**Hive SQL**

**1.  Hive在HDFS上的默认存储路径**

Hive的数据都是存储在HDFS上的，默认有一个根目录，在hive-site.xml中，由参数 hive.metastore.warehouse.dir 指定。默认值为/user/hive/warehouse.

**2.  Hive中的数据库(Database)**

* 进入Hive命令行，执行 show databases; 命令，可以列出hive中的所有数据库，默认有一个default数据库，进入Hive-Cli之后，即到default数据库下。
* 使用 use databasename; 可以切换到某个数据库下，同mysql；

**3.  Hive中的表(Table)**

1）查看所有的表：show tables;

2）内部表和外部表：

Hive中的表分为内部表(MANAGED\_TABLE)和外部表(EXTERNAL\_TABLE)。

内部表和外部表最大的区别：

内部表DROP时候 会删除 HDFS上的数据;

外部表DROP时候 不会删除 HDFS上的数据;

内部表适用场景：

Hive中间表、结果表、一般不需要从外部（如本地文件、HDFS上load数据）的情况。

外部表适用场景：

源表，需要定期将外部数据映射到表中。

我们的使用场景：

每天将收集到的网站日志定期流入HDFS文本文件，一天一个目录；

在Hive中建立外部表作为源表，通过添加分区的方式，将每天HDFS上的原始日志映射到外部表的天分区中；

在外部表（原始日志表）的基础上做大量的统计分析，用到的中间表、结果表使用内部表存储，数据通过SELECT+INSERT进入内部表。

CREATE EXTERNAL TABLE t\_lxw1234 (

id INT,

ip STRING COMMENT ‘访问者IP’,

avg\_view\_depth DECIMAL(5,1),

bounce\_rate D

ECIMAL(6,5)

) COMMENT ‘lxw的大数据田地-lxw1234.com’

PARTITIONED BY (day STRING)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ‘,’

STORED AS textfile

LOCATION ‘hdfs://cdh5/tmp/lxw1234/';

关键字 EXTERNAL ：

表示该表为外部表，如果不指定EXTERNAL关键字，则表示内部表

关键字 COMMENT

为表和列添加注释

关键字 PARTITIONED BY

表示该表为分区表，分区字段为day,类型为string

关键字 ROW FORMAT DELIMITED

指定表的分隔符，通常后面要与以下关键字连用：

FIELDS TERMINATED BY ‘,’ //指定每行中字段分隔符为逗号

LINES TERMINATED BY ‘\n’ //指定行分隔符

COLLECTION ITEMS TERMINATED BY ‘,’ //指定集合中元素之间的分隔符

MAP KEYS TERMINATED BY ‘:’ //指定数据中Map类型的Key与Value之间的分隔符

count(\*)和count(字段名) 基本结果是一样的

但是一种情况例外，就是当某字段名下边的数据有null值的时候，不计入这个count中，\*则全部列入count中。

row\_number() OVER (PARTITION BY COL1 ORDER BY COL2) 表示根据COL1分组，在分组内部根据 COL2排序，而此函数计算的值就表示每组内部排序后的顺序编号（组内连续的唯一的)

1 无参数调用：UNIX\_TIMESTAMP()

返回值：自'1970-01-01 00:00:00'的到当前时间的秒数差

例子：SELECT UNIX\_TIMESTAMP() => 1339123415

2 有参数调用：UNIX\_TIMESTAMP(date)

其中date可以是一个DATE字符串，一个DATETIME字符串，一个TIMESTAMP或者一个当地时间的YYMMDD或YYYMMDD格式的数字

返回值：自'1970-01-01 00:00:00'与指定时间的秒数差

举例说明：

DATE字符串格式：（日期类型）

SELECT UNIX\_TIMESTAMP(‘2012-06-08’) => 1339084800

SELECT UNIX\_TIMESTAMP(CURRENT\_DATE()) =>1339084800

注：CURRENT\_DATE ()的返回值是一个DATE字符串格式

以下几种格式返回的结果相同：

SELECT UNIX\_TIMESTAMP('20120608');

SELECT UNIX\_TIMESTAMP('2012-6-8');

SELECT UNIX\_TIMESTAMP('2012-06-08');

结果都是：1339084800

### UNIX时间戳转日期函数: from\_unixtime

|  |
| --- |
| 语法: from\_unixtime(bigint unixtime[, string format])  返回值: string  说明: 转化UNIX时间戳（从1970-01-01 00:00:00 UTC到指定时间的秒数）到当前时区的时间格式  举例：  hive> select from\_unixtime(1323308943,’yyyyMMdd’) from dual;  20111208 |

### 获取当前UNIX时间戳函数: unix\_timestamp

|  |
| --- |
| 语法: unix\_timestamp()  返回值: bigint  说明: 获得当前时区的UNIX时间戳  举例：  hive> select unix\_timestamp() from dual;  1323309615 |

### 日期转UNIX时间戳函数: unix\_timestamp

|  |
| --- |
| 语法: unix\_timestamp(string date)  返回值: bigint  说明: 转换格式为“yyyy-MM-dd HH:mm:ss“的日期到UNIX时间戳。如果转化失败，则返回0。  举例：  hive> select unix\_timestamp(’2011-12-07 13:01:03′) from dual;  1323234063 |

### 指定格式日期转UNIX时间戳函数: unix\_timestamp

|  |
| --- |
| 语法: unix\_timestamp(string date, string pattern)  返回值: bigint  说明: 转换pattern格式的日期到UNIX时间戳。如果转化失败，则返回0。  举例：  hive> select unix\_timestamp(’20111207 13:01:03′,’yyyyMMdd HH:mm:ss’) from dual;  1323234063 |

### 日期时间转日期函数: to\_date

|  |
| --- |
| 语法: to\_date(string timestamp)  返回值: string  说明: 返回日期时间字段中的日期部分。  举例：  hive> select to\_date(’2011-12-08 10:03:01′) from dual;  2011-12-08 |

### 日期转年函数: year

|  |
| --- |
| 语法: year(string date)  返回值: int  说明: 返回日期中的年。  举例：  hive> select year(’2011-12-08 10:03:01′) from dual;  2011  hive> select year(’2012-12-08′) from dual;  2012 |

### 日期转月函数: month

|  |
| --- |
| 语法: month (string date)  返回值: int  说明: 返回日期中的月份。  举例：  hive> select month(’2011-12-08 10:03:01′) from dual;  12  hive> select month(’2011-08-08′) from dual;  8 |

### 日期转天函数: day

|  |
| --- |
| 语法: day (string date)  返回值: int  说明: 返回日期中的天。  举例：  hive> select day(’2011-12-08 10:03:01′) from dual;  8  hive> select day(’2011-12-24′) from dual;  24 |

### 日期转小时函数: hour

|  |
| --- |
| 语法: hour (string date)  返回值: int  说明: 返回日期中的小时。  举例：  hive> select hour(’2011-12-08 10:03:01′) from dual;  10 |

### 日期转分钟函数: minute

|  |
| --- |
| 语法: minute (string date)  返回值: int  说明: 返回日期中的分钟。  举例：  hive> select minute(’2011-12-08 10:03:01′) from dual;  3 |

### 日期转秒函数: second

|  |
| --- |
| 语法: second (string date)  返回值: int  说明: 返回日期中的秒。  举例：  hive> select second(’2011-12-08 10:03:01′) from dual;  1 |

### 日期转周函数: weekofyear

|  |
| --- |
| 语法: weekofyear (string date)  返回值: int  说明: 返回日期在当前的周数。  举例：  hive> select weekofyear(’2011-12-08 10:03:01′) from dual;  49 |

### 日期比较函数: datediff

|  |
| --- |
| 语法: datediff(string enddate, string startdate)  返回值: int  说明: 返回结束日期减去开始日期的天数。  举例：  hive> select datediff(’2012-12-08′,’2012-05-09′) from dual;  213 |

### 日期增加函数: date\_add

|  |
| --- |
| 语法: date\_add(string startdate, int days)  返回值: string  说明: 返回开始日期startdate增加days天后的日期。  举例：  hive> select date\_add(’2012-12-08′,10) from dual;  2012-12-18 |

### 日期减少函数: date\_sub

|  |
| --- |
| 语法: date\_sub (string startdate, int days)  返回值: string  说明: 返回开始日期startdate减少days天后的日期。  举例：  hive> select date\_sub(’2012-12-08′,10) from dual;  2012-11-28 |

# 数据类型

## 1.基本的数据类型

|  |  |  |
| --- | --- | --- |
| 数据类型 | 字节 | 范围 |
| TINYINT | 1字节有符号整数 | -128-127 |
| SMALLINT | 2字节有符号整数 | -32768-32767 |
| INT | 4字节有符号整数 | -2,147,483,648到2,147,483,647 |
| BIGINT | 8字节有符号整数 | -9,223,372,036,854,775,808到9,223,372,036,854,775,807 |
| FLOAT | 4字节单精度浮点数 |  |
| DOUBLE/DOUBLE PRECISION | 8字节单精度浮点数 |  |
| DECIMAL | 在Hive 0.11.0中引入了38位精度 | Hive 0.13.0引入了用户定义的精度和尺度 |

新增的数据类型TIMESTAMP可以是整数、浮点数、字符串。整数是距离Unix新纪元时间1970.1.1 00:00:00的秒数。浮点数精确到纳秒，字符串则是YYYY-MM-DD hh:mm:ss.fffffffff；

BINARY字节数组。

如果对不同类型的值做对比，hive会隐形将比较小的那个类型转换成大的数据类型；

如果用户需要强转：cast(s AS INT)；

## 2.集合数据类型

## 1）struct和c类似STRUCT｛first STRING,last STRING｝，字段名.first就可以引用第一个元素。

## 2）MAP：可以通过字段名[’first’]来获取。

## 3) ARRAY:数组Array(‘JOHN’,’Doe’)通过数组名[1]来进行应用；

# Hive QL：数据定义

## 1.Hive中的数据库本质上是表的一个目录或者命名空间。Hive会为每个数据库创建一个目录。数据库中的表将会以这个数据库目录的子目录形式存储。除了default数据库的表。因为这个数据库本身就没有自己的目录。数据库所在目录位于属性hive.metastore.warehouse.dir所指定的顶层目录之后。默认配置是/user/hive/warehouse；数据库的文件目录是以.db结尾的。

## DESCRIB DATABASE 数据库名；会显示文件目录。

## 默认情况，Hive不允许用户删除一个包含有表的数据库，要么先删除数据库中的表，然后再删除数据库，或者DROP DATABASE IF EXISTS 数据库名 CASCADE；

## 2.管理表/内部表：删除表时，Hive也会删除这个表的数据

## 外部表：数据源在外部，通过EXTERNAL来声明，通过LOCATION ‘/…’来告诉hive数据位于哪个路径下。删除的时候不会删除数据，只会删除元数据。

## 

## 3.修改表：大多数表属性可以通过ALTER TABLE语句进行修改。这种操作会修改元数据但不会修改数据本身。

## ALTER TABLE old\_name RENAME TO new\_name；表重命名

## ALTER TABLE table\_name ADD IF NOT EXISTS PARTITION(year=2001) LOCATION '/log/2001'；增加分区

## ALTER TABLE table\_name DROP IF EXISTS PARTITION(year=2001)删除分区

## ALTER TABLE table\_name ADD COLUMNS(columns\_name type)

# 三、HiveQL：数据操作

## 1.向管理表中装载数据：

## Load data local inpath ‘…’ overwrite into table table\_name partition (ds);

## 如果分区目录不存在，这个命令会先创建分区目录，然后再将数据拷贝到该目录下。

## load data local inpath 'a.txt' into table a;

## overwrite的作用是覆盖；

## 2.通过查询语句向表中插入数据

## insert语句允许用户通过查询语句向目标表中插入数据。

## Insert overwrite table table\_name partition(ds) select \* from ();

## 

## 3.单个查询语句中创建表并加载数据

## Create table table\_name as select name,salary,address from t\_user where …;

## 不能用于外部表;

## 4.导出数据

## <http://blog.csdn.net/lifuxiangcaohui/article/details/40588929>

## 四、HiveQL：查询

## 1.select语句：

## 1）数组集合的引用：字段名[索引]；

## map集合的引用：字段名[键值]；

## struct集合的引用：字段名.属性名

## 

## 

## 

## 

## 

## CASE…WHEN…THEN…句式

## CASE

## WHEN … THEN '列的结果'

## …

## ELSE ‘’

## END AS 字段名