Date and Time Functions

- In this tutorial we will see the different date and time functions
- More details are available in the following link-

  https://dev.mysql.com/doc/refman/5.6/en/date-and-time-functions.html

Current_date

- CURRENT_DATE(): It returns the current date.
- Example:
  - **select current_date()**
  - **select curdate()**

Current Time

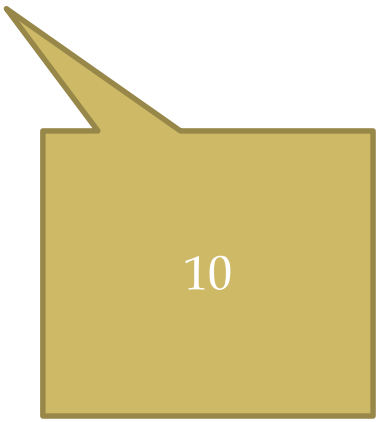
- CURRENT_TIME(): returns the current time as a value in 'HH:MM:SS' or 'HHMMSS' format
- Example:
  - **select current_time**()
  - **select now():** returns the time at which the function or triggering statement began to execute (includes the date also)
  - **select sysdate():** returns the time at which it executes (includes date also)

DATE & Extract

- DATE(expr): extracts date part from the expression *expr*
- Example:
  - **select date**('1998-12-19 11:34:45')

Extract

- Extract(unit from date)
- Unit can be day, week, month, year, quarter, hour, minute, second, etc.
  - Example: select extract (month from '1999-10-19 10:11:12')



DATEDIFF(exp1,exp2)

- DATEDIFF(exp1, exp2): returns ***expr1*** – ***expr2*** expressed as a value in days from one date to the other
- ***expr1*** and ***expr2*** are date or date-and-time expressions.
- Only the date parts of the values are used in the calculation
- Example:
  - **SELECT** DATEDIFF('2013-09-04 23:59:59', '2013-08-30')

DATE_ADD/ADDDATE

- Performs date arithmetic
- DATE_ADD(date, INTERVAL expr_unit):
  - date argument specifies the starting date or datetime value.
  - expr is an expression specifying the interval value to be added to or subtracted from the date.
  - unit may be Microsecond, second, minute, day, week, month, quarter, year, etc.

- Example:
  - **SELECT** DATE_ADD('2013-08-31', **INTERVAL** 10 **DAY**)
  - **SELECT** DATE_ADD('2001-09-17 23:59:59', **INTERVAL** 1 **SECOND**)

DATE_SUB

- A similar function but it will perform date subtraction
- Example:
  - **SELECT** DATE_SUB('2013-09-04', **INTERVAL** 100 **DAY**)

DATE_FORMAT

- DATE_FORMAT(date, format): it formats the date value according to the format string
- In the ***format*** string specifier is used along with the %symbol
- Example:
  - **SELECT** DATE_FORMAT('1998-10-18', '%D %b %Y')

  - **SELECT** DATE_FORMAT('1998-10-18', '%d %c %y')

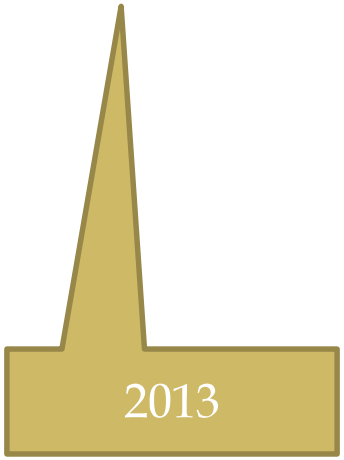
Specifies Table

Specifier	Description
%a	Abbr. weekday name (like Sun..Sat)
%b	Abbr. month name (like Jan..Dec)
%c	Month numeric (0..12)
%D	Day of the month with English suffix (0^{th}, 1^{st}, 2^{nd}, 3^{rd} , …)
%d	Day of the month, numeric (00..31)
%M	Month name (January,…, December)
%m	Month, numeric (00,..,12)
%Y	Year numeric (4 digits)
%H	Hour (00..23)
%h	Hour (01..12)
%i	Minutes, numeric (0..59)
%s	Seconds (00,..59)
%p	AM or PM

- DAY(date): returns the day of the month
- DAYOFMONTH(date): returns the day of the month
- DAYNAME(date): returns the name of the weekday
- DAYOFYEAR(date): returns the day of the year

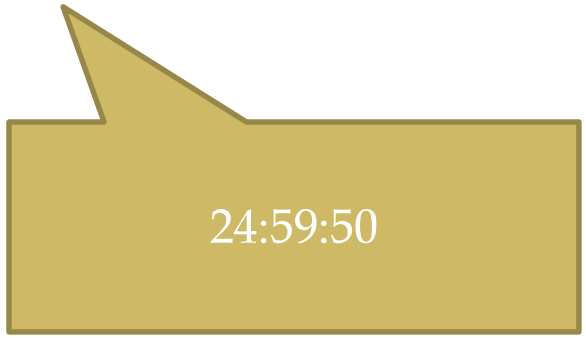
YEAR

- YEAR(date): returns the year for date in the range from 1000 to 9999
- Example:
  - **select year**('2013-01-01')



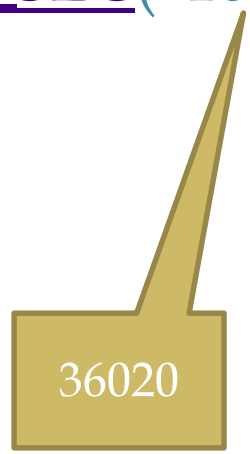
TIMEDIFF

- TIMEDIFF(expr1, expr2): returns ***expr1*** – ***expr2*** expressed as a time value.
- ***expr1*** and ***expr2*** are time or date-and-time expressions, but both must be of the same type.
- Example:
  - **SELECT** TIMEDIFF('2017-08-16 12:00:00', '2017-08-15 11:00:10')



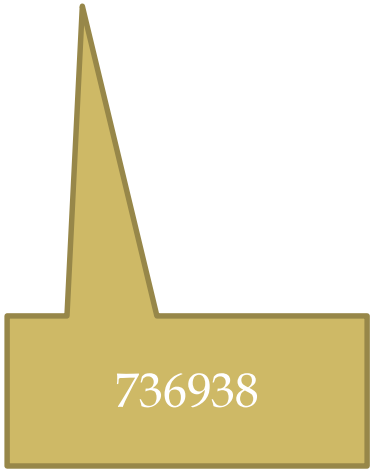
TIME_TO_SEC

- TIME_TO_SEC(time): returns the time argument converted to seconds
- Example:
  - **SELECT** TIME_TO_SEC('10:00:20')



TO_DAYS

- TO_DAYS(date): Given a date ***date,*** returns a day number (the number of days since year 0).
- Example:
  - **SELECT** TO_DAYS('2017-09-01')



WEEK

- WEEK(date[, mode]): returns the week number for date
- The mode (next slide) enable to specify whether the week starts on Sunday or Monday and the return value should be in the range of 0-53 or 1-53
- Example:
  - select week('2013-01-01') returns 0
  - select week('2013-01-01',1) returns 1
  - select week('2013-01-01',7) returns 53

Week mode

Mode	1st day of week	Range	Week 1 is the first week
0	Sunday	0-53	With a Sunday in this year
1	Monday	0-53	With more than 3 days in this year
2	Sunday	1-53	With a Sunday in this year
3	Monday	1-53	With more than 3 days in this year
4	Sunday	0-53	With more than 3 days in this year
5	Monday	0-53	With a Monday in this year
6	Sunday	1-53	With more than 3 days in this year
7	Monday	1-53	With a Monday in this year

WEEKOFYEAR

- WEEKOFYEAR(date): returns the calendar week of the date as a number in the range 1 to 53.
- Equivalent to WEEK(date,3)
- EXAMPLE:
  - select weekofyear('2013-01-28')

YEARWEEK

- YEARWEEK(date[,mode]): returns year and week of a date argument
- Example:
  - select yearweek('2012-01-01')
  - select yearweek('2013-01-01')

WEEKDAY

- WEEKDAY(date): returns the weekday index for date
- 0: Monday, 1: Tuesday, …, 6:Sunday
- Example:
  - select weekday('2013-09-04')

UNIX_TIMESTAMP

- UNIX_TIMESTAMP(): returns a Unix timestamp (seconds since '1970-01-01 00:00:00' UTC) as an unsigned integer
- UNIX_TIMESTAMP(date): returns the value of the argument as seconds since '1970-01-01 00:00:00' UTC

GET_FORMAT

- GET_FORMAT({date|time|datetime},{'EUR'|'USA'|'JIS'|'ISO'|'INTERNAL'})
- Returns a format string
- The function is useful in combination with the DATE_FORMAT() and the STR_TO_DATE() functions

Function Call	Result
GET_FORMAT(DATE,'USA')	'%m.%d.%Y'
GET_FORMAT(DATE,'EUR')	'%d.%m.%Y'
GET_FORMAT(DATE,'INTERNAL')	'%Y%m%d'
GET_FORMAT(TIME,'USA')	'%h : %i : %s : %p'
GET_FORMAT(TIME,'EUR')	'%H : %i %s'
GET_FORMAT(TIME,'INTERNAL')	'%H%i%s'
GET_FORMAT(DATETIME,'USA')	'%Y-%m-%d %H.%i.%s'
GET_FORMAT(DATETIME,'INTERNAL')	%Y%m%d%H%i%s'

- EXAMPLE:
  - **SELECT** DATE_FORMAT('2013-01-08', GET_FORMAT(DATE, 'USA'))

  - **SELECT** STR_TO_DATE('01.08.2013', GET_FORMAT(DATE, 'USA'))