

一. hive 数据导入.

1. load

单个 `hive> load data local inpath '/root/data/student01.txt' into table`

所有 `hive> load data local inpath '/root/data/' overwrite into`

`> table t3;`

`hive> load data inpath hdfs中 '/input/student01.txt' overwrite table`

导入HBase: `hive> load data local inpath ^ '/root/student01.txt' into table`

`> partition-table partition on (gender='M');`

2. sqoop 工具

需要: ① tar 工具

② `test/profile`

③ `bin/`

④ `jdbc.jar` $\xrightarrow{\text{拷贝}}$ `sqoop/lib/下`

数据迁移:

i. Oracle \rightarrow hdfs



```
$ ./sqoop import --connect jdbc:oracle:thin:@192.168.0.102:1521:orcl
--username scott --password tiger --table emp --columns
empno, ename, job, sal, deptno -m 1 --target-dir 'sqoop/emp'
```

解析为 M/R job.

2. Oracle → hive

```
$ ./sqoop import --connect << --username << --password <<
--table emp --columns ' << '
```

注: 未指定表名, 则使用原表的表名, 增加 --hive-table ^{emp} << 可指定
where 条件导入 --where 'deptno=10'

query 条件导入 --query 'select * from emp
where sal < 2000'
注: 要删除 --table emp
--target-dir 'sqoop/emp5'
--hive-table emp5.

增加 and 条件

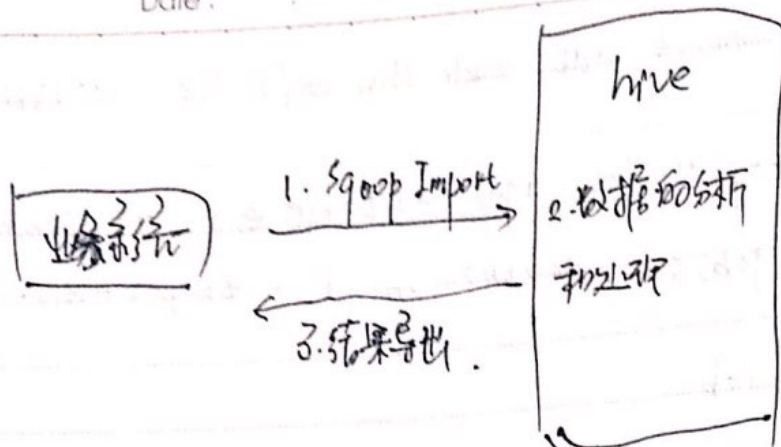
导出数据: hive → Oracle

```
./sqoop export --connect << --username << --password << -m 1
```

```
--table myemp --export-dir '/input/my/emp.txt'
```

到结构一致.





三、hive 数据查询.

hive > select emp^{emp}no, ename, sal, sal*12, 10mm, sal*12 + nvl(comm, 0) fr

简单查询: 没有函数, 没有排序, fetch task. 而不是 map/reduce job.

① hive > set hive.fetch.task.conversion=more;

② hive > hive --hiveconf hive.fetch.task.conversion=more;

③. 修改 hive-site.xml. --永久生效.

查询: 过滤与排序.

hive > explain ... -- 使用, 看法: 从右向左, 从下到上

like "0% \|-%" \- 转义符.

where 不是 MR job.



order by 列, 表名, 别名, 序号

hive → set hive.groupby.orderby.position.alias=true;

数值排序

null 升序: 最前边. (默认) null (comm, 0)

降序: 最后边.

函数:

1. 内置.

2. JAVA 自定义.

数学:

round: 四舍五入.

ceil: 向上取整.

字符:

lower: 转小写.

upper: 转大写.

length: 长度/字符数.

concat: 拼接字符串.

substr: 截取.

trim: 去空格.

lpad: 左填充.

rpad: 右填充.



No.

Date:

收集与转换.

size (map(<key, value>, <key, value>))

select size(map(1, 'Tom', 2, 'Mary'));

cast (1 as float)

cast ('2015-10-20' as date)

日期函数.

to-date: select to-date('2015-04-13 11:23:11');

year, month, day: select year('{{')), month('{{')), ^{day('{{'))} day('{{'));

week of year: select weekofyear('{{'))

date diff: select datediff('2015-04-23 11:23:11', '2014-04-23 11:23:11')
365

date-add: select date-add('{{', 2), date-sub('{{', 2);

date-sub:

条件函数.

coalesce: 找到第一个不为 null 的列值

case when 条件 then 显值 else 显值 end.

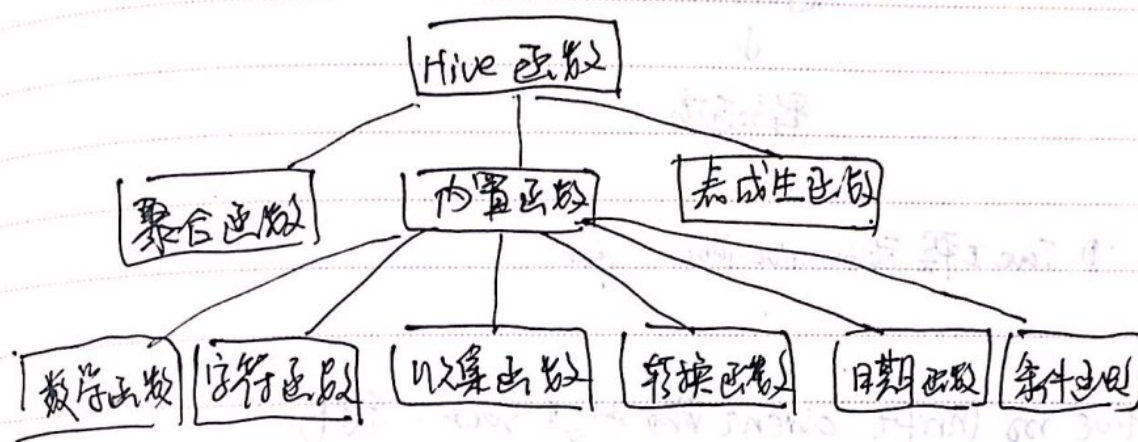


聚合函数:

```
select count(*), max(sal), sum(sal), min(sal), avg(sal) from emp.
```

表生成函数:

```
select explode (map(1, 'Tom', 2, 'Mary', 3, 'Mike'));
```



Hive 表连接:

等值连接

from a, b where a.id = b.id.

不等值连接:

from a, b where a.age between b.losal and s.hisal.

外连接:

新不成立, 但正确. from a ~~b~~ where a right join b on (a.id = b.id)

自连接:

select a.ename, b.ename from emp a, emp b where a.mgr = b.empno.

Hive 子查询:

① select e.ename. from emp e where e.deptno. in (select d.deptno from dept d where d.dname = 'SALES' or d.dname = 'Ac');

② select * from emp e where e.empno not in (select e1.mgr from emp e1 where e1.mgr is not null)

含null, 则查不到值



Hive的JDBC操作

获取连接



创建运行环境



执行SQL



处理结果



释放连接

①. 在工程添加 hive-jdbc-...jar

Hive 的 Thrift client 端创建 socket 连接.

Hive 自定义函数.

① 自定义程序 打包 → jar

② 进入hive的 cli 添加jar hive> add jar [...].jar

③ 创建临时函数

hive> create temporary function <函数名>

as 'Java类名';

hive> drop temporary function <函数名>



Java Demo:

①. 创建 UDF 类.

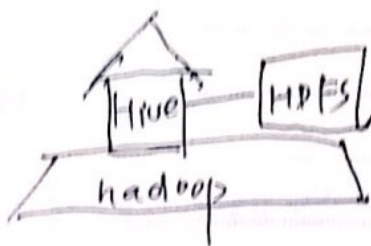
```
public class ConcatString extends UDF {
    public Text evaluate(Text a, Text b) {
        return new Text(a.toString() + "##" + b.toString());
    }
}
```

②. export jar.

③. hbase jar.

> create function temporary
> select myconcat('cell', 'word');

7. 总结



HQL 作业提交

