Database Console Commands

DBCC (Database Console Commands) in SQL Server is a set of commands used for maintenance, validation, and status checking of databases, tables, indexes, and other SQL Server objects. Here's a summary of some key DBCC commands with examples and their outputs:

1. DBCC CHECKDB

- **Purpose**: Checks the logical and physical integrity of all the objects in a specified database.
- **Syntax**:

```sql
DBCC CHECKDB (database_name)
```

- **Example:**

```sql
DBCC CHECKDB (AdventureWorks2019)
```

- **Output**: This will display a series of messages indicating if there are any consistency issues or if everything is fine. A typical output might look like:

```yaml
DBCC results for 'AdventureWorks2019'.
Service Broker Msg 9675, State 1: Message Types analyzed: 14.
Service Broker Msg 9676, State 1: Service Contracts analyzed: 6.
DBCC results for 'AdventureWorks2019'.
No allocation or consistency errors were found in database 'AdventureWorks2019'.
DBCC execution completed. If DBCC printed error messages, contact your system administ
```

2. DBCC CHECKTABLE

- **Purpose**: Checks the integrity of a single table or indexed view.
- **Syntax**:

```sql
DBCC CHECKTABLE (table_name)
```

- **Example**:

```sql
DBCC CHECKTABLE ('AdventureWorks2019.Person.Person')
```

- **Output**: This will check the table and display any errors found. The output could be:

```sql
DBCC results for 'Person'.
There are 19972 rows in 137 pages for object 'Person'.
DBCC execution completed. If DBCC printed error messages, contact your system administ
```

3. DBCC CHECKALLOC

- **Purpose**: Verifies the consistency of disk space allocation structures for a specific database.
- **Syntax**:

```sql
DBCC CHECKALLOC (database_name)
```

- **Example**:

```sql
DBCC CHECKALLOC (AdventureWorks2019)
```

- **Output**: The output will show whether there are allocation errors in the database.

```go
DBCC results for 'AdventureWorks2019'.
No allocation errors found in database 'AdventureWorks2019'.
DBCC execution completed. If DBCC printed error messages, contact your system administ
```

4. DBCC SQLPERF

- **Purpose**: Provides statistics on transaction log space usage.
- **Syntax**:

```sql
DBCC SQLPERF(logspace)
```

- **Example**:

```sql
DBCC SQLPERF(logspace)
```

- **Output**: It will show how much of the transaction log is in use for each database.

```markdown
Database Name               Log Size (MB)  Log Space Used (%)  Status
-------------------------------------------------------------
AdventureWorks2019           1024           7.5                 0
master                       50             3.5                 0
```

5. DBCC SHRINKDATABASE

- **Purpose**: Shrinks the size of the data and log files in the specified database.
- **Syntax**:

```sql
DBCC SHRINKDATABASE (database_name, target_percent)
```

- **Example**:

```sql
DBCC SHRINKDATABASE (AdventureWorks2019, 10)
```

- **Output**: It will shrink the database files and display the results showing the new size. The output will look like:

```markdown
DbId   FileId   CurrentSize   MinimumSize   UsedPages   EstimatedPages
---------------------------------------------------------------------
8      1        10240         2048          8192        7680
```

6. DBCC SHRINKFILE

- **Purpose**: Shrinks the size of a specific data or log file.
- **Syntax**:

```sql
DBCC SHRINKFILE (file_name, target_size)
```

- **Example**:

```sql
DBCC SHRINKFILE ('AdventureWorks2019_Log', 100)
```

- **Output**: This command will reduce the size of the specified file. The output will be:

```markdown
DbId   FileId   CurrentSize   MinimumSize   UsedPages   EstimatedPages
---------------------------------------------------------------------
8      2        1024          512           512         256
```

7. DBCC DROPCLEANBUFFERS

- **Purpose**: Clears the buffer cache, useful in testing the performance of queries with a "cold" cache.
- **Syntax**:

```sql
DBCC DROPCLEANBUFFERS
```

- **Example**:

```sql
DBCC DROPCLEANBUFFERS
```

- **Output**: No result set is returned unless you use the WITH NO_INFOMSGS option. The buffer cache will be cleared silently.

8. DBCC FREEPROCCACHE

- **Purpose**: Clears the procedure cache, forcing SQL Server to recompile queries the next time they are executed.
- **Syntax**:

```sql
DBCC FREEPROCCACHE
```

- **Example**:

```sql
DBCC FREEPROCCACHE
```

- **Output**: This clears the cache and does not return a result set, but the cache will now be free.

9. DBCC CHECKIDENT

- **Purpose**: Checks the current identity value for a table and corrects it if necessary.
- **Syntax**:

```sql
DBCC CHECKIDENT (table_name, NORESEED | RESEED, new_seed)
```

- **Example**:

```sql
DBCC CHECKIDENT ('AdventureWorks2019.Person.Person', RESEED, 1000)
```

- **Output**: This will reset the identity value of the table to the specified value and display:

```sql
Checking identity information: current identity value '999', current column value '999
DBCC execution completed. If DBCC printed error messages, contact your system administ
```

10. DBCC TRACEON and DBCC TRACEOFF

- **Purpose**: Enables or disables a specific trace flag for troubleshooting purposes.
- **Syntax**:

```sql
DBCC TRACEON (trace_flag)
DBCC TRACEOFF (trace_flag)
```

- **Example**:

```sql
DBCC TRACEON (3604)  -- Displays output to the client
DBCC TRACEOFF (3604) -- Turns off the output
```

- **Output**: No output will be generated unless the trace flag produces it.

11. DBCC INPUTBUFFER

- **Purpose**: Displays the last statement sent from a client to SQL Server for a specific session.
- **Syntax**:

```sql
DBCC INPUTBUFFER (session_id)
```

- **Example**:

```sql
EventType      Parameters    EventInfo
-------------------------------------
Language       1             SELECT * FROM AdventureWorks2019.Person.Person
```

- **Output**: This will display the last statement executed by the session.

12. DBCC SHOW_STATISTICS

- **Purpose**: Displays statistics information for a table or indexed view.
- **Syntax**:

```sql
DBCC SHOW_STATISTICS (table_or_indexed_view_name, target)
```

- **Example**:

```sql
DBCC SHOW_STATISTICS ('AdventureWorks2019.Person.Person', 'IX_Person_LastName')
```

- **Output**: Displays the statistics for the specified index or table, showing data distribution for optimization.

```markdown
Updated              Rows     Rows Sampled    Steps    Density     Average Key Length
-------------------------------------------------------------------------------
2023-01-15 10:30 AM  500000   500000          200      0.0025      12
```

These commands can be highly useful for maintaining database health, troubleshooting performance issues, and diagnosing integrity problems within your SQL Server environment.