# DoNeMP 2016Q12 – Module 1 – Home Task 1

Contents

[DoNeMP 2016Q12 – Module 1 – Home Task 1 1](#_Toc447922526)

[General Info 1](#_Toc447922527)

[Domain Model 1](#_Toc447922528)

[Git Repository 2](#_Toc447922529)

[Task 1 2](#_Toc447922530)

[**Task 2** 3](#_Toc447922531)

[Task 3 3](#_Toc447922532)

[SQL Name Convention 3](#_Toc447922533)

[Criteria for assessment: 4](#_Toc447922534)

[ER model 5](#_Toc447922535)

## General Info

You need to create a database for American domestic shipping company which has the fleet of trucks and serves several shipments between largest cities in USA.

Create the SQL script which creates database tables, keys and constraint according to the domain model, described below.

Please use the SQL Naming Conventions listed below.

Once database is created, please populate it with a test data located in spreadsheets in the .xlsx file.

## Domain Model

The Company XYZ provides a service of **shipment** of **cargo** by **trucks** between its **warehouses** located in the cities in the USA.

**Customer** can bring their **cargo** to one of the company’s warehouses and chose the destination of the **cargo** from the existing list of **warehouses**. Company’s shipment prices are depends on the distance between origin and destination warehouses, the **weight** (in kilograms) and the **volume** (m3) of the **cargo**.

Each customer leaves its own **contact** **information**, like Last name, First name and contact cell phone. He also registers the contact information of the recipient.

Company has fleet of **trucks** which serves different **routes** between **warehouses**.

Each truck has the **brand name**, registration plate number, **payload** in kilograms, **volume** of cargo in m3 and **fuel consumption** in liters per 100 kilometers.

Route includes the origin **warehouse** and destination **warehouse**. Each route has an estimated **distance**.

Every day several **shipments** are performed for each **route**. Operator selects a **truck** and a **driver** to serve the **shipment**.

Each **truck** has one or several **drivers** who can drive it. When operator starts the **shipment**, he picks the **truck** and then the **driver** from the list of **drivers** who assigned to the **truck**.



## Git Repository

* Create new folder **Module1\task1** under your git repository and place all files for this home task in this folder.
* Create file Readme.md in this folder and keep all your communication with mentor in this file (for example review comments, implementation details, etc.).
* All results must be created in the shared SQL Server in your personal database. Connection details will be provided in separate email.

## Task 1

* Using the domain model description and ER model, design and implement the MS SQL Server database.
* It must be a single SQL file will all database objects and constraints.
* The script must be created in the MS SQL Management Studio 2012/2014
* Create ER diagram from your physical model in the database. Upload it to the git and put the link into the readme.md file.

## Task 2

* Import sample data from the attached Excel files into the created database using any option. You can use:
  + Copy & paste, replace rows with regular expression in SSMS window to produce the INSERT SQL which is placed in another ImportData.sql file
  + Save result as separate csv file and import using BULK INSERT, c# or Powershell command.
  + You own approach
* Populate Route table with all possible pairs of warehouses, excluding the case when origin and destination are the warehouse.
* Set Distance for route as Random between 100 and 3000.
* Insert 10000 random values into Cargo table. You can use the same sender and recipient contacts.
* Populate Shipment table with 1000 records using random Truck Id, and Route Id. Randomly choose valid combinations of Truck and Driver for each shipment.
* Describe the option you have used in readme.md.
* Make sure all data presents on the shared SQL server.

## Task 3

* Create a view which returns following information about all shipment and includes:
  + Route origin city
  + Route destination city
  + Truck brand
  + Date and Time shipment has started
  + Date and Time shipment has been completed
  + Total weight of all shipped cargo
  + Total volume of all shipped cargo
  + Fuel spend on the route as calculated field using the formula (distance \* truck’s fuel consumption / 100).
* Add appropriate indexes into the tables involved into this view to increase query performance
* Use at least three approaches including CTE and CROSS APPLY to implement this query
* Put all three views in one SQL file named ShipmentView.sql and using the SQL comments mark the one query that has the most optimal Execution Plan.

Good luck and enjoy!

## SQL Name Convention

* Keep all Transact-SQL keywords UPPERCASE
* Structure the code using intends for easy reading
* Table names are not pluralized ("User" table not "Users")
* No underscores in object names (table, view, SP)
* Use PascalCase for Names
* Use “v” prefix in View names, like vOrder
* Use prefix in lowcase for all constraints, like pk\_Order.
* For foreign keys use fk\_TableName\_ReferenceTableName

## Criteria for assessment:

Max number of points: max **10 points**

Database Schema Design (Task 1):

1. Correct Entities and Relationships: 0 - 3 points
2. Correct data types: 0 – 1 point
3. Correct column names and following name convention: 0 - 1 point

Import sample data (Task 2):

1. Successfully populated tables with test data: 0 - 2 points.

Create View and SQL (Task 3)

1. View returns required data: 0 – 1 point
2. Used CTE (Common table expression): 0 – 1 point
3. Used CROSS APPLY: 0 – 1 point

## ER model

