

hive严格模式

hive提供了一个严格模式，可以防止用户执行那些可能产生意想不到的不好的效果的查询。即某些查询在严格模式下无法执行。通过设置hive.mapred.mode的值为strict，可以禁止3中类型的查询。

1) 带有分区的表的查询

如果在一个分区表执行hive，除非where语句中包含分区字段过滤条件来显示数据范围，否则不允许执行。换句话说，就是用户不允许扫描所有的分区。进行这个限制的原因是，通常分区表都拥有非常大的数据集，而且数据增加迅速。

如果没有进行分区限制的查询可能会消耗令人不可接受的巨大资源来处理这个表：

```
hive> SELECT DISTINCT(planner_id) FROM fracture_ins WHERE planner_id=5;
```

```
FAILED: Error in semantic analysis: No Partition Predicate Found for Alias
```

```
"fracture_ins" Table "fracture_ins"
```

如下这个语句在where语句中增加了一个分区过滤条件（也就是限制了表分区）：

```
hive> SELECT DISTINCT(planner_id) FROM fracture_ins
```

```
> WHERE planner_id=5 AND hit_date=20120101;
```

```
... normal results ...
```

2) 带有orderby的查询

对于使用了orderby的查询，要求必须有limit语句。因为orderby为了执行排序过程会讲所有的结果分发到同一个reducer中

进行处理，强烈要求用户增加这个limit语句可以防止reducer额外执行很长一段时间：

```
hive> SELECT * FROM fracture_ins WHERE hit_date>2012 ORDER BY planner_id;
```

```
FAILED: Error in semantic analysis: line 1:56 In strict mode,
```

```
limit must be specified if ORDER BY is present planner_id
```

只需要增加limit语句就可以解决这个问题：

```
hive> SELECT * FROM fracture_ins WHERE hit_date>2012 ORDER BY planner_id  
> LIMIT 100000;  
... normal results ...
```

3) 限制笛卡尔积的查询

对关系型数据库非常了解的用户可能期望在执行join查询的时候不使用on语句而是使用where语句，这样关系数据库的执行

优化器就可以高效的将where语句转换成那个on语句。不幸的是，hive不会执行这种优化，因此，如果表足够大，那么这个查询就会

出现不可控的情况：

```
hive> SELECT * FROM fracture_act JOIN fracture_ads  
> WHERE fracture_act.planner_id = fracture_ads.planner_id;  
FAILED: Error in semantic analysis: In strict mode, cartesian product  
is not allowed. If you really want to perform the operation,  
+set hive.mapred.mode=nonstrict+
```

下面这个才是正确的使用join和on语句的查询：

```
hive> SELECT * FROM fracture_act JOIN fracture_ads  
> ON (fracture_act.planner_id = fracture_ads.planner_id);  
... normal results ...
```