# 第一部分：建表sql

-- 骑行数据

CREATE TABLE trips (

user\_id BIGINT COMMENT '用户id',

start\_time DATETIME COMMENT '骑行开始时间',

stop\_time DATETIME COMMENT '骑行结束时间',

bike\_id BIGINT COMMENT '单车id',

time\_sec BIGINT COMMENT '骑行时长',

from\_station\_id BIGINT COMMENT '起始站ID',

from\_station\_name STRING COMMENT '起始站名称',

to\_station\_id BIGINT COMMENT '终点站id',

to\_station\_name STRING COMMENT '终点站名称',

user\_type STRING COMMENT '用户类型',

gender STRING COMMENT '用户性别',

birth\_year STRING COMMENT '出生年份'

);

-- 站点信息

CREATE TABLE stations (

station\_id BIGINT COMMENT '站点ID',

station\_name STRING COMMENT '站点名称',

station\_city STRING COMMENT '站点所属地市',

station\_lat STRING COMMENT '站点纬度',

station\_lon STRING COMMENT '站点经度'

);

-- 热门线路

CREATE TABLE hot\_line (

order\_id BIGINT COMMENT '序号',

from\_station\_id BIGINT COMMENT '起始站ID',

from\_station\_name STRING COMMENT '起始站名称',

to\_station\_id BIGINT COMMENT '终点站id',

to\_station\_name STRING COMMENT '终点站名称',

cnt BIGINT COMMENT '频次',

ctime DATETIME COMMENT '创建时间'

) partitioned by (pt string);

-- 热门站点

CREATE TABLE hot\_stations (

order\_id BIGINT COMMENT '序号',

station\_id BIGINT COMMENT '站点ID',

station\_name STRING COMMENT '站点名称',

cnt BIGINT COMMENT '频次',

ctime DATETIME COMMENT '创建时间'

)partitioned by (pt string);

-- 热门时段

CREATE TABLE hot\_hour (

order\_id BIGINT COMMENT '序号',

start\_hour STRING COMMENT '时刻',

cnt BIGINT COMMENT '频次',

day\_of\_week STRING COMMENT '星期几',

ctime DATETIME COMMENT '创建时间'

)partitioned by (pt string);

-- KPI指标

CREATE TABLE kpi\_index (

index\_id STRING COMMENT '指标ID',

index\_name STRING COMMENT '指标名称',

index\_type STRING COMMENT '指标单位',

index\_value STRING COMMENT '指标值',

ctime DATETIME COMMENT '创建时间'

)partitioned by (pt string);

-- 站点车辆调度信息

CREATE TABLE bike\_control (

order\_id BIGINT COMMENT '序号',

start\_hour STRING COMMENT '时间',

station\_id BIGINT COMMENT '站点ID',

station\_name STRING COMMENT '站点名称',

stock\_cnt BIGINT COMMENT '存量',

avg\_cnt BIGINT COMMENT '需求量',

allot\_cnt BIGINT COMMENT '可分配量',

ctime DATETIME COMMENT '创建时间'

)partitioned by (pt string);

# 第二部分：数据分析sql

## 任务一：共享单车骑行分析

### 1）热门线路

**计算口径**：

统计近30天内，出发地到目的地，有效骑行次数（骑行时长超过1分钟）排名前10

**参数配置**：

sdate=$[yyyy-mm-dd-30]

yesterday=$[yyyy-mm-dd-1]

**统计sql**:

insert overwrite table hot\_line partition (pt)

select ROW\_NUMBER() OVER (ORDER BY cnt DESC),

a.\* ,getdate(),'${yesterday}'

from (

select from\_station\_id

,from\_station\_name

,to\_station\_id

,to\_station\_name

,count(1) as cnt

from trips

where substr(start\_time,1,10)>='${sdate}'

and substr(start\_time,1,10)<='${yesterday}'

and time\_sec>'60'

group by from\_station\_id ,from\_station\_name,to\_station\_id,to\_station\_name

) a

limit 10

### 2）热门站点

**计算口径**：

统计近30天内，起始站点有效骑行次数（骑行时长超过1分钟）最多的前5名

**参数配置**：

sdate=$[yyyy-mm-dd-30]

yesterday=$[yyyy-mm-dd-1]

**统计sql**:

insert overwrite table hot\_stations partition (pt)

select ROW\_NUMBER() OVER (ORDER BY cnt DESC),

a.\* ,getdate(),'${yesterday}'

from (

select from\_station\_id

,from\_station\_name

,count(1) as cnt

from trips

where substr(start\_time,1,10)>='${sdate}'

and substr(start\_time,1,10)<='${yesterday}'

and time\_sec>'60'

group by from\_station\_id ,from\_station\_name

) a

limit 5

### 3 ) 高峰时段

**计算口径**：

统计当天各个时间段所有站点的骑行需求，利用曲线趋势图进行分析。

**参数配置**：

yesterday=$[yyyy-mm-dd-1]

**统计sql**:

insert overwrite table hot\_hour partition (pt)

select ROW\_NUMBER() OVER (ORDER BY cnt DESC),

a.\* ,

weekday(to\_date('${yesterday}', 'yyyy-mm-dd'))+1 ,

getdate(),

'${yesterday}'

from (

select substr(start\_time,12,2)

,count(1) as cnt

from trips

where substr(start\_time,1,10)='${yesterday}'

and time\_sec>'60'

group by substr(start\_time,12,2)

) a

### 4）KPI指标

**计算口径**：

说明：以下统计均为有效骑行记录（骑行时长超过1分钟）

单车日利用率=单车有效使用数/单车总数

单车日周转率=日骑行总次数/单车总数

日活跃用户=当日有有效骑行记录的独立用户数。

日骑行次数=当日骑行总次数

日均骑行时长=当日骑行总时长/当日骑行总次数

**参数配置**：

yesterday=$[yyyy-mm-dd-1]

**统计sql**:

insert overwrite table kpi\_index partition (pt)

select 'IDX\_001','单车日利用率','%',

round(count(distinct bike\_id)/max(bike\_id), 4) \*100,

getdate(),'${yesterday}'

from trips

where substr(start\_time,1,10)='${yesterday}' and time\_sec>'60';

insert into table kpi\_index partition (pt)

select 'IDX\_002','单车日周转率','次',

round(count(1)/max(bike\_id),4) ,

getdate(),'${yesterday}'

from trips

where substr(start\_time,1,10)='${yesterday}' and time\_sec>'60';

insert into table kpi\_index partition (pt)

select 'IDX\_003','日活跃用户','人',

count(distinct user\_id),

getdate(),'${yesterday}'

from trips

where substr(start\_time,1,10)='${yesterday}' and time\_sec>'60';

insert into table kpi\_index partition (pt)

select 'IDX\_004','日骑行次数','次',

count(1),

getdate(),'${yesterday}'

from trips

where substr(start\_time,1,10)='${yesterday}' and time\_sec>'60';

insert into table kpi\_index partition (pt)

select 'IDX\_005','日均骑行时长','秒/次',

sum(time\_sec)/count(1),

getdate(),'${yesterday}'

from trips

where substr(start\_time,1,10)='${yesterday}' and time\_sec>'60';

## 任务二：共享单车调度任务

### 1）、站点存量数据

**计算口径**：

统计当前站点车辆库存量。本实验中通过获取每辆单车的最新骑行数据来判断单车停靠站点。而在实际生产环境中，每辆车的启动、停靠都会对站点车辆信息进行更新操作，可以直接读取。

**参数配置**：

day=$[yyyy-mm-dd]

time=$[hh24:mi:ss]

state=$[yyyy-mm-dd\_hh24]

**统计sql**:

-- 车辆最新状态信息

DROP TABLE IF EXISTS tmp\_bike\_status;

CREATE TABLE IF NOT EXISTS tmp\_bike\_status

as

SELECT a.\*

FROM (

select bike\_id

,max(start\_time) as start\_time

from trips

where start\_time<'${day} ${time}'

group by bike\_id

) b JOIN trips a ON a.bike\_id = b.bike\_id AND a.start\_time = b.start\_time;

-- 站点车辆存量信息

DROP TABLE IF EXISTS tmp\_station\_stock;

CREATE TABLE IF NOT EXISTS tmp\_station\_stock

as

select to\_station\_id

,to\_station\_name

,count(1) as cnt

, '${sdate}' as start\_hour

from tmp\_bike\_status

group by to\_station\_id,to\_station\_name

### 2）、站点需求量

**计算口径**：

当前时刻取历史4次的均值。（如当前时刻是周一八点，则往前推四周，取每周一的八点数据，然后求平均值）

**参数配置**：

state=$[yyyy-mm-dd\_hh24]

last7day=$[yyyy-mm-dd-7]

last14day=$[yyyy-mm-dd-14]

last21day=$[yyyy-mm-dd-21]

last28day=$[yyyy-mm-dd-28]

thishour=$[hh24]

**统计sql**:

DROP TABLE IF EXISTS tmp\_station\_need;

CREATE TABLE IF NOT EXISTS tmp\_station\_need

as

select from\_station\_id

, from\_station\_name

, ceil(count(1)/4) as cnt

, '${sdate}' as start\_hour

from trips

where substr(start\_time,1,10) in('${last7day}','${last14day}','${last21day}','${last28day}')

and substr(start\_time,12,2)='${thishour}'

group by from\_station\_id,from\_station\_name

### 3）、站点调度信息

**计算口径**：

可分配量=存量-需求量

**参数配置**：

state=$[yyyy-mm-dd\_hh24]

today=$[yyyy-mm-dd]

**统计sql**:

insert overwrite table bike\_control partition (pt)

select ROW\_NUMBER() OVER (ORDER BY allot\_cnt ),

'${sdate}',y.\*,getdate(),'${today}'

from (

select x.station\_id

,x.station\_name

,case when m.cnt is null then 0 else m.cnt end stock\_cnt

,case when n.cnt is null then 0 else n.cnt end avg\_cnt

,(case when m.cnt is null then 0 else m.cnt end)

-(case when n.cnt is null then 0 else n.cnt end) as allot\_cnt

from stations x

left outer join tmp\_station\_stock m on x.station\_id=m.to\_station\_id

left outer join tmp\_station\_need n on x.station\_id=n.from\_station\_id

) y