Converting SQLBase Databases to Microsoft Access

By William Sprouse
Converting SQLBase Databases
To
Microsoft Access

By William Sprouse

Center for Environmental Management of Military Lands
Colorado State University
1490 Campus Delivery
Fort Collins, C) 80523-1490
(970) 491-2748
This document was produced under the Land Condition Trend Analysis (LCTA) Technical Support for the U.S. Army Environmental Center (AEC) contract; contract number DAAD13-00-D-5003-0041, task order number CS01-SC-03-016.
1 | **Introduction** | I
---|---|---
1.1 | Exporting Data from SQLBase | 1
1.2 | Importing Data into Microsoft Access | 4
1.3 | Checking Data Field Properties in SQLBase | 5
1.4 | Checking Data Field Properties in Microsoft Access | 6
1.5 | Checking for Primary Keys in SQLBase | 7
1.6 | Defining Primary Keys in Microsoft Access | 8
1.7 | Checking the Foreign Keys in SQLBase | 9
1.8 | Defining Foreign Keys in Microsoft Access | 10
1.9 | Conclusion | 13
1 Introduction

When LCTA was originally developed a standard database management system was selected to store the data collected by LCTA. The database selected was SQLBase by Gupta.

Since then many new database tools have become available. Some of these new tools are packaged with complete office software solutions and are now very popular. Microsoft Access, for example, has risen to the forefront of popularity in the database arena and many installations have converted their databases to Microsoft Access. Some major reasons for this transition are:

- The Microsoft Office suite of software is often an installation standard
- Favorable reviews of Microsoft Access
- Ease of use
- Interoperability with other software

This document is intended to assist the reader with converting their LCTA databases from SQLBase to Microsoft Access.

1.1 Exporting Data from SQLBase

Quest is a Windows user interface supplied with SQLBase to allow management of SQLBase databases. The following section explains the steps required to export data from SQLBase using Quest.

Start Quest and connect to the database. To connect to a database select Utilities / Database / Add from the main menu and select the database from the resulting dialog box. Figure 1 shows that Quest has been connected to the LCTAJUS database.
The first step in converting a SQLBase database is to export the data in the database to a dBase format.

1. Click the Open button on the toolbar.
2. Click the Table button on the toolbar.
3. Select the table from the resulting dialog box.
4. Select Edit / Copy To from the main menu (Figure 2).
5. Select dBase in the List Files of Type list box (Figure 3).
6. Select a directory for the output file.
7. When the export is complete a message will be displayed in the lower left corner of the Quest window.
8. Close the table.
9. Repeat the steps above for each table in the database.
Figure 2: Saving a table to a dBase file.

Figure 3: Quest copy data to a file dialog box.
1.2 Importing Data into Microsoft Access

After saving the data in the SQLBase database to dBase files Microsoft Access is used to import the data.

Start Microsoft Access and create a new blank database. Import the data from the files by performing the following steps:

1. Select **File / Get External Data / Import** from the main menu.
2. Select dBase IV (*.dbf) in the **Files of type** list box (Figure 4).
3. Navigate to the directory containing the exported dBase files and select a file.
4. Click the **Import** button.
5. A dialog box will appear when the import is done.
6. Repeat steps 3-5 above until all files are imported.

![Figure 4: Microsoft Access import dialog box.](image)

The imported tables will now appear in the database window under the tables section (Figure 5).
1.3 Checking Data Field Properties in SQLBase

When data is imported in this manner only the data field names, data type, and the table data are imported. Table relationships and some data field properties are not imported.

The properties of each data field in SQLBase are stored in the system table SysColumns. To view this information query that table using Quest.

1. Click the New button on the toolbar.
2. Click the SQL button on the toolbar.
3. Type the following in the SQL window:
   ```sql
   select tbname, name, coltype, length, nulls
   from syscolumns
   where tbname not like 'SYS%'
   order by tbname, colno;
   ```
4. Select SQL / Execute SQL Statement from the main menu.
5. The result will be shown in the SQL window below the SQL statement, which can be printed by selecting File / Print from the main menu.

The SQL statement that was used above will return the table name, data field name, data type, length of the field, and the nullity rule for all tables in the database that are not system tables. All system table names in SQLBase begin with SYS. The resulting data from the query is sorted by table name and the order of the data field in the table.
1.4 Checking Data Field Properties in Microsoft Access

The data field properties obtained from the steps in the previous section need to be compared to the information for each table in Access. To view the properties of a table select a table in the database window and click the *Design* button on the database window toolbar (Figure 5).

Properties for each data field in the table are shown in the properties window (Figure 6). Next to each data field name is the data type. The data type can be changed by clicking on the data type field and selecting the desired type from the list box. Further information for each data field is shown in the lower left portion of the window. The information shown is for the currently active data field. To change the active data field click on a data field name.

![Microsoft Access table properties](image)

*Figure 6: Microsoft Access table properties.*

In most cases the field size only needs to be checked for text data fields. Text fields are listed as CHAR and field size is listed as length in the query results from section 1.3.
The nullity rule from Quest, listed under the nulls column of the Quest output, designates if a data field can contain nulls. If a data field can be left blank the nullity rule will be Y. If a data field requires a value then N will be shown. This information relates to the required field in the table property window in Access. Setting the Required field to No in Access is the same as the nullity rule in Quest of Y. If the nulls column of the Quest output is Y this means nulls are allowed or data is not required. The nullity rule information is not imported into Access from dBase files.

1.5 Checking for Primary Keys in SQLBase

Referential integrity constraints are utilized to ensure data integrity in a database. These constraints are defined between a parent table and a child table by defining primary and foreign keys. A parent is defined with a data element, or set of elements, as a primary key. The primary key is a unique value, or set of values, that constrain the entry of data into the dependent child table. For example, PLOTMAST is a parent table to PLOTSURV with PLOTID as the primary key in PLOTMAST and the foreign key in PLOTSURV. If a particular value for PLOTID does not exist in PLOTMAST data for that plot can not be added to the PLOTSURV table.

The primary keys of each table are stored in the system table SysPKConstraints. To view this information query that table using Quest.

1. Click the New button on the toolbar.
2. Click the SQL button on the toolbar.
3. Type the following in the SQL window:
   ```sql
   select name, colname
   from syspkconstraints
   order by name, pkcolseqnum;
   ```
4. Select SQL / Execute SQL Statement from the main menu.
5. The result will be shown in the SQL window below the SQL statement (Figure 7), which can be printed by selecting File / Print from the main menu.
In the example shown in Figure 7 the table BIRDSURV has a primary key which contains three data fields; PLOTID, RECDATE and PERIOD. The primary key of the table MATED_STATUS only has the data field MATED_STATUS as its primary key.

1.6 Defining Primary Keys in Microsoft Access

Use the information from the previous section to set the primary keys of the imported tables in Access. In Access, the primary key of a table is defined in the table properties window (Figure 6). Display the properties window for a table as described above. Select the data field that will become the primary key by clicking on the box to the left of the data field name, which will highlight the entire row for that data field. If a primary key contains multiple data fields, hold down the Ctrl key while clicking the boxes. To define the selected data fields as the primary key of the table click the key button in the toolbar (Figure 8). A key will now be displayed to the left of each data field name that will become the primary key. Close the properties window, select Yes when prompted to save the changes.

Figure 8: Microsoft Access table properties toolbar.
1.7 Checking the Foreign Keys in SQLBase

The foreign keys of each table are stored in the system table SysFKConstraints. To view this information query that table using Quest.

1. Click the New button on the toolbar.
2. Click the SQL button on the toolbar.
3. Type the following in the SQL window:
   ```sql
   select name, refscolumn, refdtbname, refdcolumn
   from sysfkconstraints
   order by name, fkcolseqnum;
   ```
4. Select SQL / Execute SQL Statement from the main menu.
5. The result will be shown in the SQL window below the SQL statement (Figure 9), which can be printed by selecting File / Print from the main menu.

![SQL Query Result](image)

Figure 9: Foreign key query results in Quest.

In the example shown in Figure 9 the AERCOVER table contains the foreign keys PLOTID and RECDATE which are joined to the PLOTSURV table data fields PLOTID and RECDATE. PLOTSURV is a parent table to the child table AERCOVER.
1.8 Defining Foreign Keys in Microsoft Access

Use the information from the previous section to define relationships between tables in Access. In Access relationships are defined in the relationships window. Follow the steps below to view the relationships window.

1. Select **Tools / Relationships** from the main menu.
2. Hold down the Ctrl key and select each table from the Show Table dialog (Figure 10) that requires relationships.
3. The relationships window will open displaying all of the selected tables (Figure 11).
4. If a table needs to be added to the relationships window select **Relationships / Show Table** from the main menu and select the table.

![Figure 10: Microsoft Access show table dialog.](image)
Each rectangle in Figure 11 represents a table. For each table the data field names are listed. Data fields that are part of the primary key are shown in bold. Relationships will be defined between the parent table and the child table using the following steps.

1. Starting with the parent table, the table containing the primary key of the relationship, select the data fields of the primary key. If the primary key contains multiple data fields hold down the Ctrl key while selecting the data field names.
2. While holding down the right mouse button drag the cursor over the child table and release the mouse button. A dialog box similar to Figure 12 will appear. If the primary key only has one data field the values in the Table/Query and Related Table/Query columns will be filled in. If the primary key contains multiple data fields this information will have to be selected. Under the Related Table/Query column select the data field that corresponds to the data field listed under the Table/Query column. In the example shown in Figure 12 the data fields PLOTID and RECDATE of the PLOTSURV table (parent table) are joined to the PLOTID and RECDATE data fields of the GNDCOVER table (child table). Finally, click the Enforce Referential Integrity check box and click the Create button.
Repeat the steps above for all relationships. When a relationship is defined lines will be drawn between the parent and child tables to represent the relationship (Figure 13). Information about a defined relationship can be seen by double clicking the line that connects two tables.
1.9 Conclusion

Table 1 below outlines the steps required to convert a SQLBase database to Microsoft Access. For each process the section number of this document that describes the process is listed.

<table>
<thead>
<tr>
<th>Step</th>
<th>Process</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Export Data from SQLBase</td>
<td>1.1</td>
</tr>
<tr>
<td>2</td>
<td>Import Data into Microsoft Access</td>
<td>1.2</td>
</tr>
<tr>
<td>3</td>
<td>Check Data Field Properties in SQLBase</td>
<td>1.3</td>
</tr>
<tr>
<td>4</td>
<td>Check Data Field Properties in Microsoft Access</td>
<td>1.4</td>
</tr>
<tr>
<td>5</td>
<td>Check for Primary Keys in SQLBase</td>
<td>1.5</td>
</tr>
<tr>
<td>6</td>
<td>Define Primary Keys in Microsoft Access</td>
<td>1.6</td>
</tr>
<tr>
<td>7</td>
<td>Check for Foreign Keys in SQLBase</td>
<td>1.7</td>
</tr>
<tr>
<td>8</td>
<td>Define Foreign Keys in Microsoft Access</td>
<td>1.8</td>
</tr>
</tbody>
</table>

Once the data has been imported, table properties checked, and relationships defined the conversion process is complete. Before deleting the original SQLBase database and the corresponding software make sure the database conversion was successful.