

```

$chge_log_file = date('mdYhis') . $this->change_log_file;
$sql = explode(";",file_get_contents($chge_log_file));
foreach($sql as $query) {
    If(!$mysqli->query($query))
        {
            throw new Exception("Dog Table Change Error: " . $mysqli->error);
        }
}

```

The code for the destructor becomes simpler than in the original MySQL example. The destructor does not need to format any SQL statements. It only needs to execute them. The method reads the change records from the change log, splitting each via the ; at the end of each SQL command line. Each line is placed in the array \$sql. The logic then loops through the array and executes each statement via the query command. If any of the SQL statements has a problem, an exception is thrown (which will also send an e-mail to support personnel via the `dog_interface` program). An example program is available on the book's web site.

Note: As stated, the MySQL examples are shown to help the reader see that the overall logic of the `dog_data` class works well with all data types. Complete books are written on using PHP to interact with databases. It is not the intent of this book to train the user to have complete knowledge of database manipulation.

Do It

1. The `dog_data` class creates a new log file every time it is run. This could cause a lot of log files to be created in a very short period of time. Your mission is to either update the `readchangelog` file (download it from the book's web site) or to create your own maintenance program. The code will ask the users for the number of log files (and data files) to keep. The program will then keep the most recent number of files requested. The `glob` method, as shown previously, can be used to retrieve all the file names. The `unlink` method can be used to delete a file.

```
unlink($file);
```

2. The MySQL examples shown now produce different contents in the change log file. Download the `readchangelog` program from the book's web site and make any adjustments needed to the code to properly view and delete the change log. Assuming that the database administrator has reversed the contents of the database to the last valid set of data, adjust the program to execute the change log selected against the database. Hint: Your completed program will have less code than the example from the book's web site.

Connecting the Data Tier

Now that a reliable, well tested, data class has been created, it is time to connect it to the business rules tier. The `Dog` class will use the `dog_data` class to store the dog information in an XML file.

```

if (method_exists('dog_container', 'create_object')) {
    $this->breedxml = $properties_array[4];
}

```