

# 5

## Database Construction & Structure

When building a database, there are a number of important decisions to be made, and these decisions have profound effects not only on what the database looks like, but how it can be searched and used. In this chapter, we'll discuss two of these decisions: the construction of the inverted file and the use of structure.

First of all, it's not really necessary to use either of these in searching at all. The documents or document surrogates you want to include in your search system are usually stored in their full, native form in something called a **linear file**. Suppose you wanted to build an information system to allow you to search your old electronic mail messages. You might choose to simply save them in one big word processing file and not go to the trouble of building an inverted file or adding any structural elements at all. Almost all word processors have simple search facilities, and would allow you, say, to look for all the occurrences of the word "project" to find email messages from your boss about your current project.

Which might not be what you wind up with. Searching on the word "project" will get you documents which include that word, but there are other uses of "project" besides "the thing I'm working on now". It can also be used as a verb "I *project* this book will sell really well" or "*Project* the slides on that wall over there." Most word processor search engines also automatically truncate, so you'd get words like "projectile" and "projection" as well. Just some of the difficulties we encounter in searching. - JWJ

For collections of documents which are small or rarely searched, such a setup is probably not all that impractical. But, if you've got a database of several million documents, all of a sudden searching through each word of each document each time you're looking for something, makes use of both the inverted files and structure begin to look very attractive indeed.

### Inverted Files

The inverted file is used to make searching easier by providing access to the content-bearing words in all documents without needing to search through the entire texts of the documents themselves. The best way to discuss this is to show it, so let's walk through an example of the construction of an inverted file, using a small "toy" collection of documents.

#### Steps 0. Make some initial decisions.

Before actually starting the process of creating the inverted file itself, there are some important decisions to be made up front. For example:

*Collection & Coverage* What is the database about? How will you decide which documents will be included and which ones won't? Where will the collection come from?

*Technological Infrastructure* What kinds of hardware and software (including search engine) will you use in building and searching the database?

*Technical Details* What, if any, words will be on your stop list? What will you do with punctuation? Will you preserve capitalization, or not? What fields will you use for records? Will you include any phrases or just individual words?

The first two categories are beyond the scope of this chapter, but we'll discuss these technical matters as we go and see the effect the decisions will have. As an example, we'll construct a file in the way that DIALOG does. Although other systems will do things in somewhat different ways, especially Internet-based systems, the same basic steps apply.

### **Step 1. Make a list including all the words in each field of the record.**

Our toy database has three fields: title, abstract (really short abstracts, to be sure), and descriptor. We'll talk more about records and structure later in this chapter, but for now you need to know that a **field** is an individual piece of information about a document, and that a **record** is a collection of fields about the same document.

Here are our documents, with their document numbers:

101

The Origins of Don Giovanni

Discusses the history and sources Mozart used in his opera Don Giovanni.

DE: Mozart, Opera, Historical Analysis

102

Handel: Two Great Operas

Plot summaries, textual analysis of libretti, and musical explication of two of Handel's operas: Giulio Cesare (1724) and his first Italian opera, Rodrigo (1707).

DE: Handel, Opera, Musical Analysis

103

English Orchestral Music of the Early 18th Century

A discussion of the major features in English music of the mid-1700s, focusing on Handel and his "Music for the Royal Fireworks" and his capacity for realizing the common mood.

DE: Handel, Orchestral Music, English Music, Musical Analysis

104

The Art of the Oratorio

One of the greatest writers of the English oratorio, Handel, is featured, with extensive focus on Messiah, Alexander's Feast and his final work, Jephtha.

DE: Handel, Oratorio, English Music, Musical Analysis

In this stage, we also deal with a couple of other technical details. We remove all punctuation, including commas, periods, hyphens, apostrophes, quotation marks, etc., and replace them with spaces. We

also choose to ignore capitalization, so we convert all words to capital letters. This means, for example, that the words

**"Alexander's Feast"**

in document 104 will become

**ALEXANDER S FEAST**

with no quotation marks. There are other ways you could handle this-- for example, decide to preserve capitalization to make it easier to search for capitalized words or not insert spaces in place of punctuation marks, to keep words like "Alexander's" together. There's nothing magical or even ideal about these decisions; as we said, they are DIALOG's way of doing things, but you will begin to see the kinds of effects they have later on searching.

Doing all these things gets us a list which looks like this:

101  
THE  
ORIGINS  
OF  
DON  
GIOVANNI

DISCUSSES  
THE  
HISTORY  
AND  
SOURCES  
MOZART  
USED  
IN  
HIS  
OPERA  
DON  
GIOVANNI

MOZART  
OPERA  
HISTORICAL  
ANALYSIS

102  
HANDEL  
TWO  
GREAT  
OPERAS

PLOT  
SUMMARIES  
TEXTUAL  
ANALYSIS  
OF  
LIBRETTI  
AND  
MUSICAL  
EXPLICATION  
OF  
TWO  
OF  
HANDEL  
S  
OPERAS

GIULIO  
CESARE  
1724  
AND  
HIS  
FIRST  
ITALIAN  
OPERA  
RODRIGO  
1707

HANDEL  
OPERA  
MUSICAL  
ANALYSIS

103  
ENGLISH  
ORCHESTRAL  
MUSIC  
OF  
THE  
EARLY  
18TH  
CENTURY

A  
DISCUSSION  
OF  
THE  
MAJOR  
FEATURES  
IN  
ENGLISH  
MUSIC  
OF  
THE  
MID  
1700S  
FOCUSING  
ON  
HANDEL  
AND  
HIS  
MUSIC  
FOR  
THE  
ROYAL  
FIREWORKS  
AND  
HIS  
CAPACITY  
FOR  
REALIZING  
THE  
COMMON  
MOOD

HANDEL  
ORCHESTRAL  
MUSIC  
ENGLISH  
MUSIC  
MUSICAL  
ANALYSIS

104  
THE  
ART  
OF  
THE  
ORATORIO

ONE  
OF  
THE  
GREATEST  
WRITERS  
OF  
THE  
ENGLISH  
ORATORIO  
HANDEL  
IS  
FEATURED  
WITH  
EXTENSIVE  
FOCUS  
ON  
MESSIAH  
ALEXANDER  
S  
FEAST  
AND  
HIS  
FINAL  
WORK  
JEPHTHA

HANDEL  
ORATORIO  
ENGLISH  
MUSIC  
MUSICAL  
ANALYSIS

**Step 2. Number all the words, including phrases and excluding stop words.**

We now assign a number to each of these words. These numbers will serve as pointers to the documents from which they come; when we search the file for a given word, it will tell us its exact location in the file. So, for example, DISCUSSES will get this code:

101 AB 1

to indicate that it comes from document 101, is in the abstract field, and is the first word in that abstract.

An inverted file is intended to aid in searching, but this aid does not come without some cost. You can already get a sense from looking at the preliminary list above that not only does a great deal of thought go into the creation of an inverted file, so also does a considerable amount of computing time and storage. There are ways to make the storage issue a bit easier, without sacrificing too much access; one of them is the use of a list of words which are either so common or devoid of search potential that they are excluded at this stage and thus not able to be used in searching. Some systems have quite extensive lists

of these **stop words** (sometimes called **noise words**); some have none at all. DIALOG, our exemplar system, uses these 9:

AN AND BY FOR FROM OF THE TO WITH

These words will be excluded from the inverted file, but only **after** the words have been numbered. This will permit more precise searching of multiple-word combinations of words.

The other detail to be dealt with here are the phrases in the descriptor field. DIALOG and other systems use a technique here called **phrase indexing**, the inclusion of complete phrases from the descriptor field (and sometimes other fields as well) in the inverted file as well as the individual words themselves. So the descriptor

#### MUSICAL ANALYSIS

would receive three entries: one for MUSICAL, one for ANALYSIS, and the third for MUSICAL ANALYSIS.

We call this multiple indexing **double posting**. It is particularly useful to be able to search phrases in this fashion as they are more specific search terms than single words. --GW

This is possible in fields such as descriptor because it's clear what the phrases are. In other fields (titles and abstracts), people are really good at telling where phrases begin and end but it's been very difficult to get computers to figure it out. A number of researchers in natural language processing and automatic indexing have been working on this for many years with some success, but such techniques are nowhere near being ready for commercial systems yet. In the meantime, we do the best we can with only **word indexing** of those fields, and make up for it by being more clever in searching. More about this when we discuss free-text searching technique.

When these three steps have been completed, our list now looks like this:

ORIGINS	101	TI	2
DON	101	TI	4
GIOVANNI	101	TI	5
DISCUSSES	101	AB	1
HISTORY	101	AB	3
SOURCES	101	AB	5
MOZART	101	AB	6
USED	101	AB	7
IN	101	AB	8
HIS	101	AB	9
OPERA	101	AB	10
DON	101	AB	11
GIOVANNI	101	AB	12
MOZART	101	DE	1
OPERA	101	DE	2
HISTORICAL	101	DE	3
ANALYSIS	101	DE	4
HISTORICAL ANALYSIS	101	DE	3,4
HANDEL	102	TI	1
TWO	102	TI	2
GREAT	102	TI	3

OPERAS	102	TI	4
PLOT	102	AB	1
SUMMARIES	102	AB	2
TEXTUAL	102	AB	3
ANALYSIS	102	AB	4
LIBRETTI	102	AB	6
MUSICAL	102	AB	8
EXPLICATION	102	AB	9
TWO	102	AB	11
HANDEL	102	AB	13
S	102	AB	14
OPERAS	102	AB	15
GIULIO	102	AB	16
CESARE	102	AB	17
1724	102	AB	18
HIS	102	AB	20
FIRST	102	AB	21
ITALIAN	102	AB	22
OPERA	102	AB	23
RODRIGO	102	AB	24
1707	102	AB	25
HANDEL	102	DE	1
OPERA	102	DE	2
MUSICAL	102	DE	3
ANALYSIS	102	DE	4
MUSICAL ANALYSIS	102	DE	3,4
ENGLISH	103	TI	1
ORCHESTRAL	103	TI	2
MUSIC	103	TI	3
EARLY	103	TI	6
18TH	103	TI	7
CENTURY	103	TI	8
A	103	AB	1
DISCUSSION	103	AB	2
MAJOR	103	AB	5
FEATURES	103	AB	6
IN	103	AB	7
ENGLISH	103	AB	8
MUSIC	103	AB	9
MID	103	AB	12
1700S	103	AB	13
FOCUSING	103	AB	14
ON	103	AB	15
HANDEL	103	AB	16
HIS	103	AB	18
MUSIC	103	AB	19
ROYAL	103	AB	22
FIREWORKS	103	AB	23
HIS	103	AB	25
CAPACITY	103	AB	26
REALIZING	103	AB	28
COMMON	103	AB	30
MOOD	103	AB	31
HANDEL	103	DE	1

ORCHESTRAL	103 DE	2
MUSIC	103 DE	3
ENGLISH	103 DE	4
MUSIC	103 DE	5
MUSICAL	103 DE	6
ANALYSIS	103 DE	7
ORCHESTRAL MUSIC	103 DE	2,3
ENGLISH MUSIC	103 DE	4,5
MUSICAL ANALYSIS	103 DE	6,7
ART	104 TI	2
ORATORIO	104 TI	5
ONE	104 AB	1
GREATEST	104 AB	4
WRITERS	104 AB	5
ENGLISH	104 AB	8
ORATORIO	104 AB	9
HANDEL	104 AB	10
IS	104 AB	11
FEATURED	104 AB	12
EXTENSIVE	104 AB	14
FOCUS	104 AB	15
ON	104 AB	16
MESSIAH	104 AB	17
ALEXANDER	104 AB	18
S	104 AB	19
FEAST	104 AB	20
HIS	104 AB	22
FINAL	104 AB	23
WORK	104 AB	24
JEPHTHA	104 AB	25
HANDEL	104 DE	1
ORATORIO	104 DE	2
ENGLISH	104 DE	3
MUSIC	104 DE	4
MUSICAL	104 DE	5
ANALYSIS	104 DE	6
ENGLISH MUSIC	104 DE	3,4
MUSICAL ANALYSIS	104 DE	5,6

### Step 3. Alphabetize the list.

This is really the easiest part, but alphabetization is what makes an inverted file so darn useful. Put these entries in alphabetical order (actually, usually in the order of the ASCII character set used by computers--numbers first, followed by letters), and you have an inverted file.

1707	102 AB	25
1724	102 AB	18
1700S	103 AB	13
18TH	103 TI	7
A	103 AB	1
ALEXANDER	104 AB	18
ANALYSIS	101 DE	4
ANALYSIS	102 AB	4



ANALYSIS	102 DE	4
ANALYSIS	103 DE	7
ANALYSIS	104 DE	6
ART	104 TI	2
CAPACITY	103 AB	26
CENTURY	103 TI	8
CESARE	102 AB	17
COMMON	103 AB	30
DISCUSSES	101 AB	1
DISCUSSION	103 AB	2
DON	101 TI	4
DON	101 AB	11
EARLY	103 TI	6
ENGLISH	103 TI	1
ENGLISH	103 AB	8
ENGLISH	103 DE	4
ENGLISH	104 AB	8
ENGLISH	104 DE	3
ENGLISH MUSIC	103 DE	4,5
ENGLISH MUSIC	104 DE	3,4
EXPLICATION	102 AB	9
EXTENSIVE	104 AB	14
FEAST	104 AB	20
FEATURED	104 AB	12
FEATURES	103 AB	6
FINAL	104 AB	23
FIREWORKS	103 AB	23
FIRST	102 AB	21
FOCUS	104 AB	15
FOCUSING	103 AB	14
GIOVANNI	101 TI	5
GIOVANNI	101 AB	12
GIULIO	102 AB	16
GREAT	102 TI	3
GREATEST	104 AB	4
HANDEL	102 TI	1
HANDEL	102 AB	13
HANDEL	102 DE	1
HANDEL	103 AB	16
HANDEL	103 DE	1
HANDEL	104 AB	10
HANDEL	104 DE	1
HIS	101 AB	9
HIS	102 AB	20
HIS	103 AB	18
HIS	103 AB	25
HIS	104 AB	22
HISTORICAL	101 DE	3
HISTORICAL ANALYSIS	101 DE	3,4
HISTORY	101 AB	3
IN	101 AB	8
IN	103 AB	7
IS	104 AB	11
ITALIAN	102 AB	22

JEPHTHA	104 AB	25
LIBRETTI	102 AB	6
MAJOR	103 AB	5
MESSIAH	104 AB	17
MID	103 AB	12
MOOD	103 AB	31
MOZART	101 AB	6
MOZART	101 DE	1
MUSIC	103 TI	3
MUSIC	103 AB	9
MUSIC	103 AB	19
MUSIC	103 DE	3
MUSIC	103 DE	5
MUSIC	104 DE	4
MUSICAL	102 AB	8
MUSICAL	102 DE	3
MUSICAL	103 DE	6
MUSICAL	104 DE	5
MUSICAL ANALYSIS	102 DE	3,4
MUSICAL ANALYSIS	103 DE	6,7
MUSICAL ANALYSIS	104 DE	5,6
ON	103 AB	15
ON	104 AB	16
ONE	104 AB	1
OPERA	101 AB	10
OPERA	101 DE	2
OPERA	102 AB	23
OPERA	102 DE	2
OPERAS	102 TI	4
OPERAS	102 AB	15
ORATORIO	104 TI	5
ORATORIO	104 AB	9
ORATORIO	104 DE	2
ORCHESTRAL	103 TI	2
ORCHESTRAL	103 DE	2
ORCHESTRAL MUSIC	103 DE	2,3
ORIGINS	101 TI	2
PLOT	102 AB	1
REALIZING	103 AB	28
RODRIGO	102 AB	24
ROYAL	103 AB	22
S	102 AB	14
S	104 AB	19
SOURCES	101 AB	5
SUMMARIES	102 AB	2
TEXTUAL	102 AB	3
TWO	102 TI	2
TWO	102 AB	11
USED	101 AB	7
WORK	104 AB	24
WRITERS	104 AB	5

And that's how an inverted file is created. Almost all information retrieval systems and search engines you use will have something like

this (if not precisely this structure or format) underlying it. New documents are processed in the same way and added to the file; if documents are removed from the database, their pointers are excised from the file and life goes on.

### Why Inverted Files Aren't Panaceas

A couple of things to think about at this stage: looking through the final inverted file, you notice words like WORK, ROYAL, ENGLISH, MESSIAH, and MOOD. If you know that the database covers only music, some of these have special meanings. However, in a file with, for example, newspaper or magazine articles, these words might have other connotations. We also have here four documents about operas and oratorios, both kinds of vocal compositions. None of them, though, contain the words VOICE or VOCAL. Hmmm.

Imagine also a religion database which nonetheless has a document about the oratorio Messiah. Searching in that database for the word MESSIAH will get you that document, but also probably a great deal more, so you'll have to refine your search with other words and techniques.

The computer can only say yes, these documents have those words in them and no, those don't, but not much more than that. The system alone is not able to take advantage of the context of the way words are used in a document, which people can do very easily. It is simply unable to identify relationship among terms.

This is one of the most important things for you to know about information systems in general. The only things that go into inverted files are **words**--either directly from the texts of documents or from some indexing added later (called "controlled vocabulary"). These words are, at best, clues to the actual content and subject matter of these documents, but because of the nature of language, they are often not perfect (or, for that matter, even very good) indicators of that content.

If a document has the word ORATORIO in the text, it's probably about oratorios--that's a pretty specific word. If ORATORIO is in the title, your odds go up. What about the word pitcher?

Here's the bottom line, and what pretty much the whole profession of library and information science is about: **we want to look for concepts, but we are forced to search for words**. The better you understand that, and the better you are able to cope with it, the better you'll be at the whole searching thing.

There. Now that we've revealed the secret of the universe, all the rest should be pretty easy.

## Structure

Several times in the preceding discussion, we have talked about the structure of the records to be included in the database. This is one of the senses of "structure" that we'll use here--in this section we'll review what both senses are, give some examples of each, talk about how and why they arose, how they can be used in searching, now and in the future, and finally elaborate further on overhead issues in database building.

### Two Kinds of Structure

In the inverted file construction example above, we used a crude kind of document in a toy database. That document had only a very few pieces of information. In most bibliographic retrieval systems, the records have much more information in them and contain more kinds of information. We might say these records have more elaborate (or at least bigger) **record structures**.

Here, for example, is a record from the *ERIC* database

```

AN  EJ355024    TM511910
TI  An Experimental, Exploratory Study of Causes of Bias
    in Test Items
AU  Scheuneman, Janice Dowd
JO  Journal of Educational Measurement, v24 n2 p97-118 Sum 1987
AV  Available from: UMI
LA  Language: English
DT  Document Type: JOURNAL ARTICLE (080); RESEARCH REPORT (143)
JA  Journal Announcement: CIJSEP87
AB  This study evaluated 16 hypotheses concerning possible sources
of bias in test items on the Graduate Record Examination General
Test. Ten of the hypotheses showed interactions between group
membership and the item performance of Black and White examinees.
(Author/LMO)
DE  Descriptors: *Blacks; *College Entrance Examinations; Higher
Education; Hypothesis Testing; *Racial Differences; Sex Differences;
Statistical Bias; *Test Bias; *Test Items; *Whites
ID  Identifiers: *Graduate Record Examinations; Log Linear
Analysis

```

Figure 5.1 ERIC Database Record

The two-letter codes point to the different fields of the record. A field is an individual piece of information about a document, and the collection of these fields about the same document is called a record. The fields shown above are described here:

**AN - accession number;** a number assigned by the database producer as a document is entered into the database. This number uniquely identifies each record in the file. Documents in the ERIC database have two accession numbers: one assigned by the individual ERIC clearinghouse where the document was produced (here TM 511 910), and one by the overall ERIC system (here EJ 355 024).

**TI - title;** the title of the original document.

**AU - author;** the author of the original document. There may be more than one author; if so, all may or may not be listed. An agency or organization may also be credited with authorship. This is referred to as a corporate author.

**JO - journal name and citation;** the name of the journal where the original document appeared (if it is indeed a journal article; if not, identifying information about the original source is given). In addition, the journal's volume, number, pages, and year of publication are given.

**AV - availability;** where the document may be obtained, in addition to the source journal. In this case, the document is available from University Microfilms International (UMI).

**LA - language;** the language in which the original document is written.

**DT - document type;** ERIC assigns a code to each document to describe its "type": journal article, guidebook, manual, dissertation, report, etc. Other databases have similar information, although the specific types involved will differ.

**JA - journal announcement;** all documents in the ERIC database are also listed in the two ERIC manual indexes: CIJE (Current Index to Journals in Education) for journal articles and RIE (Resources in Education) for all other documents. This field shows that this document appeared in CIJE in September of 1987.

**AB - abstract;** a brief summary of the document (typically a paragraph), which may either have been written by the original author, or by the indexers. This abstract was written by the author, then later edited--the initials are of the indexer.

**DE - descriptors;** index terms assigned, generally from a predetermined list, by a professional indexer to represent this document and assist searchers in looking for it. This list is known as a "controlled vocabulary", and we will discuss this further later. The descriptors in this record are taken from the *Thesaurus of ERIC Descriptors*. The starred descriptors are referred to as major descriptors, which have been identified by the indexer as the terms which best describe what the document is about and are the only descriptors which appear in the print version of the file.

**ID - identifiers;** terms assigned by the indexer--similar in form to descriptors, but these are freely assigned and are not from a predetermined list. Often, identifiers are terms so new in an area that they are not yet widely used or known and have not yet been added to the accepted vocabularies. In ERIC this field is also used for proper names (i.e., names of places, people, projects and programs).

Different databases have different record structures, as we have seen: different fields, different codes for the same fields, and different orderings of the fields. But this ERIC record is a good example of the type of bibliographic record stored in an online database.

This record will act as a surrogate for the real document, an article which appeared in the *Journal of Educational Measurement* in 1987.

As an aside, you might also be familiar with another kind of record in an information system: here's the MARC (MAchine-Readable Cataloging) record for the first edition of this book.

```

001 28257554
003 OCoLC
005 19950309081456.0
008 930520s1993 coua b 001 0 eng pam a
010 93004955
020 1563080710 (cloth) :
020 1563081571 (paper) :
040 DLC|cDLC
043 n-us---
050 00 Z699.35.O55|bW35 1993
082 00 025.5/24|220
100 1 Walker, Geraldene
245 10 Online retrieval :|ba dialogue of theory and practice /
|cGeraldene Walker, Joseph Janes
260 Englewood, Colo. :|bLibraries Unlimited,|c1993
300 xi, 221 p. :|bill. ;|c28 cm
440 0 Database searching series

```

504	Includes bibliographical references (p. 2-8) and index
650 0	Online bibliographic searching zUnited States
650 0	DIALOG (Information retrieval system)
700 10	Janes, Joseph

Figure 5.2 MARC Record

The catalog of a library which owns this book will include a MARC record very like this, and will create an inverted file much as we've seen to allow people to search for it and find it based on words in the title, one or both of our names, or one of the two subject headings. You'll notice that although there are many similarities between the two records and the systems which search them, there are some important differences. The MARC record, representing a 200+ page book, has only those two subject headings and no abstract to represent the subject matter of the original document; the ERIC record for a 20 page journal article has a paragraph-length abstract and a dozen subject indicators. Something to bear in mind--searchers will typically have fewer access points to retrieve book records than journal article records because of the nature of the records used to represent them.

The other use of the word "structure" has a somewhat different meaning. Over the last several years, there has been a greater attention and awareness of **structured text**, largely fueled by its use in the publishing industry and the Internet.

Using structured text is a way of representing the internal structure of documents (acts of a play, chapters of a book, stanzas of a poem, captions of photographs, etc.) as well as meta-information such as version, edition, authorship, date, and so on. As you can imagine, this can be an enormous aid in textual analysis and scholarship, printing, description, and, of course, searching.

There are two common schemes used in creating structured text: **HTML (HyperText Markup Language)** and **SGML (Standard Generalized Markup Language)**. You note that they are both called "markup languages", and we often refer to the creation or conversion of structured documents as "marking them up". Here are examples of each of these.

This is an HTML document. HTML is the language used to create documents which can be served and viewed over the World-Wide Web on the Internet. It is the "About the Library" page from the Internet Public Library:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
<head>
<title>IPL About the Library</title>
</head>
<body bgcolor="#FFFFFF">

<h3><a href="/"></a>the Internet Public Library</h3>

<h1>About the Internet Public Library</h1>

<p> <a href="iplfaq.html">Frequently Asked Questions about the IPL</a>
(updated 7 June 96)</p>
<p> <strong>NEW!</strong> A more user-friendly <a
href="stats.html">summary of our statistics</a></p>

<ul>
<li><a href="message.html">A welcome message from the Director</a>
<li><a href="bios.html">Meet the IPL Staff</a>
```

```

<li><a href="statement.html">Our Statement of Principles</a>
<li><a href="newmission.html">Our Mission Statement and Goals</a>
<li><a href="update.html">Our listserv</a>
<li><a href="releases.html">Press Releases</a>
<li><a href="awards.html">Awards Received by the IPL</a>
<li><a href="ifpol.html">Policy regarding requests to reconsider
resources</a>
<li><a href="circpol.html">Policy regarding release of access log
information</a>
<li><a href="repropol.html">Policy regarding reproduction of our pages
and images</a>
<li><a href="telecom.html">Position on S.652 HR.1555</a>, the
Telecommunications Reform Bill.
<li>Thoughts on <a href="bannedbooks.html">Banned Books</a> Week 1996.
</ul>

<p>

<p>The Library is hosted by the <A href="/cgi-
bin/redirect?http://www.si.umich.edu/">School of Information</A> at the
<A href="/cgi-bin/redirect?http://www.umich.edu/">University of
Michigan</A>.

<p><STRONG>You may also want to <A HREF="mailto:ipl@ipl.org">Send Us
Feedback</A> | View the IPL <A HREF="/cgi-bin/stats.pages.pl">Access
Statistics</A>.</STRONG></p>

<p><strong>Return to <a href="/">the IPL Main Lobby</a>.</strong>
<hr>

<address>the Internet Public Library - = - http://www.ipl.org/ - = -
ipl@ipl.org</address>
Last Updated Oct 16, 1996

</body>
</html>

```

Figure 5.3 Internet Public Library HTML Page

Here's what this page would look like when viewed over the Web:

<insert Figure 5.4>

Figure 5.4 Internet Public Library Page (as viewed)

If you compare the two, you can see that the text of the document is intermingled with lots of things in angle brackets, like <title> and <h1> and <li> and so on. These are HTML **tags**, and they define the structure of the document and how it will be displayed by a Web browser. The <h1> tag is a first-level header, and so its text is displayed as large type. The <ul> tag defines the beginning of an unordered list, and each <li> is a list item, which appear as bullets in the page.

You can imagine ways in which that would be quite useful for searching. The <title> tag is an obvious one, but being able to search in important headers or addresses could also come in handy. We'll talk more about Web searching in the next chapter.

This next one is an SGML document; SGML has a more complete set of tags and is used by many publishing companies to assist in formatting and printing. We are indebted to the Humanities Text Initiative of the

University of Michigan for this document, which is the descriptive information about and opening lines of a book of Longfellow poetry.

```

<TEI.2 ID="BAD8947">
  <TEIHEADER>
    <FILEDESC>
      <TITLESTMT>
        <TITLE TYPE="245">Courtship of Miles Standish : and other
          poems / Henry Wadsworth Longfellow [electronic text]</TITLE>
        <AUTHOR>Longfellow, Henry Wadsworth, 1807-1882</AUTHOR>
      <RESPSTMT>
        <NAME>Nigel Kerr, Mark Holman, Anne Noakes, William
          M. Wines, University of Michigan Humanities Text
          Initiative</NAME> <RESP>creation of machine-readable
          edition</RESP>
      </RESPSTMT>
      <RESPSTMT>
        <NAME>Jason J. Chu, University of Michigan Humanities Text
          Initiative</NAME> <RESP>correction of machine-readable
          edition</RESP>
      </RESPSTMT>
      <RESPSTMT>
        <NAME>Kevin Butterfield, University of Michigan</NAME>
        <RESP>creation of AACR2-conformant header</RESP>
      </RESPSTMT>
    </TITLESTMT>
    <EXTENT>ca. 172 kb.</EXTENT>
    <PUBLICATIONSTMT>
      <PUBLISHER>University of Michigan Humanities Text
        Initiative</PUBLISHER> <PUBPLACE>Ann Arbor, Mich.</PUBPLACE>
      <IDNO>LongfCourt</IDNO>
    <AVAILABILITY>
      <P>This work is the property of the University of Michigan.
        It may be copied freely by individuals for personal use,
        research, and teaching (including distribution to classes)
        as long as this statement of availability is included in
        the text. It may be linked to freely in Internet editions
        of all kinds, including for-profit works.</P>
      <P>Publishers, libraries, and other information providers
        intere</P> <P>Publishers, libraries, and other information
        providers interested in providing this text in a
        commercial or non-profit product or from an information
        server must contact the University of Michigan Press for
        licensing and cost information.</P> <P>Scholars interested
        in changing or adding to these texts by, for example,
        creating a new edition of the text (electronically or in
        print) with substantive editorial changes, may do so with
        the permission of the University of Michigan Press. This
        is the case whether the new publication will be made
        available at a co st or free of charge.</P> <P>Accessible
        at http://www.hti.umich.edu/english/amverse/</P>
    </AVAILABILITY>
    <DATE>1996</DATE>
  </PUBLICATIONSTMT>
  <SOURCEDESC>
  <BIBLFULL>
    <TITLESTMT>
      <TITLE>The Courtship of Miles Standish, and other
poems</TITLE>
      <AUTHOR>Henry Wadsworth Longfellow</AUTHOR>
    </TITLESTMT>
    <PUBLICATIONSTMT>
      <PUBLISHER>Ticknor and Fields</PUBLISHER>
      <PUBPLACE>Boston</PUBPLACE>

```



```

<DATE>1859</DATE>
</PUBLICATIONSTMT>
<NOTESSTMT>
  <NOTE><P>Call number: 828 L853c</P></NOTE>
</NOTESSTMT>
</BIBLFULL>
</SOURCEDESC>
</FILEDESC>
<ENCODINGDESC>
  <EDITORIALDECL>
    <P>All poems, line groups, and lines are represented.
      Indentation has not been preserved.</P>
  </EDITORIALDECL>
</ENCODINGDESC>
</TEIHEADER>
<TEXT>
  <FRONT>
    <TITLEPAGE>
      <DOCTITLE>
        <TITLEPART TYPE="main">THE COURTSHIP OF MILES STANDISH,
          <LB>AND <LB>OTHER POEMS.</TITLEPART>
      </DOCTITLE>
      <BYLINE>BY <DOCAUTHOR>HENRY WADSWORTH LONGFELLOW.</DOCAUTHOR>
      </BYLINE>
      <DOCIMPRINT>
        <PUBPLACE>BOSTON:</PUBPLACE> <PUBLISHER> TICKNOR AND
          FIELDS.</PUBLISHER> <DOCDATE>M DCCC LIX.</DOCDATE> <PB
          ID="P1" N=" [verso]">Entered according to Act of Congress, in
          the year 1858, by <LB>HENRY WADSWORTH LONGFELLOW, <LB>in the
          Clerk's Office of the District Court of the District of
          Massachusetts. <LB>CAMBRIDGE: <LB> ELECTROTYPED AND PRINTED
          BY <NAME>METCALF AND COMPANY.</NAME>
        </DOCIMPRINT>
      </TITLEPAGE>
      <PB ID="P2" N=" [iii]">
    </FRONT>
    <BODY>
      <PB ID="P4" N=" [5]">
      <DIV0 ID="DIV0.2" N="1" TYPE="poem">
      <HEAD>THE COURTSHIP OF MILES STANDISH</HEAD>
      <PB ID="P5" N=" [6]">
      <PB ID="P6" N=" [7]">
      <DIV1 ID="DIV1.3" TYPE="section">
        <HEAD>I. <LB>MILES STANDISH.</HEAD>
        <LG ID="LG1" TYPE="stanza">
          <L ID="L1">IN the Old Colony days, in Plymouth the land of
            the Pilgrims,</L>
          <L ID="L2">To and fro in a room of his simple and
            primitive dwelling,</L>
          <L ID="L3">Clad in doublet and hose and boots of Cordovan
            leather,</L>
          <L ID="L4">Strode, with a martial air, Miles Standish the
            Puritan Captain.</L>
          <L ID="L5">Buried in thought he seemed, with his hands
            behind him, and pausing</L>
          <L ID="L6">Ever and anon to behold his glittering weapons
            of warfare,</L>
          <PB ID="P7" N="8">
          <L ID="L7">Hanging in shining array along the walls of the
            chamber, &mdash;</L>
          <L ID="L8">Cutlass and corslet of steel, and his trusty
            sword of Damascus,</L>
          <L ID="L9">Curved at the point and inscribed with its
            mystical Arabic sentence,</L>

```

```

<L ID="L10">While underneath, in a corner, were
  fowling-piece, musket, and matchlock.</L>
<L ID="L11">Short of stature he was, but strongly built
  and athletic, </L>
<L ID="L12">Broad in the shoulders, deep-chested, with
  muscles and sinews of iron;</L>
<L ID="L13">Brown as a nut was his face, but his russet
  beard was already</L>
<L ID="L14">Flaked with patches of snow, as hedges
  sometimes in November.</L>
<L ID="L15">Near him was seated John Alden, his friend,
  and household companion,</L>
<L ID="L16">Writing with diligent speed at a table of pine
  by the window;</L>
<PB ID="P8" N="9">
<L ID="L17">Fair-haired, azure-eyed, with delicate Saxon
  complexion,</L>
<L ID="L18">Having the dew of his youth, and the beauty
  thereof, as the captives</L>
<L ID="L19">Whom Saint Gregory saw, and exclaimed, "Not
  Angles but Angels."</L>
<L ID="L20">Youngest of all was he of the men who came in
  the May Flower.</L>
</LG>

```

Figure 5.5 SGML Document

You see even more indications of the structure of the document in the SGML document, giving the lines of the poem, where the pages begin and end, and so on. But you also see a large amount of information about the work itself (publisher, date, author, title, imprint) and this electronic version (who created the SGML version, the size of the file, its availability, etc.). This is often called **meta-information**, and, since it is the same file as the work itself, can also be used in searching in rather sophisticated ways.

### Searching and Structure

The use of the structure of bibliographic records in searching is pretty basic. It permits searches to be restricted to a particular field, so you don't have to search for all occurrences of the word "Bush" if you only want documents written by someone named Bush, for example. This kind of searching is often used as an auxiliary method to searching by content. We'll see more of this in a later chapter, but consider a search for documents about the Challenger disaster--if you wanted only things written right after it happened, you could construct a search for things with the word CHALLENGER and then restrict the result to those written in 1986.

Some of the benefits of structured text in searching should be clear. It permits searching by field, but in a much more detailed way than in the bibliographic example--not just words in fields like author, title, abstract, etc., but also where they occur, in which chapter or heading or subheading or table or in a caption, and so on. It also allows searching in the meta-information such as version or edition.

It's true that in this situation you've got the full text available to be searched, which you don't have with just a bibliographic record. That may seem like an inherent advantage, but it's not necessarily the case. It may well be that the addition of structure helps the full-text searching problem, but this is as yet an emerging area of investigation. As yet, searching using structured documents is pretty crude, and largely limited to the HTML/Internet domain, where some search engines allow you to restrict searching to words in the <title> or <h1> tags,

for example. And, of course, it goes without saying that this kind of searching depends on the fact that the right tags have to be assigned to the appropriate parts of the document at the input stage and the search engine has to be able to take advantage of them.

### **Overhead Issues**

In both cases, a lot of work is involved in implementation. Indexing or marking up documents takes a great deal of time and intellectual effort, and the more you want to be able to use this structure as an aid to searching these documents, the more work it will be.

If you want to be able to search within chapters or captions, you have to add HTML or SGML tags. For short texts such as poems, this is not an enormous burden, but imagine the work involved in novels, technical documents incorporating formulas, graphs, diagrams and pictures, or the works of Shakespeare. Since it's difficult to predict exactly what kinds of searching people are going to want to do on these, you probably will have to include a great deal of structure to allow for a variety of possibilities. This can be incredibly tedious and very costly. More documents are being created with SGML in "native" form, but there are a lot of pre-SGML documents out there, and their conversion is a daunting prospect indeed.

In a bibliographic database, if you just want to be able to search for words anywhere, your inverted file doesn't have to be all that complicated--just list the words and what documents they're in and you're all set.

#### **ENGLISH 104**

If, though, you want to be able to restrict your searching by field (only look for ENGLISH in the abstract field, for example), you've got to include the field indicator in your pointer.

#### **ENGLISH 104 AB**

And to be able to search for multiple-word phrases, you have to be able to find those words near each other, so you must include position within the field in the inverted field entry. To be able to find the phrase ENGLISH ORATORIO, you have to know if they ever occur next to each other, which they do:

#### **ENGLISH 104 AB 8**

#### **ORATORIO 104 AB 9**

All of this falls into the general category of overhead--the more you want to be able to do in searching a database, the more preparation and processing will be needed. We'll see this several times in the discussions to come, and you should be thinking about this for all kinds of features of information systems.