Introduction
Perhaps the most common tasks database developers face are those related to creating new databases and populating them with existing data. Typically, these require developers to employ a variety of manual and automated processes using many different tools and techniques.

Manual SQL Commands
In the scope of relational databases that presents a SQL based interface, creation of a database is part of the Data Definition Language. Specifically, there is a CREATE command that may be used to define a database, its tables, and specific properties of each. For example, the CREATE command in T-SQL allows users to define the size of database files and policies about the database. The variation, CREATE TABLE, allows developers to specify tables, columns, datatypes, and attributes. A user may directly enter these commands using a command prompt, a SQL tool, or another mechanism for executing SQL.

Example 1: Manually Creating a Database and Table using SQL Data Definition Language
First, a database named “LTRACK” is created. LOG ON will specify where to store the database log. Lastly, lhistory table is created in LTRACK.

USE Master
GO
CREATE Database LTRACK
ON
( NAME = LTRACK,
 FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL\Data\LTRACK.mdf',
 SIZE = 10,
 FILEGROWTH = 5)

LOG ON
( NAME = LTRACK_Log',
 FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL\Data\LTRACK_Log.ldf',
 SIZE = 5,
 FILEGROWTH = 5)
GO

USE LTRACK
GO
CREATE TABLE lhistory ( DrawDate smalldatetime NOT NULL,
Creating and Managing Databases

DrawValue1 smallint NOT NULL,
DrawValue2 smallint NOT NULL
)

Tool Support
Because the SQL commands for creating a database are powerful and because there are many parameters that must be specified, directly writing SQL scripts to create databases is a tedious and complex process. It can lead to errors. For many products such as Microsoft’s SQL Server, functionality to create a database using SQL is encapsulated by a graphical interface that gives users an alternative method to specify requirements and attributes for a new database. Instead of keying SQL commands, developers only select options and specify parameters. The functionality supporting the interface creates the necessary SQL statements and executes them.

Example 2
The following is an example of using Enterprise Manager to automatically create the table and database previously described in Example 1.
Automation
A third option is to employ a tool that moves data from an external data store to a database. Often, data is stored externally from a database in a format that already describes the schema information necessary to create a database. A tool, such as Data Transfer Services (DTS) in SQL Server, can locate such data stores and automatically generate the SQL statements to create a database with the same schema. In a similar fashion, many external data applications contain similar functionality that allows users to export data in a specific database format or even directly to a database server.

Example 3
Here are the DTS screens used to specify source, destination, and table conversions. DTS automatically creates databases and tables if necessary.
Creating and Managing Databases

No One Solution is Best
Developers should not, however, rely solely on one single mechanism for creating databases. Automated tools and graphical interfaces that generate SQL scripts for creating databases remove control from the process. There is no guarantee that they will provide a user with desirable or even controllable results. For example, an import tool might automate specification of data types for columns in a table, which is a convenience, but if it decides the column is a numeric field when it should be a text field, it will cause future problems. Similarly, tools for creating tables often make assumptions such as how to treat null values without necessarily exposing those assumptions to users. Manually creating a new database, however, may not be feasible when given time and scope restrictions on a project. Generally, creating a database of any complexity will require the use of both tools and specific SQL commands to achieve a desired result.

**Summary**

Developers who are required to create tables or carry out related data import and export tasks should be proficient with both tools and the SQL data definition language. Developers should consider both the advantages and limitations of all available methods, whether manual or automated, and tailor solutions accordingly.