ALGUNOS CONSEJOS PARA CARGAR LOS DATOS EN LA BASE DE DATOS

EDAT 2017-18

El procedimiento general para cargar datos en la base de datos ya ha sido comentado en el enunciado de la práctica y, supongo, en clase. Como ya probablemente os habeis dado cuenta, la parte más complidaca es conseguir leer los ficheros en las tablas temporales. Una vez que los datos estén alí, hay mmuchas maneras de tratarlos y obligarlos a hacer lo que queremos, pero sobre los ficheros no tenemos ningún control, y el COPY no es una instrucciń especialmente potente.

La idea aquí es conseguir meter los datos en una tabla, sea como sea, y luego manipularlos con cómodo una vez que ya estén en la base de datos. Esto siempre reserva sorpresas y problemas que hay que ir solucionando a medida de que se presentan. Por ejemplo, casi todos los consejos que os voy a dar en este documento derivan de problemas de que yo no tenía ni idea, problemas que se presentaron en la calse de práctica y que hemos solucionado entre todos; algunas de las soluciones son mias, otras se le ocurrieron a algún colega vuestro.

Uno de los problemas quw muchos han encontrado se debe a la presecia de "tildes" (á, é, etc.) de eñes y otros caracteres que el COPY no reconoce. Si habéis intentado subir el fichero, seguro que os habrá llegado un error del tipo "caracter erroneo en la codificaión UTF-8" o cosas del estilo. Una solución es limpiar el fichero con un *global replace* de "é" con "e", "ñ" con "n" o "ny" etc. No es un trabajo muy largo: con un buen editor (yo uso emacs) y algunos intentos para ver donde hay errores, se puede limpiar el fichero en unos 15 minutos.

Hay un comando del sistema operativo, *iconv*, que podéis usar para convertir el fichero a UTF-8 de un golpe. Sé que el programa existe (es estándar en todas las distribucciones de Linux), pero no lo he usado nunca, así que para usarlo habrá que chaucear un poco y leerse el "man" ("man iconv" en la terminal os da la documentación del comando). Para saber como está codificado el fichero, podéis usar otro comando Linux: el comando *file*. En mi caso, por ejemplo, tengo esta información:

```
p2> file books_01.txt
books_01.txt: ISO-8859 English text, with very long lines, with CRLF line terminators
p2>
```

Un compañero vuestro ha encontrado una solución mejor: el comando COPY tiene una opción para cambiar la codificación de la entrada, y algunas condificaciones sí que aceptan tildes y todo el resto. Se puede cambiar la codificación que el COPY reconoce usando la opción ENCODING. Una codificación que parece funcionar es *ISO-8859-1*, por tanto dando el comando

```
COPY MiTable from '/user/myself/code/myfile' WITH DELIMITER "     " ENCODING 'ISO-8859-1'
```

En este caso también, habrá que chapucear un poco para encontrar la condificación que funciona.

* * *

Dado que la copia de datos es algo delicada, es una buena idea crear las tablas temporales con tipos de datos que se tragan todo como, por ejemplo, VARCHAR. Un problema que alguien ha enido con el fichero de libro es que en algunos de ellos la fecha de publicación es '0000-00-00' y si intentáis leer esta fecha en una columna de tipo DATE, os da error. La solución es crear la tabla temporal con la columna de tipo VARCHAR(20)[1]. Puedo, por ejemplo, creat una tabla temporal así (ahora no tengo a mano el fichero de libros y no me acuerdo que campos hay... lo creo un poco de fantasía... you get the gist...)

```
CREATE TABLE MyTemp(titulo VARCHAR(300), autor VARCHAR(200),
                    editor VARCHAR(200), fecha VARCHAR(50),
                    tipo VARCHAR(200))
```

En esta table puedo leer las fechas '0000-00-00' sin problema. Pero... ¿ahora cómo lo concverto? Fácil. Digamos que mi table final tiene título, autor, editor y fecha, dode la fecha se ha declarado como DATE. Puedo insertar en la tabla sólo las tuplas con fecha válida, haciendo al mismo tiempo la conversión de tipo:

```
INSERT INTO MyFinal  SELECT titulo, autor, editor, to_date(fecha, 'YYYY-MM-DD')
                     FROM MyTemp
                     WHERE fecha <> '0000-00-00'
```

Esta instrucción inserta en la tabla sólo las fechas válidas, haciendo la conversión. Si no queréis tirar las tuplas con la fecha no válida, se le puede añadir una fecha de fantasía, por ejemplo la fecha de hoy. EN este caso se hará

```
INSERT INTO MyFinal  SELECT titulo, autor, editor, to_date(fecha, 'YYYY-MM-DD')
                     FROM MyTemp
                     WHERE fecha <> '0000-00-00'

INSERT INTO MyFinal  SELECT titulo, autor, editor, to_date('2016-10-23', 'YYYY-MM-DD')
                     FROM MyTemp
                     WHERE fecha = '0000-00-00'
```

o, alternativamente:

```
INSERT INTO MyFinal  (SELECT titulo, autor, editor, to_date(fecha, 'YYYY-MM-DD')
                     FROM MyTemp
                     WHERE fecha <> '0000-00-00')
                     UNION
                     (SELECT titulo, autor, editor, to_date('2016-10-23', 'YYYY-MM-DD')
                     FROM MyTemp
                     WHERE fecha = '0000-00-00')
```

El fichero de libros tiene tuplas con el ISBN duplicado. En algunos casos esto no debería dar problemas, en cuanto se trata de libros con distintos autores, y en el fichero hay una tupla por cada

[1] EN la declaración de varchar, no seáis tirado con las longitudes. Hay gente que ha tenido que volver a crear la table tres vees porque el título o no se que no le cabía. Se trata de tablas temporales que luego no se van a usar: ¿qué os cuesta declarar todo VARCHAR(300)? Es gratis y os evita problemas.

autor. Dado que en la tabla de ediciones (en que el ISBN es una clave) no aparece el autor, una inserción con uan consulta con un SELECT DISTINCT eliminará las tuplas duplicadas.

Por ejemplo, en el fichero aparecen estas dos tuplas (las divido en varias líneas por claridad, pero cada una es una línea del fichero:

```
Arturo Prez-Reverte Konigin des Sudens Libro de bolsillo Libro de bolsillo
       Suhrkamp Verlag Ag  2016-02-08 Aleman 3518466585

Angelica Ammar Konigin des Sudens Libro de bolsillo Libro de bolsillo
        Suhrkamp Verlag Ag  2016-02-08 Aleman 3518466585
```

Si la tabla en que lo insertamos tiene campos *titulo, tipo, editorial, fecha, ISBN* la instrucción

```
INSERT INTO edicion select distinct titulo, tipo, editorial, fecha, ISBN from libros
```

resuelve el problema. Lamentablemente, hay casos en que la cosa no funciona en cuanto el libro es el mismo pero en el fichero aparece de manera ligeramente diferente. Por ejemplo:

```
Noam Chomsky Manufacturing Consent The Political Economy of the Mass Media by Chomsky,
                Noam ( Author ) ON Jan-03-1998, Libro de bolsillo
           Libro de bolsillo Libro de bolsillo Vintage  1998-01-03 Ingls 0099533111

Noam Chomsky Manufacturing Consent: The Political Economy of the Mass Media
           Libro de bolsillo  432 pages Vintage  1995-04-20 Ingls 0099533111
```

En la primera línea, todo lo que hay hasta el primer *Libro de Bolsillo* es título. El problema es que queremos sólo una de estas dos líneas en nuestra tabla de ediciones. La solución (es un poco un apaño, pero no se me ocurre otra) es hacer un group-by sobre ISBN para garantizarnos que sólo vamos a tener uno. Pero si hacemos un group-by de ISBN todo lo demás que aparece en el select debe tener un operador de agregación. Es decir, de todos los títulos que aparecen con el mismo ISBN hay que elegir uno, y así de todos los tipos, etc. Podemos usar el operador MIN, que funciona con cadenas (nos devuelve la que viene primero en orden alfabetico o la más corta). por tanto la inserción en la tabla de ediciones ya no sería la de antes, sino:

```
INSERT INTO edicion
    select min(titulo), min(tipo), min(editorial), min(fecha), ISBN
    from libros
    group by ISBN
```

Con esto debería funcionar.