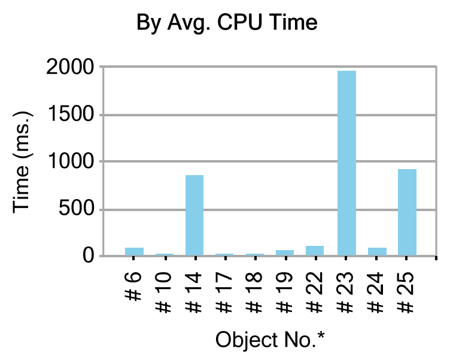
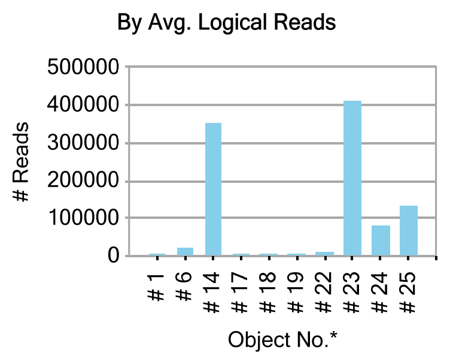
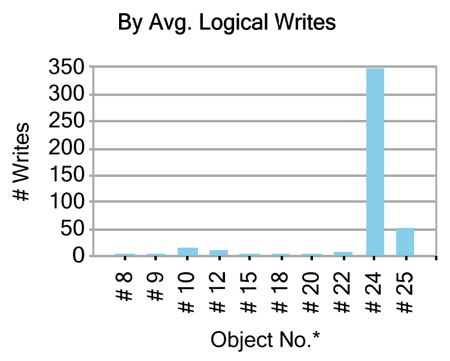
Object Execution Statistics
[BeaconProdV1]
on S45-40-143-129 at 7/29/2016 12:19:15 AM

This report provides detailed historical execution data for all currently cached plans for objects within the Database. This execution data is aggregated over the time during which the plan has been in the cache.

Top Executable Objects


By Avg. CPU Time


By Avg. Logical Reads


By Avg. Logical Writes

* See column 'Object No.' from table 'All Objects' for value of 'Object No.' It is the unique number assigned to each Object.

All Executable Objects

Shows statement wise execution statistics for all the executable objects.

Object No.*	Object Name	Object Type		Avg. CPU Time (ms.)	Total CPU Time (ms.)	Total CPU Time (%)				# Avg. Logical Reads	# Total Logical Reads				# Avg. Logical Writes	# Total Logical Writes				# Avg. Logical IO	# Total Logical IO	Total Logical IO (%)			
1	[dbo]..uspAllAccountForScheduledPeriod	SQL Stored-Procedure		5.69	153.74	1.88				2,268.15	61,240				0.00	0				2,268.15	61,240	2.27			
	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	SELECT DISTINCT Account.Id,Account.AccountNumber AS AccountNumber FROM AssignedDC DC INNER JOIN Account AS House ON DC.HouseID = House.Id INNER JOIN Account ON House.ParentId = Account.Id WHERE (DC.SchedulePeriod = @SchedulePeriod) And (DC.isClosed=1) ORDER BY AccountNumber	27	1	5.69	153.74		7.09	5.10	7.68	2,268.15	61,240	5,772	1,776	6,132	0.00	0	0	0	0	2,268.15	61,240		5,772	1,776	6,132
2	[dbo]..uspAllAccountHomeForSchedulePeriod	SQL Stored-Procedure		4.43	70.83	0.87				1,799.06	28,785				0.00	0				1,799.06	28,785	1.07			
	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	SELECT DISTINCT House.Id,House.AccountNumber AS AccountNumber FROM AssignedDC DC INNER JOIN Account AS House ON DC.HouseID = House.Id INNER JOIN Account ON House.ParentId = Account.Id WHERE (DC.SchedulePeriod = @SchedulePeriod) And DC.AccountID = @AccountID And (DC.isClosed=1) ORDER BY AccountNumber	16	1	4.43	70.83		4.31	4.20	4.87	1,799.06	28,785	1,784	1,775	2,000	0.00	0	0	0	0	1,799.06	28,785		1,784	1,775	2,000

Object No.*	Object Name	Object Type		Avg. CPU Time (ms.)	Total CPU Time (ms.)	Total CPU Time (%)				# Avg. Logical Reads	# Total Logical Reads				# Avg. Logical Writes	# Total Logical Writes				# Avg. Logical IO	# Total Logical IO	Total Logical IO (%)			
3	[dbo]..uspDCMessageReport	SQL Stored-Procedure		9.83	49.14	0.60				860.00	4,300				0.00	0				860.00	4,300	0.16			
	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	Select * from (select Case When M.CreatedBy ='Employee' then E.EMP_FIRST_NAME + ' ' + E.EMP_LAST_NAME else Case when E2.Emp_ID is not null then E2.EMP_FIRST_NAME + ' ' + E2.EMP_LAST_NAME else M.CreatedBy end end [SendBy], M.[Message], M.CreatedDate, E.EMP_FIRST_NAME + ' ' + E.EMP_LAST_NAME [SendTo], Case When M.CreatedBy ='Employee' then 'Employee' else 'Beacon' end [SendByType], M.DirectCareID [DCID] From [Messages] M Inner join Employee E on E.Emp_ID = M.DirectCareID Left join Users U on U.UserName = M.CreatedBy and (U.UserTypeID=5 or U.UserID=2) Left join Employee E2 on E2.Emp_ID = U.EmpId Where M.Type='Msg' and (M.CreatedDate >= @StartDate or @StartDate ='') and (M.CreatedDate <= @EndDate or @EndDate ='') and (/*(M.CreatedBy ='Employee' and M.DirectCareID = @DCID) Or (M.CreatedBy !='Employee' and U.EmpId= @DCID) or*/ M.DirectCareID = @DCID or U.EmpId= @DCID or (isnull (@DCID,0) = 0)) and(isnull(@SendToDC,0)=0 or M.DirectCareID = @SendToDC) and ((@MessageFor ='') or (@MessageFor = 'Mass Text' and M.MessageFor like 'Mass Text%') or (@MessageFor = 'Without Mass Text' and M.MessageFor not like 'Mass Text%'))) as tmp Order by CASE WHEN @OrderByColumn + ' ' + @OrderbyDir = 'SendTo ASC' THEN [SendTo] END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderbyDir = 'SendTo DESC' THEN [SendTo] END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderbyDir = 'SendBy ASC' THEN [SendBy] END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderbyDir = 'SendBy DESC' THEN [SendBy] END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderbyDir = 'CreatedDate ASC' THEN CreatedDate END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderbyDir = 'CreatedDate DESC' THEN CreatedDate END DESC, CreatedDate	5	1	9.83	49.14		9.31	9.31	10.34	860.00	4,300	860	860	860	0.00	0	0	0	0	860.00	4,300		860	860	860
4	[dbo]..uspAssignDCHistoryInfo	SQL Stored-Procedure		0.61	1.84	0.02				92.00	276				0.00	0				92.00	276	0.01			
	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	select AC.AccountNumber as AccountName ,HN.AccountNumber as HouseName ,ISNULL (Emp.EMP_FIRST_NAME, '') + ' ' + ISNULL (Emp.EMP_LAST_NAME, '') as DCName from AssignedDC DC Left Join Employee Emp on Emp.EMP_ID = DC.DCID Left Join Account AC on AC.ID = DC.AccountID Left Join Account HN on HN.ID = DC.HouseID where DC.ID = @ID	3	1	0.61	1.84		0.62	0.59	0.62	92.00	276	92	92	92	0.00	0	0	0	0	92.00	276		92	92	92
5	[dbo]..utrAssignedDCModified	SQL Trigger		0.22	0.22	0.00				36.00	36				0.00	0				36.00	36	0.00			
	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	Select @Old_DCID=DcId,@Old_PunchCardClosed =PunchCardClosed From deleted	1	1	0.02	0.02		0.02	0.02	0.02	6.00	6	6	6	6	0.00	0	0	0	0	6.00	6		6	6	6
	Select @Cur_DCID=DcId,@Cur_PunchCardClosed =PunchCardClosed,@Cur_SchedulePeriod=SchedulePeriod From Inserted -- Update StartDateTime, EndDateTime, ActualStartDateTime, ActualEndDateTime	1	1	0.01	0.01		0.01	0.01	0.01	6.00	6	6	6	6	0.00	0	0	0	0	6.00	6		6	6	6

Object No.*	Object Name	Object Type		Avg. CPU Time (ms.)	Total CPU Time (ms.)	Total CPU Time (%)				# Avg. Logical Reads	# Total Logical Reads				# Avg. Logical Writes	# Total Logical Writes				# Avg. Logical IO	# Total Logical IO	Total Logical IO (%)			
	update AssignedDC set StartDateTime = Convert (datetime, ADC2.ScheduleDate + ' ' + ADC2.StartTime), EndDateTime = Case when ADC2.EndTime ='12:00 AM' or datediff(minute,cast (ADC2.scheduledate +' '+ ADC2.starttime as datetime),cast (ADC2.scheduledate +' '+ ADC2.EndTime as datetime)) < 0 then Convert(datetime, DateAdd(d, 1, ADC2.ScheduleDate) + ' ' + ADC2.EndTime) else Convert (datetime, ADC2.ScheduleDate + ' ' + ADC2.EndTime) end, ActualStartDateTime = Convert(datetime, ADC2.ScheduleDate + ' ' + ADC2.ActualStartTime), ActualEndDateTime = Case when ADC2.EndTime ='12:00 AM' or datediff(minute,cast (ADC2.scheduledate +' '+ ADC2.starttime as datetime),cast (ADC2.scheduledate +' '+ ADC2.EndTime as datetime)) < 0 then Convert(datetime, DateAdd(d, 1, ADC2.ScheduleDate) + ' ' + ADC2.ActualEndTime) else Convert(datetime, ADC2.ScheduleDate + ' ' + ADC2.ActualEndTime) end from Inserted ADC2 where AssignedDC.ID = ADC2.ID	1	1	0.08	0.08		0.08	0.08	0.08	15.00	15	15	15	15	0.00	0	0	0	0	15.00	15		15	15	15
	Select @Old_DCID=DcId,@Old_PunchCardClosed =PunchCardClosed From deleted	1	1	0.03	0.03		0.03	0.03	0.03	2.00	2	2	2	2	0.00	0	0	0	0	2.00	2		2	2	2
	Select @Cur_DCID=DcId,@Cur_PunchCardClosed =PunchCardClosed,@Cur_SchedulePeriod=SchedulePeriod From Inserted -- Update StartDateTime, EndDateTime, ActualStartDateTime, ActualEndDateTime	1	1	0.01	0.01		0.01	0.01	0.01	2.00	2	2	2	2	0.00	0	0	0	0	2.00	2		2	2	2
	update AssignedDC set StartDateTime = Convert (datetime, ADC2.ScheduleDate + ' ' + ADC2.StartTime), EndDateTime = Case when ADC2.EndTime ='12:00 AM' or datediff(minute,cast (ADC2.scheduledate +' '+ ADC2.starttime as datetime),cast (ADC2.scheduledate +' '+ ADC2.EndTime as datetime)) < 0 then Convert(datetime, DateAdd(d, 1, ADC2.ScheduleDate) + ' ' + ADC2.EndTime) else Convert (datetime, ADC2.ScheduleDate + ' ' + ADC2.EndTime) end, ActualStartDateTime = Convert(datetime, ADC2.ScheduleDate + ' ' + ADC2.ActualStartTime), ActualEndDateTime = Case when ADC2.EndTime ='12:00 AM' or datediff(minute,cast (ADC2.scheduledate +' '+ ADC2.starttime as datetime),cast (ADC2.scheduledate +' '+ ADC2.EndTime as datetime)) < 0 then Convert(datetime, DateAdd(d, 1, ADC2.ScheduleDate) + ' ' + ADC2.ActualEndTime) else Convert(datetime, ADC2.ScheduleDate + ' ' + ADC2.ActualEndTime) end from Inserted ADC2 where AssignedDC.ID = ADC2.ID	1	1	0.07	0.07		0.07	0.07	0.07	5.00	5	5	5	5	0.00	0	0	0	0	5.00	5		5	5	5
6	[dbo]..uspMessageSummaryReport	SQL Stored-Procedure		95.04	190.09	2.32				19,698.00	39,396				0.00	0				19,698.00	39,396	1.46			
	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	Select @OEmployees = @OEmployees +',[' + Case when E2.Emp_ID is not null then E2.EMP_FIRST_NAME + ' ' + E2.EMP_LAST_NAME else M.CreatedBy end + ']', @OEmployeesCol = @OEmployeesCol +',isnull([' + Case when E2.Emp_ID is not null then E2.EMP_FIRST_NAME + ' ' + E2.EMP_LAST_NAME else M.CreatedBy end + '],0)'+ '[' + Case when E2.Emp_ID is not null then E2.EMP_FIRST_NAME + ' ' + E2.EMP_LAST_NAME else M.CreatedBy end + ']' from [Messages] M Inner join Employee E on E.Emp_ID = M.DirectCareID Left join Users U on U.Email = M.CreatedBy and (U.UserTypeID=5 or U.UserID=2) Left join Employee E2 on E2.Emp_ID = U.EmpId Where M.[Type]='Msg' and M.CreatedBy !='Employee' and (M.CreatedDate >= @StartDate and M.CreatedDate <= @EndDate) and ((@MessageFor ='') or (@MessageFor = 'Mass Text' and M.MessageFor like 'Mass Text%') or (@MessageFor = 'Without Mass Text' and M.MessageFor not like 'Mass Text%')) Group by Case when E2.Emp_ID is not null then E2.EMP_FIRST_NAME + ' ' + E2.EMP_LAST_NAME else M.CreatedBy end	2	1	47.32	94.63		47.90	46.73	47.90	9,849.00	19,698	9,849	9,849	9,849	0.00	0	0	0	0	9,849.00	19,698		9,849	9,849	9,849

Object No.*	Object Name	Object Type		Avg. CPU Time (ms.)	Total CPU Time (ms.)	Total CPU Time (%)				# Avg. Logical Reads	# Total Logical Reads				# Avg. Logical Writes	# Total Logical Writes				# Avg. Logical IO	# Total Logical IO	Total Logical IO (%)			
	Select @OEmployees = @OEmployees +',[' + Case when E2.Emp_ID is not null then E2.EMP_FIRST_NAME + ' ' + E2.EMP_LAST_NAME else M.CreatedBy end + ']', @OEmployeesCol = @OEmployeesCol +',isnull([' + Case when E2.Emp_ID is not null then E2.EMP_FIRST_NAME + ' ' + E2.EMP_LAST_NAME else M.CreatedBy end + '],0)'+ '[' + Case when E2.Emp_ID is not null then E2.EMP_FIRST_NAME + ' ' + E2.EMP_LAST_NAME else M.CreatedBy end + ']' from [Messages] M Inner join Employee E on E.Emp_ID = M.DirectCareID Left join Users U on U.Email = M.CreatedBy and (U.UserTypeID=5 or U.UserID=2) Left join Employee E2 on E2.Emp_ID = U.EmpId Where M.[Type]='Msg' and M.CreatedBy !='Employee' and (M.CreatedDate >= @StartDate and M.CreatedDate <= @EndDate) and ((@MessageFor ='') or (@MessageFor = 'Mass Text' and M.MessageFor like 'Mass Text%') or (@MessageFor = 'Without Mass Text' and M.MessageFor not like 'Mass Text%')) Group by Case when E2.Emp_ID is not null then E2.EMP_FIRST_NAME + ' ' + E2.EMP_LAST_NAME else M.CreatedBy end	2	1	47.73	95.46		47.91	47.55	47.91	9,849.00	19,698	9,849	9,849	9,849	0.00	0	0	0	0	9,849.00	19,698		9,849	9,849	9,849
7	[dbo]..uspWorksAtListSelect	SQL Stored-Procedure		5.88	123.50	1.51				1,776.90	37,315				0.00	0				1,776.90	37,315	1.38			
	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	SELECT WAT.* ,A1.Id as AccountId ,A1.AccountNumber ,A2.AccountNumber as AccountName ,A1.HS_ADDRESS ,A1.City ,A1.State ,A1.ZIP ,A1.Email ,A1.HS_PHONE ,1 AS 'TotalRows', (Select Max(Convert (datetime, ADC.ScheduleDate + ' ' + ADC.ActualStartTime)) from AssignedDC ADC where ADC.houseID=WAT.HouseId and ADC.DCID=@EmpId and isnull(ADC.ActualStartTime,'')! ='') [LastWorkTime] FROM [WorksAtList] WAT INNER JOIN Account A1 ON WAT.HouseId = A1.Id INNER JOIN Account AS A2 ON A1.ParentId = A2.Id WHERE [EmpId]= COALESCE (@EmpId,[EmpId])	21	1	5.88	123.50		5.37	5.07	14.76	1,776.90	37,315	1,777	1,776	1,777	0.00	0	0	0	0	1,776.90	37,315		1,777	1,776	1,777
8	[dbo]..uspUsersSelect	SQL Stored-Procedure		0.52	13.00	0.16				30.20	755				0.04	1				30.24	756	0.03			
	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	SELECT TblRange.* From (SELECT U.* ,A.Type as AccountType ,UT.Type as UserType ,UT.HomePage ,ROW_NUMBER() OVER (ORDER BY UserId ASC) AS RowNum ,TotalRows=Count(UserId) OVER() FROM [Users] U Left Join Account A on U.AccountId = A.Id Left Join UserTypes UT on UT.Id = U.UserTypeId --where condition where (@UserName is null or U.UserName = @UserName) AND (@Password is null or U. [Password] = @Password) AND (@Email is null or U.Email = @Email) AND (@ResetToken is null or U.ResetToken = @ResetToken) AND (@UserTypeID is null or U.UserTypeID = @UserTypeID))TblRange WHERE RowNum BETWEEN ISNULL(@StartRange,1) AND ISNULL (@EndRange,TblRange. [TotalRows]) ORDER BY TblRange.UserId ASC	25	1	0.52	13.00		0.53	0.37	0.86	30.20	755	32	21	32	0.04	1	0	0	1	30.24	756		32	21	33
9	[dbo]..uspAccountSelect	SQL Stored-Procedure		1.17	3.50	0.04				134.00	402				0.67	2				134.67	404	0.01			
	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	SELECT TblRange.* From (SELECT * ,ROW_NUMBER() OVER (ORDER BY Id ASC) AS RowNum ,TotalRows=Count(Id) OVER () FROM [Account] where (@AccountNumber is null or AccountNumber = @AccountNumber) And (ISNULL(@ParentId, 0) = 0 or ParentId = @ParentId) And (@Type is null or Type = @Type) AND (@HS_PHONE is null or HS_PHONE = @HS_PHONE)																								

	)TblRange WHERE RowNum BETWEEN ISNULL(@StartRange,1) AND ISNULL(@EndRange,TblRange.[TotalRows]) ORDER BY CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Id ASC' THEN TblRange.Id END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Id DESC' THEN TblRange.Id END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'ParentId ASC' THEN TblRange.ParentId END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'ParentId DESC' THEN TblRange.ParentId END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Type ASC' THEN TblRange.Type END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Type DESC' THEN TblRange.Type END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'AccountNumber ASC' THEN TblRange.AccountNumber END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'AccountNumber DESC' THEN TblRange.AccountNumber END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CreatedDate ASC' THEN TblRange.CreatedDate END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CreatedDate DESC' THEN TblRange.CreatedDate END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CreatedBy ASC' THEN TblRange.CreatedBy END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CreatedBy DESC' THEN TblRange.CreatedBy END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'UpdatedDate ASC' THEN TblRange.UpdatedDate END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'UpdatedDate DESC' THEN TblRange.UpdatedDate END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'UpdatedBy ASC' THEN TblRange.UpdatedBy END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'UpdatedBy DESC' THEN TblRange.UpdatedBy END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'HS_NAME ASC' THEN TblRange.HS_NAME END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'HS_NAME DESC' THEN TblRange.HS_NAME END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'HS_PHONE ASC' THEN TblRange.HS_PHONE END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'HS_PHONE DESC' THEN TblRange.HS_PHONE END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'INVOICEEMAIL ASC' THEN TblRange.INVOICEEMAIL END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'INVOICEEMAIL DESC' THEN TblRange.INVOICEEMAIL END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'DeptCode ASC' THEN TblRange.DeptCode END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'DeptCode DESC' THEN TblRange.DeptCode END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Code ASC' THEN TblRange.Code END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Code DESC' THEN TblRange.Code END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'HS_ADDRESS ASC' THEN TblRange.HS_ADDRESS END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'HS_ADDRESS DESC' THEN TblRange.HS_ADDRESS END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'City	3	1	1.17	3.50		0.83	0.83	1.60	134.00	402	101	100	201	0.67	2	0	0	2	134.67	404		101	100	203

	ASC' THEN TblRange.City END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'City DESC' THEN TblRange.City END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'State ASC' THEN TblRange.State END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'State DESC' THEN TblRange.State END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'ZIP ASC' THEN TblRange.ZIP END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'ZIP DESC' THEN TblRange.ZIP END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Email ASC' THEN TblRange.Email END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Email DESC' THEN TblRange.Email END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'DefaultRate ASC' THEN TblRange.DefaultRate END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'DefaultRate DESC' THEN TblRange.DefaultRate END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'AMAPRate ASC' THEN TblRange.AMAPRate END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'AMAPRate DESC' THEN TblRange.AMAPRate END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Password ASC' THEN TblRange.Password END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Password DESC' THEN TblRange.Password END DESC																								
10	[dbo]..uspEmployeeSelect	SQL Stored-Procedure		26.80	777.20	9.50				1,637.79	47,496				15.69	455				1,653.48	47,951	1.78			
	SQL Statement	**# Executions (With Last Plan)**	**# Plans Generated**	**Avg. CPU Time (ms.)**	**Total CPU Time (ms.)**		**Last CPU Time (ms.)**	**Min. CPU Time (ms.)**	**Max. CPU Time (ms.)**	**# Avg. Logical Reads**	**# Total Physical Reads**	**# Last Logical Reads**	**# Min. Logical Reads**	**# Max. Logical Reads**	**# Avg. Logical Writes**	**# Total Logical Writes**	**# Last Logical Writes**	**# Min. Logical Writes**	**# Max. Logical Writes**	**# Avg. Logical IO**	**# Total Logical IO**		**# Last Logical IO**	**# Min. Logical IO**	**# Max. Logical IO**
	SELECT TblRange.* From (SELECT E.*,isnull (U.UserId,0) [LoginUserID] ,ROW_NUMBER() OVER (ORDER BY E.EMP_ID ASC) AS RowNum ,TotalRows=Count (E.EMP_ID) OVER() FROM [Employee] E Left join [Users] U on U.EmpId = E.EMP_ID where (@EMP_ID is null or E.EMP_ID = @EMP_ID) AND (@EMP_TYPE IS NULL OR E.EMP_TYPE = @EMP_TYPE) AND (@AccountId IS NULL OR E.AccountId = @AccountId) AND (@EMP_FIRST_NAME IS NULL OR E.EMP_FIRST_NAME like '%' + @EMP_FIRST_NAME + '%') AND (@EMP_LAST_NAME IS NULL OR E.EMP_LAST_NAME like '%' + @EMP_LAST_NAME + '%') AND (@EMP_EMAIL IS NULL OR E.EMP_EMAIL= @EMP_EMAIL) AND (@EMP_CELL IS NULL OR E.EMP_CELL= @EMP_CELL) AND (@EMP_ADDRESS IS NULL OR E.EMP_ADDRESS= @EMP_ADDRESS) AND (@Status IS NULL OR E.Status= @Status) AND (@HouseID is null OR (isnull (U.UserId,0) >0 and U.HouseID =@HouseID)))TblRange WHERE RowNum BETWEEN ISNULL(@StartRange,1) AND ISNULL (@EndRange,TblRange. [TotalRows]) ORDER BY CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EMP_ID ASC' THEN TblRange.EMP_ID END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EMP_ID DESC' THEN TblRange.EMP_ID END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EMP_FIRST_NAME ASC' THEN TblRange.EMP_FIRST_NAME END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EMP_FIRST_NAME DESC' THEN TblRange.EMP_FIRST_NAME END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EMP_LAST_NAME ASC' THEN TblRange.EMP_LAST_NAME END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EMP_LAST_NAME DESC' THEN TblRange.EMP_LAST_NAME END DESC,																								

```
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'BEACON_EMP_CODE ASC'
THEN
TblRange.BEACON_EMP_CO
DE END ASC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'BEACON_EMP_CODE DESC'
THEN
TblRange.BEACON_EMP_CO
DE END DESC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'EMP_EMAIL ASC' THEN
TblRange.EMP_EMAIL END
ASC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'EMP_EMAIL DESC' THEN
TblRange.EMP_EMAIL END
DESC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'EMP_CELL ASC' THEN
TblRange.EMP_CELL END
ASC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'EMP_CELL DESC' THEN
TblRange.EMP_CELL END
DESC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'HOME_PHONE ASC' THEN
TblRange.HOME_PHONE
END ASC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'HOME_PHONE DESC' THEN
TblRange.HOME_PHONE
END DESC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'EMP_CELL2 ASC' THEN
TblRange.EMP_CELL2 END
ASC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'EMP_CELL2 DESC' THEN
TblRange.EMP_CELL2 END
DESC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'HOME_PHONE2 ASC' THEN
TblRange.HOME_PHONE2
END ASC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'HOME_PHONE2 DESC'
THEN
TblRange.HOME_PHONE2
END DESC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'EMP_ADDRESS ASC' THEN
TblRange.EMP_ADDRESS
END ASC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'EMP_ADDRESS DESC' THEN
TblRange.EMP_ADDRESS
END DESC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection = 'CITY
ASC' THEN TblRange.CITY
END ASC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection = 'CITY
DESC' THEN TblRange.CITY
END DESC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection = 'STATE
ASC' THEN TblRange.STATE
END ASC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection = 'STATE
DESC' THEN TblRange.STATE
END DESC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection = 'ZIP ASC'
THEN TblRange.ZIP END
ASC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection = 'ZIP
DESC' THEN TblRange.ZIP
END DESC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection = 'APT_NO
ASC' THEN
TblRange.APT_NO END ASC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection = 'APT_NO
DESC' THEN
TblRange.APT_NO END
DESC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'EMP_CERTIFICATE ASC'
THEN
TblRange.EMP_CERTIFICATE
END ASC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'EMP_CERTIFICATE DESC'
THEN
TblRange.EMP_CERTIFICATE
END DESC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'EMP_SKILLS ASC' THEN
TblRange.EMP_SKILLS END
ASC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'EMP_SKILLS DESC' THEN
TblRange.EMP_SKILLS END
DESC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'EMP_GENDER ASC' THEN
TblRange.EMP_GENDER END
ASC,
 CASE WHEN
@OrderByColumn + ' ' +
@OrderByDirection =
'EMP_GENDER DESC' THEN
TblRange.EMP_GENDER END
DESC,
 CASE WHEN
@OrderByColumn + ' ' +
```

	@OrderByDirection = 'EMP_TYPE ASC' THEN TblRange.EMP_TYPE END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EMP_TYPE DESC' THEN TblRange.EMP_TYPE END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EMP_APMP ASC' THEN TblRange.EMP_APMP END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EMP_APMP DESC' THEN TblRange.EMP_APMP END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CREATEDBY ASC' THEN TblRange.CREATEDBY END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CREATEDBY DESC' THEN TblRange.CREATEDBY END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CREATEDDATE ASC' THEN TblRange.CREATEDDATE END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CREATEDDATE DESC' THEN TblRange.CREATEDDATE END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'UPDATEDBY ASC' THEN TblRange.UPDATEDBY END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'UPDATEDBY DESC' THEN TblRange.UPDATEDBY END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'UPDATEDDATE ASC' THEN TblRange.UPDATEDDATE END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'UPDATEDDATE DESC' THEN TblRange.UPDATEDDATE END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'ApprovalDate ASC' THEN TblRange.ApprovalDate END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'ApprovalDate DESC' THEN TblRange.ApprovalDate END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'LastDayWorked ASC' THEN TblRange.LastDayWorked END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'LastDayWorked DESC' THEN TblRange.LastDayWorked END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'AMAP ASC' THEN TblRange.AMAP END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'AMAP DESC' THEN TblRange.AMAP END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'HasId ASC' THEN TblRange.HasId END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'HasId DESC' THEN TblRange.HasId END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Drives ASC' THEN TblRange.Drives END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Drives DESC' THEN TblRange.Drives END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'MaleFemale ASC' THEN TblRange.MaleFemale END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'MaleFemale DESC' THEN TblRange.MaleFemale END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EmployeeId ASC' THEN TblRange.EmployeeId END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EmployeeId DESC' THEN TblRange.EmployeeId END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Location ASC' THEN TblRange.Location END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Location DESC' THEN TblRange.Location END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Title ASC' THEN TblRange.Title END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Title DESC' THEN TblRange.Title END DESC,	29	2	26.80	777.20		31.18	2.75	32.39	1,637.79	47,496	1,879	101	1,881	15.69	455	16	0	26	1,653.48	47,951		1,895	101	1,907

	CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'PayCode ASC' THEN TblRange.PayCode END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'PayCode DESC' THEN TblRange.PayCode END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Status ASC' THEN TblRange.Status END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Status DESC' THEN TblRange.Status END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EMP_PASSWORD ASC' THEN TblRange.EMP_PASSWORD END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EMP_PASSWORD DESC' THEN TblRange.EMP_PASSWORD END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EMP_USERNAME ASC' THEN TblRange.EMP_USERNAME END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EMP_USERNAME DESC' THEN TblRange.EMP_USERNAME END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'SSN ASC' THEN TblRange.SSN END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'SSN DESC' THEN TblRange.SSN END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'MSO_EFFECTIVE_DATE ASC' THEN TblRange.MSO_EFFECTIVE_DATE END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'MSO_EFFECTIVE_DATE DESC' THEN TblRange.MSO_EFFECTIVE_DATE END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'MSO_EXPIRY_DATE ASC' THEN TblRange.MSO_EXPIRY_DATE END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'MSO_EXPIRY_DATE DESC' THEN TblRange.MSO_EXPIRY_DATE END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'ALT_PHONE ASC' THEN TblRange.ALT_PHONE END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'ALT_PHONE DESC' THEN TblRange.ALT_PHONE END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'OTHER_CONTACT ASC' THEN TblRange.OTHER_CONTACT END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'OTHER_CONTACT DESC' THEN TblRange.OTHER_CONTACT END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CONTRACTOR ASC' THEN TblRange.CONTRACTOR END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CONTRACTOR DESC' THEN TblRange.CONTRACTOR END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'SERVICE ASC' THEN TblRange.SERVICE END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'SERVICE DESC' THEN TblRange.SERVICE END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'TYPE ASC' THEN TblRange.TYPE END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'TYPE DESC' THEN TblRange.TYPE END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'DOB ASC' THEN TblRange.DOB END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'DOB DESC' THEN TblRange.DOB END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'ACC_NUM ASC' THEN TblRange.ACC_NUM END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'ACC_NUM DESC' THEN TblRange.ACC_NUM END DESC,																								

	CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'LIC_NUM ASC' THEN TblRange.LIC_NUM END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'LIC_NUM DESC' THEN TblRange.LIC_NUM END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CHEST_XRAY ASC' THEN TblRange.CHEST_XRAY END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CHEST_XRAY DESC' THEN TblRange.CHEST_XRAY END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CONTRACTOR_TYPE ASC' THEN TblRange.CONTRACTOR_TYPE END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CONTRACTOR_TYPE DESC' THEN TblRange.CONTRACTOR_TYPE END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EIN ASC' THEN TblRange.EIN END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EIN DESC' THEN TblRange.EIN END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'NYS_DRV_LIC ASC' THEN TblRange.NYS_DRV_LIC END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'NYS_DRV_LIC DESC' THEN TblRange.NYS_DRV_LIC END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'DISCIPLINE ASC' THEN TblRange.DISCIPLINE END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'DISCIPLINE DESC' THEN TblRange.DISCIPLINE END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'DATE_HIRED ASC' THEN TblRange.DATE_HIRED END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'DATE_HIRED DESC' THEN TblRange.DATE_HIRED END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'BOX_NUMBER ASC' THEN TblRange.BOX_NUMBER END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'BOX_NUMBER DESC' THEN TblRange.BOX_NUMBER END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Department ASC' THEN TblRange.Department END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Department DESC' THEN TblRange.Department END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'AccountId ASC' THEN TblRange.AccountId END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'AccountId DESC' THEN TblRange.AccountId END DESC																								
11	[dbo]..uspEmployeeFormTemplatesSelect	SQL Stored-Procedure		0.18	0.18	0.00				12.00	12				0.00	0				12.00	12	0.00			
	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	SELECT *,1 AS 'TotalRows' FROM [EmployeeFormTemplates] WHERE [ID]= COALESCE(@ID,[ID])	1	1	0.18	0.18		0.18	0.18	0.18	12.00	12	12	12	12	0.00	0	0	0	0	12.00	12		12	12	12
12	[dbo]..uspAssignedDCInviteSave	SQL Stored-Procedure		4.94	4.94	0.06				114.00	114				11.00	11				125.00	125	0.00			
	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	Insert into AssignedDCInvite (SchedulePeriod,AssignDCIDs,DirectCareIDs,CreatedBy,CreatedOn) Values (@SchedulePeriod,@AssignDCIDs,@DirectCareIDs,@CreatedBy,getdate());	1	1	0.14	0.14		0.14	0.14	0.14	9.00	9	9	9	9	2.00	2	2	2	2	11.00	11		11	11	11
	Insert into AssignedDCInviteDetail (AssignedDCInviteID,AssignedDCID,Emp_ID) Select @Id,t.value('id[1]','int'),t.value('empid[1]','int') from @XML.nodes('/xml/row') x(t)	1	1	0.50	0.50		0.50	0.50	0.50	27.00	27	27	27	27	1.00	1	1	1	1	28.00	28		28	28	28
	Update AssignedDC Set IsConfirm='S' Where ID in (select value from dbo.fnsplit(@AssignDCIDs,',')) and isnull(PunchCardClosed,0)=0;	1	1	4.30	4.30		4.30	4.30	4.30	78.00	78	78	78	78	8.00	8	8	8	8	86.00	86		86	86	86

Object No.*	Object Name	Object Type		Avg. CPU Time (ms.)	Total CPU Time (ms.)	Total CPU Time (%)				# Avg. Logical Reads	# Total Logical Reads				# Avg. Logical Writes	# Total Logical Writes				# Avg. Logical IO	# Total Logical IO	Total Logical IO (%)			
13	[dbo]..uspTextMessageSettingSelect	SQL Stored-Procedure		0.12	4.83	0.06				13.98	559				0.00	0				13.98	559	0.02			
	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	SELECT TblRange.* From (SELECT ES.* ,ROW_NUMBER() OVER (ORDER BY ES.ID ASC) AS RowNum ,TotalRows=Count(ES.ID) OVER() FROM TextMessageSetting ES where (@Type is null or Type = @Type))TblRange WHERE RowNum BETWEEN ISNULL(@StartRange,1) AND ISNULL (@EndRange,TblRange.[TotalRows]) ORDER BY CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'ID ASC' THEN TblRange.ID END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'ID DESC' THEN TblRange.ID END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Type ASC' THEN TblRange.Type END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Type DESC' THEN TblRange.Type END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CreateDate ASC' THEN TblRange.CreateDate END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CreateDate DESC' THEN TblRange.CreateDate END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CreatedBy ASC' THEN TblRange.CreatedBy END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CreatedBy DESC' THEN TblRange.CreatedBy END DESC	40	1	0.12	4.83		0.09	0.08	0.31	13.98	559	14	13	14	0.00	0	0	0	0	13.98	559		14	13	14
14	[dbo]..uspAssignDCDateWiseDClist	SQL Stored-Procedure		854.53	1,709.06	20.89				351,551.00	703,102				0.00	0				351,551.00	703,102	26.04			
	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	Select Top 20 E.Emp_ID,E.EMP_FIRST_NAME + ' '+ E.EMP_LAST_NAME + ' (' + Cast(Count(ADC2.DCID) as varchar(10)) +')' [EmpName],Count (ADC2.DCID) [Total], 1[DCSR] From assignedDC ADC2 Inner join Employee E on E.Emp_ID = ADC2.DCID Where ADC2 .Scheduleperiod=@Scheduleperiod and ADC2.PunchCardClosed=0 and ADC2.IsClosed=1 and ADC2.DCID in (Select ADC.DCID From assignedDC ADC Where ADC.Scheduleperiod=@Scheduleperiod and ADC.ScheduleDate= @ScheduleDate and ADC.DCID >0 and ADC.PunchCardClosed=0 and ADC.IsClosed=1) Group by Emp_ID,E.EMP_FIRST_NAME + ' '+ E.EMP_LAST_NAME Union All Select -1 [Emp_ID],'NONAME (' + Cast(Count(ADC3.ID) as varchar(10)) +')' [EmpName],Count(ADC3.ID) [Total], 2 [DCSR] From assignedDC ADC3 Where ADC3 .Scheduleperiod=@Scheduleperiod and ADC3.PunchCardClosed=0 and ADC3.IsClosed=1 and isnull (ADC3.DCID,0)=0 and (Select Count(1) from assignedDC DC4 Where DC4 .Scheduleperiod=@Scheduleperiod and DC4.PunchCardClosed=0 and DC4.IsClosed=1 and isnull(DC4.DCID,0)=0 and DC4.ScheduleDate= @ScheduleDate) > 0 Having Count(ADC3.ID) >0 Order by DCSR,3 desc;	2	1	854.53	1,709.06		1,694.55	14.51	1,694.55	351,551.00	703,102	697,814	5,288	697,814	0.00	0	0	0	0	351,551.00	703,102		697,814	5,288	697,814
15	[dbo]..uspAuditLogSave	SQL Stored-Procedure		0.70	2.09	0.03				24.00	72				2.67	8				26.67	80	0.00			
	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	INSERT INTO [AuditLog] ([EntityName], [ObjectType], [PrimaryId], [ActionBy], [Action], [Value], [Date], PrimaryKeys) VALUES (@EntityName, @ObjectType, @PrimaryId, @ActionBy, @Action, @Value, isnull(@Date,getdate()), @PrimaryKeys)	3	1	0.70	2.09		0.50	0.50	1.05	24.00	72	28	20	28	2.67	8	3	1	4	26.67	80		31	21	32

Object No.*	Object Name	Object Type		Avg. CPU Time (ms.)	Total CPU Time (ms.)	Total CPU Time (%)				# Avg. Logical Reads	# Total Logical Reads				# Avg. Logical Writes	# Total Logical Writes				# Avg. Logical IO	# Total Logical IO	Total Logical IO (%)			
16	[dbo]..uspEmailSettingsSelect	SQL Stored-Procedure		1.89	3.79	0.05				449.00	898				0.00	0				449.00	898	0.03			
	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	SELECT TblRange.* From (SELECT ES.* ,(SELECT STUFF((SELECT ',' + CAST (ESE.EMP_ID AS VARCHAR (10)) [text()] FROM EmailSettingEmployees ESE where ESE.EmailSettingID = ES.ID FOR XML PATH(''), TYPE) .value('.','NVARCHAR (MAX)'),1,1,'')) as EmployeeIDs ,(SELECT STUFF((SELECT ',' + CAST (E.EMP_Email AS VARCHAR (100)) [text()] FROM EmailSettingEmployees ESE Left Join Employee E on E.EMP_ID = ESE.EMP_ID where ESE.EmailSettingID = ES.ID FOR XML PATH(''), TYPE) .value('.','NVARCHAR (MAX)'),1,1,'')) as EmployeeEmails ,ROW_NUMBER() OVER (ORDER BY ES.ID ASC) AS RowNum ,TotalRows=Count(ES.ID) OVER() FROM [EmailSettings] ES where (@Type is null or Type = @Type))TblRange WHERE RowNum BETWEEN ISNULL(@StartRange,1) AND ISNULL (@EndRange,TblRange. [TotalRows]) ORDER BY CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'ID ASC' THEN TblRange.ID END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'ID DESC' THEN TblRange.ID END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Type ASC' THEN TblRange.Type END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Type DESC' THEN TblRange.Type END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EmailSubject ASC' THEN TblRange.EmailSubject END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EmailSubject DESC' THEN TblRange.EmailSubject END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EmailBody ASC' THEN TblRange.EmailBody END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EmailBody DESC' THEN TblRange.EmailBody END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CreateDate ASC' THEN TblRange.CreateDate END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CreateDate DESC' THEN TblRange.CreateDate END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CreatedBy ASC' THEN TblRange.CreatedBy END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'CreatedBy DESC' THEN TblRange.CreatedBy END DESC	2	1	1.89	3.79		3.22	0.57	3.22	449.00	898	832	66	832	0.00	0	0	0	0	449.00	898		832	66	832
17	[dbo]..uspEmployeesByHouseForHomeSchedular	SQL Stored-Procedure		10.94	21.88	0.27				3,623.00	7,246				0.00	0				3,623.00	7,246	0.27			
	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	SELECT DISTINCT * FROM AssignedDC WHERE SchedulePeriod=@SchedulePeriod and HouseId=@houseID;	2	1	4.88	9.75		4.92	4.83	4.92	1,764.00	3,528	1,764	1,764	1,764	0.00	0	0	0	0	1,764.00	3,528		1,764	1,764	1,764
	select Concat (EMP_FIRST_NAME,' ',EMP_LAST_NAME) as EmpName, E.EMP_ID FROM EMPLOYEE E INNER JOIN WorksAtList W ON W.EmpId = E.EMP_ID WHERE E.[Status]='Active' AND emp_type='2' AND W.HouseId = @houseID AND EMP_ID NOT IN(Select DCID from [BlackListEmployee] WHERE IsBlackList=1) ORDER BY EmpName	2	1	1.13	2.25		1.14	1.11	1.14	95.00	190	95	95	95	0.00	0	0	0	0	95.00	190		95	95	95
	Select DCID,ScheduleDate,CAST (ScheduleDate + ' ' + StartTime as dateTIME) [StartDt],CAST (ScheduleDate + ' ' + EndTime as dateTIME) [EndDt] From AssignedDC Where cast(ScheduleDate as date) >= @StartDate and cast (ScheduleDate as date) <= @EndDate And HouseID= @houseID Order by ScheduleDate,3,4	2	1	4.94	9.88		4.43	4.43	5.45	1,764.00	3,528	1,764	1,764	1,764	0.00	0	0	0	0	1,764.00	3,528		1,764	1,764	1,764
18	[dbo]..uspClosedAssignDC	SQL Stored-Procedure		31.31	31.31	0.38				3,655.00	3,655				2.00	2				3,657.00	3,657	0.14			

	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	Select DC.*, E.EMP_FIRST_NAME + ' ' + E.EMP_LAST_NAME as EmployeeName, E.EMP_CELL AS EmpCell, E.HOME_PHONE AS EmpPhone, DC.MsgSent, DC.PunchCardClosed ,(SELECT STUFF ((SELECT '<br />' + Emp.EMP_FIRST_NAME + ' ' + Emp.EMP_LAST_NAME [text()] FROM AssignedDCCancelHistory ADCH left join Employee Emp on Emp.EMP_ID = ADCH.EMP_ID WHERE AssignedDCID = DC.ID FOR XML PATH("), TYPE) .value('.','NVARCHAR (MAX)'),1,6,") List_Output) as CancelledEmployee ,(SELECT STUFF ((SELECT '<br />' + ADCH.CancelReason [text()] FROM AssignedDCCancelHistory ADCH WHERE AssignedDCID = DC.ID FOR XML PATH("), TYPE) .value('.','NVARCHAR (MAX)'),1,6,")List_Output) as CancelledReason, isnull(DC.BillClosed,0) [BillClosed], isnull(AA.AccountNumber,") [AccountName], isnull(AH.AccountNumber,") [HouseName], isnull(DC.isNotToFill,0) [IsNotToFill2], Case When datediff (mi,DateAdd(hour,3,Cast (DC.ScheduleDate + ' '+ StartTime as datetime)),getdate ()) >0 then 1 else 0 end [PastTime], E.EMP_FIRST_NAME + ' ' + E.EMP_LAST_NAME + ' - ' + Case When isnull (WL.id,0)>0 then 'HY' else 'HN' end + ' - M' + cast(cast(isnull (EHD.Distance,0) as int) as varchar(10)) + ' - H' + Format(isnull (ESM.TotalMinutes,0) /60.0,'0') [EmpDetail] From [AssignedDC] DC LEFT JOIN Employee E ON (Case when isnull(DC.DcId,0) =0 then @SuggestedDCID else DC.DCID end) = E.Emp_Id Left join EmployeeSchedulePeriodMinute ESM on ESM.EmployeeID = (Case when isnull(DC.DcId,0) =0 then @SuggestedDCID else DC.DCID end) and ESM.SchedulePeriod =DC.SchedulePeriod Left join EmployeeHouseDistances EHD on EHD.EmployeeID = (Case when isnull(DC.DcId,0) =0 then @SuggestedDCID else DC.DCID end) and EHD.HouseID = DC.HouseID Left join WorksAtList WL on WL.EmpId = DC.DcId and WL.HouseId = DC.HouseID Left Join Account AA on AA.Id = DC.AccountID Left Join Account AH on AH.Id = DC.HouseID WHERE DC.IsClosed=1 AND DC.SchedulePeriod=@SchedulePeriod and (DC.AccountID=@AccountID or @AccountID=0) and (DC.HouseID=@HouseID or @HouseID=0) and (DC.DCID=@DCID or @DCID=0 or (@DCID= -1 and isnull(DC.DCID,0)=0)) and (DC.ScheduleDate=@ScheduleDate or isnull (@ScheduleDate,")=") and ((@AssignedFilter=" and (@isConfirmed =" or (@isConfirmed ='Y' and DC.IsConfirm like '%Y') or (@isConfirmed ='N' and DC.IsConfirm like '%N'))) Or (@AssignedFilter='Y' and (@isConfirmed =" or (@isConfirmed ='Y' and DC.PunchCardClosed =1) or (@isConfirmed ='N' and DC.PunchCardClosed =0)))) and ((@AssignedFilter='Y' and isnull(DC.isNotToFill,0) = @NotToFill) or @AssignedFilter=") and (@AssignedDC=" or DC.ID in (select value from dbo.fnsplit(@AssignedDC,','))) Order by CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'ScheduleDate ASC' THEN cast ((DC.ScheduleDate + ' ' + DC.StartTime) as datetime) END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'ScheduleDate DESC' THEN cast((DC.ScheduleDate + ' ' + DC.StartTime) as datetime) END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EmployeeName ASC' THEN (E.EMP_FIRST_NAME + ' ' + E.EMP_LAST_NAME) END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'EmployeeName DESC' THEN (E.EMP_FIRST_NAME + ' ' + E.EMP_LAST_NAME) END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Closed ASC' THEN DC.PunchCardClosed END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderByDirection = 'Closed	1	25	30.69	30.69		30.69	30.69	30.69	3,485.00	3,485	3,485	3,485	3,485	1.00	1	1	1	1	3,486.00	3,486		3,486	3,486	3,486

	DESC' THEN DC.PunchCardClosed END DESC OFFSET @OffsetRows ROWS FETCH NEXT @TotalRecords ROWS ONLY OPTION (RECOMPILE) -- Insert calling log																								
	Select top 1 ADC.*,Case when E.Emp_ID is null then U.UserName else E.EMP_FIRST_NAME + ' ' + E.EMP_LAST_NAME end as EmployeeName from AssignDCCallLog ADC LEFT JOIN Users U ON ADC.CreatedBy = U.UserId Left join Employee E ON U.EmpId = E.Emp_Id Where SearchedDCID=@DCID and SchedulePeriod=@SchedulePe riod Order by ADC.ID Desc	1	1	0.37	0.37		0.37	0.37	0.37	80.00	80	80	80	80	0.00	0	0	0	0	80.00	80		80	80	80
	select * from Employee where emp_ID=2	1	1	0.17	0.17		0.17	0.17	0.17	85.00	85	85	85	85	0.00	0	0	0	0	85.00	85		85	85	85
	Insert into AssignDCCallLog (SchedulePeriod,SearchedDCI D,CreatedBy,CreatedOn) values (@SchedulePeriod,@DCID,@ CreatedBy, Dateadd(hour,3, getdate()))	1	1	0.09	0.09		0.09	0.09	0.09	5.00	5	5	5	5	1.00	1	1	1	1	6.00	6		6	6	6
19	[dbo]..uspAssignedDCSelect	SQL Stored-Procedure		58.76	617.71	7.55				5,291.83	28,222				0.00	0				5,291.83	28,222	1.05			
	SQL Statement	**# Executions (With Last Plan)**	**# Plans Generated**	**Avg. CPU Time (ms.)**	**Total CPU Time (ms.)**		**Last CPU Time (ms.)**	**Min. CPU Time (ms.)**	**Max. CPU Time (ms.)**	**# Avg. Logical Reads**	**# Total Physical Reads**	**# Last Logical Reads**	**# Min. Logical Reads**	**# Max. Logical Reads**	**# Avg. Logical Writes**	**# Total Logical Writes**	**# Last Logical Writes**	**# Min. Logical Writes**	**# Max. Logical Writes**	**# Avg. Logical IO**	**# Total Logical IO**		**# Last Logical IO**	**# Min. Logical IO**	**# Max. Logical IO**
	SELECT *,1 AS 'TotalRows' FROM [AssignedDC] WHERE [ID]= COALESCE(@ID,[ID])	2	1	4.22	8.43		4.20	4.20	4.23	1,764.00	3,528	1,764	1,764	1,764	0.00	0	0	0	0	1,764.00	3,528		1,764	1,764	1,764
	Select Distinct cast(0 as bigint) [ID],SchedulePeriod,cast (substring (SchedulePeriod,0,charindex ('-',SchedulePeriod,1)-1) as date) [ScheduleDate1], '' [StartTime], '' [EndTime],cast(0 as bit) [IsClosed], cast(0 as bit) [IsCustom], substring (SchedulePeriod,0,charindex ('-',SchedulePeriod,1)) [ScheduleDate] From AssignedDC Where (HouseId=@HouseID or @HouseID is null) and (@IsClosed is null or IsClosed=@IsClosed) Order by 3	12	1	50.02	600.24		55.69	5.60	55.69	1,763.83	21,166	1,764	1,763	1,764	0.00	0	0	0	0	1,763.83	21,166		1,764	1,763	1,764
	SELECT distinct cast(0 as bigint) [ID],SchedulePeriod,ScheduleD ate, StartTime,EndTime, IsClosed,IsCustom From AssignedDC Where SchedulePeriod=@SchedulePe riod and HouseID=@HouseID;	2	1	4.52	9.04		4.45	4.45	4.58	1,764.00	3,528	1,764	1,764	1,764	0.00	0	0	0	0	1,764.00	3,528		1,764	1,764	1,764
20	[dbo]..fnSplit	SQL Table-valued-Function		0.18	7.22	0.09				4.26	286				0.04	2				4.31	288	0.01			
	SQL Statement	**# Executions (With Last Plan)**	**# Plans Generated**	**Avg. CPU Time (ms.)**	**Total CPU Time (ms.)**		**Last CPU Time (ms.)**	**Min. CPU Time (ms.)**	**Max. CPU Time (ms.)**	**# Avg. Logical Reads**	**# Total Physical Reads**	**# Last Logical Reads**	**# Min. Logical Reads**	**# Max. Logical Reads**	**# Avg. Logical Writes**	**# Total Logical Writes**	**# Last Logical Writes**	**# Min. Logical Writes**	**# Max. Logical Writes**	**# Avg. Logical IO**	**# Total Logical IO**		**# Last Logical IO**	**# Min. Logical IO**	**# Max. Logical IO**
	INSERT INTO @tableList (value) VALUES (@value)	29	1	0.16	4.64		0.01	0.01	3.90	2.66	77	1	1	15	0.03	1	0	0	1	2.69	78		1	1	16
	INSERT INTO @tableList (value) VALUES (@value)	130	1	0.02	2.58		0.01	0.01	0.12	1.61	209	1	1	12	0.01	1	0	0	1	1.62	210		1	1	13
21	[dbo]..uspEmployeeFormTem plateRequestSelect	SQL Stored-Procedure		5.72	5.72	0.07				328.00	328				0.00	0				328.00	328	0.01			
	SQL Statement	**# Executions (With Last Plan)**	**# Plans Generated**	**Avg. CPU Time (ms.)**	**Total CPU Time (ms.)**		**Last CPU Time (ms.)**	**Min. CPU Time (ms.)**	**Max. CPU Time (ms.)**	**# Avg. Logical Reads**	**# Total Physical Reads**	**# Last Logical Reads**	**# Min. Logical Reads**	**# Max. Logical Reads**	**# Avg. Logical Writes**	**# Total Logical Writes**	**# Last Logical Writes**	**# Min. Logical Writes**	**# Max. Logical Writes**	**# Avg. Logical IO**	**# Total Logical IO**		**# Last Logical IO**	**# Min. Logical IO**	**# Max. Logical IO**
	SELECT ISNULL (Emp.EMP_FIRST_NAME, '') + ' ' + ISNULL (Emp.EMP_LAST_NAME, '') as EmployeeName ,EFT.Title as TemplateName ,EFTR.* ,1 AS 'TotalRows' FROM [EmployeeFormTemplateReque st] EFTR Inner Join Employee Emp on Emp.EMP_ID = EFTR.EMP_ID Inner Join EmployeeFormTemplates EFT on EFT.ID = EFTR.EmployeeFormTemplateI D WHERE EFTR.[ID]= COALESCE (@ID,EFTR.[ID])	1	1	5.72	5.72		5.72	5.72	5.72	328.00	328	328	328	328	0.00	0	0	0	0	328.00	328		328	328	328
22	[dbo]..uspHouseImportSched uleProcess	SQL Stored-Procedure		99.06	99.06	1.21				10,241.00	10,241				6.00	6				10,247.00	10,247	0.38			
	SQL Statement	**# Executions (With Last Plan)**	**# Plans Generated**	**Avg. CPU Time (ms.)**	**Total CPU Time (ms.)**		**Last CPU Time (ms.)**	**Min. CPU Time (ms.)**	**Max. CPU Time (ms.)**	**# Avg. Logical Reads**	**# Total Physical Reads**	**# Last Logical Reads**	**# Min. Logical Reads**	**# Max. Logical Reads**	**# Avg. Logical Writes**	**# Total Logical Writes**	**# Last Logical Writes**	**# Min. Logical Writes**	**# Max. Logical Writes**	**# Avg. Logical IO**	**# Total Logical IO**		**# Last Logical IO**	**# Min. Logical IO**	**# Max. Logical IO**
	Select @DCConfirmCount=Count(1) FROM AssignedDC adc INNER join HouseScheduleImport hsi on hsi.SchedulePeriod=ADC.Sche dulePeriod and hsi.AccountID = adc.AccountID and hsi.HouseID =adc.HouseID Where (hsi.HouseID = @HouseID or isnull (@HouseID,0)=0) and isnull (adc.PunchCardClosed,0) =1 and hsi.SessionID=@SessionID;	1	1	18.47	18.47		18.47	18.47	18.47	3,937.00	3,937	3,937	3,937	3,937	0.00	0	0	0	0	3,937.00	3,937		3,937	3,937	3,937

Object No.*	Object Name	Object Type		Avg. CPU Time (ms.)	Total CPU Time (ms.)	Total CPU Time (%)				# Avg. Logical Reads	# Total Logical Reads				# Avg. Logical Writes	# Total Logical Writes				# Avg. Logical IO	# Total Logical IO	Total Logical IO (%)			
	Delete AssignedDC FROM AssignedDC adc INNER join HouseScheduleImport hsi on hsi.SchedulePeriod=ADC.SchedulePeriod and hsi.AccountID = adc.AccountID and hsi.HouseID =adc.HouseID Where (hsi.HouseID = @HouseID or isnull (@HouseID,0)=0) and hsi.sessionID =@SessionID;	1	1	19.60	19.60		19.60	19.60	19.60	3,937.00	3,937	3,937	3,937	3,937	0.00	0	0	0	0	3,937.00	3,937		3,937	3,937	3,937
	Insert into AssignedDC (SchedulePeriod,ScheduleDate ,AccountID,HouseID,StartTime, EndTime, [Type],DcId,IsClosed,PunchCardClosed,StartDateTime,EndDateTime) OUTPUT inserted.ID INTO @output Select hsi.SchedulePeriod,hsi.SDATE, hsi.AccountID,hsi.HouseID,hsi. StartTime,hsi.EndTime, Case When isnull(hsi.DCID,0) =0 then 'Any' else E.EMP_FIRST_NAME + ' '+ E.EMP_LAST_NAME end [Type], Case When isnull(hsi.DCID,0) =0 then null else hsi.DCID end [DCID], 1 [IsClosed], Case When DCStatus='Yes' and isnull (hsi.DCID,0) > 0 then 1 else 0 end [PunchCardClosed], Convert(datetime, hsi.SDATE + ' ' + hsi.StartTime), Case when hsi.EndTime ='12:00 AM' or datediff (minute,cast(hsi.SDATE +' '+ hsi.StartTime as datetime),cast (hsi.SDATE +' '+ hsi.EndTime as datetime)) < 0 then Convert(datetime, DateAdd(d, 1, hsi.SDATE) + ' ' + hsi.EndTime) else Convert (datetime, hsi.SDATE + ' ' + hsi.EndTime) end From HouseScheduleImport hsi Left join Employee E on E.EMP_ID = HSI.DCID Where hsi.sessionID =@SessionID and isnull (hsi.AccountID ,0) >0 and isnull (hsi.HouseID,0) > 0 And (hsi.HouseID = @HouseID or isnull (@HouseID,0)=0) Order by SchedulePeriod,hsi.AccountID, hsi.HouseID,convert (date,hsi.SDATE);	1	1	60.91	60.91		60.91	60.91	60.91	2,364.00	2,364	2,364	2,364	2,364	6.00	6	6	6	6	2,370.00	2,370		2,370	2,370	2,370
	Select @IDs= @IDs + ',' + cast (id as varchar(10)) from @output;	1	1	0.09	0.09		0.09	0.09	0.09	3.00	3	3	3	3	0.00	0	0	0	0	3.00	3		3	3	3
23	[dbo]..uspHouseImportScheduleSummary	SQL Stored-Procedure		1,967.78	1,967.78	24.06				412,509.00	412,509				0.00	0				412,509.00	412,509	15.28			
	SQL Statement	**# Executions (With Last Plan)**	**# Plans Generated**	**Avg. CPU Time (ms.)**	**Total CPU Time (ms.)**		**Last CPU Time (ms.)**	**Min. CPU Time (ms.)**	**Max. CPU Time (ms.)**	**# Avg. Logical Reads**	**# Total Physical Reads**	**# Last Logical Reads**	**# Min. Logical Reads**	**# Max. Logical Reads**	**# Avg. Logical Writes**	**# Total Logical Writes**	**# Last Logical Writes**	**# Min. Logical Writes**	**# Max. Logical Writes**	**# Avg. Logical IO**	**# Total Logical IO**		**# Last Logical IO**	**# Min. Logical IO**	**# Max. Logical IO**

	Select isnull (tmp.SchedulePeriod,ADC.SchedulePeriod) [SchedulePeriod], isnull(tmp.AccountID,isnull (ADC.AccountID,0)) [AccountID],isnull (tmp.HouseID,isnull (ADC.HouseID,0)) [HouseID], isnull(tmp.ImportedShift,0) [ImportedShift], isnull(ADC.CurrentShift,0) [CurrentShift],isnull (ADC.ClosedInDB,0) [ClosedInDB], Case when isnull (ADC.AccountID,0) > 0 then A1.AccountNumber else A.AccountNumber end [AccountName], Case when isnull (ADC.HouseID,0) > 0 then H1.AccountNumber else H.AccountNumber end [HouseName], isnull(tmp.AgencyName,'') [AgencyName],isnull (tmp.DCNotMatched,0) [DCNotMatched], isnull(tmp.ClosedInExcel,0) [ClosedInExcel], --1. imported records != existing no of records 2. No of matched records != imported records Case When isnull (ADC.CurrentShift,0) != isnull ([ImportedShift],0) or isnull (match.[MatchRecords],0) != [ImportedShift] then 'No' else 'Yes' end [DtNotMatched], Case When isnull (ADC.CurrentShift,0) != isnull ([ImportedShift],0) or isnull (match.[MatchRecords],0) != [ImportedShift] then 'red' else '' end [NotMatchedColor], Case when isnull (ADC.AccountID,0) >0 then A1.AccountNumber + ' - '+ H1.AccountNumber else A.AccountNumber + ' - '+ H.AccountNumber end [AgencyName1] from (Select hsi.SchedulePeriod,isnull (hsi.AccountID,0) [AccountID],isnull (hsi.HouseID,0) [HouseID], Count(hsi.ID) [ImportedShift],hsi.AgencyName, Sum(Case when isnull (hsi.DCStatus,'') = 'YES' then 1 else 0 end) [ClosedInExcel], Sum(Case when isnull (hsi.DCID,0) = 0 and hsi.DCName !='OPEN/TBA' then 1 else 0 end) [DCNotMatched] From HouseScheduleImport hsi Where hsi.sessionID =@SessionID Group by hsi.SchedulePeriod,hsi.AgencyName,hsi.AccountID,hsi.HouseID)as tmp Left join Account A on A.ID = tmp.AccountID Left join Account H on H.ID = tmp.HouseID FULL OUTER JOIN (Select Count(ADC2.ID) [CurrentShift], Sum(Case when isnull (ADC2.PunchCardClosed,0) = 0 then 0 else 1 end) [ClosedInDB],ADC2 .AccountID,ADC2 .HouseID,ADC2 .SchedulePeriod From Assigneddc ADC2 Where ADC2.SchedulePeriod in (Select SchedulePeriod from HouseScheduleImport where sessionID =@SessionID) Group by ADC2.SchedulePeriod,ADC2 .AccountID,ADC2.HouseID) As ADC on ADC.SchedulePeriod = tmp.SchedulePeriod and ADC.AccountID = tmp.AccountID and ADC.HouseID = tmp.HouseID Left join Account A1 on A1.ID = ADC.AccountID Left join Account H1 on H1.ID = ADC.HouseID Left JOIN (Select hsi.SchedulePeriod,isnull (hsi.HouseID,0) [HouseID],Count(1) [MatchRecords] From HouseScheduleImport hsi Inner join AssignedDC ADC on ADC.SchedulePeriod = hsi.SchedulePeriod and ADC.HouseID = hsi.HouseID and ADC.ScheduleDate = hsi.SDATE and ADC.StartTime = hsi.starttime and ADC.EndTime = hsi.EndTime and (isnull (hsi.DCStatus,'No') = Case when isnull (ADC.PunchCardClosed,0) = 1 then 'Yes' else 'No' end) Left join Employee E on E.Emp_ID = ADC.DCID Where hsi.sessionID =@SessionID and ADC.scheduleperiod in (Select SchedulePeriod from HouseScheduleImport where sessionID =@SessionID) and ADC.houseid in (Select HouseID from HouseScheduleImport where sessionID =@SessionID) and ((isnull(ADC.DCID,0) = isnull(hsi.DCID,0) and isnull (hsi.DCID,0) >0) or (isnull(hsi.DCID,0) =0 and hsi.DCName ='OPEN/TBA' and isnull(ADC.DCID,0)=0 and ADC.[Type]='Any')) Group by hsi.SchedulePeriod,hsi.HouseID) as match on match.SchedulePeriod = tmp.SchedulePeriod and match.[HouseID] = tmp.HouseID Order by isnull (tmp.SchedulePeriod,ADC.SchedulePeriod),isnull (tmp.AgencyName,'ZZ')	1	18	1,952.22	1,952.22		1,952.22	1,952.22	1,952.22	407,006.00	407,006	407,006	407,006	407,006	0.00	0	0	0	0	407,006.00	407,006		407,006	407,006	407,006

Object No.*	Object Name	Object Type		Avg. CPU Time (ms.)	Total CPU Time (ms.)	Total CPU Time (%)				# Avg. Logical Reads	# Total Logical Reads				# Avg. Logical Writes	# Total Logical Writes				# Avg. Logical IO	# Total Logical IO	Total Logical IO (%)			
	Insert into #TempImpDetail (SchedulePeriod,AccountID,HouseID,AgencyName,DCName, Confirmed, SDate,StartTime,EndTime,DCNotMapped,DetailNotMatched,CurrentShift,ScheduleDCName,ScheduleConfirm,InsertType) Select Case when hsi.SchedulePeriod is not null then hsi.SchedulePeriod else ADC.SchedulePeriod end [SchedulePeriod], isnull(hsi.AccountID,0) [AccountID],isnull(hsi.HouseID,0) [HouseID], hsi.AgencyName,hsi.DCName, hsi.[Confirmed], Case when hsi.SDATE is not null then hsi.SDATE else ADC.scheduledate end [SDATE], Case when hsi.StartTime is not null then hsi.StartTime else ADC.StartTime end [StartTime], Case when hsi.EndTime is not null then hsi.EndTime else ADC.EndTime end [EndTime], isnull(hsi.[DCNotMatched],0) [DCNotMapped], Case When isnull(ADC.ID,0) = 0 or (ADC.DCID != hsi.DCID) or (isnull(hsi.[Confirmed],'') != isnull(ADC.[ScheduleConfirm],'')) then 1 else 0 end [DetailNotMatched], isnull(ADC.ID,0) [CurrentShift], isnull(ADC.DCName,'') [ScheduleDCName],isnull(ADC.ScheduleConfirm,'') [ScheduleConfirm],1 From (Select hsi1.SchedulePeriod,isnull(hsi1.AccountID,0) [AccountID],isnull(hsi1.HouseID,0) [HouseID], hsi1.AgencyName,hsi1.DCName,hsi1.DCStatus [Confirmed], hsi1.SDATE,hsi1.StartTime,hsi1.EndTime,isnull(hsi1.DCID,0) [DCID], Case when isnull(hsi1.DCID,0) = 0 and hsi1.DCName !='OPEN/TBA' then 1 else 0 end [DCNotMatched], row_number() over (partition by hsi1.SchedulePeriod,hsi1.SDATE,hsi1.StartTime,hsi1.EndTime, Case when hsi1.DCName ='OPEN/TBA' then 'Any' else hsi1.DCName end order by hsi1.SDATE,hsi1.StartTime,hsi1.EndTime,hsi1.DCName) [SRNo] From HouseScheduleImport hsi1 with (NOLOCK) Where hsi1.sessionID =@SessionID and hsi1.HouseID = @HouseID) as hsi Left JOIN (Select DC.ID,scheduleperiod,scheduledate,starttime,endtime, Case when isnull(E.Emp_ID,0) >0 then EMP_LAST_NAME+ ', '+ EMP_FIRST_NAME else DC.[Type] end [DCName], Case when isnull(DC.PunchCardClosed,0) = 1 then 'Yes' else 'No' end [ScheduleConfirm] ,DC.IsClosed,isnull(DC.DCID,0) [DCID], row_number() over (partition by SchedulePeriod,scheduledate, StartTime,EndTime, isnull(DCID,0) order by scheduledate,StartTime,EndTime,DCID) [SRNo] from AssignedDC DC with (NOLOCK) Left join Employee E on E.Emp_ID = DC.DCID where scheduleperiod in (Select SchedulePeriod from HouseScheduleImport where sessionID =@SessionID) and DC.houseid=@HouseID) As ADC on ADC.SchedulePeriod = hsi.SchedulePeriod and ADC.ScheduleDate = hsi.SDATE and ADC.StartTime = hsi.starttime and ADC.EndTime = hsi.EndTime and ((isnull(ADC.DCID,0) = isnull(hsi.DCID,0) and isnull(hsi.DCID,0) >0) or (isnull(hsi.DCID,0) =0 and hsi.DCName ='OPEN/TBA' and isnull(ADC.DCID,0)=0 and ADC.[DCName]='Any')) and hsi.SRNo = ADC.SRNo	1	13	5.40	5.40		5.40	5.40	5.40	1,861.00	1,861	1,861	1,861	1,861	0.00	0	0	0	0	1,861.00	1,861		1,861	1,861	1,861

Object No.*	Object Name	Object Type		Avg. CPU Time (ms.)	Total CPU Time (ms.)	Total CPU Time (%)				# Avg. Logical Reads	# Total Logical Reads				# Avg. Logical Writes	# Total Logical Writes				# Avg. Logical IO	# Total Logical IO	Total Logical IO (%)			
	Insert into #TempImpDetail (SchedulePeriod,AccountID,HouseID,AgencyName,DCName, Confirmed, SDate,StartTime,EndTime,DCNotMapped,DetailNotMatched,CurrentShift,ScheduleDCName,ScheduleConfirm,InsertType) Select old.SchedulePeriod, old.AccountID,old.HouseID,old.AgencyName,old.DCName, old.Confirmed, old.SDATE, old.StartTime,old.EndTime,old.DCNotMapped, Case When ADC.ID is not null then 1 else old.DetailNotMatched end [DetailNotMatched], Case When ADC.ID is not null then ADC.ID else old.CurrentShift end [CurrentShift], Case When ADC.ID is not null then isnull(ADC.DCName,'') else old.ScheduleDCName end [ScheduleDCName], Case When ADC.ID is not null then isnull (ADC.ScheduleConfirm,'') else old.ScheduleConfirm end [ScheduleConfirm],2 --,isnull (ADC.ID,0) [NewID] From (Select Q.SchedulePeriod, Q.AccountID,Q.HouseID,Q.AgencyName,Q.DCName, Q.Confirmed, Q.SDATE, Q.StartTime,Q.EndTime,Q.DCNotMapped, Q.DetailNotMatched, Q.CurrentShift,Q.ScheduleDCName,Q.ScheduleConfirm, row_number() over (partition by SchedulePeriod,SDATE,StartTime,EndTime order by SDATE,StartTime,EndTime) [SRNo] From #TempImpDetail as Q) as old Left JOIN (Select DC.ID,scheduleperiod,scheduledate,starttime,endtime, Case when isnull (E.Emp_ID,0) >0 then EMP_LAST_NAME+ ', '+ EMP_FIRST_NAME else DC.[Type] end [DCName], Case when isnull (DC.PunchCardClosed,0) = 1 then 'Yes' else 'No' end [ScheduleConfirm] ,DC.IsClosed,isnull(DC.DCID,0) [DCID], row_number() over (partition by SchedulePeriod,scheduledate, StartTime,EndTime order by scheduledate,StartTime,EndTime,DCID) [SRNo] from AssignedDC DC with (NOLOCK) Left join Employee E on E.Emp_ID = DC.DCID where scheduleperiod in (Select SchedulePeriod from HouseScheduleImport where sessionID =@SessionID) and DC.houseid=@HouseID and DC.ID not in (Select CurrentShift From #TempImpDetail)) As ADC on ADC.SchedulePeriod = old.SchedulePeriod and ADC.ScheduleDate = old.SDATE and ADC.StartTime = old.starttime and ADC.EndTime = old.EndTime and old.CurrentShift =0 and old.SRNo = ADC.SRNo;	1	14	5.39	5.39		5.39	5.39	5.39	1,830.00	1,830	1,830	1,830	1,830	0.00	0	0	0	0	1,830.00	1,830		1,830	1,830	1,830

Object No.*	Object Name	Object Type		Avg. CPU Time (ms.)	Total CPU Time (ms.)	Total CPU Time (%)				# Avg. Logical Reads	# Total Logical Reads				# Avg. Logical Writes	# Total Logical Writes				# Avg. Logical IO	# Total Logical IO	Total Logical IO (%)			
	Select * From (Select P.SchedulePeriod, P.AccountID,P.HouseID,P.AgencyName,P.DCName, P.Confirmed, P.SDATE, P.StartTime,P.EndTime,P.DCNotMapped,P.DetailNotMatched, P.CurrentShift,P.ScheduleDCName,P.ScheduleConfirm,convert(datetime,SDATE + ' '+ StartTime) [stDate] --,[NewID] From #TempImpDetail as P Where InsertType = 2 Union all Select DC.Scheduleperiod,DC.AccountID,DC.HouseID,'' [AgencyName],'' [DCName],'' [Confirmed], DC.ScheduleDate,DC.StartTime,DC.EndTime,0 [DCNotMapped],1 [DetailNotMatched], isnull(DC.ID,0) [CurrentShift], Case when isnull (E.Emp_ID,0) >0 then EMP_LAST_NAME+ ', '+ EMP_FIRST_NAME else DC.[Type] end [ScheduleDCName], Case when isnull (DC.PunchCardClosed,0) = 1 then 'Yes' else 'No' end [ScheduleConfirm],convert (datetime,ScheduleDate + ' '+ StartTime) [stDate]--,0 from AssignedDC DC with (NOLOCK) Left join Employee E on E.Emp_ID = DC.DCID where scheduleperiod in (Select SchedulePeriod from HouseScheduleImport where sessionID =@SessionID) and DC.houseid=@HouseID and DC.ID not in (Select CurrentShift From #TempImpDetail)) as t Order by CASE WHEN @OrderByColumn + ' ' + @OrderbyDir = 'SDATE ASC' THEN [stDate] END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderbyDir = 'SDATE DESC' THEN [stDate] END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderbyDir = 'StartTime ASC' THEN SDATE END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderbyDir = 'StartTime DESC' THEN SDATE END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderbyDir = 'DCName ASC' THEN DCName END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderbyDir = 'DCName DESC' THEN DCName END DESC, CASE WHEN @OrderByColumn + ' ' + @OrderbyDir = 'ScheduleDCName ASC' THEN ScheduleDCName END ASC, CASE WHEN @OrderByColumn + ' ' + @OrderbyDir = 'ScheduleDCName DESC' THEN ScheduleDCName END DESC, [stDate] -- ,SchedulePeriod	1	15	4.78	4.78		4.78	4.78	4.78	1,812.00	1,812	1,812	1,812	1,812	0.00	0	0	0	0	1,812.00	1,812		1,812	1,812	1,812
24	[dbo]..uspGetHouseDirectBySchedulePeriod	SQL Stored-Procedure		93.15	1,397.24	17.08				78,270.53	1,174,058				349.13	5,237				78,619.67	1,179,295	43.68			
	SQL Statement	# Executions (With Last Plan)	# Plans Generated	Avg. CPU Time (ms.)	Total CPU Time (ms.)		Last CPU Time (ms.)	Min. CPU Time (ms.)	Max. CPU Time (ms.)	# Avg. Logical Reads	# Total Physical Reads	# Last Logical Reads	# Min. Logical Reads	# Max. Logical Reads	# Avg. Logical Writes	# Total Logical Writes	# Last Logical Writes	# Min. Logical Writes	# Max. Logical Writes	# Avg. Logical IO	# Total Logical IO		# Last Logical IO	# Min. Logical IO	# Max. Logical IO
	SELECT Employee.EMP_ID, Employee.EMP_FIRST_NAME + ' ' + Employee.EMP_LAST_NAME as EmpName, Employee.EMP_GENDER, Employee.EMP_TYPE, Employee.AMAP, EMP_FIRST_NAME + ' ' + Employee.EMP_LAST_NAME + -- case when EHD.Distance is not null then ' - ' + cast(cast (EHD.Distance as int) as varchar(10)) + 'M ' else '' end + ' - ' + Case When isnull (WL.id,0)>0 then 'HY' else 'HN' end + ' - M' + cast(cast(isnull (EHD.Distance,0) as int) as varchar(10)) + ' - H' + Format(isnull (ESM.TotalMinutes,0) /60.0,'0') [EmpDetail] --+ ':' + Format (isnull(tmpHours.[Minutes],0) % 60.0,'00') FROM Employee Left join EmployeeSchedulePeriodMinute ESM on ESM.EmployeeID = Employee.EMP_ID and ESM.SchedulePeriod =@SchedulePeriod Left join EmployeeHouseDistances EHD on EHD.EmployeeID = Employee.EMP_ID and EHD.HouseID = @HouseID Left join WorksAtList WL on WL.EmpId = Employee.EMP_ID and WL.HouseId = @HouseID Left join BlackList BL on BL.EmpId = Employee.EMP_ID and BL.HouseId = @HouseID WHERE (Employee.EMP_TYPE = '2') AND (isnull(BL.Id,0)=0) and (isnull(Employee.[Status],'')= 'Active') and ((@DCType = 'Male' and Employee.EMP_GENDER ='MALE') or (@DCType = 'FEMALE' and Employee.EMP_GENDER ='FEMALE') or (@DCType = 'ANY AMAP' and Employee.EMP_APMP=1) or (@DCType = 'AMAP MALE' and Employee.EMP_GENDER ='MALE' and Employee.EMP_APMP=1) or (@DCType = 'AMAP FEMALE' and Employee.EMP_GENDER																								

	='FEMALE' and Employee.EMP_APMP=1) or (@DCType not in ('Male','FEMALE','ANY AMAP','AMAP MALE','AMAP FEMALE'))) and (Employee.EMP_ID Not IN (Select isnull(DcId,0) From AssignedDC Where ScheduleDate=@ScheduleDate and (((charindex ('PM',StartTime) = 0 or charindex('AM',EndTime) = 0) and ((@ScheduleStart > CAST (ScheduleDate + ' ' + StartTime as DATETIME) and @ScheduleStart < CAST (ScheduleDate + ' ' + EndTime as DATETIME)) or (@ScheduleEnd > CAST (ScheduleDate + ' ' + StartTime as DATETIME) and @ScheduleEnd < CAST (ScheduleDate + ' ' + EndTime as DATETIME)) or (CAST(ScheduleDate + ' ' + StartTime as DATETIME) > @ScheduleStart and CAST (ScheduleDate + ' ' + StartTime as DATETIME) < @ScheduleEnd) or (CAST(ScheduleDate + ' ' + EndTime as DATETIME) > @ScheduleEnd and CAST (ScheduleDate + ' ' + EndTime as DATETIME) < @ScheduleEnd))) or ((charindex ('PM',StartTime) > 0 and charindex('AM',EndTime) > 0) and ((@ScheduleStart > CAST (ScheduleDate + ' ' + StartTime as DATETIME) and @ScheduleStart < Dateadd (d,1,CAST(ScheduleDate + ' '+ EndTime as DATETIME))) or (@ScheduleEnd > CAST (ScheduleDate + ' ' + StartTime as DATETIME) and @ScheduleEnd < Dateadd (d,1,CAST(ScheduleDate + ' '+ EndTime as DATETIME))) or (CAST(ScheduleDate + ' ' + StartTime as DATETIME) > @ScheduleStart and CAST (ScheduleDate + ' ' + StartTime as DATETIME) < @ScheduleEnd) or (Dateadd(d,1,CAST (ScheduleDate + ' '+ EndTime as DATETIME)) > @ScheduleEnd and Dateadd (d,1,CAST(ScheduleDate + ' '+ EndTime as DATETIME)) < @ScheduleEnd))) or (@ScheduleStart = CAST (ScheduleDate + ' ' + StartTime as DATETIME) and @ScheduleEnd = CAST (ScheduleDate + ' ' + EndTime as DATETIME)))) or (EMP_ID =isnull (@SelectedDC,0))) AND (ISNULL(@Keyword, '') = '' or Employee.EMP_FIRST_NAME like @Keyword + '%' or Employee.EMP_LAST_NAME like @Keyword + '%') ORDER BY Case when @SortDCBy ='Mileage' then isnull (EHD.Distance,0) End, Case when @SortDCBy ='Hours' then isnull (ESM.TotalMinutes,0) End, Case When isnull(WL.id,0)>0 then 1 else 2 end, Employee.EMP_FIRST_NAME + ' ' + Employee.EMP_LAST_NAME	15	1	93.15	1,397.24		89.51	88.77	103.12	78,270.53	1,174,058	77,974	77,961	79,156	349.13	5,237	276	276	366	78,619.67	1,179,295		78,250	78,237	79,522
25	[dbo]..uspHouseImportScheduleSetColumns	SQL Stored-Procedure		919.60	924.10	11.30				132,320.00	132,556				51.00	51				132,371.00	132,607	4.91			
	SQL Statement	**# Executions (With Last Plan)**	**# Plans Generated**	**Avg. CPU Time (ms.)**	**Total CPU Time (ms.)**		**Last CPU Time (ms.)**	**Min. CPU Time (ms.)**	**Max. CPU Time (ms.)**	**# Avg. Logical Reads**	**# Total Physical Reads**	**# Last Logical Reads**	**# Min. Logical Reads**	**# Max. Logical Reads**	**# Avg. Logical Writes**	**# Total Logical Writes**	**# Last Logical Writes**	**# Min. Logical Writes**	**# Max. Logical Writes**	**# Avg. Logical IO**	**# Total Logical IO**		**# Last Logical IO**	**# Min. Logical IO**	**# Max. Logical IO**
	Update HSI Set DCID = (select top 1 emp_id From Employee Where (Rtrim(Ltrim (EMP_LAST_NAME))+ ', '+ Rtrim(Ltrim (EMP_FIRST_NAME)) = Rtrim (Ltrim(hsi.DCName)) or Rtrim (Ltrim(EMP_LAST_NAME))+ ','+ Rtrim(Ltrim (EMP_FIRST_NAME)) = Rtrim (Ltrim(hsi.DCName)))), SchedulePeriod= dbo.ufnSchedulePeriodFromDate(HSI.SDate), AccountID = (Select Top 1 ID from Account Where [type] ='ACCOUNT' and Rtrim(Ltrim (AccountNumber)) = Rtrim (Ltrim(substring (HSI.AgencyName,1,charindex ('-',HSI.AgencyName+'-')-1)))) From houseScheduleImport HSI Where SessionID= @SessionID; -- Update HouseID	1	5	544.63	544.63		544.63	544.63	544.63	75,372.00	75,372	75,372	75,372	75,372	51.00	51	51	51	51	75,423.00	75,423		75,423	75,423	75,423

Object No.*	Object Name	Object Type		Avg. CPU Time (ms.)	Total CPU Time (ms.)	Total CPU Time (%)				# Avg. Logical Reads	# Total Logical Reads				# Avg. Logical Writes	# Total Logical Writes				# Avg. Logical IO	# Total Logical IO	Total Logical IO (%)			
	Update HSI Set HouseID = (Select Top 1 ID from Account Where [type] ='House' and ParentID=HSI.AccountID and Rtrim(Ltrim(AccountNumber)) = Rtrim(Ltrim(substring (HSI.AgencyName,charindex ('-',HSI.AgencyName)+1,500)))) From houseScheduleImport HSI Where charindex ('-',HSI.AgencyName) > 0 and SessionID= @SessionID; -- Update Confirm	1	6	17.02	17.02		17.02	17.02	17.02	3,921.00	3,921	3,921	3,921	3,921	0.00	0	0	0	0	3,921.00	3,921		3,921	3,921	3,921
	Update houseScheduleImport set DCStatus = case when isnull(DCStatus,") =" then 'No' else DCStatus end Where SessionID= @SessionID; -- Delete OPEN/TBA record when import for Actual schedule	3	1	2.25	6.75		3.06	1.58	3.06	118.00	354	166	70	166	0.00	0	0	0	0	118.00	354		166	70	166
	Update HSI Set DCID = (select top 1 emp_id From Employee Where (Rtrim(Ltrim (EMP_LAST_NAME))+ ', '+ Rtrim(Ltrim (EMP_FIRST_NAME)) = Rtrim (Ltrim(hsi.DCName)) or Rtrim (Ltrim(EMP_LAST_NAME))+ ','+ Rtrim(Ltrim (EMP_FIRST_NAME)) = Rtrim (Ltrim(hsi.DCName)))), SchedulePeriod= dbo.ufnSchedulePeriodFromDate(HSI.SDate), AccountID = (Select Top 1 ID from Account Where [type] ='ACCOUNT' and Rtrim(Ltrim (AccountNumber)) = Rtrim (Ltrim(substring (HSI.AgencyName,1,charindex ('-',HSI.AgencyName+'-')-1)))) From houseScheduleImport HSI Where SessionID= @SessionID; -- Update HouseID	1	1	345.32	345.32		345.32	345.32	345.32	50,110.00	50,110	50,110	50,110	50,110	0.00	0	0	0	0	50,110.00	50,110		50,110	50,110	50,110
	Update HSI Set HouseID = (Select Top 1 ID from Account Where [type] ='House' and ParentID=HSI.AccountID and Rtrim(Ltrim(AccountNumber)) = Rtrim(Ltrim(substring (HSI.AgencyName,charindex ('-',HSI.AgencyName)+1,500)))) From houseScheduleImport HSI Where charindex ('-',HSI.AgencyName) > 0 and SessionID= @SessionID; -- Update Confirm	1	2	9.44	9.44		9.44	9.44	9.44	2,765.00	2,765	2,765	2,765	2,765	0.00	0	0	0	0	2,765.00	2,765		2,765	2,765	2,765
	Update houseScheduleImport set DCStatus = case when isnull(DCStatus,") =" then 'No' else DCStatus end Where SessionID= @SessionID; -- Delete OPEN/TBA record when import for Actual schedule	1	1	0.95	0.95		0.95	0.95	0.95	34.00	34	34	34	34	0.00	0	0	0	0	34.00	34		34	34	34