**US Accidents Query Solutions**

***By Somendra, Pablo and Balaji***

## **Query 1**

WITH sev1 as(SELECT DISTINCT count(id) as Acc\_Sev1, state

FROM us\_accident\_v3

WHERE severity = 1

GROUP BY state),

sev2 as(SELECT DISTINCT count(id) as Acc\_Sev2, state

FROM us\_accident\_v3

WHERE severity = 2

GROUP BY state),

sev3 as(SELECT DISTINCT count(id) as Acc\_Sev3, state

FROM us\_accident\_v3

WHERE severity = 3

GROUP BY state),

sev4 as(SELECT DISTINCT count(id) as Acc\_Sev4, state

FROM us\_accident\_v3

WHERE severity = 4

GROUP BY state)

SELECT sev1.state, Acc\_Sev1, Acc\_Sev2, Acc\_Sev3, Acc\_Sev4

FROM sev1, sev2, sev3, sev4

WHERE sev1.state = sev2.state AND

sev2.state = sev3.state AND

sev3.state = sev4.state

ORDER BY sev1.state ASC;

A screenshot of a social media post

Description automatically generated

## **Query 3**

SELECT DISTINCT

(SELECT avg(date\_diff('second',start\_time,end\_time))

FROM us\_accident\_v3

WHERE severity = 4) AS Avg\_dur\_sev\_4,

(SELECT stddev(date\_diff('second',start\_time,end\_time))

FROM us\_accident\_v3

WHERE severity = 4) AS Stddev\_dur\_sev\_4,

(SELECT max(date\_diff('second',start\_time,end\_time))

FROM us\_accident\_v3

WHERE severity = 4) AS Max\_dur\_sev\_4,

(SELECT avg(date\_diff('second',start\_time,end\_time))

FROM us\_accident\_v3

WHERE severity = 3) AS Avg\_dur\_sev\_3,

(SELECT stddev(date\_diff('second',start\_time,end\_time))

FROM us\_accident\_v3

WHERE severity = 3) AS Stddev\_dur\_sev\_3,

(SELECT max(date\_diff('second',start\_time,end\_time))

FROM us\_accident\_v3

WHERE severity = 3) AS Max\_dur\_sev\_3,

(SELECT avg(date\_diff('second',start\_time,end\_time))

FROM us\_accident\_v3

WHERE severity = 2) AS Avg\_dur\_sev\_2,

(SELECT stddev(date\_diff('second',start\_time,end\_time))

FROM us\_accident\_v3

WHERE severity = 2) AS Stddev\_dur\_sev\_2,

(SELECT max(date\_diff('second',start\_time,end\_time))

FROM us\_accident\_v3

WHERE severity = 2) AS Max\_dur\_sev\_2,

(SELECT avg(date\_diff('second',start\_time,end\_time))

FROM us\_accident\_v3

WHERE severity = 1) AS Avg\_dur\_sev\_1,

(SELECT stddev(date\_diff('hour',start\_time,end\_time))

FROM us\_accident\_v3

WHERE severity = 1) AS Stddev\_dur\_sev\_1,

(SELECT max(date\_diff('second',start\_time,end\_time))

FROM us\_accident\_v3

WHERE severity = 1) AS Max\_dur\_sev\_1

FROM us\_accident\_v3



## **Query 4**

with cte1 as

(select date\_format(start\_time, '%Y-%m-%d') AS date,

weather\_condition,

case

when weather\_condition = 'Clear' then 'GOOD'

when weather\_condition = 'Cloudy' then 'GOOD'

when weather\_condition = 'Cloudy / Windy' then 'GOOD'

when weather\_condition = 'Drizzle' then 'GOOD'

when weather\_condition = 'Drizzle / Windy' then 'GOOD'

when weather\_condition = 'Fair' then 'GOOD'

when weather\_condition = 'Fair / Windy' then 'GOOD'

when weather\_condition = 'Haze' then 'GOOD'

when weather\_condition = 'Light Blowing Snow' then 'GOOD'

when weather\_condition = 'Light Drizzle' then 'GOOD'

when weather\_condition = 'Light Drizzle / Windy' then 'GOOD'

when weather\_condition = 'Light Fog' then 'GOOD'

when weather\_condition = 'Light Freezing Drizzle' then 'GOOD'

when weather\_condition = 'Light Freezing Fog' then 'GOOD'

when weather\_condition = 'Light Freezing Rain' then 'GOOD'

when weather\_condition = 'Light Freezing Rain / Windy' then 'GOOD'

when weather\_condition = 'Light Hail' then 'GOOD'

when weather\_condition = 'Light Haze' then 'GOOD'

when weather\_condition = 'Light Ice Pellets' then 'GOOD'

when weather\_condition = 'Light Rain' then 'GOOD'

when weather\_condition = 'Light Rain / Windy' then 'GOOD'

when weather\_condition = 'Light Rain Shower' then 'GOOD'

when weather\_condition = 'Light Rain Shower / Windy' then 'GOOD'

when weather\_condition = 'Light Rain Showers' then 'GOOD'

when weather\_condition = 'Light Sleet' then 'GOOD'

when weather\_condition = 'Light Snow' then 'GOOD'

when weather\_condition = 'Light Snow and Sleet' then 'GOOD'

when weather\_condition = 'Light Snow Grains' then 'GOOD'

when weather\_condition = 'Light Snow Shower' then 'GOOD'

when weather\_condition = 'Light Snow Showers' then 'GOOD'

when weather\_condition = 'Light Thunderstthen' then 'GOOD'

when weather\_condition = 'Light Thunderstthen' then 'GOOD'

when weather\_condition = 'Mist' then 'GOOD'

when weather\_condition = 'Mostly Cloudy' then 'GOOD'

when weather\_condition = 'Mostly Cloudy / Windy' then 'GOOD'

when weather\_condition = 'N/A Precipitation' then 'GOOD'

when weather\_condition = 'Overcast' then 'GOOD'

when weather\_condition = 'Partial Fog' then 'GOOD'

when weather\_condition = 'Partial Fog / Windy' then 'GOOD'

when weather\_condition = 'Partly Cloudy' then 'GOOD'

when weather\_condition = 'Partly Cloudy / Windy' then 'GOOD'

when weather\_condition = 'Patches of Fog' then 'GOOD'

when weather\_condition = 'Scattered Clouds' then 'GOOD'

when weather\_condition = 'Shallow Fog' then 'GOOD'

when weather\_condition = 'Showers in the Vicinity' then 'GOOD'

when weather\_condition = 'Small Hail' then 'GOOD'

when weather\_condition = 'Thunder in the Vicinity' then 'GOOD'

else 'BAD' end as weather\_type

, severity

from us\_accident\_v3),

cte2 as

(select \*,

case

when severity = 1 and weather\_type = 'GOOD' then 's1\_G'

when severity = 1 and weather\_type = 'BAD' then 's1\_B'

when severity = 2 and weather\_type = 'GOOD' then 's2\_G'

when severity = 2 and weather\_type = 'BAD' then 's2\_B'

when severity = 3 and weather\_type = 'GOOD' then 's3\_G'

when severity = 3 and weather\_type = 'BAD' then 's3\_B'

when severity = 4 and weather\_type = 'GOOD' then 's4\_G'

when severity = 4 and weather\_type = 'BAD' then 's4\_B'

else 'N/A' end as count\_G\_B

from cte1),

cte3 as(

select date,

count\_G\_B,

count(\*) as totalNumber

from cte2

group by date,count\_G\_B

order by date,count\_G\_B asc),

cte4 as(

select a.date,

count\_G\_B,

a.totalNumber,

cast(a.totalNumber as double)\*100/b.tot\_accident\_good as Acc\_Percent

from cte3 a

inner join

(select date,

count(\*) as tot\_accident\_good

from cte2

where weather\_type = 'GOOD'

group by date) b

on a.date = b.date

where count\_G\_B in ('s1\_G', 's2\_G', 's3\_G', 's4\_G')),

cte5 as (

select a.date,

count\_G\_B,

a.totalNumber,

cast(a.totalNumber as double)\*100/b.tot\_accident\_bad as Acc\_Percent

from cte3 a

inner join

(select date, count(\*) as tot\_accident\_bad from cte2

where weather\_type = 'BAD'

group by date) b

on a.date = b.date where count\_G\_B in ('s1\_B', 's2\_B', 's3\_B', 's4\_B')),

cte6 as (

select \* from cte4

union all

select \* from cte5)

select date, q1['s1\_G'] as Sev\_1\_perc\_GOOD\_weather,

q2['s2\_G'] as Sev\_2\_perc\_GOOD\_weather,

q3['s3\_G'] as Sev\_3\_perc\_GOOD\_weather,

q4['s4\_G'] as Sev\_4\_perc\_GOOD\_weather,

q5['s1\_B'] as Sev\_1\_perc\_BAD\_weather,

q6['s2\_B'] as Sev\_2\_perc\_BAD\_weather,

q7['s3\_B'] as Sev\_3\_perc\_BAD\_weather,

q8['s4\_B'] as Sev\_4\_perc\_BAD\_weather

from (

select date, map\_agg(count\_G\_B, Acc\_Percent) q1,

map\_agg(count\_G\_B, Acc\_Percent) q2,

map\_agg(count\_G\_B, Acc\_Percent) q3,

map\_agg(count\_G\_B, Acc\_Percent) q4,

map\_agg(count\_G\_B, Acc\_Percent) q5,

map\_agg(count\_G\_B, Acc\_Percent) q6,

map\_agg(count\_G\_B, Acc\_Percent) q7,

map\_agg(count\_G\_B, Acc\_Percent) q8

from cte6

group by date

A close up of a piece of paper

Description automatically generated order by date)

## **Query 5**

**temperature**

select date, temp\_range\_in\_f, severity, weather\_condn, count(\*) as cnt

from

(select date(start\_time) as date,

case

when temperature between -80 and -71 then '-80 to -71'

when temperature between -70 and -61 then '-70 to -61'

when temperature between -60 and -51 then '-60 to -51'

when temperature between -50 and -41 then '-50 to -41'

when temperature between -40 and -31 then '-40 to -31'

when temperature between -30 and -21 then '-30 to -21'

when temperature between -20 and -11 then '-20 to -11'

when temperature between -10 and -1 then '-10 to -01'

when temperature between 0 and 9 then '0 to 9'

when temperature between 10 and 19 then '10 to 19'

when temperature between 20 and 29 then '20 to 29'

when temperature between 30 and 39 then '30 to 39'

when temperature between 40 and 49 then '40 to 49'

when temperature between 50 and 59 then '50 to 59'

when temperature between 60 and 69 then '60 to 69'

when temperature between 70 and 79 then '70 to 79'

when temperature between 80 and 89 then '80 to 89'

when temperature between 90 and 99 then '90 to 99'

when temperature between 100 and 109 then '100 to 109'

when temperature between 110 and 119 then '110 to 119'

when temperature between 120 and 129 then '120 to 129'

when temperature between 130 and 139 then '130 to 139'

when temperature between 140 and 149 then '140 to 149'

when temperature between 150 and 159 then '150 to 159'

when temperature between 160 and 169 then '160 to 169'

when temperature between 170 and 179 then '170 to 179'

else 'temperature unknown'

end as temp\_range\_in\_f,

case

when weather\_condition = 'Clear' then 'good'

when weather\_condition like 'Cloudy%' then 'good'

when weather\_condition = 'Drizzle' then 'good'

when weather\_condition = 'Drizzle / Windy' then 'good'

when weather\_condition like 'Fair%' then 'good'

when weather\_condition = 'Haze' then 'good'

when weather\_condition like 'Light%' then 'good'

when weather\_condition = 'Mist' then 'good'

when weather\_condition like 'Mostly%' then 'good'

when weather\_condition = 'N/A Precipitation' then 'good'

when weather\_condition = 'Overcast' then 'good'

when weather\_condition like 'Partial%' then 'good'

when weather\_condition like 'Partly%' then 'good'

when weather\_condition = 'Patches of Fog' then 'good'

when weather\_condition = 'Scattered Clouds' then 'good'

when weather\_condition = 'Shallow Fog' then 'good'

when weather\_condition = 'Showers in the Vicinity' then 'good'

when weather\_condition = 'Small Hail' then 'good'

when weather\_condition = 'Thunder in the Vicinity' then 'good'

else 'bad'

end as weather\_condn, severity

from us\_accident\_v3

where year(start\_time) = 2019 and month(start\_time) = 12 and day(start\_time) = 31

order by month(start\_time), date(start\_time))

group by weather\_condn, severity, temp\_range\_in\_f, date

order by temp\_range\_in\_f, severity desc, weather\_condn, cnt desc;



**visibility**

**wind-speed**

**Crossing**

select date, range, severity, weather\_condn, count(\*) as cnt

from

(select date(start\_time) as date,

case

when crossing = false then '0'

when crossing = true then '1'

end as range,

case

when weather\_condition = 'Clear' then 'good'

when weather\_condition like 'Cloudy%' then 'good'

when weather\_condition = 'Drizzle' then 'good'

when weather\_condition = 'Drizzle / Windy' then 'good'

when weather\_condition like 'Fair%' then 'good'

when weather\_condition = 'Haze' then 'good'

when weather\_condition like 'Light%' then 'good'

when weather\_condition = 'Mist' then 'good'

when weather\_condition like 'Mostly%' then 'good'

when weather\_condition = 'N/A Precipitation' then 'good'

when weather\_condition = 'Overcast' then 'good'

when weather\_condition like 'Partial%' then 'good'

when weather\_condition like 'Partly%' then 'good'

when weather\_condition = 'Patches of Fog' then 'good'

when weather\_condition = 'Scattered Clouds' then 'good'

when weather\_condition = 'Shallow Fog' then 'good'

when weather\_condition = 'Showers in the Vicinity' then 'good'

when weather\_condition = 'Small Hail' then 'good'

when weather\_condition = 'Thunder in the Vicinity' then 'good'

else 'bad'

end as weather\_condn, severity

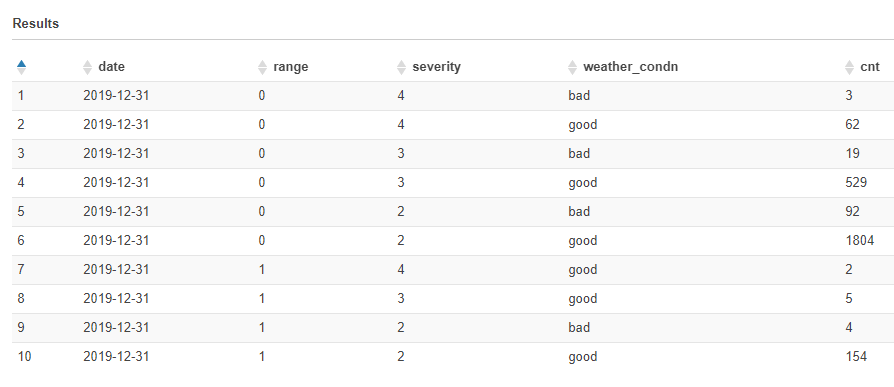
from us\_accident\_v3

where year(start\_time) = 2019 and month(start\_time) = 12 and day(start\_time) = 31

order by month(start\_time), date(start\_time))

group by weather\_condn, severity, range, date

order by range, severity desc, weather\_condn, cnt desc;



**Junction**

select date, range, severity, weather\_condn, count(\*) as cnt

from

(select date(start\_time) as date,

case

when junction = false then '0'

when junction = true then '1'

end as range,

case

when weather\_condition = 'Clear' then 'good'

when weather\_condition like 'Cloudy%' then 'good'

when weather\_condition = 'Drizzle' then 'good'

when weather\_condition = 'Drizzle / Windy' then 'good'

when weather\_condition like 'Fair%' then 'good'

when weather\_condition = 'Haze' then 'good'

when weather\_condition like 'Light%' then 'good'

when weather\_condition = 'Mist' then 'good'

when weather\_condition like 'Mostly%' then 'good'

when weather\_condition = 'N/A Precipitation' then 'good'

when weather\_condition = 'Overcast' then 'good'

when weather\_condition like 'Partial%' then 'good'

when weather\_condition like 'Partly%' then 'good'

when weather\_condition = 'Patches of Fog' then 'good'

when weather\_condition = 'Scattered Clouds' then 'good'

when weather\_condition = 'Shallow Fog' then 'good'

when weather\_condition = 'Showers in the Vicinity' then 'good'

when weather\_condition = 'Small Hail' then 'good'

when weather\_condition = 'Thunder in the Vicinity' then 'good'

else 'bad'

end as weather\_condn, severity

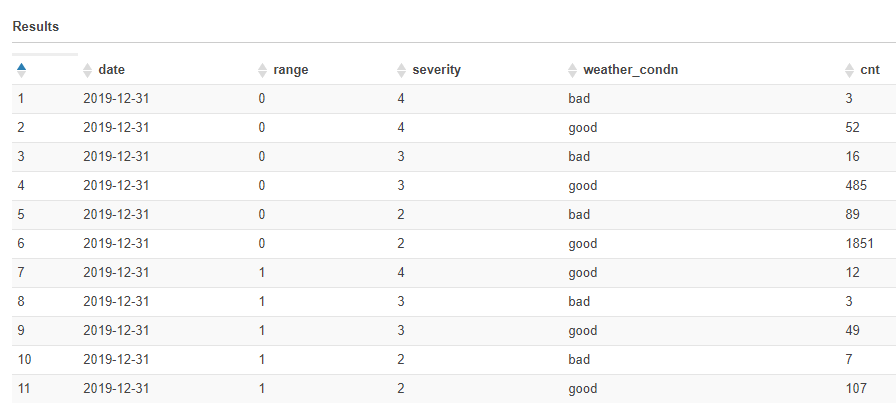
from us\_accident\_v3

where year(start\_time) = 2019 and month(start\_time) = 12 and day(start\_time) = 31

order by month(start\_time), date(start\_time))

group by weather\_condn, severity, range, date

order by range, severity desc, weather\_condn, cnt desc;



**Traffic\_Signal**

select date, range, severity, weather\_condn, count(\*) as cnt

from

(select date(start\_time) as date,

case

when traffic\_signal = false then '0'

when traffic\_signal = true then '1'

end as range,

case

when weather\_condition = 'Clear' then 'good'

when weather\_condition like 'Cloudy%' then 'good'

when weather\_condition = 'Drizzle' then 'good'

when weather\_condition = 'Drizzle / Windy' then 'good'

when weather\_condition like 'Fair%' then 'good'

when weather\_condition = 'Haze' then 'good'

when weather\_condition like 'Light%' then 'good'

when weather\_condition = 'Mist' then 'good'

when weather\_condition like 'Mostly%' then 'good'

when weather\_condition = 'N/A Precipitation' then 'good'

when weather\_condition = 'Overcast' then 'good'

when weather\_condition like 'Partial%' then 'good'

when weather\_condition like 'Partly%' then 'good'

when weather\_condition = 'Patches of Fog' then 'good'

when weather\_condition = 'Scattered Clouds' then 'good'

when weather\_condition = 'Shallow Fog' then 'good'

when weather\_condition = 'Showers in the Vicinity' then 'good'

when weather\_condition = 'Small Hail' then 'good'

when weather\_condition = 'Thunder in the Vicinity' then 'good'

else 'bad'

end as weather\_condn, severity

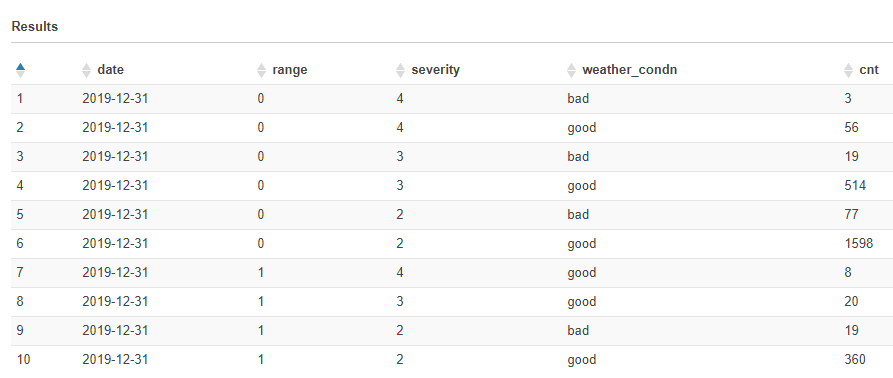
from us\_accident\_v3

where year(start\_time) = 2019 and month(start\_time) = 12 and day(start\_time) = 31

order by month(start\_time), date(start\_time))

group by weather\_condn, severity, range, date

order by range, severity desc, weather\_condn, cnt desc;



## Query 6

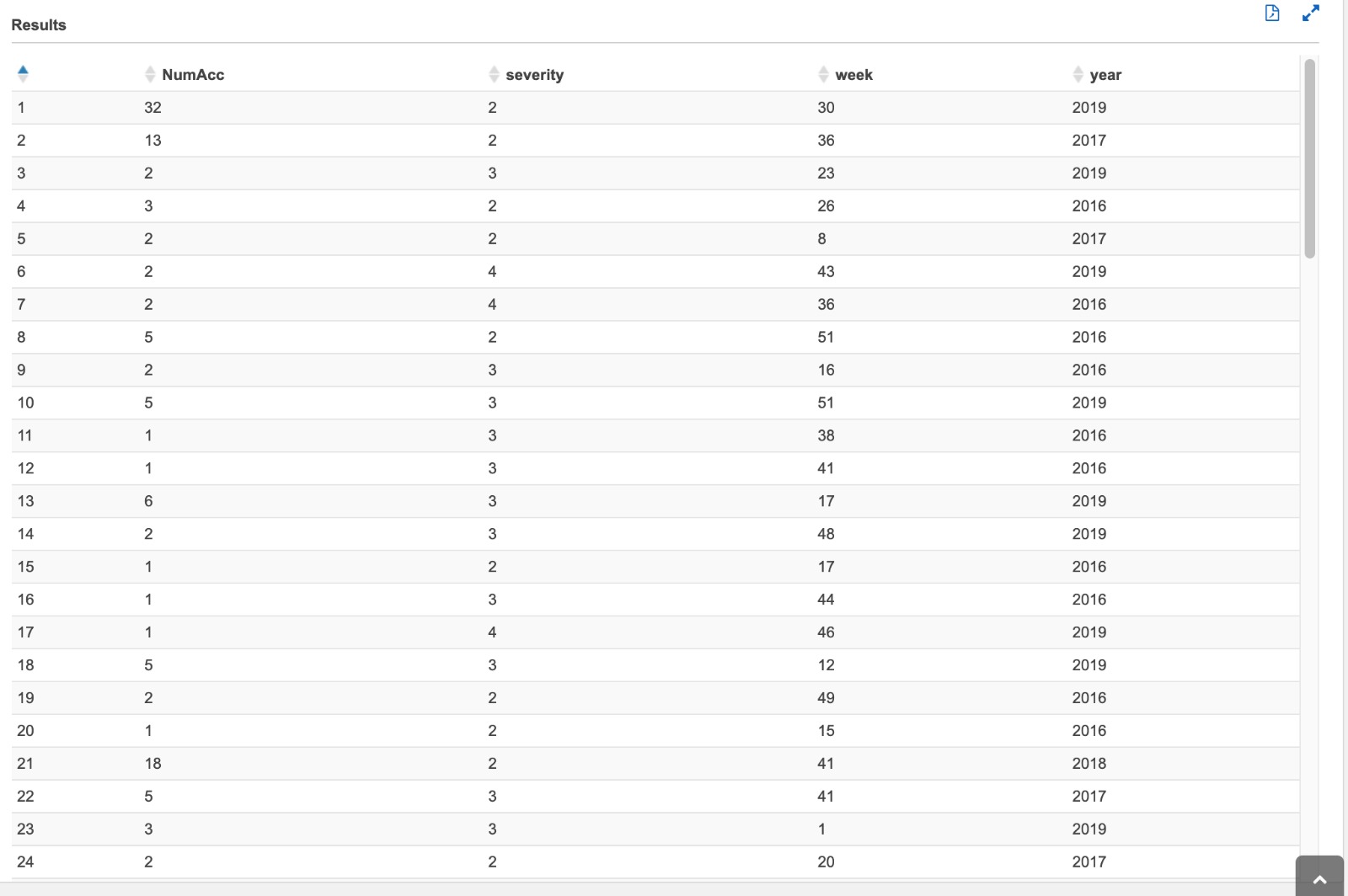
select count(id) as NumAcc, severity, extract(week from start\_time) as week, extract(year from start\_time) as year

from timezone\_table us\_accident\_v3

where (extract(year from start\_time) > 2015 and extract(month from start\_time)>1) or

(extract(year from start\_time) > 2016)

group by severity, extract(week from start\_time), extract(year from start\_time)



## Query 7

SELECT severity,

avg(distance) AS AvgDis,

stddev(distance) AS StdDevDis,

max(distance) AS MaxDis

FROM us\_accident\_v3

GROUP BY severity

A screenshot of a cell phone

Description automatically generatedORDER BY severity

## Query 10

select Concept, sum(Frequency) as Freq from (with cte1 AS

(SELECT b.slno,

b.pattrn,

b.concpt,

cardinality(regexp\_extract\_all(lower(a.n\_description),

lower(b.pattrn))) count

FROM us\_accidents\_dec\_19\_database.base\_corpus3 a

CROSS JOIN us\_accidents\_dec\_19\_database.concept7 b

ORDER BY slno)

SELECT DISTINCT slno AS Serial\_No,

concpt AS Concept,

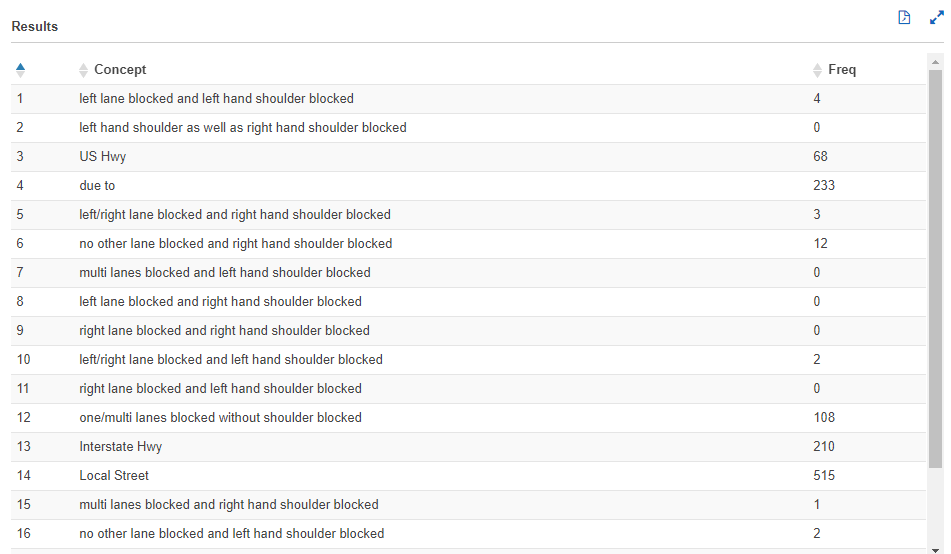
pattrn AS Pattern,

sum(count)

OVER (partition by pattrn) AS Frequency

FROM cte1

ORDER BY slno) group by Concept



## Query 13

SELECT DISTINCT state,

date,

avg(sum\_precip)

OVER (partition by date, state) AS avg\_precip2

FROM

(SELECT DISTINCT state,

zipcode,

date,

sum(avg\_precip)

OVER (partition by date, zipcode, state) AS sum\_precip

FROM

(SELECT DISTINCT state,

zipcode,

date(start\_time) AS date,

hour(start\_time) AS hour,

minute(start\_time) AS minute,

count(\*) AS count,

avg(precipitation) AS avg\_precip

FROM us\_accidents\_dec\_19\_database.us\_accident\_v3

GROUP BY state, zipcode, date(start\_time), hour(start\_time), minute(start\_time)

ORDER BY count desc)

ORDER BY state, zipcode, date)

WHERE state = 'NY'

OR state = 'NJ'

OR state = 'PA'or state = 'IN'

OR state = 'WY'

ORDER BY state, date;

