

# How to build an EXIF database to understand your photography

---

Some bits of code to get stats from the EXIF data from your photos.

---

I'm a linux user and Adobe had shut me out from their products - we don't even have and official Flash nowadays - but I wanted to examine which focal length, aperture, etc. I use the most in my photos. There is a fair amount of them ( tens of thousands) so I needed something fast & flexible. digiKam[^1] probably offers a solution for this, but it's a bit too heavy for my taste, so I went for a different approach.

## exiftool[^2]

There is program, called exiftool which can eat nearly any kind for image format, and it's pretty easy to use, even if it's a command line utility.

To install it on Ubuntu, run `sudo apt-get install exiftool` On debian it's `sudo apt-get install libimage-exiftool-perl`.

# Database

I could have used SQLite, which would make things pretty straightforward, but I went with MySQL since I had one running on the webserver. Note: for production things, use InnoDB, but writes can be slow with that. Since I do mostly writes here and it is not critical at all, I went with MyISAM.

sql

```
CREATE TABLE `files` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `fname` text NOT NULL, PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8
```

```
CREATE TABLE `exif` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `ename` text NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8
```

```
CREATE TABLE `data` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `fid` int(11) NOT NULL,  
  `eid` int(11) NOT NULL,  
  `value` text NOT NULL,  
  PRIMARY KEY (`id`),  
) ENGINE=MyISAM DEFAULT CHARSET=utf8
```

# The parser

Yes, I know, I should have given up PHP a long time ago, but I'm lazy.

php

```
<?php

$exif = stream_get_contents(STDIN);
$exif = json_decode($exif,true);
$exif = $exif[0];

$SourceFile = str_replace('./', '', $exif['SourceFile']);

unset ($exif['SourceFile']);
unset ($exif['Directory']);
$DBServer = 'db host';
$DBUser   = 'db user';
$DBPass   = 'db password';
$DBName   = 'db name';

$conn = new mysqli($DBServer, $DBUser, $DBPass, $DBName);

// check connection
if ($conn->connect_error) exit('Database connection failed:
' . $conn->connect_error);

$SourceFile = $conn->real_escape_string($SourceFile);

echo "Checking existence of ${SourceFile}\n";
$sql="SELECT ID FROM files WHERE fname='${SourceFile}'";
$rs=$conn->query($sql);

// in case the file was parsed already, quit this execution
if($rs === false) exit('Wrong SQL: ' . $sql . ' Error: ' .
$conn->error );
if ( $rs->num_rows > 0 ) exit ('File exists in db already');

// otherwise insert it into the files db
$sql = "INSERT INTO files (fname) VALUES('${SourceFile}')";
if($conn->query($sql) === false) exit ('Wrong SQL: ' . $sql .
```

```

' Error: ' . $conn->error );
$fid = $conn->insert_id;

echo "\tStarting inserting values for ${SourceFile}\n";
foreach ( $exif as $key => $value ) :

    // I don't want to store every exif parameter name every
time
    // so the parameter names are in a separate table
    $key = $conn->real_escape_string($key);
    $sql = "SELECT id FROM exif WHERE ename='${key}' LIMIT 1";
    $rs=$conn->query($sql);
    if($rs === false) exit('Wrong SQL: ' . $sql . ' Error:
' . $conn->error );
    if ( $rs->num_rows > 0 ) {
        $arr = $rs->fetch_all(MYSQLI_ASSOC);
        $eid = $arr[0]['id'];
    }
    else {
        echo "\tadding new exif param: ${key}\n";
        $sql = "INSERT INTO exif (ename) VALUES('${key}') ";
        if($conn->query($sql) === false) exit ('Wrong SQL:
' . $sql . ' Error: ' . $conn->error );
        $eid = $conn->insert_id;
    }

    // adding the exif value itself
    if (is_array($value) || is_object($value)) $value =
json_encode($value);
    $value = $conn->real_escape_string($value);
    $sql = "INSERT INTO data (fid, eid, value) VALUES('$
{fid}','${eid}','${value}')";
    if($conn->query($sql) === false) exit ('Wrong SQL: ' .
$mysql . ' Error: ' . $conn->error );

endforeach;
echo "\t${SourceFile} added to the DB\n";

```

## Glue it together

Replace \*.jpg with whatever files you want to search for and there is an -iregex option to use regex to match patterns.

```
cd directory/of/photos
```

```
bash
```

```
find . -iname *.jpg -exec bash -c "/usr/bin/exiftool -json  
'{}' | php /path/to/exifdb.php" \;
```

## Getting results

sql

```
# Your most commonly used focal lengths:
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`192.168.%` SQL
SECURITY DEFINER VIEW `top_10_focallength` AS select
`data`.`value` AS `value`,count(0) AS `count` from `data`
where (`data`.`eid` = (select `id` from `exif` where (`ename`
= 'FocalLength')))) group by `data`.`value` order by `count`
desc limit 10;
```

```
# Your most commonly used aperture:
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`192.168.%` SQL
SECURITY DEFINER VIEW `top_10_aperture` AS select
`data`.`value` AS `value`,count(0) AS `count` from `data`
where (`data`.`eid` = (select `id` from `exif` where (`ename`
= 'Aperture')))) group by `data`.`value` order by `count` desc
limit 10;
```

```
# Your lens:
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`192.168.%` SQL
SECURITY DEFINER VIEW `lens` AS select `data`.`value` AS
`value`,count(0) AS `count` from `data` where (`data`.`eid` =
(select `id` from `exif` where (`ename` = 'LensID')))) group
by `data`.`value` order by `count` desc;
```

And of course, any other query you can think about.

### Links

1. <https://www.digikam.org/>
2. <http://owl.phy.queensu.ca/~phil/exiftool/>

Created by Peter Molnar <[mail@petermolnar.net](mailto:mail@petermolnar.net)>, published at 2015-05-30 16:18 UTC, last modified at 2021-10-31 15:57 UTC , to canonical URL <https://petermolnar.net/article/how-to-build-an-exif-database-to-understand-your-photography/> , licensed under CC-BY-4.0 .