An underutilized feature of **TECHBASE** is the *Table priority* option in the *Database* menu. In a question and answer format, this **Technote** discusses what table priority is, how it functions, and better ways to use the option.

Question: What is a table priority list?

Answer: A table priority list describes the logical order of the tables in your database. You see your tables listed in this order in the database review window (the **^R** option). As you construct a database, you will notice that the tables are listed in their order of creation. The exceptions to this rule are JOIN tables, which are added to the end of the priority list, although still in the order in which they were defined.

Question: How does *Table priority* work?

Answer: The *Table priority* option in the *Database* menu has two main functions. First, it defines the search order for **TECHBASE** programs when a field is referenced. For example, if you are in *TBEDIT* and ask to view a field called **elevation**, *TBEDIT* will search through the tables in the order in which they occur in the priority list until it finds the first occurrence of **elevation**. Although **elevation** may exist in several tables, *TBEDIT* will only display the records from the first table if finds.

Second, the *Table priority* option allows you to change the order of tables in the priority. For a database with a simple structure, this is a less important issue. But when your database grows in complexity and you begin to employ JOIN tables, the priority list becomes increasingly important (especially when you begin to join JOIN tables!). Once a priority list has been changed, it is saved automatically, along with the database and filters, if you have specified *YES* for *Save Values* in the *Options* menu.

Question: What happens when you do not specify the *Table priority*?

Answer: When *Table priority* is not specified, **TECHBASE** automatically (as a default) displays the tables in the order in which they were first created, with the exception of JOIN tables, which are listed last.

Question: Why is *Table priority* so important?

Answer: *Table priority* is crucial to **TECHBASE** users because of its ability to dictate the search order for **TECHBASE** operations, especially those that require field names as the primary input. Listed below are some examples of how a neglect of table priority can lead to confusing or inaccurate results.

EXAMPLE 1. You are trying to load by KEY into your **COLLARS** table, which has a single key on the field **hole_id**. Instead, you receive the message:

```
KEY fields MUST ALL be in list for load-by-KEY
```

This load will not run because **LOAD**, when searching for **hole_id**, found a JOIN table listed ahead of **COLLARS** in the priority list. Since the JOIN table has no keyed fields, the load could not be completed.

EXAMPLE 2. When in *TRIGRID*, you specify **east, north** and **elevation** fields for your *DATA, X, Y and VALUE* entries. However, when you *GRID*, you find that your resulting map doesn't look anything like your topography. On closer examination of your priority list, you find that another table earlier in the list has the same fields, and you have actually gridded a completely different surface. *TRIGRID* found your fields first in the wrong table, and so produced an incorrect map.

EXAMPLE 3. You have a geology table joined to your collars table. You want to report average unit thickness for all holes on the east side of your property. You set a filter for your collar east-coordinate, and write a report with **HOLE_ID** and **thickness**. You look at the report and see that it has the thickness values for all the holes. What happened to your filter? Nothing! The filter is right where you left it, applying to your collar and join tables. Unfortunately, your report was made from the geology table, which does not have a filter. For the result you want, use table priority to put the join ahead of the geology table, so that the report will be made from the (filtered) join table.

Question: How can you avoid problems related to *Table priority*?

Answer: We suggest the following methods for a productive use of *Table priority*:

Sketch out a logical order for your database before you define it. Try to set up your database in a logical order you can remember. Perhaps you could start with location data, followed by analyses then modeled data.

Become familiar with your table order. Be aware of which tables you have joined and where you have the same field or fields in more than one table. You can view you priority list in any menu that allows for the database review hotkey **^R,** or by using the **^R** button on the tool bar.

Use table names. Where possible for field and table lists, such as in *REPORT* and *TBEDIT*, use the table name instead of a set of field names. A table name is a far better reference choice because table names are unique and override the normal search pattern that **TECHBASE** employs for fields.

Use unique field names. This way you can be sure that you are referencing the correct table. One popular naming convention for fields that contain similar types of data is to modify each field name with a unique suffix (e.g. for assays, have **from_a**, for geology have **from_g**, for surveys have **from_s**, and so on).

Believe the error message. **TECHBASE** error messages that result from an improper table priority are generally straightforward. If the message informs you that a field cannot be found or no relation exists, even when you know otherwise, take a moment to examine your priority list for the fields that you are requesting.

Don't forget when you change the priority list. Once you have set your Table priority to a certain order, do not forget what changes you have made! Many users set a particular order for a single procedure, forget about the re-arrangement, and then discover later that their other tasks will not work because the order is no longer correct for their application.

Permanently change table order. You can permanently change table order in *Define/Database/order Tables*. Once you have set your *Table priority* to a certain order, do not forget what changes you have made! Many users set a particular order for a single procedure, forget about the re-arrangement, and then discover later that their other tasks will not work because the order is no longer correct for their application. Remember that the table priority list is saved when *Save Values* is *YES* in the *Options* menu. Use Define/Print to produce hard copy output of the permanent order of your tables. (NOTE: if you have set table priority in other Techbase programs this order will be reflected in the onscreen view of your database accessed using the ^R hotkey, but will not be reflected when hard copy output is requested in Define/Print).

Explicitly save table priority and filters. If you want to save a set of filters, and table priority configuration for a specific purpose, consider using the *Database/saVe setup* option in any Techbase program. Your settings will be saved to a .sav file, which you can restore using *Database/Restore setup* option.

For more information on the *Table priority* option in the *Database* menu, see the **DATABASE(0x)** section of the **TECHBASE Reference Manual.** For details concerning the *REPORT*, *DEFINE*, and *TBEDIT* programs, see the **Techbase(1tb)** section of the **TECHBASE Reference Manual**.

