

5.3.7

- DW structure changes
 - added D_DelegatedState table with fields:
 - DelegatedStateId
 - DelegatedStateName
 - added F_InteractionDelegated table with fields:
 - Id
 - InteractionGUID
 - InteractionTypeid
 - BusinessProcessId
 - Converser
 - Details
 - InitiatorAgentId
 - InitiatorBusinessProcessId
 - InitiatorName
 - EntryDateTime
 - UpdateDateTime
 - CampaignId
 - Remarks
 - DelegatedStateId
 - InteractionMediaTypeId
 - IsCallback
 - InteractionStateId
 - SupervisorCloseMessage
 - CustomerId
 - CreatedLogId
 - UpdatedLogId
 - F_Interactions table new fields:
 - CustomerOptIn
 - PostAgentTimeSec
- New Dataset Common_ChangedBy_DW:
 - Parameters:
 - @DW_DBName
 - Fields:
 - ChangedBy
- 1.01 Detailed Agent Activity:
 - Parameter: @DisplayLanguage
 - parameter removed
- 1.02 Agent Detailed Alerts:
 - Parameter: @DisplayLanguage
 - parameter removed
- 1.03 Agent Transferred Interactions:
 - Parameter: @DisplayLanguage
 - parameter removed
- 1.04 Agent Breaks:
 - Parameter: @DisplayLanguage
 - parameter removed
- 1.05 Agent Rejected Interactions:
 - Parameter: @DisplayLanguage
 - parameter removed
- 1.06 Staffing:
 - Parameter: @DisplayLanguage
 - parameter removed
- 2.01 Detailed Agent Interactions:
 - Parameter: @DisplayLanguage
 - parameter removed
- 2.02 Agent Interaction Summary:
 - Parameter: @DisplayLanguage
 - parameter removed
 - Changed fields:
 - Total Ready Time -> Total Ready Idle Time
 - Total Handling Outgoing Interactions -> Total Handling Outgoing Interactions Idle
 - Total Backoffice Time -> Total Backoffice Idle Time
 - CSV Export, changed fields:
 - TotalReadyTime -> TotalReadyIdleTime
 - TotalHandlingOutgoingInteractions -> TotalHandlingOutgoingInteractionsIdle
 - TotalBackOfficeTime -> TotalBackOfficeTimeIdle
- 2.03 Agent Scoring:
 - Parameter: @DisplayLanguage
 - parameter removed
- 2.04 Agent Interaction Summary:
 - Parameter: @DisplayLanguage
 - parameter removed
 - Changed fields:
 - Ready -> Ready Idle

- Handle Outgoing -> Handle Outgoing Idle
 - Backoffice -> Backoffice Idle
- CSV Export, changed fields:
 - Ready -> Ready Idle
 - Handle Outgoing -> Handle Outgoing Idle
 - Backoffice -> Backoffice Idle
- 3.01 Agent Scoring:
 - Parameter: @DisplayLanguage
 - parameter removed
- 3.02 Query Interactions By Origin or Destination:
 - Parameter: @DisplayLanguage
 - parameter removed
- 3.03 Agents Assignments to BP Detailed:
 - Parameter: @DisplayLanguage
 - parameter removed
 - New fields:
 - StartTime
 - EndTime
 - Removed fields:
 - Action
 - Date-Time
 - Dataset Report_AgentsAssignmentsToBPDetailed, new parameters:
 - @AddCurrentAssignments
 - @AddLastEndedAssignments
 - Dataset Report_AgentsAssignmentsToBPDetailed, new fields:
 - StartTime
 - EndTime
 - CSV Export, new fields:
 - StartTime
 - EndTime
 - CSV Export, removed fields:
 - Action
 - Date-Time
- 3.04 Interactions Detail Record:
 - Parameter: @DisplayLanguage
 - parameter removed
- 3.05 Conversations:
 - Parameter: @DisplayLanguage
 - parameter removed
- 4.01 Detailed Business Entity:
 - Parameter: @DisplayLanguage
 - parameter removed
- 4.02 Destination Trace:
 - Parameter: @DisplayLanguage
 - parameter removed
- 4.03 Interaction Disposition Codes:
 - Parameter: @DisplayLanguage
 - parameter removed
- 4.04 Voice Billing:
 - Parameter: @DisplayLanguage
 - parameter removed
- 4.05 Delegated Interactions:
 - Parameter: @DisplayLanguage
 - parameter removed
 - Dataset Report_BusinessProcessCallbackPerformance, new parameters (optional):
 - @DWDBName
- 4.06 Incoming Calls By Area Code:
 - Parameter: @DisplayLanguage
 - parameter removed
- 4.07 Voice Billing By Business Process:
 - Parameter: @DisplayLanguage
 - parameter removed
- 4.08 Voice Billing By Business Process - No Abandoned Calls:
 - Parameter: @DisplayLanguage
 - parameter removed
- 4.09 Voice Billing Summary:
 - Parameter: @DisplayLanguage
 - parameter removed
- 4.10 Inbound Interactions Distribution:
 - Parameter: @DisplayLanguage
 - parameter removed
- 4.11 Interactions Analysis:
 - Parameter: @DisplayLanguage
 - parameter removed
- 4.13 Disposition Codes Distribution:
 - Parameter: @DisplayLanguage

- parameter removed
- 4.14 Demand Interval Distribution:
 - Parameter: @DisplayLanguage
 - parameter removed
- 5.01 Business Process Agent Performance:
 - Parameter: @DisplayLanguage
 - parameter removed
- 5.02 Business Process Interaction Performance:
 - Parameter: @DisplayLanguage
 - parameter removed
- 5.03 Business Process Callback Performance:
 - Parameter: @DisplayLanguage
 - parameter removed
 - New fields:
 - Demands - when "Group By" parameter includes checked value "Demands"
 - Dataset Report_BusinessProcessCallbackPerformance, new fields:
 - Demands
 - Dataset Report_BusinessProcessCallbackPerformance, new parameters (optional):
 - @DWDBName
 - @Demands
 - @GroupByDDL
 - @ClientName
 - @InboundCampaignName
 - @ProductName
 - @ProductType
 - Dataset Report_BusinessProcessCallbackPerformance_SubDs, new parameters (optional):
 - @DWDBName
 - @Demands
 - @GroupByDDL
 - @ClientName
 - @InboundCampaignName
 - @ProductName
 - @ProductType
 - CSV Export, new fields:
 - Demands
- 5.04 Business Process Detailed Alerts:
 - Parameter: @DisplayLanguage
 - parameter removed
- 5.05 Queue Interval Summary:
 - Parameter: @DisplayLanguage
 - parameter removed
- 6.01 Completed Campaign Interactions:
 - Parameter: @DisplayLanguage
 - parameter removed
- 6.02 Campaign Time Frame Specific Statistics:
 - Parameter: @DisplayLanguage
 - parameter removed
- 6.03 Campaign Inspection:
 - Parameter: @DisplayLanguage
 - parameter removed
- 6.04 Completed Callback Interactions:
 - Parameter: @DisplayLanguage
 - parameter removed
 - New fields:
 - Added "Interaction GUIDID" in the main and detailed row
 - Added "Business Process" in the main and detailed row
 - Removed fields
 - Campaign Name
 - Dialer Type
 - Dataset Report_CompletedCallbackInteraction, new fields:
 - CallbackInteractionGUIDID
 - InteractionGUIDID
 - Overflowed
 - BusinessProcessNameInitiated
 - Dataset Report_CompletedCallbackInteraction, new parameters (optional):
 - @BusinessProcessNameInitiated
 - CSV Export, new fields:
 - CallbackInteractionGUIDID
 - Overflowed
 - BusinessProcessNameInitiated
 - InteractionGUIDID
 - BusinessProcessNameCompleted
- 7.01 Customer Experience:
 - Parameter: @DisplayLanguage
 - parameter removed
- 7.02 Customer Experience Trends:
 - Parameter: @DisplayLanguage

- parameter removed
- 7.03 Contact Center Performance:
 - Parameters:
 - @DisplayLanguage parameter removed
 - @ExcludeShortAbandons changed to drop-down section:
 - "Abandon Calls and %" default value = Unchecked
 - "Answer Rate %" default value = Checked
 - "Service Level %" default value = Checked
- 7.04 Abandoned Interactions:
 - Parameter: @DisplayLanguage
 - parameter removed
- 7.05 Contact Center Performance By Demands:
 - Parameter: @DisplayLanguage
 - parameter removed
- 8.01 Calls Performance by BP and Destinations:
 - Parameter: @DisplayLanguage
 - parameter removed
- Admin. Platform Usage By Organization:
 - Parameter: @DisplayLanguage
 - parameter removed
- Admin. Validate Interactions Table:
 - Parameter: @DisplayLanguage
 - parameter removed
- SSIS project changes:
 - Upload To DW:
 - added loading data to new tables in DW database(D_DelegatedState, F_InteractionDelegated)
- LOGIC CHANGES:
 - ETL db changes:
 - procedures changed:
 - ETL.dbo.Upload_F_Interactions - loading new fields to F_interactions(CustomerOptIn,PostAgentTimeSec)
 - new procedures:
 - for 4.05 Report - move to use DWH:
 - dbo.Upload_D_DelegatedState
 - dbo.Delete_D_DelegatedState
 - dbo.Upload_F_InteractionDelegated
 - dbo.Delete_F_InteractionDelegated
 - Users db new procedures:
 - dbo.spGet_F_AgentAssignments
 - dbo.spGet_F_SupervisorAssignments
 - 2.02 Agent Interaction Summary:
 - Previous:
 - Busy = sum(Dialing + WaitingFor + Busy + BackOffice + HandlingOutgoingInteraction)
 - TotalHandlingOutgoingTime = sum(HandlingOutgoingInteraction)
 - BackOffice:
 - sum(case
 - when F_AgentLife.starttime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, F_AgentLife.EndTime)
 - when F_AgentLife.StartTime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime > D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, D_TimeSlice.tEndTime)
 - when F_AgentLife.StartTime < D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, D_TimeSlice.tBeginTime, F_AgentLife.EndTime)
 - else datediff(second, D_TimeSlice.tBeginTime, D_TimeSlice.tEndTime)
 - end)
 - F_AgentLife.Status in('Backoffice - Administrative')
 - HandlingOutgoingInteraction:
 - sum(case
 - when F_AgentLife.starttime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, F_AgentLife.EndTime)
 - when F_AgentLife.StartTime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime > D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, D_TimeSlice.tEndTime)
 - when F_AgentLife.StartTime < D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, D_TimeSlice.tBeginTime, F_AgentLife.EndTime)
 - else datediff(second, D_TimeSlice.tBeginTime, D_TimeSlice.tEndTime)
 - end)
 - F_AgentLife.Status in('Handling - Outgoing Interactions')
 - Current:
 - Busy = sum(Dialing + WaitingFor + Busy + BackOfficeBusy + HandlingOutgoingInteractionBusy)
 - TotalHandlingOutgoingTime = sum(HandlingOutgoingInteractionIdle)

- BackOfficeIdle:
 - sum(case
 - when F_AgentLife.starttime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, F_AgentLife.EndTime)
 - when F_AgentLife.StartTime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime > D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, D_TimeSlice.tEndTime)
 - when F_AgentLife.StartTime < D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, D_TimeSlice.tBeginTime, F_AgentLife.EndTime)
 - else datediff(second, D_TimeSlice.tBeginTime, D_TimeSlice.tEndTime)
 - end)
 - F_AgentLife.Status in('Backoffice - Administrative')
 - and F_AgentLife.InteractionId is null
- BackOfficeBusy:
 - sum(case
 - when F_AgentLife.starttime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, F_AgentLife.EndTime)
 - when F_AgentLife.StartTime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime > D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, D_TimeSlice.tEndTime)
 - when F_AgentLife.StartTime < D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, D_TimeSlice.tBeginTime, F_AgentLife.EndTime)
 - else datediff(second, D_TimeSlice.tBeginTime, D_TimeSlice.tEndTime)
 - end)
 - F_AgentLife.Status in('Backoffice - Administrative')
 - and F_AgentLife.InteractionId is not null
- HandlingOutgoingInteractionIdle:
 - sum(case
 - when F_AgentLife.starttime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, F_AgentLife.EndTime)
 - when F_AgentLife.StartTime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime > D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, D_TimeSlice.tEndTime)
 - when F_AgentLife.StartTime < D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, D_TimeSlice.tBeginTime, F_AgentLife.EndTime)
 - else datediff(second, D_TimeSlice.tBeginTime, D_TimeSlice.tEndTime)
 - end)
 - F_AgentLife.Status in('Handling - Outgoing Interactions')
 - and F_AgentLife.InteractionId is null
- HandlingOutgoingInteractionBusy:
 - sum(case
 - when F_AgentLife.starttime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, F_AgentLife.EndTime)
 - when F_AgentLife.StartTime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime > D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, D_TimeSlice.tEndTime)
 - when F_AgentLife.StartTime < D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, D_TimeSlice.tBeginTime, F_AgentLife.EndTime)
 - else datediff(second, D_TimeSlice.tBeginTime, D_TimeSlice.tEndTime)
 - end)
 - F_AgentLife.Status in('Handling - Outgoing Interactions')
 - and F_AgentLife.InteractionId is not null
- procedures changed:
 - DW.spRpt_AgentTimeAllocationPerformance
 - DW.spRpt_AgentTimeAllocationPerformance_SubDs1
 - dbo.spRpt_GetAgentStatusesDataset
- 2.04 Agent Interaction Summary:
 - Previous:
 - Busy = sum(Dialing + WaitingFor + Busy + BackOffice + HandlingOutgoingInteraction)
 - BackOffice = sum(BackOffice)
 - TotalHandlingOutgoingTime = sum(HandlingOutgoingInteraction)
 - BackOffice:

- sum(case
 - when F_AgentLife.starttime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, F_AgentLife.EndTime)
 - when F_AgentLife.StartTime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime > D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, D_TimeSlice.tEndTime)
 - when F_AgentLife.StartTime < D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, D_TimeSlice.tBeginTime, F_AgentLife.EndTime)
 - else datediff(second, D_TimeSlice.tBeginTime, D_TimeSlice.tEndTime)
 - end)
 - F_AgentLife.Status in('Backoffice - Administrative')
 - HandlingOutgoingInteraction:
 - sum(case
 - when F_AgentLife.starttime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, F_AgentLife.EndTime)
 - when F_AgentLife.StartTime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime > D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, D_TimeSlice.tEndTime)
 - when F_AgentLife.StartTime < D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, D_TimeSlice.tBeginTime, F_AgentLife.EndTime)
 - else datediff(second, D_TimeSlice.tBeginTime, D_TimeSlice.tEndTime)
 - end)
 - F_AgentLife.Status in('Handling - Outgoing Interactions')
- Current:
 - Busy = sum(Dialing + WaitingFor + Busy + BackOfficeBusy + HandlingOutgoingInteractionBusy)
 - BackOffice = sum(BackOfficeIdle)
 - HandlingOutgoingInteraction = sum(HandlingOutgoingInteractionIdle)
 - BackOfficeIdle:
 - sum(case
 - when F_AgentLife.starttime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, F_AgentLife.EndTime)
 - when F_AgentLife.StartTime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime > D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, D_TimeSlice.tEndTime)
 - when F_AgentLife.StartTime < D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, D_TimeSlice.tBeginTime, F_AgentLife.EndTime)
 - else datediff(second, D_TimeSlice.tBeginTime, D_TimeSlice.tEndTime)
 - end)
 - F_AgentLife.Status in('Backoffice - Administrative')
 - and F_AgentLife.InteractionId is null
 - BackOfficeBusy:
 - sum(case
 - when F_AgentLife.starttime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, F_AgentLife.EndTime)
 - when F_AgentLife.StartTime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime > D_TimeSlice.tEndTime
 - then datediff(second, F_AgentLife.StartTime, D_TimeSlice.tEndTime)
 - when F_AgentLife.StartTime < D_TimeSlice.tBeginTime and F_AgentLife.EndTime <= D_TimeSlice.tEndTime
 - then datediff(second, D_TimeSlice.tBeginTime, F_AgentLife.EndTime)
 - else datediff(second, D_TimeSlice.tBeginTime, D_TimeSlice.tEndTime)
 - end)
 - F_AgentLife.Status in('Backoffice - Administrative')
 - and F_AgentLife.InteractionId is not null
 - HandlingOutgoingInteractionIdle:

- sum(case
when F_AgentLife.starttime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime <=
D_TimeSlice.tEndTime
then datediff(second, F_AgentLife.StartTime, F_AgentLife.EndTime)
when F_AgentLife.StartTime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime >
D_TimeSlice.tEndTime
then datediff(second, F_AgentLife.StartTime, D_TimeSlice.tEndTime)
when F_AgentLife.StartTime < D_TimeSlice.tBeginTime and F_AgentLife.EndTime <=
D_TimeSlice.tEndTime
then datediff(second, D_TimeSlice.tBeginTime, F_AgentLife.EndTime)
else datediff(second, D_TimeSlice.tBeginTime, D_TimeSlice.tEndTime)
end)
F_AgentLife.Status in('Handling - Outgoing Interactions')
and F_AgentLife.InteractionId is null
 - HandlingOutgoingInteractionBusy:
 - sum(case
when F_AgentLife.starttime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime <=
D_TimeSlice.tEndTime
then datediff(second, F_AgentLife.StartTime, F_AgentLife.EndTime)
when F_AgentLife.StartTime >= D_TimeSlice.tBeginTime and F_AgentLife.EndTime >
D_TimeSlice.tEndTime
then datediff(second, F_AgentLife.StartTime, D_TimeSlice.tEndTime)
when F_AgentLife.StartTime < D_TimeSlice.tBeginTime and F_AgentLife.EndTime <=
D_TimeSlice.tEndTime
then datediff(second, D_TimeSlice.tBeginTime, F_AgentLife.EndTime)
else datediff(second, D_TimeSlice.tBeginTime, D_TimeSlice.tEndTime)
end)
F_AgentLife.Status in('Handling - Outgoing Interactions')
and F_AgentLife.InteractionId is not null
 - procedures changed:
 - DW.spRpt_AgentInteractionSummaryReport
 - dbo.spRpt_GetAgentStatusesDataset
 - 3.03 Agents Assignments to BP Detailed:
 - new source F_AgentAssignments
 - Agent Name - F_AgentLife.AgentName -> F_AgentAssignments.AgentName(Deleted) (Deleted) will added if D_Agent.Disabled = 1
 - BP Name - F_AgentLife.BusinessProcessName -> F_AgentAssignments.BusinessProcessName
 - Start Time(new field) - F_AgentAssignments.StartTime
 - End Time(new field) - F_AgentAssignments.EndTime
 - Changed By - F_AgentLife.SupervisorName -> F_AgentAssignments.ModifiedBy
 - logic added for new parametes AddCurrentAssignments, AddLastEndedAssignments
 - where
(@AddCurrentAssignments = 1 and F_AgentAssignments.EndTime is null)
or
(@AddCurrentAssignments = 0 and F_AgentAssignments.StartTime between @StartTime and @EndTime)
or
(
@AddLastEndedAssignments = 1
and exists (max(F_AgentAssignments.EndTime) where F_AgentAssignments.EndTime < @StartTime group by
F_AgentAssignments.Agent)
)
 - procedures changed:
 - DW.spRpt_AgentAssignmentToBP
 - new procedures:
 - DW.spRpt_Common_Prm_ChangedBy - fill parameter values from F_AgentAssignments.ModifiedBy
 - 4.05 Delegated Interactions:
 - Initiator - PendingTaskManager.dbo.DelegatedInteraction.InitiatorName -> F_InteractionDelegated.
InitiatorName
 - Converter - PendingTaskManager.dbo.DelegatedInteraction.Converser ->
F_InteractionDelegated.Converser
 - Entry Time - DateAdd(mi, PendingTaskManager.dbo.DelegatedInteraction.EntryTimeTicks * 1440, '1900-01-01') ->
F_InteractionDelegated.EntryDateTime
 - Interaction Type - PendingTaskManager.dbo.DelegatedInteraction.InteractionTypeName ->
F_InteractionDelegated.InteractionTypeName
 - Business Process - PendingTaskManager.dbo.DelegatedInteraction.BusinessProcessName ->
F_InteractionDelegated.BusinessProcessName
 - Last State - PendingTaskManager.dbo.DelegatedInteraction.LastState ->
F_InteractionDelegated.DelegatedState
 - Campaign Name - PendingTaskManager.dbo.DelegatedInteraction.InteractionCampaignName -> F_InteractionDelegated.
CampaignName
 - Closing Time - DateAdd(mi, PendingTaskManager.dbo.DelegatedInteraction.UpdateTimeTicks * 1440, '1900-01-01') ->
F_InteractionDelegated.UpdateDateTime
 - Agent Remark - PendingTaskManager.dbo.DelegatedInteraction.DelegateDetails ->
F_InteractionDelegated.Details

- Supervisor Remark - PendingTaskManager.dbo.DelegatedInteraction.SupervisorCloseMessage -> F_InteractionDelegated.SupervisorCloseMessage
 - Customer ID - PendingTaskManager.dbo.DelegatedInteraction.CustomerID -> F_InteractionDelegated.CustomerID
 - procedures changed:
 - DW.spRpt_DelegatedInteractions
- 5.02 Contact Center Performance:
 - Completed Callbacks(logic that has been added to an existing one):
 - if exists
 - count(distinct F_Interactions.InteractionId) > 1 group by InteractionGUIDID and F_Callbacks.LastInteractionStateID in ('Purged', 'Canceled')
 - then count as completed
 - procedures changed:
 - dbo.spRpt_GetCallbacksDataset
 - dbo.spRpt_GetInteractionsDataset
- 5.03 Business Process Callback Performance:
 - Migrated to DWH
 - new column:
 - EndedAsCanceled = if F_Interactions.InteractionResultName(previous from last Interaction or curr Interaction) = 'Canceled' or F_callbacks.LastInteractionState = 'Canceled' then F_callbacks.Completed fc.LastInteractionStateID
 - procedures changed:
 - DW.spRpt_BusinessProcessCallbackPerformance
 - DW.spRpt_BusinessProcessCallbackPerformance_SubDs1
 - dbo.spRpt_GetCallbacksDataset
 - dbo.spRpt_GetInteractionsDataset
- 6.04 Completed Callback Interactions:
 - CallbackInteractionGUIDID:
 - F_Callbacks.InteractionGUIDID
 - InteractionGUIDID:
 - F_Callbacks.InteractionGUIDID
 - Overflowed:
 - F_Callbacks.Overflowed
 - BusinessProcessInitiated:
 - F_Callbacks.RequestedBusinessProcess
 - procedures changed:
 - DW.spRpt_CompletedCallbackInteraction
- 7.01 Contact Center Performance:
 - Completed Callbacks(logic that has been added to an existing one):
 - if exists
 - count(distinct F_Interactions.InteractionId) > 1 group by InteractionGUIDID and F_Callbacks.LastInteractionStateID in ('Purged', 'Canceled')
 - then count as completed
 - procedures changed:
 - dbo.spRpt_GetCallbacksDataset
 - dbo.spRpt_GetInteractionsDataset
- 7.03 Contact Center Performance:
 - Calls Abandoned - F_Interactions.SLExcluded excluded from the formula when Exclude Short Abandons parameter = (Abandon Calls and %) unchecked
 - Abandonment Rate (%) - F_Interactions.SLExcluded excluded from the formula when Exclude Short Abandons parameter = (Abandon Calls and %) unchecked
 - AvgAbandonTime - F_Interactions.SLExcluded excluded from the formula when Exclude Short Abandons parameter = (Abandon Calls and %) unchecked
 - Answer Rate - F_Interactions.SLExcluded excluded from the formula when Exclude Short Abandons parameter = (Answer Rate %) unchecked
 - Service Level - F_Interactions.SLExcluded excluded from the formula when Exclude Short Abandons parameter = (Service Level %) unchecked
 - Completed Callbacks(logic that has been added to an existing one):
 - if exists
 - count(distinct F_Interactions.InteractionId) > 1 group by InteractionGUIDID and F_Callbacks.LastInteractionStateID in ('Purged', 'Canceled')
 - then count as completed
 - procedures changed:
 - dbo.spRpt_GetCallbacksDataset
 - dbo.spRpt_GetInteractionsDataset
 - dbo.spRpt_ContactCenterPerformance
 - dbo.spRpt_ContactCenterPerformance_Total
- 7.05 Contact Center Performance:
 - Completed Callbacks(logic that has been added to an existing one):
 - if exists
 - count(distinct F_Interactions.InteractionId) > 1 group by InteractionGUIDID

and F_Callbacks.LastInteractionStateID in ('Purged', 'Canceled')
then count as completed

- procedures changed:
 - dbo.spRpt_GetCallbacksDataset
 - dbo.spRpt_GetInteractionsDataset