Author: **Victor Sevastopol, Giprotsvetmet Institute, Moscow RUSSIA**

When loading data to several related tables, there is always a chance that data will be mis-coded in a way that creates "orphan" records in related tables. An example of an orphan record is an ASSAY record without a matching COLLAR record. This **TECHnote** will show how to find orphan records in a set of three drill hole data tables: **COLLARS, ASSAYS**, and **SURVEYS**. Each of these tables is related by a common **id** field.

Procedure:

Step 1.
With the *DEFINE* program, create a FLAT table called **check.** Add your **id** field to the table, and key the table on that **id**. Create three TEXT fields called **chk_collar, chk_assay,** and **chk_survey.** Each will be one character, LEFT justified. Add these three fields to the **check** table.

Step 2.
Use the *REPORT* program to report the records for each table in turn. For the **COLLARS** table, *Setup* the report as follows:

Report file: ***CHK_COLL.RPT***
Page layout / Field values? ***YES, all other parameters NO.***
Fields: ***id "+" COLLARS_nul***

It is not necessary to *sOrt* the report. By saying NO to parameters other than the *Field values*, the report will contain only the data values, without any headings, page breaks, etc. By including the **COLLARS_nul** field, the report must come from the **COLLARS** table. Run this report, then switch to the **ASSAYS** table:

Report file: ***CHK_ASSA.RPT***
Fields: ***id "+" ASSAYS_nul***

After running the **ASSAYS** report, switch to the **SURVEYS** table:

Report file: ***CHK_SURV.RPT***
Fields: ***id "+" SURVEYS_nul***

and run that report, for a total of three report files.

> **Note:** Another way to ensure that the report comes from the **COLLARS** table, rather than from another table containing the **id** field, is to use the *Table priority* choice in the *Database* menu. For each of the reports, simply name the required table first in the priority list.

Step 3.
Use the *LOAD* program to load each of the reported files into the **check** table. Start with the **COLLARS** data. The *Update type* will be REPLACE and the field list will be:

```
id chk_collar
```

Then load the ASSAYS data. The *Update type* will be KEY and the field list will be:

```
id chk_assay
```

Finally load the SURVEYS data. The *Update type* will be KEY and the field list will be:


```
          id  chk_survey
```

Step 4.
Report the data from your **check** table. You can sort on the **id**, if this will make the report easier to read. If all your tables have records for the **id**, you will have a plus sign (+) in each of the fields in the **check** table. If any of the tables does not have records for an **id**, the field will remain blank. The following example shows each of the possible combinations.

```
id       chk_collar  chk_assay   chk_survey
95-101   +           +           +           All tables have records
95-102               +           +           No COLLAR records
95-103   +           +                       No SURVEY records
95-104   +                                   No ASSAY or SURVEY records
95-105               +                       No COLLAR or SURVEY records
95-106                           +           No COLLAR or ASSAY records
95-107   +                       +           No ASSAY records
```

Step 5.
Examine the report carefully to resolve problems with the data. For example, is an **ASSAYS** record missing because it has not yet been received from the lab? Is a **COLLARS** record missing because the **id** was spelled inconsistently?

Variations
The "+" in the report output indicates that the table has at least one record in the indicated table. What if you want the report to tell you how many records of each type exist for each **id**? If so, the **chk_collar, chk_assay,** and **chk_survey** fields should all be defined as INTEGER ACTUAL. Then run the *REPORT* program for each table, as above, but for the **COLLARS** table *Setup* the report as follows:

Report file: C***HK_COLL.RPT***
Page layout / Field values? ***NO, all other parameters NO.***
Fields: ***id (b) COLLARS_rec (ns)***
sOrt: ***id***

Repeat this procedure for the **ASSAYS** and the **SURVEYS** tables. Now when you run the *LOAD* program, use the same procedure as before, but you will find that the final report will include numbers rather than the "+"characters:

```
id     chk_collar    chk_assaychk_survey
95-101             1 23   7
95-102               27   6
95-103             1 32
95-104             1
95-105               42
95-106                    8
95-107             1      1
```

For more information on the database setup for drill hole data, see the**"Technote: Database Setup for Drillhole Data" on July, 1991**. For information on using the *REPORT* program for generating formatted reports, see **"Page layout" on page tb-49**. For more information on using the *LOAD* program to update records by key, see **"Load – Load data files into a database" on page tb-37**.

