**大内影视库大数据分析项目实战：**

视频数据字段详解：

视频表

|  |  |  |
| --- | --- | --- |
| 字段 | 备注 | 详细描述 |
| videoId | 视频唯一id（String） | 11位字符串 |
| uploader | 视频上传者（String） | 上传视频的用户名String |
| age | 视频年龄（int） | 视频在平台上的整数天 |
| category | 视频类别（Array<String>） | 上传视频指定的视频分类 |
| length | 视频长度（Int） | 整形数字标识的视频长度 |
| views | 观看次数（Int） | 视频被浏览的次数 |
| rate | 视频评分（Double） | 满分5分 |
| Ratings | 流量（Int） | 视频的流量，整型数字 |
| conments | 评论数（Int） | 一个视频的整数评论数 |
| relatedId | 相关视频id（Array<String>） | 相关视频的id，最多20个 |

用户数据字段详解：

用户表

|  |  |  |
| --- | --- | --- |
| **字段** | **备注** | **字段类型** |
| uploader | 上传者用户名 | string |
| videos | 上传视频数 | int |
| friends | 朋友数量 | int |

1. 数据上传

在hdfs根目录中创建tv总目录，在进入其中分别创建user、video目录，将数据分别上传至目录中

1. 数据清洗
   1. hadoop jar Teduvideo-1.0-SNAPSHOT.jar cn.tedu.Teduvideo.etl.TeduvideoETLDriver /tv/video /tv/etl-video
2. 建表、关联数据

创建原始数据表：tv\_ori，tv\_user\_ori，

创建最终表：tv\_orc，tv\_user\_orc

（1）tv\_ori

create table tv\_ori(

videoId string,

uploader string,

age int,

category array<string>,

length int,

views int,

rate float,

ratings int,

comments int,

relatedId array<string>)

row format delimited fields terminated by "\t"

collection items terminated by "&"

stored as textfile;

（2）创建原始数据表: tv\_user\_ori

create table tv\_user\_ori(

uploader string,

videos int,

friends int)

row format delimited

fields terminated by "\t"

stored as textfile;

1. **创建orc存储格式带snappy压缩的表：**

（1）tv\_orc

create table tv\_orc(

videoId string,

uploader string,

age int,

category array<string>,

length int,

views int,

rate float,

ratings int,

comments int,

relatedId array<string>)

stored as orc

tblproperties("orc.compress"="SNAPPY");

（2）tv\_user\_orc

create table tv\_user\_orc(

uploader string,

videos int,

friends int)

row format delimited

fields terminated by "\t"

stored as orc

tblproperties("orc.compress"="SNAPPY");

（3）向ori表插入数据

load data inpath "/tv/etl-video" into table tv\_ori;

load data inpath "/tv/user" into table tv\_user\_ori;

（4）向orc表插入数据

insert into table tv\_orc select \* from tv\_ori;

insert into table tv\_user\_orc select \* from tv\_user\_ori;

**列转行函数**

EXPLODE(col)：将hive一列中复杂的array或者map结构拆分成多行。

LATERAL VIEW

用法：LATERAL VIEW udtf(expression) tableAlias AS columnAlias

解释：用于和split, explode等UDTF一起使用，它能够将一列数据拆成多行数据，在此基础上可以对拆分后的数据进行聚合。

**2）数据准备**

表6-7 数据准备

|  |  |
| --- | --- |
| movie | category |
| 《疑犯追踪》 | 悬疑,动作,科幻,剧情 |
| 《Lie to me》 | 悬疑,警匪,动作,心理,剧情 |
| 《战狼2》 | 战争,动作,灾难 |

**3）需求**

将电影分类中的数组数据展开。结果如下：

《疑犯追踪》 悬疑

《疑犯追踪》 动作

《疑犯追踪》 科幻

《疑犯追踪》 剧情

《Lie to me》 悬疑

《Lie to me》 警匪

《Lie to me》 动作

《Lie to me》 心理

《Lie to me》 剧情

《战狼2》 战争

《战狼2》 动作

《战狼2》 灾难

**4）创建本地movie.txt，导入数据**

[atguigu@hadoop102 datas]$ vi movie\_info.txt

《疑犯追踪》 悬疑,动作,科幻,剧情

《Lie to me》 悬疑,警匪,动作,心理,剧情

《战狼2》 战争,动作,灾难

**5）创建hive表并导入数据**

create table movie\_info(

movie string,

category string)

row format delimited fields terminated by "\t";

load data local inpath "/opt/module/hive/datas/movie\_info.txt" into table movie\_info;

**6）按需求查询数据**

SELECT movie,category\_name

FROM movie\_info

lateral VIEW

explode(split(category,",")) movie\_info\_tmp AS category\_name ;