DROP VIEW nvrpt.finance\_deck;

CREATE OR REPLACE VIEW nvrpt.finance\_deck

(

ads\_process\_type,

analyst,

badge\_color,

supervisor\_id,

supervisor\_login\_name,

week,

month,

quarter,

year,

reporting\_year,

reporting\_week,

country,

site,

cost\_center,

priority,

queue\_type,

task\_data\_language,

transformation,

root\_transformation,

transformation\_type,

task\_queue\_type,

device\_environment,

ads\_service,

order\_name,

service\_order\_id,

customer\_grp\_value,

customer,

vertical\_or\_sub\_initiative,

device\_or\_initiative,

task\_queue\_id,

task\_queue\_name,

sla\_planned\_flg,

convention,

owner,

owner\_id,

aim,

reason\_code,

rolling\_month,

rolling\_week,

rolling\_qtr,

rolling\_year,

auto\_verified\_volume,

manual\_verified\_volume,

auto\_processed\_volume,

verified\_volume,

weekly\_staff\_hrs,

weekly\_allocated\_hrs,

total\_working\_hrs,

op2\_volume,

target\_volume,

committed\_volume,

ingested\_count,

discard\_count,

auto\_skipped\_count,

overturned\_count,

processed\_volume,

production\_hours,

sla\_volume,

parent\_work\_type,

work\_type,

workflow\_app,

base\_verify\_workflow\_app,

core\_non\_core\_workflow\_app,

average\_da\_hourly\_wage,

opex\_total\_cph\_atat\_hours\_denominator,

opex\_total\_cph\_all\_hours\_denominator,

opex\_labor\_cph\_atat\_hours\_denominator,

opex\_labor\_cph\_all\_hours\_denominator,

op2\_total\_cph\_atat\_hours\_denominator,

op2\_total\_cph\_all\_hours\_denominator,

op2\_labor\_cph\_atat\_hours\_denominator,

op2\_labor\_cph\_all\_hours\_denominator,

quarterly\_total\_cph\_atat\_hours\_denominator,

quarterly\_total\_cph\_all\_hours\_denominator,

quarterly\_labor\_cph\_atat\_hours\_denominator,

quarterly\_labor\_cph\_all\_hours\_denominator,

normalized\_opex\_total\_cph\_atat\_hours\_denominator,

normalized\_opex\_total\_cph\_all\_hours\_denominator,

normalized\_opex\_labor\_cph\_atat\_hours\_denominator,

normalized\_opex\_labor\_cph\_all\_hours\_denominator,

manifest\_average\_da\_hourly\_wage,

manifest\_opex\_total\_cph\_atat\_hours\_denominator,

manifest\_opex\_total\_cph\_all\_hours\_denominator,

manifest\_opex\_labor\_cph\_atat\_hours\_denominator,

manifest\_opex\_labor\_cph\_all\_hours\_denominator,

manifest\_op2\_total\_cph\_atat\_hours\_denominator,

manifest\_op2\_total\_cph\_all\_hours\_denominator,

manifest\_op2\_labor\_cph\_atat\_hours\_denominator,

manifest\_op2\_labor\_cph\_all\_hours\_denominator,

manifest\_quarterly\_total\_cph\_atat\_hours\_denominator,

manifest\_quarterly\_total\_cph\_all\_hours\_denominator,

manifest\_quarterly\_labor\_cph\_atat\_hours\_denominator,

manifest\_quarterly\_labor\_cph\_all\_hours\_denominator,

manifest\_normalized\_opex\_total\_cph\_atat\_hours\_denominator,

manifest\_normalized\_opex\_total\_cph\_all\_hours\_denominator,

manifest\_normalized\_opex\_labor\_cph\_atat\_hours\_denominator,

manifest\_normalized\_opex\_labor\_cph\_all\_hours\_denominator)

as

(

Select

finance\_deck.\*,

parent\_work\_type,

work\_type,

workflow\_app,

base\_verify\_workflow\_app,

core\_non\_core\_workflow\_app,

b.average\_da\_hourly\_wage,

b.opex\_total\_cph\_atat\_hours\_denominator,

b.opex\_total\_cph\_all\_hours\_denominator,

b.opex\_labor\_cph\_atat\_hours\_denominator,

b.opex\_labor\_cph\_all\_hours\_denominator,

b.op2\_total\_cph\_atat\_hours\_denominator,

b.op2\_total\_cph\_all\_hours\_denominator,

b.op2\_labor\_cph\_atat\_hours\_denominator,

b.op2\_labor\_cph\_all\_hours\_denominator,

b.quarterly\_total\_cph\_atat\_hours\_denominator,

b.quarterly\_total\_cph\_all\_hours\_denominator,

b.quarterly\_labor\_cph\_atat\_hours\_denominator,

b.quarterly\_labor\_cph\_all\_hours\_denominator,

b.normalized\_opex\_total\_cph\_atat\_hours\_denominator,

b.normalized\_opex\_total\_cph\_all\_hours\_denominator,

b.normalized\_opex\_labor\_cph\_atat\_hours\_denominator,

b.normalized\_opex\_labor\_cph\_all\_hours\_denominator,

c.average\_da\_hourly\_wage as manifest\_average\_da\_hourly\_wage,

c.opex\_total\_cph\_atat\_hours\_denominator as manifest\_opex\_total\_cph\_atat\_hours\_denominator,

c.opex\_total\_cph\_all\_hours\_denominator as manifest\_opex\_total\_cph\_all\_hours\_denominator,

c.opex\_labor\_cph\_atat\_hours\_denominator as manifest\_opex\_labor\_cph\_atat\_hours\_denominator,

c.opex\_labor\_cph\_all\_hours\_denominator as manifest\_opex\_labor\_cph\_all\_hours\_denominator,

c.op2\_total\_cph\_atat\_hours\_denominator as manifest\_op2\_total\_cph\_atat\_hours\_denominator,

c.op2\_total\_cph\_all\_hours\_denominator as manifest\_op2\_total\_cph\_all\_hours\_denominator,

c.op2\_labor\_cph\_atat\_hours\_denominator as manifest\_op2\_labor\_cph\_atat\_hours\_denominator,

c.op2\_labor\_cph\_all\_hours\_denominator as manifest\_op2\_labor\_cph\_all\_hours\_denominator,

c.quarterly\_total\_cph\_atat\_hours\_denominator as manifest\_quarterly\_total\_cph\_atat\_hours\_denominator,

c.quarterly\_total\_cph\_all\_hours\_denominator as manifest\_quarterly\_total\_cph\_all\_hours\_denominator,

c.quarterly\_labor\_cph\_atat\_hours\_denominator as manifest\_quarterly\_labor\_cph\_atat\_hours\_denominator,

c.quarterly\_labor\_cph\_all\_hours\_denominator as manifest\_quarterly\_labor\_cph\_all\_hours\_denominator,

c.normalized\_opex\_total\_cph\_atat\_hours\_denominator as manifest\_normalized\_opex\_total\_cph\_atat\_hours\_denominator,

c.normalized\_opex\_total\_cph\_all\_hours\_denominator as manifest\_normalized\_opex\_total\_cph\_all\_hours\_denominator,

c.normalized\_opex\_labor\_cph\_atat\_hours\_denominator as manifest\_normalized\_opex\_labor\_cph\_atat\_hours\_denominator,

c.normalized\_opex\_labor\_cph\_all\_hours\_denominator as manifest\_normalized\_opex\_labor\_cph\_all\_hours\_denominator

From

(select

'ATAT' as ads\_process\_type,

a.analyst,

a.badge\_color,

supervisor\_id,

a.supervisor\_login\_name,

a.week,a.month,a.quarter,a.year,reporting\_year,reporting\_week,

country, site, cost\_center, priority,

queue\_type, task\_data\_language, transformation, root\_transformation, transformation\_type,

task\_queue\_type, device\_environment, ADS\_service, order\_name,

service\_order\_id,

customer\_grp\_value,

case when customer\_grp\_value is null and vertical\_or\_sub\_initiative<> 'Other' then vertical\_or\_sub\_initiative

when customer\_grp\_value is null and vertical\_or\_sub\_initiative<> 'Other' then device\_or\_initiative

else customer\_grp\_value end as customer,

vertical\_or\_sub\_initiative, device\_or\_initiative, task\_queue\_id, task\_queue\_name,sla\_planned\_flg,convention, owner, owner\_id, aim, reason\_code,

rolling\_month,

rolling\_week,

rolling\_qtr,

rolling\_year,

sum(Auto\_verified\_volume) as Auto\_Verified\_Volume, sum(Manual\_verified\_volume) as Manual\_Verified\_Volume,

sum(auto\_processed\_volume) as Auto\_processed\_volume, sum(Verified\_volume) as Verified\_volume,

avg(weekly\_staff\_hrs) as weekly\_staff\_hrs,

avg(weekly\_allocated\_hrs) as weekly\_allocated\_hrs,

avg(total\_working\_hrs) as total\_working\_hrs,

sum(op2\_volume) as op2\_volume,

sum(target\_volume) as target\_volume,

sum(committed\_volume) as committed\_volume,

sum(ingested\_count) as ingested\_count,

sum(discard\_count) as discard\_count,

sum(auto\_skipped\_count) as auto\_skipped\_count,

sum(overturned\_count) as overturned\_count,

sum(proc) as processed\_volume,

sum(hours) as production\_hours,

sum(sla\_volume) as sla\_volume

from

(select

analyst,

badge\_color,

supervisor\_id,

supervisor\_login\_name,

calendar\_date,

calendar\_week as week,

reporting\_week,

calendar\_mth as month,

calendar\_qtr as quarter,

reporting\_year,

calendar\_year as year,

root\_transformation,

device\_environment,

country,

sla\_planned\_flg,

ADS\_service,

order\_name,

convention,

owner,

owner\_id,

aim,

reason\_code,

building site,

cost\_center,

queue\_type,

task\_queue\_type,

task\_data\_language,

transformation,

transformation\_type,

priority,

vertical\_or\_sub\_initiative,

device\_or\_initiative,

task\_queue\_id,

task\_queue\_name,

service\_order\_id,

customer\_grp\_value,

rolling\_month,

rolling\_week,

rolling\_qtr,

rolling\_year,

sum(Auto\_verified\_volume) as Auto\_verified\_volume,

sum(Manual\_verified\_volume) as Manual\_verified\_volume,

sum(auto\_processed\_volume) as Auto\_processed\_volume,

sum(Verified\_volume) as Verified\_volume,

sum(op2\_volume) as op2\_volume,

sum(target\_volume) as target\_volume,

sum(committed\_volume) as committed\_volume,

sum(ingested\_count) as ingested\_count,

sum(discard\_count) as discard\_count,

sum(auto\_skipped\_count) as auto\_skipped\_count,

sum(overturned\_count) as overturned\_count,

sum(processed) proc,

sum(hours) hours,

sum(sla\_volume) as sla\_volume

from

(

select

fqat.analyst,

badge\_color,

deh.supervisor\_id,

deh.supervisor\_login\_name,

deh.department\_id cost\_center,

deh.building,

deh.country,

fqat.priority,

dtq.task\_queue\_id,

dtq.task\_queue\_name,

dtq.task\_queue\_type,

fqat.sla\_planned\_flg,

dtq.queue\_type\_tag as queue\_type,

dtq.task\_data\_language,

dtq.transformation,

fqat.convention,

fqat.owner,

fqat.owner\_id,

fqat.aim,

fqat.reason\_code,

dtq2.transformation as root\_transformation,

dtq2.device\_environment,

dtq.vertical\_or\_sub\_initiative,

dtq.device\_or\_initiative,

fqat.calendar\_date,

fso2.ADS\_service,

fso2.order\_name,

fqat.reporting\_year,

fqat.calendar\_week,

fqat.calendar\_mth,

fqat.calendar\_qtr,

fqat.reporting\_week,

fqat.calendar\_year,

fqat.transformation\_type,

dtq.service\_order\_id,

cust.customer\_grp\_value,

rolling\_month,

rolling\_week,

rolling\_qtr,

rolling\_year,

ttp.auto\_verified\_volume as Auto\_verified\_volume,

(fqat.verified\_count - nvl(ttp.auto\_verified\_volume,0)) AS Manual\_verified\_volume,

nvl(ttp\_prc.auto\_processed\_count,0) AS auto\_processed\_volume,

sum(fqat.verified\_count) Verified\_volume,

SUM(fqat.processed\_count) processed,

SUM(fqat.production\_hrs) hours,

sum(sla\_volume) as sla\_volume,

sum(op2\_volume) as op2\_volume,

sum(target\_volume) as target\_volume,

sum(committed\_volume) as committed\_volume,

sum(ingested\_count) as ingested\_count,

sum(discard\_count) as discard\_count,

sum(auto\_skipped\_count) as auto\_skipped\_count,

sum(overturned\_count) as overturned\_count

from nvrpt.daily\_queue\_analyst\_metrics\_vw fqat

JOIN nvads.dim\_task\_queue\_view dtq

on fqat.task\_queue\_id = dtq.task\_queue\_id

left join nvads.dim\_task\_queue\_chain dtqc

on dtq.task\_queue\_id=dtqc.task\_queue\_id

left join nvads.dim\_task\_queue dtq2

on dtqc.root\_task\_queue\_id=dtq2.task\_queue\_id

left join

(select distinct order\_id, max(CUST\_GRP\_ID) as cust\_grp\_id from nvads.fact\_service\_order group by 1) fso

on dtq.service\_order\_id = fso.order\_id

left join

(select distinct customer\_grp\_key, customer\_grp\_value, max(cust\_grp\_id) as cust\_grp\_id from nvads.dim\_customer\_group group by 1,2) cust

on fso.cust\_grp\_id = cust.cust\_grp\_id

left join

(select distinct order\_id, order\_create\_dt, order\_update\_dt, order\_name, aim\_work\_type as ADS\_Service

from

(select distinct order\_id, order\_create\_dt, order\_update\_dt, order\_name, aim\_work\_type, rank() over (partition by order\_id order by order\_update\_dt desc) as rank

from nvads.fact\_service\_order

where (lower(order\_name) not like '%%[closed]%%' or lower(order\_name) not like '%%[closded]%%'))

where rank = 1) fso2

on dtq.service\_order\_id = fso2.order\_id

LEFT JOIN nvads.dim\_employee\_history deh

on fqat.analyst = deh.login\_name

and fqat.calendar\_date >= deh.effective\_start\_day

and fqat.calendar\_date < deh.effective\_end\_day

and (fqat.calendar\_date < deh.termination\_day

or deh.termination\_day is null)

left join

(select distinct analyst, week\_start\_date, badge\_color

from nvdev.wbr\_associate\_skill\_map\_fact

where year>=2018) b

on fqat.analyst=b.analyst and fqat.week\_month\_start\_date=b.week\_start\_date

LEFT JOIN (SELECT ttp.processed\_date,

ttp.task\_queue\_id,

ttp.analyst,

SUM(ttp.verif\_auto\_saved\_cnt) AS auto\_verified\_volume

FROM nvads.fact\_ttop\_audit\_summary ttp

where processed\_date>='2018-01-01'

GROUP BY 1,

2,

3) ttp

ON fqat.task\_queue\_id = ttp.task\_queue\_id

AND fqat.analyst = ttp.analyst

AND fqat.calendar\_date = ttp.processed\_date

LEFT JOIN (SELECT verif\_analyst AS analyst,

verif\_date AS processed\_date,

verif\_task\_queue\_id AS task\_queue\_id,

COUNT(DISTINCT verif\_utterance\_id) AS processed\_count,

COUNT(DISTINCT CASE WHEN verif\_auto\_save = 1 THEN verif\_utterance\_id END) AS auto\_processed\_count

FROM nvads.fact\_ttop\_utterance

WHERE verif\_utterance\_Id IS NOT NULL and processed\_date>='2018-01-01'

-- AND dedup\_rn = 1

GROUP BY verif\_analyst,

verif\_date,

verif\_task\_queue\_id) ttp\_prc

ON fqat.task\_queue\_id = ttp\_prc.task\_queue\_id

AND fqat.analyst = ttp\_prc.analyst

AND fqat.calendar\_date = ttp\_prc.processed\_date

where date\_part('year',fqat.calendar\_date) >= 2018

and date\_trunc('week',fqat.calendar\_date) <> date\_trunc('week',SYSDATE)

group by 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43

)

group by 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40)a

left join

(select distinct analyst, week, datepart(quarter, week\_start\_date) as quarter, month, year, analyst\_expected\_hrs, weekly\_staff\_hrs, weekly\_allocated\_hrs, total\_working\_hrs

from nvdev.wbr\_associate\_skill\_map\_fact

where year>=2018)b

on a.analyst=b.analyst and a.week=b.week and a.month=b.month and a.quarter=b.quarter and a.year=b.year

group by 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41

union all

select

'RVD' as ads\_process\_type,

user\_login as analyst,

'RVD' as badge\_color,

'RVD' as supervisor\_id,

'RVD' as supervisor\_login\_name,

calendar\_week as week,

calendar\_mth as month,

calendar\_qtr as quarter,

calendar\_year as year,

reporting\_year,

reporting\_week,

case when country is null then 'Total' else country end as country,

site,

cost\_center,

'RVD' as priority,

'RVD' as queue\_type,

REPLACE(lang\_code, '-', '\_') as task\_data\_language,

transformation,

'RVD' as root\_transformation,

transformation as transformation\_type,

'RVD' as task\_queue\_type,

'RVD' as device\_environment,

'RVD' as ADS\_service,

'RVD' as order\_name,

'RVD' as service\_order\_id,

cycle\_customer as customer\_grp\_value,

cycle\_customer as customer,

cycle\_customer as vertical\_or\_sub\_initiative,

cycle\_customer as device\_or\_initiative,

cycle\_id :: text as task\_queue\_id,

cycle\_name as task\_queue\_name,

'Y' as sla\_planned\_flg,

'RVD' as convention,

'RVD' as owner,

null :: int as owner\_id,

'RVD' as aim,

'RVD' as reason\_code,

rolling\_month,

rolling\_week,

rolling\_qtr,

rolling\_year,

0 as Auto\_Verified\_Volume,

0 as Manual\_Verified\_Volume,

0 as Auto\_processed\_volume,

0 as Verified\_volume,

0 as weekly\_staff\_hrs,

0 as weekly\_allocated\_hrs,

0 as total\_working\_hrs,

0 as op2\_volume,

0 as target\_volume,

0 as committed\_volume,

0 as ingested\_count,

0 as discard\_count,

0 as auto\_skipped\_count,

0 as overturned\_count,

testcase\_count as processed\_volume,

processed\_hrs as production\_hours,

0 as sla\_volume

from

(select rvd.\*, deh.country, deh.building as site, deh.department\_id as cost\_center,

cycle\_type || ' ~ work state = ' || cycle\_state as transformation

from

(SELECT "Test Case Counts"."cycle\_id" AS "cycle\_id",

"Test Case Counts"."user\_id" AS "user\_id",

"Test Case Counts"."user\_login" AS "user\_login",

"Test Case Counts"."processed\_date" AS "processed\_date",

"Test Case Counts"."status" AS "status",

"Test Case Counts"."testcase\_count" AS "testcase\_count",

"Test Case Counts"."processed\_hrs" AS "processed\_hrs",

"Test Case Counts"."processed\_hrs\_rr" AS "processed\_hrs\_rr",

"date\_dim"."date\_key" AS "date\_key",

"date\_dim"."calendar\_date" AS "calendar\_date",

"date\_dim"."calendar\_half" AS "calendar\_half",

"date\_dim"."calendar\_mth" AS "calendar\_mth",

"date\_dim"."calendar\_qtr" AS "calendar\_qtr",

"date\_dim"."calendar\_week" AS "calendar\_week",

"date\_dim"."calendar\_year" AS "calendar\_year",

"date\_dim"."day\_of\_mth" AS "day\_of\_mth",

"date\_dim"."day\_of\_week" AS "day\_of\_week",

"date\_dim"."day\_of\_year" AS "day\_of\_year",

"date\_dim"."days\_in\_mth" AS "days\_in\_mth",

"date\_dim"."days\_in\_qtr" AS "days\_in\_qtr",

"date\_dim"."fiscal\_half" AS "fiscal\_half",

"date\_dim"."fiscal\_mth" AS "fiscal\_mth",

"date\_dim"."fiscal\_qtr" AS "fiscal\_qtr",

"date\_dim"."fiscal\_week" AS "fiscal\_week",

"date\_dim"."fiscal\_year" AS "fiscal\_year",

"date\_dim"."julian\_day\_num" AS "julian\_day\_num",

"date\_dim"."julian\_mth\_num" AS "julian\_mth\_num",

"date\_dim"."julian\_qtr\_num" AS "julian\_qtr\_num",

"date\_dim"."julian\_week\_num" AS "julian\_week\_num",

"date\_dim"."period\_name\_half" AS "period\_name\_half",

"date\_dim"."period\_name\_mth" AS "period\_name\_mth",

"date\_dim"."period\_name\_qtr" AS "period\_name\_qtr",

"date\_dim"."period\_name\_week" AS "period\_name\_week",

"date\_dim"."period\_name\_year" AS "period\_name\_year",

"date\_dim"."reporting\_week" AS "reporting\_week",

"date\_dim"."reporting\_year" AS "reporting\_year",

"date\_dim"."rpt\_period\_name\_week" AS "rpt\_period\_name\_week",

"date\_dim"."rpt\_period\_name\_year" AS "rpt\_period\_name\_year",

"date\_dim"."rolling\_day" AS "rolling\_day",

"date\_dim"."rolling\_week" AS "rolling\_week",

"date\_dim"."rolling\_month" AS "rolling\_month",

"date\_dim"."rolling\_qtr" AS "rolling\_qtr",

"date\_dim"."rolling\_year" AS "rolling\_year",

"tap\_cycle"."cycle\_id" AS "cycle\_id \_tap\_cycle\_",

"tap\_cycle"."created\_date" AS "created\_date",

"tap\_cycle"."modified\_date" AS "modified\_date",

"tap\_cycle"."cycle\_name" AS "cycle\_name",

"tap\_cycle"."cycle\_type" AS "cycle\_type",

"tap\_cycle"."lang\_code" AS "lang\_code",

"tap\_cycle"."target\_submissions" AS "target\_submissions",

"tap\_cycle"."cycle\_state" AS "cycle\_state",

"tap\_cycle"."cycle\_owner" AS "cycle\_owner",

"tap\_cycle"."cycle\_priority" AS "cycle\_priority",

"tap\_cycle"."cycle\_customer" AS "cycle\_customer",

"tap\_cycle"."closing\_date" AS "closing\_date",

"tap\_cycle"."review\_required" AS "review\_required",

"tap\_cycle"."sim\_issue" AS "sim\_issue",

"tap\_cycle"."close" AS "close"

FROM (SELECT cycle\_id,

user\_id,

user\_login,

processed\_date,

resulting\_testcase\_completeness\_status AS status,

COUNT(DISTINCT testcase\_id) AS testcase\_count,

SUM(total\_duration\_seconds) /(60\*60::FLOAT) AS processed\_hrs,

SUM(total\_duration\_seconds\_lex) /(60\*60::FLOAT) AS processed\_hrs\_rr

--to get most recent submission at cycle,testcase,user,date and status level(duration is added)

FROM (SELECT \*,

CASE

WHEN resulting\_testcase\_completeness\_status IN ('FINISHED') THEN SUM(

CASE

WHEN resulting\_testcase\_completeness\_status IN ('FINISHED','NEED\_REVIEW') THEN total\_duration\_seconds

ELSE 0

END ) OVER (PARTITION BY cycle\_id,testcase\_id ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED following)

ELSE 0

END AS total\_duration\_seconds\_lex

-- to get the total time taken for review req workflows(lexicon)

FROM (SELECT DISTINCT cycle\_id,

testcase\_id,

user\_id,

user\_login,

DATE (created\_date) AS processed\_date,

resulting\_testcase\_completeness\_status,

-- ROW\_NUMBER() OVER (PARTITION BY cycle\_id,DATE (created\_date),user\_id,testcase\_id,resulting\_testcase\_completeness\_status ORDER BY created\_date DESC) AS rn,

FIRST\_VALUE(submission\_id IGNORE NULLS) OVER (PARTITION BY cycle\_id,DATE (created\_date),user\_id,testcase\_id,resulting\_testcase\_completeness\_status ORDER BY created\_date ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED following) AS latest\_submission\_id,

-- FIRST\_VALUE(created\_date IGNORE NULLS) OVER (PARTITION BY cycle\_id,DATE (created\_date),user\_id,testcase\_id,resulting\_testcase\_completeness\_status ORDER BY created\_date ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED following) AS processed\_datetime,

SUM(submission\_duration\_seconds) OVER (PARTITION BY cycle\_id,testcase\_id,user\_id,resulting\_testcase\_completeness\_status ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED following) AS total\_duration\_seconds

FROM nvrvd.tap\_submission))

where processed\_date>='2019-01-01'

GROUP BY cycle\_id,

user\_id,

user\_login,

processed\_date,

resulting\_testcase\_completeness\_status) "Test Case Counts"

INNER JOIN "nvdev"."date\_dim" "date\_dim" ON ("Test Case Counts"."processed\_date" = "date\_dim"."calendar\_date")

LEFT JOIN "nvrvd"."tap\_cycle" "tap\_cycle" ON ("Test Case Counts"."cycle\_id" = "tap\_cycle"."cycle\_id"))rvd

LEFT JOIN nvads.dim\_employee\_history deh

on rvd.user\_login = deh.login\_name

and rvd.processed\_date >= deh.effective\_start\_day

and rvd.processed\_date < deh.effective\_end\_day

and (rvd.processed\_date < deh.termination\_day

or deh.termination\_day is null))

union all

select

'RAMP' as ads\_process\_type,

'RAMP' as analyst,

'RAMP' as badge\_color,

'RAMP' as supervisor\_id,

'RAMP' as supervisor\_login\_name,

cast (week as numeric) as week,

cast (actual\_month as numeric) as month,

cast (quarter as numeric) as quarter,

cast (actual\_year as numeric) as year,

cast (a.reporting\_year as numeric) as reporting\_year,

cast (b.reporting\_week as numeric) as reporting\_week,

country,

site,

'5140' as cost\_center,

'RAMP' as priority,

'RAMP' as queue\_type,

task\_data\_language,

transformation,

'RAMP' as root\_transformation,

transformation\_type,

'RAMP' as task\_queue\_type,

'RAMP' as device\_environment,

'RAMP' as ADS\_service,

task\_queue\_name as order\_name,

task\_queue\_id as service\_order\_id,

customer\_grp\_value,

customer\_grp\_value as customer,

customer\_grp\_value as vertical\_or\_sub\_initiative,

customer\_grp\_value as device\_or\_initiative,

task\_queue\_id,

task\_queue\_name,

'Y' as sla\_planned\_flg,

'RAMP' as convention,

'RAMP' as owner,

null :: int as owner\_id,

'RAMP' as aim,

'RAMP' as reason\_code,

rolling\_month,

rolling\_week,

rolling\_qtr,

rolling\_year,

0 as Auto\_Verified\_Volume,

0 as Manual\_Verified\_Volume,

0 as Auto\_processed\_volume,

0 as Verified\_volume,

0 as weekly\_staff\_hrs,

0 as weekly\_allocated\_hrs,

0 as total\_working\_hrs,

0 as op2\_volume,

0 as target\_volume,

0 as committed\_volume,

0 as ingested\_count,

0 as discard\_count,

0 as auto\_skipped\_count,

0 as overturned\_count,

sum(cast(processed\_count as double precision)) as processed\_volume,

sum(cast(production\_hours as double PRECISION)) as production\_hours,

Sum(cast(sla\_volume as double precision)) as sla\_volume

From nvads.RAMP\_edx a

Join nvdev.date\_dim b

On date(a.week\_start\_date)=b.calendar\_date

group by 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55) finance\_deck

left join

(select distinct country, year, month,

b.average\_da\_hourly\_wage,

b.opex\_total\_cph\_atat\_hours\_denominator,

b.opex\_total\_cph\_all\_hours\_denominator,

b.opex\_labor\_cph\_atat\_hours\_denominator,

b.opex\_labor\_cph\_all\_hours\_denominator,

b.op2\_total\_cph\_atat\_hours\_denominator,

b.op2\_total\_cph\_all\_hours\_denominator,

b.op2\_labor\_cph\_atat\_hours\_denominator,

b.op2\_labor\_cph\_all\_hours\_denominator,

b.quarterly\_total\_cph\_atat\_hours\_denominator,

b.quarterly\_total\_cph\_all\_hours\_denominator,

b.quarterly\_labor\_cph\_atat\_hours\_denominator,

b.quarterly\_labor\_cph\_all\_hours\_denominator,

b.normalized\_opex\_total\_cph\_atat\_hours\_denominator,

b.normalized\_opex\_total\_cph\_all\_hours\_denominator,

b.normalized\_opex\_labor\_cph\_atat\_hours\_denominator,

b.normalized\_opex\_labor\_cph\_all\_hours\_denominator

from nvads.cph\_edx b) b

on finance\_deck.country=b.country and finance\_deck.month=b.month and finance\_deck.year=b.year

left join

(select distinct country, manifest\_year, manifest\_month,

c.average\_da\_hourly\_wage,

c.opex\_total\_cph\_atat\_hours\_denominator,

c.opex\_total\_cph\_all\_hours\_denominator,

c.opex\_labor\_cph\_atat\_hours\_denominator,

c.opex\_labor\_cph\_all\_hours\_denominator,

c.op2\_total\_cph\_atat\_hours\_denominator,

c.op2\_total\_cph\_all\_hours\_denominator,

c.op2\_labor\_cph\_atat\_hours\_denominator,

c.op2\_labor\_cph\_all\_hours\_denominator,

c.quarterly\_total\_cph\_atat\_hours\_denominator,

c.quarterly\_total\_cph\_all\_hours\_denominator,

c.quarterly\_labor\_cph\_atat\_hours\_denominator,

c.quarterly\_labor\_cph\_all\_hours\_denominator,

c.normalized\_opex\_total\_cph\_atat\_hours\_denominator,

c.normalized\_opex\_total\_cph\_all\_hours\_denominator,

c.normalized\_opex\_labor\_cph\_atat\_hours\_denominator,

c.normalized\_opex\_labor\_cph\_all\_hours\_denominator

from nvads.cph\_edx c) c

on finance\_deck.country=c.country and finance\_deck.month=c.manifest\_month and finance\_deck.year=c.manifest\_year

left join

(select distinct parent\_work\_type, work\_type, workflow\_app, base\_verify\_workflow\_app, core\_non\_core\_workflow\_app

from nvads.ads\_workflows\_categorical\_mappings) d

on finance\_deck.transformation=d.workflow\_app

)

WITH NO SCHEMA binding;

GRANT SELECT, INSERT, TRIGGER, DELETE, RULE, UPDATE, REFERENCES ON nvrpt.finance\_deck TO nvort\_dev;

GRANT SELECT ON nvrpt.finance\_deck TO nvort\_rs\_etl;

GRANT SELECT ON nvrpt.finance\_deck TO ads\_rs\_science\_access;

GRANT SELECT ON nvrpt.finance\_deck TO ads\_tech\_rs\_etl;

GRANT TRIGGER, INSERT, REFERENCES, UPDATE, SELECT, RULE, DELETE ON nvrpt.finance\_deck TO nvort;

GRANT SELECT ON nvrpt.finance\_deck TO ads\_tech\_redash;

GRANT SELECT ON nvrpt.finance\_deck TO ads\_ro\_swift\_ast;