

Hive Server2 与函数操作

- What?Why?How?



Hive Beeline

- 首先启动hiveserver2
- 要在hadoop的core-site.xml添加属性：
 - `hadoop.proxyuser.root.groups`
 - `hadoop.proxyuser.root.hosts`
- 执行命令或者重启hadoop集群
 - `bin/hdfs dfsadmin -fs hdfs://node01:8020 -refreshSuperUserGroupsConfiguration`
 - `bin/hdfs dfsadmin -fs hdfs://node02:8020 -refreshSuperUserGroupsConfiguration`



Hive Beeline

- Beeline
 - Beeline 要与HiveServer2配合使用
 - 服务端启动hiveserver2
 - 客户的通过beeline两种方式连接到hive
 - 1、 beeline -u jdbc:hive2://localhost:10000/default -n root
 - 2、 beeline
 - beeline> !connect jdbc:hive2://<host>:<port>/<db>;auth=noSasl root 123
 - 默认 用户名、密码不验证



Hive JDBC

- Hive JDBC运行方式
 - 服务端启动hiveserver2后，在java代码中通过调用hive的jdbc访问默认端口10000进行连接、访问



Hive JDBC

```
public class HiveJdbcClient {  
  
    private static String driverName = "org.apache.hive.jdbc.HiveDriver";  
  
    /**  
     * @param args  
     * @throws SQLException  
     */  
    public static void main(String[] args) throws SQLException {  
        try {  
            Class.forName(driverName);  
        } catch (ClassNotFoundException e) {  
            // TODO Auto-generated catch block  
            e.printStackTrace();  
            System.exit(1);  
        }  
        // replace "hive" here with the name of the user the queries should run  
        // as  
        Connection con = DriverManager.getConnection(  
            "jdbc:hive2://node3:10000/default", "root", "");  
  
        Statement stmt = con.createStatement();  
  
        String sql = "select * from psn1 limit 5";  
        ResultSet res = stmt.executeQuery(sql);  
        while (res.next()) {  
            System.out.println(res.getString(1) + "-" + res.getString("name"));  
        }  
    }  
}
```



Hive 函数

- Hive复合数据类型

Constructor Function	Operands	Description
map	(key1, value1, key2, value2, ...)	Creates a map with the given key/value pairs.
struct	(val1, val2, val3, ...)	Creates a struct with the given field values. Struct field names will be col1, col2,
named_struct	(name1, val1, name2, val2, ...)	Creates a struct with the given field names and values. (As of Hive 0.8.0.)
array	(val1, val2, ...)	Creates an array with the given elements.
create_union	(tag, val1, val2, ...)	Creates a union type with the value that is being pointed to by the tag parameter.



Hive 函数操作

- Hive操作复合类型

Operator	Operand types	Description
A[n]	A is an Array and n is an int	Returns the nth element in the array A. The first element has index 0. For example, if A is an array comprising of ['foo', 'bar'] then A[0] returns 'foo' and A[1] returns 'bar'.
M[key]	M is a Map<K, V> and key has type K	Returns the value corresponding to the key in the map. For example, if M is a map comprising of {'f' -> 'foo', 'b' -> 'bar', 'all' -> 'foobar'} then M['all'] returns 'foobar'.
S.x	S is a struct	Returns the x field of S. For example for the struct foobar {int foo, int bar}, foobar.foo returns the integer stored in the foo field of the struct.



Hive 内置函数

- Hive 内置函数
 - Date Functions
 - Conditional Functions
 - Misc. Functions



Hive 函数

- Hive 自定义函数
 - Hive的UDF开发只需要重构UDF类的evaluate函数即可。例：
 - package com.hrj.hive.udf;
 - import org.apache.hadoop.hive.ql.exec.UDF;
 - public class helloUDF extends UDF {
 - public String evaluate(String str) {
 - try {
 - return "HelloWorld " + str;
 - } catch (Exception e) {
 - return null;
 - }
 - }
 - }



Hive 函数

- Hive 自定义函数调用
 - 将该java文件编译成helloudf.jar
 - hive> add jar helloudf.jar;
 - hive> create temporary function helloworld as 'com.hrj.hive.udf.helloUDF';
 - hive> select helloworld(t.col1) from t limit 10;
 - hive> drop temporary function helloworld;
- 注意
 - 1.helloworld为临时的函数，所以每次进入hive都需要add jar以及create temporary操作
 - 2.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

