

*MICROSOFT® SQL SERVER™  
TPC-E BENCHMARK KIT  
DATABASE SETUP REFERENCE*

Version 1.2.0-1000  
© Copyright Microsoft, 2007

Jamie Reding ([jamiere@microsoft.com](mailto:jamiere@microsoft.com))  
SQL Server Performance Team  
Microsoft Corporation

**CONTENTS**

***MICROSOFT® SQL SERVER™ TPC-E BENCHMARK KIT DATABASE SETUP ..... 1***

**INTRODUCTION ..... 3**

**TPC-E BENCHMARK DATABASE SETUP ..... 4**

## INTRODUCTION

This document provides a reference for all the steps required to build a TPC-E database using the Microsoft TPC-E Benchmark Kit. It includes the names of the scripts used and the order of execution of these scripts.

## TPC-E BENCHMARK DATABASE SETUP

1. Remove any existing TPC-E database.
  - a. This step ensures that any previous TPC-E database is removed and that SQL Server is ready to begin the creation of a new TPC-E database.
  - b. Script Name:     Remove\_Database.sql
2. Create the database.
  - a. The Create Database script defines the file groups and the files within the file groups that SQL Server will use to store the TPC-E data.
  - b. Script Name:     Create\_Database.sql
3. Create the TPC-E Data Type synonyms.
  - a. The TPC-E synonyms map the native data types in SQL Server to the data types defined in the TPC-E specification.
  - b. Example:     bigint (SQL Server native data type) → IDENT\_T (TPC-E data type)
  - c. Script Name:     Create\_TPCE\_Types.sql
4. Create the TPC-E scaling tables and appropriate check constraints.
  - a. This step creates the scaling TPC-E tables as defined in Clause 2.6.1.10 in the TPC-E specification.
  - b. The script also creates the check constraints on the scaling tables as required by the TPC-E specification in Clauses 2.2.4, 2.2.5., 2.2.6, and 2.2.7.
  - c. Script Name:     Tables\_Scaling.sql
5. Create the TPC-E growing tables.
  - a. This step creates the growing TPC-E tables as defined in Clause 2.6.1.11 in the TPC-E specification.
  - b. The script also creates the check constraints on the growing tables as required by the TPC-E specification in Clauses 2.2.4, 2.2.5., 2.2.6, and 2.2.7.
  - c. Script Name:     Tables\_Growing.sql
6. Create the TPC-E fixed tables.
  - a. This step creates the fixed TPC-E tables as defined in Clause 2.6.1.9 in the TPC-E specification.

- b. The script also creates the check constraints on the fixed tables as required by the TPC-E specification in Clauses 2.2.4, 2.2.5., 2.2.6, and 2.2.7.
        - c. Script Name:     Tables\_Fixed.sql
7. Install the TPC-E stored procedures.
  - a. Script Names:
    - BrokerVolume.sql
    - CustomerPosition.sql
    - MarketFeed.sql
    - MarketWatch.sql
    - SecurityDetail.sql
    - TradeOrder.sql
    - TradeResult.sql
    - TradeStatus.sql
    - TradeLookup.sql
    - TradeUpdate.sql
    - DataMaintenance.sql
8. Install the Trade ID generation stored procedure.
  - a. This step installs the Get\_Next\_T\_ID stored procedure. This stored procedure generates the next T\_ID (trade ID) for the runtime system to use.
  - b. Script Name:     Get\_Next\_T\_ID.sql
9. Setup the pre-load database options.
  - a. This script sets the TPC-E database options for bulk loading.
  - b. Script Name:     Database\_Options\_1.sql
10. Load the database using the TPC-E provided EGenLoader.
  - a. This step utilizes one or more Windows command files that are generated by the Microsoft TPC-E Benchmark Kit. These command files contain the appropriate EGenLoader settings based on the number of customers and the parallelism desired.
  - b. The EGenLoader.exe has the following input parameters:
    - b     Beginning customer ordinal position
    - c     Number of customers (for this instance)
    - t     Number of customers (total in the database)
    - f     Scale factor (customers per 1 tpsE)
    - w     Number of Workdays (8-hour days)
    - l     Directory for input files
    - l     Type of load
    - s     Database server
    - d     Database name

c. Example: (10,000 Customers and 2 instances of EGenLoader)

```
• ** EGenLoaderInstance1.cmd **
@ECHO OFF
@CD C:\MSTPCE.1.2.0-1000\EGen\Bin\Release\
@EGenLoader.exe -b 1 -c 5000 -t 10000 -f 500 -w 300 -i C:\MSTPCE. 1.2.0-
1000\EGen\Flat_In -l ODBC -s SQLPERFTPCE1 -d tpce
@GOTO PostProcessLoader$errorlevel%
:PostProcessLoader1
:PostProcessLoader2
:PostProcessLoader3
@ECHO.
@ECHO !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
@ECHO This instance of EGenLoader has encountered an error.
@ECHO Please capture any error information prior to pressing
@ECHO the Enter key since this window will close at that time.
@ECHO.
@ECHO The TPC-E build is being aborted.
@ECHO !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
@ECHO.
@cd..\..\SetupVBS
@copy EGenInstanceError1.txt ..\EGenInstanceError1.txt >nul
@CD ..
PAUSE
@GOTO END
:PostProcessLoader0
:END
@cd..\..\SetupVBS
@copy EGenInstanceFlag1.txt ..\EGenInstanceFlag1.txt >nul
@CD ..

• ** EGenLoaderInstance2.cmd **
@ECHO OFF
@CD C:\MSTPCE. 1.2.0-1000\EGen\Bin\Release\
@EGenLoader.exe -b 5001 -c 5000 -t 10000 -f 500 -w 300 -i C:\MSTPCE. 1.2.0-
1000\EGen\Flat_In -l ODBC -s SQLPERFTPCE1 -d tpce
@GOTO PostProcessLoader$errorlevel%
:PostProcessLoader1
:PostProcessLoader2
:PostProcessLoader3
@ECHO.
@ECHO !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
@ECHO This instance of EGenLoader has encountered an error.
@ECHO Please capture any error information prior to pressing
@ECHO the Enter key since this window will close at that time.
@ECHO.
@ECHO The TPC-E build is being aborted.
@ECHO !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
@ECHO.
```

```

@cd..\..\SetupVBS
@copy EGenInstanceError2.txt ..\EGenInstanceError2.txt >nul
@CD ..
PAUSE
@GOTO END
:PostProcessLoader0
:END
@cd..\..\SetupVBS
@copy EGenInstanceFlag2.txt ..\EGenInstanceFlag2.txt >nul
@CD ..

```

11. Create the clustered indexes on the TPC-E fixed tables.

- a. This step creates the clustered indexes on the TPC-E tables that the TPC-E specification defines as a fixed table. (Clause 2.6.1.9)
- b. Script Name: Create\_Clustered\_Indexes\_Fixed.sql

12. Create the clustered indexes on the TPC-E growing tables.

- a. This step creates the clustered indexes on the TPC-E tables that the TPC-E specification defines as a growing table. (Clause 2.6.1.11)
- b. Script Name: Create\_Clustered\_Indexes\_Growing.sql

13. Create the clustered indexes on the TPC-E scaling tables.

- a. This step creates the clustered indexes on the TPC-E tables that the TPC-E specification defines as a scaling table. (Clause 2.6.1.10)
- b. Script Name: Create\_Clustered\_Indexes\_Scaling.sql

14. Create the non-clustered indexes on the TPC-E fixed tables.

- a. This step creates the non-clustered indexes on the TPC-E tables that the TPC-E specification defines as a fixed table. (Clause 2.6.1.9)
- b. Script Name: Create\_NC\_Indexes\_Fixed.sql

15. Create the non-clustered indexes on the TPC-E scaling tables.

- a. This step creates the non-clustered indexes on the TPC-E tables that the TPC-E specification defines as a scaling table. (Clause 2.6.1.10)
- b. Script Name: Create\_NC\_Indexes\_Scaling.sql

16. Create the non-clustered indexes on the TPC-E growing tables.

- a. This step creates the non-clustered indexes on the TPC-E tables that the TPC-E specification defines as a growing table. (Clause 2.6.1.11)

- b. Script Name: Create\_NC\_Indexes\_Growing.sql

17. Create the foreign key constraints.

- a. The script creates the foreign key relationships that are defined in Clauses 2.2.4, 2.2.5, 2.2.6, and 2.2.7.
- b. Script Name: Create\_FK\_Constraints.sql

18. Setup the post-load database options.

- a. This script sets the TPC-E database options for runtime conditions.
- b. Script Name: Database\_Options\_2.sql

19. Populate the Trade ID Ranges table

- a. This step creates a TID Ranges table and sets the starting point for trade ID generation during runtime.
- b. Script Name: Create\_TID\_Ranges\_Table.sql

After the successful completion of the above steps in order, the TPC-E database is created, populated, and prepared for runtime.

