

Bike_Sharing

Asher Kirshtein

“Count the number of stations in each city.”

```
sqlite> SELECT city, COUNT(*) AS num_stations  
...> FROM station  
...> GROUP BY city  
...> ORDER BY num_stations ASC, city ASC;  
Palo Alto|5  
Mountain View|7  
Redwood City|7  
San Jose|16  
San Francisco|35  
sqlite>
```

“Find the percentage of bike trips made in each city.”

```
sqlite> SELECT station.city, ROUND(CAST(COUNT(DISTINCT trip.id) AS REAL) /  
...> (SELECT COUNT(DISTINCT id) FROM trip) , 4) AS percentage  
...> FROM trip  
...> JOIN station ON trip.start_station_id = station.station_id OR trip.end_station_id = station.station_id  
...> GROUP BY station.city  
...> ORDER BY percentage DESC, station.city ASC;  
San Francisco|0.9011  
San Jose|0.0566  
Mountain View|0.0278  
Palo Alto|0.0109  
Redwood City|0.0052  
sqlite>
```

“Which stations are the most popular for bike trips?”

```
sqlite> SELECT s.city, s.station_name, COUNT(DISTINCT t.id) AS visits
...> FROM station s
...> JOIN trip t ON s.station_id = t.start_station_id OR s.station_id = t.end_station_id
...> GROUP BY s.city, s.station_id
...> HAVING visits = (SELECT MAX(visits) FROM (
...>                     SELECT s.city, s.station_name, COUNT(DISTINCT t.id) AS visits
...>                     FROM station s
...>                     JOIN trip t ON s.station_id = t.start_station_id OR s.station_id = t.end_station_id
...>                     GROUP BY s.city, s.station_id
...>                 ) AS sub
...>                 WHERE sub.city = s.city)
...> ORDER BY s.city ASC;
Mountain View|Mountain View Caltrain Station|12735
Palo Alto|Palo Alto Caltrain Station|3534
Redwood City|Redwood City Caltrain Station|2654
San Francisco|San Francisco Caltrain (Townsend at 4th)|111738
San Jose|San Jose Diridon Caltrain Station|18782
sqlite>
```

“Find all the bikes that have been to more than one city”

```
sqlite> SELECT bike_id, COUNT(DISTINCT city) AS num_cities
...> FROM (
...>     SELECT DISTINCT bike_id, city
...>     FROM (
...>         SELECT bike_id, start_station_id, end_station_id
...>         FROM trip
...>         GROUP BY bike_id, start_station_id, end_station_id
...>     ) AS bike_trips
...>     JOIN station ON bike_trips.start_station_id = station.station_id OR bike_trips.end_station_id = station.station_id
...> ) AS bike_cities
...> GROUP BY bike_id
...> HAVING num_cities > 1
...> ORDER BY num_cities DESC, bike_id ASC
...> LIMIT 20;
15|5
25|5
27|5
31|5
43|5
51|5
56|5
59|5
64|5
69|5
76|5
90|5
94|5
116|5
119|5
123|5
136|5
137|5
158|5
164|5
sqlite>
```

“Are bikes being (incorrectly) recorded as being used in two trips concurrently?”

```
sqlite> SELECT DISTINCT
...>   T1.bike_id,
...>   T1.id as trip_id_1,
...>   T1.start_time as start_time_1,
...>   T1.end_time as end_time_1,
...>   T2.id as trip_id_2,
...>   T2.start_time as start_time_2,
...>   T2.end_time as end_time_2
...> FROM
...>   trip T1
...> JOIN trip T2 ON T1.bike_id = T2.bike_id AND T1.id < T2.id
...> WHERE
...>   T1.bike_id BETWEEN 100 AND 200
...>   AND T1.start_time < T2.end_time
...>   AND T2.start_time < T1.end_time
...> ORDER BY
...>   T1.bike_id,
...>   T1.id,
...>   T2.id;
144|815060|2015-06-19 21:26:00|2015-06-19 22:17:00|815073|2015-06-19 22:10:00|2015-06-19 22:17:00
158|576536|2014-12-15 15:05:00|2014-12-15 23:11:00|576591|2014-12-15 16:07:00|2014-12-15 16:17:00
158|576536|2014-12-15 15:05:00|2014-12-15 23:11:00|576604|2014-12-15 16:28:00|2014-12-15 16:40:00
sqlite>
```

“Find the top ten days that have the highest average bike utilization.”

```
sqlite> SELECT strftime('%Y-%m-%d', trip.start_time) AS date, ROUND(SUM(strftime('%s', trip.end_time) - strftime('%s', trip.start_time)) * 1.0 / (COUNT(DISTINCT trip.bike_id) * 60), 4) AS avg_utilization
...> FROM trip
...> WHERE trip.bike_id <= 100
...> GROUP BY date
...> ORDER BY avg_utilization DESC
...> LIMIT 10;
2015-02-08|3354.6667
2014-12-14|594.8182
2014-08-23|593.2105
2015-05-24|584.6316
2015-02-14|576.15
2015-07-01|470.68
2013-12-31|381.0
2014-07-12|376.45
2014-07-15|344.0556
2015-02-21|343.0
sqlite>
```