**TIME SERIES**

INDEX

create index idutilserv on time\_series.utilization(server\_id, event\_time);

explain select server\_id, avg(cpu\_utilization)

from time\_series.utilization

where event\_time between '2019-03-05' and '2019-03-06'

group by server\_id;

EXPLAIN in SELECT

explain select location\_id, avg(temp\_celcius)

from time\_series.location\_temp

group by location\_id

CREATE VIEW IN time series

create view time\_series.v\_util as

(select \*, server\_id % 10 dept\_id from time\_series.utilization)

LEAD

select dept\_id, server\_id, cpu\_utilization,

lead(cpu\_utilization, 2) over (partition by dept\_id

order by cpu\_utilization DESC)

from time\_series.v\_util

where event\_time between '2019-03-05' and '2019-03-06'

LAG

select dept\_id, server\_id, cpu\_utilization,

lag(cpu\_utilization, 2) over (partition by dept\_id

order by cpu\_utilization DESC)

from time\_series.v\_util

where event\_time between '2019-03-05' and '2019-03-06'

RANK

select dept\_id, server\_id, cpu\_utilization,

rank() over (partition by dept\_id

order by cpu\_utilization DESC)

from time\_series.v\_util

where event\_time between '2019-03-05' and '2019-03-06'

PERCENT\_RANK

select dept\_id, server\_id, cpu\_utilization,

percent\_rank() over (partition by dept\_id

order by cpu\_utilization DESC)

from time\_series.v\_util

where event\_time between '2019-03-05' and '2019-03-06'

COMMON TABLE EXPRESSIONS AND RECURSION

with daily\_avg\_temp as

(select date\_trunc('day', event\_time)event\_date,

avg(temp\_celcius)avg\_temp

  from time\_series.location\_temp

  group by date\_trunc('day',event\_time)

 )

 select event\_date, avg\_temp

 from daily\_avg\_temp

OVER PARTITION IN TS

select server\_id, cpu\_utilization, avg(cpu\_utilization)

over (partition by server\_id)

from time\_series.utilization

where event\_time between '2019-03-05' and '2019-03-06'

PREVIOUS DAY COMPARISION

with daily\_avg\_temp as

(select date\_trunc('day',event\_time) event\_date,

 avg(temp\_celcius) avg\_temp

 from time\_series.location\_temp

 group by date\_trunc('day',event\_time)

 )

 select event\_date, avg\_temp,

 (select avg\_temp

  from daily\_avg\_temp dat2

  where dat2.event\_date= dat1.event\_date - interval '1' day)

 from daily\_avg\_temp dat1

SLIDING WINDOW-- MOVING AVERAGE

select event\_time, server\_id,

avg(cpu\_utilization) over(order by event\_time rows between 12 preceding and current row) as hourly\_cpu\_util

from time\_series.utilization

WEIGHTED MOVING AVERAGE- Calc based on weights specified

with daily\_avg\_temp as

(select date\_trunc('day',event\_time) event\_date,

 avg(temp\_celcius) avg\_temp

 from time\_series.location\_temp

 group by date\_trunc('day',event\_time)

 )

select event\_date, round(avg\_temp,2),

(select round(avg\_temp,2)\*0.5

from daily\_avg\_temp dat2

where date\_trunc('day',dat1.event\_date)-interval '1' day=

date\_trunc('day',dat2.event\_date))+

(select round(avg\_temp,2)\*0.333

from daily\_avg\_temp dat3

where date\_trunc('day',dat1.event\_date)-interval '2' day=

date\_trunc('day',dat3.event\_date))+

  (select round(avg\_temp,2)\*0.167

from daily\_avg\_temp dat4

where date\_trunc('day',dat1.event\_date)-interval '3' day=

date\_trunc('day',dat4.event\_date))

from daily\_avg\_temp dat1

TIME SERIES FORECAST

/\* y=mx+c where m: slope; x: input val; c: intercept; y: predicted value \*/

/\* free\_memory>> y axis, cpu\_utilization>> x axis \*/

select

regr\_slope(free\_memory, cpu\_utilization) m,

regr\_intercept(free\_memory, cpu\_utilization) b

from time\_series.utilization

where event\_time between '2019-03-05' and '2019-03-06'

PREDICTED VALUE  
/\* y=mx+c where m: slope; x: input val; c: intercept; y: predicted value \*/

select

regr\_slope(free\_memory, cpu\_utilization) \* 0.65,

regr\_intercept(free\_memory, cpu\_utilization) predicted\_value

from time\_series.utilization

where event\_time between '2019-03-05' and '2019-03-06'