**­­­­­­­­­­­­­­­FOR MS SQL**

**//To convert format of trans\_date from varchar to date format(yyyy-MM-dd)**



**1-> SELECT** TRANS\_DATE ,

**convert**(**varchar**(10), **concat**(

'20', **substring**(trans\_date, 7,2),'-',

**substring**(trans\_date, 4,2) ,'-',

**substring**(trans\_date, 1,2) ) ,23) **as** final\_date

**from** Stg\_dl\_slct\_dev.db\_Stgdl.XXSCL\_DEPOT\_STOCK\_csv;

//To update trans\_date2 to trans\_date where trans\_date is in varchar format

**2-> UPDATE** Stg\_dl\_slct\_dev.db\_Stgdl.XXSCL\_DEPOT\_STOCK\_csv

**SET** TRANS\_DATE2 = **convert**(**varchar**(10), **concat**(

'20', **substring**(trans\_date, 7,2),'-',

**substring**(trans\_date, 4,2) ,'-',

**substring**(trans\_date, 1,2) ) ,23)

**where** len(trans\_date)= 8

3-> FOR POSTGRES

//TO IMPORT A CSV FILE IN POSTGRES TABLE

--->>COPY public."SO\_MASTER" FROM 'E:\SO\_MASTER.csv' WITH CSV HEADER;

4-> Converting datetime to character:

|  |  |  |  |
| --- | --- | --- | --- |
| **Without century** | **With century** | **Input/Output** | **Standard** |
| 0 | 100 | mon dd yyyy hh:miAM/PM | Default |
| 1 | 101 | mm/dd/yyyy | US |
| 2 | 102 | yyyy.mm.dd | ANSI |
| 3 | 103 | dd/mm/yyyy | British/French |
| 4 | 104 | dd.mm.yyyy | German |
| 5 | 105 | dd-mm-yyyy | Italian |
| 6 | 106 | dd mon yyyy | - |
| 7 | 107 | Mon dd, yyyy | - |
| 8 | 108 | hh:mm:ss | - |
| 9 | 109 | mon dd yyyy hh:mi:ss:mmmAM (or PM) | Default + millisec |
| 10 | 110 | mm-dd-yyyy | USA |
| 11 | 111 | yyyy/mm/dd | Japan |
| 12 | 112 | yyyymmdd | ISO |
| 13 | 113 | dd mon yyyy hh:mi:ss:mmm | Europe (24 hour clock)> |
| 14 | 114 | hh:mi:ss:mmm | 24 hour clock |
| 20 | 120 | yyyy-mm-dd hh:mi:ss | ODBC canonical (24 hour clock) |
| 21 | 121 | yyyy-mm-dd hh:mi:ss.mmm | ODBC canonical (24 hour clock) |
|  | 126 | yyyy-mm-ddThh:mi:ss.mmm | ISO8601 |
|  | 127 | yyyy-mm-ddThh:mi:ss.mmmZ | ISO8601 (with time zone Z) |
|  | 130 | dd mon yyyy hh:mi:ss:mmmAM | Hijiri |
|  | 131 | dd/mm/yy hh:mi:ss:mmmAM | Hijiri |

Converting float to real:

|  |  |
| --- | --- |
| **Value** | **Explanation** |
| 0 | Maximum 6 digits (default) |
| 1 | 8 digits |
| 2 | 16 digits |

Converting money to character:

|  |  |
| --- | --- |
| **Value** | **Explanation** |
| 0 | No comma delimiters, 2 digits to the right of decimal |
| 1 | Comma delimiters, 2 digits to the right of decimal |
| 2 | No comma delimiters, 4 digits to the right of decimal |

5-> //to select all data from table where date is '2021-03-31'

**SELECT** \*

**from** Stg\_dl\_slct\_dev.db\_Stgdl.XXSCL\_DEPOT\_STOCK\_csv

**where** TRANS\_DATE2 = **cast**('2021-03-31'**as** **date**) ;

6-> // to select all data from table where date is current date(today)

**SELECT** \*

**from** Stg\_dl\_slct\_dev.db\_Stgdl.XXSCL\_DEPOT\_STOCK\_csv

**where** TRANS\_DATE2 = **cast**(getdate() **as** **date**) ;

7-> //to select sum of trans\_qty between today and 30 days before

select sum(TRANS\_QTY)

from Stg\_dl\_slct\_dev.db\_Stgdl.XXSCL\_DEPOT\_STOCK\_csv

where TRANS\_DATE2 between DATEADD(day ,-30, GETDATE() ) and GETDATE()

and LOCATION ='ACCEPT'

8-> //to select trans\_date2 between current date and day before 4 years

**SELECT** TRANS\_DATE2 **from** Stg\_dl\_slct\_dev.db\_Stgdl.XXSCL\_DEPOT\_STOCK\_csv

**where** TRANS\_DATE2 **between** **DATEADD**(**year** ,-4, **GETDATE**()) **and** **GETDATE**()

9-> //to select trans\_date2 between current date and day before 18 months

**SELECT** TRANS\_DATE2 **from** Stg\_dl\_slct\_dev.db\_Stgdl.XXSCL\_DEPOT\_STOCK\_csv

**where** TRANS\_DATE2 **between** **DATEADD**(**month** ,-18, **GETDATE**()) **and** **GETDATE**()

9-> //to select trans\_date2 between current date and day before 550 days

**SELECT** TRANS\_DATE2 **from** Stg\_dl\_slct\_dev.db\_Stgdl.XXSCL\_DEPOT\_STOCK\_csv

**where** TRANS\_DATE2 **between** **DATEADD**(**day** ,-550, **GETDATE**()) **and** **GETDATE**()

10->//to select zone according to state,district, taluka and city even if pincode is null for some cases.

**SELECT** **ZONE**

**FROM** Stg\_dl\_slct\_dev.db\_Stgdl.SCL\_ZONE\_MAPPING\_csv

**WHERE** State = **:p\_state**

**AND** DISTRICT = **:p\_district**

**AND** TALUKA = **:p\_taluka**

**AND** ERP\_CITY = **:p\_erpcity**

**AND** Pincode = ISNULL(**:p\_pincode**,0);

11-> // to get first day of current month

**SELECT** **convert**(**date**,**DATEADD**(**MONTH**,**DATEDIFF**(**month**,0,**GETDATE**()),0))

12->// to get first day of last month

**SELECT** **convert**(**date**,**DATEADD**(MM, **DATEDIFF**(MM, 0, **GETDATE**()) - 1, 0))

13->// to get transcation quantity of current month till date

**SELECT** TRANSACTION\_QUANTITY **AS** Influencer\_Sales

**FROM** Stg\_dl\_slct\_dev.db\_Stgdl.T\_NM\_OMX\_MATERIAL\_TRANSACTIONS\_V

**WHERE** TRANSACTION\_DATE **BETWEEN** **convert**(**date**,**DATEADD**(**MONTH**,**DATEDIFF**(**month**,0,**GETDATE**()),0)) **AND**

**CONVERT**(**date**,**GETDATE**());

14-> to get transaction quantity of last month from first date to last date

**SELECT** TRANSACTION\_QUANTITY **AS** Influencer\_Sales

**FROM** Stg\_dl\_slct\_dev.db\_Stgdl.T\_NM\_OMX\_MATERIAL\_TRANSACTIONS\_V

**WHERE** TRANSACTION\_DATE **BETWEEN** **convert**(**date**,**DATEADD**(MM, **DATEDIFF**(MM, 0, **GETDATE**()) - 1, 0))

**AND** **convert**(**date**,**DATEADD**(**MONTH**,**DATEDIFF**(**month**,0,**GETDATE**()),0)-1)

15->

-- to get first date of current month

**SELECT** **convert**(**date**,**DATEADD**(**MONTH**,**DATEDIFF**(**month**,0,**GETDATE**()),0))

--to get current date of current month

**select** **convert**(**date**,**GETDATE**() )

--to get first date of previous month

**SELECT** **convert**(**date**,**DATEADD**(MM, **DATEDIFF**(MM, 0, **GETDATE**()) - 1, 0))

--to get last date of previous month

**SELECT** **convert**(**date**,**DATEADD**(**MONTH**,**DATEDIFF**(**month**,0,**GETDATE**()),0)-1)

16->**Important**

-- first date of second last month

select convert(date, DATEADD(month ,-2, DATEADD(MONTH,DATEDIFF(month,0,GETDATE()),0) ))

-- last date of second last month

select convert(date,DATEADD(month ,-1, DATEADD(MONTH,DATEDIFF(month,0,GETDATE()),0) )-1)

17->

**SELECT**

**CASE**

**WHEN** INVOICE\_DATE **between** **convert**(**date**, **DATEADD**(**month** ,-2, **DATEADD**(**MONTH**,**DATEDIFF**(**month**,0,**GETDATE**()),0) ))

**AND** **convert**(**date**,**DATEADD**(**month** ,-1, **DATEADD**(**MONTH**,**DATEDIFF**(**month**,0,**GETDATE**()),0) )-1) **THEN** NCR

**ELSE** 0

**END** **AS** Last\_to\_last\_month\_NCR,

**CASE**

**WHEN** INVOICE\_DATE **BETWEEN** **convert**(**date**,**DATEADD**(MM, **DATEDIFF**(MM, 0, **GETDATE**()) - 1, 0))

**AND** **convert**(**date**,**DATEADD**(**MONTH**,**DATEDIFF**(**month**,0,**GETDATE**()),0)-1) **THEN** NCR

**ELSE** 0

**END** **AS** Last\_month\_NCR,

**CASE**

**WHEN** INVOICE\_DATE **BETWEEN** **convert**(**date**,**DATEADD**(**MONTH**,**DATEDIFF**(**month**,0,**GETDATE**()),0))

**AND** **convert**(**date**,**GETDATE**() ) **THEN** NCR

**ELSE** 0

**END** **AS** Current\_month\_NCR

**FROM** db\_Stgdl.T\_OEBS\_SCL\_AR\_NCR\_ADVANCE\_CALC\_TAB

18-> first date of current year and last date of current year

**SELECT**

**convert**(**date**, DATEADD(yy, DATEDIFF(yy, 0, GETDATE()), 0) ) **AS** StartOfYear,

**CONVERT**(**date**, DATEADD(yy, DATEDIFF(yy, 0, GETDATE()) + 1, -1) ) **AS** EndOfYear

19-> first date of previous year

**convert**(**date**, **DATEADD**(yy, **DATEDIFF**(yy, 0, **GETDATE**()) -1,0) ) AS FIRSTDATE

20-> last date of previous year

**convert**(**date**, **DATEADD**(yy, **DATEDIFF**(yy, 0, **GETDATE**()) +0,-1) ) AS LASTDATE

21->

SELECT **DISTINCT**

**CASE**

**WHEN** xcm.State = 'DELHI' **THEN** 'NORTH1'

**WHEN** xcm.State != 'DELHI' **THEN**(

**CASE**

**WHEN** xcm.DISTRICT = szm.DISTRICT **THEN** (

**CASE**

**WHEN** szm.**ZONE** = ' ' **THEN** 'UNKNOWN'

**ELSE** szm.**zone**

**END**

)

**END**

)

**END** **AS** **ZONE**

**FROM** Stg\_dl\_slct\_dev.db\_Stgdl.T\_OEBS\_SCL\_AR\_NCR\_ADVANCE\_CALC\_TAB xcm,

Stg\_dl\_slct\_dev.db\_Stgdl.SCL\_ZONE\_MAPPING\_csv szm

**WHERE** xcm.DISTRICT = szm.DISTRICT

22-> filter to know data of which period or time span

**SELECT**

**CASE**

**WHEN** OE\_CREATION\_DT = **convert**(**date**,**getdate**()) **THEN** 'TODAY'

**WHEN** OE\_CREATION\_DT != **convert**(**date**,**getdate**()) **THEN** (

**CASE**

**WHEN** OE\_CREATION\_DT **between** **convert**(**date**,**DATEADD**(**MONTH**,**DATEDIFF**(**month**,0,**GETDATE**()),0))

**AND** **convert**(**date**,**getdate**()-1) **THEN** 'MTD'

**ELSE** (

**CASE**

**WHEN** OE\_CREATION\_DT **between** **convert**(**date**, **DATEADD**(yy, **DATEDIFF**(yy, 0, **GETDATE**()) +0,0) )

**AND** **convert**(**date**,**DATEADD**(**MONTH**,**DATEDIFF**(**month**,0,**GETDATE**()),-1)) **THEN** 'CURRENT\_YEAR\_EXCEPT\_MTD'

**ELSE** 'PREVIOUS\_DATA'

**END**

)

**END**

)

**END** **AS** BUCKET\_FILTER

**FROM** Stg\_dl\_slct\_dev.db\_Stgdl.Bangur\_Dispatch\_Data bdd

20-> to update oe-creation date from varchar to date format

UPDATE Stg\_dl\_slct\_dev.db\_Stgdl.Bangur\_Dispatch\_Data

SET OE\_CREATION\_DT = convert(varchar(10), concat(

'20', substring(OE\_CREATION\_DT, 7,2),'-',

substring(OE\_CREATION\_DT, 4,2) ,'-',

substring(OE\_CREATION\_DT, 1,2) ) ,23)

where len(OE\_CREATION\_DT)= 8,

21-> TEST FOR ZONE MAPPING

**create** **function** getZone(**@State** **as** **varchar**(100),**@district** **as** **varchar**(100))

**returns** **varchar**(100)

**as**

**begin**

**declare** **@Zone** **AS** **varchar**(100);

**Select** **@Zone**

**select** **top** 1

(**CASE**

**WHEN** **@State**=ISNULL('DELHI',**NULL**) **THEN** 'NORTH1'

**WHEN** **@State** != 'DELHI' **THEN**(

**CASE**

**WHEN** szm.DISTRICT = **@district** **THEN** (

**CASE**

**WHEN** szm.**ZONE** != ' ' **THEN** szm.**zone**

**ELSE** 'UNKNOWN'

**END**

)

**END**

)

**END**)

**FROM** Stg\_dl\_slct\_dev.db\_Stgdl.SCL\_ZONE\_MAPPING\_csv szm);

**return** **@Zone**;

**end** ;

22-> handled cases for zone mapping (but not handled)

**create** **function** getZone\_test(**@State** **as** **varchar**(100),**@district** **as** **varchar**(100))

**returns** **varchar**(100)

**as**

**begin**

**declare** **@Zone** **AS** **varchar**(100);

**Select** **@Zone** = (

**select** **top** 1

(**CASE**

**WHEN** **@State**= szm.State **THEN** (

**CASE**

**WHEN** szm.**ZONE** ! = '' **THEN** szm.**ZONE**

**ELSE** 'UNKNOWN'

**END**

)

**WHEN** **@State** != szm.State **THEN**(

**CASE**

**WHEN** szm.DISTRICT = **@district** **THEN** (

**CASE**

**WHEN** szm.**ZONE** != ' ' **THEN** szm.**zone**

**ELSE** 'UNKNOWN'

**END**

)

**ELSE** 'UNKNOWN'

**END**

)

**END**)

**FROM** Stg\_dl\_slct\_dev.db\_Stgdl.SCL\_ZONE\_MAPPING\_csv szm);

**return** **@Zone**;

**end** ;

23->

**create** **function** getZone\_test(**@State** **as** **varchar**(100),**@district** **as** **varchar**(100))

**returns** **varchar**(100)

**as**

**begin**

**declare** **@Zone** **AS** **varchar**(100);

**Select** **@Zone** = (

**select** **top** 1

(**CASE**

**WHEN** **@State**= szm.State **THEN** (

**CASE**

**WHEN** szm.**ZONE** ! = '' **THEN** szm.**ZONE**

**ELSE** (

**CASE**

**WHEN** szm.DISTRICT = **@district** **THEN** (

**CASE**

**WHEN** szm.**ZONE** != ' ' **THEN** szm.**zone**

**ELSE** 'UNKNOWN'

**END**

)

**END**

)

**END**

)

**ELSE** (

**CASE**

**WHEN** **@State** = 'DELHI' **THEN** 'NORTH1'

**ELSE** 'UNKNOWN'

**END**

)

**END**

)

**FROM** Stg\_dl\_slct\_dev.db\_Stgdl.SCL\_ZONE\_MAPPING\_csv szm);

**return** **@Zone**;

**end** ;

CHECK FOR STATE IF IT’S NULL

NO

If state is delhi THEN

Set ZONE = ‘NORTH1’

ELSE

IF ZONE IS NULL OR NOT FOUND THEN

SET ZONE = ‘UNKNOWN’

Else

SET ZONE =szm.ZONE

YES

IF DISTRICT is Not NULL THEN

Get the ZONE (szm.ZONE for given district)

IF ZONE IS null OR NOT FOUND THEN

SET ZONE= ‘UNKNOWN’

ELSE

SET ZONE= ‘SZM.zone’  
 else

SET ZONE= ‘UNKNOWN’

**22->**

**create** **function** getZone\_test(**@State** **as** **varchar**(100),**@district** **as** **varchar**(100))

**returns** **varchar**(100)

**as**

**begin**

**declare** **@Zone** **AS** **varchar**(100);

**Select** **@Zone** = (

**select** **top** 1

(**CASE**

**WHEN** szm.State != ' ' **THEN**

(

**CASE**

**WHEN** **@State**= 'DELHI' **THEN** 'NORTH1'

**ELSE** (

**CASE**

**WHEN** szm.**ZONE** != '' **THEN** szm.**ZONE**

**ELSE** 'UNKNOWN'

**END**

)

**END**

)

**ELSE** (

**CASE**

**WHEN** szm.district ! = '' **THEN** (

**CASE**

**WHEN** szm.district= **@district** **THEN** (

**CASE**

**WHEN** szm.**ZONE** != '' **THEN** szm.**ZONE**

**ELSE** 'UNKNOWN'

**END** )

**ELSE** 'UNKNOWN'

**END**

)

**END**

)

**END** )

**FROM** Stg\_dl\_slct\_dev.db\_Stgdl.SCL\_ZONE\_MAPPING\_csv szm);

**return** **@Zone**;

**end** ;

23->to convert in date-time

**SELECT** GATE\_ENTRY\_TIME,

**convert**(**varchar**(5000), **concat**(

**substring**(GATE\_ENTRY\_TIME, 7,4),'-',

**substring**(GATE\_ENTRY\_TIME, 4,2) ,'-',

**substring**(GATE\_ENTRY\_TIME, 1,2),' ',

**substring**(GATE\_ENTRY\_TIME, 12,5),' ') ,120) **as** final\_date

**from** Stg\_dl\_slct\_dev.db\_Stgdl.TAT\_DATA\_JUL21\_DEC21 ;

**24->**

**UPDATE** Stg\_dl\_slct\_dev.db\_Stgdl.TAT\_DATA\_JUL21\_DEC21

**SET** GATE\_ENTRY\_TIME = **convert**(**varchar**(5000), **concat**(

**substring**(GATE\_ENTRY\_TIME, 7,4),'-',

**substring**(GATE\_ENTRY\_TIME, 4,2) ,'-',

**substring**(GATE\_ENTRY\_TIME, 1,2),' ',

**substring**(GATE\_ENTRY\_TIME, 12,5),' ') ,120)

**WHERE** len(GATE\_ENTRY\_TIME) = 17

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | select convert(varchar, getdate(), 0) | Mon dd yyyy hh:mm AM/PM | Dec 30 2006 12:38AM |
| 9 | select convert(varchar, getdate(), 9) | Mon dd yyyy hh:mm:ss:nnn AM/PM | Dec 30 2006 12:38:54:840AM |
| 13 | select convert(varchar, getdate(), 13) | dd Mon yyyy hh:mm:ss:nnn AM/PM | 30 Dec 2006 00:38:54:840AM |
| 20 | select convert(varchar, getdate(), 20) | yyyy-mm-dd hh:mm:ss | 2006-12-30 00:38:54 |
| 21 | select convert(varchar, getdate(), 21) | yyyy-mm-dd hh:mm:ss:nnn | 2006-12-30 00:38:54.840 |
| 22 | select convert(varchar, getdate(), 22) | mm/dd/yy hh:mm:ss AM/PM | 12/30/06 12:38:54 AM |
| 25 | select convert(varchar, getdate(), 25) | yyyy-mm-dd hh:mm:ss:nnn | 2006-12-30 00:38:54.840 |
| 100 | select convert(varchar, getdate(), 100) | Mon dd yyyy hh:mm AM/PM | Dec 30 2006 12:38AM |
| 109 | select convert(varchar, getdate(), 109) | Mon dd yyyy hh:mm:ss:nnn AM/PM | Dec 30 2006 12:38:54:840AM |
| 113 | select convert(varchar, getdate(), 113) | dd Mon yyyy hh:mm:ss:nnn | 30 Dec 2006 00:38:54:840 |
| 120 | select convert(varchar, getdate(), 120) | yyyy-mm-dd hh:mm:ss | 2006-12-30 00:38:54 |
| 121 | select convert(varchar, getdate(), 121) | yyyy-mm-dd hh:mm:ss:nnn | 2006-12-30 00:38:54.840 |
| 126 | select convert(varchar, getdate(), 126) | yyyy-mm-dd T hh:mm:ss:nnn | 2006-12-30T00:38:54.840 |
| 127 | select convert(varchar, getdate(), 127) | yyyy-mm-dd T hh:mm:ss:nnn | 2006-12-30T00:38:54.840 |
|  |  |  |  |
| **ISLAMIC CALENDAR DATES** | | | |
| 130 | select convert(nvarchar, getdate(), 130) | dd mmm yyyy hh:mi:ss:nnn AM/PM | date output |
| 131 | select convert(nvarchar, getdate(), 131) | dd mmm yyyy hh:mi:ss:nnn AM/PM | 10/12/1427 12:38:54:840AM |

You can also format the date or time without dividing characters, as well as concatenate the date and time string:

|  |  |  |
| --- | --- | --- |
| **Sample statement** | **Format** | **Output** |
| select replace(convert(varchar, getdate(),101),'/','') | mmddyyyy | 12302006 |
| select replace(convert(varchar, getdate(),101),'/','') + replace(convert(varchar, getdate(),108),':','') | mmddyyyyhhmmss | 12302006004426 |

24-> to convert format of today in hours

**SELECT** FORMAT(**GETDATE**(), 'HH:mm')

25-> to get datediff (IMPORTANT) ye wala date difference ke liye hai jab column varchar mein ho

**select** **datediff**(**second**,(**convert**(**varchar**(5000), **concat**(

**substring**(GATE\_ENTRY\_TIME, 7,4),'-',

**substring**(GATE\_ENTRY\_TIME, 4,2) ,'-',

**substring**(GATE\_ENTRY\_TIME, 1,2),' ',

**substring**(GATE\_ENTRY\_TIME, 12,5),' ') ,120)),(**convert**(**varchar**(5000), **concat**(

**substring**(GATE\_EXIT\_TIME, 7,4),'-',

**substring**(GATE\_EXIT\_TIME, 4,2) ,'-',

**substring**(GATE\_EXIT\_TIME, 1,2),' ',

**substring**(GATE\_EXIT\_TIME, 12,5),' ') ,120)) )

**from** Stg\_dl\_slct\_dev.db\_Stgdl.TAT\_DATA\_JUL21\_DEC21

24-> CONVERTING DATETIME TO TIME (here tt.BID\_ALLOCATION\_TIME is in datettime format of TABLE TR)

(convert(time,tt.BID\_ALLOCATION\_TIME))

25-> TO CONVERT VARCHAR TO DECIMAL

CAST (PRIMARY\_TRANSACTION\_QUANTITY as decimal)

26-> FUNCTION TO GET DATE DIFFERENCE IN DATE HOUR:MINUTE:SECOND

**CREATE** **FUNCTION** getDateDiff\_WithDate(**@startDate** **DATETIME**, **@endDate** **DATETIME**)

**RETURNS** **VARCHAR**(10)

**AS** **BEGIN**

**DECLARE** **@seconds** **INT** = **DATEDIFF**(s, **@startDate**, **@endDate**)

**DECLARE** **@difference** **VARCHAR**(10) =

**CONVERT**(**VARCHAR**(8), **@seconds** / (3600\*24)) + ':' +

**CONVERT**(**VARCHAR**(4), **@seconds** %(3600\*24)/ 3600) + ':' +

**CONVERT**(**VARCHAR**(2), **@seconds** % 3600 / 60) + ':' +

**CONVERT**(**VARCHAR**(2), **@seconds** % 60)

**RETURN** **@difference**

**END**

**27-> QUERY TO GET DATE hh:mm:ss difference**

**SELECT** **DISTINCT**

(**CASE**

WHEN

((**DATEDIFF**(**MINUTE**, tt.DILINK, tt.PPCAL\_IN)/(24\*60))>0) **THEN** (

**CONCAT**(

(**DATEDIFF**(**MINUTE**, tt.DILINK, tt.PPCAL\_IN)/(24\*60)),' ',

(**CASE**

**WHEN**

((**DATEDIFF**(**MINUTE**, tt.DILINK, tt.PPCAL\_IN)/60)>0) **THEN** (

**CONCAT**(((**DATEDIFF**(**MINUTE**, tt.DILINK, tt.PPCAL\_IN)/60)),':',

(**DATEDIFF**(**MINUTE**, tt.DILINK, tt.PPCAL\_IN))%60)

)

**ELSE**

**CONCAT** ('00',':',((**DATEDIFF**(**MINUTE**, tt.DILINK, tt.PPCAL\_IN))%60))

**END** )

)

)

**ELSE** (

(**CASE**

**WHEN**

((**DATEDIFF**(**MINUTE**, tt.DILINK, tt.PPCAL\_IN)/60)>0) **THEN** (

**CONCAT**('00',' ',((**DATEDIFF**(**HOUR**, tt.DILINK, tt.PPCAL\_IN))),':',

(**DATEDIFF**(**MINUTE**, tt.DILINK, tt.PPCAL\_IN))%60)

)

**ELSE**

**CONCAT** ('00', ' ','00',':',((**DATEDIFF**(**MINUTE**, tt.DILINK, tt.PPCAL\_IN))%60))

**END** )

)

**END**

)

**FROM** Stg\_dl\_slct\_dev.db\_Stgdl.TRUCK\_TAT\_N tt

**WHERE** ((**DATEDIFF**(**MINUTE**, tt.DILINK, tt.PPCAL\_IN))>0)

28-> to avoid problem of overflow

CREATE FUNCTION getDateDiff\_WithDate(@startDate DATETIME, @endDate DATETIME)

RETURNS VARCHAR(5000)

AS BEGIN

DECLARE @seconds INT = DATEDIFF(s, convert(time,@startDate), convert(time, @endDate))

DECLARE @difference VARCHAR(5000) =

CONVERT(VARCHAR(100), @seconds / (3600\*24)) + ' ' +

CONVERT(VARCHAR(100), @seconds %(3600\*24)/ 3600) + ':' +

CONVERT(VARCHAR(100), @seconds % 3600 / 60) + ':' +

CONVERT(VARCHAR(100), @seconds % 60)

RETURN @difference

END;

**29-> to convert date in format MM-YYYY**

**SELECT**

**convert**(**varchar**(5000), **concat**(

**substring**(tomoqd.ORIG\_DATE\_RECEIVED, 6,2),'-',

**substring**(tomoqd.ORIG\_DATE\_RECEIVED, 1,4)

) ,105) **as** final\_dates

**from** Stg\_dl\_slct\_dev.db\_Stgdl.T\_OEBS\_MTL\_ONHAND\_QUANTITIES\_DETAIL tomoqd ;

28-> to convert varchar datatype to date

ALTER TABLE Stg\_dl\_slct\_dev.db\_Stgdl.T\_OEBS\_MTL\_ONHAND\_QUANTITIES\_DETAIL ALTER COLUMN ORIG\_DATE\_RECEIVED date ;

29-> to check datatype of any result or function

SQL\_VARIANT\_PROPERTY(getdate(), 'BaseType')

30-> to get difference of dates in time

**CREATE** **FUNCTION** getTimeDiffInTime(**@startDate** **DATETIME**, **@endDate** **DATETIME**)

**RETURNS** **time**

**AS** **BEGIN**

**DECLARE** **@seconds** **INT** = **DATEDIFF**(s, **@startDate**, **@endDate**)

**DECLARE** **@difference** **time** =

**CONVERT**(**VARCHAR**(4), **@seconds** / 3600) + ':' +

**CONVERT**(**VARCHAR**(2), **@seconds** % 3600 / 60) + ':' +

**CONVERT**(**VARCHAR**(2), **@seconds** % 60)

**RETURN** **@difference**

**END**

**31-> function to get first day of current month**

**CREATE** **FUNCTION** getFirstDayOfMonth()

returns **date**

**AS** **BEGIN**

**DECLARE** **@result** date = convert(**date**,**DATEADD**(**MONTH**,**DATEDIFF**(month,0,**GETDATE**()),0))

**RETURN** **@result**

END

**32-> function to get first day of previous month**

**create** **function** getFirstDayOfPreviousMonth()

**returns** **date**

**AS** **BEGIN**

**DECLARE** **@result** **date** = **convert**(**date**,**DATEADD**(**MONTH**,**DATEDIFF**(**month**,0,**GETDATE**())-1,0))

**RETURN** **@result**

**END**

**33-> function to get last day of previous month**

**create** **function** getLastDayOfPreviousMonth()

**returns** **date**

**AS** **BEGIN**

**DECLARE** **@result** **date** = **convert**(**date**,**DATEADD**(**MONTH**,**DATEDIFF**(**month**,0,**GETDATE**()),0)-1)

**RETURN** **@result**

**END**

**34-> to convert in ‘Oct 2021’ format**

**SELECT** FORMAT (tomoqd.ORIG\_DATE\_RECEIVED, 'MMM yyyy') **from** Stg\_dl\_slct\_dev.db\_Stgdl.T\_OEBS\_MTL\_ONHAND\_QUANTITIES\_DETAIL tomoqd

35->Test for Zone Mapping

**create** **function** getZone\_test4(**@State** **as** **varchar**(100),**@district** **as** **varchar**(100))

**returns** **varchar**(100)

**as**

**begin**

**declare** **@Zone** **AS** **varchar**(5000)

**SELECT** **@ZONE** = (

**SELECT** **TOP** 1 (**CASE**

**WHEN**

**@State** = szm.State **THEN** szm.**ZONE**

**ELSE**(

**CASE**

**WHEN** **@district** = szm.DISTRICT **THEN** szm.**ZONE**

**ELSE** 'UNKNOWN'

**END**)

**END** )

**FROM** Stg\_dl\_slct\_dev.db\_Stgdl.SCL\_ZONE\_MAPPING\_csv szm

);

**return** ISNULL(**@Zone**,'UNKNOWN');

**end** ;

36-> to select month till date of previous year on Invoice date of ncr\_calc\_tab

**SELECT** INVOICE\_DATE

**from** Stg\_dl\_slct\_dev.db\_Stgdl.T\_OEBS\_SCL\_AR\_NCR\_ADVANCE\_CALC\_TAB tosanact

**where** INVOICE\_DATE **BETWEEN** **convert**(**date**,**DATEADD**(**MONTH** ,**DATEDIFF**(**MONTH**,0,**GETDATE**())-12,0)) **AND** **convert**(**date**,**DATEADD**(**YEAR** ,-1, **GETDATE**()) )

37->

--DROP function getZone

**create** **function** getZone(**@State** **as** **varchar**(100),**@district** **as** **varchar**(100))

**returns** **varchar**(100)

**as**

**begin**

**declare** **@Zone** **AS** **varchar**(100);

**Select** **@Zone** = (

**SELECT** **TOP** 1

(**CASE**

**WHEN** **@State** = 'DELHI' **THEN** 'NORTH1'

**WHEN** ((**@State** != ' ') **AND** (**@State** **IS** **NOT** **NULL**)) **THEN**(

**SELECT** ISNULL((**SELECT** **TOP** 1 szm1.**ZONE**

**FROM** Stg\_dl\_slct\_dev.db\_Stgdl.SCL\_ZONE\_MAPPING\_csv szm1

**WHERE** szm1.State = **@State**),

( **CASE**

**WHEN** ((**@district** != ' ') **AND** (**@district** **IS** **NOT** **NULL**)) **THEN**(

**SELECT** **TOP** 1 szm2.**ZONE**

**FROM** Stg\_dl\_slct\_dev.db\_Stgdl.SCL\_ZONE\_MAPPING\_csv szm2

**WHERE** szm2.DISTRICT = **@district**

)

**ELSE** 'UNKNOWN'

**END**

)

)

)

**ELSE**

( **CASE**

**WHEN** ((**@district** != ' ') **AND** (**@district** **IS** **NOT** **NULL**)) **THEN**(

**SELECT** **TOP** 1 szm2.**ZONE**

**FROM** Stg\_dl\_slct\_dev.db\_Stgdl.SCL\_ZONE\_MAPPING\_csv szm2

**WHERE** szm2.DISTRICT = **@district**

)

**ELSE** 'UNKNOWN'

**END**

)

**END**

)

**FROM** Stg\_dl\_slct\_dev.db\_Stgdl.SCL\_ZONE\_MAPPING\_csv szm);

**if** (**@Zone** **is** **null**)

**BEGIN**

**SET** **@Zone** = 'UNKNOWN';

**END**

**return** **@Zone**;

**end** ;

CREATE VIEW ORG\_ACCT\_PERIODS#

AS