

From The Edge:

by

A. Razzak Memon

BACKUP, UNLOAD, PACK versus RELOAD

<http://www.razzak.com/fte>

01-15-1999

From the Edge: BACKUP, UNLOAD, PACK versus RELOAD
Section: Database Maintenance
Chapter: Running R:BASE Your Way!
Platform: R:BASE 6.1a/R:BASE 2000 (version 6.5)

- A. Introduction
- B. BACKUP and UNLOAD
- C. How R:BASE Stores Data
- D. When Should You Backup?
- E. File Maintenance
- F. PACK versus RELOAD
- G. What should be the choice?

A. Introduction:

Once a database is in use, it becomes another asset of the company to insure and maintain. Just as all airplanes are insured and have a regular maintenance procedure, so should database. The insurance is the BACKUP copies of the data. The maintenance plan includes procedures that will increase the response time of the database and help ensure data integrity

B. BACKUP and UNLOAD:

Since your data is valuable, a regular backup procedure should be followed. A backup is the only means of restoring data if something happens to the working copy through human error or technical malfunction. The backup should be on other media to guarantee that nothing happens to it. In other words, a copy of the database on the same hard disk is not a backup. Making a backup (used while testing a command file, for example) to the same media is acceptable for a temporary backup, but should not be treated as the permanent copy.

There are three commands in R:BASE used to back up a database or part of it: COPY, UNLOAD and BACKUP.

COPY is used to copy a small database to a floppy disk, or to another directory for a temporary backup. COPY does not compress the database to remove any space currently occupied by deleted rows or tables, nor does it span multiple floppy disks. COPY is not recommended for regular backups because of this.

BACKUP is excellent for backing up a database large enough to span multiple floppy disks. The RESTORE command is used to restore it. By using the BACKUP and RESTORE commands you can backup and restore part of a database, for example the data for a single table, or just the structure of the database. BACKUP does not backup rows in the database marked as deleted.

UNLOAD is similar to BACKUP but does not span multiple disks. UNLOAD also allows you to output data in ASCII delimited format and is often used to export data from one database to another, or from R:BASE to another products that accepts ASCII delimited format. INPUT command is used to restore the data.

The BACKUP and UNLOAD commands copy the data and structure of the database in a different format than the dbname.rb1, dbname.rb2, dbname.rb3 and dbname.rb4 files. The file created contains R:BASE commands to restore or rebuild the database. The database name and settings are written to the file along with data and table structures where applicable.

Apart from backing up an entire database there are other times when BACKUP and UNLOAD are useful. Below are situations and examples of the suggested BACKUP and UNLOAD syntax. The differences between the output are explained.

1. You have just added a new form to the database and want a new backup of all forms. Because each form

created in a database is stored as a row in SYS_FORMS2 table (R:BASE 6.1a/6.5 for Windows) or rows in SYS_FORMS table (R:BASE 6.1a/6.5 for DOS) you will backup the entire table.

UNLOAD ALL is the recommended command--allowing you to make a copy of the structure and data of the SYS_FORMS2 or SYS-FORMS table.

```
-- For R:BASE 6.1a/6.5 for DOS
OUTPUT forms.bak
UNLOAD ALL FOR SYS_FORMS
OUTPUT SCREEN
```

```
-- For R:BASE 6.1a/6.5 for Windows
OUTPUT forms.bak
UNLOAD ALL FOR SYS_FORMS2
OUTPUT SCREEN
```

Note: Using R:BASE for Windows, UNLOAD ALL FOR SYS_FORMS2 will also create an additional file, forms.LOB

Forms.bak is an ASCII file containing R:BASE commands, database information, and data (R:BASE for DOS only).

Forms.lob contains the graphical image data for all forms for R:BASE for Windows only.

2. You are responsible for backing up today's changes to table1 and table2 tables. You are not sure what the size of table1 will be when backed up as you have deleted more than one hundred rows throughout the course of the day.

BACKUP ... DATA ... WHERE is the recommended command--allowing you to backup data from two tables and span multiple disks if necessary. The WHERE clause is used to backup only today's data.

```
OUTPUT daily.bak
BACKUP DATA FOR table1 WHERE OrderDate = .#DATE
BACKUP DATA FOR table2 WHERE MaintDt = .#DATE
OUTPUT SCREEN
```

In this file you store only the data for the two tables, plus the database settings. There are no commands to define the tables.

C. How R:BASE Stores Data:

Data is stored in a database in the order that it is input. If you add 25 rows to the table1, add 10 rows to the table2, and then add more rows to the table1 the data for the table1 is not located together in the dbname.rb2 file. We would say these rows are not contiguous. When restoring from a backup you cause all data from one table to be restored before data from the next table is restored. This ensures that all data for each table is located in one area in the dbname.rb2 file, thus speeding up file access response time. R:BASE does not have to search different locations in the file for the data for one table.

D. When Should You Backup?

You should consider making a backup after all tables, forms and reports are designed, before data is added to the tables. In this way you have a copy of the structure in a single file for future development or reference.

Consider backing up forms and reports to separate files as well. Every time you change or create a form or report a new backup should be made.

If all tables in your database are updated regularly then you will probably decide to backup the entire database. Otherwise you might want to backup only the tables that change.

Daily backups of all tables that are updated should always be done.

Weekly backups of the entire database are strongly recommended.

E. File Maintenance:

When data is added, changed and deleted from a database the performance and size of the database will change.

Remember also that rows or tables that have been deleted from the database still occupy space within the database files. If you do not compress your files regularly the size of your database file may become unnecessarily large.

Not only do rows for a table become scattered within dbname.rb2 file, but the actual database files (dbname.rb1, dbname.rb3 and dbname.rb4) can become scattered on the disk. We call this disk fragmentation. As the size of the database increases so the likelihood of the database files being fragmented increases. The more fragmented the files on the hard disk are, the longer data searches take. When reloading or restoring a database, OS writes the new files in an area without fragmenting them.

Performing regular tune-ups on the files allows the files to remain as efficient as possible. There are two commands that are recommended: PACK and RELOAD.

PACK will compress the database, releasing space occupied by deleted rows and tables. The references to all indexed values are rebuilt in the dbname.rb3 file. PACK works directly with the database and will corrupt the files if interrupted. A backup should always be made before using PACK.

RELOAD creates a copy of the database in which all unusable space is recovered and all rows for a table are located contiguously within dbname.rb2. References to indexed values are also rebuilt. RELOAD builds the copy either in the same directory with another name, or in another directory with the same name. Typically RELOAD is used for file maintenance, not backups.

F. PACK versus RELOAD:

Use PACK whenever a table has been removed from the database, or many rows have been deleted from a table. This compresses the database. Choose PACK if you do not have enough room on the disk for two copies of the database.

Use RELOAD when you have enough room on a disk to create a second copy of the database. The resulting files are normally smaller than the original.

G. What should be the choice?

The schedule you choose to do either will depend on database size, the number of tables updated regularly, and the number of rows deleted.

As PACK requires less time and fewer commands, I recommend it be included in an application.

RELOAD requires more time, a knowledge of disk space and how to rename or copy database. I suggest that a RELOAD schedule be set up and carried out by the database administrator.

Remember, if the size of most of the tables in a database remains constant, and you are adding data to a few tables on a regular basis, the data in the dbname.rb2 will not be too scattered. In this case, PACK would be the best command to use weekly, with a RELOAD being done monthly.

If you do not have enough room on the disk to reload then consider using BACKUP. This will give you a copy of the database on other media. After backing up, delete the original database and restore from the backup. This achieves exactly the same results as RELOAD but requires more work on your part.