#### Working with Date and Time

Date and time are of particular interest for database applications. This is because business records often carry date/time information (e.g., orderDate, deliveryDate, paymentDate, dateOfBirth), as well as the need to time-stamp the creation and last-update of the records for auditing and security.

With date/time data types, you can sort the results by date, search for a particular date or a range of dates, calculate the difference between dates, compute a new date by adding/subtracting an interval from a given date.

##### Date By Example

Let's begin with Date (without Time) with the following example. Take note that date value must be written as a string in the format of 'yyyy-mm-dd', e.g., '2012-01-31'.

-- Create a table 'patients' of a clinic

mysql> **CREATE TABLE patients (**

**patientID INT UNSIGNED NOT NULL AUTO\_INCREMENT,**

**name VARCHAR(30) NOT NULL DEFAULT '',**

**dateOfBirth DATE NOT NULL,**

**lastVisitDate DATE NOT NULL,**

**nextVisitDate DATE NULL,**

**-- The 'Date' type contains a date value in 'yyyy-mm-dd'**

**PRIMARY KEY (patientID)**

**);**

mysql> **INSERT INTO patients VALUES**

**(1001, 'Ah Teck', '1991-12-31', '2012-01-20', NULL),**

**(NULL, 'Kumar', '2011-10-29', '2012-09-20', NULL),**

**(NULL, 'Ali', '2011-01-30', CURDATE(), NULL);**

-- Date must be written as 'yyyy-mm-dd'

-- Function CURDATE() returns today's date

mysql> **SELECT \* FROM patients;**

+-----------+---------+-------------+---------------+---------------+

| patientID | name | dateOfBirth | lastVisitDate | nextVisitDate |

+-----------+---------+-------------+---------------+---------------+

| 1001 | Ah Teck | 1991-12-31 | 2012-01-20 | NULL |

| 1002 | Kumar | 2011-10-29 | 2012-09-20 | NULL |

| 1003 | Ali | 2011-01-30 | 2012-10-21 | NULL |

+-----------+---------+-------------+---------------+---------------+

-- Select patients who last visited on a particular range of date

mysql> **SELECT \* FROM patients**

**WHERE lastVisitDate BETWEEN '2012-09-15' AND CURDATE()**

**ORDER BY lastVisitDate;**

+-----------+-------+-------------+---------------+---------------+

| patientID | name | dateOfBirth | lastVisitDate | nextVisitDate |

+-----------+-------+-------------+---------------+---------------+

| 1002 | Kumar | 2011-10-29 | 2012-09-20 | NULL |

| 1003 | Ali | 2011-01-30 | 2012-10-21 | NULL |

+-----------+-------+-------------+---------------+---------------+

-- Select patients who were born in a particular year and sort by birth-month

-- Function YEAR(date), MONTH(date), DAY(date) returns

-- the year, month, day part of the given date

mysql> **SELECT \* FROM patients**

**WHERE YEAR(dateOfBirth) = 2011**

**ORDER BY MONTH(dateOfBirth), DAY(dateOfBirth);**

+-----------+-------+-------------+---------------+---------------+

| patientID | name | dateOfBirth | lastVisitDate | nextVisitDate |

+-----------+-------+-------------+---------------+---------------+

| 1003 | Ali | 2011-01-30 | 2012-10-21 | NULL |

| 1002 | Kumar | 2011-10-29 | 2012-09-20 | NULL |

+-----------+-------+-------------+---------------+---------------+

-- Select patients whose birthday is today

mysql> **SELECT \* FROM patients**

**WHERE MONTH(dateOfBirth) = MONTH(CURDATE())**

**AND DAY(dateOfBirth) = DAY(CURDATE());**

-- List the age of patients

-- Function TIMESTAMPDIFF(unit, start, end) returns the difference in the unit specified

mysql> **SELECT name, dateOfBirth, TIMESTAMPDIFF(YEAR, dateOfBirth, CURDATE()) AS age**

**FROM patients**

**ORDER BY age, dateOfBirth;**

+---------+-------------+------+

| name | dateOfBirth | age |

+---------+-------------+------+

| Kumar | 2011-10-29 | 0 |

| Ali | 2011-01-30 | 1 |

| Ah Teck | 1991-12-31 | 20 |

+---------+-------------+------+

-- List patients whose last visited more than 60 days ago

mysql> **SELECT name, lastVisitDate FROM patients**

**WHERE TIMESTAMPDIFF(DAY, lastVisitDate, CURDATE()) > 60;**

-- Functions TO\_DAYS(date) converts the date to days

mysql> **SELECT name, lastVisitDate FROM patients**

**WHERE TO\_DAYS(CURDATE()) - TO\_DAYS(lastVisitDate) > 60;**

-- Select patients 18 years old or younger

-- Function DATE\_SUB(date, INTERVAL x unit) returns the date

-- by subtracting the given date by x unit.

##### Date/Time Functions

MySQL provides these built-in functions for getting the current date, time and datetime:

* NOW(): returns the current date and time in the format of 'YYYY-MM-DD HH:MM:SS'.
* CURDATE() (or CURRENT\_DATE(), or CURRENT\_DATE): returns the current date in the format of 'YYYY-MM-DD'.
* CURTIME() (or CURRENT\_TIME(), or CURRENT\_TIME): returns the current time in the format of 'HH:MM:SS'.

For examples,

mysql> **select now(), curdate(), curtime();**

+---------------------+------------+-----------+

| now() | curdate() | curtime() |

+---------------------+------------+-----------+

| 2012-10-19 19:53:20 | 2012-10-19 | 19:53:20 |

+---------------------+------------+-----------+

##### SQL Date/Time Types

MySQL provides these date/time data types:

* **DATETIME**: stores both date and time in the format of 'YYYY-MM-DD HH:MM:SS'. The valid range is '1000-01-01 00:00:00' to '9999-12-31 23:59:59'. You can set a value using the valid format (e.g., '2011-08-15 00:00:00'). You could also apply functions NOW() or CURDATE() (time will be set to '00:00:00'), but not CURTIME().
* **DATE**: stores date only in the format of 'YYYY-MM-DD'. The range is '1000-01-01' to '9999-12-31'. You could apply CURDATE() or NOW() (the time discarded) on this field.
* **TIME**: stores time only in the format of 'HH:MM:SS'. You could apply CURTIME() or NOW() (the date discarded) for this field.
* **YEAR(4|2)**: in 'YYYY' or 'YY'. The range of years is 1901 to 2155. Use DATE type for year outside this range. You could apply CURDATE() to this field (month and day discarded).
* **TIMESTAMP**: similar to DATETIME but stored the number of seconds since January 1, 1970 UTC (Unix-style). The range is '1970-01-01 00:00:00' to '2037-12-31 23:59:59'.  
  The differences between DATETIME and TIMESTAMP are:
  1. the range,
  2. support for time zone,
  3. TIMESTAMP column could be declared with DEFAULT CURRENT\_TIMESTAMP to set the default value to the current date/time. (All other data types' default, including DATETIME, must be a constant and not a function return value). You can also declare a TIMESTAMP column with "ON UPDATE CURRENT\_TIMESTAMP" to capture the timestamp of the last update.

The date/time value can be entered manually as a string literal (e.g., '2010-12-31 23:59:59' for DATAETIME). MySQL will issue a warning and insert all zeros (e.g., '0000-00-00 00:00:00' for DATAETIME), if the value of date/time to be inserted is invalid or out-of-range. '0000-00-00' is called a "dummy" date.

##### Example

Create a table with various date/time columns. Only the TIMESTAMP column can have the DEFAULT CURRENT\_TIMESTAMP and ON UPDATE CURRENT\_TIMESTAMP.

mysql> **CREATE TABLE IF NOT EXISTS `datetime\_arena` (**

**`description` VARCHAR(50) DEFAULT NULL,**

**`cDateTime` DATETIME DEFAULT '0000-00-00 00:00:00',**

**`cDate` DATE DEFAULT '0000-00-00',**

**`cTime` TIME DEFAULT '00:00:00',**

**`cYear` YEAR DEFAULT '0000',**

**`cYear2` YEAR(2) DEFAULT '00',**

**`cTimeStamp` TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP**

);

mysql> **DESCRIBE `datetime\_arena`;**

+-------------+-------------+------+-----+---------------------+-----------------------------+

| Field | Type | Null | Key | Default | Extra |

+-------------+-------------+------+-----+---------------------+-----------------------------+

| description | varchar(50) | YES | | NULL | |

| cDateTime | datetime | YES | | 0000-00-00 00:00:00 | |

| cDate | date | YES | | 0000-00-00 | |

| cTime | time | YES | | 00:00:00 | |

| cYear | year(4) | YES | | 0000 | |

| cYear2 | year(2) | YES | | 00 | |

| cTimeStamp | timestamp | NO | | CURRENT\_TIMESTAMP | on update CURRENT\_TIMESTAMP |

+-------------+-------------+------+-----+---------------------+-----------------------------+

Insert values manually using string literals.

mysql> **INSERT INTO `datetime\_arena`**

**(`description`, `cDateTime`, `cDate`, `cTime`, `cYear`, `cYear2`)**

**VALUES**

**('Manual Entry', '2001-01-01 23:59:59', '2002-02-02', '12:30:30', '2004', '05');**

mysql> **SELECT \* FROM `datetime\_arena` WHERE description='Manual Entry';**

+--------------+---------------------+------------+----------+-------+--------+---------------------+

| description | cDateTime | cDate | cTime | cYear | cYear2 | cTimeStamp |

+--------------+---------------------+------------+----------+-------+--------+---------------------+

| Manual Entry | 2001-01-01 23:59:59 | 2002-02-02 | 12:30:30 | 2004 | 05 | 2010-04-08 14:44:37 |

+--------------+---------------------+------------+----------+-------+--------+---------------------+

Checking the on-update for TIMSTAMP.

mysql> **UPDATE `datetime\_arena` SET `cYear2`='99' WHERE description='Manual Entry';**

mysql> **SELECT \* FROM `datetime\_arena` WHERE description='Manual Entry';**

+--------------+---------------------+------------+----------+-------+--------+---------------------+

| description | cDateTime | cDate | cTime | cYear | cYear2 | cTimeStamp |

+--------------+---------------------+------------+----------+-------+--------+---------------------+

| Manual Entry | 2001-01-01 23:59:59 | 2002-02-02 | 12:30:30 | 2004 | 99 | 2010-04-08 14:44:48 |

+--------------+---------------------+------------+----------+-------+--------+---------------------+

Insert values using MySQL built-in functions now(), curdate(), curtime().

mysql> **INSERT INTO `datetime\_arena`**

**(`description`, `cDateTime`, `cDate`, `cTime`, `cYear`, `cYear2`)**

**VALUES**

**('Built-in Functions', now(), curdate(), curtime(), now(), now());**

mysql> **SELECT \* FROM `datetime\_arena` WHERE description='Built-in Functions';**

+--------------------+---------------------+------------+----------+-------+--------+---------------------+

| description | cDateTime | cDate | cTime | cYear | cYear2 | cTimeStamp |

+--------------------+---------------------+------------+----------+-------+--------+---------------------+

| Built-in Functions | 2010-04-08 14:45:48 | 2010-04-08 | 14:45:48 | 2010 | 10 | 2010-04-08 14:45:48 |

+--------------------+---------------------+------------+----------+-------+--------+---------------------+

Insert invalid or out-of-range values. MySQL replaces with all zeros.

mysql> **INSERT INTO `datetime\_arena`**

**(`description`, `cDateTime`, `cDate`, `cTime`, `cYear`, `cYear2`)**

**VALUES**

**('Error Input', '2001-13-31 23:59:59', '2002-13-31', '12:61:61', '99999', '999');**

mysql> **SELECT \* FROM `datetime\_arena` WHERE description='Error Input';**

+-------------+---------------------+------------+----------+-------+--------+---------------------+

| description | cDateTime | cDate | cTime | cYear | cYear2 | cTimeStamp |

+-------------+---------------------+------------+----------+-------+--------+---------------------+

| Error Input | 0000-00-00 00:00:00 | 0000-00-00 | 00:00:00 | 0000 | 00 | 2010-04-08 14:46:10 |

+-------------+---------------------+------------+----------+-------+--------+---------------------+