```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--
===================================================================================
==
-- dba16_met_DMGetActiveBlockingConnectionHead
--
--
-- Calls:               None
--
-- Description: Get a list of active blocking connections. Including the Header.
--
-- Date                 Modified By                     Changes
-- 02/19/2014   Aron E. Tekulsky    Initial Coding.
-- 02/13/2018   Aron E. Tekulsky    Update to V140.
--
===================================================================================
==
--      Copyright©2001 - 2025 Aron Tekulsky.  World wide rights reserved.
--      Use of this code is free of charge but does not in any way imply transfer of

--      Copyright ownership to the user.
--
===================================================================================
==
--
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED

        DECLARE @blktab TABLE (
                        session_id                                      smallint,
                        ------blocking_session_id                       smallint,
                        blocking_session_id                     varchar(12),
                        start_time                                      datetime,
-- when the request arrived.
                        status
nvarchar(30),
                        ------transaction_isolation             smallint,       --
transaction_isolation_level
                        transaction_isolation           varchar(16),    --
transaction_isolation_level
```

```
                        last_wait_type                          nvarchar(60),
                        wait_time                                       int,
                        wait_resource                           nvarchar(256),
                        cpu_time                                        int,
-- CPU
                        IOReads                                         bigint, --=
r.logical_reads + r.reads
                        IOWrite                                         bigint, --
IOWrite
                        OpenTranCount                           int,    --
open_transaction_count
                        command
nvarchar(32), -- CommandType
                        database_id                                     smallint,
                        databasename                            nvarchar(128),
                        connect_time                            datetime,
                        protocol_type                           nvarchar(40),
                        SQLStmt
nvarchar(max),
                        query_plan                                      xml,
                        login_name
nvarchar(128),
                        login_time                                      datetime,
                        Host
nvarchar(128),
                        Protocol
nvarchar(40), --  net_transport
                        ConnectionWrites                        int,
                        ConnectionReads                         int,
                        ClientAddress                           varchar(48),
                        Authenticationn                         nvarchar(40),
                        SQLHandle
varbinary(64)
        )

-- ***** Get the Blocked processes *****

        INSERT INTO @blktab
        (session_id,
                                blocking_session_id,
                                start_time,
                                status,
                                transaction_isolation,
                                last_wait_type,
                                wait_resource,
                                wait_time,
                                cpu_time,
                                IOReads,
                                IOWrite,
                                OpenTranCount,
                                command,
                                database_id,
                                databasename,
                                connect_time,
                                protocol_type,
                                SQLStmt,
                                query_plan,
                                login_name,
                                login_time,
                                Host,
                                Protocol,
                                ConnectionWrites,
                                ConnectionReads,
                                ClientAddress,
                                Authenticationn,
                                SQLHandle)
        SELECT
                        s.session_id
                        ,CASE
                                WHEN R.blocking_session_id IS NULL THEN 'Head
Blocker'
                                ELSE CONVERT(varchar(12),R.blocking_session_id)
                                END AS blocking_session_id
                        , R.start_time -- when the request arrived.
                        ,CASE
                                WHEN R.status IS NULL THEN
                                        s.status
                                ELSE R.status
                         END AS TheStatus
                        , CASE s.transaction_isolation_level
                                WHEN 0 THEN 'Unspecified'
                                WHEN 1 THEN 'Read Uncommitted'
                                WHEN 2 THEN 'Read Committed'
                                WHEN 3 THEN 'Repeatable'
                                WHEN 4 THEN 'Serializable'
                                WHEN 5 THEN 'Snapshot'
                        END AS transaction_isolation
                        , r.last_wait_type AS last_wait_type
                        ,r.wait_resource
                        , ISNULL(0,r.wait_time) AS WaitTimeMs
                        , ISNULL(0,r.cpu_time) AS CPU
                        , IOReads = ISNULL(0,r.logical_reads + r.reads)
                        , ISNULL(0,r.writes) AS IOWrite
                        , CASE
                                WHEN r.open_transaction_count IS NULL THEN
s.open_transaction_count
```

```
                                ELSE r.open_transaction_count
                                END AS OpenTranCount
                        , R.command AS CommandType
                        ,CASE
                                WHEN R.database_id IS NULL THEN s.database_id
                                ELSE r.database_id
                                END AS TheDbid
                        , CASE
                                WHEN R.database_id IS NULL THEN
DB_NAME(s.database_id)
                                ELSE  DB_NAME(r.database_id)
                                END AS database_name
                        , C.connect_time
                        , C.protocol_type
                        ,q.text AS SQLStmt
                        ,p.query_plan
                        ,S.login_name
                        , S.login_time
                        , s.host_name AS Host
                        , c.net_transport AS Protocol
                        , c.num_writes AS ConnectionWrites
                        , c.num_reads AS ConnectionReads
                        , c.client_net_address AS ClientAddress
                        , c.auth_scheme AS Authentication
                        ,r.sql_handle
                FROM sys.dm_exec_sessions S
                        LEFT OUTER JOIN sys.dm_exec_requests R on (R.session_id =
S.session_id)
                        LEFT OUTER JOIN sys.dm_exec_connections C on
(R.connection_id = C.connection_id)
                        CROSS APPLY sys.dm_exec_sql_text(R.sql_handle) as q
                        CROSS APPLY sys.dm_exec_query_plan(R.plan_handle) p
        WHERE s.session_id > 50
                AND (s.database_id > 4 OR r.database_id > 4)
                AND (r.status IN ('suspended', 'sleeping') OR s.status IN ('Sleeping
'))
                AND R.blocking_session_id IS NOT NULL

-- ***** Add The Head blockers *****

        INSERT INTO @blktab
        (session_id,
                                blocking_session_id,
                                start_time,
                                status,
                                transaction_isolation,
                                last_wait_type,
                                wait_resource,
                                wait_time,
                                cpu_time,
                                IOReads,
                                IOWrite,
                                OpenTranCount,
                                command,
                                database_id,
                                databasename,
                                connect_time,
                                protocol_type,
                                SQLStmt,
                                login_name,
                                login_time,
                                Host,
                                Protocol,
                                ConnectionWrites,
                                ConnectionReads,
                                ClientAddress,
                                Authenticationn--,
                                )

                SELECT
                        c.session_id
                        ,0
                        , R.start_time -- when the request arrived.
                        ,CASE
                                WHEN R.status IS NULL THEN
                                        s.status
                                ELSE R.status
                         END AS TheStatus
                        , CASE s.transaction_isolation_level
                                WHEN 0 THEN 'Unspecified'
                                WHEN 1 THEN 'Read Uncommitted'
                                WHEN 2 THEN 'Read Committed'
                                WHEN 3 THEN 'Repeatable'
                                WHEN 4 THEN 'Serializable'
                                WHEN 5 THEN 'Snapshot'
                        END AS transaction_isolationS
                        , ISNULL('MISCELLANEOUS',r.last_wait_type) AS last_wait_type
                        ------, r.last_wait_type AS last_wait_type
                        ,r.wait_resource
                        , ISNULL(0,r.wait_time) AS WaitTimeMs
                        , ISNULL(0,r.cpu_time) AS CPU
                        , IOReads = ISNULL(0,r.logical_reads + r.reads)
                        , ISNULL(0,r.writes) AS IOWrite
                        , CASE
                                WHEN r.open_transaction_count IS NULL THEN
s.open_transaction_count
```

```
                                ELSE r.open_transaction_count
                                END AS OpenTranCount
                        , ISNULL('AWAITING COMMAND', R.command) AS CommandType
                        ----------, R.command AS CommandType
                        ,CASE
                                WHEN R.database_id IS NULL THEN s.database_id
                                ELSE r.database_id
                                END AS TheDbid
                        , CASE
                                WHEN R.database_id IS NULL THEN
DB_NAME(s.database_id)
                                ELSE  DB_NAME(r.database_id)
                                END AS database_name
                        , C.connect_time
                        , C.protocol_type
                        ,q.text AS SQLStmt
                        ,S.login_name
                        , S.login_time
                        , s.host_name AS Host
                        , c.net_transport AS Protocol
                        , c.num_writes AS ConnectionWrites
                        , c.num_reads AS ConnectionReads
                        , c.client_net_address AS ClientAddress
                        , c.auth_scheme AS Authentication
                FROM sys.dm_exec_connections C
                        LEFT OUTER JOIN sys.dm_exec_requests R on (R.session_id =
c.session_id)
                        JOIN sys.dm_exec_sessions s ON (c.session_id = s.session_id
)
                        CROSS APPLY sys.dm_exec_sql_text(c.most_recent_sql_handle )
as q
        WHERE c.session_id > 50
                AND (s.database_id > 4 OR r.database_id > 4)
                AND R.blocking_session_id IS NULL


-------- ***** END head blocker *****

        SELECT t.session_id,
                                t.blocking_session_id,
                                t.start_time,
                                t.status,
                                t.transaction_isolation AS transaction_isolation,
                                t.last_wait_type,
                                t.wait_time,
                                t.wait_resource,
                                t.cpu_time,
                                t.IOReads,
                                t.IOWrite,
                                t.OpenTranCount,
                                t.command,
                                t.database_id,
                                t.databasename,
                                t.connect_time,
                                t.protocol_type,
                                t.SQLStmt,
                                t.query_plan,
                                t.login_name,
                                t.login_time,
                                t.Host,
                                t.Protocol,
                                t.ConnectionWrites,
                                t.ConnectionReads,
                                t.ClientAddress,
                                t.Authenticationn
                        FROM @blktab t
                ORDER BY t.blocking_session_id ASC




END
GO
```