Search in PDB

Based on a lecture/workshop held at the

Annual Meeting of the Schwalbe 2019 in Bielefeld

Thomas Brand

Contents

List of Figures

Preliminaries

"Chess **P**roblem **D**ata**B**ase" (PDB) is a free data base containing more than 425,000 published chess problems of all kinds. You may search in PDB with very flexible criteria; you can freely access it under

https://pdb.dieschwalbe.de

PDB was developed by Gerd Wilts and Hans Peter Reich. It was maintained for some 20 years by Gerd Wilts, and for the last two years on a commercial server funded by "Schwalbe, deutsche Vereinigung für Problemschach e.V.", the German Problem Chess Society, administrated by Gerd.

This paper will provide support in searching chess problems in PDB. It is not intended as a comprehensive manual to describe all PDB details, particularly the technical ones, but it will provide an introduction into using PDB and additionally show helpful information integrated into PDB.

I'm very grateful to Bernd Gräfrath and Gerd Wilts for carefully proof reading the (German version of this) text and additionally Gerd Wilts for many helpful discussions and hints. Without him I would not have been able to provide this paper. Indeed, this paper would be useful for no-one, since without him there would be no PDB at all.

I'm also very grateful to Andrew Buchanan for his proof reading of this English translation.

Please send any comments or corrections to t.brand@gmx.net

Bornheim, December 15, 2019 Thomas Brand

Site Structure

PDB internet link:
https://pdb.dieschwalbe.de

HINT PDB site is bi-lingual English / German. If it starts in German language you may switch to English by clicking `Englisch` in the upper menu line OR define English as main language when starting PDB with the URL

https://pdb.dieschwalbe.de/index.jsp?langw=EN

TIP You may save this address in your browser favorites to start immediately on the English site.

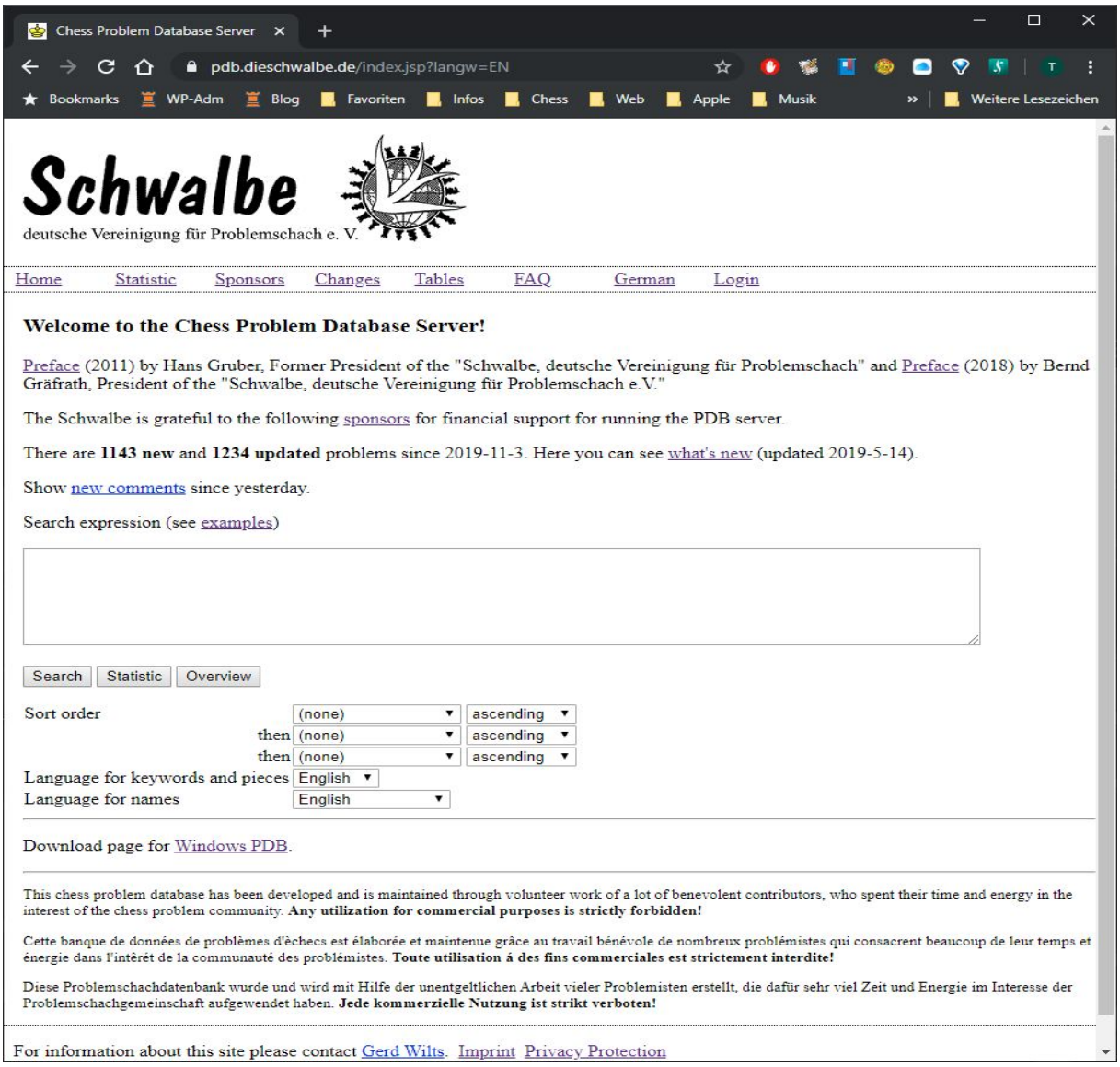

Figure 1 PDB start screen

Let's start with the (upper) menu line:

- **`Start`** Back to the start screen
- **`Statistic`** General statistical information on PDB size and usage
- **`Sponsors`** List of the PDB sponsors supporting "Schwalbe, deutsche Vereinigung für Problemschach e.V." by a donation for providing PDB
- **`Changes`** new comments and problems for the last ten days
- **`Tables`** We'll come back to them later in detail
- **`German`** Switch to German user interface.
- **`Login`** is not required for searching. We will come back to login in the last section of this text.

The subsequent text above the "big empty box" provides some additional links:

- Two **`Prefaces`** by the (former) Schwalbe presidents, dealing with PDB's aims and history. Worth reading, but mostly in German!
- An additional link to the **`sponsors`** list
- **`what's new`** A chronologic overview on updates of PDB software and integration of different problem collections
- **`new comments`** leads directly to a predefined PDB search for problems with new comments since the previous day
- **`examples`** File of PDB example queries and methods to combine queries. We will deal with this topic in detail in the next two sections. **TIP** You might want to keep open this page in parallel to your "PDB working site"; so it's easy to find help to formulate PDB queries.

The "big empty box" in the center will be our main working place for our searches: it is the editor box for our queries. Search will be started by pressing one of the buttons below the box, but first have a look below these boxes.

Here you may define sort order for the problems found, e.g. ascending for publication year, descending for piece count. Define these before starting the search!

We will deal with the meaning of the three buttons later on; now we will have a look to the PDB query editor, our "big empty box".

What Can Be Searched For?

Here we deal with which searches are possible and how queries have to be formulated.

Please keep the “example” page opened in a separate window. The direct address is:

https://pdb.dieschwalbe.de/article.jsp?idx=3

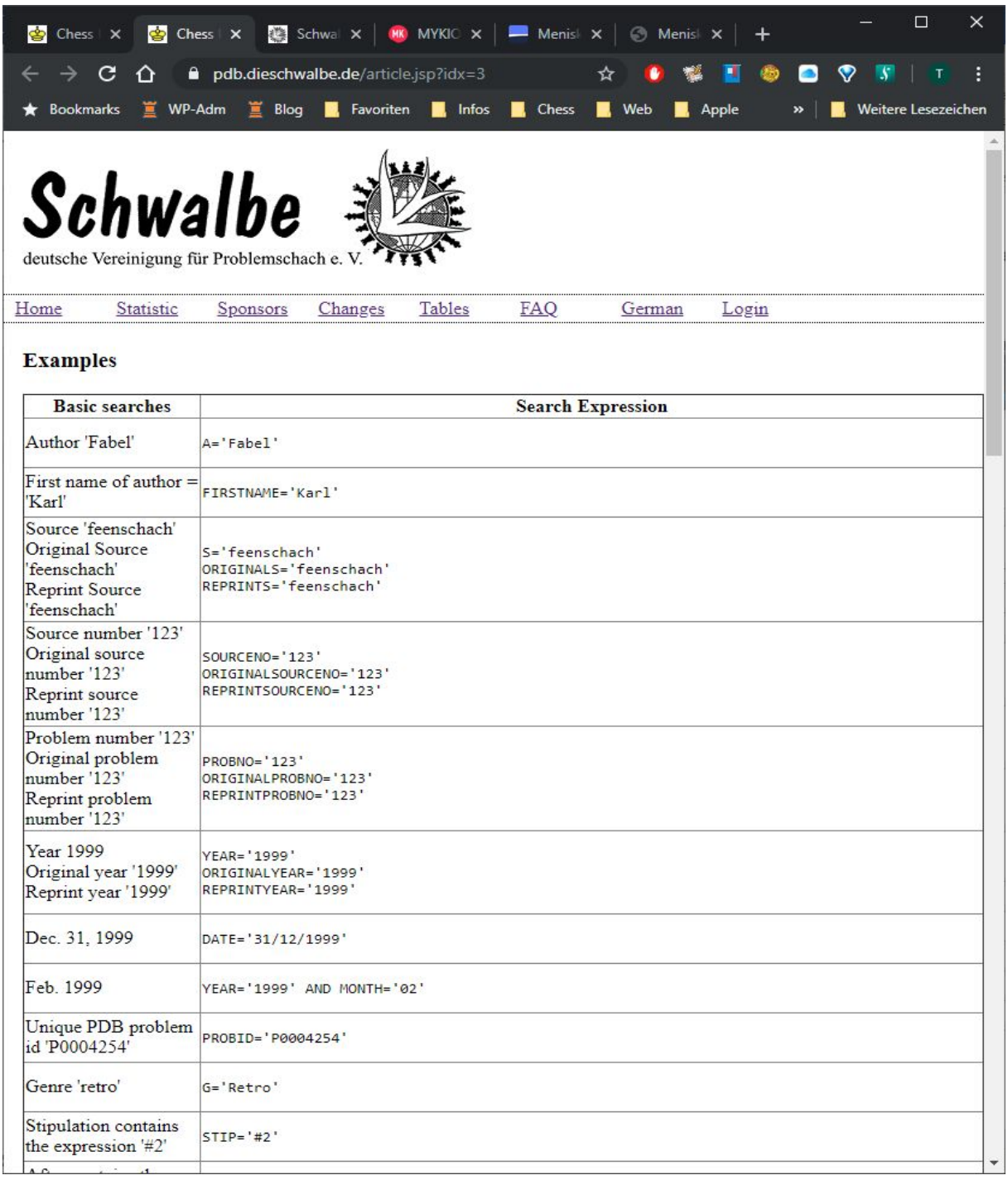

Figure 2 PDB ‘Examples’ page

The structure of a simple query is:

- Start with a token (e.g. `PROBID`)
- Then a relational operator follows (e.g. an equal sign)
- The third part is a search parameter included in single quotes.

This has to be entered into the "big empty box", the query editor in the middle of the PDB main page. To start the query you have to press one of the three buttons:

- **Search:**
  Will display the matching problems immediately (max 100 per page; to see more you have to request them actively at the end of the page).
  TIP If you expect just a small number of resulting problems, use "Search".
- **Statistic:**
  Will show a statistical report on the hit quantity (helpful for a first impression of the query results).
- **Overview:**
  Will also provide statistical information and additionally allows you to refine the query and to show the resulting problems of the (refined) query.
  TIP If you have no clue about the number of resulting problems or you might want to refine the query, use "Overview".

Now we want to formulate some queries and examine the results:

TIP Enter the queries as demonstrated here into the query editor and modify them to see what happens.

- Search for a composer via his **name**:
  A (and if necessary FIRSTNAME)
  EXAMPLES
  `A='Brand'` *or* `A='Brand' AND FIRSTNAME='Thomas'` and press the "Overview" button.
  HINT Only use 'upright single quotes' on your keyboard! If you are entering from a mobile phone, keep the apostrophe character pressed, and a range of other quote characters will appear: you must pick the upright one.

Now you should see something like this:

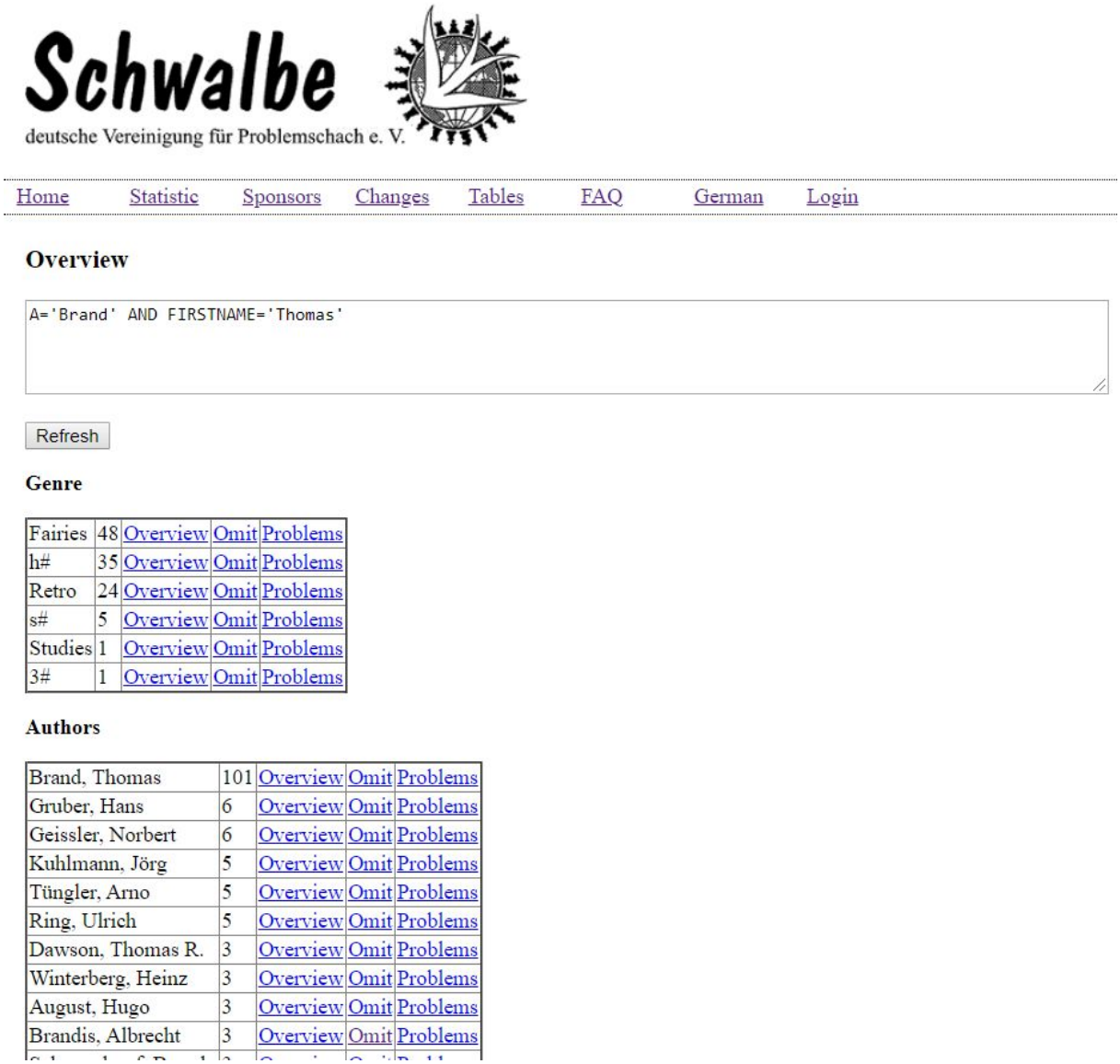

Figure 3 Query `A='Brand' AND FIRSTNAME='Thomas'`

You see many names – most of them resulting from joint compositions – but what about "Albrecht Brandis"?

`A='Brand'` not only matches "Brand" but also matches with the prefix "Brand" of " Brandis". And we also find Thomas R. Dawson in the results because he had joint compositions with Brandis.

TIP We can slightly refine our query resulting in a list of problems only composed by Thomas Brand:

```
A=='Brand' AND FIRSTNAME=='Thomas'
```

The double equal signs look for exact matching and not for names with the search parameter as prefix. Try out the difference and examine the new overview result.

HINT Alternatively you might click "Omit" in the Brandis column to exclude his problems from further processing, or you might edit your request in the editor field and press "Refresh".

HINT Sometimes it's confusing that there is NO implicit link of parts of the queries. The query `A='Müller' AND FIRSTNAME='Henry'` yields a hit, although there is no Henry Müller in PDB – but a joint composition of Henry Tanner and Frank Müller.

- Search for **sources**: S, ORIGINALS, REPRINTS, different problem numbers, dedications and distinctions:
  "S" searches for a source of a problem. This might be the original source, but also the source of a reprint. You may refine your query: "ORIGINALS" searches for the original source, "REPRINTS" for the source of reprints.
  HINT As mentioned above there is no implicit link of parts of a query: `S='Die Schwalbe' AND SOURCENO='123'` will find not only problems published in *Die Schwalbe*, issue 123, but also problems with original source *Die Schwalbe* with a reprint in, say, *feenschach* issue 123. Here `ORIGINALS='Die Schwalbe' AND ORIGINALSOURCENO='123'` might be helpful.
- Search for **date information**: YEAR, ORIGINALYEAR, REPRINTYEAR; DATE; MONTH.
  Here the same general differentiation as for the sources is possible, say year of original publication and year of reprints. You may also add month (MONTH) or even the exact publication date (DATE).
  HINT "DATE" uses a slash "/" to delimit day, month, and year.
  HINT The correct DATE format is <dd/mm/yyyy>, e.g. 31/12/2019.
- Search for the **genre** of problems: Here the eight *FIDE Album* sections are used (2#, 3#, n#, Studies, s#, h#, Fairies, Retro) plus "r#" and "mathematics".
  Example: `A=='Brand' AND G='Retro'` matches the Retro problems of this document's author.
- **PROBID** – this is the famous "**P number**" sometimes used to reference problems in PDB. This number is unique, so it's the easiest war to identify a problem in PDB.

Example: `PROBID='P0501340'`

HINT Even though the "P" is redundant you have to give it in the search parameter.

Now we have dealt with most of the info above a diagram. There are some additional ones, e.g. "D" for dedications, "AFTER" for acknowledgement, "AWARD" for distinctions. You will find details in the example page.

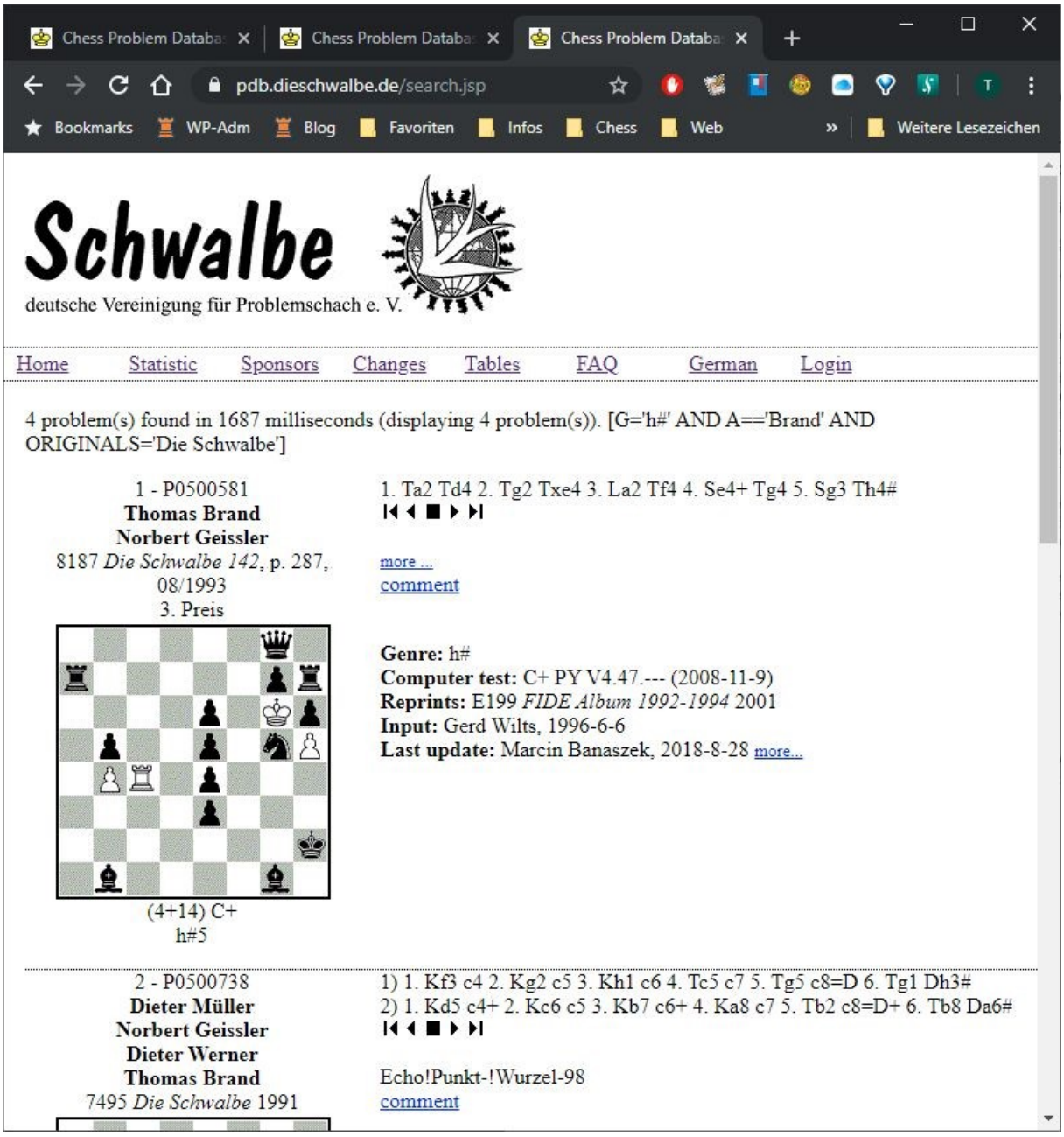

Figure 4 Result of a query

We can make the explanation shorter for the information typically printed below a diagram, since now you know the principles of querying.

- Search for the **stipulation** with the token `STIP`.
  Example: `STIP='h#4'`.
- Search for the number of pieces in the diagram: (white, black, neutral, all)
  Here we prefix the token `PIECES` with a letter: (`W`, `B`, `N`, `A`).
  Example: `APIECES <= 7 AND NPIECES > 0` matches all miniatures with neutral men.
  HINT Do not use quotation marks for the numbers!
- Additionally we have two tokens which, I think, are self-explanatory: `CPLUS` and `COOKED`.

Now we want to deal with information belonging to the diagram itself:

- `PIECELIST` asks for pieces **present in the diagram**.
  Uppercase letters are used for white, lowercase letters for black pieces.
  Examples:
  `PIECELIST='KkLs'` There are *at least* a black king, and white king, a white bishop and a black knight in the diagram.
  `PIECELIST=='KkLs'` (two equal signs!) There are *exactly* a black king, and white king, a white bishop and a black knight in the diagram.
  HINT Use German abbreviations KDTLSB for KQRBNP!
- You can also check for the **position** of pieces in the diagram:
  `POSITION='wTa1 wSb1 wLc1 wDd1'` looks for diagrams with these pieces exactly on the given squares.
- Also you might check for **patterns** of pieces in the diagram:
  `PATTERN='wTa1 wSb1 wLc1 wDd1'` looks for diagrams with the given pieces in exactly this pattern, but the pattern might be shifted, turned, or mirrored. So this query e.g. matches wTh5 wLh4 wSh3 and wDh2. This query usually needs a little bit longer search time.
- You might also search for fairy pieces used:
  `PIECE='Erlking'` matches problems with an erlking in the diagram.
  HINT You will find the fairy pieces recognized by PDB in the “Fairy Pieces“ table you may access via “Tables / Fairy Pieces“ in the PDB menu or directly under the URL https://pdb.dieschwalbe.de/pieces.jsp. This table does not only list the recognized pieces, but also contains piece definitions and a specific search function.

TIP If you work much with fairy pieces keep this table opened in an extra browser window.

- To find any text in the **solutions** or the **comments**:
  `SOL='1. e4'` looks for the key move (P)e4.
  HINT For the PDB solution animation feature to work there must be a space between the move number and the move itself. Any exceptions to this you see in a solution are errors. This blank should be respected in your queries.
  `SOLTEXT='Albino'` The solution contains the word "Albino".
  `COMMENT='Albino'` The comment contains the word "Albino".

Now we come to the most powerful search method: **keywords**. These keywords distinguish the problems, they inform about relevant themes in a formal way, so these problems can be found by queries.

HINT Please keep in mind that this indexing is not be done automatically, but by a person recording or commenting the problem for PDB – and so errors can't be excluded.

TIP Keywords are listed in a dedicated PDB table you will find via the PDB Menu (Tables / Keywords) or directly via https://pdb.dieschwalbe.de/keywords.jsp. Keep it open if you use it intensively.

TIP The table is divided into sub-tables (Fairy Conditions, Themes, Stipulation, Description). "All" gives the complete view of all keywords.

HINT Like the Fairy Pieces table the keywords in these tables sometimes contain definitions. If a definition exists, it will be displayed in a sub-window of the results browser window, if you click a keyword. After a second click this sub-window will disappear.

Some keywords are assigned to a lot of problems (some 10,000 Allumwandlungs!), so you might sometimes wish to filter your search.

Among other topics we will deal with this question in the next section.

Filter Search

In the last section we learned a lot of different ways of search, we just combined different searches by `AND`.

We not only can combine simple searches by AND, but also by all **Boolean Operators**:

- To match `A AND B` both A and B must match (so `A AND B` shrinks A).
- To match `A OR B` A or B (or both) must match (so `A OR B` expands A).
- `NOT A` matches if and only if A does not match.

Using **brackets** you can combine expressions and change priority:

- `AND` binds more strongly than `OR` – this can be changed by brackets.
  Example: `A=='Brand' AND G='Retro' OR G='h#'` is equivalent to `(A=='Brand' AND G='Retro') OR G='h#'`, but this has a different meaning to `A=='Brand' AND (G='Retro' OR G='h#')`
  The last query results in much less matches than the first ones. It searches for problems by Brand with at least one genre "helpmate" or "retro", while the other ones search for Brand's retros or any (so: all!) helpmates.
- In queries you may also use **placeholders**:
  - '`%`' for any (even an empty) string:
    `A='Br%d'` matches Br**e**de, but also Br**an**d, B**ree**de and Br**enan**der
  - '`_`' for exactly one character:
    `A='M__er'` e..g matches M**ai**er, M**ay**er, M**ei**er, M**ey**er, M**os**er
    HINT These placeholders also work for searches in the Albrecht Twomovers Collection, maintained by Udo Degener (http://www.schach-udo.de/dab/daten.htm).
  - '`[x]`' matches one character out of a given sequence of characters:
    `A='M[ae][iy]er'` matches Maier, Mayer, Meier, Meyer
- Filtering the keywords:
  - If you want to search problems with the fairy condition "Anticirce", the query `STIP='Anticirce'` will match, but

there is no keyword like “Anticirce”. The keyword “Circe” is used for the whole Circe family of conditions. The **colon** is used to filter keywords, here `K='Circe:Anti'`.

This example shows how a keyword (in this case "Circe") may have a parameter after it (in this case "Anti"). While inexact matching for keywords is by prefix (e.g. "Circ" will work), inexact matching for parameters is by substring (e.g. "nti" will work). In this way you may search for thematic pieces with some keywords:

- `K='Pronkin:DT'` searches for problems with Pronkin promotions of queen and rook. It will only match those problems where the thematic pieces appear in the parameter string in just this order.
- So we understand: `K='Pronkin:DT'` and `K='Pronkin:TD'` are different! And both queries don’t match Pronkin problems with promoted men noted as TLD!
- Here we may write: `K='Pronkin:D;T'` This (delimit the pieces by semicolon , which means OR) will match all three examples!
- This doesn’t work for multiple occurrences of one kind of pieces. For searching Pronkins with two rooks we might write `K='Pronkin:T%T'` (matches TT and TLT, too)
- If you want to search for any Pronkins without queen promotion, but promotions to rook, bishop and knight, you might write:
  `K='Pronkin:T;S;L' AND NOT K='Pronkin:D'`

- Re-use queries!
  - By pressing “Home“ in the main menu you will find the last query in the main editor window, where you might modify it for your next query.
  - It's a good idea to save complicated queries in a pure text file for future use.

  TIP In this way you can build up your own library of (complex) queries.

  TIP Use a pure ASCII editor, not a text processing program like Word. These programs tend to replace your 'upright single quotes with more “beautiful“ ones – this would result in PDB syntax errors when copying back these queries to the PDB main editor.

Improve PDB Data Quality

This section deals with the question how we may all improve PDB data quality. Support further PDB development!

To do so you must create a PDB account. Hit “Login” on the start page, and then "create an account". You must enter a valid email address (for password reset 😉), login name and password. Of course data protection is ensured.

Now you can comment on problems. Send a mail to Gerd Wilts to request additional functions:

- Add new problems to the PDB
- Correct problems (e.g. diagram errors)
- Index new or existing problems with (additional) keywords.

The last topic is extremely important, since thematic search is mostly done via keywords. Queries will only match problems with appropriate keywords!

To help you support further PDB development an introduction into editing PDB problems is planned.