Table Load/Unload   Sep 21, 2012

To unload the contents of a database table to a file:

psql dbname

==> \COPY tablename TO filename USING DELIMITERS '|' WITH NULL AS ''

To load the contents of a file into a database table:

 psql dbname

==> \COPY tablename FROM filename USING DELIMITERS '|' WITH NULL AS ''

Note that these statements have no ";" (semi colon) at the end!

In version 8.3.x of postgres (used by AWIPS II), the above commands will be

        \COPY tablename FROM filename WITH DELIMITER '|' NULL ''

        \COPY tablename TO filename   WITH DELIMITER '|' NULL ''

\COPY TO and \COPY FROM stop at the first error. If an error occurs while loading a table using \COPY … FROM, the records already loaded are not visible or accessible but TAKE UP SPACE ON THE DISK. VACUUM must be run to make this space available for reuse.

Default delimiter for \COPY is TAB. The above example shows how to change this to use the pipe symbol '|' as a delimiter.

Both COPY TO and COPY FROM and \COPY … TO and \COPY … FROM allow a list of columns to be specified for copying. For example, to copy the lid and ts fields for all records in the hourlypp table to a file named "file.unl":

psql db_name
\copy hourlypp (lid, obsdate) to file.unl

Using a VIEW with \COPY FROM is not allowed!

To copy the contents of a table directly from one database to another (without writing to a file):

        psql hd_ob83fwr -c "copy height to stdout" | \
        psql hd_ob90fwr -c "copy height from stdin"

COPY TO and COPY FROM can also be run from the command line. You must be

logged in as user = postgres to COPY TO or COPY FROM using a file. If you are doing a COPY FROM stdin or COPY TO stdout, you do not have to be logged in as user=postgres. COPY TO and COPY FROM executed from the command line are more efficient than \COPY through psql.

Copying Part of a Table to a File
To copy the lid and obsdate fields for records in the hourlypp table with obsdate > 2007-06-21 to a file named "file.unl":

psql db_name
\copy (select lid, obsdate from hourlypp where obsdate > '2007-06-21') to file.unl with delimiter '|'

OR

\copy (select lid, obsdate from hourlypp where obsdate > '2007-06-21') to file.unl using delimiters '|' with null as ''

The above is for postgres Version 8.2.x and 8.3.x only.

Unload File Format
If | is the delimiter:
      field1 | field2 | field3   (no delimiter at end of line)

In Informix, the format was
      field1 | field2 | field3 |

Use of NULLs
- COPY FROM command needs the "WITH NULL AS" option to allow NULL values in
      datetime fields and integer fields to be loaded properly

postgres uses \N for null and back-to-back delimiters for empty string

Character Translation
postgres translates non-text characters; e.g., outputs \n for newlines and \t for tabs, if fields contain them

Skipping Records On Insert
Informix had the dbload utility which allowed skipping of records on insert which violate the primary key constraint. Postgresql has two options. An EXCEPT operator which can be used as follows:

INSERT INTO TimeZone
SELECT DISTINCT tzone FROM Location EXCEPT SELECT tzone FROM TimeZone;

The EXCEPT operator computes the set of rows in the result set of the left SELECT statement that are not in the result set of the right SELECT statement

Or the pgloader utility:

http://pgfoundry.org/projects/pgloader/

It is a Python program that 'manages' COPY. It will kick out bad rows and keep loading data.

Reducing Table Bloat
Another trick I like to use is to alter a table field from type integer to type integer. This causes pg to rewrite the table without trying to do any sorting and is MVCC safe to boot.

Copying a Table From One db to Another
You may need to set user & other parameters, but a quick example:

 to dump a table, rename it and recreate in a different db on another server on the fly:

```
pg_dump -h  -t   |  sed 's/src table/target table/g'  |  psql -h  -d
```

By piping through sed you can also change column names, just as this does the table name.