

Expanded Date Table Technical Paper

Introduction:

This paper will discuss the creation and population of an expanded date reference table. Most of the functionality contained in the table can be built independently through either functions or object-oriented programming. The advantage that the table affords in advanced date calculations and manipulations is the reduced overhead to compute the end result date parameters.

The code and sql contained in this paper is open source and available for your unrestricted use. There are no warranties applied to any of the following code.

Intended Audience:

The content of this paper is intended for intermediate to advanced PHP and MySQL programmers who have a need to probe date type of data on several levels. If you are just starting out in either PHP or MySQL, this paper may exceed your present skill set.

Table Structure:

The sql for creating the blank table can be found further down in the paper. The table is a MyISAM table with two indexes (fx, fdate_ms). If your application requires it, additional index keys may be added after the table is created. A separate record is created for each date in the table. Each record will consist of the following fields:

Field	Type	Description	Example
fx	mediumint(8)	Auto-incrementing index	
fdate_ms	date	MySQL format date	2009-01-01
fyr	smallint(5)	4 digit year	2009
fmo	tinyint(3)	Integer value of the month of the year	1
fmo_ms	char(2)	MySQL format for the month	01
fmo_day1	char(3)	Day of the week of month day 1	Mon
fmonth	char(3)	3 Char month name	Jan
fwk_mo	tinyint(3)	Week of the month (Sunday day 1)	1
fwk_yr	tinyint(3)	Week number of the year	1
fwk_day1	char(3)	Weekday name of the 1 st day of the week	Sun
fwk_da1	tinyint(3)	Integer date of 1 st day of the week	1
fda_mo	tinyint(3)	Integer day of the month	12
fda_yr	smallint(3)	Integer day of the year	15
fday	char(3)	3 Char day of the week	Mon
fserial	int(10)	Unix serialize value for the day	1167886800
fyr_idx	tinyint(3)	Incrementing index of the year (starts with 1)	2
fmo_idx	smallint(5)	Incrementing index of the month(starts with 1)	3

Expanded Date Table Technical Paper

fwk_idx	smallint(5)	Incrementing index of the week (starts with 1)	17
fda_idx	mediumint(8)	Incrementing index of the date (starts with 1)	233

Table Creation:

The following sql code will create the table for you.

```
CREATE TABLE IF NOT EXISTS `tr50` (  
  `fx` mediumint(8) unsigned NOT NULL auto_increment,  
  `fdate_ms` date NOT NULL,  
  `fyr` smallint(5) unsigned NOT NULL,  
  `fmo` tinyint(3) unsigned NOT NULL,  
  `fmo_ms` char(2) NOT NULL,  
  `fmo_day1` char(3) NOT NULL,  
  `fmonth` char(3) NOT NULL,  
  `fwk_mo` tinyint(3) unsigned NOT NULL,  
  `fwk_yr` tinyint(3) unsigned NOT NULL,  
  `fwk_day1` char(3) NOT NULL,  
  `fwk_da1` tinyint(3) unsigned NOT NULL,  
  `fda_mo` tinyint(3) unsigned NOT NULL,  
  `fda_yr` smallint(5) unsigned NOT NULL,  
  `fday` char(3) NOT NULL,  
  `fserial` int(10) unsigned NOT NULL,  
  `fyr_idx` tinyint(3) unsigned NOT NULL,  
  `fmo_idx` smallint(5) unsigned NOT NULL,  
  `fwk_idx` smallint(5) unsigned NOT NULL,  
  `fda_idx` mediumint(8) unsigned NOT NULL,  
  PRIMARY KEY (`fx`),  
  KEY `fdate_ms` (`fdate_ms`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Populating the Table;

The following program will populate the table with a distinct record for each day within the start and end date specified in the code. Please note that the code to make the MySQL connection and database access has been left out. You will need to insert the applicable code for your instance in the program.

Expanded Date Table Technical Paper

```
<?PHP
#*****
#
#   This program is authored by Timothy Bonesho.  You are free to use or modify the contents
#   of this program as required.  There are no warranties applicable to it's use or functionality.
#   Use at your own risk.
#
#*****

function calc_mysql_date($mysql_date_in,$offset)
{
    if($mysql_date_in<="0000-00-00")
    {
        return;
    }
    $year      = substr($mysql_date_in,0,4);
    $month     = substr($mysql_date_in,5,2);
    $day       = substr($mysql_date_in,8,2);
    $anchor    = mktime(0,0,0,$month,$day+$offset,$year);
    $nu_yr     = date("Y",$anchor);
    $nu_mo     = date("m",$anchor);
    $nu_da     = date("d",$anchor);
    $nu_date= $nu_yr.'-'. $nu_mo.'-'. $nu_da;
    return($nu_date);
}

#   insert the first date for the table in MySQL date format here
$date_in = '2007-01-01';
#   insert the last date for the table in MySQL date format here
$last = '2014-06-30';

#   insert your MySQL connection and database access equivalent code here

#   Empty the table
$sq100="truncate tr50";
$res100=mysql_query($sq100);

#   Initialization of the variables
$old_yr="";
$old_mo="";
$old_wk="";
$fyr_idx =0;
$fmo_idx =0;
$fwk_idx =1;
$fda_idx =0;

#   date record generation code and sql statement for insertion of the record into the table
#
while($date_in<$last)
{
    $fdate_ms = $date_in;
    $fyr = substr($date_in,0,4);
    $fmo_ms = substr($date_in,5,2);
```

Expanded Date Table Technical Paper

```
if(substr($fmo_ms,0,1) == '0')
{
    $fmo = substr($fmo_ms,1,1);
}
else
{
    $fmo = $fmo_ms;
}
$fda_ms = substr($date_in,8,2);
if(substr($fda_ms,0,1) == '0')
{
    $fda_mo = substr($fda_ms,1,1);
}
else
{
    $fda_mo = $fda_ms;
}

$fserial = mktime(0,0,0,$fmo,$fda_mo, $fyr);

if($fda_mo == "1")
{
    $fmo_day1 = strftime("%a",$fserial);
}

if($fyr != $old_yr)
{
    $test_wk = strftime("%U", $fserial);
    if($test_wk == '00')
    {
        $week_correction = 1;
    }
    else
    {
        $week_correction = 0;
    }
    $old_yr = $fyr;
    $fyr_idx++;
}
$fwk_yr = strftime("%U",$fserial) + $week_correction;
$fday = strftime("%a",$fserial);

$fda_yr = strftime("%j",$fserial);
if(substr($fda_yr,0,2) == '00')
{
    $fda_yr = substr($fda_yr,2,1);
}
elseif(substr($fda_yr,0,1) == '0')
{
    $fda_yr = substr($fda_yr,1,2);
}

if($fmo != $old_mo)
{
    $fmo_day1 = strftime("%a",$fserial);
    $fmo_da1 = $fday;
}
```

Expanded Date Table Technical Paper

```
$old_mo = $fmo;
$fwk_mo = 1;
$fwk_da1 = $fda_mo;
$fwk_day1 = $fday;
$fmonth = strftime("%b",$fserial);
$fmo_idx++;
}
if($fday == 'Sun' && $fmo_day1 == 'Sun' && $fda_mo > 7)
{
    $fwk_mo ++;
    $fwk_da1 = $fda_mo;
    $fwk_day1 = $fday;
    $fwk_idx++;
}
elseif($fday == 'Sun' && $fmo_day1 != 'Sun')
{
    $fwk_mo++;
    $fwk_da1 = $fda_mo;
    $fwk_day1 = $fday;
    $fwk_idx++;
}
}
$fda_idx++;
$sq1="insert into tr50
    (
        fdate_ms,
        fyr,
        fmo,
        fmo_ms,
        fmo_day1,
        fmonth,
        fwk_mo,
        fwk_yr,
        fwk_day1,
        fwk_da1,
        fda_mo,
        fda_yr,
        fday,
        fserial,
        fyr_idx,
        fmo_idx,
        fwk_idx,
        fda_idx
    )
values
    (
        '$fdate_ms',
        '$fyr',
        '$fmo',
        '$fmo_ms',
        '$fmo_day1',
        '$fmonth',
        '$fwk_mo',
        '$fwk_yr',
        '$fwk_day1',
        '$fwk_da1',
        '$fda_mo',
```

Expanded Date Table Technical Paper

```
        '$fda_yr',
        '$fday',
        '$fserial',
        '$fyr_idx',
        '$fmo_idx',
        '$fwk_idx',
        '$fda_idx'

    )";
    $res1=mysql_query($sq1);

    unset($fserial);
    $date_in = calc_mysql_date($date_in,1);
}
mysql_close();

print '<br><b>Finished populating table tr50<br></b>';

exit;
?>
```

Application:

This project started out with a need to expand the analysis level of my personal investment tracking programs. Approximately 65 different stocks and mutual funds are tracked to closing prices on a daily basis. The data spans several years of activity. The analysis requirement was to build an application that would build the weekly trade delta's between the week's first day of trade and the last day of trade. The scope was to examine 4 weeks of this activity with respect to the current date. This application relied quite heavily on the week index contained in the table. A cross reference to the date recorded in the financial model provided the basis for the date-based data.

This table also proved to be invaluable in the construction of a very large-scale html Gantt chart (in the work-work setting). Because of the large number of project tasks and date scaling, traditional PHP Gantt Chart generation applications were not up to the task (Jpgraph). The challenge in this application was to group the data in month blocks subdivided into weekly units. The use of the above table made this a workable solution for the generation of the Gantt Chart.

Expanded Date Table Technical Paper

Conclusion:

I hope the contents of this paper prove to be of value to you. The use of the tabled dates has saved considerable time and effort when constructing advanced date-based applications.

If you have an application based on the use of this table that you would like to share here, please use the contact information on the website.

Because of timing commitments, I can only make limited support-type of responses for this table.

Timothy Bonesho