

Fundamentos de Programación

MERGE

APAREO

Merge

- Se mezclan dos o más archivos que tienen el mismo formato.
- Los archivos tienen que estar ordenados por algún campo.
- Se genera un archivo extra que los une a todos y queda ordenado también por ese campo.
- Ante igualdad de valores podemos optar por guardar en primer lugar los registros que provienen de cierto archivo o, también, podemos decidir que el orden es indistinto.

Merge: ejemplo

Por ejemplo, de un negocio hay varias sucursales y, de cada una, se genera un archivo diario de ventas. Ambos archivos están ordenados por fecha.

`fecha, vendedor, monto`

Merge con 2 archivos

Merge con dos archivos

Hay varias formas de realizar el merge, pero la más eficiente es la siguiente:

- Se procede a la lectura adelantada de una línea de cada archivo
- Luego se ingresa en un ciclo mientras alguna de las lecturas tenga una clave (fecha en este ejemplo) menor a la clave máxima.
- Dentro del ciclo, se calcula la clave mínima a procesar.
- Mientras la clave 1 sea igual a la clave mínima, se procesa (se graba) un registro del archivo 1 y se vuelve a leer de ese mismo archivo.
- El paso anterior se repite exactamente igual para el archivo 2. Mientras la clave 2 sea igual a la clave mínima, se procesa (se graba) un registro del archivo 2 y se vuelve a leer de ese mismo archivo.

Función merge

def mezcla (suc1, suc2, unificado):

 fecha1, vendedor1, monto1 = leer(suc1)

 fecha2, vendedor2, monto2 = leer(suc2)

 while (fecha1 < MAX) or (fecha2 < MAX):

 minimo = min(fecha1, fecha2)

 while (fecha1 == mínimo):

 guardar(fecha1, vendedor1, monto1, unificado)

 fecha1, vendedor1, monto1 = leer(suc1)

 while (fecha2 == mínimo):

 guardar(fecha2, vendedor2, monto2, unificado)

 fecha2, vendedor2, monto2 = leer(suc2)

Merge con 3 o más archivos

Similar a dos archivos, pero queda un código más largo. Si tuviéramos que hacer una mezcla, por ejemplo, con diez archivos, una opción podría consistir en ir haciendo mezclas de a pares: sucursal 1 con sucursal 2. Luego esta unión con la sucursal 3, y así sucesivamente.

Otra opción más eficiente sería realizar la mezcla entre la sucursal 1 y la 2, la 3 con la 4, etcétera. De esta manera, en el caso de que fueran diez archivos, nos quedarían cinco archivos. Luego, volver a realizar mezclas entre pares, quedando tres archivos, etcétera.

Apareo

Similar a una mezcla, pero en general se hace siempre con dos archivos en donde tenemos: un archivo con datos, que llamaremos archivo maestro y otro con novedades. El objetivo es actualizar el archivo maestro con las novedades que contiene el otro archivo. Generalmente el archivo actualizado nos quedará en un nuevo archivo, ya que no podemos modificar las líneas de un archivo.

Apareo: ejemplo

Por ejemplo, tenemos un archivo con los números de legajo y los sueldos de cada empleado de una empresa. El archivo de novedades podría tener la actualización de esos sueldos.

Apareo: variante

El archivo novedades también podría tener alta de empleados y baja de alguno de ellos. Entonces debería tener un carácter extra que indique: A (alta), B (baja), M (modificación). En estos casos podríamos tener que generar un archivo *log* o *errores* en donde registremos ciertos casos anómalos: como querer modificar o dar de baja a un legajo que no existe o querer dar de alta a un legajo existente.

Apareo: algoritmo

El procesamiento es similar al realizado en un *merge*: se van leyendo los registros en paralelo: uno del archivo maestro y uno del archivo de novedades. Se pregunta por el menor y se procesa en consecuencia:

Apareo: algoritmo

- Si el menor es del archivo maestro, significa que no hay novedades, por lo que se graba en el archivo definitivo o maestro actualizado. Se vuelve a leer un registro del archivo maestro.
- Si el menor es del archivo novedades, debe ser un alta, de lo contrario, es un error.
 - Si es un alta, se guarda en el archivo maestro actualizado.
 - Si no, se guarda el registro en el archivo *log*.
 - En ambos casos se vuelve a leer el archivo de novedades.
- Si las claves de los registros del archivo maestro y del archivo de novedades son iguales, tenemos tres casos:
 - Si es alta, es un error. El registro maestro se guarda en el archivo actualizado y el de novedades va al archivo *log*.
 - Si es baja, los registros no se guardan, es decir, no se hace nada.
 - Si es modificación, se actualiza el registro del archivo maestro con el de novedades, y se guarda en el archivo actualizado.
 - En cualquiera de los tres casos, se leen ambos archivos.

Fin
