# THE CHESS QUERY LANGUAGE: CQL

*G. Costeff* [1]

San Francisco, CA, USA

ABSTRACT

The Chess Query Language, CQL, allows for the matching of complex thematic requirements that cannot be accomplished using prior facilities. The target users of CQL are writers, researchers, composers, composition tournament directors, and judges. This combination describes how CQL works, the reachable levels of thematic complexity, and the history and thinking behind it.

## 1. INTRODUCTION

The recording of chess games and chess problems start with the Arab manuscripts of the 9th century (Hooper and Whyld, 1996). For over one thousand years, paper continued as the sole medium of record for chess information. To facilitate the search for specific themes required first collecting a significant number of games or studies, then manually classifying them according to some system (cf. Harman, 1967; Van der Heijden, 1998). The result reflected the limitations of classification as well as the slow rate of manual search.

Around 1990 the computer replaced print as the primary chess archiving and research tool. Large collections of games and studies (Van der Heijden, 2000) became available and one could use the software search facilities to look for positions with specific characteristics such as move sequences, material balance, some measure of logical operators, and a host of game data such as players, opening type, rating, result and so on.

Tools such as CHESSBASE™ were an important step forward, but probably because most customers are players, not researchers, the search facilities in such commercial products were still insufficient to describe a variety of more complex themes and theme combinations. This is illustrated by the queries presented in this article, none of which can be currently described with non-CQL tools. As a result of this state of affairs, thematic research remained largely dependent on past classification systems, as they appeared in books or in large personal collections (see web link karlonline).

In 2001 the 7th tri-annual World Chess Composition Tourney was announced with a challenging theme in the studies section (see web link saunalahti). The author was discussing the theme with Lewis Stiller when we wondered how many published examples of the theme exist. There were no tools to answer the question so we set out to find out on our own. This was the beginning of CQL.

## 2. DESIGN OF CQL

CQL was designed to search for arbitrary combinations of positions, moves, and relationship attributes that can describe a vast number of chess themes. To achieve its goals CQL was implemented as a general query language along the lines of the ubiquitous SQL (see web link). Other designs were considered, but the need for descriptive power coupled with a user base of non-programmers made a query language a natural solution.

Accordingly, CQL contains a rich set of primitives (tags), which can be combined with the logical operators (OR, NOT, AND) to produce arbitrarily complicated constructs. This design allows CQL to describe a wide variety of complex themes and theme combinations.

---

[1] 178 Andover Street, San Francisco, CA 94110, U.S.A. Email: costeff@yahoo.com.

Unlike SQL, CQL is domain specific. Consequently, it was possible to take advantage of chess nomenclature. For example, terms such as: *position Kb6 Pc6 ka1 rd5*, *check*, *mate*, and *stalemate* are familiar to any chess player. As a result, most CQL tags are immediately comprehensible to even non-technical users.

CQL operates on any game, study, or problem database. The only requirement is for the database to be in PGN format (see web link), which enjoys universal support. Each game is read, parsed, and then filtered against the query. If the game matches the query it is written into the output file. The output file is in PGN as well, so the results may be viewed with a variety of software. It also means that CQL output can be used as input to another CQL query.

CQL treats each game as a collection of all the positions within it. For example, a 40-move game with another 30 moves in the variations yields 140 distinct positions. The most important consequence of this design is that the main line presentation (in a study) and the scope and depth of variations (for games and studies) may significantly affect query results. The syntax of a CQL query has the generic structure given in Figure 1. The Subsections 2.1 to 2.5 provide five distinct examples of the use of CQL.

```
(match
 :pgn input_filename
 :output output_filename
   (position ... )+
)
```

**Figure 1:** The generic structure of a CQL query.

In this example, the "..." represent various keywords and commands and "+" means that one or more **positions** can be specified.

## 2.1 A first CQL example

In Figure 2 we provide an example of a specification of the generic structure of a CQL query.

```
; stalemate.cql         ; comments begin with a semicolon (;)

(match
:pgn heijden.pgn        ; heijden.pgn is the input database
:output stalemate.pgn   ; stalemate.pgn is the query result database
(position
 [KQRBNkqrbn]d2         ; 4 officers of any color on d2
 [KQRBNkqrbn]d3         ; on d3
 [KQRBNkqrbn]d4         ; on d4
 [KQRBNkqrbn]d5         ; and on d5

 :stalemate             ; the position is stalemate
 :shift                 ; The piece configuration is location
                        ; independent
 :flip                  ; The piece configuration is orientation
                        ; independent
 :variations            ; include variations in the search
))
```

**Figure 2:** An instantiated example of the generic structure.

The ***:shift*** and ***:flip*** tags are rather powerful, allowing for succinct description of large sets of specific configurations. The above query yields 99 studies with the following two stalemate positions (Diagrams 2 and 3) from the same study by Kozirev (1996) (see Diagram 1).

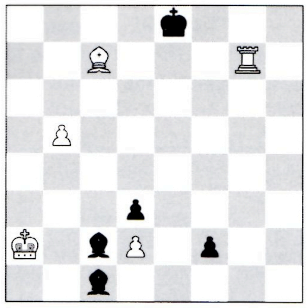
**Diagram 1:** Kozirev (1996).

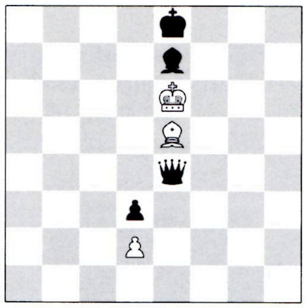
**Diagram 2:** Stalemate 1.

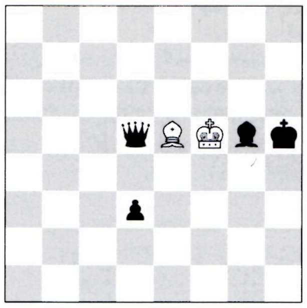
**Diagram 3:** Stalemate 2.

1. b6 f1Q 2. b7 Bb3+! 3. K:b3 Qd1+ 4. Kc4! Qa4+ 5. Kd5! Qb5+ 6. Ke6! Qb3+ 7. Kf6! Q:b7 8. Re7+ Kf8 9. Rf7+ Kg8 10. Rg7+ Kh8 11. Be5! with:
11. ...Qe4! 12. Rc7! Ba3 13. Ke6+ Kg8 14. Rg7+ Kf8 15. Rf7+ Ke8 16. Re7+ B:e7 stalemate (Diagram 2).
11. ...Qd5! 12. Rg3! B:d2 13. Kf5+ Kh7 14. Rg7+ Kh6 15. Rg6+ Kh5 16. Rg5+! B:g5 stalemate (Diagram 3).

## 2.2 A second CQL example

There are other ways to specify location-independent and orientation-independent themes. Figure 3 provides an example.

```
; mirrormate.cql

(match
 :pgn heijden.pgn
 :output mirrormate.pgn

 (position
  :attackcount k . 8    ; the black King attacks 8 empty (.) squares
                        ; ( no pieces next to the black King )
  :mate
))
```

**Figure 3:** An example of the specification of location-independent and orientation-independent themes.

Diagram 5 is one of the more striking results (Roycroft and Blundell, 1993). The starting position is given in Diagram 4.

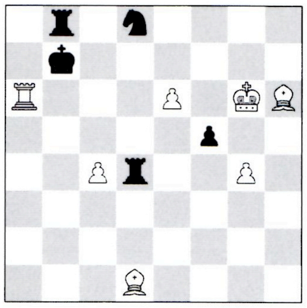
**Diagram 4:** Roycroft and Blundell (1993).

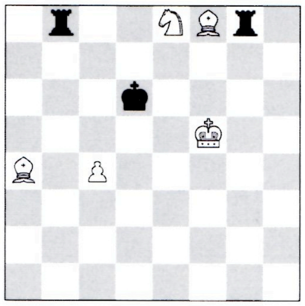
**Diagram 5:** What was White's last move?

1. e7 Nc6 2. Rxc6 Kxc6 3. Ba4+ Kd6 4. Bf8! Rxg4+ 5. Kxf5 Rg8 6. e8N! mate

### 2.3 A tagging example

Another powerful feature of CQL is *tagging*. Tagging is the notion of marking a piece and thus being able to follow its movements along the game score. The query of Figure 4 looks for games that contain the Bristol (see web link) theme along the diagonal.

```
; bristol.cql

(match
 :forany head [QBK]      ; define the labels head and tail as
 :forany tail [QBK]      ; white queen, bishop or king
 :pgn heijden.pgn
 :output bristol.pgn

 (position
 :shift :flip :wtm       ; white to move
 :tagmatch head [QBK]b2  ; find diagonally adjacent white pieces
 :tagmatch tail [QBK]a1  ; and assign them the labels head and tail
 :sequence (             ; a sequence of positions (ply) follows
  (position :movefrom $head :moveto .[c3,d4,e5,f6,g7,h8])
  (position :nocheck  :raydiagonal ($head $tail))
  (position :movefrom $tail :moveto .[b2,c3,d4,e5,f6,g7])
                         ; tail follows the head = Bristol theme
```

**Figure 4:** A query for the Bristol theme.

Tagging opens a wide range of possible themes for exploration. A prime example is the family of geometric shapes such as when the same piece visits all 4 corners.

### 2.4 An abstract example

Our next example shows an *abstract* theme. Such themes are made of more complex relationships than the explicit piece configurations or routes used by the previous examples. A good example of an abstract theme is the query given in Figure 5.

```
; wcct7.cql

(match
 :pgn games.pgn
 :output wcct7.pgn
 :result 1-0             ; White won the game

 (position
                         ; a position with no parameters like here
                         ; matches everything

  :relation (:missingpiececount A 1 15) ; see text
))
```

**Figure 5:** A query for an abstract theme.

By default a ***:relation*** is identical to the encapsulating **position** but this can be modified using the parameters following the ***:relation*** tag. Here these parameters say that the related position is identical (implicitly) but at least one white piece is missing. ('**A**' representing any white piece, "**1 15**" the range of the number of missing pieces.) This example was run against a games database returning among others a famous figure of 19th century chess (see Diagram 6).

**Blackburne, J - Schwarz, A**
**Vienna (1), 28.07.1873**
1. e4 e6 2. d4 d5 3. Nc3 Bb4 4. exd5 exd5 5. Bd3 Be6 6. Nf3 h6 7. 0–0 Bxc3 8. bxc3 Nf6 9. Ba3 Nc6 10. Re1 Ne7 11. c4 c6 12. cxd5 Nfxd5 13. c4 Nf6 14. Rb1 Qc7 15. Qb3 0–0–0 16. Qa4 a6 17. Bc5 Rde8 18. Qa3 Nf5 19. Bb6 Qf4 20. Ne5 Nd7 21. Qb2 Nxe5 22. Rxe5 Ne7 23. Bc5 b5 24. Qa3 Kd7 25. Qxa6 Rc8 26. cxb5 Rhd8 27. bxc6+ Ke8 28. Qa7 Qf6 29. Rb7 Nxc6 30. Bb5 Qxe5 Match (Diagram 6) 31. Re7+ Kf8 32. Re8+ Kxe8 Match2 (Diagram 7) 33. Qe7 mate

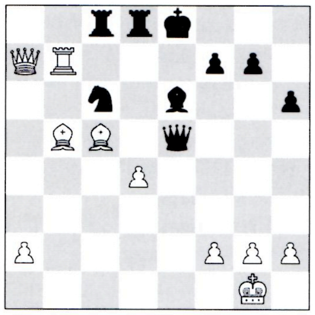
**Diagram 6:** After 30. ... Qxe5.

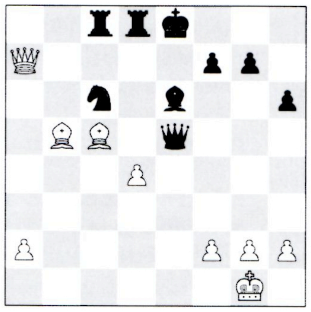
**Diagram 7:** After 32. ... Kxe8.

After 31. Re7+ Kf8 32. Re8++! Kxe8 (Diagram 7) the same position occurs as in the Diagram 6 but Rb7 is now missing, clearing the way for 33. Qe7 mate.

Of course the term *abstract* is a human label reflecting the author's perception that the lack of a specific piece placement makes the search quite difficult to humans. For CQL this is a straightforward comparison.

## 2.5 A fortress example

If the above query was straightforward, our final example is anything but. In fact, I originally wrote "However, there are themes, such as fortress, that are beyond even CQL." Then I started thinking about what a fortress is in chess terms and I found the following:

1. the result is a draw;
2. Black has a significant material advantage;
3. Black has significant freedom of movement;
4. the white Pawns are static;
5. there are no captures;
6. White has no threats.

Other than item 6, all attributes can be defined in CQL (see Figure 6). Below we provide two examples of a fortress in the form of a study, viz. by Gurvitch (1952) and by Simkovitch (1923).

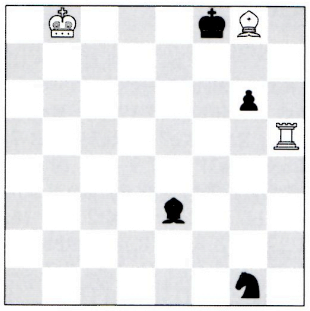
**Diagram 8:** Gurvitch (1952).

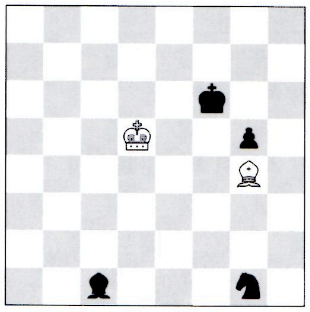
**Diagram 9:** Positional draw.

A. Gurvitch
1st Prize
Dagestan Committee Fizkultura i Sport 1952 (Diagram 8)

Kb8,Rh5,Bg8 = 3
Kf8,Be3,Ng1,Pg6 = 4
1. Rh8! Kg7 2. Bh7! g5! 3. Bf5!! Kxh8 4. Bg4!! Kg7 5. Kc7 Kf6 6. Kd6 Bc1 7. Kd5 Ba3 8. Ke4 Ke7 9. Kd5 Kd8 10. Kc6 Bc1 11. Kb7 Ke7 12. Kc6 Kf6 13. Kd5 positional draw (Diagram 9)

```
; fortress.cql

(match
 :pgn heijden.pgn
 :output fortress.pgn
 :result 1/2                        ; the result is a draw

(position
 :piececount A 2 16
 :piececount Pa-h7 0               ; white does not threaten to promote
 :attackcount K a 0                ; white is not in check
 :wtm
 :sequence (                       ; The structure is a sequence of 4 ply
(position
 :attackcount A [nbrq] 0           ; white attacks no black officers
 :powerdifference U -1000 -4       ; black's ahead at least 4 pawns
                                   ; ( P=1 [NB]=3, R=5, Q=9 )
 :nocheck
 :movefrom [KQRBN]                 ; no white pawn move
 :moveto . )                       ; no capture: (move to an empty square (.)

 (position
 :attackcount A [nbrq] 0
 :powerdifference U -1000 -4
 :nocheck   :moveto .)

 (position
 :attackcount A [nbrq] 0
 :powerdifference U -1000 -4
 :nocheck   :moveto .
 :movefrom [NBRQK])

 (position :and (
    (position   :not :stalemate)
    (position   :attackcount A [nbrq] 0
                :powerdifference U -1000 -4
                :nocheck  :terminal
                :attackcount [qrbn]  . 4 100)
)))))
```

**Figure 6:** The definition of the attributes 1 to 5 for a fortress.

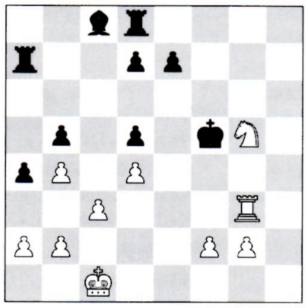
**Diagram 10:** Simkovitch (1923).

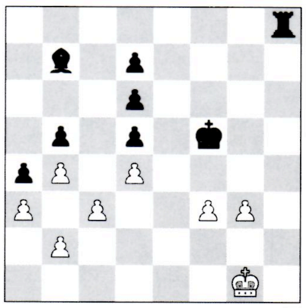
**Diagram 11:** Positional draw.

F. Simkovitch
3rd H.M., L'Italia Scacchistica 1923
(Diagram 10)

Kc1,Rg3,Ng5,Pa2,b2,b4,c3,d4,f2,g2 = 10
Kf5,Ra7,Rd8,Bc8,Pa4,b5,d5,d7,e7 = 9
1. Nf7 Re8 2. Nd6+ exd6 3. Rf3+ Kg6 4. Rg3+ Kf7 5. Rf3+ Ke7 6. Re3+ Kd8 7. Rxe8+ Kxe8 8. a3!! Bb7 9. Kd1 d1 Kf7 10. Ke1 Ra8 11. Kf1 Rh8 12. Kg1! Kf6 13. g3 Kf5 14. f3 Re8 15. Kf2 Re6 16. Kf1 Re3 17. Kf2 Re8 18. Kf1 Rh8 19. Kg2 Rh6 20. Kg1 Rh3 21. Kg2 Rh8 22. Kg1 positional draw (Diagram 11)

CQL's inability to identify threats beyond a single ply, accounts for just about all the 'false positives,' which in the above query amount to a quarter of the 90 studies returned. Removing some of the restrictions will increase somewhat the number of true hits but at a decreasing ratio. Similarly, once the results have been examined, better filters can be applied. For example, over a third of the 'false positives' can be eliminated if the query excluded the class N-NNP. Such a balancing act is typical of query systems and it takes knowledge of the data and what one is looking for to make an optimal query.

The *fortress* query demonstrates an important underlying concept of CQL. It is the combination of simple individual attributes that creates complex themes, which in turn can be further combined to make more complex themes, e.g., systematic manoeuvres. This mechanism accounts for CQL's ability to deconstruct what seems like abstract or fuzzy themes. Fundamentally, the mechanism accounts for the richness of chess.

## 3. APPLICATIONS OF CQL

CQL is being used for a variety of purposes. Chess writers use CQL to find material for articles and books. Study composers use it to research their ideas while tournament directors and judges use it for anticipation checking. In the latest development, players can now use CQL from within the CHESS ASSISTANT™ database product. Perhaps there will be other unanticipated uses of CQL.

Research by writers and composers has been the biggest beneficiary of CQL. The simplest search for historical examples of a theme was difficult, primary due to the lack of ***:shift*** capability. To mimic such a capability required dedicated researchers to input manually the configuration up to 64 times for each query. When it came to more complex themes requiring the notions of ***:relation*** or tagging, even dedication could not help. Without the ability to query directly source material, the use of published thematic collections remained the main research aid. These collections were most often too generic to find specialized themes. This meant that innovation depressed by the lack of a good research tool.

In this respect, CQL has already revolutionized chess research by providing results used for numerous articles and study tournaments. The speed savings are easy to quantify, typically between three and four orders of magnitude. More importantly, the new capabilities have redefined what thematic research can provide. Researchers can pursue themes that are more granular than ever before as well as more complex in terms of their combinations. The result is a far greater variation and richness.

An example of this can be seen in the writings of Tim Krabbé (see web link). His web site and books are dedicated to the most offbeat and strange themes, which now can be researched directly against a database of millions of games, determining within minutes whether quintupled Pawns had ever occurred in tournament play. His input has led to improvements designed to tackle the sort of themes that occur more often in games.

Similarly, given that less public advances have been made in originality checking of endgame study tournaments, the previously mentioned 7th tri-annual World Chess Composition Tourney was the start of a new era. Over one-thousand studies, the entire body of previous work matching the thematic definition, was published online for the benefit of composers and judges alike.

When Lewis Stiller and the author built CQL, our hope was that eventually commercial entities such as CHESSBASE™ would implement these (or their own) powerful search facilities within their products. It remains to be seen whether the CHESS ASSISTANT™ experiment to imbed CQL into their product is as useful for players as it is for researchers.

## 4. FUTURE DEVELOPMENTS

The adoption of CQL has exceeded its authors' expectations. Users worldwide supplied bug reports, comments, magazine reviews, and ideas for future developments. Some of the ideas led to new functionality, such as *tagging*, whereas others are deemed beyond the scope of CQL.

In general, however, the current functionality is sufficient for all but the most esoteric needs. Even those are typically addressed by more sophisticated query formulation rather than by adding new functionality.

There are no plans for further elaborations of CQL; however, other software tools that make use of CQL are a possibility. For example, automatic originality checking of studies would be highly desirable. The idea is to take a game, analyze its attributes and search for prior games exhibiting similar characteristics. At its simplest form this means searching for the existence of any of the current game's positions in the database. At a higher level this requires a more sophisticated approach that can weigh the relative importance of attributes and themes in a given game. This is a very ambitious project.

A more modest idea is to try to improve mechanically a game's annotation. This requires a chess engine, which would run as a preprocessor and complete the additional variations to a predetermined level of detail. Here too, one runs into the issues of how deep and wide to annotate, not to mention the fallibility of chess engines.

Within CQL itself there is much scope for improvement, especially to support the problem domain, where the internal relationship between moves and variations is important. For example, in three variations the same three white moves occur in cyclic order (ABC, BCA, CBA).

There are no current plans for any of these projects.

## 5. TECHNICAL MATTERS

CQL can be downloaded freely at http://www.rbnn.com/cql. It comes in a zipped archive that includes the query engine, help files, and examples. The same help and examples are available on the web site as well. The underlying CQL engine is written in C++ for efficiency. The code also uses the Scid library (see web link).

Running CQL on the Windows™ shell is simple enough though Emil Vlasák (see web link) has provided Visual-CQL, a utility that allows users to edit and run CQL queries from a more familiar windows environment.

There are no minimum hardware requirements as CQL is far more modest in its requirements than most native Windows™ applications. However, CQL is CPU intensive, so running a sophisticated query over a very large database will noticeably slow other applications.

The syntax of CQL is reminiscent of Lisp, however it is more closely related to query languages and shares some of their logical constraints. Although there are nominally nearly 70 tags, most users rarely use more than 20. The rest of the tags are parameters to other tags or rarely used optimizations such as ***:fliphorizontal***.

Although CQL is powerful when compared to current tools, like other query languages, it forgoes the full power of programming languages in order to be usable by a larger community. A future, thematically unlimited alternative to CQL could be CPL (Chess Programming Library), which would use a programming language such as Lisp allowing unlimited descriptive power.

CQL performance is only an issue when using very large game databases, which can contain several million games. With each game containing about 100 positions the search space is quite large, even without accounting for a specific query. In such cases Krabbé generates smaller thematic databases such as all games that end in mate. These can be used later to find specific themes that include a mate requirement.

The other way to speed up large database searches is to optimize queries. As in most query languages the performance depends on the data and the specifics of the query. However, a first-order optimization should take account of the fact that every ***:shift*** incurs up to a 64-fold increase in execution time.

## 6. CONCLUSION

CQL is a boon to chess research due to its great descriptive power and computer speed. This combination enables chess research to reach levels of granularity that were hitherto impossible. Consequently, CQL's immediate impact has been to enable a richer and more diverse chess literature as well as improved chess compositions.

## 7. ACKNOWLEDGEMENTS

The author gratefully acknowledges the fruitful discussions with Lewis Stiller, the contribution by Emil Vlasák, and the many recommendations by the anonymous referees (see Diagram 8).

## 8. REFERENCES

Gurvitch, A. (1952). Published study: 1st Prize, Dagestan Committee Fizkultura i Sport (see Diagram 8).

Harman, J. (1967). The Classification of Endgame Studies. *EG*, 7, pp. 24-31.
(http://www.gadycosteff.com/eg/eg7.pdf#page=24).

Heijden, H. van der (1998). My computerised collection. *EG*, 8, 130, pp. 400-413.
(http://www.gadycosteff.com/eg/eg130.pdf#page=38)

Heijden, H. van der (2000). Study Database 2000 (a collection of all known 58801 studies). CHESSBASE™ (chessbase.com).

Hooper D. and Whyld, K. (1996). *The Oxford Companion to Chess* . 2nd edition (November 1, 1996). Oxford University Press, Oxford, UK. ISBN 0192800493.

Kozirev, V. (1996). Study published as #54 in *64-Shakhmatnoye Obozrenye* (see Diagram 1).

Roycroft, J. and Blundell, D. (1993). Published study: 2nd H.M., Boris 10 Jubilee Tourney (see Diagram 4).

Simkovitch, F. (1923). Published study: 3rd H.M. L'Italia Scacchistica (see Diagram 10).

### 8.1 Web links

karlonline: http://www.karlonline.org/1_04.htm

saunalahti: http://www.saunalahti.fi/~stniekat/pccc/7_wcct.htm

Structured Query Language is the standard query language of relational databases (Oracle, Informix, Sybase, SQL-Server). There are thousands of sites devoted to SQL including
http://www.w3schools.com/sql/sql_intro.asp

Portable Game Notation. See (http://www.yacdb.com/pgn/pgn_lidx.htm) for the PGN specification.

For the Bristol theme definition and the seminal article about it see:
http://www.matplus.org.yu/BRISTOL.HTM

Tim Krabbé has written his own review of CQL (see item 242 in his chess diary:
http://www.xs4all.nl/~timkr/chess2/cql.htm)

The Scid library by Shane Hudson is part of a free chess database. See: http://scid.sourceforge.net

Emil Vlasák, http://web.quick.cz/EVCOMP/vcql.htm