WHAT ARE KEY FIELDS?

It is essential to understand what makes a record unique before key fields can be used effectively. Key fields are index fields designed to make each record in a table unique. Key fields are created to:

1. Preserve database integrity.
2. Load data by key.
3. Join related tables.

In this Technote we will concentrate on examples of how to set up key fields, using key fields to preserve database integrity and loading data by key. Join tables and their uses will be covered in a later Technote.

WHERE ARE KEY FIELDS SET?

In the *DEFINE* program, the choice *Key field* can be used to assign the desired keys only if there is no data currently in the table you wish to key. If data has already been loaded into the table then use the *DEFINE* program and the *Rekey* choice, to add or remove key fields.

WHEN WOULD A KEY BE USED?

Preserve Database Integrity: One example might be laboratory results that are continually being updated. For instance, when you receive the original results, you create a database and load the data into the database

```
Table LOCATION   is a FLAT table with   4 records
            LOCATION_rec      LOCATION_nul      *Hole_id          Easting
            Northing          Td
Table VALUES     is a FLAT table with 16 records
            VALUES_rec        VALUES_nul        *Hole_id          *From
            To                Value
```

Note: Key fields are flagged with an asterisk "*" in the database review.

In the above example, the field *Hole_id* in the LOCATION table is the key field. Therefore, each record must have a single *Hole_id* (i.e., there is only one record per hole_id.) This is illustrated by the LOCATION data shown below.

In the VALUES table *Hole_id* alone is not unique for each record because the hole identification (id) is repeated for each interval down the hole (i.e., there are many records with the same hole id). To make a record unique in the VALUES table, more than one field must be keyed. The following example shows the data before key fields are added and the results you would get by keying different fields in the VALUES table.

LOCATION Data:

Hole_id	Easting	Northing	Td
1	2050.04	18964.10	40
2	2691.08	19279.20	30
3	3532.12	19497.70	30
4	3350.27	19010.30	35

VALUES Data:

Hole_id	From	To	Value
1	0	10	.05
1	10	20	.03
1	20	30	.04
1	30	40	.05
2	0	15	.04
2	15	20	.02
2	20	25	.02
2	325	30	.04

Hole_id	From	To	Value
3	0	10	.04
3	10	15	.06
3	15	.20	.5
3	20	25	.05
3	25	30	.04
4	0	10	.04
4	10	25	.04
4	25	35	.04

If only *Hole_id* is keyed in the VALUES table then only one record per hole is preserved and all the other data would be deleted. The same is true if only *From* or *To* is keyed in the VALUES table, then only one *From* or *To* would be preserved for each record. The results of the different keys would be:

Only *Hole_id* keyed

Hole_id	From	To	Value1
1	30	40	.05
2	25	30	.04
3	25	30	.04
4	25	35	.04

Only *From* keyed

Hole_id	From	To	Value1
1	30	40	.05
3	20	25	.05
3	15	20	.05
4	0	10	.04
4	10	25	.04
4	25	35	.04

Only *To* keyed

Hole_id	From	To	Value1
1	30	40	.05
3	25	30	.04
3	10	15	.06
3	15	20	.05
4	0	10	.04
4	10	25	.04
4	25	35	.04

One way to make a record unique in the VALUES table is to key *Hole_id* and *From*. When these fields are keyed, only one record exists for each interval. The same is true if *Hole_id* and *To* is keyed. Then if the laboratory sends corrections to some of the old data plus new data, key field(s) would be used to update existing records with the new data.

> **Note:** Keys do not imply any specific order. Specifying a Key field does not automatically sort the table on that field.

Load Data by Key: Once the fields are properly keyed then "Key" can be used as the *Update Type* in the *Load* program. If "Key" is chosen as the *Update Type* and you have new records that did not exist previously, the message "Matching KEY NOT FOUND for record "will occur when the data is loaded. This is merely, an informative message. If "Append" is chosen as the *Update Type* you will get the same results as in a "Key" load. However, a different message, "Matching KEY already exists for record" will occur. Again this is just an informative message. "Overlay" cannot be used as an *Update Type* when a table is keyed. "Replace" will delete all old data and replace it with new data. When "Replace" is used the old data is permanently deleted.

When loading data by key, any corrected values will be matched with the existing records, and any new data will be appended. Also, the integrity of the data is maintained by disallowing duplicate records. Following are two data files and the results of *Load* by "Key":

File 1

Hole_id	From	To	Value1
1	0	10	.05
1	10	20	.03
1	20	30	.04

File 2

Hole_id	From	To	Value2
1	10	20	.35
6	0	10	.05
2	20	25	.25

Load by Key Results

Hole_id	From	To	Value1	Value2
1	0	10	.05	
1	10	20	.03	.35
1	20	30	.04	
2	20	25		.25
6	0	10		.05

TECHBASE will check for an existing record that holds the exact *Hole_id* and *From* before it loads the value into the matching record.

See Also

For more information on **"Key field" on page tb-20**. For more information on changing Key fields see **"reKey" on page tb-29**. For a description of JOIN tables, see **"TECHBASE Tables" on page 0-3** For information on creating a JOIN table, see **"JOIN" on page tb-16**. For instructions on loading data, see **"Load – Load data files into a database" on page tb-37**.

