

4. Suche die tatsächlich realisierten Phoneme in den Wörtern 'ist' und 'fast'

```
select distinct ort.label, mau.label
from segment ort
 join links l on l.lto = ort.id
 join segment mau on mau.id = l.lfrom
where mau.tier = 'MAU'
 and ort.tier = 'ORT'
 and ort.label in ('ist', 'fast')
order by ort.label, mau.label;
```

5. Gib Alter, Geschlecht, Ort des Sprechers, die Wörter und realisierten Phonem-Segmente sowie die f0 Werte für die Aufnahme in der Signaldatei 'AAA4784B2_0' zurück und ordne das Ergebnis nach Samplewert

```
select spk.age, spk.sex, geo.label, ort.label, mau.label, p.sample, p.f0
from speaker spk
 join signalfile sig on sig.speaker_id = spk.id
 join geolocation geo on spk.geolocation_id = geo.id
 join segment ort on sig.id = ort.signalfile_id
 join links l on ort.id = l.lto
 join segment mau on mau.id = l.lfrom
 join pitch p on sig.id = p.signalfile_id
where ort.tier = 'ORT'
 and mau.tier = 'MAU'
 and sig.filename = 'AAA4784B2_0'
 and p.sample between mau.begin and (mau.begin + mau.duration)
order by p.sample;
```

   Fällt Ihnen was auf? **Tipp**: Sind alle erwarteten Phoneme in der Ergebnisrelation?

Die letzte Abfrage liefert sehr viele Ergebnisse, was ziemlich unübersichtlich ist. Wie können Sie nur den Durchschnittswert der f0-Werte pro Phonem berechnen? Ist das sinnvoll?