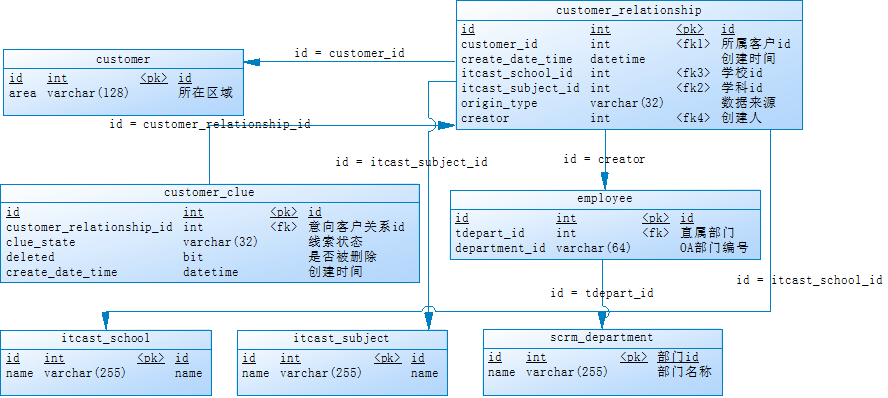
# 意向客户主题看板

## 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

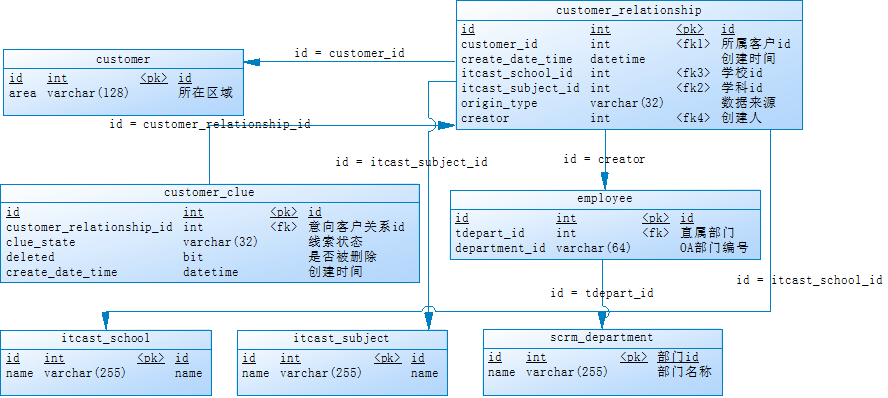
粒度：天，可以下钻到小时数据。

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; |

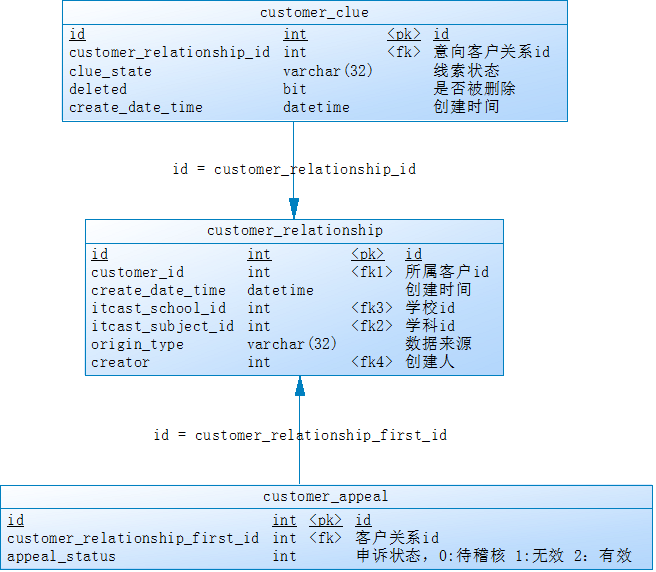
### itcast\_subject学科表

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; |

# 有效线索主题看板

## 主题需求



### 有效线索转化率

说明：统计期内，访客咨询产生的有效线索的占比。有效线索量 / 咨询量，有效线索指的是拿到电话且电话有效。

展现：线状图。双轴：有效线索量、有效线索转化率。

条件：年、月、线上线下

维度：年、月、线上线下

指标：访客咨询率=有效线索量/咨询量

粒度：天

数据来源：客户管理系统的customer\_clue线索表、customer\_relationship意向表、customer\_appeal申诉表；咨询系统的web\_chat\_ems访问咨询表

SQL：

1. --咨询量(暂时以2019年7月的数据为例)：
2. **SELECT**
3. count(1)
4. **FROM**
5. web\_chat\_ems\_2019\_07
6. **WHERE**
7. msg\_count >= 1
8. AND create\_time >= '2019-07-01'
9. AND create\_time <= '2019-07-15 23:59:59';
10. --有效线索量：
11. **SELECT**
12. count(1)
13. **FROM**
14. *customer\_clue* cc
15. LEFT JOIN *customer\_relationship* cr **ON** cc.customer\_relationship\_id = cr.id
16. **WHERE**
17. cc.clue\_state IN (
18. 'VALID\_NEW\_CLUES', --新客户新线索
19. 'VALID\_PUBLIC\_NEW\_CLUE' --老客户新线索
20. )
21. AND cc.customer\_relationship\_id NOT IN (
22. **SELECT**
23. ca.customer\_relationship\_first\_id
24. **FROM** --投诉表，投诉成功的数据为无效线索
25. customer\_appeal ca
26. **WHERE**
27. ca.appeal\_status = 1 AND ca.customer\_relationship\_first\_id != 0
28. )
29. AND cr.origin\_type IN ('NETSERVICE','PRESIGNUP') --线上（排除挖掘录入量）
30. AND ! cc.deleted
31. AND cc.create\_date\_time <= '2019-07-01'
32. AND cc.create\_date\_time <= '2019-07-15 23:59:59';

### 有效线索转化率时间段趋势

说明：统计期内，1-24h之间，每个时间段的有效线索转化率。横轴：1-24h，间隔为1h，纵轴：每个时间段的有效线索转化率。

展现：线状图

条件：天、线上线下

维度：天、线上线下

指标：某小时的总有效线索转化率

粒度：区间内小时段（区间内同一个时间点的总有效线索转化率）

数据来源：客户管理系统的customer\_clue线索表、customer\_relationship意向表、customer\_appeal申诉表；咨询系统的web\_chat\_ems访问咨询表

SQL：同上

### 有效线索量

说明：统计期内，新增的咨询客户中，有效线索的数量。

展现：线状图。

条件：年、月、线上线下

维度：年、月、线上线下

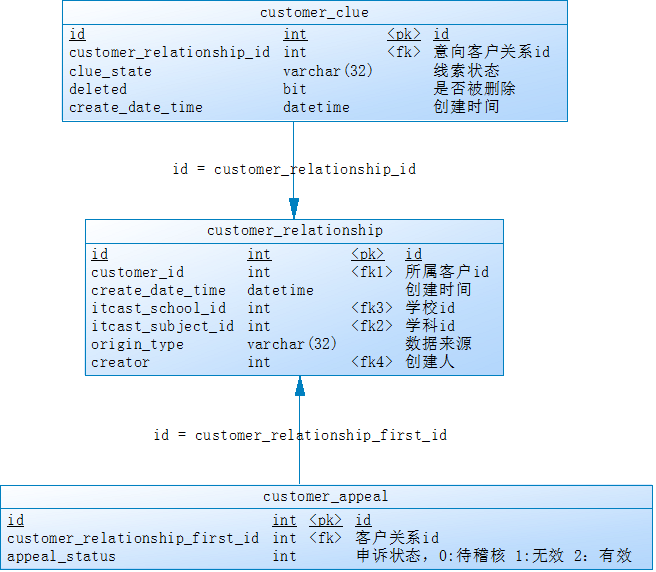
指标：有效线索的数量

粒度：天

数据来源：客户管理系统的customer\_clue线索表、customer\_relationship意向表、customer\_appeal申诉表

SQL：同上

## 原始数据结构



有效线索指标的原始数据为客户管理系统的customer\_clue线索表和customer\_relationship意向客户表。

customer\_clue是线索事实表，customer\_relationship表主要是用来判断数据来源为线上还是线下。

这两张表在意向客户指标的ODS层已经抽取过，此处可以直接复用。

customer\_appeal表是线索申诉表，主要用来判断客户线索被投诉无效。

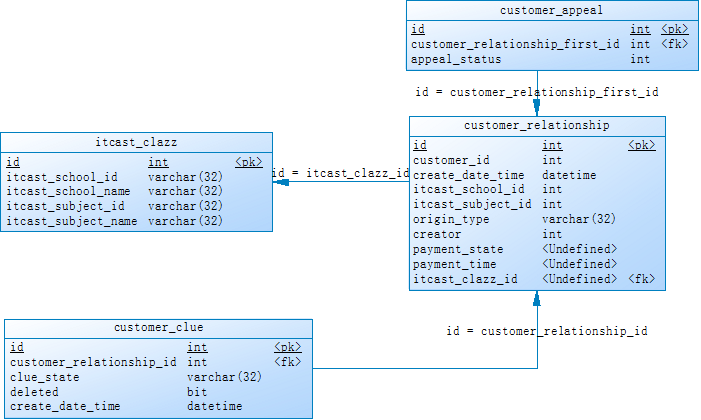
测试数据：已包含在意向客户主题测试sql中，无需重复导入。

|  |
| --- |
| CREATE TABLE `customer\_appeal` (  `id` int(11) NOT NULL AUTO\_INCREMENT COMMENT '主键',  `customer\_relationship\_first\_id` int(11) NOT NULL COMMENT '第一条客户关系id',  `employee\_id` int(11) DEFAULT NULL COMMENT '申诉人',  `employee\_name` varchar(64) COLLATE utf8\_bin DEFAULT NULL COMMENT '申诉人姓名',  `employee\_department\_id` int(11) DEFAULT NULL COMMENT '申诉人部门',  `employee\_tdepart\_id` int(11) DEFAULT NULL COMMENT '申诉人所属部门',  `appeal\_status` int(1) NOT NULL COMMENT '申诉状态，0:待稽核 1:无效 2：有效',  `audit\_id` int(11) DEFAULT NULL COMMENT '稽核人id',  `audit\_name` varchar(255) COLLATE utf8\_bin DEFAULT NULL COMMENT '稽核人姓名',  `audit\_department\_id` int(11) DEFAULT NULL COMMENT '稽核人所在部门',  `audit\_department\_name` varchar(255) COLLATE utf8\_bin DEFAULT NULL COMMENT '稽核人部门名称',  `audit\_date\_time` datetime DEFAULT NULL COMMENT '稽核时间',  `create\_date\_time` datetime DEFAULT *CURRENT\_TIMESTAMP* COMMENT '创建时间（申诉时间）',  `update\_date\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP COMMENT '更新时间',  `deleted` bit(1) NOT NULL DEFAULT b'0' COMMENT '删除标志位',  `tenant` int(11) NOT NULL DEFAULT '0',  PRIMARY KEY (`id`),  KEY `id` (`id`,`appeal\_status`) USING BTREE,  KEY `idx\_customer\_relationship\_first\_id` (`customer\_relationship\_first\_id`) USING BTREE ) ENGINE=InnoDB AUTO\_INCREMENT=2012358 DEFAULT CHARSET=utf8 COLLATE=utf8\_bin; |

# 报名用户主题看板

## 主题需求

此主题下指标需要能够下钻到小时数据。



### 校区报名柱状图

说明：统计期内，全部报名客户中，各校区报名人数分布。

展现：柱状图

条件：年、月，校区

维度：天区间，按查询条件来定

指标：报名人数

粒度：天/线上线下/校区

数据来源：客户管理系统的customer\_relationship、itcast\_clazz报名课程表

SQL：

1. **SELECT**
2. count( 1 ) '报名数量',
3. DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) '时间',
4. **ic.itcast\_school\_name** '校区名称'
5. **FROM**
6. **customer\_relationship** cr
7. LEFT JOIN **itcast\_clazz** ic **ON** **cr.itcast\_clazz\_id = ic.id**
8. **WHERE**
9. **cr.payment\_state =** **'PAID'**
10. AND DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) >= '2019-08-01'
11. AND DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) <= '2019-12-01'
12. **GROUP** **BY**
13. DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ),
14. **ic.itcast\_school\_id**
15. **ORDER** **BY**
16. DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) **ASC**,
17. count( 1 ) **DESC**;

### 学科报名柱状图

说明：统计期内，全部报名客户中，各学科报名人数分布。

展现：柱状图

条件：年、月，学科

维度：天区间，按查询条件来定

指标：报名人数

粒度：天/线上线下/学科

数据来源：客户管理系统的customer\_relationship、itcast\_clazz报名课程表

SQL：

1. **SELECT**
2. count( 1 ) '报名数量',
3. DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) '时间',
4. **ic.itcast\_subject\_name** '学科名称'
5. **FROM**
6. customer\_relationship cr
7. LEFT JOIN itcast\_clazz ic **ON** cr.itcast\_clazz\_id = ic.id
8. **WHERE**
9. cr.payment\_state = 'PAID'
10. AND DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) >= '2019-08-01'
11. AND DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) <= '2019-12-01'
12. **GROUP** **BY**
13. DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ),
14. **ic.itcast\_subject\_id**
15. **ORDER** **BY**
16. DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) **ASC**,
17. count( 1 ) **DESC**;

### 总报名量

说明：统计期内，已经缴费的报名客户总量。

展现：数值。

条件：年、月

维度：年、月

指标：报名客户总量

粒度：天

数据来源：客户管理系统的customer\_relationship表

SQL：

1. **SELECT**
2. count( 1 ) '报名数量',
3. DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) '时间'
4. **FROM**
5. **customer\_relationship cr**
6. **WHERE**
7. cr.payment\_state = 'PAID'
8. AND DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) >= '2019-08-01'
9. AND DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) <= '2019-12-01'
10. **GROUP** **BY**
11. DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' )
12. **ORDER** **BY**
13. DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) **ASC**,
14. count( 1 ) **DESC**;

### 线上报名量

说明：总报名量中来源渠道为线上访客渠道的报名总量

展现：线状图。

条件：年、月

维度：天区间，按查询条件来定

指标：报名客户总量

粒度：天

数据来源：客户管理系统的customer\_relationship表

SQL：

1. **SELECT**
2. count( 1 ) '报名数量',
3. DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) '时间'
4. **FROM**
5. customer\_relationship cr
6. **WHERE**
7. cr.payment\_state = 'PAID'
8. **AND cr.origin\_type IN ('NETSERVICE','PRESIGNUP')**
9. AND DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) >= '2019-08-01'
10. AND DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) <= '2019-12-01'
11. **GROUP** **BY**
12. DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' )
13. **ORDER** **BY**
14. DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) **ASC**,
15. count( 1 ) **DESC**;

### 意向用户报名转化率

说明：统计期内，新增的意向客户中报名的客户占比。**全部报名人数 / 全部新增的意向人数**

展现：线状图。双轴：全部报名人数、报名转化率。

条件：年、月

维度：天/线上线下

指标：报名转化率=全部报名人数/全部新增的意向人数

粒度：天

数据来源：客户管理系统的customer\_relationship表

SQL：

1. **SELECT**
2. CONCAT(
3. **signup\_num.num / total\_num.num \* 100**,
4. '%'
5. ) '报名转化率',
6. total\_num.date\_day '时间'
7. **FROM**
8. **(**
9. **SELECT**
10. count( 1 ) **AS** num,
11. DATE\_FORMAT( cr.create\_date\_time, '%Y-%m-%d' ) **AS** date\_day
12. **FROM**
13. customer\_relationship cr
14. **GROUP** **BY**
15. DATE\_FORMAT( cr.create\_date\_time, '%Y-%m-%d' )
16. **) total\_num**
17. **INNER** JOIN **(**
18. **SELECT**
19. count( 1 ) **AS** num,
20. DATE\_FORMAT( cr.create\_date\_time, '%Y-%m-%d' ) **AS** date\_day
21. **FROM**
22. customer\_relationship cr
23. **WHERE**
24. **cr.payment\_state = 'PAID'**
25. **GROUP** **BY**
26. DATE\_FORMAT( cr.create\_date\_time, '%Y-%m-%d' )
27. **)** **signup\_num** **ON** total\_num.date\_day = signup\_num.date\_day

### 有效线索报名转化率

说明：**线上报名量 / 线上有效线索量**，与上一个指标类似，此处的线索量需要排除已申诉数据。

展现：线状图。双轴：线上报名人数、线上报名转化率。

条件：年、月

维度：天/线上线下

指标：线上报名转化率=线上报名人数/线上有效线索量

粒度：天

数据来源：客户管理系统的customer\_relationship表、customer\_clue表、customer\_appeal表

SQL：

1. **SELECT**
2. CONCAT(
3. signup\_num.num / total\_num.num \* 100,
4. '%'
5. ) '线上报名转化率',
6. total\_num.date\_day '时间'
7. **FROM**
8. **(**
9. **SELECT**
10. count( 1 ) **AS** num,
11. DATE\_FORMAT( cr.create\_date\_time, '%Y-%m-%d' ) **AS** date\_day
12. **FROM**
13. customer\_clue cc
14. **LEFT** JOIN customer\_relationship cr **ON** cr.id = cc.customer\_relationship\_id
15. **WHERE**
16. cc.clue\_state IN ( 'VALID\_NEW\_CLUES', 'VALID\_PUBLIC\_NEW\_CLUE' )
17. AND !cc.deleted
18. AND cr.id NOT IN ( **SELECT** customer\_relationship\_first\_id **FROM** customer\_appeal **WHERE** appeal\_status = '1' )
19. **GROUP** **BY**
20. DATE\_FORMAT( cr.create\_date\_time, '%Y-%m-%d' )
21. **) total\_num**
22. **INNER** JOIN **(**
23. **SELECT**
24. count( 1 ) **AS** num,
25. DATE\_FORMAT( cr.create\_date\_time, '%Y-%m-%d' ) **AS** date\_day
26. **FROM**
27. customer\_relationship cr
28. **WHERE**
29. cr.payment\_state = 'PAID'
30. **GROUP** **BY**
31. DATE\_FORMAT( cr.create\_date\_time, '%Y-%m-%d' )
32. **) signup\_num** **ON** total\_num.date\_day = signup\_num.date\_day

### 日报名趋势图

说明：统计期内，每天报名人数的趋势图。

展现：线状图。

条件：年、月

维度：天/线上线下

指标：报名人数

粒度：天

数据来源：客户管理系统的customer\_relationship表

SQL：

1. **SELECT**
2. count( 1 ) '报名人数',
3. DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' )
4. **FROM**
5. **customer\_relationship cr**
6. **WHERE**
7. cr.payment\_state = 'PAID'
8. AND DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) >= '2019-08-01'
9. AND DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' ) <= '2019-12-01'
10. **GROUP** **BY**
11. DATE\_FORMAT( cr.payment\_time, '%Y-%m-%d' )

### 校区学科的报名学员TOP

说明：统计期内，全部报名学员中，校区学科排行榜，topN。A校区b学科第一，B校区a学科第二等等。

展现：柱状图

条件：年、月，校区，学科，数据量N

维度：天/线上线下

指标：报名学员人数

粒度：各校区各学科的报名人数和

数据来源：客户管理系统的customer\_relationship表、itcast\_clazz表

SQL：

1. **SELECT**
2. count(1) '总数',
3. itc.itcast\_school\_id,
4. itc.itcast\_school\_name,
5. itc.itcast\_subject\_id,
6. itc.itcast\_subject\_name,
7. cr.payment\_state,
8. cr.payment\_time
9. **FROM**
10. customer\_relationship cr
11. LEFT JOIN itcast\_clazz itc **ON** cr.itcast\_clazz\_id = itc.id
12. **WHERE**
13. cr.payment\_state = 'PAID'
14. AND cr.payment\_time >= '2019-10-01'
15. AND cr.payment\_time <= '2020-10-31 23:59:59'
16. **GROUP** **BY**
17. **itc.itcast\_school\_id,**
18. **itc.itcast\_subject\_id**
19. **ORDER** **BY**
20. count(1) **DESC**;

### 来源渠道占比

说明：统计期内，全部报名学员中，不同来源渠道的报名学员占比情况。

展现：饼状图

条件：年、月

维度：天/线上线下/来源渠道

指标：比值

数据来源：客户管理系统的customer\_relationship表

SQL：

1. **SELECT**
2. count( 1 ) '总数',
3. cr.origin\_type,
4. cr.payment\_state,
5. cr.payment\_time
6. **FROM**
7. customer\_relationship cr
8. **WHERE**
9. cr.payment\_state = 'PAID'
10. AND cr.payment\_time >= '2019-10-01'
11. AND cr.payment\_time <= '2019-10-31 23:59:59'
12. **GROUP** **BY**
13. **cr.origin\_type**;

### 咨询中心报名贡献

说明：统计期内，全部报名学员中，各咨询中心的报名学员人数占比情况。

展现：饼状图

条件：年、月，咨询中心

维度：天/线上线下/咨询中心

指标：报名学员人数

粒度：天/报名学员人数

数据来源：客户管理系统的customer\_relationship表、employee表、scrm\_department表

SQL：

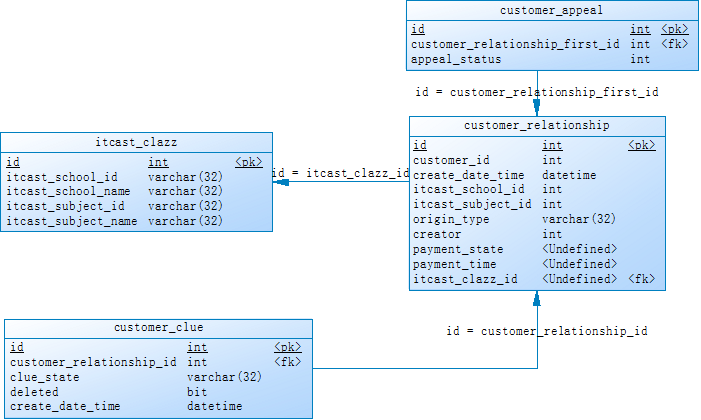
1. **SELECT**
2. count( 1 ),
3. e.tdepart\_id,
4. sd.`**name**`
5. **FROM**
6. customer\_relationship cr
7. LEFT JOIN employee e **ON** cr.creator = e.id
8. LEFT JOIN scrm\_department sd **ON** e.tdepart\_id = sd.id
9. **WHERE**
10. cr.payment\_state = 'PAID'
11. AND cr.payment\_time >= '2019-10-01'
12. AND cr.payment\_time <= '2019-10-31 23:59:59'
13. **GROUP** **BY**
14. **e.tdepart\_id**;

## 原始数据结构

customer\_relationship、employee、scrm\_department三张表，在之前的主题统计中已经实现，可以复用。

itcast\_clazz报名课程表可以获取客户报名的校区和学科。

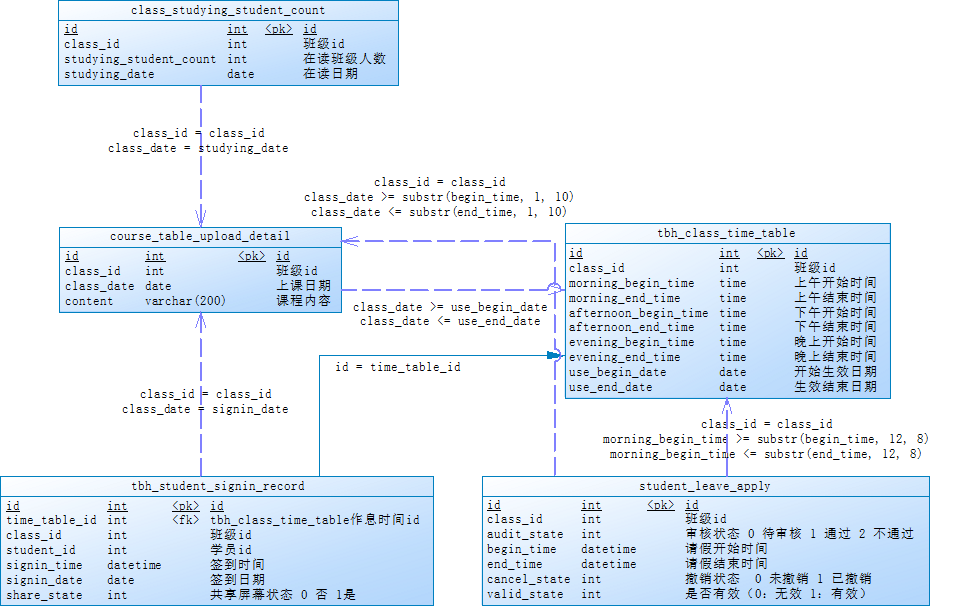
测试数据：【Home\讲义\第7章 报名用户主题看板\mysql测试数据】



|  |
| --- |
| CREATE TABLE `itcast\_clazz` (  `id` int(11) NOT NULL COMMENT 'ems课程id(非自增)',  `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）',  `itcast\_school\_id` varchar(32) COLLATE utf8\_bin DEFAULT '' COMMENT 'ems校区ID',  `itcast\_school\_name` varchar(32) COLLATE utf8\_bin DEFAULT '' COMMENT 'ems校区名称',  `itcast\_subject\_id` varchar(32) COLLATE utf8\_bin DEFAULT '' COMMENT 'ems学科ID',  `itcast\_subject\_name` varchar(32) COLLATE utf8\_bin DEFAULT '' COMMENT 'ems学科名称',  `itcast\_brand` varchar(32) COLLATE utf8\_bin DEFAULT NULL COMMENT 'ems品牌',  `clazz\_type\_state` varchar(32) COLLATE utf8\_bin DEFAULT NULL COMMENT '班级类型状态',  `clazz\_type\_name` varchar(32) COLLATE utf8\_bin DEFAULT '' COMMENT '班级类型名称',  `teaching\_mode` varchar(32) COLLATE utf8\_bin DEFAULT NULL COMMENT '授课模式',  `start\_time` datetime DEFAULT NULL COMMENT '开班时间',  `end\_time` datetime DEFAULT NULL COMMENT '毕业时间',  `comment` varchar(255) COLLATE utf8\_bin DEFAULT '' COMMENT '备注',  `detail` varchar(32) COLLATE utf8\_bin DEFAULT NULL COMMENT '详情(比如：27期)',  `uncertain` bit(1) DEFAULT b'0' COMMENT '待定班(0:否,1:是)',  `tenant` int(11) NOT NULL DEFAULT '0',  PRIMARY KEY (`id`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_bin; |

# 学生出勤看板

## 主题需求



### 班级出勤人数

说明：统计指定时间段内，不同班级的出勤人数。打卡时间在上课前40分钟(否则认为无效)~上课时间点之内，且未早退，则为正常上课打卡。可以下钻到具体学生的出勤数据。跨天数据直接累加。

指标：出勤人数

维度：年、月、天

粒度：上午、下午、晚自习

条件：年、月

数据来源：教学实施与保障系统teach的course\_table\_upload\_detail班级课表、tbh\_student\_signin\_record学生打卡记录表、tbh\_class\_time\_table班级作息时间表。

### 班级出勤率

说明：统计指定时间段内，不同班级的学生出勤率。可以下钻到具体学生的出勤数据。出勤率=出勤人数/当日在读学员人数。

指标：出勤率

维度：年、月、天

粒度：上午、下午、晚自习

条件：年、月

数据来源：教学实施与保障系统的course\_table\_upload\_detail班级课表、tbh\_student\_signin\_record学生打卡记录表、tbh\_class\_time\_table班级作息时间表、class\_studying\_student\_count班级在读学生人数。

### 班级迟到人数

说明：统计指定时间段内，不同班级的迟到人数。上课10分钟后视为迟到。可以下钻到具体学生的迟到数据。跨天数据直接累加。

指标：迟到人数

维度：年、月、天

粒度：上午、下午、晚自习

条件：年、月

数据来源：教学实施与保障系统的course\_table\_upload\_detail班级课表、tbh\_student\_signin\_record学生打卡记录表、tbh\_class\_time\_table班级作息时间表。

### 班级迟到率

说明：统计指定时间段内，不同班级的学生迟到率。上课10分钟后视为迟到。可以下钻到具体学生的迟到数据。迟到率=迟到人数/当日在读学员人数。

指标：迟到率

维度：年、月、天

粒度：上午、下午、晚自习

条件：年、月

数据来源：教学实施与保障系统的course\_table\_upload\_detail班级课表、tbh\_student\_signin\_record学生打卡记录表、tbh\_class\_time\_table班级作息时间表、class\_studying\_student\_count班级在读学生人数。  
SQL：

|  |
| --- |
| select dt.every\_date,  ctud.class\_id,  tssr.student\_id,  *if*(  *#上午正常打卡为0，迟到10分钟以上为1，其他(请假+旷课)为2  sum*(  case  *#上午打卡时间是否在上课前40分钟~下课时间段之内* when *time*(tssr.signin\_time) between *TIMESTAMPADD*(minute, -40, tctt.morning\_begin\_time) and tctt.morning\_end\_time  then 1 *#上午来了* else 0 end *#上午没来* ) > 0, *#打卡多次，只要有一次正常打卡，就会>0，返回true；否则没来，返回false  if*(  *sum*(  case  *#上午打卡时间是否在上课前40分钟~上课后10分钟之内* when *time*(tssr.signin\_time) between *TIMESTAMPADD*(minute, -40, tctt.morning\_begin\_time) and *TIMESTAMPADD*(minute, 10, tctt.morning\_begin\_time)  then 1 *#正常出勤* else 0 end *#迟到* ) > 0, *#有一次打卡是正常出勤，就会>0，返回true；否则迟到，返回false* 0, *#正常出勤* 1 *#迟到* ),  2 *#上午没来* ) as morning\_signin,  *if*(  *#下午正常打卡为0，迟到10分钟以上为1，其他(请假+旷课)为2  sum*(case  when *time*(tssr.signin\_time) between *TIMESTAMPADD*(minute, -40, tctt.afternoon\_begin\_time) and tctt.afternoon\_end\_time  then 1  else 0 end) > 0,  *if*(*sum*(case  when *time*(tssr.signin\_time) between *TIMESTAMPADD*(minute, -40, tctt.afternoon\_begin\_time) and *TIMESTAMPADD*(minute, 10, tctt.afternoon\_begin\_time)  then 1  else 0 end) > 0, 0, 1), 2) as afternoon\_signin,  *if*(  *#晚自习正常打卡为0，迟到10分钟以上为1，其他(请假+旷课)为2  sum*(case  when *time*(tssr.signin\_time) between *TIMESTAMPADD*(minute, -20, tctt.evening\_begin\_time) and tctt.evening\_end\_time  then 1  else 0 end) > 0,  *if*(*sum*(case  when *time*(tssr.signin\_time) between *TIMESTAMPADD*(minute, -20, tctt.evening\_begin\_time) and *TIMESTAMPADD*(minute, 10, tctt.evening\_begin\_time)  then 1  else 0 end) > 0, 0, 1), 2) as evening\_signin from (  *#获取今天之前一周内的日期* select datelist as every\_date from calendar where datelist between '2019-09-01' and '2019-09-30'  ) dt  *#日期课表不为空且不是开班典礼* left join course\_table\_upload\_detail ctud  on ctud.class\_date = dt.every\_date and *ifnull*(ctud.content, '') != '' and  ctud.content != '开班典礼'  *#学生打卡记录日期和班级匹配，且开启共屏进入学习* left join tbh\_student\_signin\_record tssr  on tssr.class\_id = ctud.class\_id and tssr.signin\_date = dt.every\_date and  tssr.share\_state = 1  *#获取班级作息时间以判断是否按时出勤* left join tbh\_class\_time\_table tctt on tctt.id = tssr.time\_table\_id  *#按照日期、班级、学生分组统计* group by dt.every\_date, ctud.class\_id, tssr.student\_id; |

### 班级请假人数

说明：统计指定时间段内，不同班级的请假人数。跨天数据直接累加。

指标：请假人数

维度：年、月、天

粒度：上午、下午、晚自习

条件：年、月

数据来源：教学实施与保障系统的student\_leave\_apply学生请假申请表、tbh\_class\_time\_table班级作息时间表、course\_table\_upload\_detail班级课表。  
SQL：

|  |
| --- |
| select cud.class\_date as dateinfo,  cud.class\_id,  *count*(distinct sla.student\_id) as morning\_leave\_count from student\_leave\_apply sla,  tbh\_class\_time\_table ct,  course\_table\_upload\_detail cud *-- 表关联* where sla.class\_id = ct.class\_id = cud.class\_id *-- 课程表，当天有课程内容* AND cud.content IS NOT NULL  AND cud.content != '开班典礼' *-- 作息时间表，数据在生效期范围内* and cud.class\_date between ct.use\_begin\_date and ct.use\_end\_date *-- 请假表，请假状态已审核通过，且没有取消、数据有效* and sla.audit\_state = 1  and sla.cancel\_state = 0  and sla.valid\_state = 1 *-- 关联判断请假周期，请假时间周期要与课程和作息时间对比 -- cud.class\_date 课程表的上课日期 2020-09-16 -- ct.morning\_begin\_time 作息表的早上上课时间 09:00:00 -- 请假结束时间 >= 2020-09-16 09:00:00 >= 请假开始时间，认为上午请假了* and *concat*(cud.class\_date, ' ', ct.morning\_begin\_time) >= sla.begin\_time  and *concat*(cud.class\_date, ' ', ct.morning\_begin\_time) <= sla.end\_time group by cud.class\_date, cud.class\_id; |

### 班级请假率

说明：统计指定时间段内，不同班级的学生请假率。可以下钻到具体学生的请假数据。请假率=请假人数/当日在读学员人数。

指标：请假率

维度：年、月、天

粒度：上午、下午、晚自习

条件：年、月

数据来源：教学实施与保障系统的student\_leave\_apply学生请假申请表、class\_studying\_student\_count班级在读学生人数。

### 班级旷课人数

说明：统计指定时间段内，不同班级的旷课人数。跨天数据直接累加。旷课人数=当日在读学员人数-出勤人数-请假人数。

指标：旷课人数

维度：年、月、天

粒度：上午、下午、晚自习

条件：年、月

数据来源：教学实施与保障系统的course\_table\_upload\_detail班级课表、tbh\_student\_signin\_record学生打卡记录表、tbh\_class\_time\_table班级作息时间表、student\_leave\_apply学生请假申请表。

### 班级旷课率

说明：统计指定时间段内，不同班级的学生旷课率。旷课率=旷课人数/当日在读学员人数。

指标：旷课率

维度：年、月、天

粒度：上午、下午、晚自习

条件：年、月

数据来源：教学实施与保障系统的course\_table\_upload\_detail班级课表、tbh\_student\_signin\_record学生打卡记录表、tbh\_class\_time\_table班级作息时间表、student\_leave\_apply学生请假申请表、class\_studying\_student\_count班级在读学生人数。

SQL：

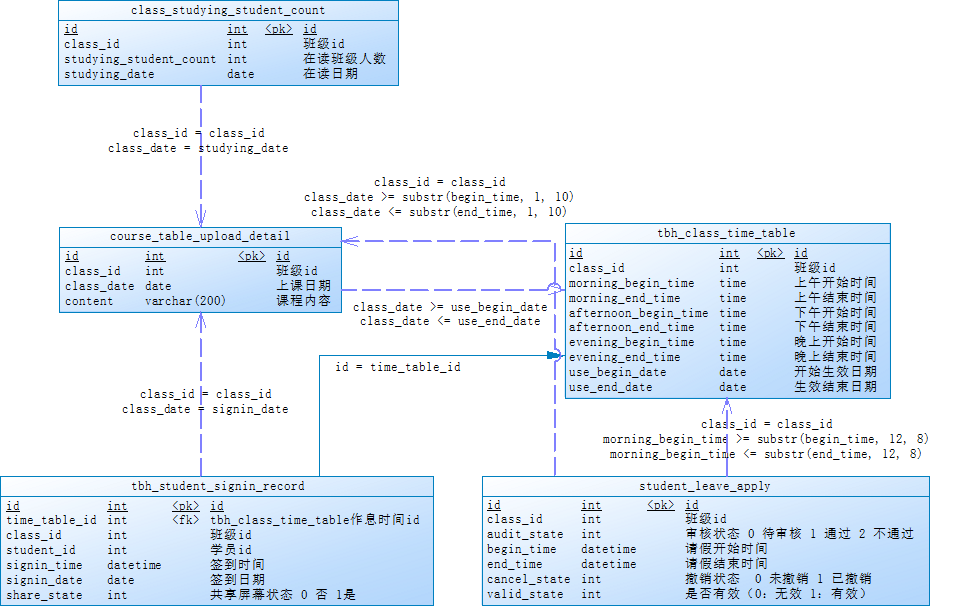
|  |
| --- |
| select *date\_format*(tmp3.every\_date, '%Y/%m/%d'),  tmp3.class\_count,  tmp3.student\_count,  tmp3.morning\_att\_count,  tmp3.morning\_late\_count,  tmp3.morning\_leave\_count,  *#减出旷课人数* (tmp3.student\_count - tmp3.morning\_att\_count - tmp3.morning\_leave\_count) as morning\_truant\_count,  *concat*(*cast*((tmp3.morning\_att\_count / tmp3.student\_count) \* 100 as decimal(8, 2)), '%') as '上午出勤率',  *concat*(*cast*((tmp3.morning\_late\_count / tmp3.student\_count) \* 100 as decimal(8, 2)), '%') as '上午迟到率',  *concat*(*cast*((tmp3.morning\_leave\_count / tmp3.student\_count) \* 100 as decimal(8, 2)), '%') as '上午请假率',  *concat*(*cast*(((tmp3.student\_count - tmp3.morning\_att\_count - tmp3.morning\_leave\_count) / tmp3.student\_count) \*  100 as decimal(8, 2)), '%') as '上午旷课率',  tmp3.afternoon\_att\_count,  tmp3.afternoon\_late\_count,  tmp3.afternoon\_leave\_count,  (tmp3.student\_count - tmp3.afternoon\_att\_count -  tmp3.afternoon\_leave\_count) as afternoon\_truant\_count,  *concat*(*cast*((tmp3.afternoon\_att\_count / tmp3.student\_count) \* 100 as decimal(8, 2)), '%') as '下午出勤率',  *concat*(*cast*((tmp3.afternoon\_late\_count / tmp3.student\_count) \* 100 as decimal(8, 2)), '%') as '下午迟到率',  *concat*(*cast*((tmp3.afternoon\_leave\_count / tmp3.student\_count) \* 100 as decimal(8, 2)), '%') as '下午请假率',  *concat*(*cast*(((tmp3.student\_count - tmp3.afternoon\_att\_count - tmp3.afternoon\_leave\_count) / tmp3.student\_count) \*  100 as decimal(8, 2)), '%') as '下午旷课率',  tmp3.evening\_att\_count,  tmp3.evening\_late\_count,  tmp3.evening\_leave\_count,  (tmp3.student\_count - tmp3.evening\_att\_count - tmp3.evening\_leave\_count) as evening\_truant\_count,  *concat*(*cast*((tmp3.evening\_att\_count / tmp3.student\_count) \* 100 as decimal(8, 2)), '%') as '晚上出勤率',  *concat*(*cast*((tmp3.evening\_late\_count / tmp3.student\_count) \* 100 as decimal(8, 2)), '%') as '晚上迟到率',  *concat*(*cast*((tmp3.evening\_leave\_count / tmp3.student\_count) \* 100 as decimal(8, 2)), '%') as '晚上请假率',  *concat*(*cast*(((tmp3.student\_count - tmp3.evening\_att\_count - tmp3.evening\_leave\_count) / tmp3.student\_count) \*  100 as decimal(8, 2)), '%') as '晚上旷课率' from (  select tmp2.every\_date,  *count*(tmp2.class\_id) as class\_count,  *sum*(tmp2.student\_count) as student\_count,  *sum*(tmp2.morning\_att\_count) as morning\_att\_count,  *sum*(tmp2.morning\_late\_count) as morning\_late\_count,  *sum*(tmp2.morning\_leave\_count) as morning\_leave\_count,  *sum*(tmp2.afternoon\_att\_count) as afternoon\_att\_count,  *sum*(tmp2.afternoon\_late\_count) as afternoon\_late\_count,  *sum*(tmp2.afternoon\_leave\_count) as afternoon\_leave\_count,  *sum*(tmp2.evening\_att\_count) as evening\_att\_count,  *sum*(tmp2.evening\_late\_count) as evening\_late\_count,  *sum*(tmp2.evening\_leave\_count) as evening\_leave\_count  from (  select tmp.every\_date,  tmp.class\_id,  *#班级人数* (select cssc.studying\_student\_count  from class\_studying\_student\_count cssc  where cssc.studying\_date = tmp.every\_date  and cssc.class\_id = tmp.**class\_id**) as student\_count,  *#上午出勤人数(包括迟到)  count*(distinct (  case  *#如果正常打卡，或者迟到，只要来了，就算到出勤人数中，返回student\_id后去重* when tmp.morning\_signin = 0 or tmp.morning\_signin = 1 then tmp.student\_id  *#如果没来，返回null，count不计算到出勤人数* else null end  )) as morning\_att\_count,  *#上午迟到人数  count*(distinct (  *#如果迟到，则返回student\_id后去重* case  when tmp.morning\_signin = 1 then tmp.student\_id  else null end  )) as morning\_late\_count,  *#下午出勤人数(包括迟到)  count*(distinct (case  when tmp.afternoon\_signin = 0 or tmp.afternoon\_signin = 1  then tmp.student\_id  else null end)) as afternoon\_att\_count,  *#下午迟到人数  count*(distinct  (case when tmp.afternoon\_signin = 1 then tmp.student\_id else null end)) as afternoon\_late\_count,  *#晚自习出勤人数(包括迟到)  count*(distinct (case  when tmp.evening\_signin = 0 or tmp.evening\_signin = 1 then tmp.student\_id  else null end)) as evening\_att\_count,  *#晚自习迟到人数  count*(distinct  (case when tmp.evening\_signin = 1 then tmp.student\_id else null end)) as evening\_late\_count,  *#上午请假学生人数，审批通过、未撤销、有效、班级匹配、请假时间在课表上课时间之内* (select *count*(distinct sla.student\_id)  from student\_leave\_apply sla  where sla.audit\_state = 1  and sla.cancel\_state = 0  and sla.valid\_state = 1  and sla.class\_id = tmp.**class\_id** and *concat*(tmp.every\_date, ' ', tctt2.**morning\_begin\_time**) >= sla.begin\_time  and *concat*(tmp.every\_date, ' ', tctt2.**morning\_begin\_time**) <=  sla.end\_time) as morning\_leave\_count,  *#下午请假学生人数，审批通过、未撤销、有效、班级匹配、请假时间在课表上课时间之内* (select *count*(distinct sla.student\_id)  from student\_leave\_apply sla  where sla.audit\_state = 1  and sla.cancel\_state = 0  and sla.valid\_state = 1  and sla.class\_id = tmp.**class\_id** and *concat*(tmp.every\_date, ' ', tctt2.**afternoon\_begin\_time**) >= sla.begin\_time  and *concat*(tmp.every\_date, ' ', tctt2.**afternoon\_begin\_time**) <=  sla.end\_time) as afternoon\_leave\_count,  *#晚自习请假学生人数，审批通过、未撤销、有效、班级匹配、请假时间在课表上课时间之内* (select *count*(distinct sla.student\_id)  from student\_leave\_apply sla  where sla.audit\_state = 1  and sla.cancel\_state = 0  and sla.valid\_state = 1  and sla.class\_id = tmp.**class\_id** and *concat*(tmp.every\_date, ' ', tctt2.**evening\_begin\_time**) >= sla.begin\_time  and *concat*(tmp.every\_date, ' ', tctt2.**evening\_begin\_time**) <=  sla.end\_time) as evening\_leave\_count  from (  *#上午、下午、晚自习：正常打卡为0，迟到10分钟以上为1，其他(请假+旷课)为2* select dt.every\_date,  ctud.class\_id,  tssr.student\_id,  *if*(  *#上午正常打卡为0，迟到10分钟以上为1，其他(请假+旷课)为2  sum*(  case  *#上午打卡时间是否在上课前40分钟~下课时间段之内* when *time*(tssr.signin\_time) between *TIMESTAMPADD*(minute, -40, tctt.morning\_begin\_time) and tctt.morning\_end\_time  then 1 *#上午来了* else 0 end *#上午没来* ) > 0, *#打卡多次，只要有一次正常打卡，就会>0，返回true；否则没来，返回false  if*(  *sum*(  case  *#上午打卡时间是否在上课前40分钟~上课后10分钟之内* when *time*(tssr.signin\_time) between *TIMESTAMPADD*(minute, -40, tctt.morning\_begin\_time) and *TIMESTAMPADD*(minute, 10, tctt.morning\_begin\_time)  then 1 *#正常出勤* else 0 end *#迟到* ) > 0, *#有一次打卡是正常出勤，就会>0，返回true；否则迟到，返回false* 0, *#正常出勤* 1 *#迟到* ),  2 *#上午没来* ) as morning\_signin,  *if*(  *#下午正常打卡为0，迟到10分钟以上为1，其他(请假+旷课)为2  sum*(case  when *time*(tssr.signin\_time) between *TIMESTAMPADD*(minute, -40, tctt.afternoon\_begin\_time) and tctt.afternoon\_end\_time  then 1  else 0 end) > 0,  *if*(*sum*(case  when *time*(tssr.signin\_time) between *TIMESTAMPADD*(minute, -40, tctt.afternoon\_begin\_time) and *TIMESTAMPADD*(minute, 10, tctt.afternoon\_begin\_time)  then 1  else 0 end) > 0, 0, 1), 2) as afternoon\_signin,  *if*(  *#晚自习正常打卡为0，迟到10分钟以上为1，其他(请假+旷课)为2  sum*(case  when *time*(tssr.signin\_time) between *TIMESTAMPADD*(minute, -20, tctt.evening\_begin\_time) and tctt.evening\_end\_time  then 1  else 0 end) > 0,  *if*(*sum*(case  when *time*(tssr.signin\_time) between *TIMESTAMPADD*(minute, -20, tctt.evening\_begin\_time) and *TIMESTAMPADD*(minute, 10, tctt.evening\_begin\_time)  then 1  else 0 end) > 0, 0, 1), 2) as evening\_signin  from (  *#获取今天之前一周内的日期* select datelist as every\_date  from calendar  where datelist between '2019-09-01' and '2019-09-30'  ) dt  *#日期课表不为空且不是开班典礼* left join course\_table\_upload\_detail ctud  on ctud.class\_date = dt.every\_date and *ifnull*(ctud.content, '') != '' and  ctud.content != '开班典礼'  *#学生打卡记录日期和班级匹配，且开启共屏进入学习* left join tbh\_student\_signin\_record tssr  on tssr.class\_id = ctud.class\_id and tssr.signin\_date = dt.every\_date and  tssr.share\_state = 1  *#获取班级作息时间以判断是否按时出勤* left join tbh\_class\_time\_table tctt on tctt.id = tssr.time\_table\_id  *#按照日期、班级、学生分组统计* group by dt.every\_date, ctud.class\_id, tssr.student\_id  ) as tmp  *#获取班级作息时间以判断是否按时出勤* left join tbh\_class\_time\_table tctt2  *#班级id相等，且作息数据当天正在生效中* on tctt2.class\_id = tmp.class\_id and tmp.every\_date >= tctt2.use\_begin\_date and  tmp.every\_date <= tctt2.use\_end\_date  *#按照日期和班级统计* group by tmp.every\_date, tmp.class\_id  ) as tmp2  *#按照日期统计* group by tmp2.every\_date  ) as tmp3; |

## 原始数据结构

**建库：**

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

测试数据：【Home\讲义\第8章 学生出勤主题看板\mysql测试数据】



### 在读学员人数指标

#### class\_studying\_student\_count班级在读学生人数

根据班级和日期，查询出班级在读的学生人数。

|  |
| --- |
| create table class\_studying\_student\_count  (  id int auto\_increment  primary key,  school\_id int null comment '校区id',  subject\_id int null comment '学科id',  class\_id int null comment '**班级id**',  studying\_student\_count int null comment '**在读班级人数**',  studying\_date date null comment '**在读日期**'  )  comment '在读班级的每天在读学员人数';  create index idx\_classid\_in\_classstudyingstucount\_0  on class\_studying\_student\_count (class\_id);  create index idx\_schoolid\_in\_classstudyingstucount\_0  on class\_studying\_student\_count (school\_id);  create index idx\_studyingdate\_in\_classstudystudentcount\_4  on class\_studying\_student\_count (studying\_date);  create index idx\_subjectid\_in\_classstudyingstucount\_3  on class\_studying\_student\_count (subject\_id); |

### 出勤人数指标

#### course\_table\_upload\_detail班级课表

用来判断班级是否有课，而且不是开班典礼。

|  |
| --- |
| create table course\_table\_upload\_detail (  id int auto\_increment comment 'id'  primary key,  base\_id int null comment '课程主表id',  class\_id int null comment '班级id',  class\_date date null comment '上课日期',  content varchar(200) null comment '课程内容',  teacher\_id int null comment '老师id',  teacher\_name varchar(20) null comment '老师名字',  job\_number varchar(45) null comment '工号',  classroom\_id int null comment '教室id',  classroom\_name varchar(50) null comment '教室名称',  is\_outline int null comment '是否大纲 0 否 1 是',  class\_mode int null comment '上课模式 0 传统全天 1 AB上午 2 AB下午 3 线上直播',  is\_stage\_exam int null comment '是否阶段考试（0：否 1：是）',  is\_pay int null comment '代课费（0：无 1：有）',  tutor\_teacher\_id int null comment '晚自习辅导老师id',  tutor\_teacher\_name varchar(20) null comment '辅导老师姓名',  tutor\_job\_number varchar(45) null comment '晚自习辅导老师工号',  is\_subsidy int null comment '晚自习补贴（0：无 1：有）',  answer\_teacher\_id int null comment '答疑老师id',  answer\_teacher\_name varchar(20) null comment '答疑老师姓名',  answer\_job\_number varchar(45) null comment '答疑老师工号',  remark varchar(2000) null comment '备注',  create\_time datetime null comment '创建时间' )  comment '班级课表明细表';  create index idx\_tutorueacheranddatesubsidy\_in\_ctud\_4  on course\_table\_upload\_detail (tutor\_teacher\_id, class\_date, is\_subsidy);  create index idx\_tutorueacheranddatesubsidy\_in\_ctud\_5  on course\_table\_upload\_detail (class\_date, tutor\_teacher\_id, is\_subsidy);  create index index\_coursetableupload\_classId\_2  on course\_table\_upload\_detail (class\_id);  create index index\_coursetableupload\_classId\_classDate\_6  on course\_table\_upload\_detail (class\_id, class\_date);  create index index\_coursetableupload\_jobNumber\_7  on course\_table\_upload\_detail (job\_number);  create index index\_coursetableuploaddetail\_answer\_teacher\_id\_7  on course\_table\_upload\_detail (answer\_teacher\_id);  create index index\_coursetableuploaddetail\_teacher\_id\_1  on course\_table\_upload\_detail (teacher\_id);  create index index\_coursetupload\_2\_baseid\_0  on course\_table\_upload\_detail (base\_id); |

#### tbh\_student\_signin\_record学生打卡记录表

用来判断学生是否开启共屏进入学习。

|  |
| --- |
| create table tbh\_student\_signin\_record (  id int auto\_increment comment '主键id'  primary key,  normal\_class\_flag int null comment '是否正课 1 正课 2 自习 3 休息',  time\_table\_id int null comment '作息时间id normal\_class\_flag=2 关联tbh\_school\_time\_table 或者 normal\_class\_flag=1 关联 tbh\_class\_time\_table',  class\_id int null comment '班级id',  student\_id int null comment '学员id',  signin\_time datetime null comment '签到时间',  signin\_date date null comment '签到日期',  inner\_flag int null comment '内外网标志 0 外网 1 内网',  signin\_type int null comment '签到类型 1 心跳打卡 2 老师补卡 3 直播打卡',  share\_state int default 0 null comment '共享屏幕状态 0 否 1是 在上午或下午段有共屏记录，则该段所有记录该字段为1，内网默认为1 外网默认为0 (暂不用)',  inner\_ip varchar(20) null comment '内网ip地址',  create\_time datetime null comment '创建时间' );  create index idx\_sharestateinnerflag\_in\_tbhstusigninrecord\_2  on tbh\_student\_signin\_record (share\_state, inner\_flag);  create index idx\_signindate\_in\_tbhstusigninrecord\_4  on tbh\_student\_signin\_record (signin\_date);  create index idx\_signintime\_in\_tbhstusigninrecord\_1  on tbh\_student\_signin\_record (signin\_time);  create index idx\_timetableid\_in\_tbhsigninrecord\_3  on tbh\_student\_signin\_record (time\_table\_id);  create index index\_student\_signin\_record\_class\_student\_id  on tbh\_student\_signin\_record (class\_id, student\_id, inner\_ip); |

#### tbh\_class\_time\_table班级作息时间表

用来获取班级作息时间以判断学生是否按时出勤。

|  |
| --- |
| create table tbh\_class\_time\_table (  id int auto\_increment comment '主键id'  primary key,  class\_id int null comment '班级id',  morning\_template\_id int null comment '上午出勤模板id',  morning\_begin\_time time null comment '上午开始时间',  morning\_end\_time time null comment '上午结束时间',  afternoon\_template\_id int null comment '下午出勤模板id',  afternoon\_begin\_time time null comment '下午开始时间',  afternoon\_end\_time time null comment '下午结束时间',  evening\_template\_id int null comment '晚上出勤模板id',  evening\_begin\_time time null comment '晚上开始时间',  evening\_end\_time time null comment '晚上结束时间',  use\_begin\_date date null comment '使用开始日期(开始生效时间)',  use\_end\_date date null comment '使用结束日期(生效结束时间)',  create\_time datetime null comment '创建时间',  create\_person int null comment '创建人',  remark varchar(500) null comment '备注' );  create index idx\_classid\_state\_in\_classtimetable\_0  on tbh\_class\_time\_table (class\_id); |

### 迟到人数指标

同出勤人数指标。

### 请假人数指标

#### student\_leave\_apply学生请假申请表

用来获取请假学生，条件：审批通过、未撤销、有效、班级匹配、请假时间包含了课表上课时间。

|  |
| --- |
| create table student\_leave\_apply  (  id int auto\_increment  primary key,  class\_id int null comment '**班级id**',  student\_id int null comment '学员id',  audit\_state int default 0 null comment '**审核状态** 0 待审核 1 通过 2 不通过',  audit\_person int null comment '审核人',  audit\_time datetime null comment '审核时间',  audit\_remark varchar(500) null comment '审核备注',  leave\_type int null comment '请假类型 1 请假 2 销假 （查询是否请假不用过滤此类型，通过有效状态来判断）',  leave\_reason int null comment '请假原因 1 事假 2 病假',  begin\_time datetime null comment '**请假开始时间**',  begin\_time\_type int null comment '1：上午 2：下午 3：晚自习',  end\_time datetime null comment '**请假结束时间**',  end\_time\_type int null comment '1：上午 2：下午 3：晚自习',  days float null comment '请假/已休天数',  cancel\_state int default 0 null comment '**撤销状态** 0 未撤销 1 已撤销',  cancel\_time datetime null comment '撤销时间',  old\_leave\_id int null comment '原请假id，只有leave\_type =2 销假的时候才有',  leave\_remark varchar(500) null comment '请假/销假说明',  valid\_state int default 1 null comment '**是否有效**（0：无效 1：有效）',  create\_time datetime null comment '创建时间'  );  create index index\_student\_leave\_apply\_begin\_time\_1  on student\_leave\_apply (begin\_time);  create index index\_student\_leave\_apply\_class\_student\_id  on student\_leave\_apply (class\_id, student\_id);  create index index\_student\_leave\_apply\_end\_time\_2  on student\_leave\_apply (end\_time); |

### 旷课人数指标

旷课人数是通过之前的指标结果计算出来的。

旷课人数=班级在读学生人数 - 出勤人数（包括迟到）- 旷课人数。

因此只要前面的指标统计出来，旷课人数的结果就有了。