INSTITUT FÜR DEUTSCHE SPRACHE

OPAL

Online publizierte Arbeiten zur Linguistik

ISSN 1860-9422

1/2010

Roman Schneider

# Planning for new media

## The Bibliography of German Grammar goes online

(Erstpublikation / first published 2000 in ORBIS,
http://www.ids-mannheim.de/grammis/orbis/)

**OPAL – Online publizierte Arbeiten zur Linguistik**
Herausgegeben vom Institut für Deutsche Sprache

Institut für Deutsche Sprache
Postfach 10 16 21
68016 Mannheim
opal@ids-mannheim.de

Technische Redaktion: Norbert Volz

Roman Schneider

# Planning for new media – The Bibliography of German Grammar goes online

**(First published 2000 in ORBIS, http://www.ids-mannheim.de/grammis/orbis/)**

## 1. Introduction

The evolution of computer technologies and the introduction of the World Wide Web (WWW) have substantially changed the way scientific articles and books are published today. Besides writing for "traditional" print media, more and more authors decide to reach a larger audience and to decrease distribution time by offering their works on the internet. The electronic medium not only facilitates the spread of information, it also adds new value by extending the possibilities of knowledge retrieval. Of course the same is true for structured data collections like scientific glossaries, dictionaries or bibliographies. They particularly profit from the web when being accessible via user-friendly and effective frontends.

The following chapters deal with the transformation of the Bibliography of German Grammar ("Bibliografie zur deutschen Grammatik") from a data pool primarly used for print publishing to a relational database application offering a basis for media-independent distribution. Starting with a short description of the beginnings of the bibliography, the focus of this article lies on the explanation of our current database design as well as on the presentation of the web-based user interface.

The Bibliography of German Grammar, currently hosted at the Institute for the German Language ("Institut für Deutsche Sprache") at Mannheim/Germany, is an up-to-date collection of more than 10.0000 scientific publications dedicated to German Grammar. Besides the usual bibliographic information (author names, title, year, place, publishing house, etc.), these entries are enriched by additional headwords, object words and language information to enable users to search this bibliography by different criteria. Since 1999, the retrieval frontend is part of the GRAMMIS information system (see [1]), which is an approach to grammatical knowledge on the internet. It can be accessed via http://www.ids-mannheim.de/grammis/.

## 2. The Starting Point

The origins of the Bibliography of German Grammar date from the year 1980, when Peter Eisenberg decided to gather information about scientific publications in the fields of German grammar. Starting with a small team of collaborators, located at the "Freie Universität" at Berlin and at the University of Hannover, he systematically tried to include every book and every article dedicated to German grammar that was published since the early sixties. This work was carried out by analysing various specialized publications like linguistic journals or catalogues from publishing houses. In the beginning, the entries were collected using simple index cards. The first print edition of the bibliography was published in 1985 (see [3]), followed by a second edition in 1988 (see [4]). In the late eighties, after the introduction of the personal computer, the index cards were replaced by a database system running under MS-DOS. This made it easier to maintain the increasing amount of bibliographic entries, which was already larger than 5,000 items. But this first database just mirrored the functionality of the index cards without adding any new relational structures. In 1992 the bibliographic work was continued at the University of Potsdam. Some years later, after the publication of the third print edition (see [5]), the project group decided to use the Standard Generalized Markup Language (SGML) for the logical formatting of the bibliographic data. The aim was to make the process of collecting and maintaining the entries fully independent from any underlying computer platform or operating system. Whereas the old, proprietary database system reduced the scope to a single platform, the use of SGML allowed the

effective processing of the data on literally every computer system. Another aspect was the handling of language-specific characters (e.g., special characters in the names of east-european authors), for which SGML offers a simple and effective solution.

In 1999 the location and composition of the project group changed again. Since then, the Bibliography of German Grammar – with now more than 10,000 entries – is maintained at the Institute for the German Language (Institut für Deutsche Sprache) in Mannheim/Germany. The current goal is not only to keep the bibliography permanently up-to-date, but to make the administration more effective and secure. Besides, the plan is to broaden the potential readership by adding new value and offering the contents on the World Wide Web. To achieve this, the first step was to change the way the data is stored and edited. In order to extend the possibilities of SGML by the advantages of a database management system (DBMS), plans were worked up to migrate the data to the object-relational DBMS Oracle™. This decision was based on the following arguments (of course a lot more could be said about the general advantages of database management systems, just think of aspects like scaling, high availability, data analysis, etc.):

1. Robustness: Aside from the stability of the hardware and the operating system itself, database management systems provide a robust environment for the development of data collections and applications as well as advanced backup and recovery strategies.

2. Security: A DBMS can control access to every piece of data and prevents the unauthorized modification or deletion of existing database entries. This seems important especially for multi-user environments, where users with different responsibilities work on a complex collection of data (such as a bibliography).

3. Accurate Data: Database management systems promise data accuracy and the avoidance of redundancy by offering automatic consistency and integrity checks.

4. Retrieval: Stored in a database, the complex bibliographic data can be easily accessed using the mighty Structured Query Language (SQL). Furthermore the execution of queries can be offered to a global audience by developing a gateway to the World Wide Web (WWW).

5. Platform independency: Since Oracle™ runs on nearly every computer platform, the migration to another machine or operating system at some future date should be no problem. Besides, it seems easy to generate well-formed SGML out of database tables.

The data used as starting point for the migration was organized in three separate SGML-files: One for the collected articles, one for the books (monographs as well as anthologies) and one for periodicals (journals, series, etc.). To get an impression of the internal structure, just take a look at the following examples.

Sample article entry:

```
<aufsatz id="t6830" jahr="1992" bd3="bd3" utyp="11" relevant="ja">
  <autor typ="autor">
    <nachname>Strecker</nachname><vorname>Bruno</vorname>
  </autor>
  <titel>Zum Begriff des Satzes</titel>
  <verweis idref="t6815" jahr="1992" von="408" bis="416">
  <unstruk>Hoffmann, Ludger</unstruk>
  <schlagwort>Satz, Definition</schlagwort>
</aufsatz>
```

Sample book entry:

```
<buch id="t1657" jahr="1980" bd1="teil-b" typ="mo" utyp="50" relevant="ja">
  <autor typ="autor">
    <nachname>Heringer</nachname>
    <vorname>Hans</vorname><vorname>JÜrgen</vorname>
  </autor>
  <autor typ="autor">
    <nachname>Wimmer</nachname><vorname>Rainer</vorname>
  </autor>
  <autor typ="autor">
    <nachname>Strecker</nachname><vorname>Bruno</vorname>
  </autor>
 <titel>Syntax. Fragen - LÖsungen - Alternativen</titel>
 <pubdata>
   <ort>MÜnchen</ort>
   <verlag>Fink</verlag>
   <reihe ref="b208" band="251">UTB</reihe>
 </pubdata>
</buch>
```

Sample periodical entry:

```
<periodikum typ="RW" id="b43">
  <titel>Forschungsberichte des Instituts fÜr deutsche Sprache</titel>
  <sigle>IdS</sigle>
  <ort>TÜbingen</ort>
  <verlag>Narr</verlag>
</periodikum>
```

## 3. The Migration Work

Keeping in mind that the migration of the bibliographic data should facilitate the future administration work and allow the processing of "intelligent" structured queries, some additional steps had to be carried out before building the final database. First of all, the complexity of the topic required the developing of appropriate data models for describing the representation of information in terms of data. In other words, we had to determine the types of information that needed capturing. This had to be done under structural and operational aspects and produced a so-called "conceptual data model" (sometimes called "logical data model" as well, but these terminological questions can be ignored here). It contained all knowledge necessary to model the miniworld of bibliographies irrespective of the concrete software to be used for the later implementation. The realization of this conceptual data model was done using entity-relationship (E-R) diagrams. They are based on the distinction between "entity types" that represent the logical "things" our miniworld is composed of and "relationship types" among those entity types. Below, you will find the E-R diagram for the Bibliography of German Grammar, which was created with the powerful CASE tool Oracle Designer/2000 (see [5]). Entity types are displayed as rounded rectangles, with their names and their attributes listed inside. Relationship types are presented according to the following conventions: Single lines represent "one-to-one"-relationships, crow's feet represent "one-to-many"- or "many-to-many"-relationships; lines stand for "must"-relationships, broken lines stand for "may"-relationships. For example, the following state-

ments are coded in the figure below: “Each book (buch) must be published by a publishing house (verlag).” and “Each publishing house (verlag) may publish one ore more book(s) (buch).”. The relationship between publishing houses and periodicals is an example for a “many-to-many” (n:m)-relationship: “Different publishing houses can publish the same periodical. Different periodicals can be published by the same publishing house.”.

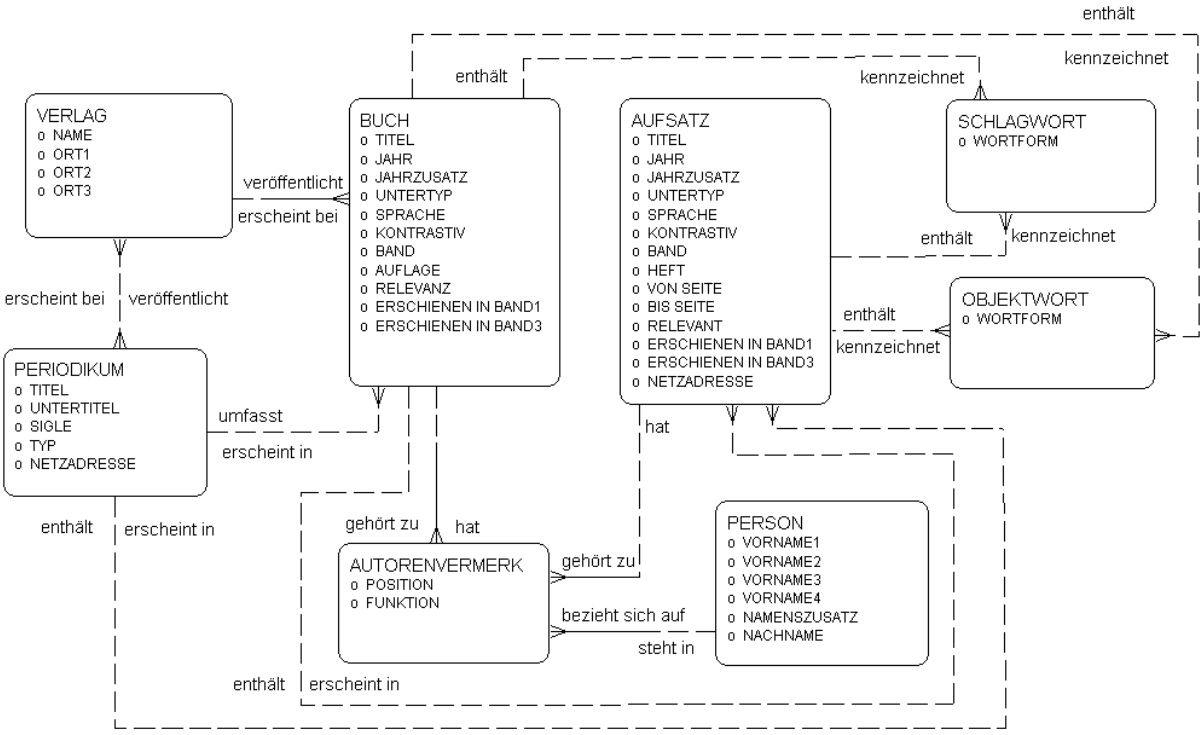


Figure 1: Conceptual data model for the Bibliography of German Grammar

So what is the advantage of mapping bibliographic constructs into an E-R representation? One important aspect is that it seems to be a good way to understand the logical structures that organize our information. Based on this understanding and with the help of relational algebra, the later development of database applications can take these structures into account and offer solutions to certain problems. For example, let's think of the relationship between books and authors. The non-relational way to store this information would be similar to the index card solution: For each book, we had to enter the names of the authors concerned again and again. Given the fact that 80 out of 2000 authors included in our bibliography are responsible for about 25% of all book titles, this is obviously not very effective. Besides the multiple work, it preprogrammes the storing of different spellings of author names. The introduction of separate entities for the book and the author opens up the possibility to store the information about the authors at a different location and to use this information when entering a new book entry (readers familiar with database design will notice that this already gets us to the physical data and application modelling, which is the next step that comes after the conceptual modelling). Of course it is sometimes difficult to determine the “real” name of an author. Especially transliterations from other alphabets raise problems, as well as titles of nobility (“von”, “de”, “van”, etc.) or the fact that people sometimes completely change their surname after marriage. To model the last-mentioned case, we could add an additional recursive relationship “is identical with” to the entity “person” in the E-R diagram above.

When designing a database based on the former analysis of functional dependencies, attention has to be paid to the normalization of the defined relations. The primary goal here is to reduce an arbitrary relation to an equivalent collection of 3NF (third normal form) relations. Despite of some special cases described below, the relations defined within the database model for the Bibliography of German

Grammar fulfill the demands of E.F. Codd's original three normal forms (for more information on normal forms, see [2]).

When examining our conceptual database design, relational purists would possibly argue that the conception of the two entities “person” and “verlag” (“publishing house”) does not exhaust the full potential of E-R modelling. It is obvious: Since people can have one ore more first names, it would be accurate to split the entity “person”, to create a new entity called “first name” and to connect these two entities with a “one-to-many”-relationship. The same is true for the “verlag” entity: Publishing houses can be based at one or more different cities, so under ideal circumstances the modelling should reproduce this. We intentionally broke the rules and restricted the maximal number of first names to four and the maximal number of cities to three. This was done after analysing the complete SGML data to be sure that none of the existing entries needs more than that. The advantages we derived from avoiding additional entities are a substantial ease of administration work in the resulting database application as well as less complex and therefore faster retrieval queries.

As always, the conceptual model finally used for the later creation of our database tables is a compromise and far from being the only possible view on the mini-world it describes. Maybe changing demands will make it necessary to re-model some parts of it. For example, we assumed that each book or article can only be written in one language and that it may cover only one contrastive language. When detecting that some of the original SGML entries contain a contrastive language attribute called “Verschiedene” (mixed), we decided not to extend our model because of administrative reasons. If some future application will make it desirable to explicitly distinguish between different contrastive languages, it should be manageable to change the way we looked on the data and to add the changes to the bibliographic database.

Another aspect that gives reasons for the use of conceptual data models has to do with system independency; those models can be used as a basis when migrating to a different DBMS. As already mentioned, the translation of conceptual data models into something a database system can work with is called “physical data modelling”. It can be defined as a low-level description of data, as a description of how to store information physically in database tables. In contrast to the abstract modelling we did before, the physical data model shown in the figure below precisely describes the internal structure of the database tables and the foreign keys used to join two tables together.

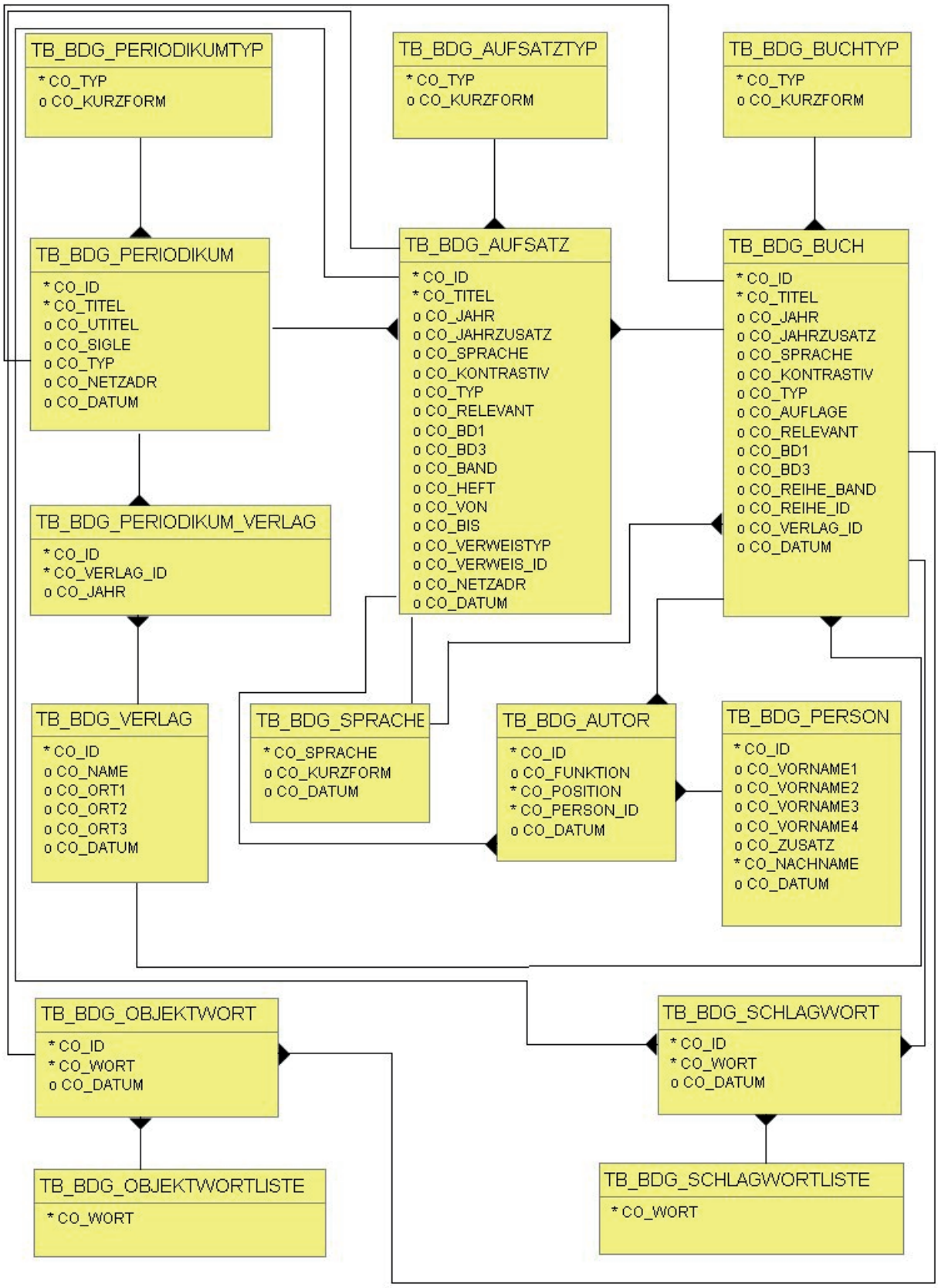


Figure 2: Physical data model for the Bibliography of German Grammar

For a better understanding of the role of the database tables, a few words should be said about their content and its origin. Most parts of the information come from the SGML files described in the previous chapter, but besides structural reorganization and addition of formerly implicitly stored information we also added some new content. The descriptions below start with the formal characterization of each table column, followed by a short explanation. The tables are listed in alphabetical order.

Description of table TB_BDG_AUFSATZ:

```
CO_ID                    NOT NULL VARCHAR2(10)
CO_TITEL                 NOT NULL VARCHAR2(400)
CO_JAHR                           NUMBER(4)
CO_JAHRZUSATZ                     VARCHAR2(2)
CO_SPRACHE                        VARCHAR2(50)
CO_KONTRASTIV                     VARCHAR2(50)
CO_TYP                            VARCHAR2(50)
CO_RELEVANT                       VARCHAR2(10)
CO_BD1                            VARCHAR2(10)
CO_BD3                            VARCHAR2(5)
CO_BAND                           VARCHAR2(15)
CO_HEFT                           VARCHAR2(5)
CO_VON                            VARCHAR2(5)
CO_BIS                            VARCHAR2(5)
CO_VERWEISTYP                     VARCHAR2(1)
CO_VERWEIS_ID                     VARCHAR2(10)
CO_NETZADR                        VARCHAR2(200)
CO_DATUM                          DATE
```

This table stores information about scientific articles that were published in journals, anthologies etc. The column CO_ID contains an unambiguous identifier for each article. CO_TITEL stores its title, CO_JAHR stores the year of publication and CO_JAHRZUSATZ an addition like “a”, “b”, “c”, etc. CO_SPRACHE stores the language the article is written in, CO_KONTRASTIV stores the language the article is about (both correspond to the column CO_SPRACHE of table TB_BDG_SPRACHE). CO_TYP stores the article type and corresponds to the column CO_TYP of table TB_BDG_AUFSATZTYP. CO_RELEVANT stores information about the relevance of the entry, CO_BD1 and CO_BD3 say something about whether the entry was published in the first and third print edition of the biography or not (these columns are remains of the original SGML data records and will not be maintained in future). The column CO_VERWEISTYP contains information about the type of publication the article was published in (book or journal); CO_VERWEIS_ID is a link to the publication concerned and corresponds to an entry in TB_BDG_BOOK or TB_BDG_PERIODIKUM. The columns CO_BAND and CO_HEFT contain – if available – information about volume and issue. CO_VON and CO_BIS store the page numbers. A new type of information is collected in column CO_NETZADR: To keep up with current developments in scientific publication, it is now possible to enter the internet address of electronically published articles. Finally, CO_DATUM stores the last modification date for each record set.

Description of table TB_BDG_AUFSATZTYP:

```
CO_TYP                   NOT NULL VARCHAR2(50)
CO_KURZFORM                       VARCHAR2(5)
```

This table holds the information about possible article types. The column CO_TYP contains the different types (“journal article”, “review”, “summary”, “foreword”, etc.), CO_KURZFORM stores the corresponding abbreviations.

Description of table TB_BDG_AUTOR:

```
CO_ID                    NOT NULL VARCHAR2(10)
CO_FUNKTION                       VARCHAR2(20)
CO_POSITION              NOT NULL NUMBER(2)
CO_PERSON_ID             NOT NULL NUMBER(5)
CO_DATUM                          DATE
```

This table stores information about the persons responsible for a certain publication. Column CO_ID holds the unique identifier of the publication (a book or an article), CO_PERSON_ID refers to a personal name stored in TB_BDG_PERSON. CO_FUNKTION says something about the function of this person (“editor”, “author”, “co-author”, etc.). Since the names on a book cover are not always arranged in alphabetical order, CO_POSITION explicitly stores the position of each name. CO_DATUM stores the last modification date for each entry.

Description of table TB_BDG_BUCH:

```
CO_ID                    NOT NULL VARCHAR2(10)
CO_TITEL                 NOT NULL VARCHAR2(400)
CO_JAHR                           NUMBER(4)
CO_JAHRZUSATZ                     VARCHAR2(2)
CO_SPRACHE                        VARCHAR2(50)
CO_KONTRASTIV                     VARCHAR2(50)
CO_TYP                            VARCHAR2(50)
CO_AUFLAGE                        VARCHAR2(50)
CO_RELEVANT                       VARCHAR2(10)
CO_BD1                            VARCHAR2(10)
CO_BD3                            VARCHAR2(5)
CO_REIHE_BAND                     VARCHAR2(10)
CO_REIHE_ID                       VARCHAR2(10)
CO_VERLAG_ID                      NUMBER(5)
CO_DATUM                          DATE
```

This table stores information about books relevant for grammar science. The column CO_ID contains an unambiguous identifier for each book. CO_TITEL stores its title, CO_JAHR stores the year of publication and CO_JAHRZUSATZ an addition like “a”, “b”, “c”, etc. CO_SPRACHE stores the language the book is written in, CO_KONTRASTIV stores the language the book is about (both correspond to the column CO_SPRACHE of table TB_BDG_SPRACHE). CO_TYP stores the book type and corresponds to the column CO_TYP of table TB_BDG_BUCHTYP. CO_AUFLAGE holds information about the book's print edition. CO_RELEVANT stores information about the relevance of the entry, CO_BD1 and CO_BD3 say something about whether the entry was published in the first and third print edition of the bibliography or not (as for TB_BDG_ARTICLE, these columns are remains of the original SGML data records and will not be maintained in future). CO_REIHE_ID refers to a series a book belongs to (stored in table TB_BDG_PERIODIKUM), CO_REIHE_BAND says under which number the book appears in this series. Column CO_VERLAG_ID stores a link to the publishing house in table TB_BDG_VERLAG. As always, CO_DATUM stores the last modification date for each entry.

Description of table TB_BDG_BUCHTYP:

```
CO_TYP                    NOT NULL VARCHAR2(50)
CO_KURZFORM                        VARCHAR2(5)
```

This table holds the information about possible book types. The column CO_TYP contains the different types (“monograph”, “anthology”, “ph.d. thesis”, etc.), CO_KURZFORM stores the corresponding abbreviations.

Description of table TB_BDG_OBJEKTWORT:

```
CO_ID                     NOT NULL VARCHAR2(10)
CO_WORT                   NOT NULL VARCHAR2(200)
CO_DATUM                           DATE
```

This table connects books and articles with so-called object words (German: “Objektworte”) that are relevant in the publication's context. Object words can be full words as well as prefixes, suffixes, etc. and are stored in column CO_WORT. The column CO_ID refers to the book's or article's unique identifier, CO_DATUM stores the last modification date of the record set.

Description of table TB_BDG_OBJEKTWORTLISTE:

```
CO_WORT                   NOT NULL VARCHAR2(200)
```

This is a look-up table which contains the list of possible object words used in table TB_BDG_OBJEKTWORT.

Description of table TB_BDG_PERIODIKUM:

```
CO_ID                     NOT NULL VARCHAR2(10)
CO_TITEL                  NOT NULL VARCHAR2(400)
CO_UTITEL                          VARCHAR2(400)
CO_SIGLE                           VARCHAR2(50)
CO_TYP                             VARCHAR2(50)
CO_NETZADR                         VARCHAR2(200)
CO_DATUM                           DATE
```

This table stores information about periodicals like journals or book series. The column CO_ID contains an unambiguous identifier for each periodical, CO_TITEL stores its title and CO_UTITEL holds the subtitle. In CO_SIGLE we find the sigle – a kind of abbreviation – for the periodical and CO_TYP stores its type. Like in table TB_BDG_ARTICLE, the column CO_NETZADR refers to an internet address (e.g., for e-journals). CO_DATUM stores the last modification date for each record set.

Description of table TB_BDG_PERIODIKUMTYP:

```
CO_TYP                    NOT NULL VARCHAR2(50)
CO_KURZFORM                        VARCHAR2(5)
```

This is again a look-up table, this time with a list of possible types of periodicals used in table TB_BDG_PERIODIKUM. The types are found in column CO_TYP, whereas CO_KURZFORM stores an abbreviation.

Description of table TB_BDG_PERIODIKUM_VERLAG:

```
CO_ID                              VARCHAR2(10)
CO_VERLAG_ID                       NUMBER(5)
CO_JAHR                            NUMBER(4)
```

This table was created for modelling the relationship between periodicals and publishing houses. Since the publishing house responsible for a certain periodical can change with the time, this table holds the data necessary for connecting one periodical with different publishing houses. Column CO_ID stores the unique identifier of a periodical, CO_VERLAG_ID refers to a publishing house stored in table TB_BDG_VERLAG. In column CO_JAHR it is possible to enter the year a periodical moved to this publishing house.

Description of table TB_BDG_PERSON:

```
CO_ID                    NOT NULL NUMBER(5)
CO_VORNAME1                       VARCHAR2(200)
CO_VORNAME2                       VARCHAR2(200)
CO_VORNAME3                       VARCHAR2(200)
CO_VORNAME4                       VARCHAR2(200)
CO_ZUSATZ                         VARCHAR2(20)
CO_NACHNAME              NOT NULL VARCHAR2(200)
CO_DATUM                          DATE
```

As already mentioned, table TB_BDG_PERSON stores the names of the persons responsible for an article or a book. The column CO_ID contains an unambiguous identifier for each entry. CO_VORNAME1 to CO_VORNAME4 hold the first names of a person, CO_NACHNAME stores the surname and CO_ZUSATZ other parts like titles of nobility (“von”, “de”, “van”, etc.). CO_DATUM stores the entry's last modification date.

Description of table TB_BDG_SCHLAGWORT:

```
CO_ID                    NOT NULL VARCHAR2(10)
CO_WORT                  NOT NULL VARCHAR2(200)
CO_DATUM                          DATE
```

This table connects books and articles with relevant headwords. Column CO_ID refers to the book's or article's unique identifier, in CO_WORT we find the headword, and CO_DATUM stores the last modification date.

Description of table TB_BDG_SCHLAGWORTLISTE:

```
CO_WORT                  NOT NULL VARCHAR2(200)
```

This is a look-up table which contains the list of headwords used in table TB_BDG_SCHLAGWORT.

Description of table TB_BDG_SPRACHE:

```
CO_SPRACHE               NOT NULL VARCHAR2(50)
CO_KURZFORM                       VARCHAR2(5)
CO_DATUM                          DATE
```

This table holds the different languages used in TB_BDG_ARTICLE and TB_BDG_BOOK. The column CO_SPRACHE contains the fully written languages (from “Afrikaans” to “Zurich-German”), CO_KURZFORM stores corresponding abbreviations. CO_DATUM stores the last modification date for each entry.

Description of table TB_BDG_VERLAG:

```
CO_ID                    NOT NULL NUMBER(5)
CO_NAME                           VARCHAR2(200)
CO_ORT1                           VARCHAR2(200)
```

| | |
|---|---|
| CO_ORT2 | VARCHAR2(200) |
| CO_ORT3 | VARCHAR2(200) |
| CO_DATUM | DATE |

Finally, this table stores information about publishing houses. The column CO_ID contains an unambiguous identifier for each entry. CO_NAME stores the name of the publishing house, CO_ORT1 to CO_ORT3 stores up to three the city names and in CO_DATUM we keep the last modification date.

## 4. The Online Presentation

After migrating the SGML data to a relational database, the question of how to access and maintain the bibliographic content had to be solved. Knowing about the advantages of the World Wide Web (WWW) as a global and easy-to-use media, we decided to use web technology for entering, modifying and querying the information stored in the bibliography database. This means that normal users as well as people entrusted with the maintaining of the database should be able to use a simple web browser to fulfill their needs. Accordingly, an appropriate WWW application had to cover both sides of database access: data input and data output. But since the end-user pages of the application seem to be of greater public interest than the administrative components, the following lines describe only the frontend as it appears for the majority of users out on the internet. This frontend is based on dynamic web pages created by PL/SQL-procedures that are stored directly inside the database. This was done because of security reasons as well as to reduce the response time for people accessing the bibliography from distant locations.

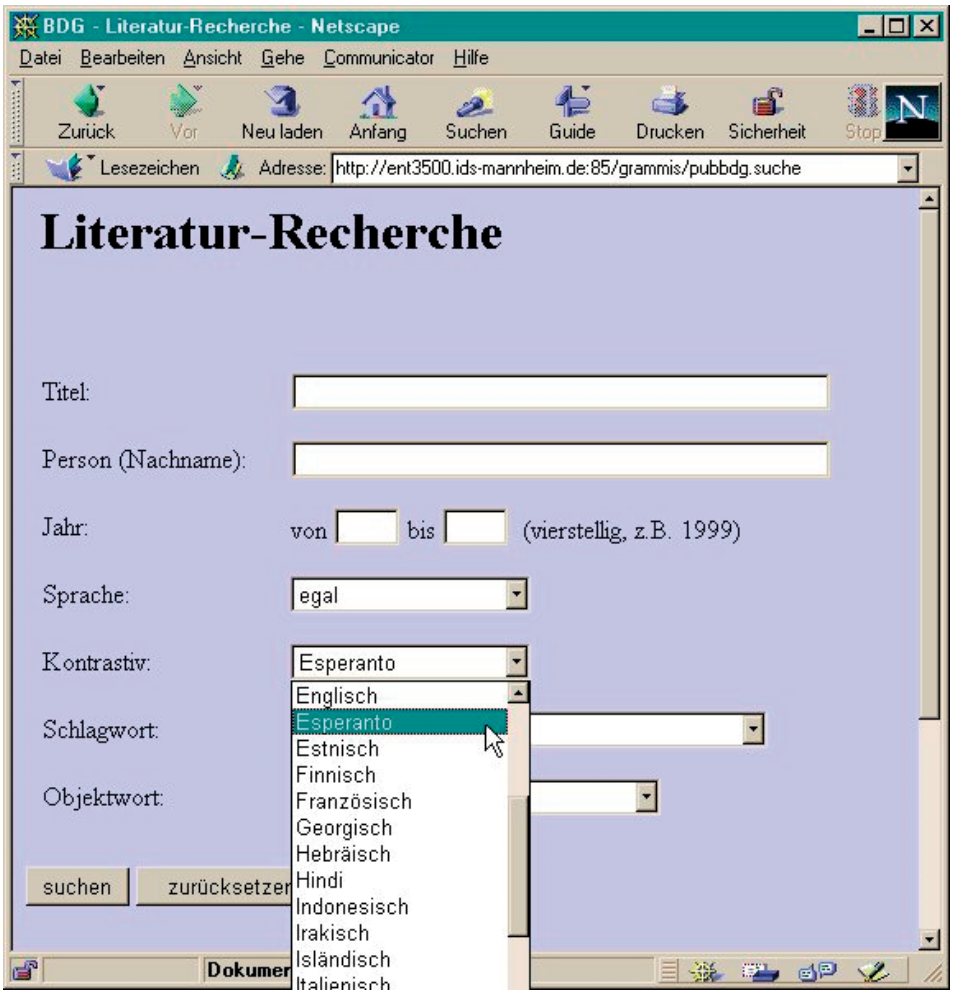


Figure 3: WWW frontend for complex retrieval queries

To allow the processing of complex queries, we built a special form for the online retrieval of bibliographic entries (see figure above). Users can fill in combinations of entry title, author name or publishing year – possibly using wildcards if they don't know the exact spelling. In addition, they can choose from pulldown lists to specify the entry's language, the contrastive language, the object words, and the headwords. After pushing the “search” button, the user gets a result list of all entries matching his criteria.

Besides structured search, it is also possible to request an overview of all articles and books stored inside the bibliographic database. As in the result list mentioned above, each entry is presented as a hyperlink, connecting the user with detailed information about the specific book or article.

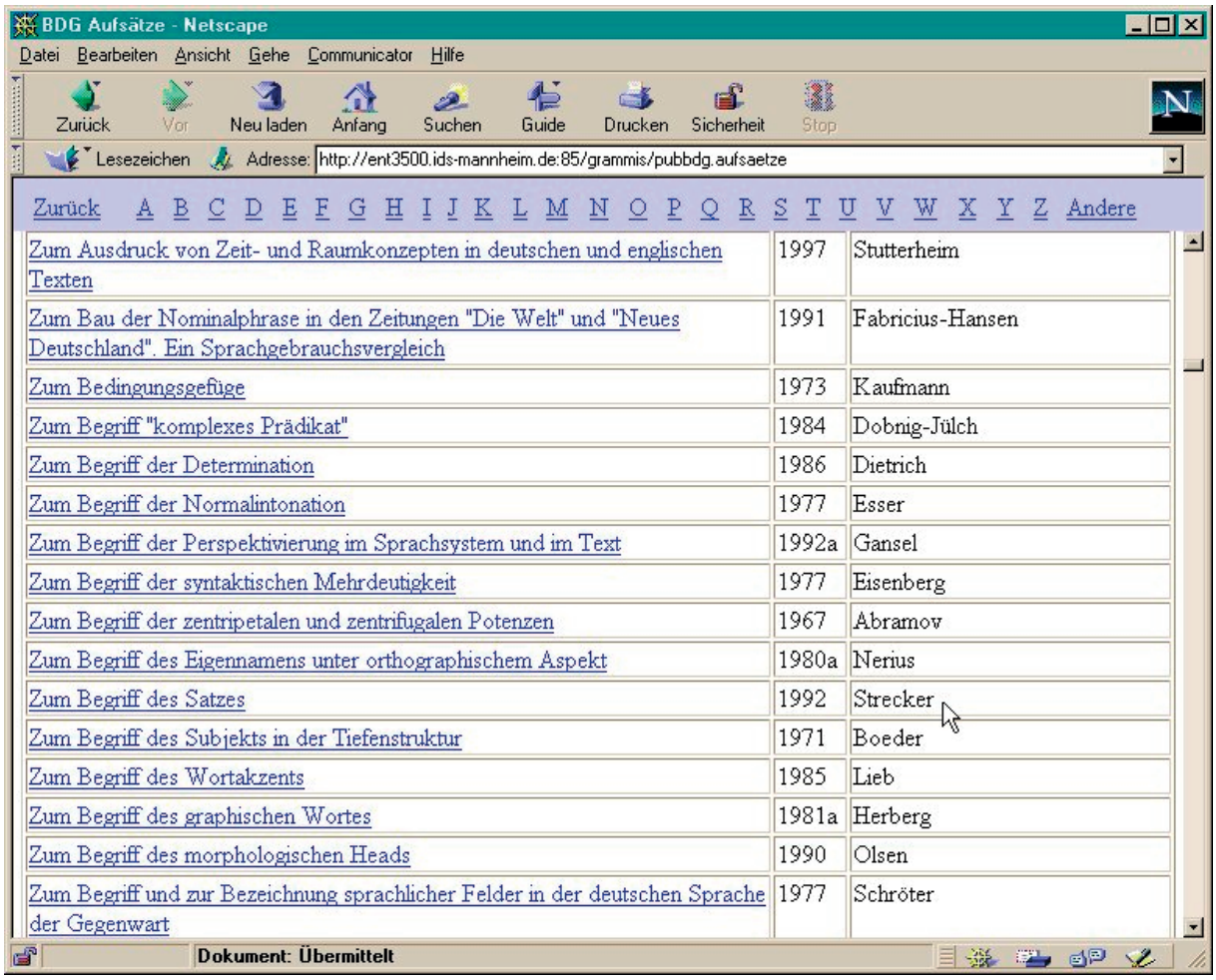


Figure 4: WWW presentation of all collected articles

After the user clicks on a single article title, the system reads all information available for this entry and presents the result in a new browser window.

If the article is published in an anthology, the system offers a hyperlink to this book. The same goes for articles published in journals etc. When clicking on this hyperlink (see figure 5 below), the user is connected with a page presenting information about the anthology (see figure 6 below).

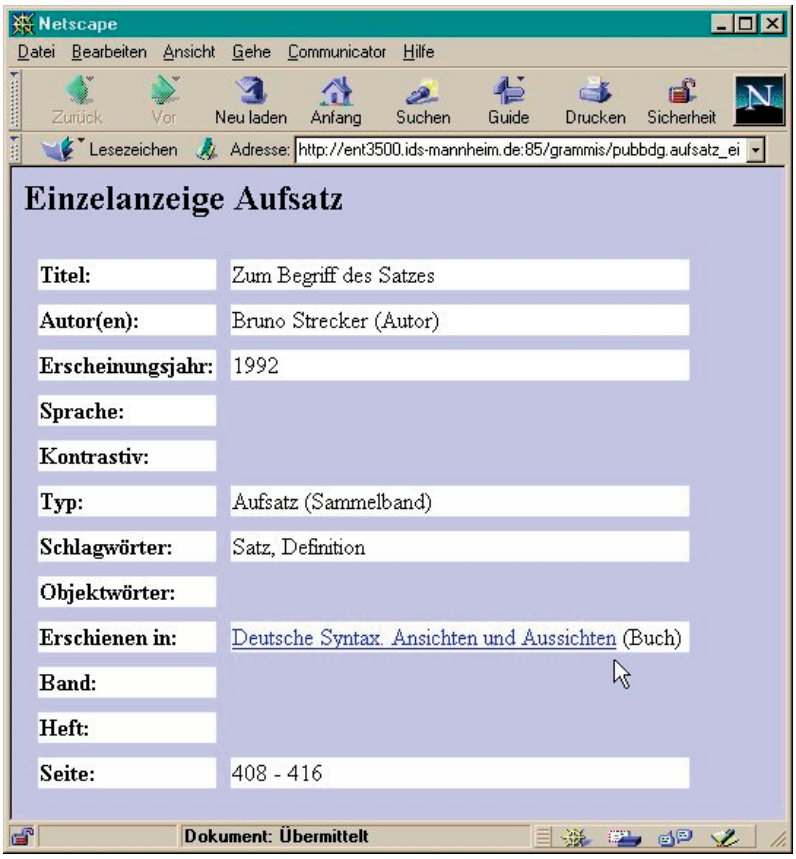


Figure 5: WWW presentation of a single article entry

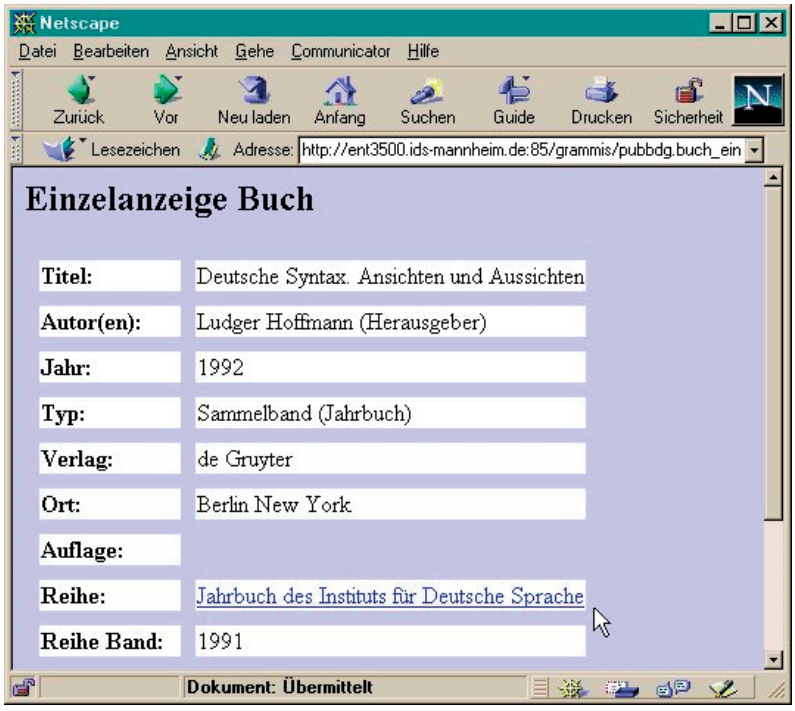


Figure 6: WWW presentation of a single book entry

Furthermore, the system automatically checks the database table of stored periodicals and builds a hyperlink connecting a book with the series it belongs to.


Figure 7: WWW presentation of a single periodical entry

Even if enriched with detailed illustrations, no text-based presentation can substitute the individual exploration of web-based applications. As already mentioned, the Bibliography of German Grammar is offered as a part of the grammatical information system GRAMMIS. To get a real-life impression of the database retrieval options, please visit us at http://www.ids-mannheim.de/grammis/.

## 5. References

[1] E. Breindl, R. Schneider, B. Strecker (2000): GRAMMIS – Ein Projekt stellt sich vor, In: Sprachreport. Informationen und Meinungen zur deutschen Sprache. Heft 1/2000, 19-24.

[2] C.J. Date (1995): An Introduction To Database Systems, Sixth Edition, Addison-Wesley Publishing Company.

[3] P. Eisenberg, A.H. Gusovius (1985): Bibliographie zur deutschen Grammatik. 1965-1983. Narr.

[4] P. Eisenberg, A.H. Gusovius (1988): Bibliographie zur deutschen Grammatik. 1965-1986. Narr.

[5] P. Eisenberg, B. Wiese (1995): Bibliographie zur deutschen Grammatik. 1984-1994. Stauffenburg.

[6] P. Koletzke, P. Dorsey (1998): Oracle Designer Handbook, Second Edition, Oracle Press, Osborne McGraw-Hill.