

第十一节、HIVE常用操作

一、常用函数

1. 查看数据库列表
2. 查看数据表列表
3. 创建表
4. 查看表信息
5. 数据导入
6. 连接查询
7. 排序
8. 散列操作
9. 查看可用函数列表

二、关系运算

1. 等值比较(=)
2. 不等值比较(<>)
3. 小于 (等于) 比较(<=)
4. 大于 (等于) 比较(>=)
5. 空值判断
6. LIKE比较
7. 正则匹配 (RLIKE)

三、数学运算

1. 四则运算
2. 取余运算 (%)
3. 按位与运算 (&)
4. 按位或运算 (|)
5. 位异或运算 (^)
6. 位取反操作 (~)

四、逻辑运算

1. 逻辑与 (AND)
2. 逻辑或 (OR)
3. 逻辑非 (NOT)

五、数值运算

1. 取整函数
2. 向下取整函数
3. 向上取整函数
4. 生成随机数
5. 自然指数函数
6. 对数函数
7. 幂函数
8. 平方根函数：sqrt ({value})
9. 立方根函数：cbrt ({value})
10. 进制函数
11. 绝对值函数：abs ({value})
12. 正取余函数
13. 三角函数

14. 弧度转角度函数：degrees ({value})
15. 角度转弧度函数：radians ({value})
16. 相反数函数：negative ({value})
17. 符号函数：sign ({value})
18. 常数函数
19. 最值函数
20. 阶乘函数：factorial ({value})

六、日期函数

1. 获取日期函数：unix_timestamp ()
2. 时间戳转换函数
3. 日期截取函数
4. 日期计算函数

七、条件函数

1. if函数
2. 非空查找函数
3. 条件判断函数

八、字符串函数

1. 字符串长度函数：length ({stringValue})
2. 字符串反转函数：reverse ({stringValue})
3. 字符串连接函数
4. 字符串截取函数
5. 大小写转换函数
6. 去空格函数
7. 正则表达式函数：
8. URL解析函数
9. json解析函数：get_json_object({jsonString},'\${jsonObjectKey}')
10. 字符串生成函数
11. 首字符ASCII码函数：ascii ({stringValue})
12. 字符串补足函数
13. 字符串分割函数：split ({stringValue} , {splitValue})
14. 字符串查找函数：find_in_set ({findString} , {stringValue})

九、集合统计函数

1. 计数函数
2. 求和函数
3. 平均值函数
4. 最值函数
5. 非空集合总体变量函数
6. 非空集合样本变量函数
7. 总体标准偏离函数
8. 样本标准偏离函数
9. 中位数函数
10. 直方图函数

十、复合类型函数

1. Map类型函数

- 2. Struct类型函数
- 3. Array类型函数
- 4. 类型转换函数

十一、用户自定义函数

- 1. 函数编写
- 2. 函数注册
- 3. 函数调用

第十一节、HIVE常用操作

一、常用函数

1. 查看数据库列表

```
1. show databases;
```

2. 查看数据表列表

```
1. show tables [like '*name*'];
```

3. 创建表

内部表和外部表区别：内部表即完全交给hive管理表，在创建时会把数据移动到数据仓库所在的路径，删除时会删除数据源文件。外部表即增加hive管理的数据文件，创建时需要记录数据所在的路径，不会移动数据源文件，删除时不会删除数据源文件

- 内部表创建

```
1. create table {tableName}(  
2.     {columnName} {columnType},  
3.     {columnName} {columnType}  
4. )  
5. [row format delimited fields terminated by '\t'];
```

- 外部表创建

```
1. create external table {tableName}(  
2.     {columnName} {columnType},  
3.     {columnName} {columnType}  
4. )  
5. [row format delimited fields terminated by '\t' location  
   '{HDFS_path}'];
```

- 分区表创建

```
1. create [external] table {tableName}(  
2.     {columnName} {columnType},  
3.     {columnName} {columnType}  
4. )  
5. partitioned by(  
6.     {columnName} {columnType},  
7.     {columnName} {columnType}  
8. )  
9. [row format delimited fields terminated by '\t' location  
   '{HDFS_path}'];
```

- 从查询结果创建

```
1. create table {tableName} as {selectSql};
```

默认分隔符为'\001'

4. 查看表信息

- 查看表结构

```
1. desc {tableName};  
2. {desc formatted}/{describe} {tableName};
```

- 查看分区信息

```
1. show partitions {tableName};
```

5. 数据导入

- 文件导入

```
1. load data inpath '{HDFSPath}' [overwrite] into table {tableName};
2. load data local inpath '{localPath}' [overwrite] into table {tableName};
```

- hive数据查询导入

```
1. insert [overwrite] into table {dstTableName} select {needColumns} from {srcTableName};
```

- Sqoop查询导入

```
1. sqoop import --connect {jdbcUrl} --username {databaseUserName} --password {databaseUserPwd} [--columns {columnName}] --table {srcTableName} --where '{conditionSql}' / --query '{querySql}' --hive-import --hive-database {dstDatabaseName} [--hive-overwrite] --hive-table {dstTableName} [--delete-target-dir] --target-dir {tempPath} -m 1
```

- 动态分区数据导入

```
1. set hive.exec.dynamic.partition=true;
2. set hive.exec.dynamic.partition.mode=nostrick;
3. insert [overwrite] into table {dstTableName} partition({partitionName}) select {needColumns},{partitionName} from {srcTableName};
```

6. 连接查询

- 内连接

常见的连接查询，也可以使用where实现

```
1. select {t1.columns},{t2.columns} from {tableName t1} inner join {tableName t2} on {t1.columnName = t2.columnName};
2. select {t1.columns},{t2.columns} from {tableName t1},{tableName t2} where {t1.columnName = t2.columnName};
```

- 左连接

将左边的表作为主表，右边的表作为从表，会将左边的表的元组全部取出

```
1. select {t1.columns},{t2.columns} from {tableName t1} left [outer] join
   {tableName t2} on {t1.columnName = t2.columnName};
```

- 右连接

将右边的表作为主表，左边的表作为从表，会将右边的表的元组全部取出

```
1. select {t1.columns},{t2.columns} from {tableName t1} right [outer] join
   {tableName t2} on {t1.columnName = t2.columnName};
```

- 全连接

会将左右两个表的数据全部取出，不管是否有匹配项

```
1. select {t1.columns},{t2.columns} from {tableName t1} full [outer] join
   {tableName t2} on {t1.columnName = t2.columnName};
```

7. 排序

使用order by关键字，默认升序

- 升序：ASC
- 降序：DESC

8. 散列操作

可以将文件拆分或者组合，默认关闭，值为-1

```
1. set mapred.reduce.tasks = n;
```

9. 查看可用函数列表

```
1. show functions;
```

```
1. desc function {functionName};
```

二、关系运算

时间的比较需要先使用to_date ({timeColumn}) 函数

1. 等值比较(=)

2. 不等值比较(<>)

3. 小于（等于）比较(<=)

4. 大于（等于）比较(>=)

5. 空值判断

- 空值判断：IS NULL
- 非空判断：IS NOT NULL

6. LIKE比较

如果比较的字段为NULL则返回NULL，如果匹配则返回TRUE，否则为FALSE

- 模糊查询：%代表任意数量的字符，_代表任意单个字符
- 否定比较：使用NOT关键字

7. 正则匹配（RLIKE）

如果比较的字段为NULL则返回NULL，如果匹配则返回TRUE，否则为FALSE

三、数学运算

取前n条数据，limit n (mysql语法)

1. 四则运算

最高精度为double，精确到小数点后16位

- 加法操作 (+)
- 减法操作 (-)
- 乘法操作 (*)
- 除法操作 (/)

2. 取余运算 (%)

返回除法运算的余数，结果的数值类型等于两数据最小父类型。

3. 按位与运算 (&)

4. 按位或运算 (|)

5. 位异或运算 (^)

相同为0，不同为1

6. 位取反操作 (~)

四、逻辑运算

如果结果为真，则返回TRUE，否则返回FALSE

1. 逻辑与 (AND)

2. 逻辑或 (OR)

3. 逻辑非 (NOT)

五、数值运算

1. 取整函数

- 四舍五入法：round ({value}[, precision])

1~4：舍，5~9：进，传入两个参数时可指定精度

- 银行家舍入法：bround ({value}[, precision])

1~4：舍，6~9：进，5->前一位是偶数：舍，前一位是奇数：进，传入两个参数时可指定精度

2. 向下取整函数

floor ({value})

3. 向上取整函数

ceil/ceiling ({value})

4. 生成随机数

rand ([{seed}])：返回一个0到1范围内的随机数，传入参数时可生成稳定的随机数

5. 自然指数函数

- 自然指数e的n次方： $\exp(\{n\})$

6. 对数函数

- 以10为底的对数函数： $\log_{10}(\{value\})$
- 以2为底的对数函数： $\log_2(\{value\})$
- 以e为底的对数函数： $\ln(\{value\})$
- 对数函数： $\log(\{base\}, \{value\})$

7. 幂函数

```
pow/power ( {base} , {exponent} )
```

8. 平方根函数： $\text{sqrt}(\{value\})$

9. 立方根函数： $\text{cbrt}(\{value\})$

10. 进制函数

- 转二进制函数： $\text{bin}(\{value\})$
- 转十六进制函数： $\text{hex}(\{value\})$
- 反转十六进制函数： $\text{unhex}(\{value\})$
- 进制转换函数： $\text{conv}(\{value\}, \{fromBase\}, \{toBase\})$

11. 绝对值函数： $\text{abs}(\{value\})$

12. 正取余函数

```
pmod ( {value} , {value} )
```

13. 三角函数

- 正弦函数：sin ({value})
- 反正弦函数：asin ({value})
- 余弦函数：cos ({value})
- 反余弦函数：acos ({value})
- 正切函数：tan ({value})
- 反正切函数：atan ({value})

14. 弧度转角度函数：degrees ({value})

15. 角度转弧度函数：radians ({value})

16. 相反数函数：negative ({value})

17. 符号函数：sign ({value})

18. 常数函数

- 自然指数e：pi ()
- 圆周率 π ：pie ()

19. 最值函数

- 最大值函数：greatest ({valueList})
- 最小值函数：least ({valueList})

20. 阶乘函数：factorial ({value})

六、日期函数

1. 获取日期函数：unix_timestamp ()

2. 时间戳转换函数

- UNIX时间戳转日期：from_unixtime ({unixTime}[, {formatString}])
- 日期转UNIX时间戳：unix_timestamp ({timeString}[, {formatString}])

3. 日期截取函数

- 返回日期部分：to_date ({timeString})
- 返回日期的年：year ({timeString})
- 返回日期的月：month ({timeString})
- 返回日期的天：day ({timeString})
- 返回日期的时：hour ({timeString})
- 返回日期的分：minute ({timeString})
- 返回日期的秒：second ({timeString})
- 返回日期的周：weekofyear ({timeString})

4. 日期计算函数

- 日期比较函数：datediff ({endDate}, {startDate})
- 日期增加函数：date_add ({startData}, {days})
- 日期减少函数：date_sub ({startData}, {days})

七、条件函数

1. if函数

if ({condition}, {trueValue}, {falseValue}) : 当条件为true时，返回trueValue的值，当条件为false时，返回falseValue的值

2. 非空查找函数

coalesce ({values}) : 返回参数中的第一个非空值，如果都为NULL，则返回NULL

3. 条件判断函数

case {value} when {compareValue} then {resultValue} else {resultValue} : 如果值相等则返回相应的值，否则返回else后的值

八、字符串函数

1. 字符串长度函数：length ({stringValue})

2. 字符串反转函数：reverse ({stringValue})

3. 字符串连接函数

- 无分隔符连接函数：concat ({stringValues})
- 分隔符连接函数：concat_ws ({separator}, {stringValues})

4. 字符串截取函数

- substr/substring ({stringValue}, {index})：当index为正数时，截取从index至结尾的字符串，当index为负数时，截取后index个字符，index的值不能超过字符串长度
- substr/substring ({stringValue}, {index}, {length})：截取从index开始，长度为length的字符，index为正数时，索引从左边开始，index为负数时，索引从右边开始

5. 大小写转换函数

- 转大写函数：upper/ucase ({stringValue})
- 转小写函数：lower/lcase ({stringValue})

6. 去空格函数

- 两边去空格函数：trim ({stringValue})
- 左边去空格函数：ltrim ({stringValue})
- 右边去空格函数：rtrim ({stringValue})

7. 正则表达式函数：

- 正则替换函数：regexp_replace ({stringValue}, {regexpString}, {replaceValue})
- 正则解析函数：regexp_extract ({stringValue}, {regexpString}, {index})

8. URL解析函数

```
parse_url ( {stringValue} , '{extractPart}' ) [ , '{extractKey}' ] )
```

- HOST : 获取主机名 (域名)
- PATH : 获取访问路径
- QUERY : 参数解析 , 需要配合extractKey一起使用
- REF : 获取锚点信息
- PROTOCOL : 获取网络协议
- FILE : 获取路径及参数信息

9. json解析函数 :

get_json_object({jsonString}, '\$. {jsonObjectKey}')

10. 字符串生成函数

- 空格字符串函数 : space (n)
- 重复字符串函数 : repeat ({stringValue} , n)

11. 首字符ASCII码函数 : ascii ({stringValue})

12. 字符串补足函数

将原字符串用指定的追加字符串补足为指定长度的字符串

- 左补足函数 : lpad ({stringValue} , {length} , {appendValue})
- 右补足函数 : rpad ({stringValue} , {length} , {appendValue})

13. 字符串分割函数 : split ({stringValue} , {splitValue})

14. 字符串查找函数 : find_in_set ({findString} , {stringValue})

九、集合统计函数

1. 计数函数

2. 求和函数

3. 平均值函数

4. 最值函数

5. 非空集合总体变量函数

6. 非空集合样本变量函数

7. 总体标准偏离函数

8. 样本标准偏离函数

9. 中位数函数

10. 直方图函数

十、复合类型函数

1. Map类型函数

2. Struct类型函数

3. Array类型函数

4. 类型转换函数

十一、用户自定义函数

1. 函数编写

2. 函数注册

3. 函数调用