

Description of the Date type:

In SQL DATE values describe a particular year/month/day, in the form YYYY-MM-DD. For example, DATE '2013-01-01'. Date types do not have a time of day component.

The range of values supported for the Date type will be 0000-01-01 to 9999-12-31, dependent on support by the primitive Java Date type.

Representation of Date values

Writable Type

The DateWritable class uses an int representation, representing the day offset from a defined epoch value of 1970-01-01. This results in a smaller value required to represent the Date value compared to a representation based on seconds or milliseconds since the epoch, since we do not need to worry about the time of day for Date types. The range of dates that we can represent with such an integer representation would be +/- 2 billion days, or 5.8M years, which should be more than sufficient for the YYYY-MM-DD format that is used for date values.

There is a fairly limited number of usable operations that can be done with this integer representation - really just comparison operations, and adding/subtracting days. That should still be enough to do many of the common operations that would be done on Date types (group by, order by), without having to convert from the integer representation. For more powerful date operations, such as generating the string value, converting to/from other types, or applying date calculations, the integer representation in the DateWritable would need to be converted into some kind of date value which supports those operations.

Primitive Java Type

The primitive Java type for Date uses the java.sql.Date class. The java.sql.Date object has an internal representation in milliseconds since the epoch, so when converting from a DateWritable to a Java primitive value, the Date value will be set to a milliseconds value corresponding to midnight for the year/month/day, in the local timezone.

Operations on Date types

Date types support comparison operations, and type casting.

UDFs with Date support

There are a number of existing UDFs in Hive which support additional date-related functionality, and these will be compatible with the Date type as well.

```
date_add()  
date_sub()  
datediff()  
day()  
dayofmonth()  
from_utc_timestamp()
```

month()
to_date()
to_utc_timestamp()
unix_timestamp()
week_of_year()
year()

Aggregate Functions with Date support

max()
min()

Compatibility with other types:

Date types can only be converted to/from Date, Timestamp, or String types. There are UDFs available that the user can use to convert Date into numeric values.

Valid casts to/from date type	Result
cast(date as date)	Same date value
cast(timestamp as date)	The year/month/day of the timestamp is determined, based on the local timezone, and returned as a date value.
cast(string as date)	If the string is in the form 'YYYY-MM-DD', then a date value corresponding to that year/month day is returned. If the string value does not match this format then NULL is returned.
cast(date as timestamp)	A timestamp value is generated corresponding to midnight of the year/month/day of the date value, based on the local timezone.
cast(date as string)	The year/month/day represented by the Date is formatted as a string in the form 'YYYY-MM-DD'.