**PROCEDURE TO CONVERT ‘na’ VALUES:**

CREATE or replace PROCEDURE usp\_convertnavalues ()

AS '

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

--SET NOCOUNT ON;

update stg.stg\_ontime

SET

"year"=''0000''

where "year"=''NA'' or "year"=''None'' ;

update stg.stg\_ontime

SET

"month"=''00''

where "month"=''NA'';

update stg.stg\_ontime

SET

dayofmonth=''0000''

where dayofmonth=''NA'';

update stg.stg\_ontime

SET

dayofweek=''0000''

where dayofweek=''NA'';

update stg.stg\_ontime

SET

flightnum=''0000''

where flightnum=''NA'';

update stg.stg\_ontime

SET

crselapsedtime=''0000''

where crselapsedtime=''NA'';

update stg.stg\_ontime

SET

DepTime=''0000''

where DepTime=''NA'';

update stg.stg\_ontime

SET

CRSDepTime=''0000''

where CRSDepTime=''NA'';

update stg.stg\_ontime

SET

ArrTime=''0000''

where ArrTime=''NA'';

update stg.stg\_ontime

SET

CRSArrTime=''0000''

where CRSArrTime=''NA'';

update stg.stg\_ontime

SET

ActualElapsedTime=''0000''

where ActualElapsedTime=''NA'';

update stg.stg\_ontime

SET

AirTime=''0000''

where AirTime=''NA'';

update stg.stg\_ontime

SET

ArrDelay=''0000''

where ArrDelay=''NA'';

update stg.stg\_ontime

SET

DepDelay=''0000''

where DepDelay=''NA'';

update stg.stg\_ontime

SET

Distance=''0000''

where Distance=''NA'';

update stg.stg\_ontime

SET

TaxiIn=''0000''

where TaxiIn=''NA'';

update stg.stg\_ontime

SET

TaxiOut=''0000''

where TaxiOut=''NA'';

update stg.stg\_ontime

SET

cancelled=''0000''

where cancelled=''NA'';

END;

'

language PLPGSQL

;

**Procedures for loading Dimension tables:**

1. **Dim\_DestAirport:**

create or replace procedure load\_destairport()

as'

declare newdatetime timestamp;

begin

--load new records

insert into dwh.dim\_destairport(destairportkey,iata,airportname,city,state,country,lattitude,longitude,start\_date,end\_date,flag)

select row\_number() over (order by stg.iata),

stg.iata

,stg.airport

,stg.city

,stg.state

,stg.country

,stg.lat

,stg.long ,

getdate(),

''12/31/2030'',

''0''

from stg.stg\_airports stg

where not exists(select \* from dwh.dim\_destairport d where d.iata=stg.iata);

newdatetime := getdate();

update dwh.dim\_destairport

set end\_date =newdatetime ,flag=1

from dwh.dim\_destairport d

inner join stg.stg\_airports stg

on d.iata=stg.iata

where(d.end\_date=''12/31/2030'')

and (stg.airport<>d.airportname or stg.country<>d.country or stg.city<>d.city or stg.state<>d.state or stg.lat<>d.lattitude or stg.long<>d.longitude);

insert into dwh.dim\_destairport(destairportkey,iata,airportname,city,state,country,lattitude,longitude,start\_date,end\_date,flag)

select row\_number() over (order by stg.iata),

stg.iata

,stg.airport

,stg.city

,stg.state

,stg.country

,stg.lat

,stg.long

,newdatetime,

''12/31/2030'',

''0''

from stg.stg\_airports stg

inner join dwh.dim\_destairport d

on d.iata=stg.iata

and d.end\_date=newdatetime;

end;

'

language 'plpgsql';

1. **Dim\_OrgAirport:**

create or replace procedure load\_orgairport()

as'

declare newdatetime timestamp;

begin

--load new records

insert into dwh.dim\_orgairport(orgairportkey,iata,airportname,city,state,country,lattitude,longitude,start\_date,end\_date,flag)

select row\_number() over (order by stg.iata),

stg.iata

,stg.airport

,stg.city

,stg.state

,stg.country

,stg.lat

,stg.long ,

getdate(),

''12/31/2030'',

''0''

from stg.stg\_airports stg

where not exists(select \* from dwh.dim\_orgairport d where d.iata=stg.iata);

newdatetime := getdate();

update dwh.dim\_orgairport

set end\_date =newdatetime ,flag=1

from dwh.dim\_orgairport d

inner join stg.stg\_airports stg

on d.iata=stg.iata

where(d.end\_date=''12/31/2030'')

and (stg.airport<>d.airportname or stg.country<>d.country or stg.city<>d.city or stg.state<>d.state or stg.lat<>d.lattitude or stg.long<>d.longitude);

insert into dwh.dim\_orgairport(orgairportkey,iata,airportname,city,state,country,lattitude,longitude,start\_date,end\_date,flag)

select row\_number() over (order by stg.iata),

stg.iata

,stg.airport

,stg.city

,stg.state

,stg.country

,stg.lat

,stg.long

,newdatetime,

''12/31/2030'',

''0''

from stg.stg\_airports stg

inner join dwh.dim\_orgairport d

on d.iata=stg.iata

and d.end\_date=newdatetime;

end;

'

language 'plpgsql';

1. **Dim\_Plane:**

create or replace procedure load\_dim\_plane ()

as '

begin

insert into DWH.dim\_plane(planekey,tailnum,planetype,manufacturer,issuedate,model,status,aircrafttype,enginetype,mfgyear,loadtimestamp,loaduserid)

select row\_number() over (order by tailnum),tailnum,type, manufacturer ,

issue\_date,

model,

status,

aircraft\_type,

engine\_type,

year,getdate() as loadtimestamp,null as loaduserid

from STG.stg\_planes;

end;

'Language plpgsql;

1. **Dim\_Carriers:**

CREATE OR REPLACE PROCEDURE LOAD\_DIM\_CARRIER()

AS

'

BEGIN

insert into dwh.dim\_carrier(carrierkey,carriercode,carrierdescription,loadtimestamp,loaduserid)

select row\_number() over (order by code),code,description,getdate() as loadtimestamp,null as loaduserid

from stg.stg\_carriers;

END;

'

LANGUAGE PLPGSQL;

**5.Dim\_Date:**

CREATE OR REPLACE PROCEDURE public.usp\_loaddimdate(

strdt date,

enddt date)

LANGUAGE 'plpgsql'

AS '

declare ctrDt date:= strdt;

BEGIN

drop table if exists tempDt;

create table tempDt as

SELECT cast(to\_char(getdate(), ''yyyymmdd'') as int) as DateKey

,current\_date as FullDateAlternateKey

,extract(dow from getdate())+1 as DayNumberOfWeek

,TO\_CHAR(getdate(), ''Day'') as DayNameOfWeek

,extract(day from getdate()) as DayNumberOfMonth

,extract(DOY from getdate())as DayNumberOfYear

,extract(week from getdate())+1 as WeekNumberOfYear

,TO\_CHAR(getdate(), ''month'') as MonthName

,extract(month from getdate()) as MonthNumberOfYear

,extract(quarter from getdate()) as CalendarQuarter

,extract(month from getdate()) as CalendarSemester

,extract(year from getdate()) as CalendarYear

, getdate() as LoadTimestamp

,''SQLServer'' as LoadUserId;

truncate table tempDt;

while ctrDt <=enddt LOOP

INSERT into tempDt

SELECT cast(to\_char(ctrDt, ''yyyymmdd'') as int) as DateKey

,cast(to\_char(ctrDt, ''yyyymmdd'') as date) as FullDateAlternateKey

,extract(dow from ctrDt) as DayNumberOfWeek

,TO\_CHAR(ctrDt, ''Day'') as DayNameOfWeek

,extract(day from ctrDt) as DayNumberOfMonth

,extract(DOY from ctrDt)as DayNumberOfYear

,extract(week from ctrDt)+1 as WeekNumberOfYear

,TO\_CHAR(ctrDt, ''month'') as MonthName

,extract(month from ctrDt) as MonthNumberOfYear

,extract(quarter from ctrDt) as CalendarQuarter

,extract(month from ctrDt) as CalendarSemester

,extract(year from ctrDt) as CalendarYear

,getdate() as LoadTimestamp

,''SQLServer'' as LoadUserId;

ctrDt = ctrDt + INTERVAL ''1 day'';

END LOOP ;

INSERT INTO dwh.dim\_date select \* from tempDt;

DROP TABLE tempDt;

COMMIT;

END;

'

call usp\_LoadDimDate('01/01/1990', '12/31/2020');

6. **Fact\_ontime**:

create or replace procedure load\_fact\_ontime ()

as '

begin

--insert into fact table

insert into dwh.fact\_ontime(fact\_key,fk\_carrier\_key, fk\_planekey,

fk\_orgairportkey, fk\_destairportkey, datekey,

flight\_year, flightmonth, dayofmonth, dayofweek, deptime,

crsdeptime, arrtime, crsarrtime, uniquecarrier, flightnum, tailnum,

actualelapsedtime, crselapsedtime, airtime, arrdelay, depdelay, origin,

dest, distance, taxiin, taxiout, cancelled, cancellationcode, diverted,

carrierdelay, weatherdelay, nasdelay, securitydelay, lateaircraftdelay

)

select row\_number() over (order by dayofweek),c.carrierkey,p.planekey,a.orgairportkey,de.destairportkey,d.datekey

,cast(s.year as int)as "year",

cast(s.month as int) as "month" ,

cast(s.dayofmonth as int) as "dayofmonth",

cast(s.dayofweek as int) as "dayofweek",

cast(s.deptime as int) as "deptime",

cast(s.crsdeptime as int) as "crsdeptime",

cast(s.arrtime as int) as "arrtime",

cast(s.crsarrtime as int) as "crsarrtime",

s.uniquecarrier,cast(s.flightnum as int) as "flightnum", s.tailnum,

cast(s.actualelapsedtime as int) as "actualelapsedtime",

cast(s.crselapsedtime as int) as "crselapsedtime",

cast(s.airtime as int) as "airtime",

cast(s.arrdelay as int) as "arrdelay",

cast(s.depdelay as int) as "depdelay",

s.origin,s.dest,

cast(s.distance as int) as "distance",

cast(s.taxiin as int) as "taxiin",

cast(s.taxiout as int) as "taxiout",

cast(s.cancelled as int) as "cancelled",s.cancellationcode,

cast(s.diverted as int) as "diverted",s.carrierdelay,

s.weatherdelay,s.nasdelay, s.securitydelay,s.lateaircraftdelay

from stg.stg\_ontime s

left join dim\_plane p on s.tailnum=p.tailnum

join dim\_orgairport a on s.origin=a.iata

join dim\_destairport de on s.dest=de.iata

join dim\_carrier c on s.uniquecarrier=c.carriercode

join dim\_date d on cast(s.year as int)=d.calendaryear

and cast(s.month as int)=d.monthnumberofyear

and cast(s.dayofmonth as int)=d.daynumberofmonth;

end;

'

language 'plpgsql';