Alternatives to the BLOB Queries

rev. 6/9/2006 for Voyager Version 6.1
http://www.carli.illinois.edu/I-Share/secure/sql/resources/alttoblob.pdf

The BLOB functions in Voyager allow you to retrieve any field or subfield from a MARC record. They are very powerful and very useful. But, depending on the size of your database, a query using the BLOB functions may run for hours. While such a query is running, a network glitch might disconnect Microsoft Access from the Voyager server so that you will never get results, and you'll have to restart it.

This document will look at some alternatives to the BLOB queries. If you can use them, you will get faster results from your queries, even on smaller databases. If you can't, a separate document, "A Strategy for Using BLOB Functions on Large Databases," is available at http://www.carli.illinois.edu/I-Share/secure/sql/resources/bloblargedb.pdf

To get commonly used fields from bibliographic records

Is the field that you want available in the BIB_TEXT table? There's a document on the CARLI I-Share site that lists the MARC fields and subfields that are in each field in BIB_TEXT.
http://www.carli.illinois.edu/I-Share/secure/sql/resources/marcfields.pdf

To get fixed field data from bibliographic records

If you need information from the fixed fields of bibliographic records, consider using the MARC*_VW tables. There's a different table for each of the MARC bib formats. Here's a list of them, along with the Voyager field name, the OCLC display name, and the locations in the MARC 008 field. Refer to the OCLC or LC MARC documentation for a list of the values in these fields.

Use MARCBOOK_VW for monographs with record types a, h, and t. It provides these fields:

audience	Audn	008 byte 22
biography	Biog	008 byte 34
conferencepub	Conf	008 byte 29
governmentpub	GPub	008 byte 28
itemform	Form	008 byte 23
literaryform	LitF	008 byte 33

Use MARCCOMPUTER_VW for monographs with record type m. It provides these fields:

audience	Audn	008 byte 22
filetype	File	008 byte 26
governmentpub	GPub	008 byte 28

Use MARCMAP_VW for monographs with record types e and f. It provides these fields:

cartographictype	CrTp	008 byte 25
governmentpub	GPub	008 byte 28
indexed	Indx	008 byte 31
projection	Proj	008 bytes 22-23

Use MARCMUSIC_VW for monographs with record types c, d, i, and j. It provides these fields:

audience	Audn	008 byte 22
compositionform	Comp	008 bytes 18-19
itemform	Form	008 byte 23
musicformat	FMus	008 byte 20

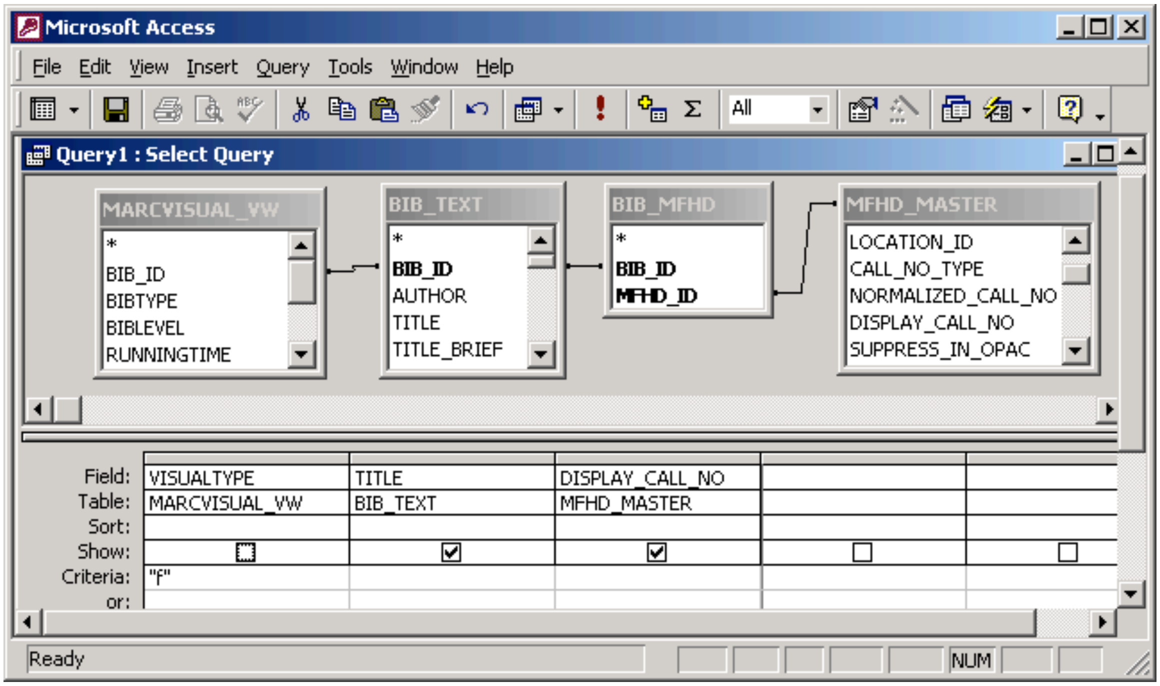
Use MARCSERIAL_VW for bib levels b and s. It provides these fields:

conferencepub	Conf	008 byte 29
entirenature	EntW	008 byte 24
frequency	Freq	008 byte 18
governmentpub	GPub	008 byte 28
itemform	Form	008 byte 23
originalform	Orig	008 byte 22
regularity	Regl	008 byte 19
type	SrTp	008 byte 21

Use MARCVISUAL_VW for monographs with record types g, k, n, o, and r. It provides these fields:

audience	Audn	008 byte 22
governmentpub	GPub	008 byte 28
runningtime	Time	008 bytes 18-20
technique	Tech	008 byte 34
visualtype	TMat	008 byte 33

Example: I want a list of my filmstrips, along with their titles and call numbers:



Field:	VISUALTYPE	TITLE	DISPLAY_CALL_NO		
Table:	MARCVISUAL_VW	BIB_TEXT	MFHD_MASTER		
Sort:					
Show:	□	☑	☑	□	□
Criteria:	"f"				
or:					

```
SELECT BIB_TEXT.TITLE, MFHD_MASTER.DISPLAY_CALL_NO
FROM ((MARCVISUAL_VW INNER JOIN BIB_TEXT
ON MARCVISUAL_VW.BIB_ID = BIB_TEXT.BIB_ID) INNER JOIN BIB_MFHD
ON BIB_TEXT.BIB_ID = BIB_MFHD.BIB_ID) INNER JOIN MFHD_MASTER
ON BIB_MFHD.MFHD_ID = MFHD_MASTER.MFHD_ID
WHERE (((MARCVISUAL_VW.VISUALTYPE)="f"));
```

To get URLs for MFHDs, bibliographic records, and electronic reserve items

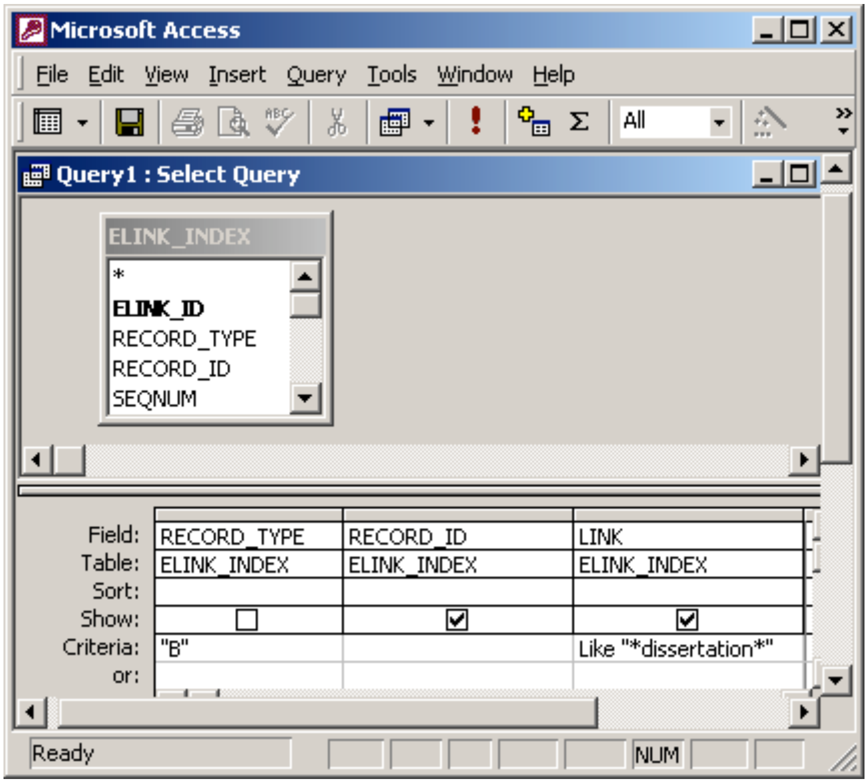
ELINK_INDEX is a very handy place to find URLs from various types of records.

Record_type tells you the kind of record in which the URL is located: B for Bibliographic, E for Electronic item, and M for MFHD.

The record_id is either a bib_id, an eitem_id, or a mfhd_id, depending on the value of record_type.

Most of the time, the link field is 856$u, and the link_text field is subfields $z and $3. The exceptions, which are rather complicated, are in Appendix 1.

Example: I need to find the bib records that point to my electronic dissertations. The URLs for these records include the word "dissertation".



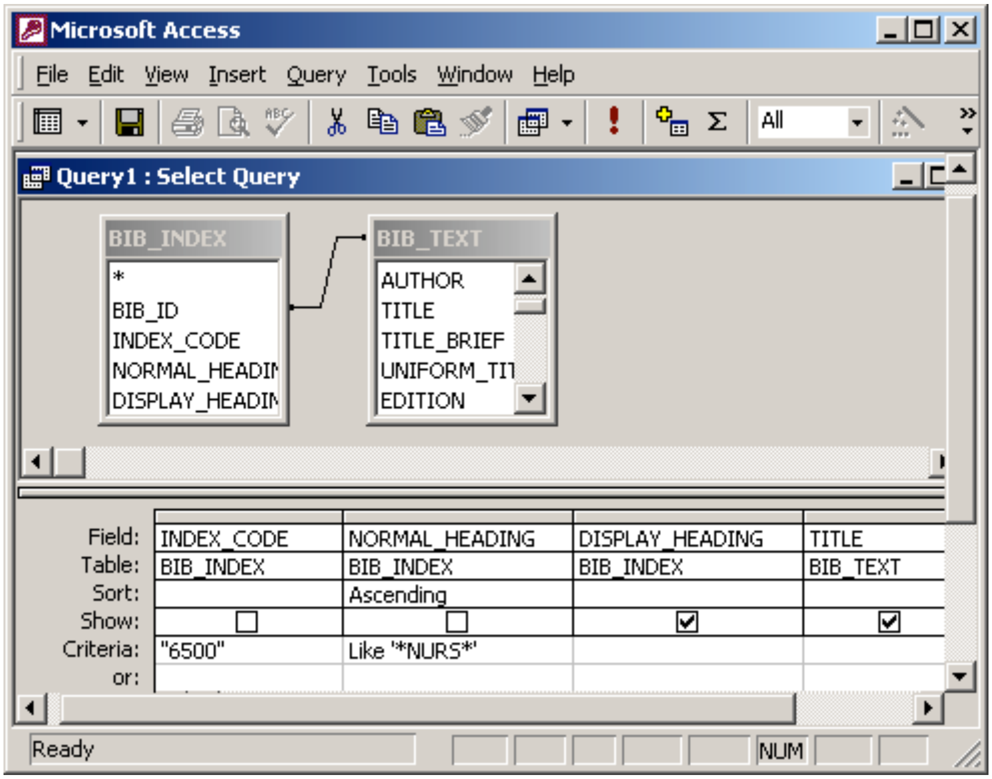
Field:	RECORD_TYPE	RECORD_ID	LINK
Table:	ELINK_INDEX	ELINK_INDEX	ELINK_INDEX
Sort:			
Show:	☐	☑	☑
Criteria:	"B"		Like "*dissertation*"
or:			

```
SELECT ELINK_INDEX.RECORD_ID, ELINK_INDEX.LINK
FROM ELINK_INDEX
WHERE (((ELINK_INDEX.RECORD_TYPE)="B")
AND ((ELINK_INDEX.LINK) Like "*dissertation*"));
```

To get data from fields in bibliographic records that have left-anchored indexes in Voyager

If the data that you want can be searched using a left-anchored index, consider using the BIB_INDEX table. To use BIB_INDEX, you need to know what value of INDEX_CODE to use. You can look it up in the SEARCHPARM table or in Appendix 2 to this document. The NORMAL_HEADING is a version of the search string that is normalized for capitalization, punctuation, subfield coding, and other formatting. The DISPLAY_HEADING has subfield coding removed but is otherwise pretty much what you find in the bibliographic record.

Example: I need a list of all our books on nursing. We use only LC subject headings. I want LC subjects that contain the string "nurs" so I get both "nurses" and "nursing".



Field:	INDEX_CODE	NORMAL_HEADING	DISPLAY_HEADING	TITLE
Table:	BIB_INDEX	BIB_INDEX	BIB_INDEX	BIB_TEXT
Sort:		Ascending		
Show:	☐	☐	☑	☑
Criteria:	"6500"	Like '*NURS*'		
or:				

```
SELECT BIB_INDEX.DISPLAY_HEADING, BIB_TEXT.TITLE
FROM BIB_INDEX INNER JOIN BIB_TEXT ON BIB_INDEX.BIB_ID = BIB_TEXT.BIB_ID
WHERE (((BIB_INDEX.INDEX_CODE)="6500")
AND ((BIB_INDEX.NORMAL_HEADING) Like '*NURS*'))
ORDER BY BIB_INDEX.NORMAL_HEADING;
```

Notice that I sorted this query by NORMAL_HEADING. This makes it easy for me to find and delete the false hits that I got by searching for "*NURS*" – Things like "NURSERY RHYMES".

Appendix 1: What’s in the LINK and LINK_TEXT fields of ELINK_INDEX?

The values in the LINK and LINK_TEXT fields in the ELINK_INDEX table are derived from the 856 field of the corresponding bib, MFHD, or authority.

The value of the LINK field:

```
If there is a $u
then    if $u has a valid prefix (e.g. http, telnet, ftp, file, etc)
        then LINK = $u
        else LINK = blank
else    if there is a $g
        then LINK = $g
        else    if $d and $f and $o are all present
                then    if there is a $2
                        then LINK = blank
                        else    if $o = 'dos'
                                then LINK = $d, a backslash character, $f
                                else LINK = $d, a slash character, $f
          else LINK = blank
```

The value of the LINK_TEXT field:
There are 4 cases, depending on whether $3 and $z are present.

```
1.      If there is a $z and no $3
        then LINK_TEXT = $z

2.      If there is a $z and $3
        then    if there is $u or $g
                then LINK_TEXT = $3 followed by $z
                else LINK_TEXT = $z

3.      If there is a $3 and no $z
        then    if there is $u or $g
                then LINK_TEXT = $3
        else    if $d and $f and $o are all present
                then LINK_TEXT = $3
                else    if $d and $f are both present
                        then LINK_TEXT = blank
                        else LINK_TEXT = all subfields of the 856 with subfield codes

4.      If neither $3 nor $z is present and there’s something in the LINK field
        then LINK_TEXT = LINK
        else LINK_TEXT = all subfields of the 856 with subfield codes
```

Appendix 2: The INDEX_CODE field from the BIB_INDEX table and the MARC fields that are indexed with it.

Index Codes	MARC Tags
008D	008/Date1
008L	008/Lang
008P	008/Ctry
010A	010 a
010Z	010 z
019A	019 a
020A	020 a
020N	020 a
020R	020 z
020Z	020 z
ISB3	020 a special rules for 10-digit ISBNs
022A	022 a
022Y	022 y
022Z	022 z
024A	024 a
024I	024 a when Ind13
024Y	024 z when Ind13
024Z	024 z
027A	027 a
027Z	027 z
028A	028 a
028B	028 b
030A	030 a
030Z	030 z
0350	035 a
0359	035 9
035A	035 a
035Z	035 z
074A	074 a
074Z	074 z
100H	100 abcdefgklnpqtu
110H	110 abcdefgklnptu
111H	111 abcdefgklnpqtu
1300	130 adfghklmnoprst
1301	130 alnps
1302	130 p
2100	210 ab
2110	211 ab
2120	212 a
2140	214 a
2220	222 ab
2400	240 adfghklmnoprs
2401	240 alnps
2402	240 p
2450	245 abfghknps
2451	245 ab
2452	245 abh
2460	246 abfghnp
2470	247 abfghnpx
260D	260 d
4000	400 t
400H	400 abcdefgklnptuv
4100	410 t
410H	410 abcdefgklnptuv
4110	411 t
411H	411 abcdefgklnpqtuv
4400	440 anpvx
440A	440 a
505T	505 t
600H	600 abcdfgklmnopqrtvxyz
600T	600 t
610H	610 abcdfgklmnoprstvxyz
611H	611 abcdefgklnpqstvxyz
630H	630 adfgklmnoprsvxyz
648H	648 avxyz
6500	650 abcdvxyz when Ind20
6501	650 abcdvxyz when Ind21
6502	650 abcdvxyz when Ind22
6503	650 abcdvxyz when Ind23
6504	650 abcdvxyz when Ind24
6505	650 abcdvxyz when Ind25
6506	650 abcdvxyz when Ind26
6507	650 abcdvxyz when Ind27
651H	651 abvxyz
654H	654 abvxyz
655H	655 abvxyz
656A	656 avxyz
657A	657 avxyz
658A	658 abc
6900	690 abcdvxyz
6910	691 abvxyz
700H	700 abcdefghklmnopqrstu
700X	700 t
710H	710 abcdefghklmnoprstu
710X	710 t
711H	711 abcdefghklnpqstu
711X	711 t
7300	730 adfghklmnoprstx
7301	730 alnps
7302	730 p
7400	740 ahnp

7401	740 p
7402	740 a
7600	760 t
7620	762 t
7670	767 abst
7720	772 a
7721	772 t
7730	773 a
7731	773 t
7800	780 t
7850	785 t
7910	791 abcdefgklnptu
8000	800 t
800H	800 abcdefghklmnopqrstuv
8100	810 t
810H	810 abcdefghklmnoprstuv
8110	811 t
811H	811 abcdefghklnpqstuv
8300	830 adfghklmnoprstv
8301	830 alnps
8302	830 p