深入Hive

谭唐华

课程大纲

- Hive数据库操作
- Hive表的操作
- Hive数据类型
- Hive数据迁移
- Hive常见查询
- HiveUDF编程

Hive数据库操作

- 默认数据库"default"
- 使用#hive命令后，不使用hive>use <数据库名>，系统默认的数据库。可以显式使用hive> use default;
- 创建一个新库

Hive数据库操作

- 数据库操作：
  - hive > create database db_hive_01;
  - hive > show databases;
  - hive > use databases;
  - hive > desc database db_hive_01;
  - hive > drop database db_hive_01 cascade;

课程大纲

- Hive数据库操作
- Hive表的操作
- Hive数据类型
- Hive数据迁移
- Hive常见查询
- HiveUDF编程

创建表语法

```
## 创建表,指定表的类型,表的名称
CREATE [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.]table_name
  ### 表的列,包含列名称,列类型
  [(col_name data_type [COMMENT col_comment], ...)]
  ### 表的说明
  [COMMENT table_comment]
  ### 分区表,指定分区的列名称,列类型
  [PARTITIONED BY (col_name data_type [COMMENT col_comment], ...)]
  ### 数据文件中数据存储格式,每行分隔,每列分隔,数据文件类型
  [
   [ROW FORMAT row_format]
   [STORED AS file_format]
     | STORED BY 'storage.handler.class.name' [WITH SERDEPROPERTIES (...)]
  ]
  ### 数据文件存储在HDFS上的位置
  [LOCATION hdfs_path]
  ### 表的属性设置
  [TBLPROPERTIES (property_name=property_value, ...)]
  ### 子查询
  [AS select_statement];
```

创建表语法

```
## 创建表,指定表的类型,表的名称
CREATE [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.]table_name
  ### 指定表结构相同的表名称
  LIKE existing_table_or_view_name
  ### 数据文件存储在HDFS上的位置
  [LOCATION hdfs_path];

## 表中列的数据类型
data_type
  : primitive_type
  | array_type
  | map_type
  | struct_type
  | union_type
```

创建表语法

```
### 数据文件中数据存储格式
row_format
   : DELIMITED
     #### 每列数据分隔符
     [FIELDS TERMINATED BY char ]
     #### 每行数据分隔符
     [LINES TERMINATED BY char]

### 数据文件存储格式
file_format:
   : SEQUENCEFILE
   | TEXTFILE
   | RCFILE
   | ORC
   | PARQUET
   | INPUTFORMAT input_format_classname
     OUTPUTFORMAT output_format_classname
```

Hive表的操作

员工表：

```
create table emp(
empno int,
ename string,
job string,
mgr int,
hiredate string,
sal double,
comm double,
deptno int
)
row format delimited fields terminated by '\t';
```

部门表：

```
create table dept(
deptno int,
dname string,
loc string
)
row format delimited fields terminated by '\t';
```

- 加载数据

```
LOAD DATA [LOCAL] INPATH 'filepath'
[OVERWRITE] INTO TABLE tablename
[PARTITION (partcol1=val1, partcol2=val2 ...)]
```

  - LOCAL：从本地文件加载数据到hive表；否则从HDFS加载数据到hive表；
  - OVERWRITE：是否覆盖表中已有数据；
- load data local inpath '/emp.txt' overwrite into table emp;
- load data local inpath '/dept.txt' overwrite into table dept;

删除表与删除表数据

```
DROP TABLE [IF EXISTS] table_name [PURGE];
```

```
TRUNCATE TABLE table_name [PARTITION partition_spec];

partition_spec:
  : (partition_column = partition_col_value, partition_column = partition_col_value, ...)
```

Hive表的操作

- Hive两种表类型
  - 管理表（内部表）
  - 外部表
- 创建管理表
- hive>create table inner_table (key string);

外部表创建语法

```
CREATE EXTERNAL TABLE page_view(viewTime INT, userid BIGINT,
     page_url STRING, referrer_url STRING,
     ip STRING COMMENT 'IP Address of the User',
     country STRING COMMENT 'country of origination')
 COMMENT 'This is the staging page view table'
 ROW FORMAT DELIMITED FIELDS TERMINATED BY '\054'
 STORED AS TEXTFILE
 LOCATION '<hdfs_location>';
```

Hive表的操作

- 管理表
  - 内部表也称之为MANAGED_TABLE；
  - 默认存储在/user/hive/warehouse下，也可以通过location指定；
  - 删除表时，会删除表数据以及元数据；
- 外部表
  - 外部表称之为EXTERNAL_TABLE；
  - 在创建表时可以自己指定目录位置(LOCATION)；
  - 删除表时，只会删除元数据不会删除表数据；

Hive表的操作

- 分区表
  - 分区可以理解为分类，通过分类把不同类型的数据放到不同的目录下。
  - 分类的标准就是分区字段，可以一个，也可以多个。
  - 分区表的意义在于优化查询。查询时尽量利用分区字段。如果不使用分区字段，就会全部扫描。

分区表案例

```
hive (default)> create EXTERNAL table IF NOT EXISTS default.emp_partition(
              > empno int,
              > ename string,
              > job string,
              > mgr int,
              > hiredate string,
              > sal double,
              > comm double,
              > deptno int
              > )
              > partitioned by (month string,day string)
              > ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' ;
OK
Time taken: 0.183 seconds
```

分区表加载数据

- 加载数据：

load data local inpath ‘emp.txt’ into table emp_part partition(date=‘20160830’ ,hour=’20’) ;

- 查询数据：

select * from emp_part where date=‘20160830’ and hour=‘20’;

- 查看分区表：

show partitions student_part ;

- 删除分区表：

alter table student_part drop partition(date='20161030');

分区表特点

- 创建表的时候只需要指定分区字段,分区范围是在加载数据的时候才指定的
- 对大表数据进行分类存储,有效提高了数据质量方面的安全
- 使用分区表能大大的提高查询效率

创建表方式总结

- Hive中创建表的几种方式
  - 第一种：

```
create table db_1031.emp(
        empno int,
        ename string,
        job string,
        mgr int,
        hiredate string,
        sal double,
        comm double,
        deptno int
)
row format delimited fields terminated by '\t';
```

  - 第二种：--想把一张表的某几个字段抽取出来创建成一张新表

```
create table db_0831.emp_as as select * from emp ;
```

  - 第三种：--复制表结构

```
create table db_0831.emp_like like emp ;
```

课程大纲

- Hive数据库操作
- Hive表的操作
- Hive数据类型
- Hive数据迁移
- Hive常见查询
- HiveUDF编程

Hive数据类型

基本数据类型		
类型	描述	示例
TINYINT	1个字节（8位）有符号整数	1
SMALLINT	2字节（16位）有符号整数	1
INT	4字节（32位）有符号整数	1
BIGINT	8字节（64位）有符号整数	1
FLOAT	4字节（32位）单精度浮点数	1.0
DOUBLE	8字节（64位）双精度浮点数	1.0
BOOLEAN	true/false	true
STRING	字符串	‘xia’ ,” xia”

Hive数据类型

复杂数据类型		
类型	描述	示例
ARRAY	一组有序字段。字段的类型必须相同	Array(1,2)
MAP	一组无序的键/值对。键的类型必须是原子的，值可以是任何类型，同一个映射的键的类型必须相同，值得类型也必须相同	Map(‘a’,1,’b’,2)
STRUCT	一组命名的字段。字段类型可以不同	Struct(‘a’,1,1,0)

课程大纲

- Hive数据库操作
- Hive表的操作
- Hive数据类型
- Hive数据迁移
- Hive常见查询
- HiveUDF编程

表数据导入步骤

- 加载本地文件到hive表
- 加载hdfs文件到hive中
- 加载数据覆盖表中已有的数据
- 创建表时通过select加载
- 创建表通过insert加载
- 创建表的时候通过location指定加载

表数据导入步骤

- 1、加载本地文件到Hive表

  load data local inpath 'path/file' into table 表名称 ;

- 2、加载HDFS文件到Hive表

  load data inpath 'path/file' into table 表名称 ;

- 3、加载数据覆盖表中已有的数据

  load data local inpath 'path/file' overwrite into table 表名称 ;

- 4、创建表时通过select加载

  create table db_0831.emp_as as select * from emp ;

表数据导入步骤

- 5、用insert命令加载（先要创建好表，然后再写入数据

  应用场景：把用select命令分析的结果写入某　　个临时表

  append 追加写入　--默认

  overwrite 覆盖写入 --使用最多

  insert into table 表名 select * from emp ;

  insert overwrite table 表名 select * from emp ;

表数据导入步骤

- 6、创建表的时候通过location指定加载

  create table 表名(

  ……

)

  partitioned by　　--注意顺序

  row format ..

  location "" ;

Hive导出表的操作

- 表数据导出几种方式：
  - 1、insert..local directory导出到本地

    insert overwrite local diretory "path/" select ....

    >insert overwrite local directory '/opt/modules/mydata'

    > row format delimited fields terminated by '\t'

    > select * from student;
  - 2、insert..directory导出到HDFS

    insert overwrite directory "path/" select * from emp ;
  - 3、hive -e

    $ bin/hive -e "select ...." > /home/beifeng/emp.log
  - 4、sqoop工具

    Hive表数据 --> mysql表

数据导出四种方式

- 通过insert…directory导出
- 通过hadoop命令导出
- 通过hive shell命令 + 管道(hive -f/-e | sed/grep/awk > file)
- 使用sqoop

课程大纲

- Hive数据库操作
- Hive表的操作
- Hive数据类型
- Hive数据迁移
- Hive常见查询
- HiveUDF编程

Hive常见查询

- Select Syntax
  - WHERE Clause
  - ALL and DISTINCT Clauses
  - Partition Based Queries
  - HAVING Clause
  - LIMIT Clause
  - REGEX Column Specification
  - More Select Syntax
    - GROUP BY
    - SORT BY, ORDER BY, CLUSTER BY, DISTRIBUTE BY
    - JOIN
    - UNION
    - TABLESAMPLE
    - Subqueries
    - Virtual Columns
    - Operators and UDFs
    - LATERAL VIEW
    - Windowing, OVER, and Analytics

Select基本语法

```
SELECT [ALL | DISTINCT] select_expr, select_expr, ...
    FROM table_reference
    [WHERE where_condition]
    [GROUP BY col_list]
    [CLUSTER BY col_list
     | [DISTRIBUTE BY col_list] [SORT BY col_list]
    ]
    [LIMIT number]
```

select常见子句

- 全表查询、指定字段查询
- = />= / <= /between and /limit
- (not) in / is (not) null
- max/min/count/sum/avg
- group by / having
- join

- 去重
  - select distinct(deptno) from db_0831.emp ;
- 关联查询，聚合，分组查询
  - select b.dname,max(a.sal) from emp a inner join dept b on a.deptno = b.deptno group by a.deptno,b.dname having max(a.sal)>3000;
- 内链接、左链接、右链接

Hive中select新特性

- Hive中select新特性
  - Order By：全局排序，一个Reduce
  - Sort By：每个reduce内部进行排序，全局不是排序
  - Distribute By：类似MR中partition，进行分区，结合sort by使用

课程大纲

- Hive数据库操作
- Hive表的操作
- Hive数据类型
- Hive数据迁移
- Hive常见查询
- HiveUDF编程

HiveUDF编程

- 查看有哪些函数

  > show functions ;

- 查看具体某个函数的用法

  > desc function extended max ;

- 常用函数

  concat函数 --拼接字符串

  > select concat(empno,"_",ename) from emp;

  substr函数 --截取字符串

  > select substr(hiredate,1,4) from emp;

  day函数 --获取天

  > select day(hiredate) from emp;

  cast --类型转换

  > select cast(1472704474123/1000 as int) ;

HiveUDF编程

- Hive自带了一些函数，比如:max/min等，但是数量有限，自己可以通过自定义UDF来方便的扩展。
- 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。
- 编写UDF函数的时候需要注意以下几点：
  - 自定义UDF需继成org.apache.hadoop.hive.ql.UDF
  - 需要实现evaluate函数，evaluate函数支持重载
  - UDF必须要有返回类型，可以返回null,但是返回类型不能为void；
  - UDF中常用Text/LongWritable等类型，不推荐使用java类型；

环境准备

- 修改pom依赖包
  - <dependency>
  - <groupId>org.apache.hive</groupId>
  - <artifactId>hive-jdbc</artifactId>
  - <version>0.13.1</version>
  - </dependency>
  - <dependency>
  - <groupId>org.apache.hive</groupId>
  - <artifactId>hive-exec</artifactId>
  - <version>0.13.1</version>
  - </dependency>
- 上传repository.tar.gz,并解压到/home/hadoop/.m2/

  （使用hadoop开发mapreduce环境，无需再解压）

环境准备

- 修改maven的配置文件：/opt/modules/apache-maven-3.0.5/conf: settings.xml

```
<mirror>
        <id>nexus-osc</id>
        <mirrorOf>central</mirrorOf>
        <name>Nexus osc</name>
        <url>http://maven.oschina.net/content/groups/public/</url>
</mirror>
```

```
$ cp settings.xml /home/beifeng/.m2/
```

- 重新更新maven工程

HiveUDF编程

```
package org.apache.hadoop.udf;

import org.apache.commons.lang.StringUtils;
import org.apache.hadoop.hive.ql.exec.UDF;
import org.apache.hadoop.io.Text;

public class MyLower extends UDF {
	public Text evaluate(Text str){
		if(null == str){
			return null;
		}
		if(StringUtils.isBlank(str.toString())){
			return null;
		}
		return new Text(str.toString().toLowerCase());
	}
}
```

HiveUDF编程

- 实现步骤
  - 把程序打包放到目标机器上去；
  - 进入hive客户端，添加jar包：hive>add jar /run/jar/udf_test.jar;
  - 创建临时函数：hive>CREATE TEMPORARY FUNCTION add_example AS 'hive.udf.Add';
  - 查询HQL语句：
  - 销毁临时函数：hive> DROP TEMPORARY FUNCTION add_example;

注：UDF只能实现一进一出的操作，如果需要实现多进一出，则需要实现UDAF

HiveUDF编程

- UDF几种类型
  - UDF(User-Defined-Function)
    - 一进一出
  - UDAF(User-Defined Aggregation Funcation)
    - 聚集函数，多进一出；
    - 类似于：count/max/min
  - UDTF(User-Defined Table-Generating Functions)
    - 一进多出；
    - 如lateral view explore()

HiveUDF编程

- UDF函数日常应用中注意事项：
  - 以上方式添加的UDF函数是临时生效的,只要退出当前hive窗口就失效了
  - 一般我们不会把UDF函数设置为永久生效
  - 日常应用当某个job任务需要用到该函数的时候,我们才会去添加这个函数，也即也就是说每个函数都是跟着业务(job)走。

总结

- Hive数据库和数据表
- Hive表类型和数据类型
- Hive数据导入导出
- Hive分析查询语句
- 自定义UDF函数编程