实习1

关系数据库创建和查询

理解概念

- 非结构化数据库
 - 一致性 → 最终一致性
 - 互联网上的某些数据,如微博关注数,并非需要实时一致性,只需在特定时间内达到一致性

OLTP

- 联机事务处理,如订票、银行转账、选课、成绩登记、微博转发、点赞等
- 通常涉及部分数据,对应insert、update和delete语句

OLAP

- 联机分析处理,如查询商品在不同地区或时间销售额、微博热榜、成绩统计分析等
- 通常涉及大量数据,对应select语句

关系创建 - 外键约束

• drop表再创建表问题:运行第一次没问题,运行第二次就出错,但是并不是每个表都是如此,station表有这种问题

```
In [3]: "Sol drop table if exists station:
        CREATE TABLE station (
             station_id smallint not null primary key,
            station name text,
            lat real.
            long real,
            dock count smallint,
            city text,
            installation_date date,
            zip code text
           1245
           1246
                        except BaseException as e:
        D:\py\python\envs\python27\lib\site-packages\sqlalchemy\engine\default.pyc in do execute(self, cursor, statem
        ent, parameters, context)
            550
                    def do execute (self, cursor, statement, parameters, context=None):
            551
        —> 552
                        cursor.execute(statement, parameters)
            553
            554
                    def do_execute_no_params(self, cursor, statement, context=None):
        InternalError: (psycopg2.InternalError) 错误: 无法删除 表 station 因为有其它对象倚赖它
        DETAIL: 在表 trip上的约束trip start station id fkey 倚赖干 表 station
        在表 trip上的约束trip_end_station_id_fkey 倚赖于 表 station
        HINT: 使用 DROP .. CASCADE 把倚赖对象一并删除.
         [SQL: drop table if exists station:]
         (Background on this error at: http://sqlalche.me/e/2j85)
```

数据插入 - 主键约束

• 数据重复插入问题

```
In [6]: Sign copy station from 'E://station.txt' delimiter ',';
         -> 1244
                                         cursor, statement, parameters, context
            1245
            1246
                         except BaseException as e:
         D:\py\python\envs\python27\lib\site-packages\sqlalchemy\engine\default.pyc in do_execute(self, cursor, statem
         ent, parameters, context)
             550
                     def do_execute(self, cursor, statement, parameters, context=None):
             551
         —> 552
                         cursor.execute(statement, parameters)
             553
             554
                     def do_execute_no_params(self, cursor, statement, context=None):
         IntegrityError: (psycopg2. IntegrityError) 错误: 重复键讳反唯一约束 "station pkey"
         DETAIL: 键值《(station id)=(2)》 已经存在
         CONTEXT: COPY station, line 1
         [SQL: copy station from 'E://station.txt' delimiter ',':]
         (Background on this error at: http://sqlalche.me/e/gkpj)
```

关系和数据在第一次运行后,就永久保存在数据库中,以后运行时,无需再次创建和插入数据

SQL

- 课堂练习2(简单/普通模式) → 实习1(普通/困难模式)

_ 看清楚题目要求,题意理解有问题可以在QQ群提问

- 建议首先使用with简单SQL语句构建临时关系,实现题目要求,再将with语句通过子查询嵌入到select/from/where

子句

start_time, timestamp start_station_name, text start_station_id, smallint end_time, timestamp end_station_name, text end_station_id, smallint bike_id, smallint		
id, integer not null duration, integer start_time, timestamp start_station_name, text start_station_id, smallint end_time, timestamp end_station_name, text end_station_id, smallint bike_id, smallint station_station_d dock_co city, text installati zip_code PRIMAR	Trip	[table]
duration, integer start_time, timestamp start_station_name, text start_station_id, smallint end_time, timestamp end_station_name, text end_station_id, smallint bike_id, smallint station_name, text installation zip_code, PRIMARY	id, integer not null	
start_station_name, text start_station_id, smallint end_time, timestamp end_station_name, text end_station_id, smallint bike_id, smallint latitude, re longitude, dock_cource city, text installatio zip_code,t PRIMARY	duration, integer	
start_station_name, text start_station_id, smallint end_time, timestamp end_station_name, text end_station_id, smallint bike_id, smallint latitude, re longitude, dock_cource city, text installatio zip_code,t PRIMARY	start time timesta	mp
start_station_id, smallint end_time, timestamp end_station_name, text end_station_id, smallint bike_id, smallint longitude, dock_cource city, text installation zip_code,t		
end_time, timestamp end_station_name, text end_station_id, smallint bike_id, smallint dock_count city, text installatio zip_code,t	start_station_nam	ie, text
end_station_name, text end_station_id, smallint bike_id, smallint city, text installatio zip_code,t PRIMARY	start_station_id, s	mallint
end_station_name, text end_station_id, smallint bike_id, smallint primary primary	end_time, timestam	ıp
end_station_id, smallint bike_id, smallint PRIMARY	end_station_name	e, text
bike_id, smallint PRIMARY	end_station_id, sn	nallint
PRIMARY	bike id, smallint	
TRIMARI_REF(R)		n
	111111111111111111111111111111111111111	.,

Weather [table] date, date not null max temp, real mean temp, real min_temp, real max_visibility_miles, real mean_visibility_miles, real min_visibility_miles, real max wind speed mph, real mean_wind_speed_mph, real max_gust_speed_mph, real cloud_cover, real events, text wind dir degrees, real zip_code, text not null PRIMARY_KEY(date, zip_code)

SELECT [ALL|DISTINCT]

<目标列表达式>[别名][, <目标列表达式>[别名]]...

FROM <表名或视图名>[别名]

[, <表名或视图名>[别名]]...

[WHERE <条件表达式>]

[GROUP BY <列名1>[, <列名2>]...

[HAVING <条件表达式>]]

[ORDER BY <列名1> [ASC|DESC]

[, <列名2> [ASC|DESC]]...];

• $\pi_{A1, A2, ..., An}$ ($\sigma_{condition}(R_1 \times R_2 \times ... \times R_n)$)

语义上的执行顺序:

- 1. FROM: 关系笛卡尔积
- 2. WHERE: 选择满足条件的行
- 3. GROUP BY: 根据属性分组
- 4. HAVING: 选择满足条件的组
- 5. SELECT: 投影需要的列
- 6. ORDER BY: 结果排序

- 0. 自行车位最多的站点
 - 最大值问题,SQL常见写法

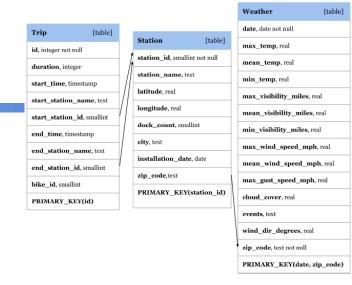
select station_id, dock_count
from station

where dock_count = (select max(dock_count) from station);

select station_id, dock_count
from station
where dock_count >= all(select dock_count from station);

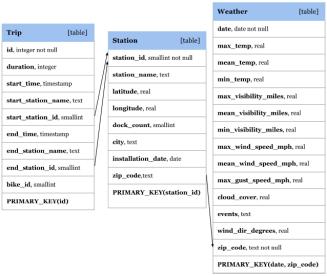
select station_id, dock_count
from station,

(select max(dock_count) as max_dock_count from station) as mt where dock_count = max_dock_count



- 1. 每个城市站点数
 - 不包含子查询,没有比这更简单了

select city, count(station_id) as number from station
group by city
order by number desc, city asc



- 2. 距离最近的站点对
 - 最小值问题,套用0题模版
 - 不能使用limit,函数使用,去重

```
Weather
                                                                                              [table]
                                                                    date, date not null
Trip
                                  Station
                                                                    max_temp, real
id, integer not null
                                  station_id, smallint not null
                                                                    mean_temp, real
duration, integer
                                  station name, text
                                                                    min_temp, real
start time, timestamp
                                  latitude, real
                                                                    max_visibility_miles, real
                                  longitude, real
                                                                    mean_visibility_miles, real
start_station_id, smallint
                                  dock count, smallin
                                                                    min_visibility_miles, real
                                                                    max_wind_speed_mph, real
end_station_name, text
                                  installation_date, date
                                                                    mean wind speed mph, rea
end station id, smallint
                                  zip_code,text
                                                                   max_gust_speed_mph, real
bike id, smallint
                                  PRIMARY KEY(station id)
                                                                    cloud_cover, real
PRIMARY_KEY(id)
                                                                    events, text
                                                                    wind_dir_degrees, real
                                                                    zip_code, text not null
                                                                    PRIMARY KEY(date, zip_code)
```

```
select S1.station id as station id A,
       S2.station id as station id B,
       dist(S1.lat, S1.long, S2.lat, S2.long) as distance
from station S1, station S2
where S1.station id < S2.station_id and
       dist(S1.lat, S1.long, S2.lat, S2.long) =
       (select min(dist(S3.lat, S3.long, S4.lat, S4.long))
       from station S3, station S4)
```

- 2. 距离最近的站点对
 - 最小值问题, 套用0题模版
 - _ 不能使用limit,函数使用,去重

```
Weather
                                                                                              [table]
                                                                    date, date not null
Trip
                                  Station
                                                                    max_temp, real
id, integer not null
                                  station_id, smallint not null
                                                                    mean_temp, real
duration, integer
                                  station name, text
                                                                    min_temp, real
start time, timestamp
                                  latitude, real
                                                                    max_visibility_miles, real
                                  longitude, real
                                                                    mean_visibility_miles, real
start_station_id, smallint
                                  dock count, smallin
                                                                    min_visibility_miles, real
                                                                    max_wind_speed_mph, real
end_station_name, text
                                  installation_date, date
                                                                    mean wind speed mph, rea
end station id, smallint
                                  zip_code,text
                                                                    max_gust_speed_mph, real
bike id, smallint
                                  PRIMARY KEY(station id)
                                                                    cloud_cover, real
PRIMARY_KEY(id)
                                                                    events, text
                                                                    wind_dir_degrees, real
                                                                    zip_code, text not null
                                                                    PRIMARY KEY(date, zip_code)
```

(select min(dist(S3.lat, S3.long, S4.lat, S4.long) as md) from station S3, station S4) as D

```
where S1.station_id < S2.station_id and dist(S1.lat, S1.long, S2.lat, S2.long) = md
```

- 2. 每个站点距离最近的站点
 - 每个站点之间的距离

select S1.station_id as A,

S2.station_id as B,

dist(S1.lat, S1.long, S2.lat, S2.long) as d

from Station S1, Station S2

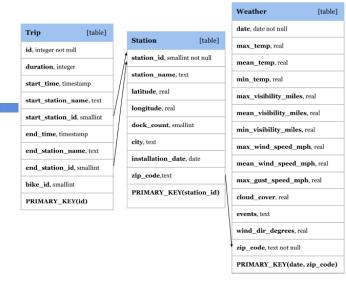
where S1.station_id <> S2.station_id

- 每个站点选择距离最近的站点

select A, B, d

from (...) T1

where d <= any(select d from (...) as T2 where T1.A = T2.A)



- 2. 每个站点距离最近的站点
 - 每个站点之间的距离 (A, B, d)
 - 每个站点选择距离最近的站点

```
select A, B, d
from (...) T1
```

where d <= any(select d from (...) as T2 where T1.A = T2.A)

去重(A, B, d)和(B, A, d)

select A, B, d

from (...) T1

where d <= any(select d from (...) as T2 where T2.A = T1.A) and (A < B or

d > any(select d from (...) as T3 where T3.A = T1.B))

```
Weather
                                                                                              [table]
                                                                    date, date not null
Trip
                                  Station
                                                                    max_temp, real
id, integer not null
                                  station_id, smallint not null
                                                                    mean_temp, real
duration, integer
                                  station name, text
                                                                    min_temp, real
start_time, timestamp
                                  latitude, real
                                                                    max_visibility_miles, real
start_station_name, text
                                  longitude, real
                                                                    mean_visibility_miles, real
start_station_id, smallint
                                                                    min_visibility_miles, real
                                                                    max_wind_speed_mph, real
end_station_name, text
                                  installation_date, date
                                                                    mean wind speed mph, rea
end_station_id, smallint
                                  zip_code,text
                                                                   max_gust_speed_mph, real
bike id, smallint
                                  PRIMARY KEY(station id)
                                                                    cloud cover, real
PRIMARY_KEY(id)
                                                                    wind_dir_degrees, real
                                                                    zip code, text not null
                                                                    PRIMARY KEY(date, zip_code)
```

2. 每个站点距离最近的站点select A, B, d

from (select S1.station_id as A, S2.station
 dist(S1.lat, S1.long, S2.lat, S2.long) as d
 from Station S1, Station S2
 where S1.station_id <> S2.station_id) T1

where d <= any(select dist(S3.lat, S3.long, S4.lat, S4.long)
from Station S3, Station S4
where S3.station_id = T1.A and S4.station_id <> T1.A)

Weather

max_temp, real

mean_temp, real

min_temp, real

max_visibility_miles, real

mean_visibility_miles, real

min_visibility_miles, real

max_wind_speed_mph, real

mean wind speed mph, rea

max_gust_speed_mph, real

wind_dir_degrees, real

zip_code, text not null

PRIMARY_KEY(date, zip_code)

cloud_cover, real

events, text

Station

station_id, smallint not null

station_name, text

latitude, real

longitude, real

zip_code,text

dock count, smallin

installation date, date

PRIMARY KEY(station id)

id, integer not null

start_time, timestamp

end_station_id, smalling

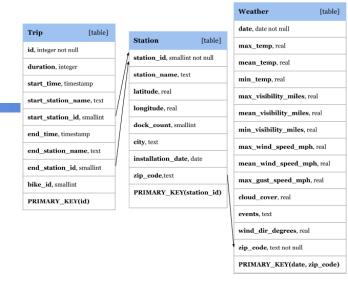
bike id, smallint

PRIMARY_KEY(id)

and (A < B or

d > any(select dist(S5.lat, S5.long, S6.lat, S6.long)
from Station S5, Station S6
where S5.station id = T1.B and S6.station id <> T1.B))

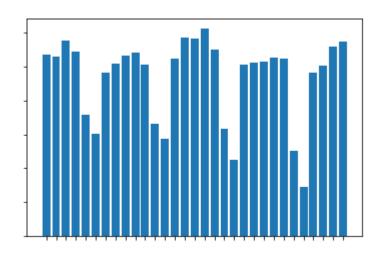
- 3. 2013年10月每天租车量
 - _ 和1题一样,没有比这更简单了
 - _ 时间判断和函数使用



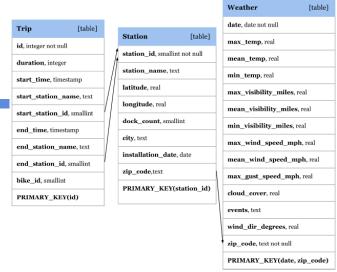
select date(start_time) as date, count(*) as number
from trip

where start_time > '2013-10-01' and start_time < '2013-11-1'

group by date order by date



- 4. 租车记录最多的20个站点对
 - _ 和1题一样,没有比这更简单了



select start_station_id, end_station_id, count(*) as trip_count
from trip

group by start_station_id, end_station_id
order by trip_count desc
limit 20

- 查询结果分析
 - 有的对称,有的不对称
 - 原因: 从站点A到站点B是下坡,而相反方向是上坡,租车人相 对减少; 旅游景点站点A-B-C,入口在站点A,出口在站点C;马 路两侧都有自行车站点,一侧借车,另一侧还车

- 5. 自行车#697的累积行驶时间
 - 自行车#697租车记录
 - select * from trip where bike_id = 697
 - _ 自行车#697还车时间为t时,累积行驶时间
 - select sum(duration) from trip
 - where bike_id = 697 and end_time <= t
 - 自行车#697所有还车时间时,计算累积行驶时间 select end_time,
 - (select sum(duration) from trip T2 where T2.bike_id = 697 and T2.end_time <= T1.end_time) as ctd

from trip T1
where bike_id = 697
order by end_time



• 选择语句

```
SELECT A_1, A_2, ..., A_n #3: what to return FROM R_1, R_2, ..., R_n #1: relations to query
```

WHERE condition #2: combine, filter relations

```
Answer = {}
for x_1 in R_1(SQL) do
for x_2 in R_2 do
.....

for x_n in R_n do
if conditions(x_1, ..., x_n) then (conditions = SQL)
A_1 = x_1.a_1
A_2 = \text{SQL}(x_1, ..., x_n) \text{ // 要求输出一个值}
Answer = Answer \cup {(A<sub>1</sub>, A<sub>2</sub>, ..., A<sub>n</sub>)}
return Answer
```

- 6. 每个城市最受欢迎的站点
 - 分组最大值问题
 - 不区分租车和还车站点

with visits as

((select start_station_id as sid from trip)

union all

(select end_station_id as sid from trip))

- 每个站点的使用次数

select sid, count(*) from visits group by sid

select city, station_name, count(*)

from station, visits

where station_id = sid

group by city, sid, station_name



思考:可以使用union?

- 6. 每个城市最受欢迎的站点
 - 分组最大值问题
 - _ 基于城市分组,使用次数最多的站点

select city, station_name, count(*)

from station S1, visits

where station_id = sid

group by city, sid, station_name

having count(*) >=

all(select count(*) from visits

where sid in (select station_id from station S2

where S2.city = S1.city)

group by sid)

Weather Trip Station max_temp, real id, integer not null station_id, smallint not null mean_temp, real duration, integer station_name, text min_temp, real start time, timestamp latitude, real max_visibility_miles, real longitude, real mean_visibility_miles, real start_station_id, smallint dock count, smallin min_visibility_miles, real max_wind_speed_mph, real end_station_name, text installation_date, date mean wind speed mph, rea end_station_id, smallint zip_code,text max_gust_speed_mph, real bike id, smallint PRIMARY KEY(station id) cloud_cover, real PRIMARY_KEY(id) wind_dir_degrees, real zip_code, text not null PRIMARY KEY(date, zip_code)

注意:可能存在多种 SQL查询方法,嵌套 查询 vs. Group By

order by city

该城市的使用次数最多的站点

• 6. 每个城市最受欢迎的站点

city	station_name	count
Mountain View	Mountain View Caltrain Station	13263
Palo Alto	Palo Alto Caltrain Station	3930
Redwood City	Redwood City Caltrain Station	2809
San Francisco	San Francisco Caltrain (Townsend at 4th)	112271
San Jose	San Jose Diridon Caltrain Station	18973

caltrain

查看此网页的中文翻译,请点击翻译此页

Board of Directors Advisory Committees Media Relations Government Affairs Social Media Doing Business Sustainability Statistics & Reports History Caltrain's ...

www.caltrain.com/ ▼ - 百度快照 - 评价

Schedules - 时间表

Pdf schedules - download and view...

Stations - 站

Caltrain serves dozens of stations...

Weekday Timetable - 平日的时间表

Weekday timetable weekend timetable...

System Map - 系统图

Caltrain zone map caltrain serves...

旧金山Caltrain火车站交通攻略 旧金山Caltrain火车站地址 旧金山...



2017年2月11日 - 旧金山Caltrain火车站交通攻略,穷游网为旅行者提供旧金山Caltrain火车站价格、简介、地址、电话、网址、交通线路、开放时间、周边信息,网友点评等,为穷游er制订旧金山...

place.qyer.com/poi/V2c... ▼ - 百度快照 - 281条评价

旧金山有火车站吗 百度知道



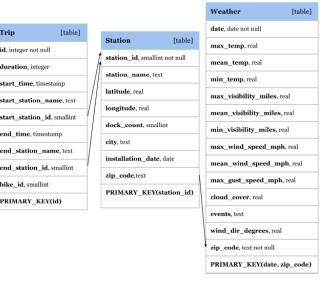
3个回答 - 最新回答: 2015年08月18日 - 5人觉得有用问题描述: 我只知道奥克兰有一个,旧金山市中心有吗

[专业] 答案:旧金山有火车站吗旧金山湾区有两套铁路系统,一套是 湾区地铁 简称BART,全称Bay Area Railway Train),主要是用来连接湾区北部几...

更多关于caltrain的问题>>

zhidao.baidu.com/link?... ▼ - 百度快照 - 评价

- 7. 每个站点当前的自行车数目
 - Trip只记录了已完成还车的租出记录
 - _ 假设每辆自行车至少被租出一次, 且当前所有车已归还
 - 自行车认为在最后一次还车站点,最后一次还车时间 select bike_id, max(end_time) as end_time from trip group by bike_id
 - 每辆自行车当前在哪个站点 select trip.bike_id, end_station_id from trip, (...) as foo where trip.bike_id = foo.bike_id and trip.end_time = foo.end_time



Trip

id, integer not null

duration, integer

start_time, timestamp

end_station_name, text

bike id, smallint

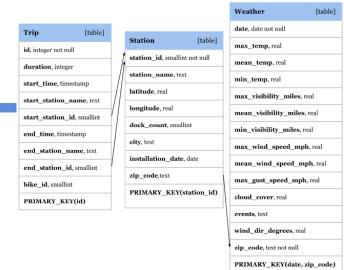
PRIMARY_KEY(id)

• 7. 每个站点当前的自行车数目

- 每个站点的自行车数目

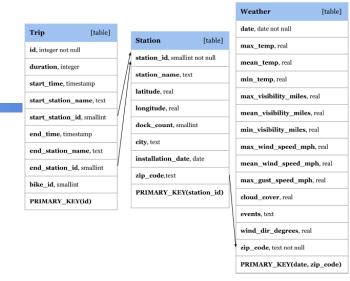
- 每辆自行车最后一次还车时间
- 每辆自行车当前在哪个站点
 select trip.bike_id, end_station_id
 from trip, (...) as foo
 where trip.bike_id = foo.bike_id and
 trip.end_time = foo.end_time
- select end_station_id, count(bike_id) as bike_count
 from (...) as temp
 group by end_station_id

order by bike_count desc, end_station_id asc



- 7. 每个站点当前的自行车数目
 - 每辆自行车最后一次还车时间
 - 每辆自行车当前在哪个站点
 - 每个站点的自行车数目

(select bike_id, max(end_time) as end_time from trip group by bike_id) as foo



- 7. 每个站点当前的自行车数目
 - 每辆自行车最后一次还车时间
 - 每辆自行车当前在哪个站点
 - 每个站点的自行车数目

select end_station_id as station_id,
 count(trip.bike_id) as bike_count

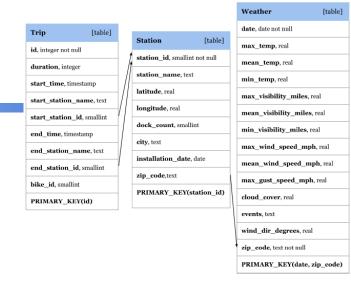
from trip natural join

(select bike_id, max(end_time) as end_time

from trip group by bike_id) as foo

group by end_station_id

order by bike_count desc, station_id asc



- 7. 每个站点当前的自行车数目
 - 每辆自行车最后一次还车时间
 - 每辆自行车当前在哪个站点
 - 每个站点的自行车数目
 - _ 当站点车位停满后,无法在该站点还车
 - _ Trip没有记录人为调度的自行车记录

station_id	dock_count
2	27
61	27
67	27
77	27

Trip

id, integer not null
duration, integer
start_time, timestamp

start_station_id, smallint

end_station_name, text end_station_id, smallint

bike_id, smallint
PRIMARY_KEY(id)

[table]

			Weather	[table]
			date, date not null	
	Station	[table]	max_temp, real	
A	station_id, smallint no	t null	mean_temp, real	
	station_name, text		min_temp, real	
$/\!\!/\!\!\!\perp$	latitude, real		max_visibility_miles,	real
	longitude, real		mean_visibility_miles	, real
	dock_count, smallint		min_visibility_miles,	real
	city, text		max_wind_speed_mp	h, real
	installation_date, date	е	mean_wind_speed_n	ph, real
	zip_code,text		max_gust_speed_mp	h, real
	PRIMARY_KEY(stati	on_id)	cloud_cover, real	
			events, text	
			wind_dir_degrees, rea	1
			zip_code, text not null	
			PRIMARY_KEY(date,	zip_code)

station_id	bike_count
69	50
70	45
50	26
55	24
61	23
6	21
2	19
54	18
3	17

- 8. 天气与租车关系
 - _ 不包含子查询,没有比这更简单了
 - 需要关联trip和weather, 通过date
 select lower(events), count(*) as number
 from trip, weather
 where date(start_time) = date
 group by lower(events)
 - 需要关联trip, weather和station, 通过station_id, zip_code select lower(events) as events, count(*) as number from weather, trip, station where date(start_time) = date and start_station_id = station_id and station.zip_code = weather.zip_code group by lower(events)

 思考: group by用events, 结果是否相同?

Weather

max_temp, real

mean_temp, real

min_temp, real

max_visibility_miles, real

mean_visibility_miles, real min_visibility_miles, real

max_wind_speed_mph, real

mean wind speed mph, rea

max_gust_speed_mph, real

wind_dir_degrees, real

zip_code, text not null

PRIMARY_KEY(date, zip_code)

cloud_cover, real

Station

station_id, smallint not null

station_name, text

latitude, real

longitude, real

zip_code,text

installation_date, date

PRIMARY KEY(station id)

id, integer not null

duration, integer

start time, timestamp

end_station_id, smallint

bike id. smallint

PRIMARY KEY(id)

- 8. 天气与租车关系

select lower(events) as events, count(*) as number

from weather, trip, station

where date(start_time) = date and

start_station_id = station_id and

station.zip_code = weather.zip_code

group by lower(events)

4	1
27	
\sim \square	ν

- 当天气为rain-thunderstorm时,选择租车的可能性最小?
- 在rain时选择租车的可能性大于在fog时选择租车的可能性?
- 不能用绝对数量代替平均数量或概率得出结论
 - 城市犯罪人数越多越不安全?省市GDP越高人民越富裕?...

					Weather	[table]
Trip	[table]				date, date not null	
id, integer not n	null	Stati		[table]	max_temp, real	
duration, integ	ger	statio	on_id, smallint	not null	mean_temp, real	
start_time, tim	nestamp	statio	on_name, text		min_temp, real	
start_station	name, text	latitu	ıde, real		max_visibility_mi	les, real
start_station	/	longi	itude, real		mean_visibility_n	niles, real
end time, time		dock	_count, smallin	t	min_visibility_mi	les, real
end_station_		city, t	text		max_wind_speed	_mph, real
end station		insta	llation_date, d	late	mean_wind_speed	d_mph, real
bike id, smalli		zip_code,text		max_gust_speed_mph, real		
PRIMARY_KI		PRIM	MARY_KEY(sta	ation_id)	cloud_cover, real	
	()				events, text	
') as number				_ \	wind_dir_degrees	, real
				٦ ١	zip_code, text not n	ull
					PRIMARY_KEY(d	ate, zip_code)

events

fog-rain

rain-thunderstorm

fog

rain

None

number

43676

6877

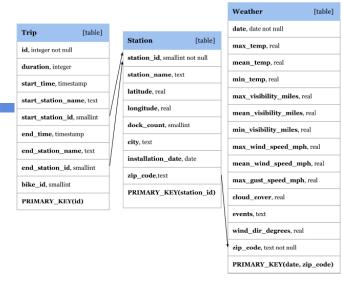
71613

546318

1475

- 8. 天气与租车关系

 - 不同天气的天数
 select lower(events) e1, count(*) n1
 from weather
 group by lower(events)
 - 不同天气的租车数量
 select lower(events) e2, count(*) n2
 from weather, trip, station
 where date(start_time) = date and
 start_station_id = station_id and
 station.zip_code = weather.zip_code
 group by lower(events)



- 8. 天气与租车关系

 - _ 不同天气的平均租车数量(self-loop)

select e1, round(n2 * 1.0 / n1, 2) from

(select lower(events) e1, count(*) n1

from weather group by lower(events)) as A

(select lower(events) e2, count(*) n2

from weather, trip, station

where date(start_time) = date and start_station_id =

group by lower(events)) as B

where e1 = e2

思考:为什么没有None(NULL)的结果?

						Weather	[table]								
Trip	[table]					date, date not null									
id, integer not	null		Station	[table]	П	max_temp, real									
duration, inte	ger	A	station_id, smal	lint not null		mean_temp, real									
		station_name, text		Н	min_temp, real										
start_time, ti	mestamp	//	latitude, real			max_visibility_miles, real									
start_station	_name, text	1	longitude, real												
start_station	_id, smallint		dock count, smallint			mean_visibility_mi	les, real								
end_time, tin	nestamp				H	min_visibility_mile	s, real								
end_station_	name, text										city, text		H	max_wind_speed_	mph, real
end_station	id. smallint		installation_da	te, date		mean_wind_speed	_mph, real								
bike id, small			zip_code,text		١	max_gust_speed_n	nph, real								
			PRIMARY_KEY(station_id)			cloud cover, real									
PRIMARY_K	EY(id)				1	events, text									
					1										
					1	wind_dir_degrees,	real								
					1	zip_code, text not nu	11								
						PRIMARY_KEY(da	te, zip_code)								

fog

rain

fog-rain

rain-thunderstorm

number

389.96

404.53

183.62

491.67

- 8. 天气与租车关系
 - 加上NULL之后,有何问题?
 - 如何进一步修改分析?

events	number
fog	389.96
fog-rain	404.53
rain	183.62
rain-thunderstorm	491.67
None	173.82

Trip	[table]		
id interes act well			Station
id, integer not null		t	station_id,
duration, integer		/1	
start_time, times	tamp	//	station_na
		III	latitude, rea
start_station_na	ime, text	///	longitude, r
start_station_id	, smallint		
end_time, timest	amp	П	dock_coun
	-		city, text
end_station_na	me, text		installation
end_station_id,	smallint		
bike_id, smallint			zip_code,te
			PRIMARY_
PRIMARY_KEY	(id)		

	Weather [tab				
	date, date not null				
[table]	max_temp, real				
, smallint not null	mean_temp, real				
ame, text	min_temp, real				
al	max_visibility_miles, real				
real	mean_visibility_miles, real				
ıt, smallint	min_visibility_miles, real				
	max_wind_speed_mph, real				
n_date, date	mean_wind_speed_mph, rea				
ext	max_gust_speed_mph, real				
_KEY(station_id)	cloud_cover, real				
	events, text				
	wind_dir_degrees, real				
,	zip_code, text not null				

PRIMARY_KEY(date, zip_code)

[table]

- 9.到达San Jose所有站点的自行车
 - 一 方法1: 除法 $r \div s = \pi_{(R-S)}(r) \pi_{(R-S)}[(\pi_{(R-S)}(r) \times s) r]$
 - 方法2: 数自行车到达的站点数目

select count(bike_id)

from station,

(select bike_id, start_station_id as sid from trip union

select bike_id, end_station_id as sid from trip) as temp

where station_id = sid and city = 'San Jose' group by bike_id

having count(distinct station_id) =

(select count(*) from station where city = 'San Jose')

