Hive 函数

1.内置运算符

1.1 关系运算符

运算符	类型	说明
A = B	所有原始类型	如果 A 与 B 相等,返回 TRUE,否则返回 FALSE
A == B	无	失败，因为无效的语法。 SQL 使用”=”，不使用”==”。
A <> B	所有原始类型	如果 A 不等于 B 返回 TRUE,否则返回 FALSE。如果 A 或 B 值为”NULL”，结果返回”NULL”。
A < B	所有原始类型	如果 A 小于 B 返回 TRUE,否则返回 FALSE。如果 A 或 B 值为”NULL”，结果返回”NULL”。
A <= B	所有原始类型	如果 A 小于等于 B 返回 TRUE,否则返回 FALSE。如果 A 或 B 值为”NULL”，结果返回”NULL”。
A > B	所有	如果 A 大于 B 返回 TRUE,否则返回 FALSE。如果 A 或 B 值为”NULL”，结果返回”NULL”。

	原始类型	
A >= B	所有原始类型	如果 A 大于等于 B 返回 TRUE,否则返回 FALSE。如果 A 或 B 值为”NULL”，结果返回”NULL”。
A IS NULL	所有类型	如果 A 值为”NULL”，返回 TRUE,否则返回 FALSE
A IS NOT NULL	所有类型	如果 A 值不为”NULL”，返回 TRUE,否则返回 FALSE
A LIKE B	字符串	如 果 A 或 B 值为”NULL”，结果返回”NULL”。字符串 A 与 B 通过 sql 进行匹配，如果相符返回 TRUE，不符返回 FALSE。B 字符串中 的”_”代表任一字符，”%”则代表多个任意字符。例如：（‘foobar’ like ‘foo’)返回 FALSE，（ ‘foobar’ like ‘foo_ _ _’或者 ‘foobar’ like ‘foo%’)则返回 TURE
A RLIKE B	字符串	如 果 A 或 B 值为”NULL”，结果返回”NULL”。字符串 A 与 B 通过 java 进行匹配，如果相符返回 TRUE，不符返回 FALSE。例如：（ ‘foobar’ rlike ‘foo’）返回 FALSE，（’ foobar’ rlike ‘^f.*r$’ ）返回 TRUE。
A REGEXP B	字符串	与 RLIKE 相同。

1.2 算术运算符

运算符	类型	说明
A + B	所有数字类型	A 和 B 相加。结果的与操作数值有共同类型。例如每一个整数是一个浮点数，浮点数包含整数。所以，一个浮点数和一个整数相加结果也是一个浮点数。
A - B	所有数字类型	A 和 B 相减。结果的与操作数值有共同类型。

A * B	所有数字类型	A 和 B 相乘，结果的与操作数值有共同类型。需要说明的是，如果乘法造成溢出，将选择更高的类型。
A / B	所有数字类型	A 和 B 相除，结果是一个 double（双精度）类型的结果。
A % B	所有数字类型	A 除以 B 余数与操作数值有共同类型。
A & B	所有数字类型	运算符查看两个参数的二进制表示法的值，并执行按位”与”操作。两个表达式的一位均为 1 时，则结果的该位为 1。否则，结果的该位为 0。
A|B	所有数字类型	运算符查看两个参数的二进制表示法的值，并执行按位”或”操作。只要任一表达式的一位为 1，则结果的该位为 1。否则，结果的该位为 0。
A ^ B	所有数字类型	运算符查看两个参数的二进制表示法的值，并执行按位”异或”操作。当且仅当只有一个表达式的某位上为 1 时，结果的该位才为 1。否则结果的该位为 0。
~A	所有数字类型	对一个表达式执行按位”非”（取反）。

1.3 逻辑运算符

运算符	类型	说明
A AND B	布尔值	A 和 B 同时正确时,返回 TRUE,否则 FALSE。如果 A 或 B 值为 NULL，返回 NULL。
A && B	布尔值	与”A AND B”相同
A OR B	布尔值	A 或 B 正确,或两者同时正确返返回 TRUE,否则 FALSE。如果 A 和 B 值同时为 NULL，返回 NULL。
A | B	布尔值	与”A OR B”相同
NOT A	布尔值	如果 A 为 NULL 或错误的时候返回 TURE，否则返回 FALSE。
! A	布尔值	与”NOT A”相同

1.4 复杂类型函数

函数	类型	说明
map	(key1, value1, key2, value2, …)	通过指定的键/值对，创建一个 map。

struct	(val1, val2, val3, …)	通过指定的字段值，创建一个结构。结构字段名称将 COL1，COL2，…
array	(val1, val2, …)	通过指定的元素，创建一个数组。

1.5 对复杂类型函数操作

函数	类型	说明
A[n]	A 是一个数组，n 为 int 型	返回数组 A 的第 n 个元素，第一个元素的索引为 0。如果 A 数组为['foo','bar']，则 A[0]返回’ foo’ 和 A[1]返回”bar”。
M[key]	M 是 Map<K, V>，关键 K 型	返回关键值对应的值，例如 mapM 为 \{ ‘f’ -> ‘foo’, ‘b’ -> ‘bar’, ‘all’ -> ‘foobar’ \}，则 M['all'] 返回’ foobar’。
S.x	S 为 struct	返回结构 x 字符串在结构 S 中的存储位置。如 foobar \{int foo, int bar\} foobar.foo 的领域中存储的整数。

2.内置函数

2.1 数学函数

返回类型	函数	说明
BIGINT	round(double a)	四舍五入
DOUBLE	round(double a, int d)	小数部分 d 位之后数字四舍五入，例如 round(21.263,2),返回 21.26
BIGINT	floor(double a)	对给定数据进行向下舍入最接近的整数。例如 floor(21.2),返回 21。
BIGINT	ceil(double a), ceiling(double a)	将参数向上舍入为最接近的整数。例如 ceil(21.2),返回 23.
double	rand(), rand(int seed)	返回大于或等于 0 且小于 1 的平均分布随机数（依重新计算而变）
double	exp(double a)	返回 e 的 n 次方

double	ln(double a)	返回给定数值的自然对数
double	log10(double a)	返回给定数值的以 10 为底自然对数
double	log2(double a)	返回给定数值的以 2 为底自然对数
double	log(double base, double a)	返回给定底数及指数返回自然对数
double	pow(double a, double p) power(double a, double p)	返回某数的乘幂
double	sqrt(double a)	返回数值的平方根
string	bin(BIGINT a)	返回二进制格式
string	hex(BIGINT a) hex(string a)	将整数或字符转换为十六进制格式
string	unhex(string a)	十六进制字符转换由数字表示的字符。
string	conv(BIGINT num, int from_base, int to_base)	将 指定数值，由原来的度量体系转换为指定的试题体系。例如 CONV(‘a’,16,2),返回。参考：’ 1010′ http://dev.mysql.com/doc/refman/5.0/en/mathematical-functions.html#function_conv
double	abs(double a)	取绝对值
int double	pmod(int a, int b) pmod(double a, double b)	返回 a 除 b 的余数的绝对值
double	sin(double a)	返回给定角度的正弦值
double	asin(double a)	返回 x 的反正弦，即是 X。如果 X 是在-1 到 1 的正弦值，返回 NULL。
double	cos(double a)	返回余弦
double	acos(double a)	返回 X 的反余弦，即余弦是 X，，如果-1<= A <= 1，否则返回 null.

int double	positive(int a) positive(double a)	返回 A 的值，例如 positive(2)，返回 2。
int double	negative(int a) negative(double a)	返回 A 的相反数，例如 negative(2),返回-2。

2.2 收集函数

返回类型	函数	说明
int	size(Map<K.V>)	返回的 map 类型的元素的数量
int	size(Array<T>)	返回数组类型的元素数量

2.3 类型转换函数

返回类型	函数	说明
指定“type”	cast(expr as <type>)	类型转换。例如将字符”1″转换为整数:cast(’1′ as bigint)，如果转换失败返回 NULL。

2.4 日期函数

返回类型	函数	说明
string	from_unixtime(bigint unixtime[, string format])	UNIX_TIMESTAMP 参数表示返回一个值’YYYY- MM - DD HH：MM：SS’或 YYYYMMDDHHMMSS.uuuuuu 格式，这取决于是否是在一个字符串或数字语境中使用的功能。该值表示在当前的时区。
bigint	unix_timestamp()	如果不带参数的调用，返回一个 Unix 时间戳（从’1970- 01 - 0100:00:00′ 到现在的 UTC 秒数）为无符号整数。
bigint	unix_timestamp(string date)	指定日期参数调用 UNIX_TIMESTAMP（），它返回参数值’1970- 01 - 0100:00:00′ 到指定日期的秒数。
bigint	unix_timestamp(string date, string pattern)	指定时间输入格式，返回到 1970 年秒数：unix_timestamp(’2009-03-20′, ‘yyyy-MM-dd’) = 1237532400

string	to_date(string timestamp)	返回时间中的年月日： to_date(“1970-01-01 00:00:00″) = “1970-01-01″
string	to_dates(string date)	给定一个日期 date，返回一个天数（0 年以来的天数）
int	year(string date)	返回指定时间的年份，范围在 1000 到 9999，或为”零”日期的 0。
int	month(string date)	返回指定时间的月份，范围为 1 至 12 月，或 0 一个月的一部分，如’0000-00-00′ 或’2008-00-00′ 的日期。
int	day(string date) dayofmonth(date)	返回指定时间的日期
int	hour(string date)	返回指定时间的小时，范围为 0 到 23。
int	minute(string date)	返回指定时间的分钟，范围为 0 到 59。
int	second(string date)	返回指定时间的秒，范围为 0 到 59。
int	weekofyear(string date)	返回指定日期所在一年中的星期号，范围为 0 到 53。
int	datediff(string enddate, string startdate)	两个时间参数的日期之差。
int	date_add(string startdate, int days)	给定时间，在此基础上加上指定的时间段。
int	date_sub(string startdate, int days)	给定时间，在此基础上减去指定的时间段。

2.5 条件函数

返回类型	函数	说明
T	if(boolean testCondition, T valueTrue, T valueFalseOrNull)	判断是否满足条件，如果满足返回一个值，如果不满足则返回另一个值。
T	COALESCE(T v1, T v2, …)	返回一组数据中，第一个不为 NULL 的值，如果均为 NULL,返回 NULL。
T	CASE a WHEN b THEN c [WHEN d THEN e]* [ELSE f] END	当 a=b 时,返回 c；当 a=d 时，返回 e，否则返回 f。
T	CASE WHEN a THEN b [WHEN c THEN d]* [ELSE e] END	当值为 a 时返回 b,当值为 c 时返回 d。否则返回 e。

2.6 字符函数

返回类型	函数	说明
int	length(string A)	返回字符串的长度
string	reverse(string A)	返回倒序字符串
string	concat(string A, string B…)	连接多个字符串，合并为一个字符串，可以接受任意数量的输入字符串
string	concat_ws(string SEP, string A, string B…)	链接多个字符串，字符串之间以指定的分隔符分开。
string	substr(string A, int start) substring(string A, int start)	从文本字符串中指定的起始位置后的字符。
string	substr(string A, int start, int len) substring(string A, int start, int len)	从文本字符串中指定的位置指定长度的字符。
string	upper(string A) ucase(string A)	将文本字符串转换成字母全部大写形式
string	lower(string A) lcase(string A)	将文本字符串转换成字母全部小写形式
string	trim(string A)	删除字符串两端的空格，字符之间的空格保留
string	ltrim(string A)	删除字符串左边的空格，其他的空格保留
string	rtrim(string A)	删除字符串右边的空格，其他的空格保留
string	regexp_replace(string A, string B, string C)	字符串 A 中的 B 字符被 C 字符替代
string	regexp_extract(string subject, string pattern, int index)	通过下标返回正则表达式指定的部分。regexp_extract(‘foothebar’, ‘foo(.*?)(bar)’, 2) returns ‘bar.’
string	parse_url(string urlString, string partToExtract [,	返回 URL 指定的部分。parse_url(‘http://facebook.com/p

	string keyToExtract])	ath1/p.php?k1=v1&k2=v2#Ref1′, ‘HOST’) 返回：’ facebook.com’
string	get_json_object(string json_string, string path)	select a.timestamp, get_json_object(a.appevents, ‘$.eventid’), get_json_object(a.appenvets, ‘$.eventname’) from log a;
string	space(int n)	返回指定数量的空格
string	repeat(string str, int n)	重复 N 次字符串
int	ascii(string str)	返回字符串中首字符的数字值
string	lpad(string str, int len, string pad)	返回指定长度的字符串，给定字符串长度小于指定长度时，由指定字符从左侧填补。
string	rpad(string str, int len, string pad)	返回指定长度的字符串，给定字符串长度小于指定长度时，由指定字符从右侧填补。
array	split(string str, string pat)	将字符串转换为数组。
int	find_in_set(string str, string strList)	返回字符串 str 第一次在 strlist 出现的位置。如果任一参数为 NULL,返回 NULL；如果第一个参数包含逗号，返回 0。
array<array<string>>	sentences(string str, string lang, string locale)	将字符串中内容按语句分组，每个单词间以逗号分隔，最后返回数组。 例如 sentences(‘Hello there! How are you?’) 返回：((“Hello”, “there”), (“How”, “are”, “you”))
array<struct<string,double>>	ngrams(array<array<string>>, int N, int K, int pf)	SELECT ngrams(sentences(lower(tweet)), 2, 100 [, 1000]) FROM twitter;
array<struct<string,double>>	context_ngrams(array<array<string>>, array<string>, int K, int pf)	SELECT context_ngrams(sentences(lower(tweet)), array(null,null), 100, [, 1000]) FROM twitter;

3.内置的聚合函数（UDAF）

返回类型	函数	说明

bigint	count(*) , count(expr), count(DISTINCT expr[, expr_., expr_.])	返回记录条数。
double	sum(col), sum(DISTINCT col)	求和
double	avg(col), avg(DISTINCT col)	求平均值
double	min(col)	返回指定列中最小值
double	max(col)	返回指定列中最大值
double	var_pop(col)	返回指定列的方差
double	var_samp(col)	返回指定列的样本方差
double	stddev_pop(col)	返回指定列的偏差
double	stddev_samp(col)	返回指定列的样本偏差
double	covar_pop(col1, col2)	两列数值协方差
double	covar_samp(col1, col2)	两列数值样本协方差
double	corr(col1, col2)	返回两列数值的相关系数
double	percentile(col, p)	返回数值区域的百分比数值点。0<=P<=1,否则返回NULL,不支持浮点型数值。
array<double>	percentile(col, array(p~1,,\ [, p,,2,,]…))	返回数值区域的一组百分比值分别对应的数值点。0<=P<=1,否则返回NULL,不支持浮点型数值。
double	percentile_approx(col, p[, B])	Returns an approximate pˆthˆ percentile of a numeric column (including floating point

		types) in the group. The B parameter controls approximation accuracy at the cost of memory. Higher values yield better approximations, and the default is 10,000. When the number of distinct values in col is smaller than B, this gives an exact percentile value.
array<double>	percentile_approx(col, array(p~1,, [, p,,2_]…) [, B])	Same as above, but accepts and returns an array of percentile values instead of a single one.
array<struct\{ ‘x’ ,’y’ \}>	histogram_numeric(col, b)	Computes a histogram of a numeric column in the group using b non-uniformly spaced bins. The output is an array of size b of double-valued (x,y) coordinates that represent the bin centers and heights
array	collect_set(col)	返回无重复记录

4.内置表生成函数（UDTF）

返回类型	函数	说明
数组	explode(array<TYPE> a)	数组一条记录中有多个参数，将参数拆分，每个参数生成一列。
	json_tuple	get_json_object 语句：select a.timestamp, get_json_object(a.appevents, ‘$.eventid’), get_json_object(a.appenvets, ‘$.eventname’) from log a; json_tuple 语句: select a.timestamp, b.* from log a lateral view json_tuple(a.appevent, ‘eventid’, ‘eventname’) b as f1, f2

5.自定义函数

自定义函数包括三种 UDF、UDAF、UDTF

UDF(User-Defined-Function) 一进一出

UDAF(User- Defined Aggregation Funcation) 聚集函数，多进一出。Count/max/min

UDTF(User-Defined Table-Generating Functions) 一进多出，如 lateral view explore()

使用方式 ： 在 HIVE 会话中 add 自定义函数的 jar 文件，然后创建 function 继而使用函数

5.1 UDF 开发

1、UDF 函数可以直接应用于 select 语句，对查询结构做格式化处理后，再输出内容。

2、编写 UDF 函数的时候需要注意一下几点：

a）自定义 UDF 需要继承 org.apache.hadoop.hive.ql.UDF。

b）需要实现 evaluate 函数，evaluate 函数支持重载。

3、步骤

a）把程序打包放到目标机器上去；

b）进入 hive 客户端，添加 jar 包：hive>add jar /run/jar/udf_test.jar;

c）创建临时函数：hive>CREATE TEMPORARY FUNCTION add_example AS 'hive.udf.Add';

d）查询 HQL 语句：

```
SELECT add_example(8, 9) FROM scores;

SELECT add_example(scores.math, scores.art) FROM scores;

SELECT add_example(6, 7, 8, 6.8) FROM scores;
```

e）销毁临时函数：hive> DROP TEMPORARY FUNCTION add_example;

5.2 UDAF 自定义集函数

多行进一行出，如 sum()、min()，用在 group by 时

1.必须继承

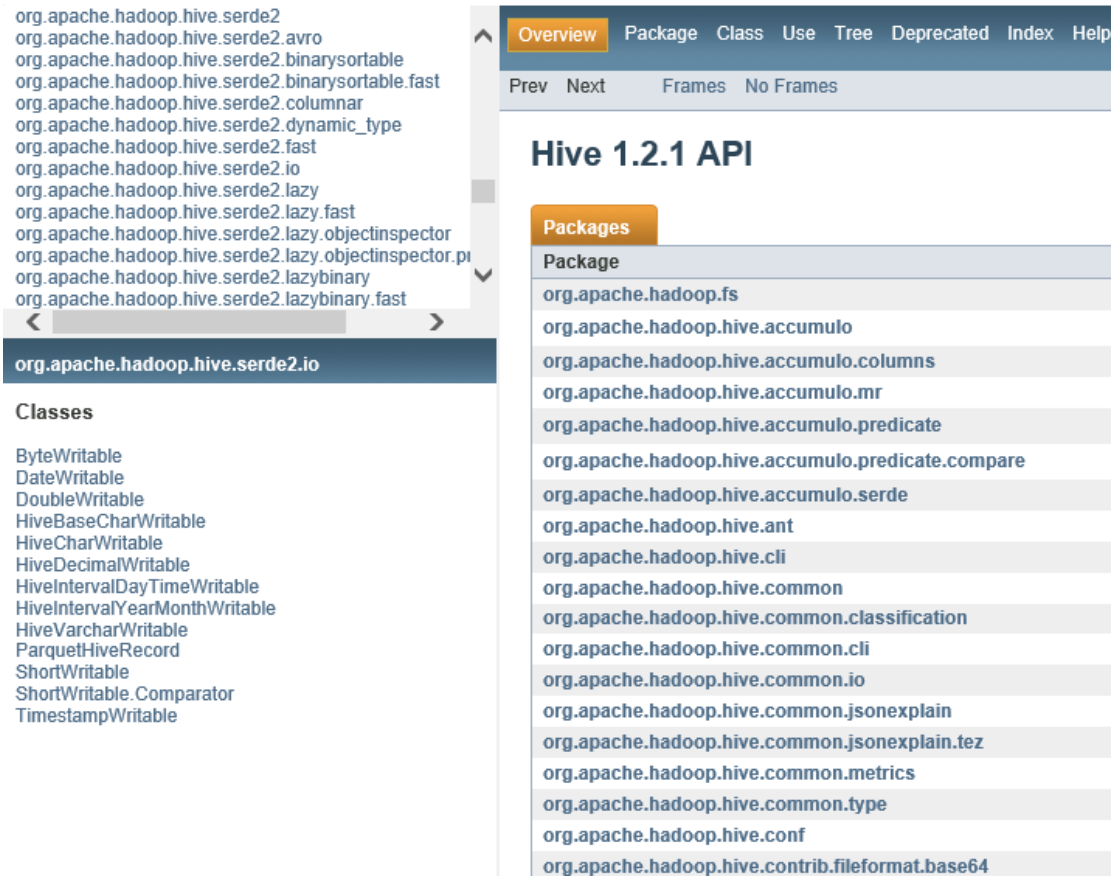
- org.apache.hadoop.hive.ql.exec.UDAF(函数类继承)
- org.apache.hadoop.hive.ql.exec.UDAFEvaluator(内部类 Evaluator 实现 UDAFEvaluator 接口)

2.Evaluator 需要实现 init、iterate、terminatePartial、merge、terminate 这几个函数

- init():类似于构造函数，用于 UDAF 的初始化
- iterate():接收传入的参数，并进行内部的轮转，返回 boolean
- terminatePartial():无参数，其为 iterate 函数轮转结束后，返回轮转数据，类似于 hadoop 的 Combiner
- merge():接收 terminatePartial 的返回结果，进行数据 merge 操作，其返回类型为 boolean
- terminate():返回最终的聚集函数结果

- 开发一个功能同：
- Oracle 的 wm_concat()函数
- Mysql 的 group_concat()

Hive UDF 的数据类型：


org.apache.hadoop.hive.serde2
org.apache.hadoop.hive.serde2.avro
org.apache.hadoop.hive.serde2.binarysortable
org.apache.hadoop.hive.serde2.binarysortable.fast
org.apache.hadoop.hive.serde2.columnar
org.apache.hadoop.hive.serde2.dynamic_type
org.apache.hadoop.hive.serde2.fast
org.apache.hadoop.hive.serde2.io
org.apache.hadoop.hive.serde2.lazy
org.apache.hadoop.hive.serde2.lazy.fast
org.apache.hadoop.hive.serde2.lazy.objectinspector
org.apache.hadoop.hive.serde2.lazybinary
org.apache.hadoop.hive.serde2.lazybinary.fast
org.apache.hadoop.hive.serde2.io
Classes
ByteWritable
DateWritable
DoubleWritable
HiveBaseCharWritable
HiveCharWritable
HiveDecimalWritable
HiveIntervalDayTimeWritable
HiveIntervalYearMonthWritable
HiveVarcharWritable
ParquetHiveRecord
ShortWritable
ShortWritable.Comparator
TimestampWritable
Overview
Package
Class
Use
Tree
Deprecated
Index
Help
Prev
Next
Frames
No Frames
Hive 1.2.1 API
Packages
Package
org.apache.hadoop.fs
org.apache.hadoop.hive.accumulo
org.apache.hadoop.hive.accumulo.columns
org.apache.hadoop.hive.accumulo.mr
org.apache.hadoop.hive.accumulo.predicate
org.apache.hadoop.hive.accumulo.predicate.compare
org.apache.hadoop.hive.accumulo.serde
org.apache.hadoop.hive.ant
org.apache.hadoop.hive.cli
org.apache.hadoop.hive.common
org.apache.hadoop.hive.common.classification
org.apache.hadoop.hive.common.cli
org.apache.hadoop.hive.common.io
org.apache.hadoop.hive.common.jsonexplain
org.apache.hadoop.hive.common.jsonexplain.tez
org.apache.hadoop.hive.common.metrics
org.apache.hadoop.hive.common.type
org.apache.hadoop.hive.conf
org.apache.hadoop.hive.contrib.fileformat.base64