# 意向客户主题看板

## 学习目标

了解意向客户主题看板需求

掌握Hive分桶的用法

掌握Map Join的用法

掌握Bucket-Map Join的用法

掌握SMB Join的用法

能够采集意向客户全量数据

能够使用Hive执行计划

能够编写意向客户指标的DWD清洗转换SQL

能够编写意向客户指标的DWM中间层SQL

能够编写意向客户指标的DWS业务层SQL

能够导出分析结果到Mysql

了解拉链表的增量采集导入过程

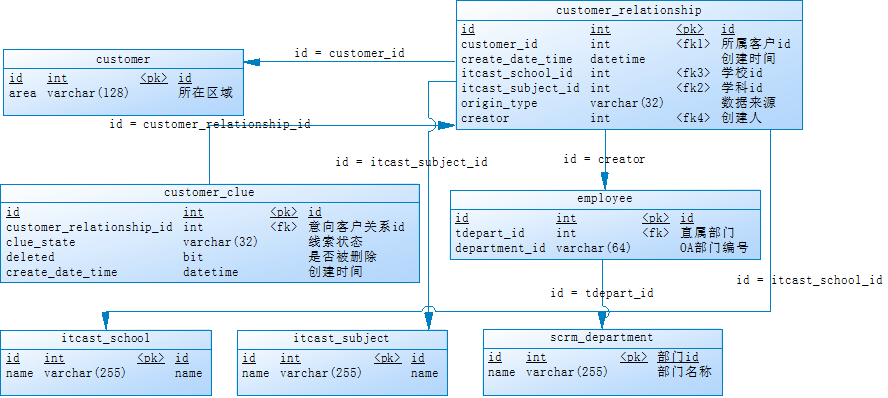
掌握变更数据的增量清洗过程

掌握变更数据的增量分析过程

能够使用Sqoop导出增量数据到Mysql

## 主题需求

包含的指标有：1、总意向量、2、意向学员位置热力图、3、意向学科排名、4、意向校区排名、5、来源渠道占比、6、意向贡献中心占比。



### 总意向量

说明：计期内，新增意向客户（包含自己录入的意向客户）总数。

展现：线状图

条件：年、月、线上线下

维度：年、月、线上线下

指标：总意向客户量

粒度：天，可以下钻到小时数据。

数据来源：客户管理系统的customer\_relationship意向表

SQL：

1. **SELECT**
2. date\_format(
3. cr.create\_date\_time,
4. '%Y-%m-%d'
5. ),
6. count(**DISTINCT** cr.customer\_id)
7. **FROM**
8. customer\_relationship cr
9. **WHERE**
10. cr.create\_date\_time >= '2019-12-01'
11. AND cr.create\_date\_time <= '2019-12-31 23:59:59'
12. **GROUP** **BY**
13. date\_format(
14. cr.create\_date\_time,
15. '%Y-%m-%d'
16. );

### 意向学员位置热力图

说明：统计指定时间段内，新增的意向客户，所在城市区域人数热力图。

展现：地图热力图

维度：年、月、线上线下

指标：按照地区聚合意向客户id数量

粒度：天，可以下钻到小时数据。

条件：年、月、线上线下

数据来源：客户管理系统的customer(客户静态信息表) 、customer\_relationship（客户意向表）

SQL：

1. **SELECT**
2. c.area '区域',
3. count(**DISTINCT** cr.customer\_id) '总数',
4. DATE\_FORMAT(cr.create\_date\_time,'%Y-%m-%d') '客户创建时间'
5. **FROM**
6. customer c, customer\_relationship cr
7. **WHERE** cr.customer\_id = c.id
8. AND cr.create\_date\_time > '2019-11-01 00:00:00'
9. AND cr.create\_date\_time < '2019-11-30 23:59:59'
10. **GROUP** **BY** DATE\_FORMAT(cr.create\_date\_time,'%Y-%m-%d'), c.area
11. **ORDER** **BY** DATE\_FORMAT(cr.create\_date\_time,'%Y-%m-%d') **ASC**, count(1) **DESC**

### 意向学科排名

说明：统计指定时间段内，新增的意向客户中，意向学科人数排行榜。学科名称要关联查询出来。

展现：柱状图

条件：年、月、线上线下

维度：年、月、线上线下、学科

指标：学科意向客户量

粒度：天，可以下钻到小时数据。

数据来源：客户管理系统的customer\_clue（客户线索表）、customer\_relationship（客户意向表）、itcast\_subject（学科表）

SQL：

意向学科，要以意向表的学科字段为准，不能以线索表为准。

1. **SELECT** cr.itcast\_subject\_id,
2. sj.**name**,
3. count(**DISTINCT** cr.customer\_id)
4. **FROM** customer\_clue cc,
5. customer\_relationship cr
6. left join itcast\_subject sj **on** cr.itcast\_subject\_id = sj.id
7. **WHERE** cc.clue\_state = 'VALID\_NEW\_CLUES' --新客户新线索
8. AND ! cc.deleted
9. AND cr.origin\_type IN ('NETSERVICE', 'PRESIGNUP') #线上（排除挖掘录入量）
10. AND cc.create\_date\_time > '2019-10-01 00:00:00'
11. AND cc.create\_date\_time < '2019-11-30 23:59:59'
12. AND cc.customer\_relationship\_id = cr.id
13. **GROUP** **BY** cr.itcast\_subject\_id
14. **ORDER** **BY** count(1) **DESC**;

### 意向校区排名

说明：统计指定时间段内，新增的意向客户中，意向校区人数排行榜。

展现：柱状图

条件：年、月、线上线下

维度：年、月、线上线下、校区

指标：校区意向客户量

粒度：天，可以下钻到小时数据。

数据来源：客户管理系统的

**注意**：学校id，同步时，0和null转换为统一数据，都转换为-1

SQL：

1. **SELECT** cr.itcast\_school\_id,
2. sc.**name**,
3. count(**DISTINCT** cr.customer\_id)
4. **FROM** customer\_clue cc,
5. customer\_relationship cr
6. left join itcast\_school sc **on** cr.itcast\_school\_id = sc.id
7. **WHERE** cc.clue\_state = 'VALID\_NEW\_CLUES' --新客户新线索
8. AND ! cc.deleted
9. AND cr.origin\_type IN ('NETSERVICE', 'PRESIGNUP') #线上（排除挖掘录入量）
10. AND cc.create\_date\_time > '2019-10-01 00:00:00'
11. AND cc.create\_date\_time < '2019-11-30 23:59:59'
12. AND cc.customer\_relationship\_id = cr.id
13. **GROUP** **BY** cr.itcast\_school\_id
14. **ORDER** **BY** count(1) **DESC**;

### 来源渠道占比

说明：统计指定时间段内，新增的意向客户中，不同来源渠道的意向客户占比。

展现：饼状图

条件：年、月、线上线下

维度：年、月、线上线下、来源渠道

粒度：天，可以下钻到小时数据。

指标：来源渠道意向客户量

数据来源：客户管理系统的customer\_clue（客户线索表）、customer\_relationship（客户意向表）

SQL：

1. **SELECT**
2. cr.origin\_type '来源渠道',
3. count(**DISTINCT** cr.customer\_id) '总数'
4. **FROM**
5. customer\_relationship cr
6. LEFT JOIN customer\_clue cc **ON** cc.customer\_relationship\_id = cr.id
7. **WHERE**
8. cc.clue\_state = 'VALID\_NEW\_CLUES'
9. AND cr.create\_date\_time < '2019-11-30 23:59:59'
10. AND cr.create\_date\_time < '2019-11-30 23:59:59'
11. AND cr.origin\_type IN ('NETSERVICE','PRESIGNUP') #线上（排除挖掘录入量）
12. AND ! cc.deleted
13. **GROUP** **BY**
14. cr.origin\_type;

### 意向贡献中心占比

说明：统计指定时间段内，新增的意向客户中，各咨询中心产生的意向客户数占比情况。

展现：饼状图

条件：年、月、线上线下

维度：年、月、线上线下、咨询中心

指标：咨询中心意向客户数

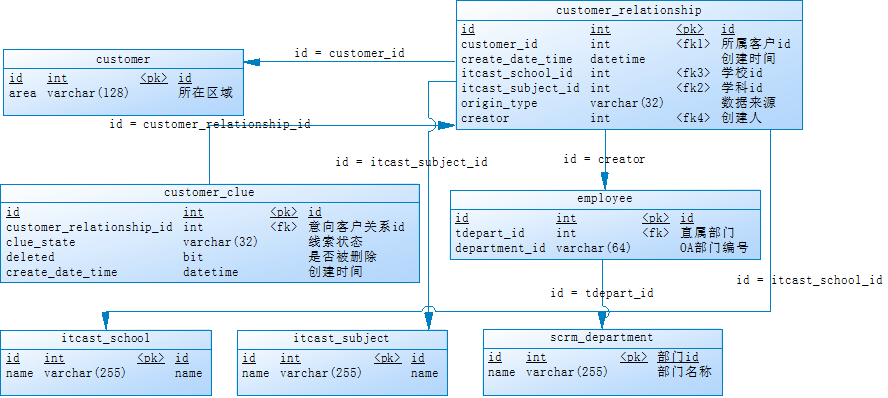
粒度：天，可以下钻到小时数据。

数据来源：客户管理系统的customer\_relationship（客户意向表）、employee（员工表）、scrm\_department（部门表）

SQL：

1. **SELECT**
2. e.tdepart\_id,
3. sd.`**name**`,
4. count(**DISTINCT** cr.customer\_id) '总数'
5. **FROM**
6. customer\_relationship cr
7. LEFT JOIN employee e **ON** cr.creator = e.id
8. LEFT JOIN scrm\_department sd **ON** e.tdepart\_id = sd.id
9. **WHERE**
10. cc.clue\_state = 'VALID\_NEW\_CLUES'
11. AND cr.create\_date\_time >= '2019-10-01 00:00:00'
12. AND cr.create\_date\_time <= '2019-11-30 23:59:59'
13. AND cr.origin\_type IN ('NETSERVICE','PRESIGNUP') #线上（排除挖掘录入量）
14. **GROUP** **BY**
15. e.tdepart\_id;

### 原始数据结构



#### 建库

意向客户数据，来源于咨询管理系统的数据库：scrm。

|  |
| --- |
| create database scrm default character set utf8mb4 collate utf8mb4\_unicode\_ci; |

**测试数据**

Mysql测试数据可以通过导入已准备好的sql文件进行创建：【Home\讲义\完整原始数据\scrm.sql】。可以通过mysql脚本导入：

|  |
| --- |
| mysql -h 192.168.52.150 -P 3306 -uroot -p  source G:\知行教育大数据平台\讲义\完整原始数据\scrm.sql |

#### customer客户静态信息表

主要用来关联获取客户的静态信息，比如地区信息。

|  |
| --- |
| CREATE TABLE `customer` (  **`id`** int(11) NOT NULL AUTO\_INCREMENT,  `customer\_relationship\_id` int(11) DEFAULT NULL COMMENT '当前意向id',  `create\_date\_time` datetime NOT NULL DEFAULT *CURRENT\_TIMESTAMP* COMMENT '创建时间',  `update\_date\_time` timestamp NOT NULL DEFAULT *CURRENT\_TIMESTAMP* ON UPDATE *CURRENT\_TIMESTAMP* COMMENT '最后更新时间',  `deleted` bit(1) NOT NULL DEFAULT b'0' COMMENT '是否被删除（禁用）',  `name` varchar(128) CHARACTER SET utf8 COLLATE utf8\_bin NOT NULL DEFAULT '' COMMENT '姓名',  `idcard` varchar(24) CHARACTER SET utf8 COLLATE utf8\_bin DEFAULT '' COMMENT '身份证号',  `birth\_year` int(5) DEFAULT NULL COMMENT '出生年份',  `gender` varchar(8) CHARACTER SET utf8 COLLATE utf8\_bin DEFAULT 'MAN' COMMENT '性别',  `phone` varchar(24) CHARACTER SET utf8 COLLATE utf8\_bin NOT NULL DEFAULT '' COMMENT '手机号',  `wechat` varchar(32) CHARACTER SET utf8 COLLATE utf8\_bin DEFAULT '' COMMENT '微信',  `qq` varchar(32) CHARACTER SET utf8 COLLATE utf8\_bin DEFAULT '' COMMENT 'qq号',  `email` varchar(56) CHARACTER SET utf8 COLLATE utf8\_bin DEFAULT '' COMMENT '邮箱',  **`area`** varchar(128) DEFAULT '' COMMENT **'所在区域'**,  `leave\_school\_date` date DEFAULT NULL COMMENT '离校时间',  `graduation\_date` date DEFAULT NULL COMMENT '毕业时间',  `bxg\_student\_id` varchar(64) DEFAULT NULL COMMENT '博学谷学员ID，可能未关联到，不存在',  `creator` int(11) DEFAULT NULL COMMENT '创建人ID',  `origin\_type` varchar(32) CHARACTER SET utf8 COLLATE utf8\_bin DEFAULT NULL COMMENT '数据来源',  `origin\_channel` varchar(32) CHARACTER SET utf8 COLLATE utf8\_bin DEFAULT NULL COMMENT '来源渠道',  `tenant` int(11) NOT NULL DEFAULT '0',  `md\_id` int(11) DEFAULT '0' COMMENT '中台id',  PRIMARY KEY (`id`),  KEY `employee\_id` (`creator`) USING BTREE,  KEY `customer\_relationship\_id` (`customer\_relationship\_id`) USING BTREE,  KEY `index\_idcard` (`idcard`) USING BTREE,  KEY `index\_phone` (`phone`) USING BTREE,  KEY `index\_create\_time` (`create\_date\_time`) USING BTREE,  KEY `index\_qq` (`qq`) USING BTREE,  KEY `idx\_update\_time` (`update\_date\_time`) USING BTREE,  CONSTRAINT `customer\_ibfk\_1` FOREIGN KEY (`creator`) REFERENCES `employee` (`id`) ) ENGINE=InnoDB AUTO\_INCREMENT=2061222 DEFAULT CHARSET=utf8; |

#### customer\_relationship客户意向表

意向客户主表，用来统计事实数据。

根据需求，客户的意向数据，会存在更新的情况，需要将更新的数据进行重新统计以得到正确的结果；同时要能够查看这些数据的历史快照。

|  |
| --- |
| CREATE TABLE `customer\_relationship` (  **`id`** int(11) NOT NULL AUTO\_INCREMENT,  **`create\_date\_time`** datetime NOT NULL DEFAULT *CURRENT\_TIMESTAMP*,  `update\_date\_time` timestamp NOT NULL DEFAULT *CURRENT\_TIMESTAMP* ON UPDATE *CURRENT\_TIMESTAMP* COMMENT '最后更新时间',  **`deleted`** bit(1) NOT NULL DEFAULT b'0' COMMENT **'是否被删除（禁用）'**,  **`customer\_id`** int(11) NOT NULL DEFAULT '0' COMMENT **'所属客户id'**,  `first\_id` int(11) DEFAULT NULL COMMENT '第一条客户关系id',  `belonger` int(11) DEFAULT NULL COMMENT '归属人',  `belonger\_name` varchar(10) DEFAULT NULL COMMENT '归属人姓名',  `initial\_belonger` int(11) DEFAULT NULL COMMENT '初始归属人',  `distribution\_handler` int(11) DEFAULT NULL COMMENT '分配处理人',  `business\_scrm\_department\_id` int(11) DEFAULT '0' COMMENT '归属部门',  `last\_visit\_time` datetime DEFAULT NULL COMMENT '最后回访时间',  `next\_visit\_time` datetime DEFAULT NULL COMMENT '下次回访时间',  **`origin\_type`** varchar(32) CHARACTER SET utf8 COLLATE utf8\_bin DEFAULT NULL COMMENT **'数据来源'**,  **`itcast\_school\_id`** int(11) DEFAULT NULL COMMENT **'校区Id'**,  **`itcast\_subject\_id`** int(11) DEFAULT NULL COMMENT **'学科Id'**,  `intention\_study\_type` varchar(32) CHARACTER SET utf8 COLLATE utf8\_bin DEFAULT NULL COMMENT '意向学习方式',  `anticipat\_signup\_date` date DEFAULT NULL COMMENT '预计报名时间',  `level` varchar(8) DEFAULT NULL COMMENT '客户级别',  **`creator`** int(11) DEFAULT NULL COMMENT **'创建人'**,  `current\_creator` int(11) DEFAULT NULL COMMENT '当前创建人：初始==创建人，当在公海拉回时为 拉回人',  `creator\_name` varchar(32) DEFAULT '' COMMENT '创建者姓名',  `origin\_channel` varchar(32) CHARACTER SET utf8 COLLATE utf8\_bin DEFAULT NULL COMMENT '来源渠道',  `comment` varchar(255) CHARACTER SET utf8 COLLATE utf8\_bin DEFAULT '' COMMENT '备注',  `first\_customer\_clue\_id` int(11) DEFAULT '0' COMMENT '第一条线索id',  `last\_customer\_clue\_id` int(11) DEFAULT '0' COMMENT '最后一条线索id',  `process\_state` varchar(32) DEFAULT NULL COMMENT '处理状态',  `process\_time` datetime DEFAULT NULL COMMENT '处理状态变动时间',  `payment\_state` varchar(32) DEFAULT NULL COMMENT '支付状态',  `payment\_time` datetime DEFAULT NULL COMMENT '支付状态变动时间',  `signup\_state` varchar(32) CHARACTER SET utf8 COLLATE utf8\_bin DEFAULT NULL COMMENT '报名状态',  `signup\_time` datetime DEFAULT NULL COMMENT '报名时间',  `notice\_state` varchar(32) DEFAULT NULL COMMENT '通知状态',  `notice\_time` datetime DEFAULT NULL COMMENT '通知状态变动时间',  `lock\_state` bit(1) DEFAULT b'0' COMMENT '锁定状态',  `lock\_time` datetime DEFAULT NULL COMMENT '锁定状态修改时间',  `itcast\_clazz\_id` int(11) DEFAULT NULL COMMENT '所属ems班级id',  `itcast\_clazz\_time` datetime DEFAULT NULL COMMENT '报班时间',  `payment\_url` varchar(1024) DEFAULT '' COMMENT '付款链接',  `payment\_url\_time` datetime DEFAULT NULL COMMENT '支付链接生成时间',  `ems\_student\_id` int(11) DEFAULT NULL COMMENT 'ems的学生id',  `delete\_reason` varchar(64) DEFAULT NULL COMMENT '删除原因',  `deleter` int(11) DEFAULT NULL COMMENT '删除人',  `deleter\_name` varchar(32) DEFAULT NULL COMMENT '删除人姓名',  `delete\_time` datetime DEFAULT NULL COMMENT '删除时间',  `course\_id` int(11) DEFAULT NULL COMMENT '课程ID',  `course\_name` varchar(64) DEFAULT NULL COMMENT '课程名称',  `delete\_comment` varchar(255) DEFAULT '' COMMENT '删除原因说明',  `close\_state` varchar(32) DEFAULT NULL COMMENT '关闭装填',  `close\_time` datetime DEFAULT NULL COMMENT '关闭状态变动时间',  `appeal\_id` int(11) DEFAULT NULL COMMENT '申诉id',  `tenant` int(11) NOT NULL DEFAULT '0' COMMENT '租户',  `total\_fee` decimal(19,0) DEFAULT NULL COMMENT '报名费总金额',  `belonged` int(11) DEFAULT NULL COMMENT '小周期归属人',  `belonged\_time` datetime DEFAULT NULL COMMENT '归属时间',  `belonger\_time` datetime DEFAULT NULL COMMENT '归属时间',  `transfer` int(11) DEFAULT NULL COMMENT '转移人',  `transfer\_time` datetime DEFAULT NULL COMMENT '转移时间',  `follow\_type` int(4) DEFAULT '0' COMMENT '分配类型，0-自动分配，1-手动分配，2-自动转移，3-手动单个转移，4-手动批量转移，5-公海领取',  `transfer\_bxg\_oa\_account` varchar(64) DEFAULT NULL COMMENT '转移到博学谷归属人OA账号',  `transfer\_bxg\_belonger\_name` varchar(64) DEFAULT NULL COMMENT '转移到博学谷归属人OA姓名',  PRIMARY KEY (`id`),  KEY `customer\_id` (`customer\_id`) USING BTREE,  KEY `appeal\_id` (`appeal\_id`) USING BTREE,  KEY `create\_date\_time` (`create\_date\_time`) USING BTREE,  KEY `next\_visit\_time` (`next\_visit\_time`) USING BTREE,  KEY `last\_visit\_time` (`last\_visit\_time`) USING BTREE,  KEY `itcast\_school\_id` (`itcast\_school\_id`) USING BTREE,  KEY `index\_delete` (`delete\_time`) USING BTREE,  KEY `index\_class\_id` (`itcast\_clazz\_id`) USING BTREE,  KEY `belonger` (`belonger`) USING BTREE,  KEY `creator` (`creator`) USING BTREE,  KEY `index\_itcast\_subject\_id` (`itcast\_subject\_id`) USING BTREE,  KEY `idex\_distribution` (`distribution\_handler`) USING BTREE,  CONSTRAINT `customer\_relationship\_ibfk\_1` FOREIGN KEY (`customer\_id`) REFERENCES `customer` (`id`) ) ENGINE=InnoDB AUTO\_INCREMENT=2060127 DEFAULT CHARSET=utf8; |

#### customer\_clue客户线索表

客户线索表主要保存的是客户咨询时留下来的手机号、微信号等联系线索。在意向客户统计时，主要用来判断是新客户还是老客户，clue\_state字段的值'VALID\_NEW\_CLUES'代表是新客户，'VALID\_PUBLIC\_NEW\_CLUE'代表是老客户。

根据需求，客户的线索数据，也会存在更新的情况，需要将更新的数据进行重新统计以得到正确的结果；同时要能够查看这些数据的历史快照。

|  |
| --- |
| CREATE TABLE `customer\_clue` (  **`id`** int(11) NOT NULL AUTO\_INCREMENT,  **`create\_date\_time`** datetime NOT NULL DEFAULT *CURRENT\_TIMESTAMP* COMMENT **'创建时间'**,  `update\_date\_time` timestamp NOT NULL DEFAULT *CURRENT\_TIMESTAMP* ON UPDATE *CURRENT\_TIMESTAMP* COMMENT '最后更新时间',  **`deleted`** bit(1) NOT NULL DEFAULT b'0' COMMENT **'是否被删除（禁用）'**,  `customer\_id` int(11) DEFAULT NULL COMMENT '客户id',  **`customer\_relationship\_id`** int(11) DEFAULT NULL COMMENT **'客户关系id'**,  `session\_id` varchar(48) COLLATE utf8\_bin DEFAULT '' COMMENT '七陌会话id',  `sid` varchar(48) COLLATE utf8\_bin DEFAULT '' COMMENT '访客id',  `status` varchar(16) COLLATE utf8\_bin DEFAULT '' COMMENT '状态（undeal待领取 deal 已领取 finish 已关闭 changePeer 已流转）',  `user` varchar(16) COLLATE utf8\_bin DEFAULT '' COMMENT '所属坐席',  `create\_time` datetime DEFAULT NULL COMMENT '七陌创建时间',  `platform` varchar(16) COLLATE utf8\_bin DEFAULT '' COMMENT '平台来源 （pc-网站咨询|wap-wap咨询|sdk-app咨询|weixin-微信咨询）',  `s\_name` varchar(32) COLLATE utf8\_bin DEFAULT '' COMMENT '用户名称',  `seo\_source` varchar(255) COLLATE utf8\_bin DEFAULT '' COMMENT '搜索来源',  `seo\_keywords` varchar(255) COLLATE utf8\_bin DEFAULT '' COMMENT '关键字',  `ip` varchar(48) COLLATE utf8\_bin DEFAULT '' COMMENT 'IP地址',  `referrer` text COLLATE utf8\_bin COMMENT '上级来源页面',  `from\_url` text COLLATE utf8\_bin COMMENT '会话来源页面',  `landing\_page\_url` text COLLATE utf8\_bin COMMENT '访客着陆页面',  `url\_title` varchar(1024) COLLATE utf8\_bin DEFAULT '' COMMENT '咨询页面title',  `to\_peer` varchar(255) COLLATE utf8\_bin DEFAULT '' COMMENT '所属技能组',  `manual\_time` datetime DEFAULT NULL COMMENT '人工开始时间',  `begin\_time` datetime DEFAULT NULL COMMENT '坐席领取时间 ',  `reply\_msg\_count` int(11) DEFAULT '0' COMMENT '客服回复消息数',  `total\_msg\_count` int(11) DEFAULT '0' COMMENT '消息总数',  `msg\_count` int(11) DEFAULT '0' COMMENT '客户发送消息数',  `comment` varchar(1024) COLLATE utf8\_bin DEFAULT '' COMMENT '备注',  `finish\_reason` varchar(255) COLLATE utf8\_bin DEFAULT '' COMMENT '结束类型',  `finish\_user` varchar(32) COLLATE utf8\_bin DEFAULT '' COMMENT '结束坐席',  `end\_time` datetime DEFAULT NULL COMMENT '会话结束时间',  `platform\_description` varchar(255) COLLATE utf8\_bin DEFAULT '' COMMENT '客户平台信息',  `browser\_name` varchar(255) COLLATE utf8\_bin DEFAULT '' COMMENT '浏览器名称',  `os\_info` varchar(255) COLLATE utf8\_bin DEFAULT '' COMMENT '系统名称',  `area` varchar(255) COLLATE utf8\_bin DEFAULT NULL COMMENT '区域',  `country` varchar(16) COLLATE utf8\_bin DEFAULT '' COMMENT '所在国家',  `province` varchar(16) COLLATE utf8\_bin DEFAULT '' COMMENT '省',  `city` varchar(255) COLLATE utf8\_bin DEFAULT '' COMMENT '城市',  `creator` int(11) DEFAULT '0' COMMENT '创建人',  `name` varchar(64) COLLATE utf8\_bin DEFAULT '' COMMENT '客户姓名',  `idcard` varchar(24) COLLATE utf8\_bin DEFAULT '' COMMENT '身份证号',  `phone` varchar(24) COLLATE utf8\_bin DEFAULT '' COMMENT '手机号',  `itcast\_school\_id` int(11) DEFAULT NULL COMMENT '校区Id',  `itcast\_school` varchar(128) COLLATE utf8\_bin DEFAULT '' COMMENT '校区',  `itcast\_subject\_id` int(11) DEFAULT NULL COMMENT '学科Id',  `itcast\_subject` varchar(128) COLLATE utf8\_bin DEFAULT '' COMMENT '学科',  `wechat` varchar(32) COLLATE utf8\_bin DEFAULT '' COMMENT '微信',  `qq` varchar(32) COLLATE utf8\_bin DEFAULT '' COMMENT 'qq号',  `email` varchar(56) COLLATE utf8\_bin DEFAULT '' COMMENT '邮箱',  `gender` varchar(8) COLLATE utf8\_bin DEFAULT 'MAN' COMMENT '性别',  `level` varchar(8) COLLATE utf8\_bin DEFAULT NULL COMMENT '客户级别',  `origin\_type` varchar(32) COLLATE utf8\_bin DEFAULT '' COMMENT '数据来源渠道',  `information\_way` varchar(32) COLLATE utf8\_bin DEFAULT NULL COMMENT '资讯方式',  `working\_years` date DEFAULT NULL COMMENT '开始工作时间',  `technical\_directions` varchar(255) COLLATE utf8\_bin DEFAULT '' COMMENT '技术方向',  `customer\_state` varchar(32) COLLATE utf8\_bin DEFAULT '' COMMENT '当前客户状态',  `valid` bit(1) DEFAULT b'0' COMMENT '该线索是否是网资有效线索',  `anticipat\_signup\_date` date DEFAULT NULL COMMENT '预计报名时间',  **`clue\_state`** varchar(32) COLLATE utf8\_bin DEFAULT 'NOT\_SUBMIT' COMMENT **'线索状态'**,  `scrm\_department\_id` int(11) DEFAULT NULL COMMENT 'SCRM内部部门id',  `superior\_url` text COLLATE utf8\_bin COMMENT '诸葛获取上级页面URL',  `superior\_source` varchar(1024) COLLATE utf8\_bin DEFAULT NULL COMMENT '诸葛获取上级页面URL标题',  `landing\_url` text COLLATE utf8\_bin COMMENT '诸葛获取着陆页面URL',  `landing\_source` varchar(1024) COLLATE utf8\_bin DEFAULT NULL COMMENT '诸葛获取着陆页面URL来源',  `info\_url` text COLLATE utf8\_bin COMMENT '诸葛获取留咨页URL',  `info\_source` varchar(255) COLLATE utf8\_bin DEFAULT NULL COMMENT '诸葛获取留咨页URL标题',  `origin\_channel` varchar(32) COLLATE utf8\_bin DEFAULT '' COMMENT '投放渠道',  `course\_id` int(32) DEFAULT NULL,  `course\_name` varchar(255) COLLATE utf8\_bin DEFAULT NULL,  `zhuge\_session\_id` varchar(500) COLLATE utf8\_bin DEFAULT NULL,  `is\_repeat` int(4) NOT NULL DEFAULT '0' COMMENT '是否重复线索(手机号维度) 0:正常 1：重复',  `tenant` int(11) NOT NULL DEFAULT '0' COMMENT '租户id',  `activity\_id` varchar(16) COLLATE utf8\_bin DEFAULT NULL COMMENT '活动id',  `activity\_name` varchar(64) COLLATE utf8\_bin DEFAULT NULL COMMENT '活动名称',  `follow\_type` int(4) DEFAULT '0' COMMENT '分配类型，0-自动分配，1-手动分配，2-自动转移，3-手动单个转移，4-手动批量转移，5-公海领取',  `shunt\_mode\_id` int(11) DEFAULT NULL COMMENT '匹配到的技能组id',  `shunt\_employee\_group\_id` int(11) DEFAULT NULL COMMENT '所属分流员工组',  PRIMARY KEY (`id`),  KEY `customer\_id` (`customer\_id`) USING BTREE,  KEY `customer\_relationship\_id` (`customer\_relationship\_id`) USING BTREE,  KEY `phone` (`phone`) USING BTREE,  KEY `idcard` (`idcard`) USING BTREE,  KEY `session\_id` (`session\_id`) USING BTREE,  KEY `index\_date\_time` (`create\_date\_time`) USING BTREE,  KEY `index\_creator` (`creator`) USING BTREE,  CONSTRAINT `customer\_clue\_ibfk\_1` FOREIGN KEY (`customer\_id`) REFERENCES `customer` (`id`),  CONSTRAINT `customer\_clue\_ibfk\_2` FOREIGN KEY (`customer\_relationship\_id`) REFERENCES `customer\_relationship` (`id`) ) ENGINE=InnoDB AUTO\_INCREMENT=2060711 DEFAULT CHARSET=utf8 COLLATE=utf8\_bin; |

#### employee员工表

主要用来关联获取员工信息，比如员工所在的部门id。

|  |
| --- |
| create table employee (  **id** int auto\_increment  primary key,  email varchar(64) not null comment '公司邮箱，OA登录账号',  real\_name varchar(32) not null comment '员工的真实姓名',  phone varchar(32) not null comment '手机号，目前还没有使用；隐私问题OA接口没有提供这个属性，',  **~~department\_id~~** varchar(64) default '0' null comment **~~'OA中的部门编号，有负值'~~**,  department\_name varchar(64) default '' null comment 'OA中的部门名',  remote\_login bit not null comment '员工是否可以远程登录',  job\_number varchar(64) null comment '员工工号',  cross\_school bit not null comment '是否有跨校区权限',  last\_login\_date datetime not null comment '最后登录日期',  creator int(32) null comment '创建人',  create\_date\_time datetime default *CURRENT\_TIMESTAMP* not null comment '创建时间',  update\_date\_time timestamp default *CURRENT\_TIMESTAMP* not null on update *CURRENT\_TIMESTAMP* comment '最后更新时间',  deleted bit default b'0' not null comment '是否被删除（禁用）',  scrm\_department\_id int(32) null comment 'SCRM内部部门id',  leave\_office bit null comment '离职状态',  leave\_office\_time datetime null comment '离职时间',  reinstated\_time datetime null comment '复职时间',  superior\_leaders\_id int null comment '上级领导ID',  **tdepart\_id** int null comment **'直属部门'**,  tenant int default 0 not null,  ems\_user\_name varchar(32) null )  comment '员工信息表'; |

#### scrm\_department部门表

用来获取部门名称等信息。

|  |
| --- |
| **CREATE TABLE** `scrm\_department` (   **`id` int**(11) **NOT NULL AUTO\_INCREMENT COMMENT '部门id'**,  **`name` varchar**(255) **COLLATE** utf8\_bin **DEFAULT NULL COMMENT '部门名称'**,  **`parent\_id` int**(11) **DEFAULT NULL COMMENT '父部门id'**,  **`create\_date\_time` datetime DEFAULT** *CURRENT\_TIMESTAMP* **COMMENT '创建时间'**,  **`update\_date\_time` timestamp NULL DEFAULT** *CURRENT\_TIMESTAMP* **ON UPDATE** *CURRENT\_TIMESTAMP* **COMMENT '更新时间'**,  **`deleted` bit**(1) **DEFAULT b'0' COMMENT '删除标志'**,  **`id\_path` varchar**(1000) **COLLATE** utf8\_bin **DEFAULT NULL COMMENT '编码全路径'**,  **`tdepart\_code` int**(11) **DEFAULT NULL COMMENT '直属部门'**,  **`creator` varchar**(32) **COLLATE** utf8\_bin **DEFAULT NULL COMMENT '创建者'**,  **`depart\_level` int**(4) **DEFAULT NULL COMMENT '部门层级'**,  **`depart\_sign` int**(4) **DEFAULT NULL COMMENT '部门标志，暂时默认1'**,  **`depart\_line` int**(11) **DEFAULT NULL COMMENT '业务线，存储业务线编码'**,  **`depart\_sort` int**(5) **DEFAULT NULL COMMENT '排序字段'**,  **`disable\_flag` int**(1) **DEFAULT NULL COMMENT '禁用标志'**,  **`tenant` int**(11) **NOT NULL DEFAULT '0'**,  **PRIMARY KEY** (**`id`**) ) **ENGINE**=InnoDB **AUTO\_INCREMENT**=149 **DEFAULT CHARSET**=utf8 **COLLATE**=utf8\_bin; |

#### itcast\_school学校表

用来获取学校名称等信息。

|  |
| --- |
| **CREATE TABLE** `itcast\_school` (  **`id` int**(11) **NOT NULL AUTO\_INCREMENT**,  **`create\_date\_time` datetime NOT NULL DEFAULT** *CURRENT\_TIMESTAMP* **COMMENT '创建时间'**,  **`update\_date\_time` timestamp NOT NULL DEFAULT** *CURRENT\_TIMESTAMP* **ON UPDATE** *CURRENT\_TIMESTAMP* **COMMENT '最后更新时间'**,  **`deleted` bit**(1) **NOT NULL DEFAULT b'0' COMMENT '是否被删除（禁用）'**,  **`name` varchar**(32) **COLLATE** utf8\_bin **NOT NULL DEFAULT '' COMMENT '校区名称'**,  **`code` varchar**(32) **COLLATE** utf8\_bin **NOT NULL**,  **`tenant` int**(11) **NOT NULL DEFAULT '0'**,  **PRIMARY KEY** (**`id`**) ) **ENGINE**=InnoDB **AUTO\_INCREMENT**=30 **DEFAULT CHARSET**=utf8 **COLLATE**=utf8\_bin; |

#### itcast\_subject学科表

用来获取学科名称等信息。

|  |
| --- |
| **CREATE TABLE** `itcast\_subject` (  **`id` int**(11) **NOT NULL AUTO\_INCREMENT**,  **`create\_date\_time` datetime NOT NULL COMMENT '创建时间'**,  **`update\_date\_time` timestamp NOT NULL DEFAULT** *CURRENT\_TIMESTAMP* **ON UPDATE** *CURRENT\_TIMESTAMP* **COMMENT '最后更新时间'**,  **`deleted` bit**(1) **NOT NULL DEFAULT b'0' COMMENT '是否被删除（禁用）'**,  **`name` varchar**(32) **COLLATE** utf8\_bin **DEFAULT '' COMMENT '学科名称'**,  **`code` varchar**(32) **COLLATE** utf8\_bin **DEFAULT NULL**,  **`tenant` int**(11) **NOT NULL DEFAULT '0'**,  **PRIMARY KEY** (**`id`**) )  **ENGINE**=InnoDB **AUTO\_INCREMENT**=22 **DEFAULT CHARSET**=utf8 **COLLATE**=utf8\_bin; |

## 建模分析

### 指标和维度

根据主题需求，我们来进行指标和维度的提取：

从1.1~1.6统计的分别是，地区意向客户、总意向客户数、学科意向客户、校区意向客户、来源渠道意向客户和咨询中心意向客户，维度都包含了年、月、线上线下。

每个指标都指明统计的是新增客户，我们可以将数据分为新客户和老客户进行统计。

我们可以提取出共有的指标：意向客户量。维度：年、月、线上线下、新老客户。

因为数据粒度都是展示到天，而且可以下钻到小时，所以我们的统计维度中也需要增加天和小时。

不同指标的产品属性也需要增加到维度中：

意向学员位置热力图，是将不同地区的意向客户数量进行统计；

意向学科排名，虽然最终要的结果是学科的排名，但这个排名的依据是根据学科统计出来的意向学员数量；

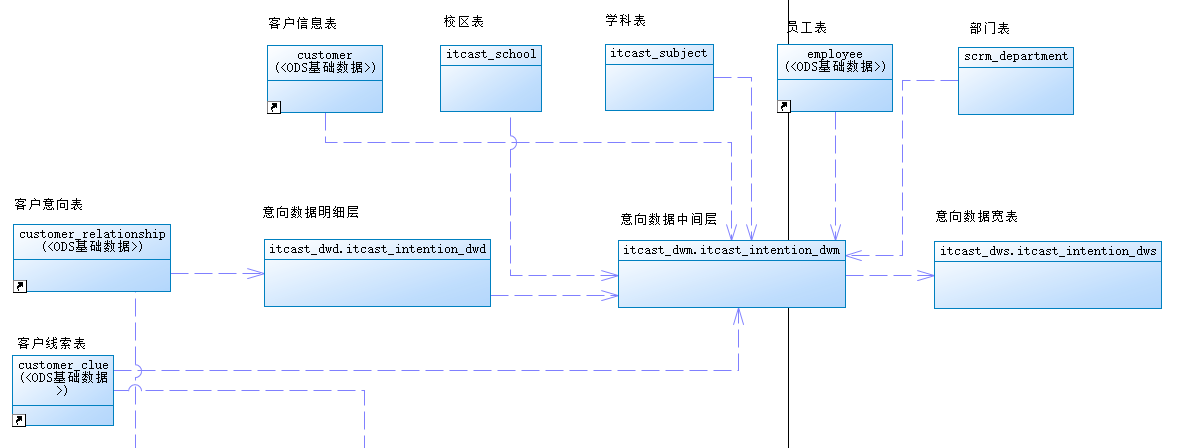
意向校区排名，要的结果是校区排名，但排名的依据也是根据校区统计出来的意向学员数量；

来源渠道占比，指的是不同来源渠道意向学员数量的总体占比，底层的依据还是意向学员数量；

意向贡献中心占比，和来源渠道占比类似，依据的是不同咨询中心的意向学员数量；

所以**维度应该包括：年、月、天、小时、线上线下、新老客户、地区、学科、校区、来源渠道、咨询中心。**

### 分层设计



我们可以采取结果导向的方式来进行倒推：

1. 最终需要统计的数据维度：年、月、天、小时、线上线下、新老客户、地区、学科、校区、来源渠道、咨询中心；
2. 在需求中，每个指标的条件都包含有时间和线上线下、新老客户，也就是说无论哪一种业务维度都需要按照时间、线上线下和新老客户来进行区分，可以将这三个维度作为单独字段；
3. 因此我们将维度分为四类：时间维度（年、月、天）、数据来源（线上线下）、客户属性（新老客户）和产品属性维度（总意向量、地区、学科、校区、来源渠道、咨询中心）；
4. 首先将数据抽取到ODS源数据层，然后将明细数据通过清洗转换后存入DWD层；
5. 在DWM，关联相关的维度数据，并转换出需要的信息；
6. DWS层在DWM关联后的数据上进行统计，得出数据集市；
7. 将OLAP需要的数据和字段同步至mysql；
8. ODS——》DWD——》DWM——》DWS。

## 实现

### 建模

#### 指标和维度

指标：意向客户量是单位时间内新增的意向客户量（包含线上线下），以天为单位显示。

维度：

* 时间维度：年、月、天、小时
* 数据来源：线上线下
* 客户属性：新客户、老客户
* 地区、学科、校区、来源渠道、咨询中心。

#### 事实表和维度表

customer\_relationship客户意向表，包含了意向客户信息；显然此表就是意向客户指标的基础事实。

customer客户静态信息表主要用来关联获取客户的静态信息，比如地区信息。是我们的维度数据。

customer\_clue客户线索表主要用来判断是新客户还是老客户；也属于要关联的维度信息；但因为此表包含了后续其他指标的事实数据，所以不放在维度DIM层。

类似的，employee员工表、scrm\_department部门表、itcast\_school学校表、itcast\_subject学科表都属于维度信息，所以作为维度表放在维度层。

#### Hive分桶

分桶是将数据集分解成更容易管理的若干部分的一个技术，是比分区更为细粒度的数据范围划分。

##### 为什么要分桶？

###### 获得更高的查询处理效率

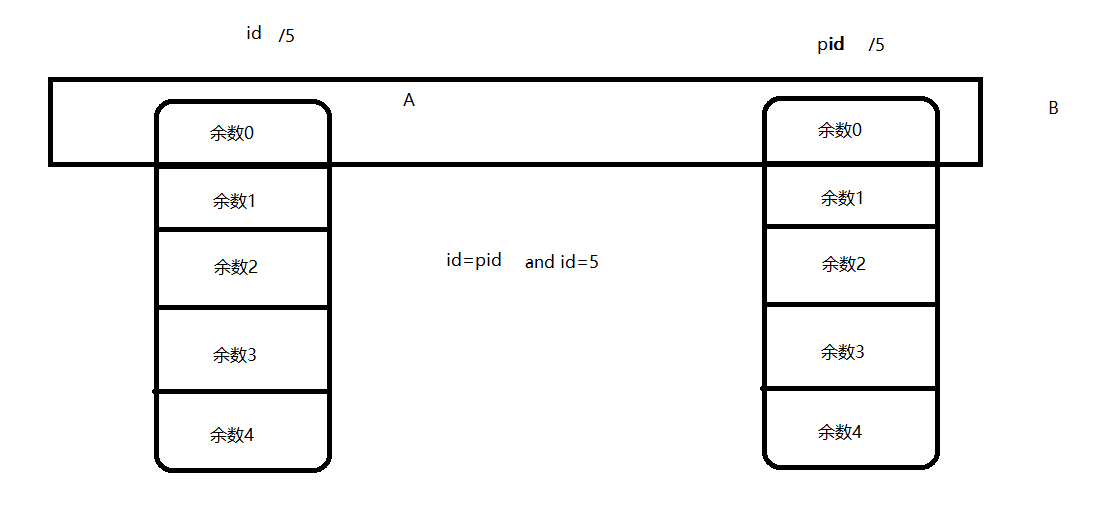
在分区数量过于庞大以至于可能导致文件系统崩溃时，或数据集找不到合理的分区字段时，我们就需要使用分桶来解决问题了。

分区中的数据可以被进一步拆分成桶，不同于分区对列直接进行拆分，桶往往使用列的哈希值对数据打散，并分发到各个不同的桶中从而完成数据的分桶过程。

注意，hive使用对分桶所用的值进行hash，并用hash结果除以桶的个数做取余运算的方式来分桶，保证了每个桶中都有数据，但每个桶中的数据条数不一定相等。

如果另外一个表也按照同样的规则分成了一个个小文件。两个表join的时候，就不必要扫描整个表，只需要匹配相同分桶的数据即可，从而提升效率。

在数据量足够大的情况下，分桶比分区有更高的查询效率。



###### 数据采样

在真实的大数据分析过程中，由于数据量较大，开发和自测的过程比较慢，严重影响系统的开发进度。此时就可以使用分桶来进行数据采样。采样使用的是一个具有代表性的查询结果而不是全部结果，通过对采样数据的分析，来达到快速开发和自测的目的，节省大量的研发成本。

##### 分桶和分区的区别

1. 分桶对数据的处理比分区更加细粒度化：分区针对的是数据的存储路径；分桶针对的是数据文件；
2. 分桶是按照列的哈希函数进行分割的，相对比较平均；而分区是按照列的值来进行分割的，容易造成数据倾斜；
3. 分桶和分区两者不干扰，可以把分区表进一步分桶。

##### 操作

1. **创建分桶表**

|  |
| --- |
| create table test\_buck(id int, name string)  clustered by(id) sorted by (id asc) into 6 buckets  row format delimited fields terminated by '\t'; |

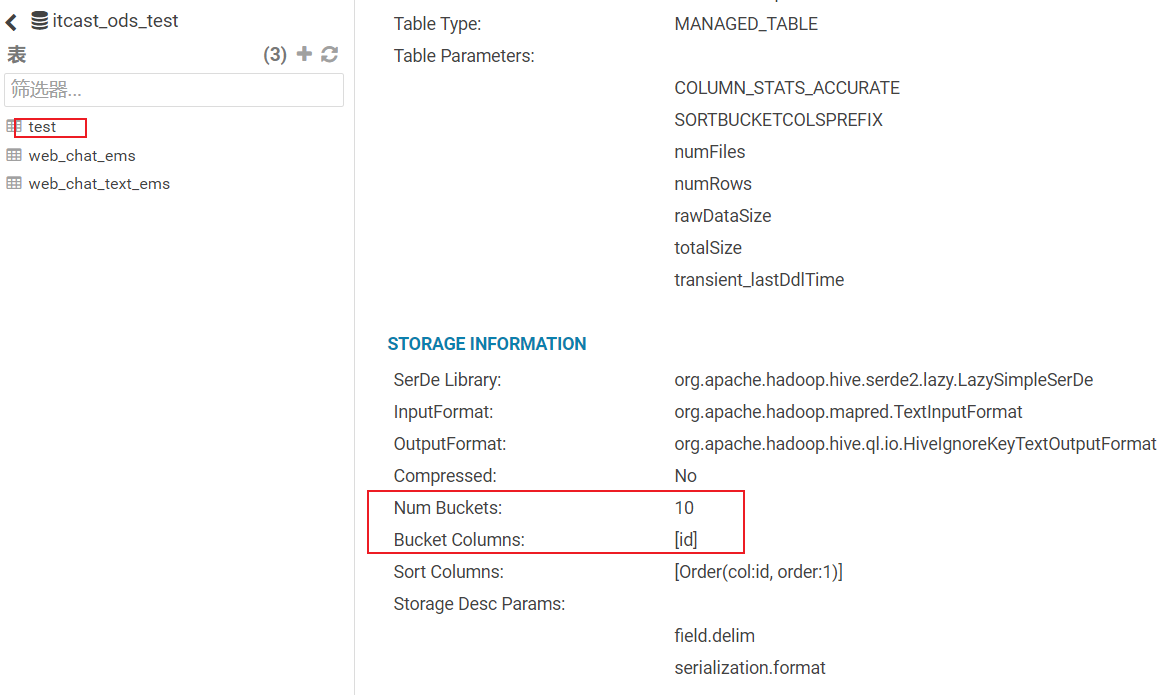
CLUSTERED BY来指定划分桶所用列；

SORTED BY对桶中的一个或多个列进行排序；

into 6 buckets指定划分桶的个数。

分桶规则：HIVE对key的hash值除bucket个数取余数，保证数据均匀随机分布在所有bucket里。

查看分桶表信息



|  |
| --- |
| desc formatted test\_buck;  20200330184514932_副本 |

1. **插入数据**

|  |
| --- |
| --启用桶表  set hive.enforce.bucketing=true;  insert into table test\_buck select id, name from temp\_buck; |

hive.enforce.bucketing：启用桶表，数据分桶是否被强制执行，默认false，如果开启，则写入table数据时会启动分桶。

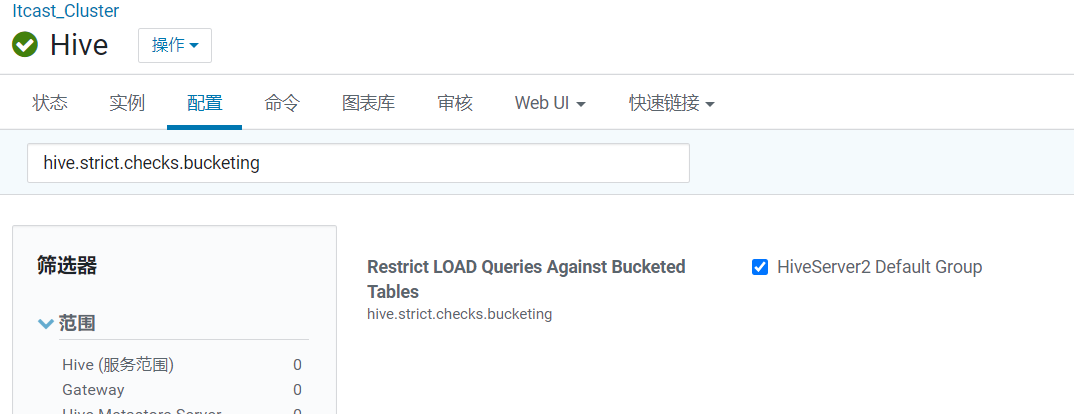
##### 文本数据处理

注意：对于分桶表，不能使用load data的方式进行数据插入操作，因为load data导入的数据不会有分桶结构。

如何避免针对桶表使用load data插入数据的误操作呢？

|  |
| --- |
| --限制对桶表进行load操作  set hive.strict.checks.bucketing = true; |

也可以在CM的hive配置项中修改此配置，当针对桶表执行load data操作时会报错。



那么对于文本数据如何处理呢？

1. 先创建临时表，通过load data将txt文本导入临时表。

|  |
| --- |
| --创建临时表  create table temp\_buck(id int, name string)  row format delimited fields terminated by '\t';  --导入数据  load data local inpath '/tools/test\_buck.txt' into table temp\_buck; |

1. 使用insert select语句间接的把数据从临时表导入到分桶表。

|  |
| --- |
| --启用桶表  set hive.enforce.bucketing=true;  --限制对桶表进行load操作  set hive.strict.checks.bucketing = true;  --insert select  insert into table test\_buck select id, name from temp\_buck;  --分桶成功  图片1_副本 |

##### 数据采样

对表分桶一般有两个目的，提高数据查询效率、抽样调查。通过前面的讲解，我们已经可以对分桶表进行正常的创建并导入数据了。一般在实际生产中，对于非常大的数据集，有时用户需要使用的是一个具有代表性的查询结果而不是全部结果，比如在开发自测的时候。这个时候Hive就可以通过对表进行抽样来满足这个需求。

**语法**

|  |
| --- |
| select \* from table tablesample(bucket x out of y on column) |

hive根据y的大小，决定抽样的比例。y必须是table总bucket数的倍数或者因子。

例如，table总共分了10份bucket，当y=2时，抽取(10/2=)5个bucket的数据，当y=10时，抽取(10/10=)1个bucket的数据。

x表示从哪个bucket开始抽取，如果需要取多个分区，以后的分区号为当前分区号加上y。

例如，table总bucket数为6，tablesample(bucket 1 out of 2)，表示总共抽取（6/2=）3个bucket的数据，从第1个bucket开始，抽取第1(x)个和第3(x+y)个和第5(x+y)个bucket的数据。

注意：x的值必须小于等于y的值。否则会抛出异常：FAILED: SemanticException [Error 10061]: Numerator should not be bigger than denominator in sample clause for table stu\_buck。

**栗子**

|  |
| --- |
| select \* from test\_buck tablesample(bucket 1 out of 10 on id); |

注意：sqoop不支持分桶表，如果需要从sqoop导入数据到分桶表，可以通过中间临时表进行过度。ODS也可以不做分桶，从DWD明细层开始分桶。

##### Map Join

MapJoin顾名思义，就是在Map阶段进行表之间的连接。而不需要进入到Reduce阶段才进行连接。这样就节省了在Shuffle阶段时要进行的大量数据传输。从而起到了优化作业的作用。

要使MapJoin能够顺利进行，那就必须满足这样的条件：除了一份表的数据分布在不同的Map中外，其他连接的表的数据必须在每个Map中有完整的拷贝。

所以并不是所有的场景都适合用MapJoin。它通常会用在如下的一些情景：在二个要连接的表中，有一个很大，有一个很小，这个小表可以存放在内存中而不影响性能。

这样我们就把小表文件复制到每一个Map任务的本地，再让Map把文件读到内存中待用。

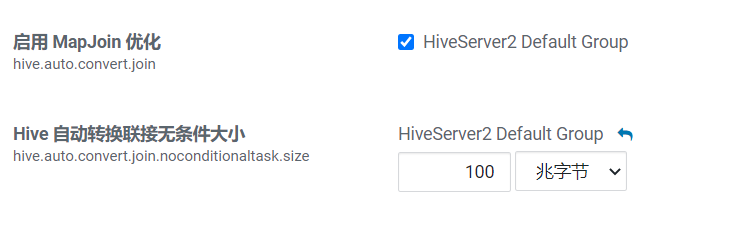
在Hive v0.7之前，需要使用hint提示 /\*+ mapjoin(table) \*/才会执行MapJoin。Hive v0.7之后的版本已经不需要给出MapJoin的指示就进行优化。现在可以通过如下配置参数来进行控制：

|  |
| --- |
| set hive.auto.convert.join=true; |

Hive还提供另外一个参数--表文件的大小作为开启和关闭MapJoin的阈值：

|  |
| --- |
| --旧版本为hive.mapjoin.smalltable.filesize  set hive.auto.convert.join.noconditionaltask.size=512000000 |

**注意**，如果hive.auto.convert.join是关闭的，则本参数不起作用。否则，如果参与连接的N个表(或分区)中的N-1个 的总大小小于512MB，则直接将连接转为Map连接。默认值为20MB。



MapJoin的使用场景：

1. 关联操作中有一张表非常小

2. 不等值的链接操作

###### 大小表关联

|  |
| --- |
| select f.a,f.b from A t join B f on ( f.a=t.a and f.ftime=20110802) |

该语句中B表有30亿行记录，A表只有100行记录，而且B表中数据倾斜特别严重，有一个key上有15亿行记录，在运行过程中特别的慢，而且在reduece的过程中遇到执行时间过长或者内存不够的问题。

MAPJION会把小表全部读入内存中，在map阶段直接拿另外一个表的数据和内存中表数据做匹配，由于在map时进行了join操作，省去了reduce运行的效率会高很多。

这样就不会由于数据倾斜导致某个reduce上落数据太多而失败。于是原来的sql可以通过使用hint的方式指定join时使用mapjoin。

|  |
| --- |
| select /\*+ mapjoin(A)\*/ f.a,f.b from A t join B f on ( f.a=t.a and f.ftime=20110802) |

在实际使用中，只要根据业务调整小表的阈值即可，hive会自动帮我们完成mapjoin，提高执行的效率。

###### 不等连接

mapjoin还有一个很大的好处是能够进行不等连接的join操作，如果将不等条件写在where中，那么mapreduce过程中会进行笛卡尔积，运行效率特别低，如果使用mapjoin操作，在map的过程中就完成了不等值的join操作，效率会高很多。

|  |
| --- |
| select A.a ,A.b from A join B where A.a**>**B.a |

##### Bucket-MapJoin

###### 作用

两个表join的时候，小表不足以放到内存中，但是又想用map side join这个时候就要用到bucket Map join。其方法是两个join表在join key上都做hash bucket，并且把你打算复制的那个（相对）小表的bucket数设置为大表的倍数。这样数据就会按照key join，做hash bucket。小表依然复制到所有节点，Map join的时候，小表的每一组bucket加载成hashtable，与对应的一个大表bucket做局部join，这样每次只需要加载部分hashtable就可以了。

###### 条件

1） set hive.optimize.bucketmapjoin = true;  
2） 一个表的bucket数是另一个表bucket数的整数倍  
3） bucket列 == join列  
4） 必须是应用在map join的场景中

注意：如果表不是bucket的，则只是做普通join。

##### SMB Join

全称Sort Merge Bucket Join。

###### 作用

大表对小表应该使用MapJoin来进行优化，但是如果是大表对大表，如果进行shuffle，那就非常可怕，第一个慢不用说，第二个容易出异常，此时就可以使用SMB Join来提高性能。SMB Join基于bucket-mapjoin的**有序bucket**，可实现在map端完成join操作，可以有效地减少或避免shuffle的数据量。SMB join的条件和Map join类似但又不同。

###### 条件

|  |  |
| --- | --- |
| **bucket mapjoin** | **SMB join** |
| set hive.optimize.bucketmapjoin = true; | set hive.optimize.bucketmapjoin = true;  set hive.auto.convert.sortmerge.join=true;  set hive.optimize.bucketmapjoin.sortedmerge = true;  set hive.auto.convert.sortmerge.join.noconditionaltask=true; |
| 一个表的bucket数是另一个表bucket数的整数倍 | 小表的bucket数**=**大表bucket数 |
| bucket列 == join列 | Bucket 列 == Join 列 == **sort 列** |
| 必须是应用在map join的场景中 | 必须是应用在bucket mapjoin 的场景中 |

###### 确保分同列排序

hive并不检查两个join的表是否已经做好bucket且sorted，需要用户自己去保证join的表数据sorted，否则可能数据不正确。

有两个办法：

1）hive.enforce.sorting 设置为 true。开启强制排序时，插数据到表中会进行强制排序，默认false。

2）插入数据时通过在sql中用distributed c1 sort by c1 或者 cluster by c1

另外，表创建时必须是CLUSTERED且SORTED，如下：

|  |
| --- |
| create table test\_smb\_2(mid string,age\_id string)  CLUSTERED BY(mid) SORTED BY(mid) INTO 500 BUCKETS; |

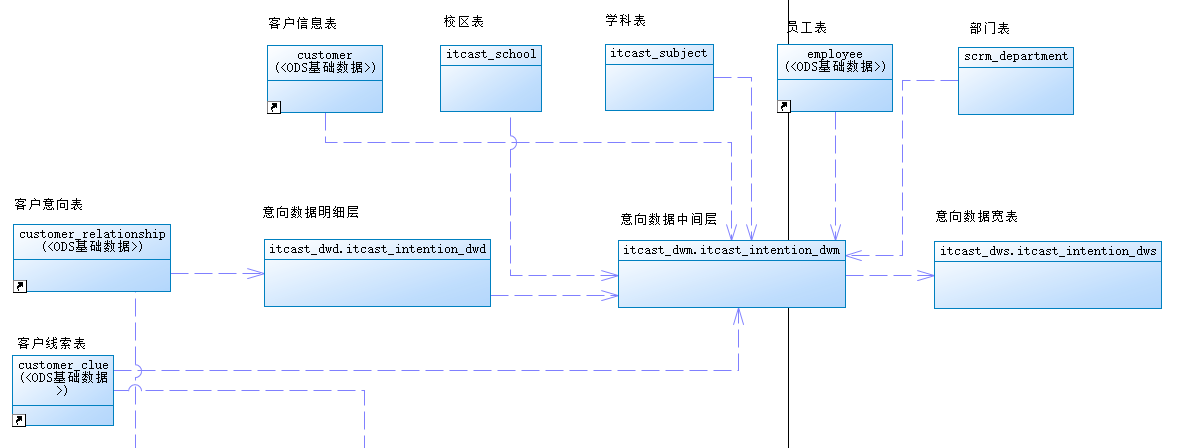
综上，涉及到分桶表操作的齐全配置为：

|  |
| --- |
| --写入数据强制分桶  set hive.enforce.bucketing=true;  --写入数据强制排序  set hive.enforce.sorting=true;  --开启bucketmapjoin  set hive.optimize.bucketmapjoin = true;  --开启SMB Join  set hive.auto.convert.sortmerge.join=true;  set hive.auto.convert.sortmerge.join.noconditionaltask=true; |

开启MapJoin的配置（hive.auto.convert.join和hive.auto.convert.join.noconditionaltask.size），还有限制对桶表进行load操作（hive.strict.checks.bucketing）可以直接设置在hive的配置项中，无需在sql中声明。

自动尝试SMB联接（hive.optimize.bucketmapjoin.sortedmerge）也可以在设置中进行提前配置。

#### 分层



##### ODS

写入时压缩生效

|  |
| --- |
| set hive.exec.orc.compression.strategy=COMPRESSION; |

**拉链表**：意向客户看板中，对意向数据有新的需求：将customer\_relationship的数据更新涉及到的维度按照最新值重新统计（比如2020年7月份的数据有修改更新，则需要将7月份的统计数据重新计算）；同时要有历史快照。

此时需要使用缓慢渐变维，推荐采用SCD2拉链表的形式来做，既能满足数据更新的需求，又能满足数据历史快照的需求。需要在start\_time字段的基础上，增加新的end\_time字段，以标识封链时间。

**内外部表**：ODS层是原始数据，一般不允许修改，所以使用外部表保证数据的安全性，避免误删除；ODS中的customer\_relationship客户意向表和customer\_clue客户线索表，因为使用拉链表需要覆盖操作，所以没有定义为外部表。

**分桶采集**：sqoop不支持分桶表，如果需要从sqoop导入数据到分桶表，需要通过中间临时表进行过度。也可以ODS不做分桶，从DWD明细层开始分桶。

**分桶关联与采样**：ODS层的customer\_relationship客户意向表和customer\_clue客户线索表是存在关联关系的，**customer\_relationship通过 id 关联customer\_clue表的 customer\_relationship\_id** ，可以获取新老客户信息。因此我们将这两个字段作为**分桶**字段。可用于数据采样和MapJoin。

**分区**：在之前的访问咨询主题看板中，为了便于后续T+1抽取数据时，方便获取昨天的数据，ODS模型要在原始mysql表的基础之上增加start\_time字段，并且可以使用start\_time字段做分区以提升查询的性能。

###### customer\_relationship客户意向表

|  |
| --- |
| DROP TABLE itcast\_ods.`customer\_relationship`; CREATE TABLE IF NOT EXISTS itcast\_ods.`customer\_relationship` (  `id` int COMMENT '客户关系id',  `create\_date\_time` STRING COMMENT '创建时间',  `update\_date\_time` STRING COMMENT '最后更新时间',  `deleted` int COMMENT '是否被删除（禁用）',  `customer\_id` int COMMENT '所属客户id',  `first\_id` int COMMENT '第一条客户关系id',  `belonger` int COMMENT '归属人',  `belonger\_name` STRING COMMENT '归属人姓名',  `initial\_belonger` int COMMENT '初始归属人',  `distribution\_handler` int COMMENT '分配处理人',  `business\_scrm\_department\_id` int COMMENT '归属部门',  `last\_visit\_time` STRING COMMENT '最后回访时间',  `next\_visit\_time` STRING COMMENT '下次回访时间',  `origin\_type` STRING COMMENT '数据来源',  `itcast\_school\_id` int COMMENT '校区Id',  `itcast\_subject\_id` int COMMENT '学科Id',  `intention\_study\_type` STRING COMMENT '意向学习方式',  `anticipat\_signup\_date` STRING COMMENT '预计报名时间',  `level` STRING COMMENT '客户级别',  `creator` int COMMENT '创建人',  `current\_creator` int COMMENT '当前创建人：初始==创建人，当在公海拉回时为 拉回人',  `creator\_name` STRING COMMENT '创建者姓名',  `origin\_channel` STRING COMMENT '来源渠道',  `comment` STRING COMMENT '备注',  `first\_customer\_clue\_id` int COMMENT '第一条线索id',  `last\_customer\_clue\_id` int COMMENT '最后一条线索id',  `process\_state` STRING COMMENT '处理状态',  `process\_time` STRING COMMENT '处理状态变动时间',  `payment\_state` STRING COMMENT '支付状态',  `payment\_time` STRING COMMENT '支付状态变动时间',  `signup\_state` STRING COMMENT '报名状态',  `signup\_time` STRING COMMENT '报名时间',  `notice\_state` STRING COMMENT '通知状态',  `notice\_time` STRING COMMENT '通知状态变动时间',  `lock\_state` STRING COMMENT '锁定状态',  `lock\_time` STRING COMMENT '锁定状态修改时间',  `itcast\_clazz\_id` int COMMENT '所属ems班级id',  `itcast\_clazz\_time` STRING COMMENT '报班时间',  `payment\_url` STRING COMMENT '付款链接',  `payment\_url\_time` STRING COMMENT '支付链接生成时间',  `ems\_student\_id` int COMMENT 'ems的学生id',  `delete\_reason` STRING COMMENT '删除原因',  `deleter` int COMMENT '删除人',  `deleter\_name` STRING COMMENT '删除人姓名',  `delete\_time` STRING COMMENT '删除时间',  `course\_id` int COMMENT '课程ID',  `course\_name` STRING COMMENT '课程名称',  `delete\_comment` STRING COMMENT '删除原因说明',  `close\_state` STRING COMMENT '关闭装填',  `close\_time` STRING COMMENT '关闭状态变动时间',  `appeal\_id` int COMMENT '申诉id',  `tenant` int COMMENT '租户',  `total\_fee` DECIMAL COMMENT '报名费总金额',  `belonged` int COMMENT '小周期归属人',  `belonged\_time` STRING COMMENT '归属时间',  `belonger\_time` STRING COMMENT '归属时间',  `transfer` int COMMENT '转移人',  `transfer\_time` STRING COMMENT '转移时间',  `follow\_type` int COMMENT '分配类型，0-自动分配，1-手动分配，2-自动转移，3-手动单个转移，4-手动批量转移，5-公海领取',  `transfer\_bxg\_oa\_account` STRING COMMENT '转移到博学谷归属人OA账号',  `transfer\_bxg\_belonger\_name` STRING COMMENT '转移到博学谷归属人OA姓名',  `end\_time` STRING COMMENT '有效截止时间') comment '客户关系表' PARTITIONED BY(start\_time STRING) clustered by(id) sorted by(id) into 10 buckets ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' stored as orc TBLPROPERTIES ('orc.compress'='ZLIB'); |

###### customer\_clue客户线索表

使用start\_time字段分区以提升条件查询性能。customer\_clue是后面有效线索主题看板的事实表，需求也要求将数据更新涉及到的维度按照最新值重新统计、要有历史快照。采用拉链表(SCD2)的形式来做，增加新的end\_time字段，以标识封链时间。

|  |
| --- |
| DROP TABLE itcast\_ods.customer\_clue; CREATE TABLE IF NOT EXISTS itcast\_ods.customer\_clue (  id int COMMENT 'customer\_clue\_id',  create\_date\_time STRING COMMENT '创建时间',  update\_date\_time STRING COMMENT '最后更新时间',  deleted STRING COMMENT '是否被删除（禁用）',  customer\_id int COMMENT '客户id',  customer\_relationship\_id int COMMENT '客户关系id',  session\_id STRING COMMENT '七陌会话id',  sid STRING COMMENT '访客id',  status STRING COMMENT '状态（undeal待领取 deal 已领取 finish 已关闭 changePeer 已流转）',  users STRING COMMENT '所属坐席',  create\_time STRING COMMENT '七陌创建时间',  platform STRING COMMENT '平台来源 （pc-网站咨询|wap-wap咨询|sdk-app咨询|weixin-微信咨询）',  s\_name STRING COMMENT '用户名称',  seo\_source STRING COMMENT '搜索来源',  seo\_keywords STRING COMMENT '关键字',  ip STRING COMMENT 'IP地址',  referrer STRING COMMENT '上级来源页面',  from\_url STRING COMMENT '会话来源页面',  landing\_page\_url STRING COMMENT '访客着陆页面',  url\_title STRING COMMENT '咨询页面title',  to\_peer STRING COMMENT '所属技能组',  manual\_time STRING COMMENT '人工开始时间',  begin\_time STRING COMMENT '坐席领取时间 ',  reply\_msg\_count int COMMENT '客服回复消息数',  total\_msg\_count int COMMENT '消息总数',  msg\_count int COMMENT '客户发送消息数',  comment STRING COMMENT '备注',  finish\_reason STRING COMMENT '结束类型',  finish\_user STRING COMMENT '结束坐席',  end\_time STRING COMMENT '会话结束时间',  platform\_description STRING COMMENT '客户平台信息',  browser\_name STRING COMMENT '浏览器名称',  os\_info STRING COMMENT '系统名称',  area STRING COMMENT '区域',  country STRING COMMENT '所在国家',  province STRING COMMENT '省',  city STRING COMMENT '城市',  creator int COMMENT '创建人',  name STRING COMMENT '客户姓名',  idcard STRING COMMENT '身份证号',  phone STRING COMMENT '手机号',  itcast\_school\_id int COMMENT '校区Id',  itcast\_school STRING COMMENT '校区',  itcast\_subject\_id int COMMENT '学科Id',  itcast\_subject STRING COMMENT '学科',  wechat STRING COMMENT '微信',  qq STRING COMMENT 'qq号',  email STRING COMMENT '邮箱',  gender STRING COMMENT '性别',  level STRING COMMENT '客户级别',  origin\_type STRING COMMENT '数据来源渠道',  information\_way STRING COMMENT '资讯方式',  working\_years STRING COMMENT '开始工作时间',  technical\_directions STRING COMMENT '技术方向',  customer\_state STRING COMMENT '当前客户状态',  valid STRING COMMENT '该线索是否是网资有效线索',  anticipat\_signup\_date STRING COMMENT '预计报名时间',  clue\_state STRING COMMENT '线索状态',  scrm\_department\_id int COMMENT 'SCRM内部部门id',  superior\_url STRING COMMENT '诸葛获取上级页面URL',  superior\_source STRING COMMENT '诸葛获取上级页面URL标题',  landing\_url STRING COMMENT '诸葛获取着陆页面URL',  landing\_source STRING COMMENT '诸葛获取着陆页面URL来源',  info\_url STRING COMMENT '诸葛获取留咨页URL',  info\_source STRING COMMENT '诸葛获取留咨页URL标题',  origin\_channel STRING COMMENT '投放渠道',  course\_id int COMMENT '课程编号',  course\_name STRING COMMENT '课程名称',  zhuge\_session\_id STRING COMMENT 'zhuge会话id',  is\_repeat int COMMENT '是否重复线索(手机号维度) 0:正常 1：重复',  tenant int COMMENT '租户id',  activity\_id STRING COMMENT '活动id',  activity\_name STRING COMMENT '活动名称',  follow\_type int COMMENT '分配类型，0-自动分配，1-手动分配，2-自动转移，3-手动单个转移，4-手动批量转移，5-公海领取',  shunt\_mode\_id int COMMENT '匹配到的技能组id',  shunt\_employee\_group\_id int COMMENT '所属分流员工组',  ends\_time STRING COMMENT '有效时间') comment '客户关系表' PARTITIONED BY(starts\_time STRING) clustered by(customer\_relationship\_id) sorted by(customer\_relationship\_id) into 10 buckets ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' stored as orc TBLPROPERTIES ('orc.compress'='ZLIB'); |

##### Dimen

为了保证数据安全，采用外部表。

建库

|  |
| --- |
| **CREATE DATABASE IF NOT EXISTS** itcast\_dimen; |

###### Customer客户静态信息表

|  |
| --- |
| CREATE TABLE IF NOT EXISTS itcast\_dimen.`customer` (  `id` int COMMENT 'key id',  `customer\_relationship\_id` int COMMENT '当前意向id',  `create\_date\_time` STRING COMMENT '创建时间',  `update\_date\_time` STRING COMMENT '最后更新时间',  `deleted` int COMMENT '是否被删除（禁用）',  `name` STRING COMMENT '姓名',  `idcard` STRING COMMENT '身份证号',  `birth\_year` int COMMENT '出生年份',  `gender` STRING COMMENT '性别',  `phone` STRING COMMENT '手机号',  `wechat` STRING COMMENT '微信',  `qq` STRING COMMENT 'qq号',  `email` STRING COMMENT '邮箱',  `area` STRING COMMENT '所在区域',  `leave\_school\_date` date COMMENT '离校时间',  `graduation\_date` date COMMENT '毕业时间',  `bxg\_student\_id` STRING COMMENT '博学谷学员ID，可能未关联到，不存在',  `creator` int COMMENT '创建人ID',  `origin\_type` STRING COMMENT '数据来源',  `origin\_channel` STRING COMMENT '来源渠道',  `tenant` int,  `md\_id` int COMMENT '中台id') comment '客户表' PARTITIONED BY(start\_time STRING) ROW FORMAT DELIMITED  FIELDS TERMINATED BY '\t' stored as orc TBLPROPERTIES ('orc.compress'='SNAPPY'); |

###### employee员工表

|  |
| --- |
| CREATE TABLE IF NOT EXISTS itcast\_dimen.employee (  id int COMMENT '员工id',  email STRING COMMENT '公司邮箱，OA登录账号',  real\_name STRING COMMENT '员工的真实姓名',  phone STRING COMMENT '手机号，目前还没有使用；隐私问题OA接口没有提供这个属性，',  department\_id STRING COMMENT 'OA中的部门编号，有负值',  department\_name STRING COMMENT 'OA中的部门名',  remote\_login STRING COMMENT '员工是否可以远程登录',  job\_number STRING COMMENT '员工工号',  cross\_school STRING COMMENT '是否有跨校区权限',  last\_login\_date STRING COMMENT '最后登录日期',  creator int COMMENT '创建人',  create\_date\_time STRING COMMENT '创建时间',  update\_date\_time STRING COMMENT '最后更新时间',  deleted STRING COMMENT '是否被删除（禁用）',  scrm\_department\_id int COMMENT 'SCRM内部部门id',  leave\_office STRING COMMENT '离职状态',  leave\_office\_time STRING COMMENT '离职时间',  reinstated\_time STRING COMMENT '复职时间',  superior\_leaders\_id int COMMENT '上级领导ID',  tdepart\_id int COMMENT '直属部门',  tenant int COMMENT '租户',  ems\_user\_name STRING COMMENT 'ems用户名称' ) comment '员工表' PARTITIONED BY(start\_time STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' stored as orc TBLPROPERTIES ('orc.compress'='SNAPPY'); |

###### scrm\_department部门表

|  |
| --- |
| **CREATE TABLE IF NOT EXISTS** itcast\_dimen.`scrm\_department` (  `id` **int COMMENT '部门id'**,  `name` **STRING COMMENT '部门名称'**,  `parent\_id` **int COMMENT '父部门id'**,  `create\_date\_time` **STRING COMMENT '创建时间'**,  `update\_date\_time` **STRING COMMENT '更新时间'**,  `deleted` **STRING COMMENT '删除标志'**,  `id\_path` **STRING COMMENT '编码全路径'**,  `tdepart\_code` **int COMMENT '直属部门'**,  `creator` **STRING COMMENT '创建者'**,  `depart\_level` **int COMMENT '部门层级'**,  `depart\_sign` **int COMMENT '部门标志，暂时默认1'**,  `depart\_line` **int COMMENT '业务线，存储业务线编码'**,  `depart\_sort` **int COMMENT '排序字段'**,  `disable\_flag` **int COMMENT '禁用标志'**,  `tenant` **int COMMENT '租户'**) **comment 'scrm部门表'** PARTITIONED **BY**(start\_time **STRING**) **ROW FORMAT** DELIMITED **FIELDS TERMINATED BY '**\t**' stored as** orc TBLPROPERTIES ('orc.compress'='SNAPPY'); |

###### itcast\_school学校表

|  |
| --- |
| **CREATE TABLE IF NOT EXISTS** itcast\_dimen.`itcast\_school` (  `id` **int COMMENT '自增主键'**,  `create\_date\_time` **timestamp COMMENT '创建时间'**,  `update\_date\_time` **timestamp COMMENT '最后更新时间'**,  `deleted` **STRING COMMENT '是否被删除（禁用）'**,  `name` **STRING COMMENT '校区名称'**,  `code` **STRING COMMENT '校区标识'**,  `tenant` **int COMMENT '租户'**) **comment '校区字典表'**  PARTITIONED **BY**(start\_time **STRING**) **ROW FORMAT** DELIMITED  **FIELDS TERMINATED BY '**\t**' stored as** orc TBLPROPERTIES ('orc.compress'='SNAPPY'); |

###### itcast\_subject学科表

|  |
| --- |
| **CREATE TABLE IF NOT EXISTS** itcast\_dimen.`itcast\_subject` (  `id` **int COMMENT '自增主键'**,  `create\_date\_time` **timestamp COMMENT '创建时间'**,  `update\_date\_time` **timestamp COMMENT '最后更新时间'**,  `deleted` **STRING COMMENT '是否被删除（禁用）'**,  `name` **STRING COMMENT '学科名称'**,  `code` **STRING COMMENT '学科编码'**,  `tenant` **int COMMENT '租户'**) **comment '学科字典表'**  PARTITIONED **BY**(start\_time **STRING**) **ROW FORMAT** DELIMITED  **FIELDS TERMINATED BY '**\t**' stored as** orc TBLPROPERTIES ('orc.compress'='SNAPPY'); |

##### DWD

ODS事实数据customer\_relationship清洗转换后存入DWD明细层。

DW和APP层是统计数据，为了使覆盖插入等操作更方便，满足业务需求的同时，提高开发和测试效率，推荐使用内部表。

|  |
| --- |
| drop table itcast\_dwd.`itcast\_intention\_dwd`; CREATE TABLE IF NOT EXISTS itcast\_dwd.`itcast\_intention\_dwd` (  `rid` int COMMENT 'id',  `customer\_id` STRING COMMENT '客户id',  `create\_date\_time` STRING COMMENT '创建时间',  `itcast\_school\_id` STRING COMMENT '校区id',  `deleted` STRING COMMENT '是否被删除',  `origin\_type` STRING COMMENT '来源渠道',  `itcast\_subject\_id` STRING COMMENT '学科id',  `creator` int COMMENT '创建人',  `hourinfo` STRING COMMENT '小时信息',  `origin\_type\_stat` STRING COMMENT '数据来源:0.线下；1.线上' ) comment '客户意向dwd表' PARTITIONED BY(yearinfo STRING,monthinfo STRING,dayinfo STRING) clustered by(rid) sorted by(rid) into 10 buckets ROW FORMAT DELIMITED  FIELDS TERMINATED BY '\t' stored as ORC TBLPROPERTIES ('orc.compress'='SNAPPY'); |

##### DWM

关联所有维表，并对获取的字段进行转换，便于统计时直接使用。

|  |
| --- |
| create database itcast\_dwm;  drop table itcast\_dwm.`itcast\_intention\_dwm`; CREATE TABLE IF NOT EXISTS itcast\_dwm.`itcast\_intention\_dwm` (  `customer\_id` STRING COMMENT 'id信息',  `create\_date\_time` STRING COMMENT '创建时间',  `area` STRING COMMENT '区域信息',  `itcast\_school\_id` STRING COMMENT '校区id',  `itcast\_school\_name` STRING COMMENT '校区名称',  `deleted` STRING COMMENT '是否被删除',  `origin\_type` STRING COMMENT '来源渠道',  `itcast\_subject\_id` STRING COMMENT '学科id',  `itcast\_subject\_name` STRING COMMENT '学科名称',  `hourinfo` STRING COMMENT '小时信息',  `origin\_type\_stat` STRING COMMENT '数据来源:0.线下；1.线上',  `clue\_state\_stat` STRING COMMENT '新老客户：0.老客户；1.新客户',  `tdepart\_id` STRING COMMENT '创建者部门id',  `tdepart\_name` STRING COMMENT '咨询中心名称' ) comment '客户意向dwm表' PARTITIONED BY(yearinfo STRING,monthinfo STRING,dayinfo STRING) clustered by(customer\_id) sorted by(customer\_id) into 10 buckets ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' stored as ORC TBLPROPERTIES ('orc.compress'='SNAPPY'); |

##### DWS

在DWM层的基础上，按照业务的要求进行统计分析；有三个常驻维度，分别增加对应的属性标识：

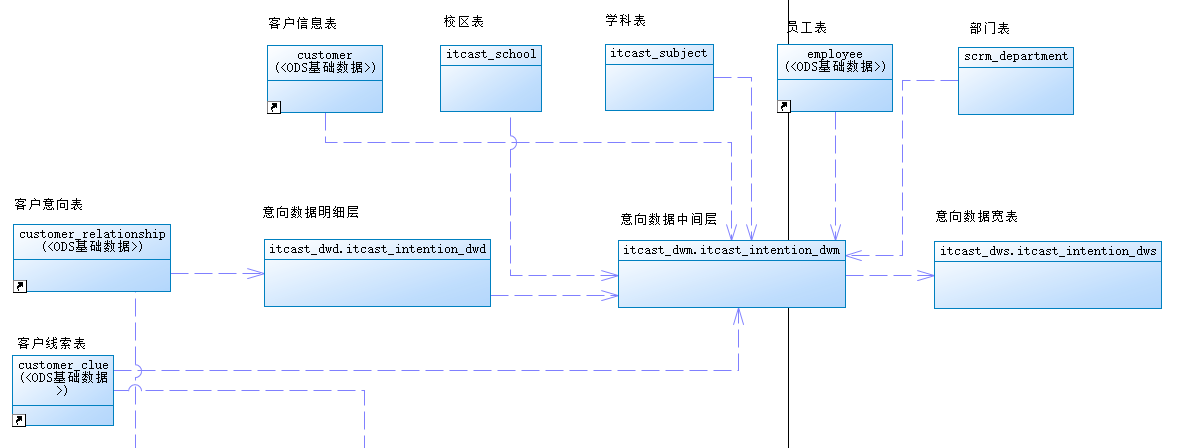
* 时间维度：1.年、2.月、3.天、4.小时
* 数据来源：0.线下；1.线上
* 客户属性：0.老客户、1.新客户
* 产品属性维度：1.总意向量；2.区域信息；3.校区、学科组合分组；4.来源渠道；5.贡献中心；

|  |
| --- |
| drop Table itcast\_dws.itcast\_intention\_dws; CREATE TABLE IF NOT EXISTS itcast\_dws.itcast\_intention\_dws (  `customer\_total` INT COMMENT '聚合意向客户数',  `area` STRING COMMENT '区域信息',  `itcast\_school\_id` STRING COMMENT '校区id',  `itcast\_school\_name` STRING COMMENT '校区名称',  `origin\_type` STRING COMMENT '来源渠道',  `itcast\_subject\_id` STRING COMMENT '学科id',  `itcast\_subject\_name` STRING COMMENT '学科名称',  `hourinfo` STRING COMMENT '小时信息',  `origin\_type\_stat` STRING COMMENT '数据来源:0.线下；1.线上',  `clue\_state\_stat` STRING COMMENT '客户属性：0.老客户；1.新客户',  `tdepart\_id` STRING COMMENT '创建者部门id',  `tdepart\_name` STRING COMMENT '咨询中心名称',  `time\_str` STRING COMMENT '时间明细',  `groupType` STRING COMMENT '产品属性类别：1.总意向量；2.区域信息；3.校区、学科组合分组；4.来源渠道；5.咨询中心;',  `time\_type` STRING COMMENT '时间维度：1、按小时聚合；2、按天聚合；3、按周聚合；4、按月聚合；5、按年聚合；' ) comment '客户意向dws表' PARTITIONED BY(yearinfo STRING,monthinfo STRING,dayinfo STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' stored as orc TBLPROPERTIES ('orc.compress'='SNAPPY'); |

##### APP

如果用户需要具体的报表展示，可以针对不同的报表页面设计APP层结构，然后导出至OLAP系统的mysql中。此系统使用FineReport，需要通过宽表来进行灵活的展现。因此APP层不再进行细化。直接将DWS层导出至mysql即可。

### 全量流程



#### 数据采集

##### Dimen层

###### Customer客户表

SQL：

|  |
| --- |
| select id,  customer\_relationship\_id,  create\_date\_time,  update\_date\_time,  deleted,  name,  idcard,  birth\_year,  gender,  phone,  wechat,  qq,  email,  area,  leave\_school\_date,  graduation\_date,  bxg\_student\_id,  creator,  origin\_type,  origin\_channel,  tenant,  md\_id,  *FROM\_UNIXTIME*(*unix\_timestamp*(), "%Y-%m-%d") as start\_time from customer; |

Sqoop：

|  |
| --- |
| sqoop import \  --connect jdbc:mysql://192.168.52.150:3306/scrm \  --username root \  --password 123456 \  --query 'select id, customer\_relationship\_id, create\_date\_time, update\_date\_time, deleted, name, idcard, birth\_year, gender, phone, wechat, qq, email, area, leave\_school\_date, graduation\_date, bxg\_student\_id, creator, origin\_type, origin\_channel, tenant, md\_id, FROM\_UNIXTIME(unix\_timestamp(),"%Y-%m-%d")as start\_time from customer where $CONDITIONS' \  --hcatalog-database itcast\_dimen \  --hcatalog-table customer \  -m 100 \  --split-by id |

###### employee员工表

SQL：

|  |
| --- |
| select id,  email,  real\_name,  -1 as phone,  department\_id,  department\_name,  remote\_login,  job\_number,  cross\_school,  last\_login\_date,  creator,  create\_date\_time,  update\_date\_time,  deleted,  scrm\_department\_id,  leave\_office,  leave\_office\_time,  reinstated\_time,  superior\_leaders\_id,  tdepart\_id,  tenant,  ems\_user\_name,  *FROM\_UNIXTIME*(*unix\_timestamp*(),"%Y-%m-%d")as start\_time from employee; |

Sqoop：

|  |
| --- |
| sqoop import \  --connect jdbc:mysql://192.168.52.150:3306/scrm \  --username root \  --password 123456 \  --query 'select id,email,real\_name,-1 as phone,department\_id,department\_name,remote\_login,job\_number,cross\_school,last\_login\_date,creator,create\_date\_time,update\_date\_time,deleted,scrm\_department\_id,leave\_office,leave\_office\_time,reinstated\_time,superior\_leaders\_id,tdepart\_id,tenant,ems\_user\_name,FROM\_UNIXTIME(unix\_timestamp(),"%Y-%m-%d")as start\_time from employee where $CONDITIONS' \  --hcatalog-database itcast\_dimen \  --hcatalog-table employee \  -m 100 \  --split-by id |

###### scrm\_department部门表

SQL：

|  |
| --- |
| select \*,  *FROM\_UNIXTIME*(*unix\_timestamp*(), "%Y-%m-%d") as start\_time from scrm\_department; |

Sqoop：

|  |
| --- |
| sqoop import \  --connect jdbc:mysql://192.168.52.150:3306/scrm \  --username root \  --password 123456 \  --query 'select \*, FROM\_UNIXTIME(unix\_timestamp(),"%Y-%m-%d")as start\_time from scrm\_department where $CONDITIONS' \  --hcatalog-database itcast\_dimen \  --hcatalog-table scrm\_department \  -m 100 \  --split-by id |

###### itcast\_school学校表

SQL：

|  |
| --- |
| select \*,  *FROM\_UNIXTIME*(*unix\_timestamp*(), "%Y-%m-%d") as start\_time from itcast\_school; |

Sqoop：

|  |
| --- |
| sqoop import \  --connect jdbc:mysql://192.168.52.150:3306/scrm \  --username root \  --password 123456 \  --query 'select \*, FROM\_UNIXTIME(unix\_timestamp(),"%Y-%m-%d")as start\_time from itcast\_school where $CONDITIONS' \  --hcatalog-database itcast\_dimen \  --hcatalog-table itcast\_school \  -m 100 \  --split-by id |

###### itcast\_subject学科表

SQL：

|  |
| --- |
| select \*,  *FROM\_UNIXTIME*(*unix\_timestamp*(), "%Y-%m-%d") as start\_time from itcast\_subject; |

Sqoop：

|  |
| --- |
| sqoop import \  --connect jdbc:mysql://192.168.52.150:3306/scrm \  --username root \  --password 123456 \  --query 'select \*, FROM\_UNIXTIME(unix\_timestamp(),"%Y-%m-%d")as start\_time from itcast\_subject where $CONDITIONS' \  --hcatalog-database itcast\_dimen \  --hcatalog-table itcast\_subject \  -m 100 \  --split-by id |

##### ODS层

Sqoop不支持分桶表，需要通过临时表的方式实现。

###### customer\_relationship意向表

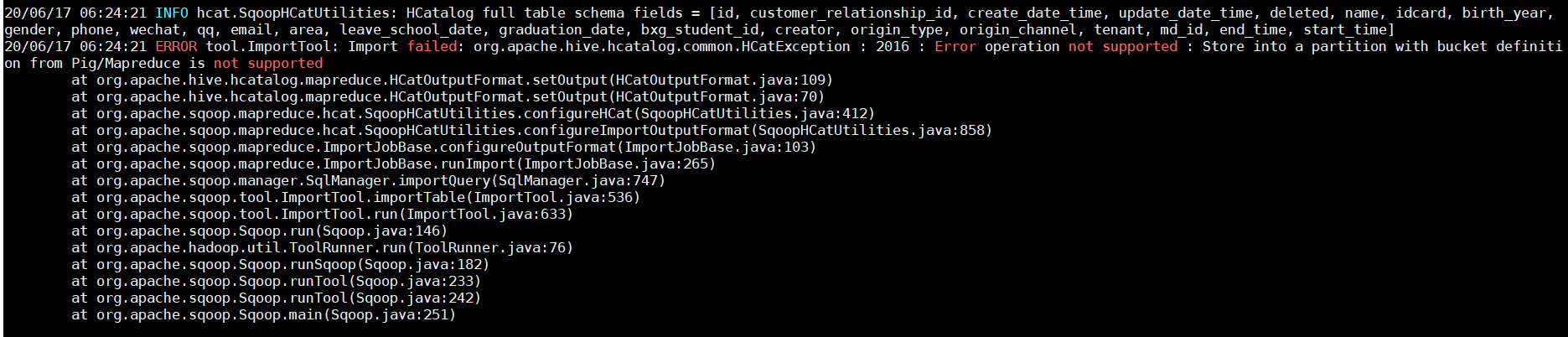
SQL：

|  |
| --- |
| select id,  create\_date\_time,  update\_date\_time,  deleted,  customer\_id,  first\_id,  belonger,  belonger\_name,  initial\_belonger,  distribution\_handler,  business\_scrm\_department\_id,  last\_visit\_time,  next\_visit\_time,  origin\_type,  itcast\_school\_id,  itcast\_subject\_id,  intention\_study\_type,  anticipat\_signup\_date,  level,  creator,  current\_creator,  creator\_name,  origin\_channel,  comment,  first\_customer\_clue\_id,  last\_customer\_clue\_id,  process\_state,  process\_time,  payment\_state,  payment\_time,  signup\_state,  signup\_time,  notice\_state,  notice\_time,  lock\_state,  lock\_time,  itcast\_clazz\_id,  itcast\_clazz\_time,  payment\_url,  payment\_url\_time,  ems\_student\_id,  delete\_reason,  deleter,  deleter\_name,  delete\_time,  course\_id,  course\_name,  delete\_comment,  close\_state,  close\_time,  appeal\_id,  tenant,  total\_fee,  belonged,  belonged\_time,  belonger\_time,  transfer,  transfer\_time,  follow\_type,  transfer\_bxg\_oa\_account,  transfer\_bxg\_belonger\_name,  *FROM\_UNIXTIME*(*unix\_timestamp*(), "%Y-%m-%d") as start\_time,  *date\_format*("9999-12-31", "%Y-%m-%d") as end\_time from customer\_relationship; |

Sqoop：

|  |
| --- |
| sqoop import \  --connect jdbc:mysql://192.168.52.150:3306/scrm \  --username root \  --password 123456 \  --query 'select id, create\_date\_time, update\_date\_time, deleted, customer\_id, first\_id, belonger, belonger\_name, initial\_belonger, distribution\_handler, business\_scrm\_department\_id, last\_visit\_time, next\_visit\_time, origin\_type, itcast\_school\_id, itcast\_subject\_id, intention\_study\_type, anticipat\_signup\_date, level, creator, current\_creator, creator\_name, origin\_channel, comment, first\_customer\_clue\_id, last\_customer\_clue\_id, process\_state, process\_time, payment\_state, payment\_time, signup\_state, signup\_time, notice\_state, notice\_time, lock\_state, lock\_time, itcast\_clazz\_id, itcast\_clazz\_time, payment\_url, payment\_url\_time, ems\_student\_id, delete\_reason, deleter, deleter\_name, delete\_time, course\_id, course\_name, delete\_comment, close\_state, close\_time, appeal\_id, tenant, total\_fee, belonged, belonged\_time, belonger\_time, transfer, transfer\_time, follow\_type, transfer\_bxg\_oa\_account, transfer\_bxg\_belonger\_name, FROM\_UNIXTIME(unix\_timestamp(),"%Y-%m-%d")as start\_time,date\_format("9999-12-31","%Y-%m-%d") as end\_time from customer\_relationship where $CONDITIONS' \  --hcatalog-database itcast\_ods \  --hcatalog-table ***customer\_relationship*** \  -m 10 \  --split-by id |

报错：



common.HCatException : 2016 : Error operation not supported : Store into a partition with bucket definition from Pig/Mapreduce is not supported

这个错误是由于sqoop不支持将数据导入分桶表所引起的问题，但是如果我们想在ODS进行分桶的话，如何来做呢？

我们可以通过临时表的方式来进行抽取数据，然后将临时表数据再同步到ODS分桶表即可。

重建ods临时表，注意不要有分桶

|  |
| --- |
| DROP TABLE itcast\_ods.`customer\_relationship\_tmp`; CREATE TABLE IF NOT EXISTS itcast\_ods.`customer\_relationship\_tmp` (  `id` int COMMENT '客户关系id',  `create\_date\_time` STRING COMMENT '创建时间',  `update\_date\_time` STRING COMMENT '最后更新时间',  `deleted` int COMMENT '是否被删除（禁用）',  `customer\_id` int COMMENT '所属客户id',  `first\_id` int COMMENT '第一条客户关系id',  `belonger` int COMMENT '归属人',  `belonger\_name` STRING COMMENT '归属人姓名',  `initial\_belonger` int COMMENT '初始归属人',  `distribution\_handler` int COMMENT '分配处理人',  `business\_scrm\_department\_id` int COMMENT '归属部门',  `last\_visit\_time` STRING COMMENT '最后回访时间',  `next\_visit\_time` STRING COMMENT '下次回访时间',  `origin\_type` STRING COMMENT '数据来源',  `itcast\_school\_id` int COMMENT '校区Id',  `itcast\_subject\_id` int COMMENT '学科Id',  `intention\_study\_type` STRING COMMENT '意向学习方式',  `anticipat\_signup\_date` STRING COMMENT '预计报名时间',  `level` STRING COMMENT '客户级别',  `creator` int COMMENT '创建人',  `current\_creator` int COMMENT '当前创建人：初始==创建人，当在公海拉回时为 拉回人',  `creator\_name` STRING COMMENT '创建者姓名',  `origin\_channel` STRING COMMENT '来源渠道',  `comment` STRING COMMENT '备注',  `first\_customer\_clue\_id` int COMMENT '第一条线索id',  `last\_customer\_clue\_id` int COMMENT '最后一条线索id',  `process\_state` STRING COMMENT '处理状态',  `process\_time` STRING COMMENT '处理状态变动时间',  `payment\_state` STRING COMMENT '支付状态',  `payment\_time` STRING COMMENT '支付状态变动时间',  `signup\_state` STRING COMMENT '报名状态',  `signup\_time` STRING COMMENT '报名时间',  `notice\_state` STRING COMMENT '通知状态',  `notice\_time` STRING COMMENT '通知状态变动时间',  `lock\_state` STRING COMMENT '锁定状态',  `lock\_time` STRING COMMENT '锁定状态修改时间',  `itcast\_clazz\_id` int COMMENT '所属ems班级id',  `itcast\_clazz\_time` STRING COMMENT '报班时间',  `payment\_url` STRING COMMENT '付款链接',  `payment\_url\_time` STRING COMMENT '支付链接生成时间',  `ems\_student\_id` int COMMENT 'ems的学生id',  `delete\_reason` STRING COMMENT '删除原因',  `deleter` int COMMENT '删除人',  `deleter\_name` STRING COMMENT '删除人姓名',  `delete\_time` STRING COMMENT '删除时间',  `course\_id` int COMMENT '课程ID',  `course\_name` STRING COMMENT '课程名称',  `delete\_comment` STRING COMMENT '删除原因说明',  `close\_state` STRING COMMENT '关闭装填',  `close\_time` STRING COMMENT '关闭状态变动时间',  `appeal\_id` int COMMENT '申诉id',  `tenant` int COMMENT '租户',  `total\_fee` DECIMAL COMMENT '报名费总金额',  `belonged` int COMMENT '小周期归属人',  `belonged\_time` STRING COMMENT '归属时间',  `belonger\_time` STRING COMMENT '归属时间',  `transfer` int COMMENT '转移人',  `transfer\_time` STRING COMMENT '转移时间',  `follow\_type` int COMMENT '分配类型，0-自动分配，1-手动分配，2-自动转移，3-手动单个转移，4-手动批量转移，5-公海领取',  `transfer\_bxg\_oa\_account` STRING COMMENT '转移到博学谷归属人OA账号',  `transfer\_bxg\_belonger\_name` STRING COMMENT '转移到博学谷归属人OA姓名',  `end\_time` STRING COMMENT '有效截止时间') comment '客户关系表' PARTITIONED BY(start\_time STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' stored as orc TBLPROPERTIES ('orc.compress'='ZLIB'); |

抽取数据到临时表

SQL：

|  |
| --- |
| select id,  create\_date\_time,  update\_date\_time,  deleted,  customer\_id,  first\_id,  belonger,  belonger\_name,  initial\_belonger,  distribution\_handler,  business\_scrm\_department\_id,  last\_visit\_time,  next\_visit\_time,  origin\_type,  itcast\_school\_id,  itcast\_subject\_id,  intention\_study\_type,  anticipat\_signup\_date,  level,  creator,  current\_creator,  creator\_name,  origin\_channel,  comment,  first\_customer\_clue\_id,  last\_customer\_clue\_id,  process\_state,  process\_time,  payment\_state,  payment\_time,  signup\_state,  signup\_time,  notice\_state,  notice\_time,  lock\_state,  lock\_time,  itcast\_clazz\_id,  itcast\_clazz\_time,  payment\_url,  payment\_url\_time,  ems\_student\_id,  delete\_reason,  deleter,  deleter\_name,  delete\_time,  course\_id,  course\_name,  delete\_comment,  close\_state,  close\_time,  appeal\_id,  tenant,  total\_fee,  belonged,  belonged\_time,  belonger\_time,  transfer,  transfer\_time,  follow\_type,  transfer\_bxg\_oa\_account,  transfer\_bxg\_belonger\_name,  *FROM\_UNIXTIME*(*unix\_timestamp*(), "%Y-%m-%d") as start\_time,  *date\_format*("9999-12-31", "%Y-%m-%d") as end\_time from customer\_relationship; |

Sqoop：

|  |
| --- |
| sqoop import \  --connect jdbc:mysql://192.168.52.150:3306/scrm \  --username root \  --password 123456 \  --query 'select id, create\_date\_time, update\_date\_time, deleted, customer\_id, first\_id, belonger, belonger\_name, initial\_belonger, distribution\_handler, business\_scrm\_department\_id, last\_visit\_time, next\_visit\_time, origin\_type, itcast\_school\_id, itcast\_subject\_id, intention\_study\_type, anticipat\_signup\_date, level, creator, current\_creator, creator\_name, origin\_channel, comment, first\_customer\_clue\_id, last\_customer\_clue\_id, process\_state, process\_time, payment\_state, payment\_time, signup\_state, signup\_time, notice\_state, notice\_time, lock\_state, lock\_time, itcast\_clazz\_id, itcast\_clazz\_time, payment\_url, payment\_url\_time, ems\_student\_id, delete\_reason, deleter, deleter\_name, delete\_time, course\_id, course\_name, delete\_comment, close\_state, close\_time, appeal\_id, tenant, total\_fee, belonged, belonged\_time, belonger\_time, transfer, transfer\_time, follow\_type, transfer\_bxg\_oa\_account, transfer\_bxg\_belonger\_name, FROM\_UNIXTIME(unix\_timestamp(),"%Y-%m-%d")as start\_time,date\_format("9999-12-31","%Y-%m-%d") as end\_time from customer\_relationship where $CONDITIONS' \  --hcatalog-database itcast\_ods \  --hcatalog-table **customer\_relationship\_tmp** \  -m 10 \  --split-by id |

将数据覆盖插入到ODS

|  |
| --- |
| *--分区* SET hive.exec.dynamic.partition=true; SET hive.exec.dynamic.partition.mode=nonstrict; set hive.exec.max.dynamic.partitions.pernode=10000; set hive.exec.max.dynamic.partitions=100000; set hive.exec.max.created.files=150000; *--hive压缩* set hive.exec.compress.intermediate=true; set hive.exec.compress.output=true; *--写入时压缩生效* set hive.exec.orc.compression.strategy=COMPRESSION; *--分桶* set hive.enforce.bucketing=true; set hive.enforce.sorting=true; set hive.optimize.bucketmapjoin = true; set hive.auto.convert.sortmerge.join=true; set hive.auto.convert.sortmerge.join.noconditionaltask=true;  **insert overwrite** table itcast\_ods.customer\_relationship partition(start\_time) select *\** from itcast\_ods.customer\_relationship**\_tmp**; |

###### Customer\_clue线索表

重建ods表，注意不要有分桶

|  |
| --- |
| DROP TABLE itcast\_ods.customer\_clue\_tmp; CREATE TABLE IF NOT EXISTS itcast\_ods.customer\_clue\_tmp (  id int COMMENT 'customer\_clue\_id',  create\_date\_time STRING COMMENT '创建时间',  update\_date\_time STRING COMMENT '最后更新时间',  deleted STRING COMMENT '是否被删除（禁用）',  customer\_id int COMMENT '客户id',  customer\_relationship\_id int COMMENT '客户关系id',  session\_id STRING COMMENT '七陌会话id',  sid STRING COMMENT '访客id',  status STRING COMMENT '状态（undeal待领取 deal 已领取 finish 已关闭 changePeer 已流转）',  users STRING COMMENT '所属坐席',  create\_time STRING COMMENT '七陌创建时间',  platform STRING COMMENT '平台来源 （pc-网站咨询|wap-wap咨询|sdk-app咨询|weixin-微信咨询）',  s\_name STRING COMMENT '用户名称',  seo\_source STRING COMMENT '搜索来源',  seo\_keywords STRING COMMENT '关键字',  ip STRING COMMENT 'IP地址',  referrer STRING COMMENT '上级来源页面',  from\_url STRING COMMENT '会话来源页面',  landing\_page\_url STRING COMMENT '访客着陆页面',  url\_title STRING COMMENT '咨询页面title',  to\_peer STRING COMMENT '所属技能组',  manual\_time STRING COMMENT '人工开始时间',  begin\_time STRING COMMENT '坐席领取时间 ',  reply\_msg\_count int COMMENT '客服回复消息数',  total\_msg\_count int COMMENT '消息总数',  msg\_count int COMMENT '客户发送消息数',  comment STRING COMMENT '备注',  finish\_reason STRING COMMENT '结束类型',  finish\_user STRING COMMENT '结束坐席',  end\_time STRING COMMENT '会话结束时间',  platform\_description STRING COMMENT '客户平台信息',  browser\_name STRING COMMENT '浏览器名称',  os\_info STRING COMMENT '系统名称',  area STRING COMMENT '区域',  country STRING COMMENT '所在国家',  province STRING COMMENT '省',  city STRING COMMENT '城市',  creator int COMMENT '创建人',  name STRING COMMENT '客户姓名',  idcard STRING COMMENT '身份证号',  phone STRING COMMENT '手机号',  itcast\_school\_id int COMMENT '校区Id',  itcast\_school STRING COMMENT '校区',  itcast\_subject\_id int COMMENT '学科Id',  itcast\_subject STRING COMMENT '学科',  wechat STRING COMMENT '微信',  qq STRING COMMENT 'qq号',  email STRING COMMENT '邮箱',  gender STRING COMMENT '性别',  level STRING COMMENT '客户级别',  origin\_type STRING COMMENT '数据来源渠道',  information\_way STRING COMMENT '资讯方式',  working\_years STRING COMMENT '开始工作时间',  technical\_directions STRING COMMENT '技术方向',  customer\_state STRING COMMENT '当前客户状态',  valid STRING COMMENT '该线索是否是网资有效线索',  anticipat\_signup\_date STRING COMMENT '预计报名时间',  clue\_state STRING COMMENT '线索状态',  scrm\_department\_id int COMMENT 'SCRM内部部门id',  superior\_url STRING COMMENT '诸葛获取上级页面URL',  superior\_source STRING COMMENT '诸葛获取上级页面URL标题',  landing\_url STRING COMMENT '诸葛获取着陆页面URL',  landing\_source STRING COMMENT '诸葛获取着陆页面URL来源',  info\_url STRING COMMENT '诸葛获取留咨页URL',  info\_source STRING COMMENT '诸葛获取留咨页URL标题',  origin\_channel STRING COMMENT '投放渠道',  course\_id int COMMENT '课程编号',  course\_name STRING COMMENT '课程名称',  zhuge\_session\_id STRING COMMENT 'zhuge会话id',  is\_repeat int COMMENT '是否重复线索(手机号维度) 0:正常 1：重复',  tenant int COMMENT '租户id',  activity\_id STRING COMMENT '活动id',  activity\_name STRING COMMENT '活动名称',  follow\_type int COMMENT '分配类型，0-自动分配，1-手动分配，2-自动转移，3-手动单个转移，4-手动批量转移，5-公海领取',  shunt\_mode\_id int COMMENT '匹配到的技能组id',  shunt\_employee\_group\_id int COMMENT '所属分流员工组',  ends\_time STRING COMMENT '有效时间') comment '客户关系表' PARTITIONED BY(starts\_time STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' stored as orc TBLPROPERTIES ('orc.compress'='ZLIB'); |

抽取数据到临时表

SQL：

|  |
| --- |
| select id,  create\_date\_time,  update\_date\_time,  deleted,  customer\_id,  customer\_relationship\_id,  session\_id,  sid,  status, user, create\_time, platform, s\_name, seo\_source, seo\_keywords, ip, referrer, from\_url, landing\_page\_url, url\_title, to\_peer, manual\_time, begin\_time, reply\_msg\_count, total\_msg\_count, msg\_count, comment, finish\_reason, finish\_user, end\_time, platform\_description, browser\_name, os\_info, area, country, province, city, creator, name, idcard, phone, itcast\_school\_id, itcast\_school, itcast\_subject\_id, itcast\_subject, wechat, qq, email, gender, level, origin\_type, information\_way, working\_years, technical\_directions, customer\_state, valid, anticipat\_signup\_date, clue\_state, scrm\_department\_id, superior\_url, superior\_source, landing\_url, landing\_source, info\_url, info\_source, origin\_channel, course\_id, course\_name, zhuge\_session\_id, is\_repeat, tenant, activity\_id, activity\_name, follow\_type, shunt\_mode\_id, shunt\_employee\_group\_id,   FROM\_UNIXTIME(unix\_timestamp(), "%Y-%m-%d") as starts\_time, date\_format("9999-12-31", "%Y-%m-%d") as ends\_time from customer\_clue; |

Sqoop：

|  |
| --- |
| sqoop import \  --connect jdbc:mysql://192.168.52.150:3306/scrm \  --username root \  --password 123456 \  --query 'select id,create\_date\_time,update\_date\_time,deleted,customer\_id,customer\_relationship\_id,session\_id,sid,status,user as users,create\_time,platform,s\_name,seo\_source,seo\_keywords,ip,referrer,from\_url,landing\_page\_url,url\_title,to\_peer,manual\_time,begin\_time,reply\_msg\_count,total\_msg\_count,msg\_count,comment,finish\_reason,finish\_user,end\_time,platform\_description,browser\_name,os\_info,area,country,province,city,creator,name,"-1" as idcard,"-1" as phone,itcast\_school\_id,itcast\_school,itcast\_subject\_id,itcast\_subject,"-1" as wechat,"-1" as qq,"-1" as email,gender,level,origin\_type,information\_way,working\_years,technical\_directions,customer\_state,valid,anticipat\_signup\_date,clue\_state,scrm\_department\_id,superior\_url,superior\_source,landing\_url,landing\_source,info\_url,info\_source,origin\_channel,course\_id,course\_name,zhuge\_session\_id,is\_repeat,tenant,activity\_id,activity\_name,follow\_type,shunt\_mode\_id,shunt\_employee\_group\_id,FROM\_UNIXTIME(unix\_timestamp(),"%Y-%m-%d")as starts\_time,date\_format("9999-12-31","%Y-%m-%d") as ends\_time from customer\_clue where $CONDITIONS' \  --hcatalog-database itcast\_ods \  --hcatalog-table customer\_clue**\_tmp** \  -m 10 \  --split-by id |

将数据覆盖插入到ODS

|  |
| --- |
| **insert overwrite** table itcast\_ods.customer\_clue partition(starts\_time) select *\** from itcast\_ods.customer\_clue\_tmp; |

#### 数据清洗转换

##### Hive执行计划

###### 作用

用户提交HiveQL查询后，Hive会把查询语句转换为MapReduce作业。Hive会自动完成整个执行过程，一般情况下，我们并不用知道内部是如何运行的。

执行计划可以告诉我们查询过程的关键信息，用来帮助我们判定优化措施是否已经生效。

###### 基础语法

EXPLAIN的使用非常简单，只需要在正常HiveQL前面加上EXPLAIN就可以了。执行计划运行时的HiveQL不会真正执行作业，只是基于优化器生成了最优的执行路径：

|  |
| --- |
| EXPLAIN [EXTENDED] query |

extended输出更加详细的信息；

###### 执行计划分为两部分

1. stage依赖(STAGE DEPENDENCIES)
   1. 这部分展示本次查询分为两个stage：Stage-1，Stage-0.
   2. 一般Stage-0是最终给查询用户展示数据用的，如LIMITE操作就会在这部分。
   3. Stage-1是mr程序的执行阶段。

|  |
| --- |
| 1 STAGE DEPENDENCIES: 2 Stage-1 is a root stage 3 Stage-0 depends on stages: Stage-1 |

1. stage详细执行计划(STAGE PLANS)
   1. 包含了整个查询所有Stage的大部分处理过程。
   2. 特定优化是否生效，主要通过此部分内容查看。
2. 名次解释

TableScan:查看表

alias: emp：所需要的表

Statistics: Num rows: 2 Data size: 820 Basic stats: COMPLETE Column stats: NONE：这张表的基本统计信息：行数、大小等；

expressions: empno (type: int), ename (type: string), job (type: string), mgr (type: int), hiredate (type: string), sal (type: double), comm (type: double), deptno (type: int)：表中需要输出的字段及类型

outputColumnNames: \_col0, \_col1, \_col2, \_col3, \_col4, \_col5, \_col6, \_col7：输出的的字段编号

compressed: true：输出是否压缩；

input format: org.apache.hadoop.mapred.SequenceFileInputFormat：文件输入调用的Java类，显示以文本Text格式输入；

output format: org.apache.hadoop.hive.ql.io.HiveSequenceFileOutputFormat：文件输出调用的java类，显示以文本Text格式输出；

###### 样例

DWD阶段执行计划：

|  |
| --- |
| 1 STAGE DEPENDENCIES: 2 Stage-1 is a root stage 3 Stage-0 depends on stages: Stage-1 4 5 STAGE PLANS: 6 Stage: Stage-1 7 Map Reduce 8 Map Operator Tree: 9 TableScan 10 alias: rs 11 Statistics: Num rows: 1109147 Data size: 236547154 Basic stats: COMPLETE Column stats: COMPLETE 12 Filter Operator 13 predicate: (((hash(id) & 2147483647) % 10) = 0) (type: boolean) 14 Statistics: Num rows: 554573 Data size: 118273474 Basic stats: COMPLETE Column stats: COMPLETE 15 Select Operator 16 expressions: id (type: int), customer\_id (type: int), create\_date\_time (type: string), *if*((itcast\_school\_id is null or (itcast\_school\_id = 0)), -1, itcast\_school\_id) (type: int), deleted (type: int), origin\_type (type: string), *if*((itcast\_subject\_id is null or (itcast\_subject\_id = 0)), -1, itcast\_subject\_id) (type: int), *substr*(create\_date\_time, 12, 2) (type: string), *if*((origin\_type = 'NETSERVICE'), '1', *if*((origin\_type = 'PRESIGNUP'), '1', '0')) (type: string), *substr*(create\_date\_time, 1, 4) (type: string), *substr*(create\_date\_time, 6, 2) (type: string), *substr*(create\_date\_time, 9, 2) (type: string) 17 outputColumnNames: \_col0, \_col1, \_col2, \_col3, \_col4, \_col5, \_col6, \_col7, \_col8, \_col9, \_col10, \_col11 18 Statistics: Num rows: 554573 Data size: 631104074 Basic stats: COMPLETE Column stats: COMPLETE 19 File Output Operator 20 compressed: false 21 Statistics: Num rows: 554573 Data size: 631104074 Basic stats: COMPLETE Column stats: COMPLETE 22 table: 23 input format: org.apache.hadoop.mapred.SequenceFileInputFormat 24 output format: org.apache.hadoop.hive.ql.io.HiveSequenceFileOutputFormat 25 serde: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe 26  27 Stage: Stage-0 28 Fetch Operator 29 limit: -1 30 Processor Tree: 31 ListSink |

##### DWD

###### 分析

在DWD层对customer\_relationship意向客户事实表做清洗转换：

清洗掉已删除的数据；

判断学校id和学科id，空值统一转换为-1；

将origin\_type来源渠道字段转换为线上/线下，如果origin\_type是NETSERVICE和PRESIGNUP类型，即为1线上，否则为0线下。

###### 代码

|  |
| --- |
| *--分区* SET hive.exec.dynamic.partition=true; SET hive.exec.dynamic.partition.mode=nonstrict; set hive.exec.max.dynamic.partitions.pernode=10000; set hive.exec.max.dynamic.partitions=100000; set hive.exec.max.created.files=150000; *--hive压缩* set hive.exec.compress.intermediate=true; set hive.exec.compress.output=true; *--写入时压缩生效* set hive.exec.orc.compression.strategy=COMPRESSION; *--分桶* set hive.enforce.bucketing=true; set hive.enforce.sorting=true; set hive.optimize.bucketmapjoin = true; set hive.auto.convert.sortmerge.join=true; set hive.auto.convert.sortmerge.join.noconditionaltask=true;  insert into table itcast\_dwd.itcast\_intention\_dwd partition (yearinfo,monthinfo,dayinfo) select  rs.id as rid,  rs.customer\_id,  rs.create\_date\_time,  ***if*(**(rs.itcast\_school\_id is null) **or** (rs.itcast\_school\_id = 0), -1, rs.itcast\_school\_id**)** as itcast\_school\_id,  rs.deleted,  rs.origin\_type,  *if*((rs.itcast\_subject\_id is null) or (rs.itcast\_subject\_id = 0), -1, rs.itcast\_subject\_id) as itcast\_subject\_id,  *substr*(rs.create\_date\_time, 12, 2) hourinfo,  *if*(rs.origin\_type='NETSERVICE', '1', *if*(rs.origin\_type='PRESIGNUP', '1', '0')) as origin\_type\_stat,  *substr*(rs.create\_date\_time, 1, 4) yearinfo,  *substr*(rs.create\_date\_time, 6, 2) monthinfo,  *substr*(rs.create\_date\_time, 9, 2) dayinfo from itcast\_ods.customer\_relationship rs where rs.deleted = 0; |

###### 测试

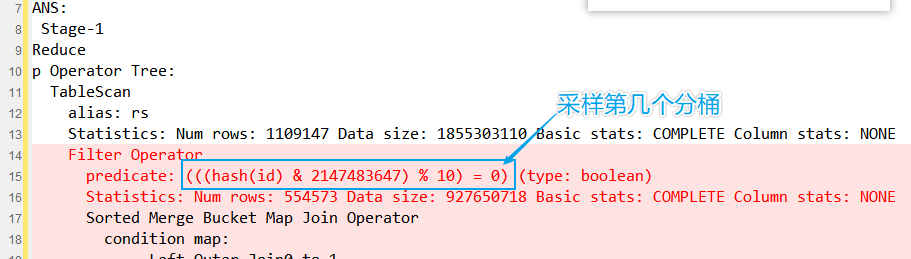
测试时，可以通过分区和分桶采样的方式。

分区针对的是固定日期，而分桶采样则侧重抽查，更具有代表性。由于第一次是全量抽取数据，所以日期分区下的数据非常庞大，此时使用分桶来进行采样测试可以提升开发和测试效率。

注意tablesample关键字所在的位置，是在表名之后，别名之前。

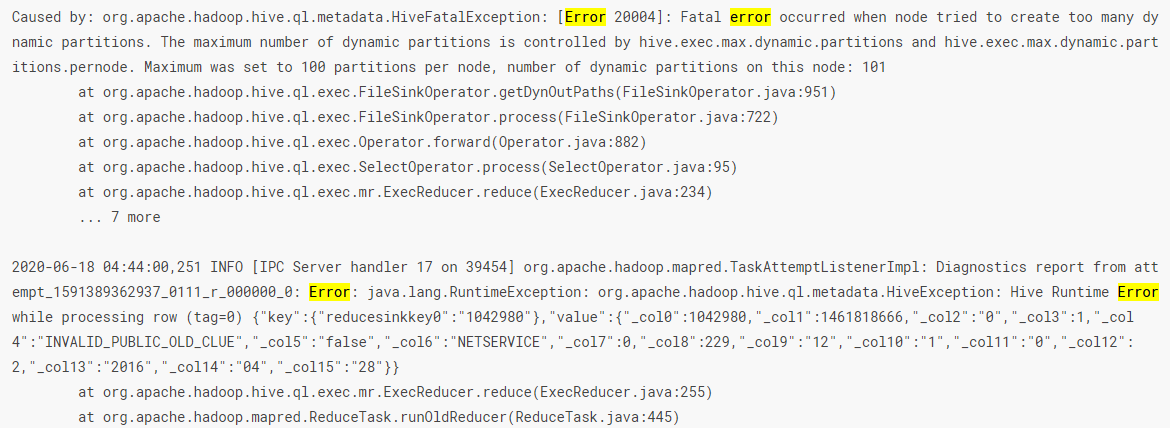
###### 执行计划验证

在select之前添加Explain，先来查看查询执行计划，可以看到分桶采样已经生效，提高了开发和测试时的执行效率。



|  |
| --- |
| insert into table itcast\_dwd.itcast\_intention\_dwd partition (yearinfo,monthinfo,dayinfo) select  rs.id as rid,  rs.customer\_id,  rs.create\_date\_time,  *if*((rs.itcast\_school\_id is null) or (rs.itcast\_school\_id = 0), -1, rs.itcast\_school\_id) as itcast\_school\_id,  rs.deleted,  rs.origin\_type,  *if*((rs.itcast\_subject\_id is null) or (rs.itcast\_subject\_id = 0), -1, rs.itcast\_subject\_id) as itcast\_subject\_id,  *substr*(rs.create\_date\_time, 12, 2) hourinfo,  *if*(rs.origin\_type='NETSERVICE', '1', *if*(rs.origin\_type='PRESIGNUP', '1', '0')) as origin\_type\_stat,  *substr*(rs.create\_date\_time, 1, 4) yearinfo,  *substr*(rs.create\_date\_time, 6, 2) monthinfo,  *substr*(rs.create\_date\_time, 9, 2) dayinfo from itcast\_ods.customer\_relationship tablesample(bucket 1 out of 10 on id) rs where rs.deleted = 0; |

###### 动态分区报错



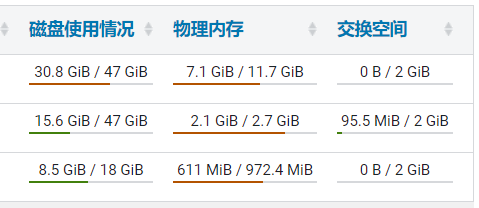
提高动态分区数量和文件数量，在sql前添加：

|  |
| --- |
| set hive.exec.max.dynamic.partitions.pernode=10000; set hive.exec.max.dynamic.partitions=100000; set hive.exec.max.created.files=150000; |

###### 内存溢出

**注意**，如果遇到因硬件配置而导致的内存溢出问题，有以下几种处理办法：

硬件内存充足



按照访问咨询看板中增加内存的设置进行配置：

1. 提高Yarn的NodeManager内存配置

修改参数yarn.nodemanager.resource.memory-mb。

1. 提高MR的内存配置

修改参数mapreduce.map.java.opts、mapreduce.reduce.java.opts、mapreduce.map.memory.mb、mapreduce.reduce.memory.mb。

硬件内存不足

开启有序动态分区，并关闭Map Join，但过程会比较慢。

也可以通过where条件，按照日期分批进行清洗转换。

查看各个年份数据分布情况：

|  |
| --- |
| select *count*(1), *substr*(create\_date\_time, 1, 4) from itcast\_ods.customer\_relationship group by *substr*(create\_date\_time, 1, 4); |



从结果可以看出，数据按年分配比较均匀，因此可以按照年份来进行分批计算。

|  |
| --- |
| insert into table itcast\_dwd.itcast\_intention\_dwd partition (yearinfo,monthinfo,dayinfo) select  rs.id as rid,  rs.customer\_id,  rs.create\_date\_time,  *if*((rs.itcast\_school\_id is null) or (rs.itcast\_school\_id = 0), -1, rs.itcast\_school\_id) as itcast\_school\_id,  rs.deleted,  rs.origin\_type,  *if*((rs.itcast\_subject\_id is null) or (rs.itcast\_subject\_id = 0), -1, rs.itcast\_subject\_id) as itcast\_subject\_id,  *substr*(rs.create\_date\_time, 12, 2) hourinfo,  *if*(rs.origin\_type='NETSERVICE', '1', *if*(rs.origin\_type='PRESIGNUP', '1', '0')) as origin\_type\_stat,  *substr*(rs.create\_date\_time, 1, 4) yearinfo,  *substr*(rs.create\_date\_time, 6, 2) monthinfo,  *substr*(rs.create\_date\_time, 9, 2) dayinfo from itcast\_ods.customer\_relationship tablesample(bucket 1 out of 10 on id) rs where rs.create\_date\_time between '2011-01-01 00:00:00' and '2012-01-01 00:00:00'; |

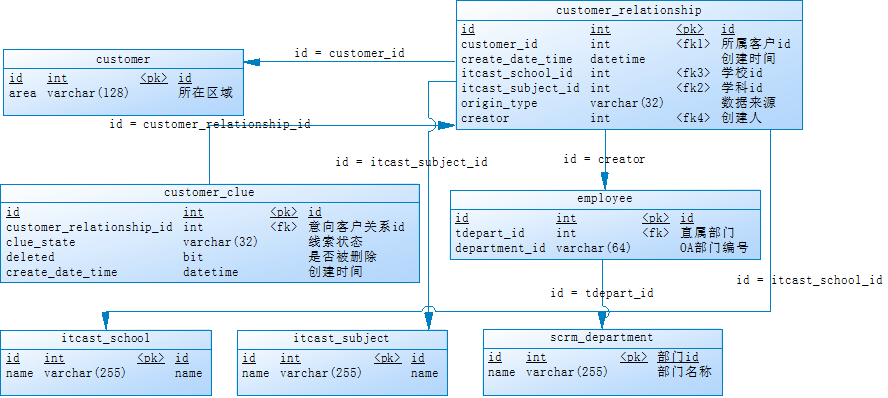
本地模式（虚拟机环境）

|  |
| --- |
| set hive.exec.mode.local.auto=true; |

##### DWM

###### 分析

意向客户量指标，最终统计的是去重后的客户；所以不能采用先count后sum的形式进行。因此在DWM中间层，我们不做统计，只将相关的维度数据进行关联，并转换出我们需要的信息。



通过id关联customer\_clue表的customer\_relationship\_id，将clue\_state状态转换为新老客户，如果clue\_state状态为VALID\_NEW\_CLUES，则为新客户，为VALID\_PUBLIC\_NEW\_CLUE，则为老客户，否则为无效数据。

通过customer\_id关联customer表id获取到区域信息area；

通过creator关联employee表获取tdepart\_id咨询中心单位id；再用employee的department\_id和scrm\_department表id关联获取单位名称name。

通过itcast\_subject\_id学科id和itcast\_subject学科表id进行关联，获取到学科名称name。

通过itcast\_school\_id学科id和itcast\_school校区表id进行关联，获取到校区名称name。

###### 代码

|  |
| --- |
| *--分区* SET hive.exec.dynamic.partition=true; SET hive.exec.dynamic.partition.mode=nonstrict; set hive.exec.max.dynamic.partitions.pernode=10000; set hive.exec.max.dynamic.partitions=100000; set hive.exec.max.created.files=150000; *--hive压缩* set hive.exec.compress.intermediate=true; set hive.exec.compress.output=true; *--写入时压缩生效* set hive.exec.orc.compression.strategy=COMPRESSION; *--分桶* set hive.enforce.bucketing=true; set hive.enforce.sorting=true; set hive.optimize.bucketmapjoin = true; set hive.auto.convert.sortmerge.join=true; set hive.auto.convert.sortmerge.join.noconditionaltask=true;  insert into table itcast\_dwm.itcast\_intention\_dwm partition (yearinfo,monthinfo,dayinfo) select  dwd.customer\_id,  dwd.create\_date\_time,  cus.area,  dwd.itcast\_school\_id,  sch.name as itcast\_school\_name,  dwd.deleted,  dwd.origin\_type,  dwd.itcast\_subject\_id,  sub.name as itcast\_subject\_name,  dwd.hourinfo,  dwd.origin\_type\_stat,  *if*(clue.clue\_state='VALID\_NEW\_CLUES', '1', *if*(clue.clue\_state='VALID\_PUBLIC\_NEW\_CLUE', '0', '-1')) as clue\_state\_stat,  e.department\_id as tdepart\_id,  dept.name as tdepart\_name,  dwd.yearinfo,  dwd.monthinfo,  dwd.dayinfo from itcast\_dwd.itcast\_intention\_dwd dwd left join itcast\_ods.customer\_clue clue on clue.customer\_relationship\_id=dwd.rid left join itcast\_dimen.customer cus on dwd.customer\_id = cus.id left join itcast\_dimen.employee e on dwd.creator = e.id left join itcast\_dimen.scrm\_department dept on e.department\_id = dept.id left join itcast\_dimen.itcast\_subject sub on dwd.itcast\_subject\_id = sub.id and sub.name is not null left join itcast\_dimen.itcast\_school sch on dwd.itcast\_school\_id = sch.id; |

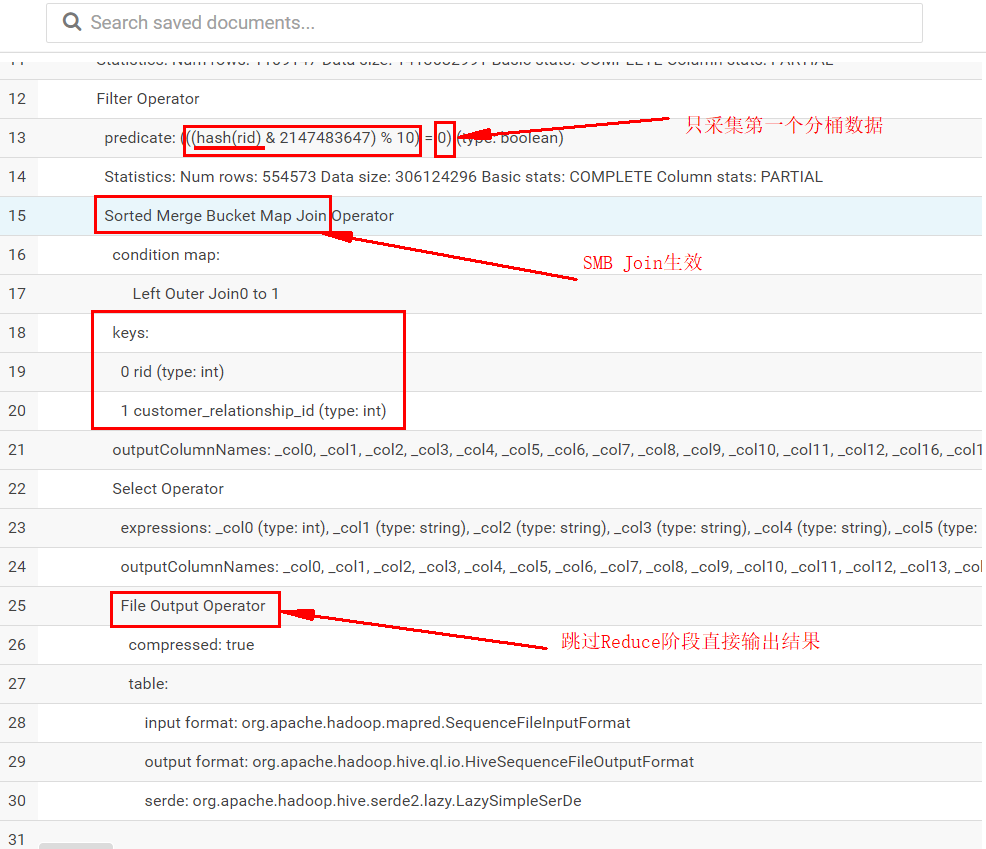
###### 测试

可以使用分桶采样来进行测试。这里因为我们在DWD层已经将数据分桶后减少了9/10，也可以不用再分桶。

执行计划验证

可以看到分桶采样，以及SMB Join都生效了，去掉Reduce过程避免了数据倾斜的问题，提升了执行效率。

|  |
| --- |
| **explain** select  dwd.customer\_id,  dwd.create\_date\_time,  cus.area,  dwd.itcast\_school\_id,  sch.name as itcast\_school\_name,  dwd.deleted,  dwd.origin\_type,  dwd.itcast\_subject\_id,  sub.name as itcast\_subject\_name,  dwd.hourinfo,  dwd.origin\_type\_stat,  *if*(clue.clue\_state='VALID\_NEW\_CLUES', '1', *if*(clue.clue\_state='VALID\_PUBLIC\_NEW\_CLUE', '0', '-1')) as clue\_state\_stat,  e.department\_id as tdepart\_id,  dept.name as tdepart\_name,  dwd.yearinfo,  dwd.monthinfo,  dwd.dayinfo from itcast\_dwd.itcast\_intention\_dwd tablesample(bucket 1 out of 10 on rid) dwd left join itcast\_ods.customer\_clue clue on clue.customer\_relationship\_id=dwd.rid left join itcast\_dimen.customer cus on dwd.customer\_id = cus.id left join itcast\_dimen.employee e on dwd.creator = e.id left join itcast\_dimen.scrm\_department dept on e.department\_id = dept.id left join itcast\_dimen.itcast\_subject sub on dwd.itcast\_subject\_id = sub.id  left join itcast\_dimen.itcast\_school sch on dwd.itcast\_school\_id = sch.id; |



运行插入

|  |
| --- |
| insert into table itcast\_dwm.itcast\_intention\_dwm partition (yearinfo,monthinfo,dayinfo) select  dwd.customer\_id,  dwd.create\_date\_time,  cus.area,  dwd.itcast\_school\_id,  sch.name as itcast\_school\_name,  dwd.deleted,  dwd.origin\_type,  dwd.itcast\_subject\_id,  sub.name as itcast\_subject\_name,  dwd.hourinfo,  dwd.origin\_type\_stat,  *if*(clue.clue\_state='VALID\_NEW\_CLUES', '1', *if*(clue.clue\_state='VALID\_PUBLIC\_NEW\_CLUE', '0', '-1')) as clue\_state\_stat,  e.department\_id as tdepart\_id,  dept.name as tdepart\_name,  dwd.yearinfo,  dwd.monthinfo,  dwd.dayinfo from itcast\_dwd.itcast\_intention\_dwd dwd left join itcast\_ods.customer\_clue clue on clue.customer\_relationship\_id=dwd.rid left join itcast\_dimen.customer cus on dwd.customer\_id = cus.id left join itcast\_dimen.employee e on dwd.creator = e.id left join itcast\_dimen.scrm\_department dept on e.department\_id = dept.id left join itcast\_dimen.itcast\_subject sub on dwd.itcast\_subject\_id = sub.id  left join itcast\_dimen.itcast\_school sch on dwd.itcast\_school\_id = sch.id; |

#### 统计分析

##### DWS

###### 分析

DWS层基于DWM清洗转换关联后的数据，使用count+distinct来统计指标。

在建模分析阶段，我们已经得到了指标相关的维度。分四大类：

* 时间维度：1.年、2.月、3.天、4.小时
* 产品属性维度：1.总意向量；2.区域信息；3.校区、学科组合分组；4.来源渠道；5.贡献中心；
* 数据来源：0.线下；1.线上
* 客户属性：0.老客户、1.新客户

代码按照产品属性分开统计；时间属性、线上线下和客户属性作为常驻字段，每一种统计分组中都要包含。

###### 代码

新增总意向量

|  |
| --- |
| *--分区* SET hive.exec.dynamic.partition=true; SET hive.exec.dynamic.partition.mode=nonstrict; set hive.exec.max.dynamic.partitions.pernode=10000; set hive.exec.max.dynamic.partitions=100000; set hive.exec.max.created.files=150000; *--hive压缩* set hive.exec.compress.intermediate=true; set hive.exec.compress.output=true; *--写入时压缩生效* set hive.exec.orc.compression.strategy=COMPRESSION; *--分桶* set hive.enforce.bucketing=true; set hive.enforce.sorting=true; set hive.optimize.bucketmapjoin = true; set hive.auto.convert.sortmerge.join=true; set hive.auto.convert.sortmerge.join.noconditionaltask=true;  *--总意向量分组（按照时间和常驻类型统计） --小时* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  '-1' itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo,'-',dayinfo,' ',hourinfo) as time\_str,  '1' as grouptype,  '1' as time\_type,  yearinfo,  monthinfo,  dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by yearinfo, monthinfo, dayinfo, hourinfo, origin\_type\_stat, clue\_state\_stat; *--天* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  '-1' itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo,'-',dayinfo) as time\_str,  '1' as grouptype,  '2' as time\_type,  yearinfo,  monthinfo,  dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by yearinfo, monthinfo, dayinfo, origin\_type\_stat, clue\_state\_stat; *--月* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  '-1' itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo) as time\_str,  '1' as grouptype,  '1' as time\_type,  yearinfo,  monthinfo,  '-1' as dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by yearinfo, monthinfo, origin\_type\_stat, clue\_state\_stat; *--年* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  '-1' itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo) as time\_str,  '1' as grouptype,  '1' as time\_type,  yearinfo,  '-1' as monthinfo,  '-1' as dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by yearinfo, origin\_type\_stat, clue\_state\_stat; |

意向学员位置热力图

|  |
| --- |
| *--地区分组（按照地区、时间和常驻类型统计） --小时* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  area,  '-1' itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo,'-',dayinfo,' ',hourinfo) as time\_str,  '2' as grouptype,  '1' as time\_type,  yearinfo,  monthinfo,  dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by **area,** yearinfo, monthinfo, dayinfo, hourinfo, origin\_type\_stat, clue\_state\_stat; *--天* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  area,  '-1' itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo,'-',dayinfo) as time\_str,  '2' as grouptype,  '2' as time\_type,  yearinfo,  monthinfo,  dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by area, yearinfo, monthinfo, dayinfo, origin\_type\_stat, clue\_state\_stat; *--月* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  area,  '-1' itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo) as time\_str,  '1' as grouptype,  '1' as time\_type,  yearinfo,  monthinfo,  '-1' as dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by area, yearinfo, monthinfo, origin\_type\_stat, clue\_state\_stat; *--年* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  area,  '-1' itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo) as time\_str,  '2' as grouptype,  '1' as time\_type,  yearinfo,  '-1' as monthinfo,  '-1' as dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by area, yearinfo, origin\_type\_stat, clue\_state\_stat; |

学科、校区排名

|  |
| --- |
| *--学科、校区分组（按照学科、校区、时间和常驻类型统计） --小时* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  itcast\_school\_id,  itcast\_school\_name,  '-1' as origin\_type,  itcast\_subject\_id,  itcast\_subject\_name,  hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo,'-',dayinfo,' ',hourinfo) as time\_str,  '3' as grouptype,  '1' as time\_type,  yearinfo,  monthinfo,  dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by itcast\_school\_id, itcast\_school\_name, itcast\_subject\_id, itcast\_subject\_name, yearinfo, monthinfo, dayinfo, hourinfo, origin\_type\_stat, clue\_state\_stat; *--天* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  itcast\_school\_id,  itcast\_school\_name,  '-1' as origin\_type,  itcast\_subject\_id,  itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo,'-',dayinfo) as time\_str,  '3' as grouptype,  '2' as time\_type,  yearinfo,  monthinfo,  dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by itcast\_school\_id, itcast\_school\_name, itcast\_subject\_id, itcast\_subject\_name, yearinfo, monthinfo, dayinfo, origin\_type\_stat, clue\_state\_stat; *--月* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  itcast\_school\_id,  itcast\_school\_name,  '-1' as origin\_type,  itcast\_subject\_id,  itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo) as time\_str,  '3' as grouptype,  '1' as time\_type,  yearinfo,  monthinfo,  '-1' as dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by itcast\_school\_id, itcast\_school\_name, itcast\_subject\_id, itcast\_subject\_name, yearinfo, monthinfo, origin\_type\_stat, clue\_state\_stat; *--年* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  itcast\_school\_id,  itcast\_school\_name,  '-1' as origin\_type,  itcast\_subject\_id,  itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo) as time\_str,  '3' as grouptype,  '1' as time\_type,  yearinfo,  '-1' as monthinfo,  '-1' as dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by itcast\_school\_id, itcast\_school\_name, itcast\_subject\_id, itcast\_subject\_name, yearinfo, origin\_type\_stat, clue\_state\_stat; |

来源渠道占比

|  |
| --- |
| *--来源渠道分组（按照来源渠道、时间和常驻类型统计） --小时* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  '-1' as itcast\_school\_id,  '-1' as itcast\_school\_name,  origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo,'-',dayinfo,' ',hourinfo) as time\_str,  '4' as grouptype,  '1' as time\_type,  yearinfo,  monthinfo,  dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by origin\_type, yearinfo, monthinfo, dayinfo, hourinfo, origin\_type\_stat, clue\_state\_stat; *--天* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  '-1' as itcast\_school\_id,  '-1' as itcast\_school\_name,  origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo,'-',dayinfo) as time\_str,  '4' as grouptype,  '2' as time\_type,  yearinfo,  monthinfo,  dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by origin\_type, yearinfo, monthinfo, dayinfo, origin\_type\_stat, clue\_state\_stat; *--月* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  '-1' as itcast\_school\_id,  '-1' as itcast\_school\_name,  origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo) as time\_str,  '4' as grouptype,  '1' as time\_type,  yearinfo,  monthinfo,  '-1' as dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by origin\_type, yearinfo, monthinfo, origin\_type\_stat, clue\_state\_stat; *--年* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  '-1' as itcast\_school\_id,  '-1' as itcast\_school\_name,  origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo) as time\_str,  '4' as grouptype,  '1' as time\_type,  yearinfo,  '-1' as monthinfo,  '-1' as dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by origin\_type, yearinfo, origin\_type\_stat, clue\_state\_stat; |

咨询中心占比

|  |
| --- |
| *--咨询中心分组（按照咨询中心、时间和常驻类型统计） --小时* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  '-1' as itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  hourinfo,  origin\_type\_stat,  clue\_state\_stat,  tdepart\_id,  tdepart\_name,  *concat*(yearinfo,'-',monthinfo,'-',dayinfo,' ',hourinfo) as time\_str,  '5' as grouptype,  '1' as time\_type,  yearinfo,  monthinfo,  dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by tdepart\_id, tdepart\_name, yearinfo, monthinfo, dayinfo, hourinfo, origin\_type\_stat, clue\_state\_stat; *--天* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  '-1' as itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  tdepart\_id,  tdepart\_name,  *concat*(yearinfo,'-',monthinfo,'-',dayinfo) as time\_str,  '5' as grouptype,  '2' as time\_type,  yearinfo,  monthinfo,  dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by tdepart\_id, tdepart\_name, yearinfo, monthinfo, dayinfo, origin\_type\_stat, clue\_state\_stat; *--月* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  '-1' as itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  tdepart\_id,  tdepart\_name,  *concat*(yearinfo,'-',monthinfo) as time\_str,  '5' as grouptype,  '1' as time\_type,  yearinfo,  monthinfo,  '-1' as dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by tdepart\_id, tdepart\_name, yearinfo, monthinfo, origin\_type\_stat, clue\_state\_stat; *--年* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  '-1' as itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  tdepart\_id,  tdepart\_name,  *concat*(yearinfo) as time\_str,  '5' as grouptype,  '1' as time\_type,  yearinfo,  '-1' as monthinfo,  '-1' as dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm group by tdepart\_id, tdepart\_name, yearinfo, origin\_type\_stat, clue\_state\_stat; |

##### 测试

由于从ODS—>DWD层—>DWM层，已经通过分桶采样减少了数据，因此在DWS层无需重复采样。

#### 导出数据

##### 创建mysql表

|  |
| --- |
| CREATE TABLE itcast\_intention\_app (  `customer\_total` int(11) COMMENT '聚合意向客户数',  `area` varchar(32) COMMENT '区域信息',  `itcast\_school\_id` varchar(32) COMMENT '校区id',  `itcast\_school\_name` varchar(32) COMMENT '校区名称',  `origin\_type` varchar(32) COMMENT '来源渠道',  `itcast\_subject\_id` varchar(32) COMMENT '学科id',  `itcast\_subject\_name` varchar(32) COMMENT '学科名称',  `hourinfo` varchar(32) COMMENT '小时信息',  `origin\_type\_stat` varchar(32) COMMENT '数据来源:0.线下；1.线上',  `clue\_state\_stat` varchar(32) COMMENT '客户属性：0.老客户；1.新客户',  `tdepart\_id` varchar(32) COMMENT '创建者',  `tdepart\_name` varchar(32) COMMENT '咨询中心名称',  `time\_str` varchar(32) COMMENT '时间明细',  `groupType` varchar(32) COMMENT '产品属性类别：1.总意向量；2.区域信息；3.校区、学科组合分组；4.来源渠道；5.贡献中心;',  `time\_type` varchar(32) COMMENT '聚合时间类型：1、按小时聚合；2、按天聚合；3、按周聚合；4、按月聚合；5、按年聚合；',  `dayinfo` varchar(32) COMMENT '日信息',  `monthinfo` varchar(32) COMMENT '月信息',  `yearinfo` varchar(32) COMMENT '年信息' ); |

##### Sqoop导出脚本

|  |
| --- |
| sqoop export \ *--connect "jdbc:mysql://192.168.52.150:3306/scrm\_bi?useUnicode=true&characterEncoding=utf-8" \ --username root \ --password '123456' \ --table itcast\_intention\_app \ --hcatalog-database itcast\_dws \ --hcatalog-table itcast\_intention\_dws \* -m 100 |

### 增量流程

#### 数据采集

##### Dimen层

###### Customer客户表

维表数据量少，可直接全部覆盖，同全量过程。

###### employee员工表

同全量过程。

###### scrm\_department部门表

同全量过程。

###### itcast\_school学校表

同全量过程。

###### itcast\_subject学科表

同全量过程。

##### ODS层

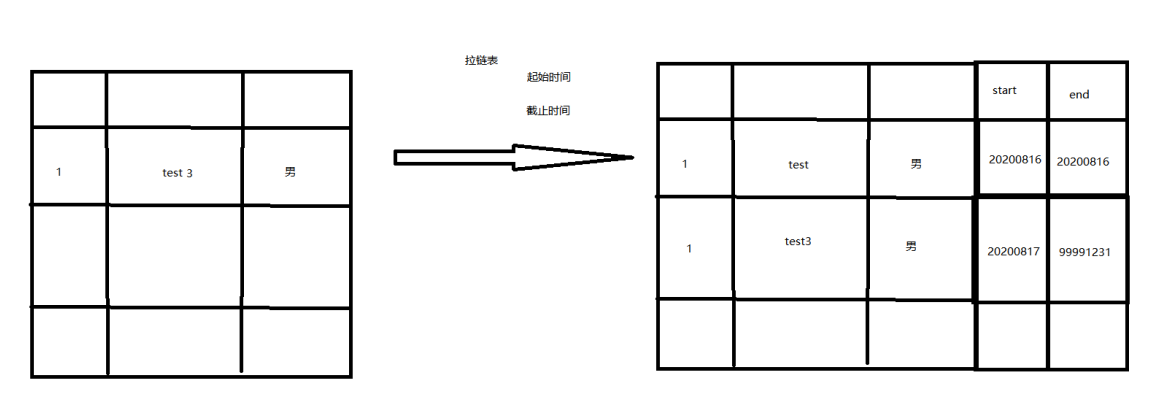
###### 拉链表采集

拉链表回顾

拉链表就是之前我们讲过的SCD2，它的优点是即满足了反应数据的历史状态，又能在最大程度上节省存储。

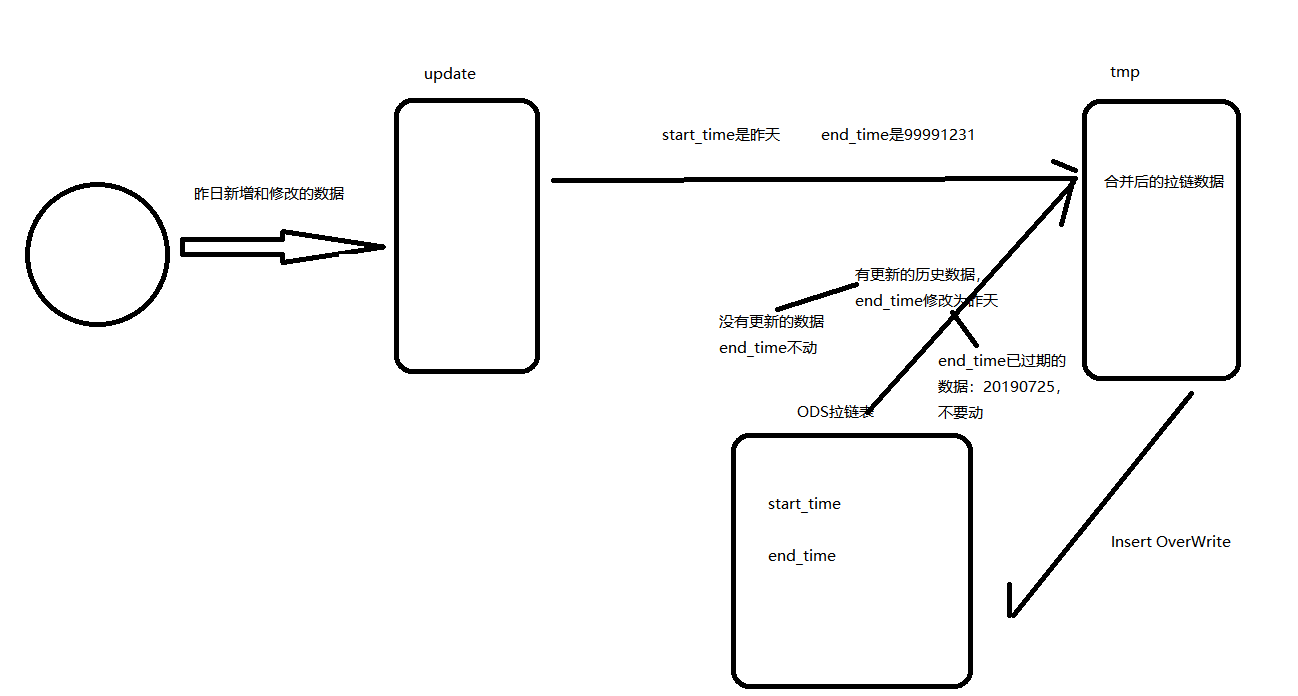
拉链表的实现需要在原始字段基础上增加两个新字段：

* start\_time(表示该条记录的生命周期开始时间——周期快照时的状态)
* end\_time(该条记录的生命周期结束时间)



采集实现步骤

1. 建立增量数据临时表update；
2. 抽取昨日增量数据(新增和更新)到update表；
3. 建立合并数据临时表tmp；
4. 合并昨日增量数据（update表）与历史数据（拉链表）
   1. 新数据end\_time设为’9999-12-31’，也就是当前有效；
   2. 如果增量数据有重复id的旧数据，将旧数据end\_time更新为前天（昨日-1），也就是从昨天开始不再生效；
   3. 合并后的数据写入tmp表；
5. 将临时表的数据，覆盖到拉链表中；
6. 下次抽取需要重建update表和tmp表。



查询拉链表数据时，可以通过start\_time和end\_time查询出快照数据。

###### Customer\_relationship

因为需求需要将customer\_relationship更新数据涉及到的维度重新统计；同时要有历史快照。推荐采用拉链表(SCD2)的形式来做。需要在start\_time字段的基础上，增加新的end\_time字段，以标识封链时间。

重建customer\_relationship\_update增量表

每次使用update表都需要重建，以避免因为数据重复而导致的问题。

|  |
| --- |
| DROP TABLE IF EXISTS itcast\_ods.customer\_relationship\_update; CREATE TABLE IF NOT EXISTS itcast\_ods.customer\_relationship\_update (  id int COMMENT '客户关系id',  create\_date\_time STRING COMMENT '创建时间',  update\_date\_time STRING COMMENT '最后更新时间',  deleted int COMMENT '是否被删除（禁用）',  customer\_id int COMMENT '所属客户id',  first\_id int COMMENT '第一条客户关系id',  belonger int COMMENT '归属人',  belonger\_name STRING COMMENT '归属人姓名',  initial\_belonger int COMMENT '初始归属人',  distribution\_handler int COMMENT '分配处理人',  business\_scrm\_department\_id int COMMENT '归属部门',  last\_visit\_time STRING COMMENT '最后回访时间',  next\_visit\_time STRING COMMENT '下次回访时间',  origin\_type STRING COMMENT '数据来源',  itcast\_school\_id int COMMENT '校区Id',  itcast\_subject\_id int COMMENT '学科Id',  intention\_study\_type STRING COMMENT '意向学习方式',  anticipat\_signup\_date STRING COMMENT '预计报名时间',  level STRING COMMENT '客户级别',  creator int COMMENT '创建人',  current\_creator int COMMENT '当前创建人：初始==创建人，当在公海拉回时为 拉回人',  creator\_name STRING COMMENT '创建者姓名',  origin\_channel STRING COMMENT '来源渠道',  comment STRING COMMENT '备注',  first\_customer\_clue\_id int COMMENT '第一条线索id',  last\_customer\_clue\_id int COMMENT '最后一条线索id',  process\_state STRING COMMENT '处理状态',  process\_time STRING COMMENT '处理状态变动时间',  payment\_state STRING COMMENT '支付状态',  payment\_time STRING COMMENT '支付状态变动时间',  signup\_state STRING COMMENT '报名状态',  signup\_time STRING COMMENT '报名时间',  notice\_state STRING COMMENT '通知状态',  notice\_time STRING COMMENT '通知状态变动时间',  lock\_state STRING COMMENT '锁定状态',  lock\_time STRING COMMENT '锁定状态修改时间',  itcast\_clazz\_id int COMMENT '所属ems班级id',  itcast\_clazz\_time STRING COMMENT '报班时间',  payment\_url STRING COMMENT '付款链接',  payment\_url\_time STRING COMMENT '支付链接生成时间',  ems\_student\_id int COMMENT 'ems的学生id',  delete\_reason STRING COMMENT '删除原因',  deleter int COMMENT '删除人',  deleter\_name STRING COMMENT '删除人姓名',  delete\_time STRING COMMENT '删除时间',  course\_id int COMMENT '课程ID',  course\_name STRING COMMENT '课程名称',  delete\_comment STRING COMMENT '删除原因说明',  close\_state STRING COMMENT '关闭装填',  close\_time STRING COMMENT '关闭状态变动时间',  appeal\_id int COMMENT '申诉id',  tenant int COMMENT '租户',  total\_fee DECIMAL COMMENT '报名费总金额',  belonged int COMMENT '小周期归属人',  belonged\_time STRING COMMENT '归属时间',  belonger\_time STRING COMMENT '归属时间',  transfer int COMMENT '转移人',  transfer\_time STRING COMMENT '转移时间',  follow\_type int COMMENT '分配类型，0-自动分配，1-手动分配，2-自动转移，3-手动单个转移，4-手动批量转移，5-公海领取',  transfer\_bxg\_oa\_account STRING COMMENT '转移到博学谷归属人OA账号',  transfer\_bxg\_belonger\_name STRING COMMENT '转移到博学谷归属人OA姓名',  end\_time STRING COMMENT '有效时间') comment '客户关系表' PARTITIONED BY(start\_time STRING) ROW FORMAT DELIMITED  FIELDS TERMINATED BY '\t' stored as orc TBLPROPERTIES ('orc.compress'='ZLIB'); |

抽取昨日新增和更新数据（逻辑删除也属于更新操作）

因为增量抽取是T+1，所以Sql中需要增加where条件，只查询昨天一天的数据（新增和更新），而不是所有表数据。

新增的数据create\_time=昨天；更新的数据update\_time=昨天。

注意，更新的数据可能是以前创建的数据，创建日期可能不是昨天。业务方将更新周期限制在30天内，也就是说，昨天更改的数据，create\_time<=’30天前的日期’，而update\_time的值就是昨天的日期。

查询条件需要包含创建日期和更新日期，因为需要将昨日新增和修改的数据都抽取到数仓中。

SQL：

|  |
| --- |
| select id,  create\_date\_time,  update\_date\_time,  deleted,  customer\_id,  first\_id,  belonger,  belonger\_name,  initial\_belonger,  distribution\_handler,  business\_scrm\_department\_id,  last\_visit\_time,  next\_visit\_time,  origin\_type,  itcast\_school\_id,  itcast\_subject\_id,  intention\_study\_type,  anticipat\_signup\_date,  level,  creator,  current\_creator,  creator\_name,  origin\_channel,  comment,  first\_customer\_clue\_id,  last\_customer\_clue\_id,  process\_state,  process\_time,  payment\_state,  payment\_time,  signup\_state,  signup\_time,  notice\_state,  notice\_time,  lock\_state,  lock\_time,  itcast\_clazz\_id,  itcast\_clazz\_time,  payment\_url,  payment\_url\_time,  ems\_student\_id,  delete\_reason,  deleter,  deleter\_name,  delete\_time,  course\_id,  course\_name,  delete\_comment,  close\_state,  close\_time,  appeal\_id,  tenant,  total\_fee,  belonged,  belonged\_time,  belonger\_time,  transfer,  transfer\_time,  follow\_type,  transfer\_bxg\_oa\_account,  transfer\_bxg\_belonger\_name,  *FROM\_UNIXTIME*(*unix\_timestamp*(), "%Y-%m-%d") as start\_time,  "9999-12-31" as end\_time from customer\_relationship where  (  create\_date\_time >= "2011-12-04 00:00:00"  and  create\_date\_time < "2011-12-05 00:00:00"  )  or  (  update\_date\_time >= "2011-12-04 00:00:00"  and  update\_date\_time < "2011-12-05 00:00:00"  ); |

Sqoop脚本：

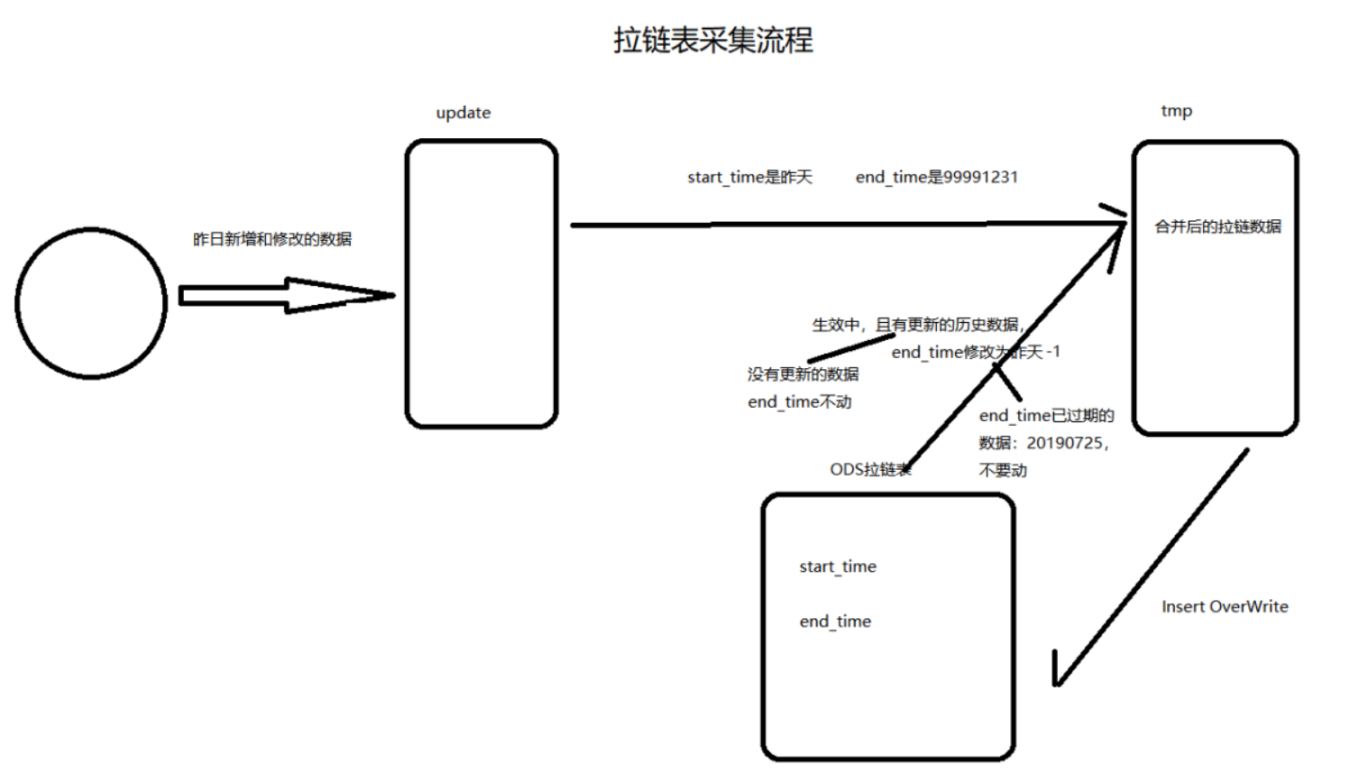
|  |
| --- |
| sqoop import \ *--connect jdbc:mysql://192.168.52.150:3306/scrm \ --username root \ --password 123456 \ --query '*  select id,  create\_date\_time,  update\_date\_time,  deleted,  customer\_id,  first\_id,  belonger,  belonger\_name,  initial\_belonger,  distribution\_handler,  business\_scrm\_department\_id,  last\_visit\_time,  next\_visit\_time,  origin\_type,  itcast\_school\_id,  itcast\_subject\_id,  intention\_study\_type,  anticipat\_signup\_date,  level,  creator,  current\_creator,  creator\_name,  origin\_channel,  comment,  first\_customer\_clue\_id,  last\_customer\_clue\_id,  process\_state,  process\_time,  payment\_state,  payment\_time,  signup\_state,  signup\_time,  notice\_state,  notice\_time,  lock\_state,  lock\_time,  itcast\_clazz\_id,  itcast\_clazz\_time,  payment\_url,  payment\_url\_time,  ems\_student\_id,  delete\_reason,  deleter,  deleter\_name,  delete\_time,  course\_id,  course\_name,  delete\_comment,  close\_state,  close\_time,  appeal\_id,  tenant,  total\_fee,  belonged,  belonged\_time,  belonger\_time,  transfer,  transfer\_time,  follow\_type,  transfer\_bxg\_oa\_account,  transfer\_bxg\_belonger\_name,  *FROM\_UNIXTIME*(*unix\_timestamp*(), "%Y-%m-%d") as start\_time,  *date\_format*("9999-12-31", "%Y-%m-%d") as end\_time from customer\_relationship where  (  create\_date\_time >= "2011-12-04 00:00:00"  and  create\_date\_time < "2011-12-05 00:00:00"  )  or  (  update\_date\_time >= "2011-12-04 00:00:00"  and  update\_date\_time < "2011-12-05 00:00:00"  )  and $CONDITIONS' \ --hcatalog-database itcast\_ods \ --hcatalog-table customer\_relationship\_update \ --hive-partition-key start\_time \ --hive-partition-value 2020-07-15 \ -m 100 \ --split-by id |

重建customer\_relationship\_tmp临时表

每次使用tmp表都需要重建，以避免因为数据重复而导致的问题。

|  |
| --- |
| DROP TABLE itcast\_ods.`customer\_relationship\_tmp`; CREATE TABLE IF NOT EXISTS itcast\_ods.`customer\_relationship\_tmp` (  `id` int COMMENT '客户关系id',  `create\_date\_time` STRING COMMENT '创建时间',  `update\_date\_time` STRING COMMENT '最后更新时间',  `deleted` int COMMENT '是否被删除（禁用）',  `customer\_id` int COMMENT '所属客户id',  `first\_id` int COMMENT '第一条客户关系id',  `belonger` int COMMENT '归属人',  `belonger\_name` STRING COMMENT '归属人姓名',  `initial\_belonger` int COMMENT '初始归属人',  `distribution\_handler` int COMMENT '分配处理人',  `business\_scrm\_department\_id` int COMMENT '归属部门',  `last\_visit\_time` STRING COMMENT '最后回访时间',  `next\_visit\_time` STRING COMMENT '下次回访时间',  `origin\_type` STRING COMMENT '数据来源',  `itcast\_school\_id` int COMMENT '校区Id',  `itcast\_subject\_id` int COMMENT '学科Id',  `intention\_study\_type` STRING COMMENT '意向学习方式',  `anticipat\_signup\_date` STRING COMMENT '预计报名时间',  `level` STRING COMMENT '客户级别',  `creator` int COMMENT '创建人',  `current\_creator` int COMMENT '当前创建人：初始==创建人，当在公海拉回时为 拉回人',  `creator\_name` STRING COMMENT '创建者姓名',  `origin\_channel` STRING COMMENT '来源渠道',  `comment` STRING COMMENT '备注',  `first\_customer\_clue\_id` int COMMENT '第一条线索id',  `last\_customer\_clue\_id` int COMMENT '最后一条线索id',  `process\_state` STRING COMMENT '处理状态',  `process\_time` STRING COMMENT '处理状态变动时间',  `payment\_state` STRING COMMENT '支付状态',  `payment\_time` STRING COMMENT '支付状态变动时间',  `signup\_state` STRING COMMENT '报名状态',  `signup\_time` STRING COMMENT '报名时间',  `notice\_state` STRING COMMENT '通知状态',  `notice\_time` STRING COMMENT '通知状态变动时间',  `lock\_state` STRING COMMENT '锁定状态',  `lock\_time` STRING COMMENT '锁定状态修改时间',  `itcast\_clazz\_id` int COMMENT '所属ems班级id',  `itcast\_clazz\_time` STRING COMMENT '报班时间',  `payment\_url` STRING COMMENT '付款链接',  `payment\_url\_time` STRING COMMENT '支付链接生成时间',  `ems\_student\_id` int COMMENT 'ems的学生id',  `delete\_reason` STRING COMMENT '删除原因',  `deleter` int COMMENT '删除人',  `deleter\_name` STRING COMMENT '删除人姓名',  `delete\_time` STRING COMMENT '删除时间',  `course\_id` int COMMENT '课程ID',  `course\_name` STRING COMMENT '课程名称',  `delete\_comment` STRING COMMENT '删除原因说明',  `close\_state` STRING COMMENT '关闭装填',  `close\_time` STRING COMMENT '关闭状态变动时间',  `appeal\_id` int COMMENT '申诉id',  `tenant` int COMMENT '租户',  `total\_fee` DECIMAL COMMENT '报名费总金额',  `belonged` int COMMENT '小周期归属人',  `belonged\_time` STRING COMMENT '归属时间',  `belonger\_time` STRING COMMENT '归属时间',  `transfer` int COMMENT '转移人',  `transfer\_time` STRING COMMENT '转移时间',  `follow\_type` int COMMENT '分配类型，0-自动分配，1-手动分配，2-自动转移，3-手动单个转移，4-手动批量转移，5-公海领取',  `transfer\_bxg\_oa\_account` STRING COMMENT '转移到博学谷归属人OA账号',  `transfer\_bxg\_belonger\_name` STRING COMMENT '转移到博学谷归属人OA姓名',  `end\_time` STRING COMMENT '有效截止时间') comment '客户关系表' PARTITIONED BY(start\_time STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' stored as orc TBLPROPERTIES ('orc.compress'='ZLIB'); |

合并增量数据与历史数据（根据需求仅更新30天之内的数据）



1. 获取update表的更新数据，新数据end\_time为’9999-12-31’，start\_time为昨日日期；
2. 获取拉链表历史数据：
   1. 更新旧数据end\_time
      1. 将历史表customer\_relationship（拉链表）与新增/更新数据表customer\_relationship\_update通过id进行关联，如果update中有与历史表重复的id，证明有此条id数据已有新的变更；
      2. end\_time不变的条件：
         1. 没有更新的数据保留原始end\_time；
         2. 历史表已是失效的数据，保留原始有效结束日期end\_time；
      3. 否则（有更新的数据，且旧数据目前正在生效），修改end\_time为前天（昨天之前）；
   2. 因为业务方将更新周期限制在30天内(只会修改30天之内的数据，即create\_time在30天之内)，所以只需查询更新30天内的数据(end\_time)即可；
3. 将 1.update 与 2.拉链表 合并，覆盖插入到临时表中。

实现：

|  |
| --- |
| **insert overwrite table** itcast\_ods.customer\_relationship\_tmp **partition** (**start\_time**) **select** *\** **from** ( *-- 一、update表更新的数据* **select  id**,  **create\_date\_time**,  **update\_date\_time**,  **deleted**,  **customer\_id**,  **first\_id**,  **belonger**,  **belonger\_name**,  **initial\_belonger**,  **distribution\_handler**,  **business\_scrm\_department\_id**,  **last\_visit\_time**,  **next\_visit\_time**,  **origin\_type**,  **itcast\_school\_id**,  **itcast\_subject\_id**,  **intention\_study\_type**,  **anticipat\_signup\_date**,  **level**,  **creator**,  **current\_creator**,  **creator\_name**,  **origin\_channel**,  **comment**,  **first\_customer\_clue\_id**,  **last\_customer\_clue\_id**,  **process\_state**,  **process\_time**,  **payment\_state**,  **payment\_time**,  **signup\_state**,  **signup\_time**,  **notice\_state**,  **notice\_time**,  **lock\_state**,  **lock\_time**,  **itcast\_clazz\_id**,  **itcast\_clazz\_time**,  **payment\_url**,  **payment\_url\_time**,  **ems\_student\_id**,  **delete\_reason**,  **deleter**,  **deleter\_name**,  **delete\_time**,  **course\_id**,  **course\_name**,  **delete\_comment**,  **close\_state**,  **close\_time**,  **appeal\_id**,  **tenant**,  **total\_fee**,  **belonged**,  **belonged\_time**,  **belonger\_time**,  **transfer**,  **transfer\_time**,  **follow\_type**,  **transfer\_bxg\_oa\_account**,  **transfer\_bxg\_belonger\_name**,  **'9999-12-31'** end\_time,  **'2020-07-15' as** start\_time  **from** itcast\_ods.customer\_relationship\_update **where** start\_time=**'2020-07-15'   union all** *-- 二、历史拉链表数据，并根据update判断更新end\_time有效期* **select** rs.**id**,  rs.**create\_date\_time**,  rs.**update\_date\_time**,  rs.**deleted**,  rs.**customer\_id**,  rs.**first\_id**,  rs.**belonger**,  rs.**belonger\_name**,  rs.**initial\_belonger**,  rs.**distribution\_handler**,  rs.**business\_scrm\_department\_id**,  rs.**last\_visit\_time**,  rs.**next\_visit\_time**,  rs.**origin\_type**,  rs.**itcast\_school\_id**,  rs.**itcast\_subject\_id**,  rs.**intention\_study\_type**,  rs.**anticipat\_signup\_date**,  rs.**level**,  rs.**creator**,  rs.**current\_creator**,  rs.**creator\_name**,  rs.**origin\_channel**,  rs.**comment**,  rs.**first\_customer\_clue\_id**,  rs.**last\_customer\_clue\_id**,  rs.**process\_state**,  rs.**process\_time**,  rs.**payment\_state**,  rs.**payment\_time**,  rs.**signup\_state**,  rs.**signup\_time**,  rs.**notice\_state**,  rs.**notice\_time**,  rs.**lock\_state**,  rs.**lock\_time**,  rs.**itcast\_clazz\_id**,  rs.**itcast\_clazz\_time**,  rs.**payment\_url**,  rs.**payment\_url\_time**,  rs.**ems\_student\_id**,  rs.**delete\_reason**,  rs.**deleter**,  rs.**deleter\_name**,  rs.**delete\_time**,  rs.**course\_id**,  rs.**course\_name**,  rs.**delete\_comment**,  rs.**close\_state**,  rs.**close\_time**,  rs.**appeal\_id**,  rs.**tenant**,  rs.**total\_fee**,  rs.**belonged**,  rs.**belonged\_time**,  rs.**belonger\_time**,  rs.**transfer**,  rs.**transfer\_time**,  rs.**follow\_type**,  rs.**transfer\_bxg\_oa\_account**,  rs.**transfer\_bxg\_belonger\_name**,  *--3、更新end\_time：如果没有匹配到变更数据，或者当前已经是无效的历史数据，则保留原始end\_time过期时间；否则变更end\_time时间为前天（昨天之前有效）  if*(**up.id is null or rs.end\_time<'9999-12-31'**, rs.**end\_time**, *date\_add*(up.**start\_time**,-1)) end\_time,  rs.**start\_time  from** itcast\_ods.customer\_relationship rs **left join** (  **select** *\** **from** itcast\_ods.customer\_relationship\_update  **where start\_time**=**'2020-07-15'** ) up  **on** rs.**id**=up.**id** *--4、时间限制：历史表中30天之内的数据才有可能变更，结果会按照所属分区进行覆盖插入* **where** rs.**start\_time** >= *date\_add*(up.**start\_time**,-30)  )his **order by** his.**id**, start\_time; |

临时表覆盖到拉链表

注意如果有分区的情况下，只会覆盖所属分区的数据，所以不用在上一个步骤中查询出所有历史数据，我们只需要查询出30天内的数据即可，30天前的数据不会被覆盖。

|  |
| --- |
| INSERT OVERWRITE TABLE itcast\_ods.customer\_relationship partition (start\_time)  SELECT *\** from itcast\_ods.customer\_relationship\_tmp; |

测试

完整执行流程后，观察拉链表中对应条件的数据是否有变化：

|  |
| --- |
| SELECT *\** from itcast\_ods.customer\_relationship WHERE create\_date\_time BETWEEN "2011-12-04 00:00:00" and "2011-12-05 00:00:00"; |

Oozie脚本

将拉链表的完整过程写入到shell脚本中。

|  |
| --- |
| #! /bin/bash  HIVE\_HOME=/usr/bin/hive  if [[ $1 == "" ]];  then  TD\_DATE=`date -d ''1 days ago'' "+%Y-%m-%d"`  else  TD\_DATE=$1  fi  output=$(${HIVE\_HOME} -S -e "  SET hive.exec.dynamic.partition=true;  SET hive.exec.dynamic.partition.mode=nonstrict;  DROP TABLE IF EXISTS itcast\_ods.customer\_relationship\_update;  CREATE TABLE IF NOT EXISTS itcast\_ods.customer\_relationship\_update (  id int COMMENT '客户关系id',  create\_date\_time STRING COMMENT '创建时间',  update\_date\_time STRING COMMENT '最后更新时间',  deleted int COMMENT '是否被删除（禁用）',  customer\_id int COMMENT '所属客户id',  first\_id int COMMENT '第一条客户关系id',  belonger int COMMENT '归属人',  belonger\_name STRING COMMENT '归属人姓名',  initial\_belonger int COMMENT '初始归属人',  distribution\_handler int COMMENT '分配处理人',  business\_scrm\_department\_id int COMMENT '归属部门',  last\_visit\_time STRING COMMENT '最后回访时间',  next\_visit\_time STRING COMMENT '下次回访时间',  origin\_type STRING COMMENT '数据来源',  itcast\_school\_id int COMMENT '校区Id',  itcast\_subject\_id int COMMENT '学科Id',  intention\_study\_type STRING COMMENT '意向学习方式',  anticipat\_signup\_date STRING COMMENT '预计报名时间',  level STRING COMMENT '客户级别',  creator int COMMENT '创建人',  current\_creator int COMMENT '当前创建人：初始==创建人，当在公海拉回时为 拉回人',  creator\_name STRING COMMENT '创建者姓名',  origin\_channel STRING COMMENT '来源渠道',  comment STRING COMMENT '备注',  first\_customer\_clue\_id int COMMENT '第一条线索id',  last\_customer\_clue\_id int COMMENT '最后一条线索id',  process\_state STRING COMMENT '处理状态',  process\_time STRING COMMENT '处理状态变动时间',  payment\_state STRING COMMENT '支付状态',  payment\_time STRING COMMENT '支付状态变动时间',  signup\_state STRING COMMENT '报名状态',  signup\_time STRING COMMENT '报名时间',  notice\_state STRING COMMENT '通知状态',  notice\_time STRING COMMENT '通知状态变动时间',  lock\_state STRING COMMENT '锁定状态',  lock\_time STRING COMMENT '锁定状态修改时间',  itcast\_clazz\_id int COMMENT '所属ems班级id',  itcast\_clazz\_time STRING COMMENT '报班时间',  payment\_url STRING COMMENT '付款链接',  payment\_url\_time STRING COMMENT '支付链接生成时间',  ems\_student\_id int COMMENT 'ems的学生id',  delete\_reason STRING COMMENT '删除原因',  deleter int COMMENT '删除人',  deleter\_name STRING COMMENT '删除人姓名',  delete\_time STRING COMMENT '删除时间',  course\_id int COMMENT '课程ID',  course\_name STRING COMMENT '课程名称',  delete\_comment STRING COMMENT '删除原因说明',  close\_state STRING COMMENT '关闭装填',  close\_time STRING COMMENT '关闭状态变动时间',  appeal\_id int COMMENT '申诉id',  tenant int COMMENT '租户',  total\_fee DECIMAL COMMENT '报名费总金额',  belonged int COMMENT '小周期归属人',  belonged\_time STRING COMMENT '归属时间',  belonger\_time STRING COMMENT '归属时间',  transfer int COMMENT '转移人',  transfer\_time STRING COMMENT '转移时间',  follow\_type int COMMENT '分配类型，0-自动分配，1-手动分配，2-自动转移，3-手动单个转移，4-手动批量转移，5-公海领取',  transfer\_bxg\_oa\_account STRING COMMENT '转移到博学谷归属人OA账号',  transfer\_bxg\_belonger\_name STRING COMMENT '转移到博学谷归属人OA姓名',  end\_time STRING COMMENT '有效时间')  comment '客户关系表'  PARTITIONED BY(start\_time STRING)  ROW FORMAT DELIMITED  FIELDS TERMINATED BY '\t'  stored as orc  TBLPROPERTIES ('orc.compress'='ZLIB');  ")  SQOOP\_HOME=/usr/bin/sqoop  output=$(${SQOOP\_HOME} import \  --connect jdbc:mysql://172.17.0.202:3306/scrm \  --username root \  --password 123456 \  --query 'select id,  create\_date\_time,  update\_date\_time,  deleted,  customer\_id,  first\_id,  belonger,  belonger\_name,  initial\_belonger,  distribution\_handler,  business\_scrm\_department\_id,  last\_visit\_time,  next\_visit\_time,  origin\_type,  itcast\_school\_id,  itcast\_subject\_id,  intention\_study\_type,  anticipat\_signup\_date,  level,  creator,  current\_creator,  creator\_name,  origin\_channel,  comment,  first\_customer\_clue\_id,  last\_customer\_clue\_id,  process\_state,  process\_time,  payment\_state,  payment\_time,  signup\_state,  signup\_time,  notice\_state,  notice\_time,  lock\_state,  lock\_time,  itcast\_clazz\_id,  itcast\_clazz\_time,  payment\_url,  payment\_url\_time,  ems\_student\_id,  delete\_reason,  deleter,  deleter\_name,  delete\_time,  course\_id,  course\_name,  delete\_comment,  close\_state,  close\_time,  appeal\_id,  tenant,  total\_fee,  belonged,  belonged\_time,  belonger\_time,  transfer,  transfer\_time,  follow\_type,  transfer\_bxg\_oa\_account,  transfer\_bxg\_belonger\_name,  FROM\_UNIXTIME(unix\_timestamp(), "%Y-%m-%d") as start\_time,  date\_format("9999-12-31", "%Y-%m-%d") as end\_time  from customer\_relationship  where  (  create\_date\_time >= FROM\_UNIXTIME(UNIX\_TIMESTAMP(CAST(SYSDATE()AS DATE) - INTERVAL 1 DAY),"%Y-%m-%d %H:%i:%s")  and  create\_date\_time < FROM\_UNIXTIME(UNIX\_TIMESTAMP(CAST(SYSDATE()AS DATE)),"%Y-%m-%d %H:%i:%s")  )  or  (  update\_date\_time >= FROM\_UNIXTIME(UNIX\_TIMESTAMP(CAST(SYSDATE()AS DATE) - INTERVAL 1 DAY),"%Y-%m-%d %H:%i:%s")  and  update\_date\_time < FROM\_UNIXTIME(UNIX\_TIMESTAMP(CAST(SYSDATE()AS DATE)),"%Y-%m-%d %H:%i:%s")  ) and $CONDITIONS' \  --hcatalog-database itcast\_ods \  --hcatalog-table customer\_relationship\_update \  --hive-partition-key start\_time \  --hive-partition-value ${TD\_DATE} \  -m 100 \  --split-by id)    output=$(${HIVE\_HOME} -S -e "  SET hive.exec.dynamic.partition=true;  SET hive.exec.dynamic.partition.mode=nonstrict;  DROP TABLE itcast\_ods.customer\_clue\_tmp;  CREATE TABLE IF NOT EXISTS itcast\_ods.customer\_clue\_tmp (  id int COMMENT 'customer\_clue\_id',  create\_date\_time STRING COMMENT '创建时间',  update\_date\_time STRING COMMENT '最后更新时间',  deleted STRING COMMENT '是否被删除（禁用）',  customer\_id int COMMENT '客户id',  customer\_relationship\_id int COMMENT '客户关系id',  session\_id STRING COMMENT '七陌会话id',  sid STRING COMMENT '访客id',  status STRING COMMENT '状态（undeal待领取 deal 已领取 finish 已关闭 changePeer 已流转）',  users STRING COMMENT '所属坐席',  create\_time STRING COMMENT '七陌创建时间',  platform STRING COMMENT '平台来源 （pc-网站咨询|wap-wap咨询|sdk-app咨询|weixin-微信咨询）',  s\_name STRING COMMENT '用户名称',  seo\_source STRING COMMENT '搜索来源',  seo\_keywords STRING COMMENT '关键字',  ip STRING COMMENT 'IP地址',  referrer STRING COMMENT '上级来源页面',  from\_url STRING COMMENT '会话来源页面',  landing\_page\_url STRING COMMENT '访客着陆页面',  url\_title STRING COMMENT '咨询页面title',  to\_peer STRING COMMENT '所属技能组',  manual\_time STRING COMMENT '人工开始时间',  begin\_time STRING COMMENT '坐席领取时间 ',  reply\_msg\_count int COMMENT '客服回复消息数',  total\_msg\_count int COMMENT '消息总数',  msg\_count int COMMENT '客户发送消息数',  comment STRING COMMENT '备注',  finish\_reason STRING COMMENT '结束类型',  finish\_user STRING COMMENT '结束坐席',  end\_time STRING COMMENT '会话结束时间',  platform\_description STRING COMMENT '客户平台信息',  browser\_name STRING COMMENT '浏览器名称',  os\_info STRING COMMENT '系统名称',  area STRING COMMENT '区域',  country STRING COMMENT '所在国家',  province STRING COMMENT '省',  city STRING COMMENT '城市',  creator int COMMENT '创建人',  name STRING COMMENT '客户姓名',  idcard STRING COMMENT '身份证号',  phone STRING COMMENT '手机号',  itcast\_school\_id int COMMENT '校区Id',  itcast\_school STRING COMMENT '校区',  itcast\_subject\_id int COMMENT '学科Id',  itcast\_subject STRING COMMENT '学科',  wechat STRING COMMENT '微信',  qq STRING COMMENT 'qq号',  email STRING COMMENT '邮箱',  gender STRING COMMENT '性别',  level STRING COMMENT '客户级别',  origin\_type STRING COMMENT '数据来源渠道',  information\_way STRING COMMENT '资讯方式',  working\_years STRING COMMENT '开始工作时间',  technical\_directions STRING COMMENT '技术方向',  customer\_state STRING COMMENT '当前客户状态',  valid STRING COMMENT '该线索是否是网资有效线索',  anticipat\_signup\_date STRING COMMENT '预计报名时间',  clue\_state STRING COMMENT '线索状态',  scrm\_department\_id int COMMENT 'SCRM内部部门id',  superior\_url STRING COMMENT '诸葛获取上级页面URL',  superior\_source STRING COMMENT '诸葛获取上级页面URL标题',  landing\_url STRING COMMENT '诸葛获取着陆页面URL',  landing\_source STRING COMMENT '诸葛获取着陆页面URL来源',  info\_url STRING COMMENT '诸葛获取留咨页URL',  info\_source STRING COMMENT '诸葛获取留咨页URL标题',  origin\_channel STRING COMMENT '投放渠道',  course\_id int COMMENT '课程编号',  course\_name STRING COMMENT '课程名称',  zhuge\_session\_id STRING COMMENT 'zhuge会话id',  is\_repeat int COMMENT '是否重复线索(手机号维度) 0:正常 1：重复',  tenant int COMMENT '租户id',  activity\_id STRING COMMENT '活动id',  activity\_name STRING COMMENT '活动名称',  follow\_type int COMMENT '分配类型，0-自动分配，1-手动分配，2-自动转移，3-手动单个转移，4-手动批量转移，5-公海领取',  shunt\_mode\_id int COMMENT '匹配到的技能组id',  shunt\_employee\_group\_id int COMMENT '所属分流员工组',  ends\_time STRING COMMENT '有效时间')  comment '客户关系表'  PARTITIONED BY(starts\_time STRING)  ROW FORMAT DELIMITED  FIELDS TERMINATED BY '\t'  stored as orc  TBLPROPERTIES ('orc.compress'='ZLIB');  insert overwrite table itcast\_ods.`customer\_relationship\_tmp` partition (start\_time)  select \* from  (  select  id,  create\_date\_time,  update\_date\_time,  deleted,  customer\_id,  first\_id,  belonger,  belonger\_name,  initial\_belonger,  distribution\_handler,  business\_scrm\_department\_id,  last\_visit\_time,  next\_visit\_time,  origin\_type,  itcast\_school\_id,  itcast\_subject\_id,  intention\_study\_type,  anticipat\_signup\_date,  level,  creator,  current\_creator,  creator\_name,  origin\_channel,  comment,  first\_customer\_clue\_id,  last\_customer\_clue\_id,  process\_state,  process\_time,  payment\_state,  payment\_time,  signup\_state,  signup\_time,  notice\_state,  notice\_time,  lock\_state,  lock\_time,  itcast\_clazz\_id,  itcast\_clazz\_time,  payment\_url,  payment\_url\_time,  ems\_student\_id,  delete\_reason,  deleter,  deleter\_name,  delete\_time,  course\_id,  course\_name,  delete\_comment,  close\_state,  close\_time,  appeal\_id,  tenant,  total\_fee,  belonged,  belonged\_time,  belonger\_time,  transfer,  transfer\_time,  follow\_type,  transfer\_bxg\_oa\_account,  transfer\_bxg\_belonger\_name,  '9999-12-31' end\_time,  FROM\_UNIXTIME(unix\_timestamp(), "%Y-%m-%d") as start\_time  from itcast\_ods.customer\_relationship\_update where start\_time=FROM\_UNIXTIME(unix\_timestamp(), "%Y-%m-%d")  union all  select  rs.id,  rs.create\_date\_time,  rs.update\_date\_time,  rs.deleted,  rs.customer\_id,  rs.first\_id,  rs.belonger,  rs.belonger\_name,  rs.initial\_belonger,  rs.distribution\_handler,  rs.business\_scrm\_department\_id,  rs.last\_visit\_time,  rs.next\_visit\_time,  rs.origin\_type,  rs.itcast\_school\_id,  rs.itcast\_subject\_id,  rs.intention\_study\_type,  rs.anticipat\_signup\_date,  rs.level,  rs.creator,  rs.current\_creator,  rs.creator\_name,  rs.origin\_channel,  rs.comment,  rs.first\_customer\_clue\_id,  rs.last\_customer\_clue\_id,  rs.process\_state,  rs.process\_time,  rs.payment\_state,  rs.payment\_time,  rs.signup\_state,  rs.signup\_time,  rs.notice\_state,  rs.notice\_time,  rs.lock\_state,  rs.lock\_time,  rs.itcast\_clazz\_id,  rs.itcast\_clazz\_time,  rs.payment\_url,  rs.payment\_url\_time,  rs.ems\_student\_id,  rs.delete\_reason,  rs.deleter,  rs.deleter\_name,  rs.delete\_time,  rs.course\_id,  rs.course\_name,  rs.delete\_comment,  rs.close\_state,  rs.close\_time,  rs.appeal\_id,  rs.tenant,  rs.total\_fee,  rs.belonged,  rs.belonged\_time,  rs.belonger\_time,  rs.transfer,  rs.transfer\_time,  rs.follow\_type,  rs.transfer\_bxg\_oa\_account,  rs.transfer\_bxg\_belonger\_name,  if(up.id is null, rs.end\_time, date\_add(up.start\_time,-1)) end\_time,  rs.start\_time  from itcast\_ods.customer\_relationship rs left join  (  select  \*  from itcast\_ods.customer\_relationship\_update  where start\_time=FROM\_UNIXTIME(unix\_timestamp(), "%Y-%m-%d")  ) up  on rs.id=up.id where rs.start\_time >= date\_sub(FROM\_UNIXTIME(UNIX\_TIMESTAMP()),30) and rs.end\_time='9999-12-31'  )his  order by his.id, start\_time;  INSERT OVERWRITE TABLE itcast\_ods.customer\_relationship partition (start\_time)  SELECT \* from itcast\_ods.customer\_relationship\_tmp;  ") |

###### Customer\_clue线索表

重建customer\_clue\_update更新表

|  |
| --- |
| DROP TABLE itcast\_ods.customer\_clue\_update; CREATE TABLE IF NOT EXISTS itcast\_ods.customer\_clue\_update (  id int COMMENT 'customer\_clue\_id',  create\_date\_time STRING COMMENT '创建时间',  update\_date\_time STRING COMMENT '最后更新时间',  deleted STRING COMMENT '是否被删除（禁用）',  customer\_id int COMMENT '客户id',  customer\_relationship\_id int COMMENT '客户关系id',  session\_id STRING COMMENT '七陌会话id',  sid STRING COMMENT '访客id',  status STRING COMMENT '状态（undeal待领取 deal 已领取 finish 已关闭 changePeer 已流转）',  users STRING COMMENT '所属坐席',  create\_time STRING COMMENT '七陌创建时间',  platform STRING COMMENT '平台来源 （pc-网站咨询|wap-wap咨询|sdk-app咨询|weixin-微信咨询）',  s\_name STRING COMMENT '用户名称',  seo\_source STRING COMMENT '搜索来源',  seo\_keywords STRING COMMENT '关键字',  ip STRING COMMENT 'IP地址',  referrer STRING COMMENT '上级来源页面',  from\_url STRING COMMENT '会话来源页面',  landing\_page\_url STRING COMMENT '访客着陆页面',  url\_title STRING COMMENT '咨询页面title',  to\_peer STRING COMMENT '所属技能组',  manual\_time STRING COMMENT '人工开始时间',  begin\_time STRING COMMENT '坐席领取时间 ',  reply\_msg\_count int COMMENT '客服回复消息数',  total\_msg\_count int COMMENT '消息总数',  msg\_count int COMMENT '客户发送消息数',  comment STRING COMMENT '备注',  finish\_reason STRING COMMENT '结束类型',  finish\_user STRING COMMENT '结束坐席',  end\_time STRING COMMENT '会话结束时间',  platform\_description STRING COMMENT '客户平台信息',  browser\_name STRING COMMENT '浏览器名称',  os\_info STRING COMMENT '系统名称',  area STRING COMMENT '区域',  country STRING COMMENT '所在国家',  province STRING COMMENT '省',  city STRING COMMENT '城市',  creator int COMMENT '创建人',  name STRING COMMENT '客户姓名',  idcard STRING COMMENT '身份证号',  phone STRING COMMENT '手机号',  itcast\_school\_id int COMMENT '校区Id',  itcast\_school STRING COMMENT '校区',  itcast\_subject\_id int COMMENT '学科Id',  itcast\_subject STRING COMMENT '学科',  wechat STRING COMMENT '微信',  qq STRING COMMENT 'qq号',  email STRING COMMENT '邮箱',  gender STRING COMMENT '性别',  level STRING COMMENT '客户级别',  origin\_type STRING COMMENT '数据来源渠道',  information\_way STRING COMMENT '资讯方式',  working\_years STRING COMMENT '开始工作时间',  technical\_directions STRING COMMENT '技术方向',  customer\_state STRING COMMENT '当前客户状态',  valid STRING COMMENT '该线索是否是网资有效线索',  anticipat\_signup\_date STRING COMMENT '预计报名时间',  clue\_state STRING COMMENT '线索状态',  scrm\_department\_id int COMMENT 'SCRM内部部门id',  superior\_url STRING COMMENT '诸葛获取上级页面URL',  superior\_source STRING COMMENT '诸葛获取上级页面URL标题',  landing\_url STRING COMMENT '诸葛获取着陆页面URL',  landing\_source STRING COMMENT '诸葛获取着陆页面URL来源',  info\_url STRING COMMENT '诸葛获取留咨页URL',  info\_source STRING COMMENT '诸葛获取留咨页URL标题',  origin\_channel STRING COMMENT '投放渠道',  course\_id int COMMENT '课程编号',  course\_name STRING COMMENT '课程名称',  zhuge\_session\_id STRING COMMENT 'zhuge会话id',  is\_repeat int COMMENT '是否重复线索(手机号维度) 0:正常 1：重复',  tenant int COMMENT '租户id',  activity\_id STRING COMMENT '活动id',  activity\_name STRING COMMENT '活动名称',  follow\_type int COMMENT '分配类型，0-自动分配，1-手动分配，2-自动转移，3-手动单个转移，4-手动批量转移，5-公海领取',  shunt\_mode\_id int COMMENT '匹配到的技能组id',  shunt\_employee\_group\_id int COMMENT '所属分流员工组',  ends\_time STRING COMMENT '有效时间') comment '客户关系表' PARTITIONED BY(starts\_time STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' stored as orc TBLPROPERTIES ('orc.compress'='ZLIB'); |

抽取昨日新增和更新数据（逻辑删除也属于更新操作）

因为增量抽取是T+1，所以Sql中需要增加where条件，只查询昨天一天的数据，而不是所有表数据。

查询条件需要包含创建日期和更新日期，因为需要将昨日新增和修改的数据都抽取到数仓中。

SQL:

|  |
| --- |
| select id,  create\_date\_time,  update\_date\_time,  deleted,  customer\_id,  customer\_relationship\_id,  session\_id,  sid,  status, user as users, create\_time, platform, s\_name, seo\_source, seo\_keywords, ip, referrer, from\_url, landing\_page\_url, url\_title, to\_peer, manual\_time, begin\_time, reply\_msg\_count, total\_msg\_count, msg\_count, comment, finish\_reason, finish\_user, end\_time, platform\_description, browser\_name, os\_info, area, country, province, city, creator, name, idcard, phone, itcast\_school\_id, itcast\_school, itcast\_subject\_id, itcast\_subject, wechat, qq, email, gender, level, origin\_type, information\_way, working\_years, technical\_directions, customer\_state, valid, anticipat\_signup\_date, clue\_state, scrm\_department\_id, superior\_url, superior\_source, landing\_url, landing\_source, info\_url, info\_source, origin\_channel, course\_id, course\_name, zhuge\_session\_id, is\_repeat, tenant, activity\_id, activity\_name, follow\_type, shunt\_mode\_id, shunt\_employee\_group\_id,  FROM\_UNIXTIME(unix\_timestamp(), "%Y-%m-%d") as starts\_time, date\_format("9999-12-31", "%Y-%m-%d") as ends\_time from customer\_clue where  (  create\_date\_time >= *FROM\_UNIXTIME*(*UNIX\_TIMESTAMP*("2019-12-04 00:00:00"),"%Y-%m-%d %H:%i:%s")  and  create\_date\_time < *FROM\_UNIXTIME*(*UNIX\_TIMESTAMP*("2019-12-04 23:59:59"),"%Y-%m-%d %H:%i:%s")  )  or  (  update\_date\_time >= *FROM\_UNIXTIME*(*UNIX\_TIMESTAMP*("2019-12-04 00:00:00"),"%Y-%m-%d %H:%i:%s")  and  update\_date\_time < *FROM\_UNIXTIME*(*UNIX\_TIMESTAMP*("2019-12-04 23:59:59"),"%Y-%m-%d %H:%i:%s")  ); |

Sqoop脚本：

|  |
| --- |
| sqoop import \ *--connect jdbc:mysql://172.17.0.202:3306/scrm \ --username root \ --password 123456 \ --query '* select id,  create\_date\_time,  update\_date\_time,  deleted,  customer\_id,  customer\_relationship\_id,  session\_id,  sid,  status, user as users, create\_time, platform, s\_name, seo\_source, seo\_keywords, ip, referrer, from\_url, landing\_page\_url, url\_title, to\_peer, manual\_time, begin\_time, reply\_msg\_count, total\_msg\_count, msg\_count, comment, finish\_reason, finish\_user, end\_time, platform\_description, browser\_name, os\_info, area, country, province, city, creator, name, idcard, phone, itcast\_school\_id, itcast\_school, itcast\_subject\_id, itcast\_subject, wechat, qq, email, gender, level, origin\_type, information\_way, working\_years, technical\_directions, customer\_state, valid, anticipat\_signup\_date, clue\_state, scrm\_department\_id, superior\_url, superior\_source, landing\_url, landing\_source, info\_url, info\_source, origin\_channel, course\_id, course\_name, zhuge\_session\_id, is\_repeat, tenant, activity\_id, activity\_name, follow\_type, shunt\_mode\_id, shunt\_employee\_group\_id,  FROM\_UNIXTIME(unix\_timestamp(), "%Y-%m-%d") as starts\_time, date\_format("9999-12-31", "%Y-%m-%d") as ends\_time from customer\_clue where  (  create\_date\_time >= *FROM\_UNIXTIME*(*UNIX\_TIMESTAMP*("2019-12-04 00:00:00"),"%Y-%m-%d %H:%i:%s")  and  create\_date\_time < *FROM\_UNIXTIME*(*UNIX\_TIMESTAMP*("2019-12-04 23:59:59"),"%Y-%m-%d %H:%i:%s")  )  or  (  update\_date\_time >= *FROM\_UNIXTIME*(*UNIX\_TIMESTAMP*("2019-12-04 00:00:00"),"%Y-%m-%d %H:%i:%s")  and  update\_date\_time < *FROM\_UNIXTIME*(*UNIX\_TIMESTAMP*("2019-12-04 23:59:59"),"%Y-%m-%d %H:%i:%s")  ) and $CONDITIONS' \ --hcatalog-database itcast\_ods \ --hcatalog-table customer\_clue\_update \ --hive-partition-key starts\_time \ --hive-partition-value 2019-12-04 \ -m 100 \ --split-by id |

重建customer\_clue\_tmp临时表

|  |
| --- |
| DROP TABLE itcast\_ods.customer\_clue\_tmp; CREATE TABLE IF NOT EXISTS itcast\_ods.customer\_clue\_tmp (  id int COMMENT 'customer\_clue\_id',  create\_date\_time STRING COMMENT '创建时间',  update\_date\_time STRING COMMENT '最后更新时间',  deleted STRING COMMENT '是否被删除（禁用）',  customer\_id int COMMENT '客户id',  customer\_relationship\_id int COMMENT '客户关系id',  session\_id STRING COMMENT '七陌会话id',  sid STRING COMMENT '访客id',  status STRING COMMENT '状态（undeal待领取 deal 已领取 finish 已关闭 changePeer 已流转）',  users STRING COMMENT '所属坐席',  create\_time STRING COMMENT '七陌创建时间',  platform STRING COMMENT '平台来源 （pc-网站咨询|wap-wap咨询|sdk-app咨询|weixin-微信咨询）',  s\_name STRING COMMENT '用户名称',  seo\_source STRING COMMENT '搜索来源',  seo\_keywords STRING COMMENT '关键字',  ip STRING COMMENT 'IP地址',  referrer STRING COMMENT '上级来源页面',  from\_url STRING COMMENT '会话来源页面',  landing\_page\_url STRING COMMENT '访客着陆页面',  url\_title STRING COMMENT '咨询页面title',  to\_peer STRING COMMENT '所属技能组',  manual\_time STRING COMMENT '人工开始时间',  begin\_time STRING COMMENT '坐席领取时间 ',  reply\_msg\_count int COMMENT '客服回复消息数',  total\_msg\_count int COMMENT '消息总数',  msg\_count int COMMENT '客户发送消息数',  comment STRING COMMENT '备注',  finish\_reason STRING COMMENT '结束类型',  finish\_user STRING COMMENT '结束坐席',  end\_time STRING COMMENT '会话结束时间',  platform\_description STRING COMMENT '客户平台信息',  browser\_name STRING COMMENT '浏览器名称',  os\_info STRING COMMENT '系统名称',  area STRING COMMENT '区域',  country STRING COMMENT '所在国家',  province STRING COMMENT '省',  city STRING COMMENT '城市',  creator int COMMENT '创建人',  name STRING COMMENT '客户姓名',  idcard STRING COMMENT '身份证号',  phone STRING COMMENT '手机号',  itcast\_school\_id int COMMENT '校区Id',  itcast\_school STRING COMMENT '校区',  itcast\_subject\_id int COMMENT '学科Id',  itcast\_subject STRING COMMENT '学科',  wechat STRING COMMENT '微信',  qq STRING COMMENT 'qq号',  email STRING COMMENT '邮箱',  gender STRING COMMENT '性别',  level STRING COMMENT '客户级别',  origin\_type STRING COMMENT '数据来源渠道',  information\_way STRING COMMENT '资讯方式',  working\_years STRING COMMENT '开始工作时间',  technical\_directions STRING COMMENT '技术方向',  customer\_state STRING COMMENT '当前客户状态',  valid STRING COMMENT '该线索是否是网资有效线索',  anticipat\_signup\_date STRING COMMENT '预计报名时间',  clue\_state STRING COMMENT '线索状态',  scrm\_department\_id int COMMENT 'SCRM内部部门id',  superior\_url STRING COMMENT '诸葛获取上级页面URL',  superior\_source STRING COMMENT '诸葛获取上级页面URL标题',  landing\_url STRING COMMENT '诸葛获取着陆页面URL',  landing\_source STRING COMMENT '诸葛获取着陆页面URL来源',  info\_url STRING COMMENT '诸葛获取留咨页URL',  info\_source STRING COMMENT '诸葛获取留咨页URL标题',  origin\_channel STRING COMMENT '投放渠道',  course\_id int COMMENT '课程编号',  course\_name STRING COMMENT '课程名称',  zhuge\_session\_id STRING COMMENT 'zhuge会话id',  is\_repeat int COMMENT '是否重复线索(手机号维度) 0:正常 1：重复',  tenant int COMMENT '租户id',  activity\_id STRING COMMENT '活动id',  activity\_name STRING COMMENT '活动名称',  follow\_type int COMMENT '分配类型，0-自动分配，1-手动分配，2-自动转移，3-手动单个转移，4-手动批量转移，5-公海领取',  shunt\_mode\_id int COMMENT '匹配到的技能组id',  shunt\_employee\_group\_id int COMMENT '所属分流员工组',  ends\_time STRING COMMENT '有效时间') comment '客户关系表' PARTITIONED BY(starts\_time STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' stored as orc TBLPROPERTIES ('orc.compress'='ZLIB'); |

合并增量数据与历史数据（仅更新30天之内的数据，根据需求）

1. 获取update表的更新数据，新数据end\_time为’9999-12-31’，start\_time为昨日日期；
2. 获取拉链表历史数据：
   1. 更新end\_time
      1. 将历史表customer\_relationship（主表）与新增/更新数据表customer\_relationship\_update通过id进行关联，如果update中有与历史表重复的id，证明有此条id数据已有新的变更；
      2. 没有更新的数据保留原始end\_time；
      3. 历史表已是失效的数据，保留原始有效结束日期end\_time；
      4. 有更新的数据，且旧数据目前正在生效，修改end\_time为前天（昨天之前）；
   2. 因为业务方将更新周期限制在30天内，所以只需查询更新30天内的数据即可；
3. 将 1.update 与 2.拉链表 合并，覆盖插入到临时表中。

实现：

|  |
| --- |
| insert overwrite table itcast\_ods.customer\_clue\_tmp partition (starts\_time) select *\** from  (  select   id,  create\_date\_time,  update\_date\_time,  deleted,  customer\_id,  customer\_relationship\_id,  session\_id,  sid,  status,  users,  create\_time,  platform,  s\_name,  seo\_source,  seo\_keywords,  ip,  referrer,  from\_url,  landing\_page\_url,  url\_title,  to\_peer,  manual\_time,  begin\_time,  reply\_msg\_count,  total\_msg\_count,  msg\_count,  comment,  finish\_reason,  finish\_user,  end\_time,  platform\_description,  browser\_name,  os\_info,  area,  country,  province,  city,  creator,  name,  idcard,  phone,  itcast\_school\_id,  itcast\_school,  itcast\_subject\_id,  itcast\_subject,  wechat,  qq,  email,  gender,  level,  origin\_type,  information\_way,  working\_years,  technical\_directions,  customer\_state,  valid,  anticipat\_signup\_date,  clue\_state,  scrm\_department\_id,  superior\_url,  superior\_source,  landing\_url,  landing\_source,  info\_url,  info\_source,  origin\_channel,  course\_id,  course\_name,  zhuge\_session\_id,  is\_repeat,  tenant,  activity\_id,  activity\_name,  follow\_type,  shunt\_mode\_id,  shunt\_employee\_group\_id,  '9999-12-31' ends\_time,  *FROM\_UNIXTIME*(*unix\_timestamp*(), "%Y-%m-%d") as starts\_time  from itcast\_ods.customer\_clue\_update where starts\_time=*FROM\_UNIXTIME*(*unix\_timestamp*(), "%Y-%m-%d")   union all   select  rs.id,  rs.create\_date\_time,  rs.update\_date\_time,  rs.deleted,  rs.customer\_id,  rs.customer\_relationship\_id,  rs.session\_id,  rs.sid,  rs.status,  rs.users,  rs.create\_time,  rs.platform,  rs.s\_name,  rs.seo\_source,  rs.seo\_keywords,  rs.ip,  rs.referrer,  rs.from\_url,  rs.landing\_page\_url,  rs.url\_title,  rs.to\_peer,  rs.manual\_time,  rs.begin\_time,  rs.reply\_msg\_count,  rs.total\_msg\_count,  rs.msg\_count,  rs.comment,  rs.finish\_reason,  rs.finish\_user,  rs.end\_time,  rs.platform\_description,  rs.browser\_name,  rs.os\_info,  rs.area,  rs.country,  rs.province,  rs.city,  rs.creator,  rs.name,  rs.idcard,  rs.phone,  rs.itcast\_school\_id,  rs.itcast\_school,  rs.itcast\_subject\_id,  rs.itcast\_subject,  rs.wechat,  rs.qq,  rs.email,  rs.gender,  rs.level,  rs.origin\_type,  rs.information\_way,  rs.working\_years,  rs.technical\_directions,  rs.customer\_state,  rs.valid,  rs.anticipat\_signup\_date,  rs.clue\_state,  rs.scrm\_department\_id,  rs.superior\_url,  rs.superior\_source,  rs.landing\_url,  rs.landing\_source,  rs.info\_url,  rs.info\_source,  rs.origin\_channel,  rs.course\_id,  rs.course\_name,  rs.zhuge\_session\_id,  rs.is\_repeat,  rs.tenant,  rs.activity\_id,  rs.activity\_name,  rs.follow\_type,  rs.shunt\_mode\_id,  rs.shunt\_employee\_group\_id,  *if*(up.id is null or rs.end\_time<'9999-12-31', rs.ends\_time, *date\_add*(up.starts\_time,-1)) ends\_time,  rs.starts\_time  from itcast\_ods.customer\_clue rs left join   (  select  *\** from itcast\_ods.customer\_clue\_update  where starts\_time=*FROM\_UNIXTIME*(*unix\_timestamp*(), "%Y-%m-%d")  ) up  on rs.id=up.id where rs.starts\_time >= *date\_add*(*FROM\_UNIXTIME*(*UNIX\_TIMESTAMP*()),-30) )his  order by his.id, starts\_time; |

临时表覆盖到拉链表

|  |
| --- |
| INSERT OVERWRITE TABLE itcast\_ods.customer\_clue partition (starts\_time)  SELECT \* from itcast\_ods.customer\_clue\_tmp; |

测试

1. 删除mysql和HDFS(外部表)中的测试数据，避免数据重复，便于验证测试结果
2. 向mysql中插入新数据
3. 验证sqoop中的sql是否能够在mysql正常查询出测试数据
4. 重建update更新表
5. 手动执行sqoop脚本抽取数据
6. 重建tmp临时表
7. 合并当天的新增和更新数据
8. 临时表覆盖到拉链表

Oozie脚本

|  |
| --- |
| #! /bin/bash HIVE\_HOME=/usr/bin/hive if [[ $1 == "" ]]; then  TD\_DATE=`date -d ''1 days ago'' "+%Y-%m-%d"` else  TD\_DATE=$1 fi output=$(${HIVE\_HOME} -S -e " SET hive.exec.dynamic.partition=true; SET hive.exec.dynamic.partition.mode=nonstrict; DROP TABLE itcast\_ods.customer\_clue\_update; CREATE TABLE IF NOT EXISTS itcast\_ods.customer\_clue\_update (  id int COMMENT 'customer\_clue\_id',  create\_date\_time STRING COMMENT '创建时间',  update\_date\_time STRING COMMENT '最后更新时间',  deleted STRING COMMENT '是否被删除（禁用）',  customer\_id int COMMENT '客户id',  customer\_relationship\_id int COMMENT '客户关系id',  session\_id STRING COMMENT '七陌会话id',  sid STRING COMMENT '访客id',  status STRING COMMENT '状态（undeal待领取 deal 已领取 finish 已关闭 changePeer 已流转）',  users STRING COMMENT '所属坐席',  create\_time STRING COMMENT '七陌创建时间',  platform STRING COMMENT '平台来源 （pc-网站咨询|wap-wap咨询|sdk-app咨询|weixin-微信咨询）',  s\_name STRING COMMENT '用户名称',  seo\_source STRING COMMENT '搜索来源',  seo\_keywords STRING COMMENT '关键字',  ip STRING COMMENT 'IP地址',  referrer STRING COMMENT '上级来源页面',  from\_url STRING COMMENT '会话来源页面',  landing\_page\_url STRING COMMENT '访客着陆页面',  url\_title STRING COMMENT '咨询页面title',  to\_peer STRING COMMENT '所属技能组',  manual\_time STRING COMMENT '人工开始时间',  begin\_time STRING COMMENT '坐席领取时间 ',  reply\_msg\_count int COMMENT '客服回复消息数',  total\_msg\_count int COMMENT '消息总数',  msg\_count int COMMENT '客户发送消息数',  comment STRING COMMENT '备注',  finish\_reason STRING COMMENT '结束类型',  finish\_user STRING COMMENT '结束坐席',  end\_time STRING COMMENT '会话结束时间',  platform\_description STRING COMMENT '客户平台信息',  browser\_name STRING COMMENT '浏览器名称',  os\_info STRING COMMENT '系统名称',  area STRING COMMENT '区域',  country STRING COMMENT '所在国家',  province STRING COMMENT '省',  city STRING COMMENT '城市',  creator int COMMENT '创建人',  name STRING COMMENT '客户姓名',  idcard STRING COMMENT '身份证号',  phone STRING COMMENT '手机号',  itcast\_school\_id int COMMENT '校区Id',  itcast\_school STRING COMMENT '校区',  itcast\_subject\_id int COMMENT '学科Id',  itcast\_subject STRING COMMENT '学科',  wechat STRING COMMENT '微信',  qq STRING COMMENT 'qq号',  email STRING COMMENT '邮箱',  gender STRING COMMENT '性别',  level STRING COMMENT '客户级别',  origin\_type STRING COMMENT '数据来源渠道',  information\_way STRING COMMENT '资讯方式',  working\_years STRING COMMENT '开始工作时间',  technical\_directions STRING COMMENT '技术方向',  customer\_state STRING COMMENT '当前客户状态',  valid STRING COMMENT '该线索是否是网资有效线索',  anticipat\_signup\_date STRING COMMENT '预计报名时间',  clue\_state STRING COMMENT '线索状态',  scrm\_department\_id int COMMENT 'SCRM内部部门id',  superior\_url STRING COMMENT '诸葛获取上级页面URL',  superior\_source STRING COMMENT '诸葛获取上级页面URL标题',  landing\_url STRING COMMENT '诸葛获取着陆页面URL',  landing\_source STRING COMMENT '诸葛获取着陆页面URL来源',  info\_url STRING COMMENT '诸葛获取留咨页URL',  info\_source STRING COMMENT '诸葛获取留咨页URL标题',  origin\_channel STRING COMMENT '投放渠道',  course\_id int COMMENT '课程编号',  course\_name STRING COMMENT '课程名称',  zhuge\_session\_id STRING COMMENT 'zhuge会话id',  is\_repeat int COMMENT '是否重复线索(手机号维度) 0:正常 1：重复',  tenant int COMMENT '租户id',  activity\_id STRING COMMENT '活动id',  activity\_name STRING COMMENT '活动名称',  follow\_type int COMMENT '分配类型，0-自动分配，1-手动分配，2-自动转移，3-手动单个转移，4-手动批量转移，5-公海领取',  shunt\_mode\_id int COMMENT '匹配到的技能组id',  shunt\_employee\_group\_id int COMMENT '所属分流员工组',  ends\_time STRING COMMENT '有效时间') comment '客户关系表' PARTITIONED BY(starts\_time STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' stored as orc TBLPROPERTIES ('orc.compress'='ZLIB'); ")  SQOOP\_HOME=/usr/bin/sqoop output=$(${SQOOP\_HOME} import \ --connect jdbc:mysql://172.17.0.202:3306/scrm \ --username root \ --password 123456 \ --query ' select id,  create\_date\_time,  update\_date\_time,  deleted,  customer\_id,  customer\_relationship\_id,  session\_id,  sid,  status, user, create\_time, platform, s\_name, seo\_source, seo\_keywords, ip, referrer, from\_url, landing\_page\_url, url\_title, to\_peer, manual\_time, begin\_time, reply\_msg\_count, total\_msg\_count, msg\_count, comment, finish\_reason, finish\_user, end\_time, platform\_description, browser\_name, os\_info, area, country, province, city, creator, name, idcard, phone, itcast\_school\_id, itcast\_school, itcast\_subject\_id, itcast\_subject, wechat, qq, email, gender, level, origin\_type, information\_way, working\_years, technical\_directions, customer\_state, valid, anticipat\_signup\_date, clue\_state, scrm\_department\_id, superior\_url, superior\_source, landing\_url, landing\_source, info\_url, info\_source, origin\_channel, course\_id, course\_name, zhuge\_session\_id, is\_repeat, tenant, activity\_id, activity\_name, follow\_type, shunt\_mode\_id, shunt\_employee\_group\_id,  FROM\_UNIXTIME(unix\_timestamp(), "%Y-%m-%d") as starts\_time, date\_format("9999-12-31", "%Y-%m-%d") as ends\_time from customer\_clue where  (  create\_date\_time >= *FROM\_UNIXTIME*(*UNIX\_TIMESTAMP*(*CAST*(SYSDATE()AS DATE) - INTERVAL 1 DAY),"%Y-%m-%d %H:%i:%s")  and  create\_date\_time < *FROM\_UNIXTIME*(*UNIX\_TIMESTAMP*(*CAST*(SYSDATE()AS DATE)),"%Y-%m-%d %H:%i:%s")  )  or  (  update\_date\_time >= *FROM\_UNIXTIME*(*UNIX\_TIMESTAMP*(*CAST*(SYSDATE()AS DATE) - INTERVAL 1 DAY),"%Y-%m-%d %H:%i:%s")  and  update\_date\_time < *FROM\_UNIXTIME*(*UNIX\_TIMESTAMP*(*CAST*(SYSDATE()AS DATE)),"%Y-%m-%d %H:%i:%s")  ) and $CONDITIONS' \ --hcatalog-database itcast\_ods \ --hcatalog-table customer\_clue\_update \ --hive-partition-key starts\_time \ --hive-partition-value ${TD\_DATE} \ -m 100 \ --split-by id)  output=$(${HIVE\_HOME} -S -e " SET hive.exec.dynamic.partition=true; SET hive.exec.dynamic.partition.mode=nonstrict;  DROP TABLE itcast\_ods.customer\_clue\_tmp; CREATE TABLE IF NOT EXISTS itcast\_ods.customer\_clue\_tmp (  id int COMMENT 'customer\_clue\_id',  create\_date\_time STRING COMMENT '创建时间',  update\_date\_time STRING COMMENT '最后更新时间',  deleted STRING COMMENT '是否被删除（禁用）',  customer\_id int COMMENT '客户id',  customer\_relationship\_id int COMMENT '客户关系id',  session\_id STRING COMMENT '七陌会话id',  sid STRING COMMENT '访客id',  status STRING COMMENT '状态（undeal待领取 deal 已领取 finish 已关闭 changePeer 已流转）',  users STRING COMMENT '所属坐席',  create\_time STRING COMMENT '七陌创建时间',  platform STRING COMMENT '平台来源 （pc-网站咨询|wap-wap咨询|sdk-app咨询|weixin-微信咨询）',  s\_name STRING COMMENT '用户名称',  seo\_source STRING COMMENT '搜索来源',  seo\_keywords STRING COMMENT '关键字',  ip STRING COMMENT 'IP地址',  referrer STRING COMMENT '上级来源页面',  from\_url STRING COMMENT '会话来源页面',  landing\_page\_url STRING COMMENT '访客着陆页面',  url\_title STRING COMMENT '咨询页面title',  to\_peer STRING COMMENT '所属技能组',  manual\_time STRING COMMENT '人工开始时间',  begin\_time STRING COMMENT '坐席领取时间 ',  reply\_msg\_count int COMMENT '客服回复消息数',  total\_msg\_count int COMMENT '消息总数',  msg\_count int COMMENT '客户发送消息数',  comment STRING COMMENT '备注',  finish\_reason STRING COMMENT '结束类型',  finish\_user STRING COMMENT '结束坐席',  end\_time STRING COMMENT '会话结束时间',  platform\_description STRING COMMENT '客户平台信息',  browser\_name STRING COMMENT '浏览器名称',  os\_info STRING COMMENT '系统名称',  area STRING COMMENT '区域',  country STRING COMMENT '所在国家',  province STRING COMMENT '省',  city STRING COMMENT '城市',  creator int COMMENT '创建人',  name STRING COMMENT '客户姓名',  idcard STRING COMMENT '身份证号',  phone STRING COMMENT '手机号',  itcast\_school\_id int COMMENT '校区Id',  itcast\_school STRING COMMENT '校区',  itcast\_subject\_id int COMMENT '学科Id',  itcast\_subject STRING COMMENT '学科',  wechat STRING COMMENT '微信',  qq STRING COMMENT 'qq号',  email STRING COMMENT '邮箱',  gender STRING COMMENT '性别',  level STRING COMMENT '客户级别',  origin\_type STRING COMMENT '数据来源渠道',  information\_way STRING COMMENT '资讯方式',  working\_years STRING COMMENT '开始工作时间',  technical\_directions STRING COMMENT '技术方向',  customer\_state STRING COMMENT '当前客户状态',  valid STRING COMMENT '该线索是否是网资有效线索',  anticipat\_signup\_date STRING COMMENT '预计报名时间',  clue\_state STRING COMMENT '线索状态',  scrm\_department\_id int COMMENT 'SCRM内部部门id',  superior\_url STRING COMMENT '诸葛获取上级页面URL',  superior\_source STRING COMMENT '诸葛获取上级页面URL标题',  landing\_url STRING COMMENT '诸葛获取着陆页面URL',  landing\_source STRING COMMENT '诸葛获取着陆页面URL来源',  info\_url STRING COMMENT '诸葛获取留咨页URL',  info\_source STRING COMMENT '诸葛获取留咨页URL标题',  origin\_channel STRING COMMENT '投放渠道',  course\_id int COMMENT '课程编号',  course\_name STRING COMMENT '课程名称',  zhuge\_session\_id STRING COMMENT 'zhuge会话id',  is\_repeat int COMMENT '是否重复线索(手机号维度) 0:正常 1：重复',  tenant int COMMENT '租户id',  activity\_id STRING COMMENT '活动id',  activity\_name STRING COMMENT '活动名称',  follow\_type int COMMENT '分配类型，0-自动分配，1-手动分配，2-自动转移，3-手动单个转移，4-手动批量转移，5-公海领取',  shunt\_mode\_id int COMMENT '匹配到的技能组id',  shunt\_employee\_group\_id int COMMENT '所属分流员工组',  ends\_time STRING COMMENT '有效时间') comment '客户关系表' PARTITIONED BY(starts\_time STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' stored as orc TBLPROPERTIES ('orc.compress'='ZLIB');  insert overwrite table itcast\_ods.customer\_clue\_tmp partition (starts\_time) select \* from  (  select   id,  create\_date\_time,  update\_date\_time,  deleted,  customer\_id,  customer\_relationship\_id,  session\_id,  sid,  status,  user,  create\_time,  platform,  s\_name,  seo\_source,  seo\_keywords,  ip,  referrer,  from\_url,  landing\_page\_url,  url\_title,  to\_peer,  manual\_time,  begin\_time,  reply\_msg\_count,  total\_msg\_count,  msg\_count,  comment,  finish\_reason,  finish\_user,  ends\_time,  platform\_description,  browser\_name,  os\_info,  area,  country,  province,  city,  creator,  name,  idcard,  phone,  itcast\_school\_id,  itcast\_school,  itcast\_subject\_id,  itcast\_subject,  wechat,  qq,  email,  gender,  level,  origin\_type,  information\_way,  working\_years,  technical\_directions,  customer\_state,  valid,  anticipat\_signup\_date,  clue\_state,  scrm\_department\_id,  superior\_url,  superior\_source,  landing\_url,  landing\_source,  info\_url,  info\_source,  origin\_channel,  course\_id,  course\_name,  zhuge\_session\_id,  is\_repeat,  tenant,  activity\_id,  activity\_name,  follow\_type,  shunt\_mode\_id,  shunt\_employee\_group\_id,  '9999-12-31' ends\_time,  FROM\_UNIXTIME(unix\_timestamp(), "%Y-%m-%d") as starts\_time  from itcast\_ods.customer\_clue\_update where starts\_time=FROM\_UNIXTIME(unix\_timestamp(), "%Y-%m-%d")   union all   select  rs.id,  rs.create\_date\_time,  rs.update\_date\_time,  rs.deleted,  rs.customer\_id,  rs.customer\_relationship\_id,  rs.session\_id,  rs.sid,  rs.status,  rs.user,  rs.create\_time,  rs.platform,  rs.s\_name,  rs.seo\_source,  rs.seo\_keywords,  rs.ip,  rs.referrer,  rs.from\_url,  rs.landing\_page\_url,  rs.url\_title,  rs.to\_peer,  rs.manual\_time,  rs.begin\_time,  rs.reply\_msg\_count,  rs.total\_msg\_count,  rs.msg\_count,  rs.comment,  rs.finish\_reason,  rs.finish\_user,  rs.ends\_time,  rs.platform\_description,  rs.browser\_name,  rs.os\_info,  rs.area,  rs.country,  rs.province,  rs.city,  rs.creator,  rs.name,  rs.idcard,  rs.phone,  rs.itcast\_school\_id,  rs.itcast\_school,  rs.itcast\_subject\_id,  rs.itcast\_subject,  rs.wechat,  rs.qq,  rs.email,  rs.gender,  rs.level,  rs.origin\_type,  rs.information\_way,  rs.working\_years,  rs.technical\_directions,  rs.customer\_state,  rs.valid,  rs.anticipat\_signup\_date,  rs.clue\_state,  rs.scrm\_department\_id,  rs.superior\_url,  rs.superior\_source,  rs.landing\_url,  rs.landing\_source,  rs.info\_url,  rs.info\_source,  rs.origin\_channel,  rs.course\_id,  rs.course\_name,  rs.zhuge\_session\_id,  rs.is\_repeat,  rs.tenant,  rs.activity\_id,  rs.activity\_name,  rs.follow\_type,  rs.shunt\_mode\_id,  rs.shunt\_employee\_group\_id,  if(up.id is null, rs.ends\_time, date\_add(up.starts\_time,-1)) ends\_time,  rs.starts\_time  from itcast\_ods.customer\_clue rs left join   (  select  \*  from itcast\_ods.customer\_clue\_update  where starts\_time=FROM\_UNIXTIME(unix\_timestamp(), "%Y-%m-%d")  ) up  on rs.id=up.id where rs.starts\_time >= date\_sub(FROM\_UNIXTIME(UNIX\_TIMESTAMP()),30) and rs.ends\_time='9999-12-31' )his  order by his.id, starts\_time;  INSERT OVERWRITE TABLE itcast\_ods.customer\_clue partition (starts\_time)  SELECT \* from itcast\_ods.customer\_clue\_tmp; ") |

#### 数据清洗转换

##### DWD

###### 分析

因为业务方将更新周期限制在30天内，而明细层不涉及统计，只有数据清洗转换操作，所以我们在进行增量统计时，只需要重新计算上个月1日至今的数据即可。

通过start\_time来指定清洗的数据时间范围（昨天：新增/更新）；

通过end\_time来指定获取当前有效的数据。

清洗掉已删除的数据；

判断学校id和学科id，把为空的字段统一转换为-1；

将origin\_type来源渠道字段转换为线上/线下，如果origin\_type是NETSERVICE和PRESIGNUP类型，即为1线上，否则为0线下。

###### 代码

SQL：

|  |
| --- |
| *--分区* SET hive.exec.dynamic.partition=true; SET hive.exec.dynamic.partition.mode=nonstrict; set hive.exec.max.dynamic.partitions.pernode=10000; set hive.exec.max.dynamic.partitions=100000; set hive.exec.max.created.files=150000; *--hive压缩* set hive.exec.compress.intermediate=true; set hive.exec.compress.output=true; *--写入时压缩生效* set hive.exec.orc.compression.strategy=COMPRESSION; *--分桶* set hive.enforce.bucketing=true; set hive.enforce.sorting=true; set hive.optimize.bucketmapjoin = true; set hive.auto.convert.sortmerge.join=true; set hive.auto.convert.sortmerge.join.noconditionaltask=true;  insert into table itcast\_dwd.itcast\_intention\_dwd partition (yearinfo,monthinfo,dayinfo) select  rs.id as rid,  rs.customer\_id,  rs.create\_date\_time,  *if*((rs.itcast\_school\_id is null) or (rs.itcast\_school\_id = 0), -1, rs.itcast\_school\_id) as itcast\_school\_id,  rs.deleted,  rs.origin\_type,  *if*((rs.itcast\_subject\_id is null) or (rs.itcast\_subject\_id = 0), -1, rs.itcast\_subject\_id) as itcast\_subject\_id,  *substr*(rs.create\_date\_time, 12, 2) hourinfo,  *if*(rs.origin\_type='NETSERVICE', '1', *if*(rs.origin\_type='PRESIGNUP', '1', '0')) as origin\_type\_stat,  *substr*(rs.create\_date\_time, 1, 4) yearinfo,  *substr*(rs.create\_date\_time, 6, 2) monthinfo,  *substr*(rs.create\_date\_time, 9, 2) dayinfo from itcast\_ods.customer\_relationship rs where rs.deleted = 0 and ***start\_time* = '${Last\_DATE}'***--2019-11-01* and **rs.end\_time = '9999-12-31'**; |

Shell脚本：

通过shell脚本获取上个月1日的日期，替换sql中的查询条件。

|  |
| --- |
| #! /bin/bash SQOOP\_HOME=/usr/bin/sqoop  #**昨天** Last\_DATE=$(date -d "-1 day" +%Y-%m-%d)   ${HIVE\_HOME} -S -e " *--分区* SET hive.exec.dynamic.partition=true; SET hive.exec.dynamic.partition.mode=nonstrict; set hive.exec.max.dynamic.partitions.pernode=10000; set hive.exec.max.dynamic.partitions=100000; set hive.exec.max.created.files=150000; *--hive压缩* set hive.exec.compress.intermediate=true; set hive.exec.compress.output=true; *--写入时压缩生效* set hive.exec.orc.compression.strategy=COMPRESSION; *--分桶* set hive.enforce.bucketing=true; set hive.enforce.sorting=true; set hive.optimize.bucketmapjoin = true; set hive.auto.convert.sortmerge.join=true; set hive.auto.convert.sortmerge.join.noconditionaltask=true;  insert into table itcast\_dwd.itcast\_intention\_dwd partition (yearinfo,monthinfo,dayinfo) select  rs.id as rid,  rs.customer\_id,  rs.create\_date\_time,  *if*((rs.itcast\_school\_id is null) or (rs.itcast\_school\_id = 0), -1, rs.itcast\_school\_id) as itcast\_school\_id,  rs.deleted,  rs.origin\_type,  *if*((rs.itcast\_subject\_id is null) or (rs.itcast\_subject\_id = 0), -1, rs.itcast\_subject\_id) as itcast\_subject\_id,  *substr*(rs.create\_date\_time, 12, 2) hourinfo,  *if*(rs.origin\_type='NETSERVICE', '1', *if*(rs.origin\_type='PRESIGNUP', '1', '0')) as origin\_type\_stat,  *substr*(rs.create\_date\_time, 1, 4) yearinfo,  *substr*(rs.create\_date\_time, 6, 2) monthinfo,  *substr*(rs.create\_date\_time, 9, 2) dayinfo from itcast\_ods.customer\_relationship rs where rs.deleted = 0 and *substr*(rs.start\_time, 1, 10) = '${Last\_DATE}'*--2019-11-01* and rs.end\_time = '9999-12-31'; " |

##### DWM

通过年月日限定，只关联上个月1日至今的数据。

###### SQL:

|  |
| --- |
| insert overwrite table itcast\_dwm.itcast\_intention\_dwm partition (yearinfo,monthinfo,dayinfo) select  dwd.customer\_id,  dwd.create\_date\_time,  cus.area,  dwd.itcast\_school\_id,  sch.name as itcast\_school\_name,  dwd.deleted,  dwd.origin\_type,  dwd.itcast\_subject\_id,  sub.name as itcast\_subject\_name,  dwd.hourinfo,  dwd.origin\_type\_stat,  *if*(clue.clue\_state='VALID\_NEW\_CLUES', '1', *if*(clue.clue\_state='VALID\_PUBLIC\_NEW\_CLUE', '0', '-1')) as clue\_state\_stat,  e.department\_id as tdepart\_id,  dept.name as tdepart\_name,  dwd.yearinfo,  dwd.monthinfo,  dwd.dayinfo from itcast\_dwd.itcast\_intention\_dwd dwd left join itcast\_ods.customer\_clue clue on clue.customer\_relationship\_id=dwd.rid left join itcast\_dimen.customer cus on dwd.customer\_id = cus.id left join itcast\_dimen.employee e on dwd.creator = e.id left join itcast\_dimen.scrm\_department dept on e.department\_id = dept.id left join itcast\_dimen.itcast\_subject sub on dwd.itcast\_subject\_id = sub.id left join itcast\_dimen.itcast\_school sch on dwd.itcast\_school\_id = sch.id where *concat\_ws*('-',dwd.yearinfo,dwd.monthinfo,dwd.dayinfo) >= '${Last\_Month\_DATE}'*--2019-11-01;* |

###### Shell:

|  |
| --- |
| #! /bin/bash SQOOP\_HOME=/usr/bin/sqoop  #**上个月1日** Last\_Month\_DATE=$(date -d "-1 month" +%Y-%m-01)   ${HIVE\_HOME} -S -e " *--分区* SET hive.exec.dynamic.partition=true; SET hive.exec.dynamic.partition.mode=nonstrict; set hive.exec.max.dynamic.partitions.pernode=10000; set hive.exec.max.dynamic.partitions=100000; set hive.exec.max.created.files=150000; *--hive压缩* set hive.exec.compress.intermediate=true; set hive.exec.compress.output=true; *--写入时压缩生效* set hive.exec.orc.compression.strategy=COMPRESSION; *--分桶* set hive.enforce.bucketing=true; set hive.enforce.sorting=true; set hive.optimize.bucketmapjoin = true; set hive.auto.convert.sortmerge.join=true; set hive.auto.convert.sortmerge.join.noconditionaltask=true;  insert into table itcast\_dwm.itcast\_intention\_dwm partition (yearinfo,monthinfo,dayinfo) select  dwd.customer\_id,  dwd.create\_date\_time,  cus.area,  dwd.itcast\_school\_id,  sch.name as itcast\_school\_name,  dwd.deleted,  dwd.origin\_type,  dwd.itcast\_subject\_id,  sub.name as itcast\_subject\_name,  dwd.hourinfo,  dwd.origin\_type\_stat,  *if*(clue.clue\_state='VALID\_NEW\_CLUES', '1', *if*(clue.clue\_state='VALID\_PUBLIC\_NEW\_CLUE', '0', '-1')) as clue\_state\_stat,  e.department\_id as tdepart\_id,  dept.name as tdepart\_name,  dwd.yearinfo,  dwd.monthinfo,  dwd.dayinfo from itcast\_dwd.itcast\_intention\_dwd dwd left join itcast\_ods.customer\_clue clue on clue.customer\_relationship\_id=dwd.rid left join itcast\_dimen.customer cus on dwd.customer\_id = cus.id left join itcast\_dimen.employee e on dwd.creator = e.id left join itcast\_dimen.scrm\_department dept on e.department\_id = dept.id left join itcast\_dimen.itcast\_subject sub on dwd.itcast\_subject\_id = sub.id left join itcast\_dimen.itcast\_school sch on dwd.itcast\_school\_id = sch.id where *concat\_ws*('-',dwd.yearinfo,dwd.monthinfo,dwd.dayinfo) >= '${Last\_Month\_DATE}'*--2019-11-01;* " |

#### 统计分析

##### 新增总意向量

可以查询2016-10-12之前的数据进行测试。

小时和天数据，重新计算上个月1日之后的数据；月份维度，计算上个月之后的数据；年份维度，计算上个月1日所在的年份之后的数据。

|  |
| --- |
| *--总意向量分组（按照时间和常驻类型统计） --小时* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  '-1' itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo,'-',dayinfo,' ',hourinfo) as time\_str,  '1' as grouptype,  '1' as time\_type,  yearinfo,  monthinfo,  dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm where *concat\_ws*('-',dwm.yearinfo,dwm.monthinfo,dwm.dayinfo) >= '${Last\_Month\_DATE}'*--2011-08-01* group by yearinfo, monthinfo, dayinfo, hourinfo, origin\_type\_stat, clue\_state\_stat; *--天* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  '-1' itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo,'-',dayinfo) as time\_str,  '1' as grouptype,  '2' as time\_type,  yearinfo,  monthinfo,  dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm where *concat\_ws*('-',dwm.yearinfo,dwm.monthinfo,dwm.dayinfo) >= '${Last\_Month\_DATE}'*--2011-08-19* group by yearinfo, monthinfo, dayinfo, origin\_type\_stat, clue\_state\_stat; *--月* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  '-1' itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo) as time\_str,  '1' as grouptype,  '1' as time\_type,  yearinfo,  monthinfo,  '-1' as dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm where ***concat\_ws*('-',dwm.yearinfo,dwm.monthinfo) >= '${V\_Month}'***--2011-08* group by yearinfo, monthinfo, origin\_type\_stat, clue\_state\_stat; *--年* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  '-1' as area,  '-1' itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo) as time\_str,  '1' as grouptype,  '1' as time\_type,  yearinfo,  '-1' as monthinfo,  '-1' as dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm where **dwm.yearinfo >= '${V\_Year}'***--2011* group by yearinfo, origin\_type\_stat, clue\_state\_stat; |

##### 意向学员位置热力图

|  |
| --- |
| *--地区分组（按照地区、时间和常驻类型统计） --小时* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  area,  '-1' itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo,'-',dayinfo,' ',hourinfo) as time\_str,  '2' as grouptype,  '1' as time\_type,  yearinfo,  monthinfo,  dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm where *concat\_ws*('-',dwm.yearinfo,dwm.monthinfo,dwm.dayinfo) >= '${Last\_Month\_DATE}'*--2011-08-19* group by area, yearinfo, monthinfo, dayinfo, hourinfo, origin\_type\_stat, clue\_state\_stat; *--天* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  area,  '-1' itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo,'-',dayinfo) as time\_str,  '2' as grouptype,  '2' as time\_type,  yearinfo,  monthinfo,  dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm where *concat\_ws*('-',dwm.yearinfo,dwm.monthinfo,dwm.dayinfo) >= '${Last\_Month\_DATE}'*--2011-08-19* group by area, yearinfo, monthinfo, dayinfo, origin\_type\_stat, clue\_state\_stat; *--月* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  area,  '-1' itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo,'-',monthinfo) as time\_str,  '1' as grouptype,  '1' as time\_type,  yearinfo,  monthinfo,  '-1' as dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm where *concat\_ws*('-',dwm.yearinfo,dwm.monthinfo) >= '${V\_Month}'*--2011-08* group by area, yearinfo, monthinfo, origin\_type\_stat, clue\_state\_stat; *--年* insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo) select  *count*(distinct customer\_id) as customer\_total,  area,  '-1' itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  '-1' as hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  *concat*(yearinfo) as time\_str,  '2' as grouptype,  '1' as time\_type,  yearinfo,  '-1' as monthinfo,  '-1' as dayinfo from itcast\_dwm.itcast\_intention\_dwm dwm where dwm.yearinfo >= '${V\_Year}'*--2011* group by area, yearinfo, origin\_type\_stat, clue\_state\_stat; |

##### 意向学科、校区排名

略。

##### 来源渠道占比

略。

##### 咨询中心占比

略。

##### OOzie Shell示例

需要在上个月1日的基础上，获取到对应的年份、月份字符，以替换sql中的变量。

意向中心贡献占比小时数据：

|  |
| --- |
| #! /bin/bash  #上个月1日  Last\_Month\_DATE=$(date -d "$(date +%Y%m)01 last month" +%Y-%m-01)  #根据TD\_DATE计算年季度月日  V\_PARYEAR=`date --date="$Last\_Month\_DATE" +%Y`  V\_PARMONTH=`date --date="$Last\_Month\_DATE" +%m`  V\_PARDAY=`date --date="$Last\_Month\_DATE" +%d`  #获取季度，**-m**为不带0，比如7，而不是07  V\_month\_for\_quarter=`date --date="$Last\_Month\_DATE" +%**-m**`  V\_PARQUARTER=$(((${V\_month\_for\_quarter}-1)/3+1))  ${HIVE\_HOME} -S -e "  SET hive.exec.dynamic.partition=true;  SET hive.exec.dynamic.partition.mode=nonstrict;  set hive.exec.max.dynamic.partitions.pernode=10000;  set hive.exec.max.dynamic.partitions=100000;  set hive.exec.max.created.files=150000;  set hive.enforce.bucketing=true;  set hive.enforce.sorting=true;  set hive.optimize.bucketmapjoin = true;  set hive.auto.convert.sortmerge.join=true;  set hive.auto.convert.sortmerge.join.noconditionaltask=true;  insert into itcast\_dws.itcast\_intention\_dws partition (yearinfo, monthinfo, dayinfo)  select  count(distinct customer\_id) as customer\_total,  '-1' as area,  '-1' itcast\_school\_id,  '-1' as itcast\_school\_name,  '-1' as origin\_type,  '-1' as itcast\_subject\_id,  '-1' as itcast\_subject\_name,  hourinfo,  origin\_type\_stat,  clue\_state\_stat,  '-1' as tdepart\_id,  '-1' as tdepart\_name,  concat(yearinfo,'-',monthinfo,'-',dayinfo,' ',hourinfo) as time\_str,  '1' as grouptype,  '1' as time\_type,  yearinfo,  monthinfo,  dayinfo  from itcast\_dwm.itcast\_intention\_dwm dwm  where concat\_ws('-',dwm.yearinfo,dwm.monthinfo,dwm.dayinfo) >= '${Last\_Month\_DATE}'  group by yearinfo, monthinfo, dayinfo, hourinfo, origin\_type\_stat, clue\_state\_stat;  " |

#### 导出数据

按照年份，先删除所在年的数据，后导出。

|  |
| --- |
| #! /bin/bash  SQOOP\_HOME=/usr/bin/sqoop  HOST=172.17.0.202  USERNAME="root"  PASSWORD="123456"  PORT=3306  DBNAME="scrm\_bi"  MYSQL=/usr/local/mysql\_5723/bin/mysql  #上个月1日  if [[ $1 == "" ]];then  Last\_Month\_DATE=$(date -d "**-1 month**" +%Y-%m-01) else  Last\_Month\_DATE=$1 fi  TD\_YEAR=$(date -d "$Last\_Month\_DATE" +%Y)  ${MYSQL} -h${HOST} -P${PORT} -u${USERNAME} -p${PASSWORD} -D${DBNAME} -e "delete from itcast\_intention\_app where yearinfo = '${Last\_Month\_DATE:0:4}'"  ${SQOOP\_HOME} export \  --connect "jdbc:mysql://${HOST}:${PORT}/${DBNAME}?useUnicode=true&characterEncoding=utf-8" \  --username ${USERNAME} \  --password ${PASSWORD} \  --table itcast\_intention\_app \  --hcatalog-database itcast\_dws \  --hcatalog-table itcast\_intention\_dws \  --hcatalog-partition-keys yearinfo \  --hcatalog-partition-values ${TD\_YEAR} \  -m 100 |