Time Solution Book Tutorial

By Eugene Asahara

Last Updated: June 4, 2025

This document serves as a living companion to tutorials and examples in the book, *Time Molecules*. It covers examples and tutorials that are either more nuanced (beyond what seems appropriate in the book) or still under development.

Status

As of June 4, 2025—the release date of *Time Molecules*—the majority of tutorials are fully implemented as explorations through the SQL Server TimeSolution database. These are packaged as a series of scripts named Time_Molecules_CodeXX.sql, located in the book_code\sql directory of the GitHub repository– https://github.com/MapRock/TimeMolecules/ Each SQL script corresponds to code listed in the book and is ready to run as-is in SQL Server Management Studio (SSMS)—just copy and execute. Specifically, from Code 4 on page 105 through Code 75 on page 245.

I'm planning an update to the GitHub material on August 1, 2025.

Page 161 – Model Stationary Distribution

The TimeSolution.py script, located in the GitHub repository under book_code/src, automates the calculation of the stationary distribution for each Markov model defined in the database. It connects to SQL Server, retrieves the transition matrix using the function [dbo].[ModelMatrix](ModelID), and runs iterative matrix multiplication to converge on the stationary probabilities.

What it does:

- Loads transition probabilities from the ModelMatrix function.
- Verifies that all events appear in both source (EventA) and target (EventB) roles, and checks for zero-probability transitions.
- Initializes a probability vector (starting in one state).
- Repeatedly multiplies the vector by the transition matrix until the probabilities converge (default 15 iterations or until steady state is reached).
- Saves the result into the Model_Stationary_Distribution table in SQL Server.

How to use it:

1. Ensure your .env file is properly configured with your SQL Server name and database name (TIMESOLUTION_SERVER_NAME, TIMESOLUTION_DATABASE_NAME).
2. Run TimeSolution.py from the book_code/src directory in Python.
3. The script clears existing values in the *Model_Stationary_Distribution* table, then processes each ModelID found in the Models table.

The result is a set of long-term probabilities for each event, giving you a snapshot of where the system tends to reside over time—a powerful insight for understanding system equilibrium.

Page 233 – Create SQL Server Metadata

This section describes how to load SQL Server and Kyvos metadata into the Time Solution environment and publish it to the Neo4j Data Catalog.

Step 1: Import Kyvos Metadata into SQL Server

1. Ensure that the CSV file kyvos_trial_cluster.csv is located in the demo_output directory.
2. Use SQL Server Management Studio (SSMS) to import the file into the table [TimeSolution].[STAGE].[kyvos_trial_cluster]:
    - Right-click the database → Tasks → Import Flat File...
    - Select the kyvos_trial_cluster.csv file.
    - Set the destination table as STAGE.kyvos_trial_cluster.
    - Follow the wizard to complete the import.

Step 2: Run SQL Server Metadata Extraction Script

1. Open sql_server_entire_server_data_catalog.sql in SSMS.
2. Execute the script against your local SQL Server instance.
3. After the result grid is populated:
    - Right-click the grid → "Copy with Headers"
    - Paste the results into a CSV file named **sql_server_entire_server_data_catalog.csv**
    - Open the CSV in a text editor and **replace all occurrences of the word NULL** (as a string) with a blank value.
    - Save the file as: **sql_server_entire_server_data_catalog.csv**

Step 3: Move Metadata to Neo4j Import Directory

1. Move or copy **sql_server_entire_server_data_catalog.csv** into the Neo4j database import directory.
    - Example path: C:/Databases/Neo4j/relate-data/dbmss/dbms-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/import/
    - (You can find the correct folder in Neo4j Desktop: open your database → click the three dots → "Open Folder" → navigate to import/.)

Step 4: Load Metadata into Neo4j

1. Make sure your Neo4j instance is running, and that it is using the **TimeMolecules** database.
2. Open Neo4j Browser (bolt://localhost:7687).
3. Open the file load_data_catalog_into_neo4j.cql in VS Code.
4. Copy and paste the contents into the Neo4j Browser.
5. Execute the code to load the data catalog into Neo4j.