

6.6 Hive内部表、外部表、分区表

Hive内部表、外部表、分区表

◆ 主要内容

6.6.1内部表与外部表

6.6.2分区表

6.6.1内部表与外部表

◆ 运用上次课学到的知识，构建以下两个表

员工表（表名：emp）

字段名	数据类型	字段含义
empno	int	员工编号
ename	string	员工姓名
job	string	职位
mgr	int	领导编号
hiredate	string	入职时间
sal	double	薪资
comm	double	奖金
deptno	int	部门编号

部门表（表名：dept）

字段名	数据类型	字段含义
deptno	int	部门编号
deptname	string	部门名称
loc	string	部门地点

员工表的数据位置（本地）：/opt/datas/emp.txt
部门表的数据位置（本地）： /opt/datas/dept.txt

6.6.1内部表与外部表

员工表的数据，分隔符为 ‘\t’

dept.txt - 记事本		
文件(F)	编辑(E)	格式(O) 查看(V) 帮助(H)
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES CHICAGO	
40	OPERATIONS	BOSTON

部门表的数据，分隔符为 ‘\t’

emp.txt - 记事本								
文件(F)	编辑(E)	格式(O)	查看(V)	帮助(H)				
7369	SMITH	CLERK	7902	1980-12-17	800.00	20		
7499	ALLEN	SALESMAN	7698	1981-2-20	1600.00	300.00	30	
7521	WARD	SALESMAN	7698	1981-2-22	1250.00	500.00	30	
7566	JONES	MANAGER	7839	1981-4-2	2975.00	20		
7654	MARTIN	SALESMAN	7698	1981-9-28	1250.00	1400.00	30	
7698	BLAKE	MANAGER	7839	1981-5-1	2850.00	30		
7782	CLARK	MANAGER	7839	1981-6-9	2450.00	10		
7788	SCOTT	ANALYST	7566	1987-4-19	3000.00	20		
7839	KING	PRESIDENT		1981-11-17	5000.00		10	
7844	TURNER	SALESMAN	7698	1981-9-8	1500.00	0.00	30	
7876	ADAMS	CLERK	7788	1987-5-23	1100.00	20		
7900	JAMES	CLERK	7698	1981-12-3	950.00	30		
7902	FORD	ANALYST	7566	1981-12-3	3000.00	20		
7934	MILLER	CLERK	7782	1982-1-23	1300.00	10		

部门表（表名：dept）

字段名	数据类型	字段含义
deptno	int	部门编号
deptname	string	部门名称
loc	string	部门地点

员工表（表名：emp）

字段名	数据类型	字段含义
empno	int	员工编号
ename	string	员工姓名
job	string	职位
mgr	int	领导编号
hiredate	string	入职时间
sal	double	薪资
comm	double	奖金
deptno	int	部门编号

6.6.1内部表与外部表

```
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';  
load data local inpath '/opt/datas/emp.txt'  
  into table emp;
```

```
create table dept(  
  deptno int,  
  dname string,  
  loc string  
)  
row format delimited fields terminated by  
  '\t';  
load data local inpath '/opt/datas/dept.txt'  
  overwrite into table dept;
```

6.6.1 内部表与外部表

1、将代码输入hive终端后，查询这两表

```
hive (emp_test)> select * from emp;
OK
7369 SMITH CLERK 7902 1980-12-17 800.0 NULL 20
7499 ALLEN SALESMAN 7698 1981-2-20 1600.0 300.0 30
7521 WARD SALESMAN 7698 1981-2-22 1250.0 500.0 30
7566 JONES MANAGER 7839 1981-4-2 2975.0 NULL 20
7654 MARTIN SALESMAN 7698 1981-9-28 1250.0 1400.0 30
7698 BLAKE MANAGER 7839 1981-5-1 2850.0 NULL 30
7782 CLARK MANAGER 7839 1981-6-9 2450.0 NULL 10
7788 SCOTT ANALYST 7566 1987-4-19 3000.0 NULL 20
7839 KING PRESIDENT NULL 1981-11-17 5000.0 NULL 10
7844 TURNER SALESMAN 7698 1981-9-8 1500.0 0.0 30
7876 ADAMS CLERK 7788 1987-5-23 1100.0 NULL 20
7900 JAMES CLERK 7698 1981-12-3 950.0 NULL 30
7902 FORD ANALYST 7566 1981-12-3 3000.0 NULL 20
7934 MILLER CLERK 7782 1982-1-23 1300.0 NULL 10
Time taken: 0.048 seconds, Fetched: 14 row(s)
```

```
hive (emp_test)> select * from dept;
OK
10 ACCOUNTING NEW YORK
20 RESEARCH DALLAS
30 SALES CHICAGO
40 OPERATIONS BOSTON
Time taken: 0.034 seconds, Fetched: 4 row(s)
hive (emp_test)>
```


6.6.1 内部表与外部表

2、使用desc formatted命令查看emp表的详细信息

```
Database:                emp_test
Owner:                   hpsk
CreateTime:              Tue May 23 16:48:44 CST 2017
LastAccessTime:          UNKNOWN
Protect Mode:            None
Retention:               0
Location:                hdfs://bigdata-training01.hpsk.com:8020/user/hive/warehouse/emp_test.db/emp
Table Type:              MANAGED_TABLE
Table Parameters:
    COLUMN_STATS_ACCURATE    true
    numFiles                  1
    numRows                   0
    rawDataSize               0
    totalSize                 656
    transient_lastDdlTime     1495529412

# Storage Information
SerDe Library:           org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:             org.apache.hadoop.mapred.TextInputFormat
OutputFormat:            org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Compressed:              No
Num Buckets:             -1
Bucket Columns:          []
```

观察发现，在Table Type这一项的值为：MANAGED_TABLE

6.6.1 内部表与外部表

◆ HIVE中表的类型

1、管理表，也称为内部表

Table Type=MANAGED_TABLE

2、外部表

Table Type=EXTERNAL_TABLE

6.6.1 内部表与外部表

◆ 如何创建外部表?

回到创建表的命令:

```
CREATE [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.]table_name
```

创建表时，如果不加EXTERNAL关键字，创建的是管理表

如果添加EXTERNAL关键字，创建的就是外部表

例: create EXTERNAL table emp_e(.....

6.6.1 内部表与外部表

◆ 通过一个实例对比管理表与外部表的区别

1、构造一个管理表emp_m，其内容与emp一样

```
create table emp_m(  
.....  
)  
row format delimited fields terminated by '\t';  
load data local inpath '/opt/datas/emp.txt' into table emp_m;
```

2、构造一个外部表emp_e，其内容也与emp一样

```
create EXTERNAL table emp_e(  
.....  
)  
row format delimited fields terminated by '\t';  
load data local inpath '/opt/datas/emp.txt' into table emp_e;
```

6.6.1 内部表与外部表

3、分别查询emp_m和emp_e两个表的内容，内容上无任何区别

```
hive (emp_test)> select * from emp_m;
```

OK

7369	SMITH	CLERK	7902	1980-12-17	800.0	NULL	20	
7499	ALLEN	SALESMAN	7698	1981-2-20	1600.0	300.0	30	
7521	WARD	SALESMAN	7698	1981-2-22	1250.0	500.0	30	
7566	JONES	MANAGER	7839	1981-4-2	2975.0	NULL	20	
7654	MARTIN	SALESMAN	7698	1981-9-28	1250.0	1400.0	30	
7698	BLAKE	MANAGER	7839	1981-5-1	2850.0	NULL	30	
7782	CLARK	MANAGER	7839	1981-6-9	2450.0	NULL	10	
7788	SCOTT	ANALYST	7566	1987-4-19	3000.0	NULL	20	
7839	KING	PRESIDENT	NULL	1981-11-17	5000.0	NULL	10	
7844	TURNER	SALESMAN	7698	1981-9-8	1500.0	0.0	30	
7876	ADAMS	CLERK	7788	1987-5-23	1100.0	NULL	20	
7900	JAMES	CLERK	7698	1981-12-3	950.0	NULL	30	
7902	FORD	ANALYST	7566	1981-12-3	3000.0	NULL	20	
7934	MILLER	CLERK	7782	1982-1-23	1300.0	NULL	10	

Time taken: 0.032 seconds, Fetched: 14 row(s)

```
hive (emp_test)> select * from emp_e;
```

OK

7369	SMITH	CLERK	7902	1980-12-17	800.0	NULL	20	
7499	ALLEN	SALESMAN	7698	1981-2-20	1600.0	300.0	30	
7521	WARD	SALESMAN	7698	1981-2-22	1250.0	500.0	30	
7566	JONES	MANAGER	7839	1981-4-2	2975.0	NULL	20	
7654	MARTIN	SALESMAN	7698	1981-9-28	1250.0	1400.0	30	
7698	BLAKE	MANAGER	7839	1981-5-1	2850.0	NULL	30	
7782	CLARK	MANAGER	7839	1981-6-9	2450.0	NULL	10	
7788	SCOTT	ANALYST	7566	1987-4-19	3000.0	NULL	20	
7839	KING	PRESIDENT	NULL	1981-11-17	5000.0	NULL	10	
7844	TURNER	SALESMAN	7698	1981-9-8	1500.0	0.0	30	
7876	ADAMS	CLERK	7788	1987-5-23	1100.0	NULL	20	
7900	JAMES	CLERK	7698	1981-12-3	950.0	NULL	30	
7902	FORD	ANALYST	7566	1981-12-3	3000.0	NULL	20	
7934	MILLER	CLERK	7782	1982-1-23	1300.0	NULL	10	

Time taken: 0.053 seconds, Fetched: 14 row(s)

4、分别查看emp_m和emp_e两个表所在的目录，两个表都在同一个数据库中：

Browse Directory

Permission	Owner	Group	Size	Replication	Block Size	Name
drwxr-xr-x	hpsk	supergroup	0 B	0	0 B	dept
drwxr-xr-x	hpsk	supergroup	0 B	0	0 B	emp
drwxr-xr-x	hpsk	supergroup	0 B	0	0 B	emp_e
drwxr-xr-x	hpsk	supergroup	0 B	0	0 B	emp_m

5、查看emp_m表的目录，里面有一个导入的emp.txt文件：

Browse Directory

Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	hpsk	supergroup	656 B	1	128 MB	emp.txt

6、查看emp_e表的目录，里面也有一个导入的emp.txt文件：

Browse Directory

Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	hpsk	supergroup	656 B	1	128 MB	emp.txt

6.6.1 内部表与外部表

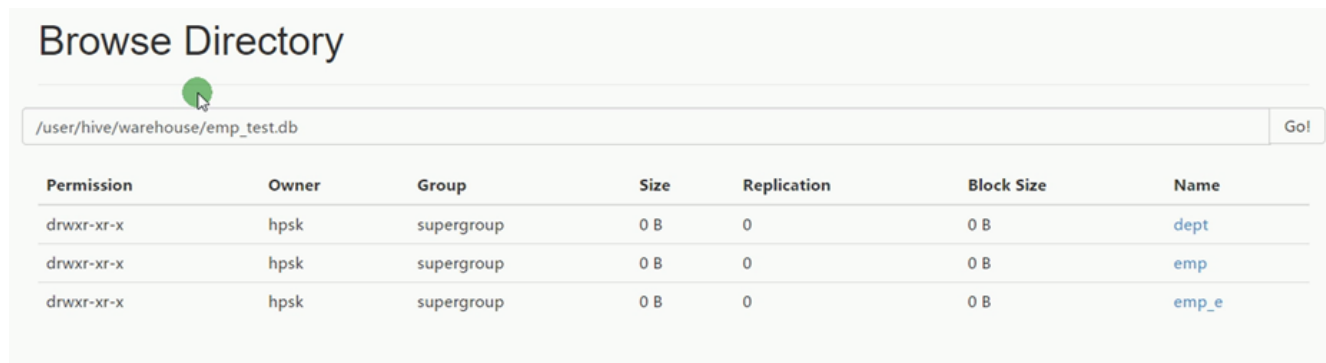
7、分别将emp_m和emp_e两个表进行删除操作

删除操作的命令为 drop table

```
hive (emp_test)> drop table emp_m;  
OK  
Time taken: 0.114 seconds  
hive (emp_test)> drop table emp_e;  
OK  
Time taken: 0.088 seconds  
hive (emp_test)> show tables;  
OK  
dept  
emp  
Time taken: 0.009 seconds, Fetched: 2 row(s)  
hive (emp_test)>
```

使用show tables可以看到，数据库中emp_m和emp_e两个表的元数据已经被删除了

8、分别进入emp_m和emp_e两个表的目录，观察数据文件是否还在



Browse Directory

/user/hive/warehouse/emp_test.db

Permission	Owner	Group	Size	Replication	Block Size	Name
drwxr-xr-x	hpsk	supergroup	0 B	0	0 B	dept
drwxr-xr-x	hpsk	supergroup	0 B	0	0 B	emp
drwxr-xr-x	hpsk	supergroup	0 B	0	0 B	emp_e

发现emp_m表的目录也被删除了，而emp_e表的目录依然还在
进入emp_e的目录



Browse Directory

/user/hive/warehouse/emp_test.db/emp_e

Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	hpsk	supergroup	656 B	1	128 MB	emp.txt

发现emp_e的目录中的数据文件依然还在

6.6.1 内部表与外部表

◆ 通过上述对比可以发现

管理表（内部表）

- 删除时会删除元数据
- 删除时会删除HDFS的数据文件

外部表

- 删除时会删除元数据
- 删除时不会删除HDFS的数据文件，保证了数据的安全性

6.6.1 内部表与外部表

◆ 管理表

hive认为完全拥有这份数据，当删除一个管理表时，Hive 也会删除这个表中数据。

管理表不适合和其他工具共享数据

◆ 外部表

Hive 并非认为其完全拥有这份数据。删除该表并不会删除掉这份数据，不过描述表的元数据信息会被删除掉

◆ 应用场景

假定每天将收集到的网站日志定期流入 HDFS 文本文件。

需要在这份日志上做大量的统计分析，或者有其他用户、其他分析软件也需要用到这份文本文件此时应建立外部表，防止数据被误删除，保证安全性（通过location指定同一份数据源）

在分析的过程中如果需要用到的中间表、结果表使用管理表存储

6.6.1 内部表与外部表

◆ 管理表与外部表的互相转换

使用命令 `alter table tablename set tblproperties('EXTERNAL'='TRUE');`

如果 `'EXTERNAL'='TRUE'`，则表示修改为外部表；

如果 `'EXTERNAL'='FALSE'`，则表示修改为管理表；

注意： `('EXTERNAL'='TRUE')`和`('EXTERNAL'='FALSE')`为固定写法，
区分大小写！

例：将student2表由管理表改为外部表

1、查询student2表的类型

```
hive (default)> desc formatted student2;  
Table Type:          MANAGED_TABLE
```

2、将student2表由管理表改为外部表

```
alter table student2 set tblproperties('EXTERNAL'='TRUE');
```

3、查询student2表的类型

```
hive (default)> desc formatted student2;  
Table Type:          EXTERNAL TABLE
```

4、将student2表由外部表改为管理表

```
alter table student2 set tblproperties('EXTERNAL'='FALSE');
```

5、查询student2表的类型

```
hive (default)> desc formatted student2;  
Table Type:          MANAGED_TABLE
```

6.6.1 内部表与外部表

◆ 外部表常用写法

与location连用

```
1 create external table test_external_location (  
2   id string comment 'ID',  
3   name string comment '名字'  
4 )  
5 comment '测试外部表location'  
6 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','  
7 location '/tmp/dkl/external_location';
```

6.6.1 内部表与外部表

◆ 问：为什么用location指定同一份源数据，而不将源数据复制多份分别给大家分析呢？

导致数据冗余，HDFS备份机制，假定一份源数据，备份机制多存了3份，再复制一份相当于再多了3份数据，导致HDFS效率降低，数据大量冗余。

如果大家都使用外部表分析，可以同时创建多张表分析不同业务，大家共享源数据，如果有人不用了，删除自己的外部表即可，不影响其他人的分析使用

Hive内部表、外部表、分区表

◆ 主要内容

6.6.1 内部表与外部表

6.6.2 分区表

6.6.2 分区表

◆ 主要内容

分区表介绍

分区表的创建及使用

6.6.2 分区表

◆ 介绍分区表之前，提出一个需求：

统计每一天的PV，UV，每一天分析前一天的数据

◆ 什么是PV，UV？

UV(Unique Visitor)：独立访客，将每个独立上网电脑视为一位访客，一天之内（00:00-24:00）访问网站的访客数量

PV（Page View）：访问量，即页面浏览量或者点击量。用户每次对网站的访问均被记录1次，用户对同一页面的多次访问，访问量值累计

6.6.2 分区表

- ◆ 统计每一天的PV，UV，每一天分析前一天的数据（即今天分析昨天的数据）

第一种情况：假定所有日志文件都存储在同一个目录log下：

 /logs/1.log

 2.log

 3.log

步骤一：预处理，由于.log文件中的日期格式通常不会是yyyyMMdd样式，因此先提取日期字段转换成yyyyMMdd样式（数据清洗）

步骤二：数据分析：假定今天为2017年2月12日，分析昨天的日志：

```
select * from logs where date='20170211';
```

.....

6.6.2 分区表

/logs/1.log

2.log

3.log

语句 “select * from logs where date='20170211';” 执行过程

- Hive首先寻找log表的目录，由于所有日志文件都存在表log的目录下，因此select * from logs 会将表中的所有文件全部加载到MapReduce中，再根据where语句在数据内寻找date='20170211'

即先加载所有数据，再查找，效率比较低

6.6.2 分区表

第二种情况：每天的日志存储在以当天日期命名的文件目录中（多了一层目录）

`/logs/20170209/1.log`

`20170210/2.log`

`20170211/3.log`

6.6.2 分区表

/logs/20170209/1.log

20170210/2.log

20170211/3.log

此时语句 “select * from logs where date='20170211';” 执行过程：

·Hive首先寻找log表的目录，找到log表目录后发现表目录下不是文件而是目录。此时，hive会根据where给出的条件寻找目录，加载所需目录下的文件里的数据

- ✓ 先进行根据子目录名称，加载需要的数据，即分区表实现的功能
- ✓ 在海量数据下（假如有一年或者几年的日志文件）提高效率

6.6.2 分区表

◆ HIVE分区表实现的功能：

将表中的数据进行分区，在进行分区检索时，直接加载对应分区的数据

对于HDFS来说，表的目录下多了一级目录

对于数据处理来说，先进行了过滤

6.6.2 分区表

◆ 如何创建分区表:

创建表的命令格式:

```
CREATE [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.]table_name
[(col_name data_type [COMMENT col_comment], ... [constraint_specification])]
[PARTITIONED BY (col_name data_type [COMMENT col_comment], ...)]
[ROW FORMAT row_format]
[STORED AS file_format]
[LOCATION hdfs_path]
[AS select_statement];
```

6.6.2 分区表

- ◆ 将构造emp_part表按照date（string类型）进行分区，字段与emp表一致

```
create table emp_part(  
  empno int,  
  ename string,  
  job string,  
  mgr int,  
  hiredate string,  
  sal double,  
  comm double,  
  deptno int  
)
```

注意：分区表中的字段是逻辑的字段，数据文件中没有实际的字段存储

emp.txt - 记事本								
文件(F)	编辑(E)	格式(O)	查看(V)	帮助(H)				
7369	SMITH	CLERK	7902	1980-12-17	800.00	20		
7499	ALLEN	SALESMAN	7698	1981-2-20	1600.00	300.00	30	
7521	WARD	SALESMAN	7698	1981-2-22	1250.00	500.00	30	
7566	JONES	MANAGER	7839	1981-4-2	2975.00	20		
7654	MARTIN	SALESMAN	7698	1981-9-28	1250.00	1400.00	30	
7698	BLAKE	MANAGER	7839	1981-5-1	2850.00	30		
7782	CLARK	MANAGER	7839	1981-6-9	2450.00	10		
7788	SCOTT	ANALYST	7566	1987-4-19	3000.00	20		
7839	KING	PRESIDENT		1981-11-17	5000.00		10	
7844	TURNER	SALESMAN	7698	1981-9-8	1500.00	0.00	30	
7876	ADAMS	CLERK	7788	1987-5-23	1100.00	20		
7900	JAMES	CLERK	7698	1981-12-3	950.00	30		
7902	FORD	ANALYST	7566	1981-12-3	3000.00	20		
7934	MILLER	CLERK	7782	1982-1-23	1300.00	10		

partitioned by (date string)

row format delimited fields terminated by '\t';

load data local inpath '/opt/datas/emp.txt' into table emp_part;

将上述代码输入后回车

```
> ename string,  
> job string,  
> mgr int,  
> hiredate string,  
> sal double,  
> comm double,  
> deptno int  
> )  
> partitioned by (date string)  
> row format delimited fields terminated by '\t';  
OK  
Time taken: 0.527 seconds  
hive (emp_test)> show tables;  
OK  
tab_name  
dept  
emp  
emp_part  
Time taken: 0.023 seconds, Fetched: 3 row(s)  
hive (emp_test)> load data local inpath '/opt/datas/emp.txt' into table emp_part;  
FAILED: SemanticException [Error 10062]: Need to specify partition columns because the destination table is partitioned  
hive (emp_test)>
```

报错，原因：加载数据到分区表必须制定分区

/logs/1.log

2.log

3.log

/logs/20170209/1.log

20170210/2.log

20170211/3.log

非分区表，logs下存放的就是数据文件

分区表，logs下存的不是数据文件，数据存放在logs下的子目录中，因此必须指定存放在哪个子目录中

6.6.2 分区表

- ◆ 因此，应修改load data语句，指定存放的分区

```
create table emp_part(  
  ...  
)
```

```
partitioned by (date string)
```

```
row format delimited fields terminated by '\t';
```

```
load data local inpath '/opt/datas/emp.txt' into table emp_part;
```



```
load data local inpath '/opt/datas/emp.txt' into table emp_part partition  
(date='20170211');
```


加载数据到分区表

```
hive (emp_test)> load data local inpath '/opt/datas/emp.txt' into table emp_part partition (date='20170211');
Copying data from file:/opt/datas/emp.txt
Copying file: file:/opt/datas/emp.txt
Loading data to table emp_test.emp_part partition (date=20170211)
Partition emp_test.emp_part{date=20170211} stats: [numFiles=1, numRows=0, totalSize=656, rawDataSize=0]
OK
Time taken: 0.956 seconds
```

使用select查询表

```
hive (emp_test)> select * from emp_part;
OK
emp_part.empno emp_part.ename emp_part.job emp_part.mgr emp_part.hiredate emp_part.sal e
mp_part.comm emp_part.deptno emp_part.date
7369 SMITH CLERK 7902 1980-12-17 800.0 NULL 20 20170211
7499 ALLEN SALESMAN 7698 1981-2-20 1600.0 300.0 30 20170211
7521 WARD SALESMAN 7698 1981-2-22 1250.0 500.0 30 20170211
7566 JONES MANAGER 7839 1981-4-2 2975.0 NULL 20 20170211
7654 MARTIN SALESMAN 7698 1981-9-28 1250.0 1400.0 30 20170211
7698 BLAKE MANAGER 7839 1981-5-1 2850.0 NULL 30 20170211
7782 CLARK MANAGER 7839 1981-6-9 2450.0 NULL 10 20170211
7788 SCOTT ANALYST 7566 1987-4-19 3000.0 NULL 20 20170211
7839 KING PRESIDENT NULL 1981-11-17 5000.0 NULL 10 20170211
7844 TURNER SALESMAN 7698 1981-9-8 1500.0 0.0 30 20170211
7876 ADAMS CLERK 7788 1987-5-23 1100.0 NULL 20 20170211
7900 JAMES CLERK 7698 1981-12-3 950.0 NULL 30 20170211
7902 FORD ANALYST 7566 1981-12-3 3000.0 NULL 20 20170211
7934 MILLER CLERK 7782 1982-1-23 1300.0 NULL 10 20170211
Time taken: 0.617 seconds, Fetched: 14 row(s)
hive (emp_test)>
```

观察发现，多了一个emp_part.data字段，该字段下还有20170211，但我们的数据文件里并没有这个字段，只是hive在逻辑上添加的一个字段

6.6.2 分区表

观察表所在的目录

Browse Directory						
<input type="text" value="/user/hive/warehouse/emp_test.db/emp_part"/>						<input data-bbox="1649 568 1711 608" type="button" value="Go!"/>
Permission	Owner	Group	Size	Replication	Block Size	Name
drwxr-xr-x	hpsk	supergroup	0 B	0	0 B	date=20170211

在表emp_part下多了个“date=20170211”目录，进入该目录即可找到数据文件

Browse Directory						
<input type="text" value="/user/hive/warehouse/emp_test.db/emp_part/date=20170211"/>						<input data-bbox="1591 1125 1653 1165" type="button" value="Go!"/>
Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	hpsk	supergroup	656 B	1	128 MB	emp.txt

如果此时，我们再将数据加载到date='20170212'下，即接着输入代码：

load data local inpath '/opt/datas/emp.txt' into table emp_part partition (date='20170212');

```
hive (emp_test)> load data local inpath '/opt/datas/emp.txt' into table emp_part partition (date='20170212');
Copying data from file:/opt/datas/emp.txt
Copying file: file:/opt/datas/emp.txt
Loading data to table emp_test.emp_part partition (date=20170212)
Partition emp_test.emp_part{date=20170212} stats: [numFiles=1, numRows=0, totalSize=656, rawDataSize=0]
OK
Time taken: 0.46 seconds
hive (emp_test)> █
```

再使用select查询表

7788	SCOTT	ANALYST	7566	1987-4-19	3000.0	NULL	20	20170211	
7839	KING	PRESIDENT		NULL	1981-11-17	5000.0	NULL	10	20170211
7844	TURNER	SALESMAN		7698	1981-9-8	1500.0	0.0	30	20170211
7876	ADAMS	CLERK	7788	1987-5-23	1100.0	NULL	20	20170211	
7900	JAMES	CLERK	7698	1981-12-3	950.0	NULL	30	20170211	
7902	FORD	ANALYST	7566	1981-12-3	3000.0	NULL	20	20170211	
7934	MILLER	CLERK	7782	1982-1-23	1300.0	NULL	10	20170211	
7369	SMITH	CLERK	7902	1980-12-17	800.0	NULL	20	20170212	
7499	ALLEN	SALESMAN		7698	1981-2-20	1600.0	300.0	30	20170212
7521	WARD	SALESMAN		7698	1981-2-22	1250.0	500.0	30	20170212
7566	JONES	MANAGER	7839	1981-4-2	2975.0	NULL	20	20170212	
7654	MARTIN	SALESMAN		7698	1981-9-28	1250.0	1400.0	30	20170212
7698	BLAKE	MANAGER	7839	1981-5-1	2850.0	NULL	30	20170212	
7782	CLARK	MANAGER	7839	1981-6-9	2450.0	NULL	10	20170212	
7788	SCOTT	ANALYST	7566	1987-4-19	3000.0	NULL	20	20170212	
7839	KING	PRESIDENT		NULL	1981-11-17	5000.0	NULL	10	20170212
7844	TURNER	SALESMAN		7698	1981-9-8	1500.0	0.0	30	20170212
7876	ADAMS	CLERK	7788	1987-5-23	1100.0	NULL	20	20170212	
7900	JAMES	CLERK	7698	1981-12-3	950.0	NULL	30	20170212	
7902	FORD	ANALYST	7566	1981-12-3	3000.0	NULL	20	20170212	
7934	MILLER	CLERK	7782	1982-1-23	1300.0	NULL	10	20170212	

Time taken: 0.055 seconds, Fetched: 28 row(s)
hive (emp_test)> █

可以发现emp_part.data字段下既有20170211的数据，也有20170212的数据

6.6.2 分区表

◆ 如果想查找emp_part表里某个分区的数据该怎么查询?

```
select * from emp_part where date='20170212'
```

```
hive (emp_test)> select * from emp_part where date='20170212';
```

```
OK
```

emp_part.empno	emp_part.ename	emp_part.job	emp_part.mgr	emp_part.hiredate	emp_part.sal	e
mp_part.comm	emp_part.deptno	emp_part.date				
7369	SMITH	CLERK 7902	1980-12-17	800.0	NULL	20 20170212
7499	ALLEN	SALESMAN	7698 1981-2-20	1600.0	300.0	30 20170212
7521	WARD	SALESMAN	7698 1981-2-22	1250.0	500.0	30 20170212
7566	JONES	MANAGER 7839	1981-4-2	2975.0	NULL	20 20170212
7654	MARTIN	SALESMAN	7698 1981-9-28	1250.0	1400.0	30 20170212
7698	BLAKE	MANAGER 7839	1981-5-1	2850.0	NULL	30 20170212
7782	CLARK	MANAGER 7839	1981-6-9	2450.0	NULL	10 20170212
7788	SCOTT	ANALYST 7566	1987-4-19	3000.0	NULL	20 20170212
7839	KING	PRESIDENT	NULL 1981-11-17	5000.0	NULL	10 20170212
7844	TURNER	SALESMAN	7698 1981-9-8	1500.0	0.0	30 20170212
7876	ADAMS	CLERK 7788	1987-5-23	1100.0	NULL	20 20170212
7900	JAMES	CLERK 7698	1981-12-3	950.0	NULL	30 20170212
7902	FORD	ANALYST 7566	1981-12-3	3000.0	NULL	20 20170212
7934	MILLER	CLERK 7782	1982-1-23	1300.0	NULL	10 20170212

```
Time taken: 0.351 seconds, Fetched: 14 row(s)
```

```
hive (emp_test)> █
```

6.6.2 分区表

◆ 观察emp_part表所在的目录

存在data=20170211的目录，也有data=20170212的目录

Browse Directory						
<input type="text" value="/user/hive/warehouse/emp_test.db/emp_part"/>						<input type="button" value="Go!"/>
Permission	Owner	Group	Size	Replication	Block Size	Name
drwxr-xr-x	hpsk	supergroup	0 B	0	0 B	date=20170211
drwxr-xr-x	hpsk	supergroup	0 B	0	0 B	date=20170212

Browse Directory						
<input type="text" value="/user/hive/warehouse/emp_test.db/emp_part/date=20170212"/>						<input type="button" value="Go!"/>
Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	hpsk	supergroup	656 B	1	128 MB	emp.txt

6.6.2 分区表

◆ 多级分区

如果需要在日期下再按小时来分区，该怎么做？

构造表二级分区表emp_part_2

```
create table emp_part_2(
```

```
.....
```

```
)
```

```
partitioned by (date string, hour string)
```

```
row format delimited fields terminated by '\t';
```

```
load data local inpath '/opt/datas/emp.txt' into table emp_part_2 partition
```

```
(date='20170211', hour='00');
```

注意：加载数据此时也要指定将数据加载到两级分区的位置，即给到最终数据文件存放的目录

加载数据文件到两级的分区目录

```
hive (emp_test)> load data local inpath '/opt/datas/emp.txt' into table emp_part_2 partition (date='20170211',hour='00');
Copying data from file:/opt/datas/emp.txt
Copying file: file:/opt/datas/emp.txt
Loading data to table emp_test.emp_part_2 partition (date=20170211, hour=00)
Partition emp_test.emp_part_2{date=20170211, hour=00} stats: [numFiles=1, numRows=0, totalSize=656, rawDataSize=0]
OK
Time taken: 0.427 seconds
hive (emp_test)>
```

使用select查询

```
hive (emp_test)> select * from emp_part_2;
OK
emp_part_2.empno      emp_part_2.ename      emp_part_2.job      emp_part_2.mgr      emp_part_2.hiredate      e
mp_part_2.sal      emp_part_2.comm      emp_part_2.deptno      emp_part_2.date      emp_part_2.hour
7369      SMITH      CLERK      7902      1980-12-17      800.0      NULL      20      20170211      00
7499      ALLEN      SALESMAN      7698      1981-2-20      1600.0      300.0      30      20170211      00
7521      WARD      SALESMAN      7698      1981-2-22      1250.0      500.0      30      20170211      00
7566      JONES      MANAGER      7839      1981-4-2      2975.0      NULL      20      20170211      00
7654      MARTIN      SALESMAN      7698      1981-9-28      1250.0      1400.0      30      20170211      00
7698      BLAKE      MANAGER      7839      1981-5-1      2850.0      NULL      30      20170211      00
7782      CLARK      MANAGER      7839      1981-6-9      2450.0      NULL      10      20170211      00
7788      SCOTT      ANALYST      7566      1987-4-19      3000.0      NULL      20      20170211      00
7839      KING      PRESIDENT      NULL      1981-11-17      5000.0      NULL      10      20170211      00
7844      TURNER      SALESMAN      7698      1981-9-8      1500.0      0.0      30      20170211      00
7876      ADAMS      CLERK      7788      1987-5-23      1100.0      NULL      20      20170211      00
7900      JAMES      CLERK      7698      1981-12-3      950.0      NULL      30      20170211      00
7902      FORD      ANALYST      7566      1981-12-3      3000.0      NULL      20      20170211      00
7934      MILLER      CLERK      7782      1982-1-23      1300.0      NULL      10      20170211      00
Time taken: 0.044 seconds, Fetched: 14 row(s)
hive (emp_test)>
```

观察发现多了两个字段，即我们指定的分区

观察emp_part_2表所在的目录

可以找到第一个分区目录 (data=20170211)

Browse Directory						
/user/hive/warehouse/emp_test.db/emp_part_2						
Permission	Owner	Group	Size	Replication	Block Size	Name
drwxr-xr-x	hpsk	supergroup	0 B	0	0 B	date=20170211

进入data=20170211目录，可以找到第二个分区目录 (hour=00)

Browse Directory						
/user/hive/warehouse/emp_test.db/emp_part_2/date=20170211						Go!
Permission	Owner	Group	Size	Replication	Block Size	Name
drwxr-xr-x	hpsk	supergroup	0 B	0	0 B	hour=00

进入hour=00目录，可以找到数据文件

Browse Directory						
/user/hive/warehouse/emp_test.db/emp_part_2/date=20170211/hour=00						Go!
Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	hpsk	supergroup	656 B	1	128 MB	emp.txt

6.6.2 分区表

◆ 问：多级分区表如何查找？

答：

```
select * from emp_part where date='20170212' and hour='00';
```

总结

◆ 主要内容

6.6.1 内部表与外部表

- 内部表外部表区别
- 如何创建内部表和外部表
- 如何相互转换
- 应用

6.6.2 分区表

- 怎么创建分区表
- 如何创建