Sk Kullanlan Gelimi T-SQL Sorgular

Form numaras ile detayl süreç ilgilerine ulamak için kullanlr.

SQL

```
select wi.Library, l.Name, p.Name, pv.Id, pv.Version, wi.Id 'WorkItemId', wi.State, wi.Name, wi.EndDate, wi.
SelectedAction
from wflInstances i,  wflWorkItems wi, wflProcessVersions pv, wflProcesses p, wflLibraries l
where i.Number=47700
and i.Id = wi.Instance
and pv.Process = p.Id
and wi.ProcessVersion = pv.Id
and l.Id = wi.Library
order by i.EndDate desc
```

Sql üzerinde optimizasyon veya tablolarla ilgili gelimi sorgu ve görüntülemeler için bu bölümde bulunan sorgular kullanabiliriz.

Tablo alanlar listeleme - tbl.name ile tablo ad içerisinde veya col.name ile alan ad içerisinde

```
select tbl.name 'TableName', col.name 'ColumnName', typ.name, col.length from syscolumns col,
 sys.tables tbl,  systypes typ
where col.id = tbl.object_id and tbl.name like 'wfl%'
and col.xtype = typ.xtype
-- and col.name like '%email%'
order by tbl.name, colid
```

Veri taban tablo alan kullanm

```
exec sp_spaceused
```

Index listesi

```
select 'table_name'=object_name(i.id)  ,i.indid
,'index_name'=i.name  ,i.groupid
,'filegroup'=f.name  ,'file_name'=d.physical_name
,'dataspace'=s.name from sys.sysindexes i
,sys.filegroups f  ,sys.database_files d
,sys.data_spaces s
where objectproperty(i.id,'IsUserTable') = 1
and f.data_space_id = i.groupid
and f.data_space_id = d.data_space_id
and f.data_space_id = s.data_space_id
order by f.name,object_name(i.id),groupid
go
```

Alanlarla ndex

```
select sys.objects.object_id,
sys.objects.name as tablename,
sys.indexes.name as indexname,
sys.columns.name as columnname
from sys.objects
inner join sys.indexes
on sys.objects.object_id = sys.indexes.object_id
inner join sys.index_columns
on sys.index_columns.object_id = sys.indexes.object_id
and sys.index_columns.index_id = sys.indexes.index_id
inner join sys.columns
on sys.columns.object_id = sys.index_columns.object_id
and sys.columns.column_id = sys.index_columns.column_id
where sys.objects.type_desc  = 'USER_TABLE'
and sys.indexes.is_primary_key = 0
order by tablename, indexname, columnname
```

Tablo isimleri, satr saylar ve kaplad alanlar listeleyen prosedür. Parametre almaz. :exec GetAllTableSizes

```
CREATE PROCEDURE GetAllTableSizes
AS
/*
    Obtains spaced used data for ALL user tables in the database
*/
DECLARE @TableName VARCHAR(100)    --For storing values in the cursor

--Cursor to get the name of all user tables from the sysobjects listing
DECLARE tableCursor CURSOR
FOR
select [name]
from dbo.sysobjects
where  OBJECTPROPERTY(id, N'IsUserTable') = 1
FOR READ ONLY

--A procedure level temp table to store the results
CREATE TABLE #TempTable
(
    tableName varchar(100),
    numberofRows int,
    reservedSize varchar(50),
    dataSize varchar(50),
    indexSize varchar(50),
    unusedSize varchar(50)
)

--Open the cursor
OPEN tableCursor

--Get the first table name from the cursor
FETCH NEXT FROM tableCursor INTO @TableName

--Loop until the cursor was not able to fetch
WHILE (@@Fetch_Status >= 0)
BEGIN
    --Dump the results of the sp_spaceused query to the temp table
    INSERT  #TempTable
        EXEC sp_spaceused @TableName

    --Get the next table name
    FETCH NEXT FROM tableCursor INTO @TableName
END

--Get rid of the cursor
CLOSE tableCursor
DEALLOCATE tableCursor

--Select all records so we can use the reults
SELECT *
FROM #TempTable order by 2 desc

--Final cleanup!
DROP TABLE #TempTable

GO
```

Veri taban üzerinde obje bulma If-Else

```
--SP
IF EXISTS (SELECT * FROM sysobjects WHERE id = object_id(N'[dbo].[SP_YIL_AY_GUN_INSERT]') AND OBJECTPROPERTY
(id, N'IsProcedure') = 1)
BEGIN
        DROP PROCEDURE dbo.[SP_YIL_AY_GUN_INSERT]
END

--index
IF EXISTS (SELECT name FROM sysindexes WHERE name = 'IDX_EBOrderNO')
        drop index [IDX_EBOrderNO] ON [dbo].[schBayer_OrderHeaders]
go

-- IF exists table
IF EXISTS (SELECT * FROM sysobjects WHERE xtype = 'U' AND name = 'schBayer_Products')
BEGIN
        DROP TABLE dbo.[schBayer_Products]
END
```

Eer veri tabanna sk sk eriiliyorsa AUTO_CLOSE OFF yaplmal

```
exec sp_MsForEachDB ' ALTER DATABASE [?] SET AUTO_CLOSE OFF  '
```

Prosedürlerin Okuma-Yazma Süreleri

```
  SELECT TOP 10
ObjectName          = OBJECT_NAME(qt.objectid)
,DiskReads          = qs.total_physical_reads -- The worst reads, disk reads
,MemoryReads        = qs.total_logical_reads  --Logical Reads are memory reads
,Executions         = qs.execution_count
,AvgDuration        = qs.total_elapsed_time / qs.execution_count
,CPUTime            = qs.total_worker_time
,DiskWaitAndCPUTime = qs.total_elapsed_time
,MemoryWrites       = qs.max_logical_writes
,DateCached         = qs.creation_time
,DatabaseName       = DB_Name(qt.dbid)
FROM sys.dm_exec_query_stats AS qs
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) AS qt
WHERE qt.dbid = db_id() -- Filter by current database
ORDER BY qs.total_elapsed_time DESC
```

En uzun süreyi alan sorgular listesini kontrol eder.

```
SELECT TOP 100
    qs.total_elapsed_time / qs.execution_count / 1000000.0 AS average_seconds,
    qs.total_elapsed_time / 1000000.0 AS total_seconds,
    qs.execution_count,
    SUBSTRING (qt.text,qs.statement_start_offset/2,
         (CASE WHEN qs.statement_end_offset = -1
            THEN LEN(CONVERT(NVARCHAR(MAX), qt.text)) * 2
          ELSE qs.statement_end_offset END - qs.statement_start_offset)/2) AS individual_query,
    o.name AS object_name,
    DB_NAME(qt.dbid) AS database_name
  FROM sys.dm_exec_query_stats qs
    CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) as qt
    LEFT OUTER JOIN sys.objects o ON qt.objectid = o.object_id
where qt.dbid = DB_ID()
  ORDER BY average_seconds DESC;
-- Queries doing most I/O:
SELECT TOP 100
    (total_logical_reads + total_logical_writes) / qs.execution_count AS average_IO,
    (total_logical_reads + total_logical_writes) AS total_IO,
    qs.execution_count AS execution_count,
    SUBSTRING (qt.text,qs.statement_start_offset/2,
         (CASE WHEN qs.statement_end_offset = -1
            THEN LEN(CONVERT(NVARCHAR(MAX), qt.text)) * 2
          ELSE qs.statement_end_offset END - qs.statement_start_offset)/2) AS indivudual_query,
    o.name AS object_name,
    DB_NAME(qt.dbid) AS database_name
  FROM sys.dm_exec_query_stats qs
    CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) as qt
    LEFT OUTER JOIN sys.objects o ON qt.objectid = o.object_id
where qt.dbid = DB_ID()
  ORDER BY average_IO DESC;
```

SQL - index fragmentation script - 30 dan yüksek parçalanmaya sahip indexleri listeler, bu indexlerin yeniden rebuild edilmesi gerekir.

```
SELECT OBJECT_NAME(ind.OBJECT_ID) AS TableName,
ind.name AS IndexName, indexstats.index_type_desc AS IndexType,
indexstats.avg_fragmentation_in_percent
FROM sys.dm_db_index_physical_stats(DB_ID(), NULL, NULL, NULL, NULL) indexstats
INNER JOIN sys.indexes ind
ON ind.object_id = indexstats.object_id
AND ind.index_id = indexstats.index_id
WHERE indexstats.avg_fragmentation_in_percent > 30
ORDER BY indexstats.avg_fragmentation_in_percent DESC
```