/\*

-----DOCUMENTATION

10/1/2020 - added big\_deal flag that flags deal over 250k

10/1/2020 - added closedate\_push\_count that counts # of times deal's close date has been pushed looking to see if certain deals push x number of times before they close irregardless whether pushes occur within the quarter or outside of quarter

1/13/2021 - added simple\_vertical to determine prediction of deals by vertical

3/1/2021 - added logic to see if deal was pushed in quarter or pushed out of quarter (then won/lost)

\*/

--drop view analytics.company\_deals\_for\_pred

--create view analytics.company\_deals\_for\_pred as

with stage\_dates as(

select distinct

sfdc\_opportunity\_id

, max(entered\_prefunnel)::date as prefunnel\_date

, max(entered\_qualify)::date as qualify\_date

, max(entered\_verify)::date as verify\_date

, max(entered\_position)::date as position\_date

, max(entered\_propose)::date as propose\_date

, max(entered\_selected)::date as selected\_date

, case when sum(forward\_count) is null then 0 else sum(forward\_count) end as forward\_total\_count

, case when sum(backward\_count) is null then 0 else sum(backward\_count) end as backward\_total\_count

, case when sum(no\_change\_count) is null then 0 else sum(no\_change\_count) end as nochange\_total\_count

from (

select distinct

sfdc\_opportunity\_id

, case when left(new\_stage\_sorted,1) like '0' then change\_date else null end as entered\_prefunnel

, case when left(new\_stage\_sorted,1) like '1' then change\_date else null end as entered\_qualify

, case when left(new\_stage\_sorted,1) like '2' then change\_date else null end as entered\_verify

, case when left(new\_stage\_sorted,1) like '3' then change\_date else null end as entered\_position

, case when left(new\_stage\_sorted,1) like '4' then change\_date else null end as entered\_propose

, case when left(new\_stage\_sorted,1) like '5' then change\_date else null end as entered\_selected

, case when stage\_direction like 'Forward' and stage\_direction is null then 0 else count(stage\_direction) end as forward\_count

, case when stage\_direction like 'Backward' and stage\_direction is null then 0 else count(stage\_direction) end as backward\_count

, case when stage\_direction like 'No Change' and stage\_direction is null then 0 else count(stage\_direction) end as no\_change\_count

from (

select distinct

sfdc\_opportunity\_id

, change\_date

, old\_stage\_sorted

, new\_stage\_sorted

, new\_stage

, stage\_direction

, rank() over (partition by sfdc\_opportunity\_id,new\_stage\_sorted order by change\_date desc) as rnk

from analytics.sfdc\_stages\_view

order by 4 asc, 2 desc

)

where rnk = 1

group by sfdc\_opportunity\_id, new\_stage\_sorted, change\_date, stage\_direction

)

group by sfdc\_opportunity\_id, forward\_count

),

deal\_change\_dates as (

select distinct

sfdc\_opportunity\_id

, min(prev\_close\_dates)::date as first\_close\_date

, max(prev\_close\_dates)::date as last\_close\_date

, min(prev\_close\_date1)::date as prev\_close\_date1

, min(prev\_close\_date2)::date as prev\_close\_date2

, min(prev\_close\_date3)::date as prev\_close\_date3

, min(prev\_close\_date4)::date as prev\_close\_date4

, min(prev\_close\_date5)::date as prev\_close\_date5

, min(prev\_close\_date6)::date as prev\_close\_date6

, min(prev\_close\_date7)::date as prev\_close\_date7

, min(prev\_close\_date8)::date as prev\_close\_date8

, min(prev\_close\_date9)::date as prev\_close\_date9

, min(prev\_close\_date10)::date as prev\_close\_date10

, min(prev\_close\_date11)::date as prev\_close\_date11

, min(prev\_close\_date12)::date as prev\_close\_date12

, min(prev\_close\_date13)::date as prev\_close\_date13

, min(prev\_close\_date14)::date as prev\_close\_date14

, min(prev\_close\_date15)::date as prev\_close\_date15

, sum(push\_counter) as pushed\_out\_of\_quarter\_count

, sum(backwards\_push) as pushed\_prior\_quarter\_count

, sum(forwards\_push) as pushes\_future\_quarter\_count

, count(prev\_close\_dates) as closedate\_push\_count -- counts # of times close\_date was pushed irregardless if it was pushed in or out of quarter

, closedate\_push\_count - pushed\_out\_of\_quarter\_count as stayed\_in\_quarter\_count

from (

select

sfdc\_opportunity\_id

, change\_date::date as change\_date

, old\_value::date as old\_value

, new\_value::date as new\_value

, rank() over (partition by sfdc\_opportunity\_id order by change\_date asc) as close\_date\_rank

, case when lag(old\_value,1) over (partition by sfdc\_opportunity\_id order by change\_date asc) is null then old\_value

else lag(old\_value,0) over (partition by sfdc\_opportunity\_id order by change\_date asc) end as prev\_close\_dates

, case

when date\_trunc('quarter', new\_value::timestamp) > date\_trunc('quarter', old\_value::timestamp) then 'Pushed Out Of Quarter'

when date\_trunc('quarter', new\_value::timestamp) = date\_trunc('quarter', old\_value::timestamp) then 'Stayed In Quarter'

when date\_trunc('quarter', new\_value::timestamp) < date\_trunc('quarter', old\_value::timestamp) then 'Pushed Prior Quarter'

end as push\_status

, case when push\_status like '%Pushed%' then 1 else 0 end as push\_counter

, case when push\_status like '%Pushed Prior%' then 1 else 0 end as backwards\_push

, case when push\_status like '%Pushed Out%' then 1 else 0 end as forwards\_push

, case when close\_date\_rank = 1 then prev\_close\_dates end as prev\_close\_date1

, case when close\_date\_rank = 2 then prev\_close\_dates end as prev\_close\_date2

, case when close\_date\_rank = 3 then prev\_close\_dates end as prev\_close\_date3

, case when close\_date\_rank = 4 then prev\_close\_dates end as prev\_close\_date4

, case when close\_date\_rank = 5 then prev\_close\_dates end as prev\_close\_date5

, case when close\_date\_rank = 6 then prev\_close\_dates end as prev\_close\_date6

, case when close\_date\_rank = 7 then prev\_close\_dates end as prev\_close\_date7

, case when close\_date\_rank = 8 then prev\_close\_dates end as prev\_close\_date8

, case when close\_date\_rank = 9 then prev\_close\_dates end as prev\_close\_date9

, case when close\_date\_rank = 10 then prev\_close\_dates end as prev\_close\_date10

, case when close\_date\_rank = 11 then prev\_close\_dates end as prev\_close\_date11

, case when close\_date\_rank = 12 then prev\_close\_dates end as prev\_close\_date12

, case when close\_date\_rank = 13 then prev\_close\_dates end as prev\_close\_date13

, case when close\_date\_rank = 14 then prev\_close\_dates end as prev\_close\_date14

, case when close\_date\_rank = 15 then prev\_close\_dates end as prev\_close\_date15

from analytics.dim\_sfdc\_opp\_field\_history

where field = 'Close\_Date'

order by 1,2,3 asc

)

group by 1

),

company\_deals as (

select distinct

o.sfdc\_opportunity\_id

, o.created\_date::date as created\_date

, o.close\_date

, o.simple\_stage

, a.simple\_vertical

, case when f.acv\_usd >= 250000 then 1 else 0 end as big\_deal\_flag

, case

when o.last\_entered\_shopping\_date is null then o.first\_entered\_shopping\_date

else o.last\_entered\_shopping\_date end as shopping\_date

, case

when shopping\_date is null then 'Never In Shopping'

else 'In Shopping'

end as shopping\_phase\_status

, case when shopping\_phase\_status = 'In Shopping' then 1 else 0 end as shopping\_phase\_flag

, date\_trunc('quarter',created\_date)::date as created\_beg\_of\_qtr

, (dateadd('quarter',1,date\_trunc('quarter',o.created\_date))-1)::date as created\_end\_of\_qtr

, date\_trunc('quarter',shopping\_date)::date as shop\_beg\_of\_qtr

, (dateadd('quarter',1,date\_trunc('quarter',shopping\_date))-1)::date as shop\_end\_of\_qtr

, date\_trunc('quarter',o.close\_date)::date as closed\_beg\_of\_qtr

, (dateadd('quarter',1,date\_trunc('quarter',o.close\_date))-1)::date as closed\_end\_of\_qtr

-- marketing experiment opportunity role info

, r.product\_count

, r.campaign\_influence\_count

, r.total\_opp\_roles

, r.sponsor

, r.champion

, r.decision\_maker

, r.economic\_decision\_maker

, r.econ\_buyer

, r.influencer

, r.technical\_buyer

, r.subject\_matter\_expert

, r.evaluator

, r.business\_user

, r.detractor

, case when r.exec\_sponsor >= 1 then 1 else 0 end as sponsor\_flag

, case when r.champion >= 1 then 1 else 0 end as champion\_flag

, case when r.decision\_maker >= 1 then 1 else 0 end as decision\_maker\_flag

, case when r.economic\_decision\_maker >= 1 then 1 else 0 end as economic\_decision\_maker\_flag

, case when r.econ\_buyer >= 1 then 1 else 0 end as econ\_buyer\_flag

, case when r.influencer >= 1 then 1 else 0 end as influencer\_flag

, case when r.technical\_buyer >= 1 then 1 else 0 end as technical\_buyer\_flag

, case when r.subject\_matter\_expert >= 1 then 1 else 0 end as subject\_matter\_expert\_flag

, case when r.evaluator >= 1 then 1 else 0 end as evaluator\_flag

, case when r.business\_user >= 1 then 1 else 0 end as business\_user\_flag

, case when r.detractor >= 1 then 1 else 0 end as detractor\_flag

, f.acv\_usd

from analytics.sfdc\_opportunity o

left join analytics.sfdc\_opportunity\_roles r on (r.sfdc\_opportunity\_id = o.sfdc\_opportunity\_id)

left join analytics.fact\_sfdc\_opportunity f on (f.sfdc\_opportunity\_id = o.sfdc\_opportunity\_id)

left join analytics.dim\_sfdc\_account a on (a.sfdc\_account\_id = o.sfdc\_account\_id)

--where o.close\_date >= '2017-07-01' and o.close\_date < '2020-12-31' --training data

where o.close\_date >= '2021-01-01' and o.close\_date < '2021-03-31' --test data

--where o.close\_date >= '2017-07-01' and o.close\_date < '2021-01-01' --PCA training/test data Unsupervised Test

and o.simple\_type in ('Type 1','Type 2')

and o.simple\_stage in ('Won', 'Lost')

and a.simple\_vertical is not null

)

select distinct

d.\*

, c.closedate\_push\_count

, c.first\_close\_date

, c.pushed\_out\_of\_quarter\_count

, c.stayed\_in\_quarter\_count

, c.pushed\_prior\_quarter\_count

, c.pushes\_future\_quarter\_count

, case

when d.close\_date < c.last\_close\_date then d.close\_date

when d.close\_date > c.last\_close\_date then c.last\_close\_date

end last\_close\_date

, c.prev\_close\_date1

, c.prev\_close\_date2

, c.prev\_close\_date3

, c.prev\_close\_date4

, c.prev\_close\_date5

, s.prefunnel\_date

, s.qualify\_date

, s.verify\_date

, s.position\_date

, s.propose\_date

, s.selected\_date

, case when nvl(datediff('day', d.created\_date, c.prev\_close\_date1),0)::float is null then 0 else abs(nvl(datediff('day', d.created\_date, c.prev\_close\_date1),0))::float end as create\_to\_prev\_close1

, case when nvl(datediff('day', d.created\_date, c.prev\_close\_date2),0)::float is null then 0 else abs(nvl(datediff('day', d.created\_date, c.prev\_close\_date2),0))::float end as create\_to\_prev\_close2

, case when nvl(datediff('day', d.created\_date, c.prev\_close\_date3),0)::float is null then 0 else abs(nvl(datediff('day', d.created\_date, c.prev\_close\_date3),0))::float end as create\_to\_prev\_close3

, case when nvl(datediff('day', d.created\_date, c.prev\_close\_date4),0)::float is null then 0 else abs(nvl(datediff('day', d.created\_date, c.prev\_close\_date4),0))::float end as create\_to\_prev\_close4

, case when nvl(datediff('day', d.created\_date, c.prev\_close\_date5),0)::float is null then 0 else abs(nvl(datediff('day', d.created\_date, c.prev\_close\_date5),0))::float end as create\_to\_prev\_close5

, case when nvl(datediff('day', d.created\_date, c.last\_close\_date ),0)::float is null then 0 else abs(nvl(datediff('day', d.created\_date, c.last\_close\_date ),0))::float end as create\_to\_prev\_closex

, case when nvl(datediff('day', c.prev\_close\_date1, c.prev\_close\_date2),0)::float is null then 0 else abs(nvl(datediff('day', c.prev\_close\_date1, c.prev\_close\_date2),0))::float end as days\_between\_prev\_close\_1\_2

, case when nvl(datediff('day', c.prev\_close\_date2, c.prev\_close\_date3),0)::float is null then 0 else abs(nvl(datediff('day', c.prev\_close\_date2, c.prev\_close\_date3),0))::float end as days\_between\_prev\_close\_2\_3

, case when nvl(datediff('day', c.prev\_close\_date3, c.prev\_close\_date4),0)::float is null then 0 else abs(nvl(datediff('day', c.prev\_close\_date3, c.prev\_close\_date4),0))::float end as days\_between\_prev\_close\_3\_4

, case when nvl(datediff('day', c.prev\_close\_date4, c.prev\_close\_date5),0)::float is null then 0 else abs(nvl(datediff('day', c.prev\_close\_date4, c.prev\_close\_date5),0))::float end as days\_between\_prev\_close\_4\_5

, case when nvl(datediff('day', c.prev\_close\_date5, c.last\_close\_date ),0)::float is null then 0 else abs(nvl(datediff('day', c.prev\_close\_date5, c.last\_close\_date ),0))::float end as days\_between\_prev\_close\_5\_x

, case when datediff('day',d.created\_date, d.close\_date) is null then 0 else abs(datediff('day',d.created\_date, d.close\_date)) end as create\_to\_final\_close

, case when nvl(datediff('day', c.first\_close\_date, c.last\_close\_date),0)::float is null then 0 else abs(nvl(datediff('day', c.first\_close\_date, c.last\_close\_date),0))::float end as days\_between\_first\_last\_close

, case when nvl(datediff('day', c.first\_close\_date, d.close\_date),0)::float is null then 0 else abs(nvl(datediff('day', c.first\_close\_date, d.close\_date),0))::float end as days\_between\_first\_final\_close

, case when nvl(datediff('day', d.created\_beg\_of\_qtr, d.created\_date),0)::float is null then 0 else nvl(datediff('day',d.created\_beg\_of\_qtr, d.created\_date),0)::float end as days\_created\_from\_qtr\_beg

, case when nvl(datediff('day', d.created\_date,d.created\_end\_of\_qtr),0)::float is null then 9 else nvl(datediff('day',d.created\_date,d.created\_end\_of\_qtr),0)::float end as days\_created\_from\_qtr\_end

, case when nvl(datediff('day', d.shop\_beg\_of\_qtr,d.shopping\_date),0)::float is null then 0 else nvl(datediff('day',d.shop\_beg\_of\_qtr,d.shopping\_date),0)::float end as days\_shop\_from\_qtr\_beg

, case when nvl(datediff('day', d.shopping\_date,d.shop\_end\_of\_qtr),0)::float is null then 0 else nvl(datediff('day',d.shopping\_date,d.shop\_end\_of\_qtr),0)::float end as days\_shop\_from\_qtr\_end

, case when nvl(datediff('day',d.created\_date,d.shopping\_date),0)::float is null then 0 else nvl(datediff('day',d.created\_date,d.shopping\_date),0)::float end as create\_to\_shop

, case when nvl(datediff('day',shopping\_date,d.close\_date),0)::float is null then 0 else nvl(datediff('day',shopping\_date,d.close\_date),0)::float end as shop\_to\_close

, case when datediff('day',d.created\_date, prefunnel\_date) is null then 0 else datediff('day',d.created\_date,prefunnel\_date) end as create\_to\_prefunnel

, case when datediff('day',d.created\_date,qualify\_date) is null then 0 else datediff('day',d.created\_date,qualify\_date) end as create\_to\_qualify

, case when datediff('day',d.created\_date,verify\_date) is null then 0 else datediff('day',d.created\_date,verify\_date) end as create\_to\_verify

, case when datediff('day',d.created\_date,position\_date) is null then 0 else datediff('day',d.created\_date,position\_date) end as create\_to\_position

, case when datediff('day',d.created\_date,propose\_date) is null then 0 else datediff('day',d.created\_date,propose\_date) end as create\_to\_propose

, case when datediff('day',d.created\_date,selected\_date) is null then 0 else datediff('day',d.created\_date,selected\_date) end as create\_to\_selected

, case when datediff('day',d.created\_date,d.close\_date) is null then 0 else datediff('day',d.created\_date,d.close\_date) end as create\_to\_close

, case when datediff('day',prefunnel\_date,qualify\_date) is null then 0 else datediff('day',prefunnel\_date,qualify\_date) end as prefunnel\_to\_qualify

, case when datediff('day',prefunnel\_date,verify\_date) is null then 0 else datediff('day',prefunnel\_date,verify\_date) end as prefunnel\_to\_verify

, case when datediff('day',prefunnel\_date,position\_date) is null then 0 else datediff('day',prefunnel\_date,position\_date) end as prefunnel\_to\_position

, case when datediff('day',prefunnel\_date,propose\_date) is null then 0 else datediff('day',prefunnel\_date,propose\_date) end as prefunnel\_to\_propose

, case when datediff('day',prefunnel\_date,selected\_date) is null then 0 else datediff('day',prefunnel\_date,selected\_date) end as prefunnel\_to\_selected

, case when datediff('day',prefunnel\_date,d.close\_date) is null then 0 else datediff('day',prefunnel\_date,d.close\_date) end as prefunnel\_to\_close

, case when datediff('day',qualify\_date,verify\_date) is null then 0 else datediff('day',qualify\_date,verify\_date) end as qualify\_to\_verify

, case when datediff('day',qualify\_date,position\_date) is null then 0 else datediff('day',qualify\_date,position\_date) end as qualify\_to\_position

, case when datediff('day',qualify\_date,propose\_date) is null then 0 else datediff('day',qualify\_date,propose\_date) end as qualify\_to\_propose

, case when datediff('day',qualify\_date,selected\_date) is null then 0 else datediff('day',qualify\_date,selected\_date) end as qualify\_to\_selected

, case when datediff('day',qualify\_date,d.close\_date) is null then 0 else datediff('day',qualify\_date,d.close\_date) end as qualify\_to\_close

, case when datediff('day',verify\_date,position\_date) is null then 0 else datediff('day',verify\_date,position\_date) end as verify\_to\_position

, case when datediff('day',verify\_date,propose\_date) is null then 0 else datediff('day',verify\_date,propose\_date) end as verify\_to\_propose

, case when datediff('day',verify\_date,selected\_date) is null then 0 else datediff('day',verify\_date,selected\_date) end as verify\_to\_selected

, case when datediff('day',verify\_date,d.close\_date) is null then 0 else datediff('day',verify\_date,d.close\_date) end as verify\_to\_close

, case when datediff('day',position\_date,propose\_date) is null then 0 else datediff('day',propose\_date,selected\_date) end as position\_to\_propose

, case when datediff('day',position\_date,selected\_date) is null then 0 else datediff('day',propose\_date,selected\_date) end as position\_to\_selected

, case when datediff('day',position\_date,d.close\_date) is null then 0 else datediff('day',propose\_date,selected\_date) end as position\_to\_close

, case when datediff('day',propose\_date,selected\_date) is null then 0 else datediff('day',propose\_date,selected\_date) end as propose\_to\_selected

, case when datediff('day',propose\_date,d.close\_date) is null then 0 else datediff('day',propose\_date,d.close\_date) end as propose\_to\_close

, case when datediff('day',selected\_date,d.close\_date) is null then 0 else datediff('day',selected\_date,d.close\_date) end as selected\_to\_close

, case when prefunnel\_date is not null then 1 else 0 end as prefunnel\_count

, case when qualify\_date is not null then 1 else 0 end as qualify\_count

, case when position\_date is not null then 1 else 0 end as position\_count

, case when verify\_date is not null then 1 else 0 end as verify\_count

, case when propose\_date is not null then 1 else 0 end as propose\_count

, case when selected\_date is not null then 1 else 0 end as selected\_count

, (prefunnel\_count + qualify\_count + position\_count + verify\_count + propose\_count + selected\_count) as total\_stage\_count

, case when total\_stage\_count = 0 then 0 else create\_to\_close/total\_stage\_count end as cycle\_per\_stage

, s.forward\_total\_count

, s.backward\_total\_count

, s.nochange\_total\_count

, case

when acv\_usd between 0 and 50000 then 1

when acv\_usd between 50000 and 100000 then 2

when acv\_usd between 100000 and 250000 then 3

when acv\_usd between 250000 and 500000 then 4

when acv\_usd between 500000 and 1000000 then 5

when acv\_usd between 1000000 and 6000000 then 6

end as deal\_size

, case

when d.simple\_stage like 'Lost' then 0

when d.simple\_stage like 'Won' then 1

end as exp\_outcome

from company\_deals d

left join stage\_dates s on (s.sfdc\_opportunity\_id = d.sfdc\_opportunity\_id)

left join deal\_change\_dates c on (c.sfdc\_opportunity\_id = d.sfdc\_opportunity\_id)

where acv\_usd > 0

and simple\_vertical = 'Legal'