Enhancing Time Zone-Sensitive Business Engagement: SQL Logic for Business Hour Validation by Time Zone

In the context of Patient SMS, Voice, and Email engagement, I was responsible for ensuring that engagement notifications are only sent to patients during designated business hours (9 AM to 5 PM) in their local timezone. This proactive approach prevents off-hours notifications and enhances the overall patient experience.

Create the Time Zone Table

```
drop table if exists [dbo].[state_timezone]
go


set ansi_nulls on
go

set quoted_identifier on
go

create table [dbo].[state_timezone]
(
  [state_cd] [varchar](50) not null,
  [state_name] [varchar](50) not null,
  [timezone] [varchar](50) not null,
  [offset] [varchar](50) not null,
  [tz_abbv] [varchar](4) null,
 constraint [pk_state_timezone] primary key clustered
(
  [state_cd] asc
)with (pad_index = off, statistics_norecompute = off, ignore_dup_key = off,
allow_row_locks = on, allow_page_locks = on, optimize_for_sequential_key = off) on
[primary]
) on [primary]
go
```

Populate The Time Zone Table

I downloaded a dataset containing state abbreviations, names, descriptions, offsets, and time zones from a public domain source. I then refined the data using Excel to prepare it for import into a SQL table. This is the query I produced to populate the Time Zone Table.

```
insert into [dbo].[state_timezone] values('AK', 'Alaska', 'Alaskan Standard Time', '-09:00', 'AKST')
insert into [dbo].[state_timezone] values('AL', 'Alabama', 'Central Standard Time', '-06:00', 'CST')
insert into [dbo].[state_timezone] values('AR', 'Arkansas', 'Central Standard Time', '-06:00', 'CST')
insert into [dbo].[state_timezone] values('AZ', 'Arizona', 'Mountain Standard Time', '-07:00', 'MST')
insert into [dbo].[state_timezone] values('CA', 'California', 'Pacific Standard Time', '-08:00', 'PST')
insert into [dbo].[state_timezone] values('CO', 'Colorado', 'Mountain Standard Time', '-07:00', 'MST')
insert into [dbo].[state_timezone] values('CT', 'Connecticut', 'Eastern Standard Time', '-05:00', 'EST')
insert into [dbo].[state_timezone] values('DC', 'District of Columbia', 'Eastern Standard Time', '-05:00', 'EST')
insert into [dbo].[state_timezone] values('DE', 'Delaware', 'Eastern Standard Time', '-05:00', 'EST')
insert into [dbo].[state_timezone] values('FL', 'Florida', 'Eastern Standard Time', '-05:00', 'EST')
insert into [dbo].[state_timezone] values('GA', 'Georgia', 'Eastern Standard Time', '-04:00', 'EST')
insert into [dbo].[state_timezone] values('HI', 'Hawaii', 'Hawaiian Standard Time', '-10:00', 'HAST')
insert into [dbo].[state_timezone] values('IA', 'Iowa', 'Central Standard Time', '-06:00', 'CST')
insert into [dbo].[state_timezone] values('ID', 'Idaho', 'Mountain Standard Time', '-07:00', 'MST')
insert into [dbo].[state_timezone] values('IL', 'Illinois', 'Central Standard Time', '-06:00', 'CST')
insert into [dbo].[state_timezone] values('IN', 'Indiana', 'Eastern Standard Time', '-05:00', 'EST')
insert into [dbo].[state_timezone] values('KS', 'Kansas', 'Central Standard Time', '-06:00', 'CST')
insert into [dbo].[state_timezone] values('KY', 'Kentucky', 'Eastern Standard Time', '-05:00', 'EST')
insert into [dbo].[state_timezone] values('LA', 'Louisiana', 'Central Standard Time', '-06:00', 'CST')
insert into [dbo].[state_timezone] values('MA', 'Massachusetts', 'Eastern Standard Time', '-05:00', 'EST')
insert into [dbo].[state_timezone] values('MD', 'Maryland', 'Eastern Standard Time', '-05:00', 'EST')
insert into [dbo].[state_timezone] values('ME', 'Maine', 'Eastern Standard Time', '-05:00', 'EST')
insert into [dbo].[state_timezone] values('MI', 'Michigan', 'Eastern Standard Time', '-05:00', 'EST')
insert into [dbo].[state_timezone] values('MN', 'Minnesota', 'Central Standard Time', '-06:00', 'CST')
insert into [dbo].[state_timezone] values('MO', 'Missouri', 'Central Standard Time', '-06:00', 'CST')
insert into [dbo].[state_timezone] values('MS', 'Mississippi', 'Central Standard Time',
'-06:00', 'CST')
insert into [dbo].[state_timezone] values('MT', 'Montana', 'Mountain Standard Time', '-
07:00', 'MST')
insert into [dbo].[state_timezone] values('NC', 'North Carolina', 'Eastern Standard
Time', '-05:00', 'EST')
insert into [dbo].[state_timezone] values('ND', 'North Dakota', 'Central Standard Time',
'-06:00', 'CST')
insert into [dbo].[state_timezone] values('NE', 'Nebraska', 'Central Standard Time', '-
06:00', 'CST')
insert into [dbo].[state_timezone] values('NH', 'New Hampshire', 'Eastern Standard Time',
'-05:00', 'EST')
insert into [dbo].[state_timezone] values('NJ', 'New Jersey', 'Eastern Standard Time', '-
05:00', 'EST')
insert into [dbo].[state_timezone] values('NM', 'New Mexico', 'Mountain Standard Time',
'-07:00', 'MST')
insert into [dbo].[state_timezone] values('NV', 'Nevada', 'Pacific Standard Time', '-
08:00', 'PST')
insert into [dbo].[state_timezone] values('NY', 'New York', 'Eastern Standard Time', '-
05:00', 'EST')
insert into [dbo].[state_timezone] values('OH', 'Ohio', 'Eastern Standard Time', '-
05:00', 'EST')
insert into [dbo].[state_timezone] values('OK', 'Oklahoma', 'Central Standard Time', '-
06:00', 'CST')
insert into [dbo].[state_timezone] values('OR', 'Oregon', 'Pacific Standard Time', '-
08:00', 'PST')
insert into [dbo].[state_timezone] values('PA', 'Pennsylvania', 'Eastern Standard Time',
'-05:00', 'EST')
insert into [dbo].[state_timezone] values('RI', 'Rhode Island', 'Eastern Standard Time',
'-05:00', 'EST')
insert into [dbo].[state_timezone] values('SC', 'South Carolina', 'Eastern Standard
Time', '-05:00', 'EST')
insert into [dbo].[state_timezone] values('SD', 'South Dakota', 'Central Standard Time',
'-06:00', 'CST')
insert into [dbo].[state_timezone] values('TN', 'Tennessee', 'Eastern Standard Time', '-
06:00', 'EST')
insert into [dbo].[state_timezone] values('TX', 'Texas', 'Central Standard Time', '-
06:00', 'CST')
insert into [dbo].[state_timezone] values('UT', 'Utah', 'Mountain Standard Time', '-
07:00', 'MST')
insert into [dbo].[state_timezone] values('VA', 'Virginia', 'Eastern Standard Time', '-
05:00', 'EST')
insert into [dbo].[state_timezone] values('VT', 'Vermont', 'Eastern Standard Time', '-
05:00', 'EST')
insert into [dbo].[state_timezone] values('WA', 'Washington', 'Pacific Standard Time', '-
08:00', 'PST')
insert into [dbo].[state_timezone] values('WI', 'Wisconsin', 'Central Standard Time', '-
06:00', 'CST')
insert into [dbo].[state_timezone] values('WV', 'West Virginia', 'Eastern Standard Time',
'-05:00', 'EST')
insert into [dbo].[state_timezone] values('WY', 'Wyoming', 'Mountain Standard
Time', '-06:00', 'MST')
```

Create Get Time Zone by State Function

Create a function that checks if the current time in a patient's local timezone falls within a specified range of business hours (9 AM to 5 PM) using the SQL SYSDATETIMEOFFSET() function to obtain the current date and time as a datetimeoffset data type from the SQL Server. This function provides both the local date and time along with the corresponding time zone offset, enabling accurate management of dates and times across various time zones. The time zone offset is a column in the Time Zone Table.

```
drop function if exists [dbo].[fn_GetTimeZoneByStateCD]
go

set ansi_nulls on
go

set quoted_identifier on
go


create function [dbo].[fn_GetTimeZoneByStateCD](@state_cd varchar(2))
returns varchar(5) as
  begin
    declare @ret_val varchar(5) = 'false'

-- define the start hour and end hour
-- the start hour and end hour values are hardcoded for demonstration purposes only
-- in a real application, the values would typically be retrieved dynamically from
-- a data source, such as a SQL query
    declare @starthour int = 8 -- 9am (SQL 24hr format)
    declare @endhour int = 18 -- 5pm (SQL 24hr format)

    declare @LocalServerTimeZone varchar(50) = null
    select @LocalServerTimeZone = [timezone] from [dbo].[state_timezone] where [state_cd]
= @state_cd
    declare @ServerDateTimeOffset datetimeoffset = sysdatetimeoffset(); -- timezone
offset of the SQL server
    declare @PatientTimeZoneDateTime datetimeoffset = @ServerDateTimeOffset at time zone
'UTC' at time zone @LocalServerTimeZone; -- patient timezone
    declare @CurrentServerDateTimeZone datetimeoffset = sysdatetimeoffset() -- '2024-05-
29 12:00:00 -05:00';
    declare @pat_tz_hour int = datepart(hour, @PatientTimeZoneDateTime) -- (SQL 24hr
format) - any time between 9am and 5pm for the patient timezone

    if (@pat_tz_hour > @starthour) and (@pat_tz_hour < @endhour)
    begin
      set @ret_val = 'true'
    end

    return(@ret_val)
  end
go
```

Description of the Function

The SQL function determines if the current time in a patient's local timezone is within business hours (9 AM to 5 PM) based on the Patient's State Abbreviation as part of their address in my database. It does this by fetching the timezone for a specified patient's state, converting the server's current time to the patient's local time, and checking if that hour falls within the defined range. The result is returned as a string value, either 'true' or 'false', indicating whether the current time is suitable for business operations.

Breakdown of the Function

Variable Declarations: @ret_val: A variable initialized to 'false', which will be used to indicate whether the current time is within business hours.

@starthour: An integer set to 8, representing the start hour for business operations (9 AM in 24-hour format).

@endhour: An integer set to 18, representing the end hour for business operations (5 PM in 24-hour format).

@LocalServerTimeZone: A variable to store the timezone of the patient's state, initially set to null.

Fetching Local Time Zone: A select statement retrieves the timezone associated with a specific state (@state_cd) from the state_timezone table and assigns it to @LocalServerTimeZone.

Date and Time Handling: @ServerDateTimeOffset: Captures the current date and time along with the timezone offset of the SQL Server using sysdatetimeoffset().

@PatientTimeZoneDateTime: Converts the server's current time to the patient's local timezone by first converting to UTC and then to the patient's local timezone using AT TIME ZONE.

@CurrentServerDateTimeZone: Simply fetches the current date and time from the SQL Server again (this variable seems to be redundant since it's not used later in the code).

@pat_tz_hour: Extracts the hour part (in 24-hour format) from @PatientTimeZoneDateTime.

Business Hours Check: An IF statement checks if the current hour in the patient's timezone (@pat_tz_hour) is greater than the start hour (@starthour, 9 AM) and less than the end hour (@endhour, 5 PM).

If the condition is met, it sets @ret_val to 'true', indicating that the current time is within business hours.

Return Statement: Finally, the function returns the value of @ret_val, which will be either 'true' or 'false', depending on whether the current time is within the defined business hours.

Sample Query Using the Function

```sql
with cte_rec_to_process as
(
  select top(1)
  [holding_tbl].[pkid],
  [holding_tbl].[order_id],
  [holding_tbl].[pat_id],
  [holding_tbl].[destination],
  [holding_tbl].[store_id],
  [holding_tbl].[date_created],
  [holding_tbl].[record_processed],
  [holding_tbl].[date_record_processed],
  [holding_tbl].[msg_cstore_param] as [msg_cstore_param],
  [holding_tbl].[twilio_from_number],
  [address].[state_cd],
  -- use address type 8 if it exists else use address type 2
  row_number() over (partition by [holding_tbl].[pkid] order by case when
[shortaddr].[addr_type_cn] = 8 then 1 else 2 end, [shortaddr].[addr_type_cn] desc) as
[rownum],
  [dbo].[fn_GetTimeZoneByStateCD]([address].[state_cd]) as [okay_to_send] -- is it within
the send time window for the specified [address].[state_cd]
  from [dbo].[MyProcessHoldingTable] as [holding_tbl]

  -- address info
  inner join [dbo].[myaddress] as [shortaddr] on [holding_tbl].[pat_id] =
[shortaddr].[pat_id]
  inner join [dbo].[theaddrid] as [address] on [shortaddr].[addr_id] =
[address].[addr_id]
  inner join [dbo].[state_timezone] as [tz] on [address].[state_cd] = [tz].[state_cd]

  -- only shipping or home address
  where [shortaddr].[addr_type_cn] in (8,2) -- select * from csct_code where ct_id = 37 -
- code_num = 8 = SHIPPING and 2 = home
  and [holding_tbl].[record_processed] = 0

  -- only get records where [okay_to_send_tf] = true
  -- see [dbo].[state_timezone] table
  and [dbo].[fn_GetTimeZoneByStateCD]([address].[state_cd]) = 'true'
)

select
[pkid],
[order_id],
[pat_id],
[destination],
[store_id],
[date_created],
[record_processed],
[date_record_processed],
[msg_cstore_param],
[twilio_from_number]
from cte_rec_to_process
```

Conclusion

The query identifies unprocessed records from the holding table and checks whether it is an appropriate time to send a notification to each patient based on their local timezone. By using a common table expression (CTE), it retrieves a single qualifying address (preferring a 'Shipping' address if available) and evaluates the timezone criteria using the function fn_GetTimeZoneByStateCD. Only records that meet the business hour conditions for notifications are returned, ensuring that communications are sent at suitable times."

http://www.seandrew.info - https://www.linkedin.com/in/seanmdrew