MySQL Date Issues(Optional)

This material is optional. But if you are going to use various web pages etc as a source of information of working with temporal data, you need to read this.

When we enter and use date values I am assuming that you will use valid, complete date values. A data value such as 2010-03-18 is a valid, complete date. A value such as 2010-03-00 is not a complete date. MySQL can allow the use of incomplete dates. Some examples here will let you see why some people might want to use them- and some people won't want to allow these dates.

Demo 01: In the a_testbed database, create a table to store date values and insert some values.

```
Create table z_dates ( id int primary key, dtm date);

insert into z_dates values (1, '2010-02-28');

-- this one fails
insert into z_dates values (2, '2010-02-29');
ERROR 1292 (22007): Incorrect date value: '2010-02-29' for column 'dtm' at row
1
insert into z_dates values (3, '2010-02-00');
insert into z_dates values (4, '2010-00-00');
insert into z_dates values (5, '0000-00-00');
insert into z_dates values (6, '2010-04-29');
insert into z_dates values (7, '2010-00-29');
insert into z_dates values (8, '2010-04-00');
insert into z_dates values (9, '2010-04-15');
insert into z_dates values (10, '0000-04-29');

select * From z_dates;
+----+------------+
| id | dtm        |
+----+------------+
|  1 | 2010-02-28 |
|  3 | 2010-02-00 |
|  4 | 2010-00-00 |
|  5 | 0000-00-00 |
|  6 | 2010-04-29 |
|  7 | 2010-00-29 |
|  8 | 2010-04-00 |
|  9 | 2010-04-15 |
| 10 | 0000-04-29 |
+----+------------+
```

The insert for id 2 did not work but the others did. We would need to have some concept of what the date values are for rows 3, 4, 5, 7, 8, and 10. Remember this is not a string column- it is a date column.

Let's do a sort. If we scan down that date column, we can see that there is some sort of ordering. The Feb date comes before the April date and April 15 comes before Apr 29.
If we look at the incomplete dates, the 0 month 0 day value comes before the 0 month 29 day value.

```
select * From z_dates
order by dtm;
+----+------------+
| id | dtm        |
+----+------------+
|  5 | 0000-00-00 |
| 10 | 0000-04-29 |
|  4 | 2010-00-00 |
|  7 | 2010-00-29 |
|  3 | 2010-02-00 |
|  1 | 2010-02-28 |
|  8 | 2010-04-00 |
|  9 | 2010-04-15 |
|  6 | 2010-04-29 |
+----+------------+
```

Now let's do some selects. These results for tests on month and day seem reasonable

Demo 02:

```
select *
from z_dates
where month(dtm) = 2;
+----+------------+
| id | dtm        |
+----+------------+
|  1 | 2010-02-28 |
|  3 | 2010-02-00 |
+----+------------+
```

Demo 03:

```
select *
from z_dates
where day(dtm) = 29;
+----+------------+
| id | dtm        |
+----+------------+
|  6 | 2010-04-29 |
|  7 | 2010-00-29 |
| 10 | 0000-04-29 |
+----+------------+
```

And some date arithmetic.

Demo 04: The month function says that if we have a month component, it will be returned; if the month component was entered as a 0 then the month function brings back a 0. That is consistent.

```
select id, dtm, month(dtm)
from z_dates;
+----+------------+------------+
| id | dtm        | month(dtm) |
+----+------------+------------+
|  1 | 2010-02-28 |          2 |
|  3 | 2010-02-00 |          2 |
|  4 | 2010-00-00 |          0 |
|  5 | 0000-00-00 |          0 |
|  6 | 2010-04-29 |          4 |
|  7 | 2010-00-29 |          0 |
|  8 | 2010-04-00 |          4 |
|  9 | 2010-04-15 |          4 |
| 10 | 0000-04-29 |          4 |
+----+------------+------------+
```

Demo 05: If I try to add 5 days to each date the incomplete dates come back with nulls. If we consider the 00 day component to mean we don't know the day, then we cannot know the value of 5 days later. The return for row 10 is interesting

```
select id, dtm, date_add(dtm, interval 5 day)
from z_dates;
+----+------------+-------------------------------+
| id | dtm        | date_add(dtm, interval 5 day) |
+----+------------+-------------------------------+
|  1 | 2010-02-28 | 2010-03-05                    |
|  3 | 2010-02-00 | NULL                           |
|  4 | 2010-00-00 | NULL                           |
|  5 | 0000-00-00 | NULL                           |
|  6 | 2010-04-29 | 2010-05-04                     |
|  7 | 2010-00-29 | NULL                           |
|  8 | 2010-04-00 | NULL                           |
|  9 | 2010-04-15 | 2010-04-20                     |
| 10 | 0000-04-29 | 0000-00-00                     |
+----+------------+--------------------------------+
```

Demo 06: Add 5 months

```
select id, dtm, date_add(dtm, interval 5 month)
from z_dates;
+----+------------+---------------------------------+
| id | dtm        | date_add(dtm, interval 5 month) |
+----+------------+---------------------------------+
|  1 | 2010-02-28 | 2010-07-28                      |
|  3 | 2010-02-00 | NULL                            |
|  4 | 2010-00-00 | NULL                            |
|  5 | 0000-00-00 | NULL                            |
|  6 | 2010-04-29 | 2010-09-29                      |
|  7 | 2010-00-29 | NULL                            |
|  8 | 2010-04-00 | NULL                            |
|  9 | 2010-04-15 | 2010-09-15                      |
| 10 | 0000-04-29 | 0000-09-29                      |
+----+------------+---------------------------------+
```

Demo 07: Add 15 months

```
select id, dtm, date_add(dtm, interval 15 month)
from z_dates;
+----+------------+----------------------------------+
| id | dtm        | date_add(dtm, interval 15 month) |
+----+------------+----------------------------------+
|  1 | 2010-02-28 | 2011-05-28                       |
|  3 | 2010-02-00 | NULL                             |
|  4 | 2010-00-00 | NULL                             |
|  5 | 0000-00-00 | NULL                             |
|  6 | 2010-04-29 | 2011-07-29                       |
|  7 | 2010-00-29 | NULL                             |
|  8 | 2010-04-00 | NULL                             |
|  9 | 2010-04-15 | 2011-07-15                       |
| 10 | 0000-04-29 | 0001-07-29                       |
+----+------------+----------------------------------+
```

Demo 08: But the Date_format function has a slightly different rule. It seems to say that it will format as much as it can.

```
select id, dtm, Date_format( dtm, '%M %e, %Y')
from z_dates;
+----+------------+--------------------------------+
| id | dtm        | Date_format( dtm, '%M %e, %Y') |
+----+------------+--------------------------------+
|  1 | 2010-02-28 | February 28, 2010              |
|  3 | 2010-02-00 | February 0, 2010               |
|  4 | 2010-00-00 | NULL                           |
|  5 | 0000-00-00 | NULL                           |
|  6 | 2010-04-29 | April 29, 2010                 |
|  7 | 2010-00-29 | NULL                           |
|  8 | 2010-04-00 | April 0, 2010                  |
|  9 | 2010-04-15 | April 15, 2010                 |
| 10 | 0000-04-29 | April 29, 0000                 |
+----+------------+--------------------------------+
```

The incomplete date values are used when you do not know the exact date. Row id 3 is a date in February 2010. Row id 7 is the 29th of some month in 2010, and Row id 10 is April 29 of some year.

If you store dates such as these, you should not expect to get correct results for functions that require complete dates such as the arithmetic functions for dates. Some of these function will return a 0 for the incomplete part; some will return a null if given an incomplete date.

The date value '0000-00-00' is called a dummy date and is sometimes used instead of using a Null

There is a setting which allows invalid dates- which accepts any month from 0 to 12 and any day from 0 to 31.

There is another setting which says that it will not accept any zero values in a date.

If you decide to use these types of dates in your tables, then you need to consider how to handle them in a consistent method when the date values are used. Also read the manual on topics such as

```
   Allow_Invalid_Dates
   No_Zero_Date
```

And display any warnings so that you have a better view of what is happening.

And don't use them for the class assignments.