



Hive Functions Cheat-sheet, by Qubole

How to create and use Hive Functions, Listing of Built-In Functions that are supported in Hive

Hive Function Meta commands

- SHOW FUNCTIONS– lists Hive functions and operators
- DESCRIBE FUNCTION [function name]– displays short description of the function
- DESCRIBE FUNCTION EXTENDED [function name]– access extended description of the function

Types of Hive Functions

- UDF– is a function that takes one or more columns from a row as argument and returns a single value or object. Eg: concat(col1, col2)
- UDAF– aggregates column values in multiple rows and returns a single value. Eg: sum(c1)
- UDTF— takes zero or more inputs and produces multiple columns or rows of output. Eg: explode()
- Macros— a function that users other Hive functions.

How To Develop UDFs

```
package org.apache.hadoop.hive.contrib.udf.example;
import java.util.Date;
import java.text.SimpleDateFormat;
import org.apache.hadoop.hive.ql.exec.UDF;
@Description(name = "YourUDFName",
  value = "_FUNC_(InputDataType) - using the input datatype X argument, "+
  "returns YYYY",
  extended = "Example:\n"+
  "+ > SELECT _FUNC_(InputDataType) FROM tablename;")

public class YourUDFName extends UDF{
  ..
  public YourUDFName(InputDataType inputValue){
    ..
  }

  public String evaluate(InputDataType inputValue){
    ..
  }
}
```

How To Develop UDFs, GenericUDFs, UDAFs, and UDTFs

```
public class YourUDFName extends UDF{
public class YourGenericUDFName extends GenericUDF {}
public class YourGenericUDAFName extends AbstractGenericUDAFResolver {}
public class YourGenericUDTFName extends GenericUDTF {}
```

How To Deploy / Drop UDFs

At start of each session:
ADD JAR /full_path_to_jar/YourUDFName.jar;
CREATE TEMPORARY FUNCTION YourUDFName AS 'org.apache.hadoop.hive.contrib.udf.example.YourUDFName';

At the end of each session:

DROP TEMPORARY FUNCTION IF EXISTS YourUDFName;

Mathematical Functions

Return Type	Name (Signature)	Description
BIGINT	round(double a)	Returns the rounded BIGINT value of the double
DOUBLE	round(double a, int d)	Returns the double rounded to d decimal places
BIGINT	floor(double a)	Returns the maximum BIGINT value that is equal or less than the double
BIGINT	ceil(double a), ceiling(double a)	Returns the minimum BIGINT value that is equal or greater than the double
double	rand(), rand(int seed)	Returns a random number (that changes from row to row) that is distributed uniformly from 0 to 1. Specifying the seed will make sure the generated random number sequence is deterministic.
double	exp(double a)	Returns ea where e is the base of the natural logarithm
double	ln(double a)	Returns the natural logarithm of the argument
double	log10(double a)	Returns the base-10 logarithm of the argument
double	log2(double a)	Returns the base-2 logarithm of the argument
double	log(double base, double a)	Return the base "base" logarithm of the argument
double	pow(double a, double p), power(double a, double p)	Return ap
double	sqrt(double a)	Returns the square root of a
string	bin(BIGINT a)	Returns the number in binary format
string	hex(BIGINT a)	If the argument is an int, hex returns the number as a string in hex format. Otherwise if the number is a string, it converts each character into its hex representation and returns the resulting string.
string	unhex(string a)	Inverse of hex. Interprets each pair of characters as a hexadecimal number and converts to the character represented by the number.
string	conv(BIGINT num, int from_base, int to_base), conv(STRING num, int from_base, int to_base)	Converts a number from a given base to another
double	abs(double a)	Returns the absolute value
int double	pmod(int a, int b) pmod(double a, double b)	Returns the positive value of a mod b
double	sin(double a)	Returns the sine of a (a is in radians)
double	asin(double a)	Returns the arc sin of x if -1<=a<=1 or null otherwise
double	cos(double a)	Returns the cosine of a (a is in radians)
double	acos(double a)	Returns the arc cosine of x if -1<=a<=1 or null otherwise
double	tan(double a)	Returns the tangent of a (a is in radians)
double	atan(double a)	Returns the arctangent of a
double	degrees(double a)	Converts value of a from radians to degrees
double	radians(double a)	Converts value of a from degrees to radians
int double	positive(int a), positive(double a)	Returns a
int double	negative(int a), negative(double a)	Returns -a
float	sign(double a)	Returns the sign of a as '1.0' or '-1.0'
double	e()	Returns the value of e
double	pi()	Returns the value of pi

Date Functions

Return Type	Name (Signature)	Description
string	from_unixtime(bigint unixtime[, string format])	Converts the number of seconds from unix epoch (1970-01-01 00:00:00 UTC) to a string representing the timestamp of that moment in the current system time zone in the format of "1970-01-01 00:00:00"
bigint	unix_timestamp()	Gets current time stamp using the default time zone.
bigint	unix_timestamp(string date)	Converts time string in format yyyy-MM-dd HH:mm:ss to Unix time stamp, return 0 if fail: unix_timestamp('2009-03-20 11:30:01') = 1237573801
bigint	unix_timestamp(string date, string pattern)	Convert time string with given pattern to Unix time stamp, return 0 if fail: unix_timestamp('2009-03-20', 'yyyy-MM-dd') = 1237532400
string	to_date(string timestamp)	Returns the date part of a timestamp string: to_date("1970-01-01 00:00:00") = "1970-01-01"
int	year(string date)	Returns the year part of a date or a timestamp string: year("1970-01-01 00:00:00") = 1970, year("1970-01-01") = 1970
int	month(string date)	Returns the month part of a date or a timestamp string: month("1970-11-01 00:00:00") = 11, month("1970-11-01") = 11
int	day(string date) dayofmonth(date)	Return the day part of a date or a timestamp string: day("1970-11-01 00:00:00") = 1, day("1970-11-01") = 1
int	hour(string date)	Returns the hour of the timestamp: hour('2009-07-30 12:58:59') = 12, hour('12:58:59') = 12
int	minute(string date)	Returns the minute of the timestamp
int	second(string date)	Returns the second of the timestamp
int	weekofyear(string date)	Return the week number of a timestamp string: weekofyear("1970-11-01 00:00:00") = 44, weekofyear("1970-11-01") = 44
int	datediff(string enddate, string startdate)	Return the number of days from startdate to enddate: datediff('2009-03-01', '2009-02-27') = 2
string	date_add(string startdate, int days)	Add a number of days to startdate: date_add('2008-12-31', 1) = '2009-01-01'
string	date_sub(string startdate, int days)	Subtract a number of days to startdate: date_sub('2008-12-31', 1) = '2008-12-30'
timestamp	from_utc_timestamp(timestamp, string timezone)	Assumes given timestamp ist UTC and converts to given timezone (as of Hive 0.8.0)
timestamp	to_utc_timestamp(timestamp, string timezone)	Assumes given timestamp is in given timezone and converts to UTC (as of Hive 0.8.0)