Subject: Re: Batch Script connecting two databases in an Extract Tranform Load (ETL) Design Pattern
Posted by rafs on Thu, 18 Jun 2015 16:52:12 GMT
View Forum Message <> Reply to Message

Hi Tony,

Can you give me some pointers? (1) Do you have a description on how to make use of the parent/child relationships defined on a table? Presently, I am only able to use a single instance of a table, and the getData with $where argument to retrieve arrays of data. I want to understand if I can make better use of the relationships between objects.

If a package has many files, and a file has many lines, where the
converse can also be true--that a line can belong to many files, and packages.
packages --< packages_files >-- files --< files_lines >-- lines

The lines already exist, but I want to create new files and packages from lines.
I want to select several sets of lines, then create a new file for each set, then create a package from these files. And, I want all of the cross/joining tables (packages_files, and files_lines) to be updated as well.

```
$lines = RDCsingleton::getInstance('lines');
$lines->sql_where    = "id > 100 AND id < 200";
$lines_data = $lines->getData();
```

(2) What is the difference between putting my where clause in sql_where, vs into $where?

```
$files_lines = RDCsingleton::getInstance('files_lines');
$files = RDCsingleton::getInstance('files');

$file_array = array('id'=>null, 'name'=>"file1");
$files_data = $files->insertRecord($file_array)[0];
```

(3) Can the parent or child table class handle these inserts into the linking tables for me? Or do I need to instantiate them and do it in a loop or something?

```
foreach ($lines_data as $i => $row) {
  $files_lines_data = $files_lines->insertRecord($files_data['id', $row['id'])[0];
  ...
}
...
$packages = RDCsingleton::getInstance('packages');
```