School of Tech

Graduate Diploma in Data Analytics (Level 7)

Cover Sheet and Student Declaration

This sheet must be signed by the student and attached to the submitted assessment.

Course Title:	**Advanced Data Engineering**	**Course code:**	GDDA707
Student Name:	Mira Torririt	**Student ID:**	764707793
Assessment No & Type:	Assessment 1[Project]	**Cohort:**	GDDA7123C
Due Date:	26/02/24	**Date Submitted:**	26/02/24
Tutor's Name:	Mohammad Norouzifard		
Assessment Weighting	40%		
Total Marks	100		

Student Declaration:

I declare that:

- I have read the New Zealand School of Education Ltd policies and regulations on assessments and understand what plagiarism is.
- I am aware of the penalties for cheating and plagiarism as laid down by the New Zealand School of Education Ltd.
- This is an original assessment and is entirely my own work.
- Where I have quoted or made use of the ideas of other writers, I have acknowledged the source.
- This assessment has been prepared exclusively for this course and has not been or will not be submitted as assessed work in any other course.
- It has been explained to me that this assessment may be used by NZSE Ltd, for internal and/or external moderation.

Student signature: [illegible]

Date:

Tutor only to complete			
Assessment results:	**Task 1** (max. 50 marks)	**Task 2** (max. 45 marks)	**Documentation** (max. 5 marks)
	Total Marks: /100	**Grade:**	

Assessment 1: GDDA707 - Advanced Data Engineering

Project 1: Relational and Non-Relational Data Models

Mira Torririt

GDDA 707

Lecturer: Mohammad Norouzifard

School of Technology

Graduate Diploma in Data Analytics (Level 7)

February 26, 2024

Table of Contents

Chapter 1: Introduction

Data is a new gold. In today's business world, you can use it to produce a marketing and strategic plan and approach. Make a new product or service out of it. Improve people's lives through innovation. Make informed decisions. Prolongs people's lives and a lot more. Data is knowledge, and knowledge is power. A powerful tool that is crucial in our everyday living – a precious piece or collection of information. In this technological era, data are everywhere. And to make it more accessible, data is readily available online.

With such importance, how do we manage and value data?

A database is a collection of data. We store data in the database, whether its in-memory or cloud, and process it for future use in various situations. Although we can make a web app or any app without a database, it is crucial for a dynamic website.

The database has four components, as stated by Simplilearn, (2023):

- Data – the information stored in the database.
- Hardware – the valuable device for entering and storing the data.
- Software – is the program that connects the user to the database to access and manipulate the data.
- Users - are responsible for performing the function in the database.

Depending on the organization's requirement, there are two databases –relational and non-relational. In Pawlan's, (n.d.) blog, a relational database is structured. The data is organized into tables, rows, and columns and can be manipulated using Structured Query Language (SQL). Since it is in tabular form, this storage is considered traditional. Most of the time, the tables or entities have relationships or dependencies with one another. Examples of the SQL databases are Oracle, Microsoft SQL Server, Postgre SQL, MySQL, and MariaDB. A non-relational database, on the other hand, also known as a No Structured Query Language (NoSQL) database, does not have a tabular form. It includes various data models such as key (like an index)-and value (data storage), document (single record storage within one document), column family (the data is stored in columns), and graph (use nodes and edges to represent data relationships). This kind of database is unstructured. Examples of NoSQL are MongoDB, Google Cloud Firestore, Cassandra, Redis, Apache Hbase, and Amazon DynamoDB. In terms of usage, relational databases are suited for applications that require complex queries, transactions, and ample storage, such as banking, inventory management, and customer relationship management systems. On the other hand, the non-relational are great for applications with flexibility, scalability, and complex data structures, such as gaming applications, social media platforms, and search engines.

Given the differences between the two, which is right for you?

According to DatabaseTown, (n.d.): online, the factors to consider in which one to use are: 1) the type of data, which is the most crucial factor; 2) performance, in which the relational database can take more complex and longer query while non-relational for simple query; and 3) scalability, in which relational databases are more straightforward to scale. It is suggested to consider the databases` characteristics and limitations and evaluate your company's requirements or needs before deciding which database to use.

Chapter 2: Project Specification

The objective of this project is to utilize data modelling techniques that will fulfill a specific business requirement using the chosen scenario. The core value is to create relational and non-relational models and implement databases for efficient data manipulation and storage.

2.1) Scenario

Most businesses engage in Supply Chain Logistics to ensure the smooth handling of order fulfillment from manufacturing to product delivery. With the primary goal to meet customers' needs and expectations, it guarantees the continuity of the organization. Customer satisfaction establishes a good relationship with the business; thus, continuity and consistency of fulfillment are required.

* Please click the link to view the web page:

https://brunel.figshare.com/articles/dataset/Supply_Chain_Logistics_Problem_Dataset/7558679

The raw data is downloaded from the website in Excel format. It has seven entities, namely:

2.1.1) Order list – orders need a route assignment (attributes: order ID, order date, origin port, carrier, TPT (transportation day count), service level, ship ahead day count, ship late day count, customer, product ID, plant code, destination port, unit quantity, and weight)

2.1.2) Freight rates – list of carriers or couriers, the weight requirements for each individual lane and its rates. (attributes: carrier, origin port, destination port, minimum weight quantity, maximum weight quantity, service level, minimum cost, rate, mode description, TPT, carrier type)

2.1.3) Warehouse costs – costs associated with storage (attributes: warehouse and cost per unit)

2.1.4) Warehouse capacities – describes the warehouse capacities on daily basis (attributes: plant ID and daily capacity)

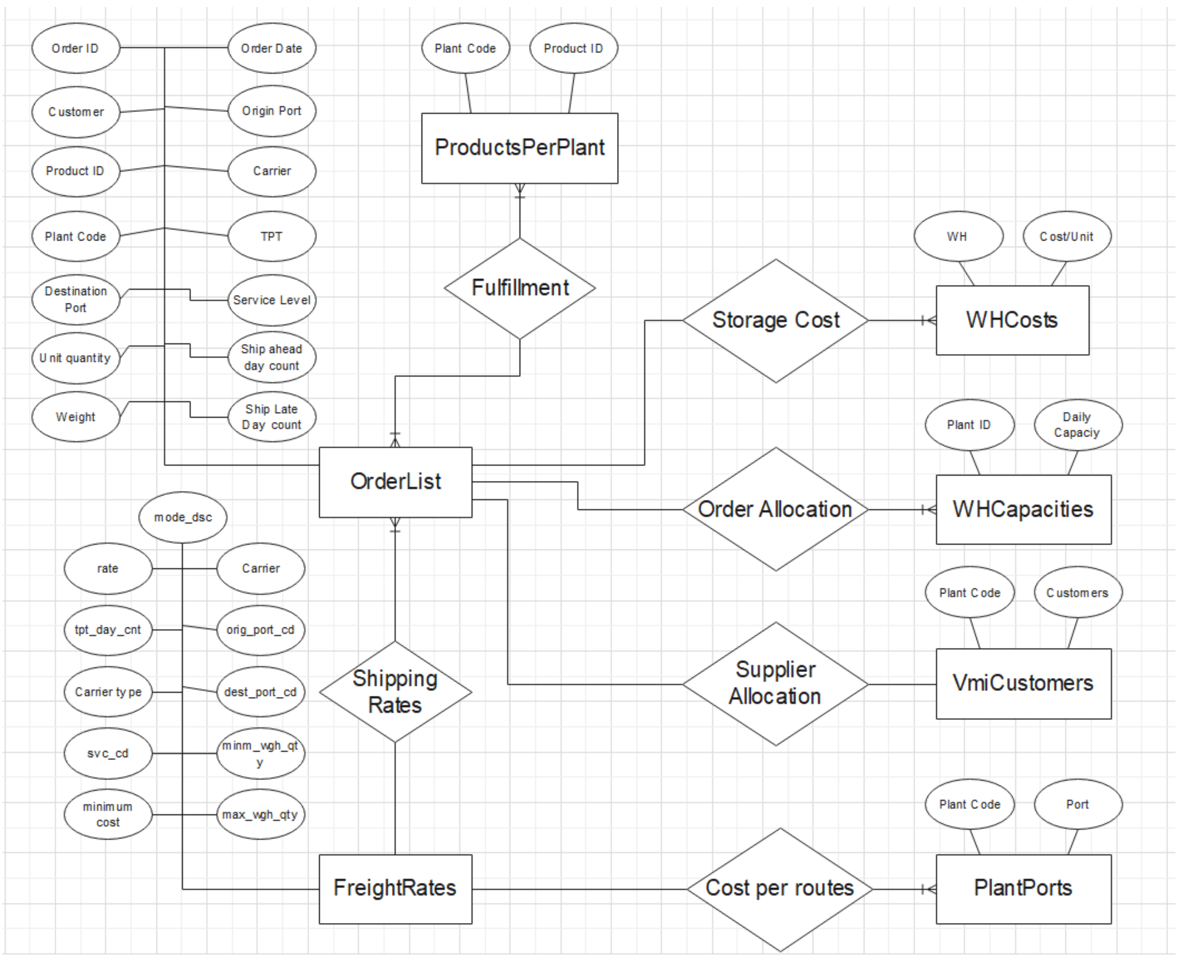
2.1.5) Products per plant – indicates the products supported by specific warehouse (attributes: plant code and product ID)

2.1.6) VMI (vendor-managed inventory) customers – a unique facility where a warehouse supports specific customers (attributes: plant code and customers)

2.1.7) Plant ports – refers to the warehousing and its corresponding shipping ports (attributes: plant code and port)

Chapter 3: Task 1 – Data Modelling

a. Relational: Entity Relationship Diagram for Supply Chain Logistics



b. Documentation of Relationships

1. **Order list (one) to products per plant (many) -** the order list includes information about the orders placed in the supply chain system and the associated logistics, service level, product, and customer details. The products per plant table refers to the specific product, assigned with a specific product ID, manufactured, or produced in a specific plant. The two entities have one to many relationships, where the order list can have one or more products per plant. The product ID links the relationships.

2. **Order list (many) to freight rates (one)** – orders can be associated with specific freight rates based on chosen shipping route and carrier.

3. **Order list (one) to warehouse costs (many)** – one order can be associated to multiple warehouses that incurred different costs.

4. **Order list (one) to warehouse capacities (many)** – one order can be associated with different warehouses based on capacities.

5. **Order list (one) to VMI customers (one)** – each customer is associated with a single order

6. **Freight rates (one) to plant ports (many) –** one freight rate can be associated with multiple warehouse ports.

c. Identifying Anomalies

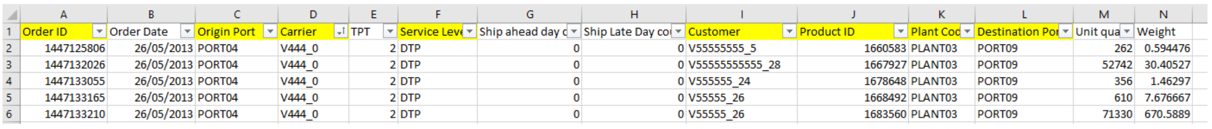
Note: Normalization is done manually in Excel

1. **Order list**
    a. Insert Anomaly occurs when an order lacks essential information (such as Carriers, Origin and Destination Ports), preventing its insertion into the table.
    b. Update Anomaly arises when modifications are made to an order's shipping details (e.g., Carrier), requiring updates to all instances of that carrier and introducing the potential for errors.
    c. Delete Anomaly involves the inadvertent removal of related information (e.g., Carrier, Customer, Product ID) when deleting an order.
2. **Freight rates**
    a. Redundancy occurs when a single freight rate is stored for multiple routes or carriers, resulting in data redundancy.
    b. Update anomaly arises when altering a freight rate for one route has unintended effects on other routes.
3. **Warehouse costs**
    a. Redundancy arises when identical cost information is stored for the same warehouse.
    b. Update anomaly occurs when modifying costs for one warehouse has no intended impact on other warehouses.
4. **Warehouse capacities**
    a. Redundancy is when the same data with identical capacity is stored in a single warehouse.
    b. Insert anomaly is when attempting to insert a warehouse into the table without capacity data causes an insertion anomaly.
5. **Products per plant**
    a. Redundancy occurs when the exact product information is saved for multiple plants.
    b. Updating anomaly is when modifying a product for one plan should not affect others.
6. **VMI customers**
    a. Redundancy occurs when the same customer information is stored across various vendor-managed inventory (VMI) setups.
    b. Inserting anomaly occurs when trying to add a customer to the table; an error occurs if there is no associated VMI agreement.
7. **Plant ports**
    a. Redundancy is when the same port information is stored more than once for a single plant.
    b. Update anomaly occurs when modifying port details for one plant affects others.

Addressing Anomalies Using Normalization

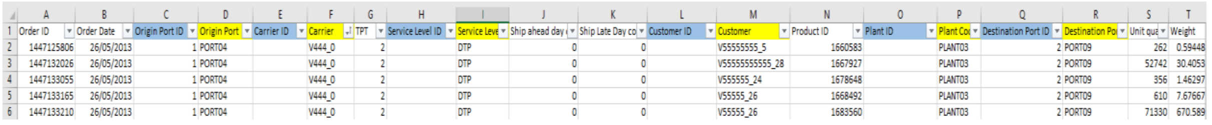
1. **First Normal Form (1NF) –** not applicable, as the obtained data is already in first normal form.
2. **Second Normal Form (2NF) -** in the Order List table, identify the independent variables and create its corresponding column which will become foreign keys in order list table. The independent variables found are:
   - Carriers to Carrier ID
   - Origin Port to Origin Port ID
   - Destination Port to Destination Port ID
   - Service Level to Service Level ID
   - Customers to Customers ID
   - Plant Code to Plants ID

Before (independent variables are highlighted in yellow)



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Order ID	Order Date	Origin Port	Carrier	TPT	Service Leve	Ship ahead day c	Ship Late Day co	Customer	Product ID	Plant Coc	Destination Por	Unit qua	Weight
2	1447125806	26/05/2013	PORT04	V444_0	2	DTP	0	0	V55555555_5	1660583	PLANT03	PORT09	262	0.594476
3	1447132026	26/05/2013	PORT04	V444_0	2	DTP	0	0	V55555555555_28	1667927	PLANT03	PORT09	52742	30.40527
4	1447133055	26/05/2013	PORT04	V444_0	2	DTP	0	0	V555555_24	1678648	PLANT03	PORT09	356	1.46297
5	1447133165	26/05/2013	PORT04	V444_0	2	DTP	0	0	V55555_26	1668492	PLANT03	PORT09	610	7.676667
6	1447133210	26/05/2013	PORT04	V444_0	2	DTP	0	0	V55555_26	1683560	PLANT03	PORT09	71330	670.5889

After (the ones highlighted in blue will replace those highlighted in yellow)

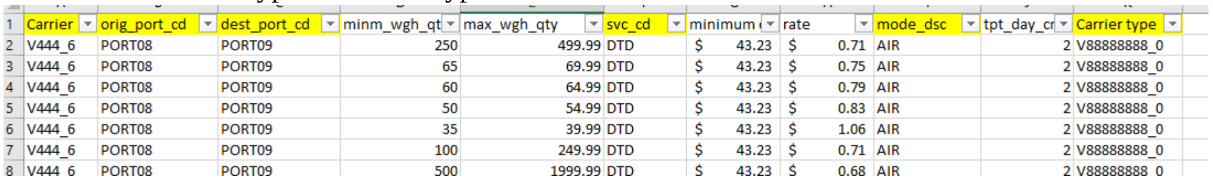


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Order ID	Order Date	Origin Port ID	Origin Port	Carrier ID	Carrier	TPT	Service Level ID	Service Leve	Ship ahead day	Ship Late Day co	Customer ID	Customer	Product ID	Plant ID	Plant Co	Destination Port ID	Destination Po	Unit qua	Weight
2	1447125806	26/05/2013	1	PORT04		V444_0	2		DTP	0	0		V55555555_5	1660583		PLANT03	2	PORT09	262	0.59448
3	1447132026	26/05/2013	1	PORT04		V444_0	2		DTP	0	0		V55555555555_28	1667927		PLANT03	2	PORT09	52742	30.4053
4	1447133055	26/05/2013	1	PORT04		V444_0	2		DTP	0	0		V555555_24	1678648		PLANT03	2	PORT09	356	1.46297
5	1447133165	26/05/2013	1	PORT04		V444_0	2		DTP	0	0		V55555_26	1668492		PLANT03	2	PORT09	610	7.67667
6	1447133210	26/05/2013	1	PORT04		V444_0	2		DTP	0	0		V55555_26	1683560		PLANT03	2	PORT09	71330	670.589

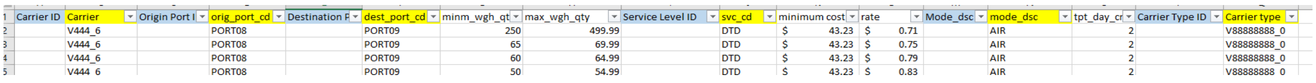
The same procedure is done in the freight rates, warehouse costs, capacities, products per plant, VMI customers, and plant port tables to prevent updating and insertion anomalies.

Freight Rates:

- Mode Description (mode_dsc) to Mode ID
- Carrier Type to Carrier Type ID



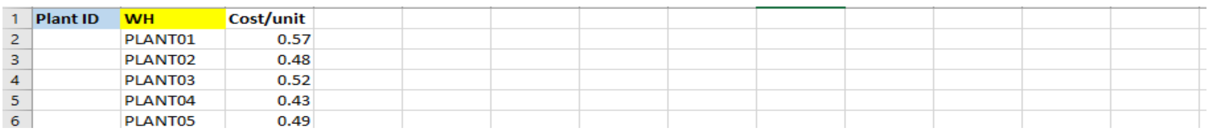
	Carrier	orig_port_cd	dest_port_cd	minm_wgh_qt	max_wgh_qty	svc_cd	minimum (	rate	mode_dsc	tpt_day_cr	Carrier type
2	V444_6	PORT08	PORT09	250	499.99	DTD	$ 43.23	$ 0.71	AIR	2	V88888888_0
3	V444_6	PORT08	PORT09	65	69.99	DTD	$ 43.23	$ 0.75	AIR	2	V88888888_0
4	V444_6	PORT08	PORT09	60	64.99	DTD	$ 43.23	$ 0.79	AIR	2	V88888888_0
5	V444_6	PORT08	PORT09	50	54.99	DTD	$ 43.23	$ 0.83	AIR	2	V88888888_0
6	V444_6	PORT08	PORT09	35	39.99	DTD	$ 43.23	$ 1.06	AIR	2	V88888888_0
7	V444_6	PORT08	PORT09	100	249.99	DTD	$ 43.23	$ 0.71	AIR	2	V88888888_0
8	V444_6	PORT08	PORT09	500	1999.99	DTD	$ 43.23	$ 0.68	AIR	2	V88888888_0



	Carrier ID	Carrier	Origin Port I	orig_port_cd	Destination P	dest_port_cd	minm_wgh_qt	max_wgh_qty	Service Level ID	svc_cd	minimum cost	rate	Mode_dsc	mode_dsc	tpt_day_cr	Carrier Type ID	Carrier type
2		V444_6		PORT08		PORT09	250	499.99		DTD	$ 43.23	$ 0.71		AIR	2		V88888888_0
3		V444_6		PORT08		PORT09	65	69.99		DTD	$ 43.23	$ 0.75		AIR	2		V88888888_0
4		V444_6		PORT08		PORT09	60	64.99		DTD	$ 43.23	$ 0.79		AIR	2		V88888888_0
5		V444_6		PORT08		PORT09	50	54.99		DTD	$ 43.23	$ 0.83		AIR	2		V88888888_0

Warehouse Costs:

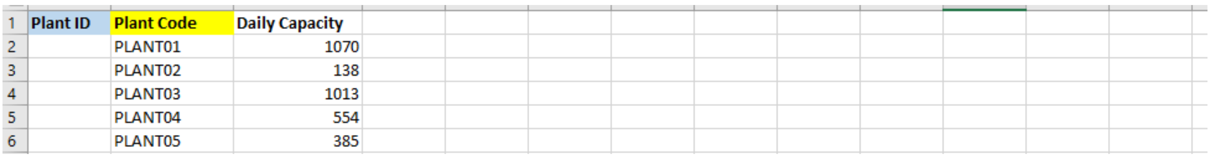
- Warehouse to Plant ID



	Plant ID	WH	Cost/unit
2		PLANT01	0.57
3		PLANT02	0.48
4		PLANT03	0.52
5		PLANT04	0.43
6		PLANT05	0.49

Warehouse Capacities:

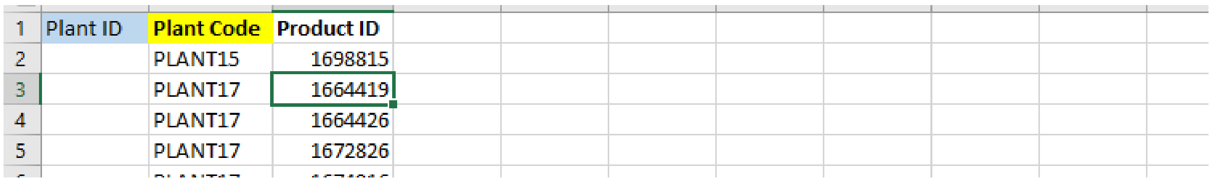
- Plant Code to Plant ID



	Plant ID	Plant Code	Daily Capacity
1	Plant ID	Plant Code	Daily Capacity
2		PLANT01	1070
3		PLANT02	138
4		PLANT03	1013
5		PLANT04	554
6		PLANT05	385

Products per plant:

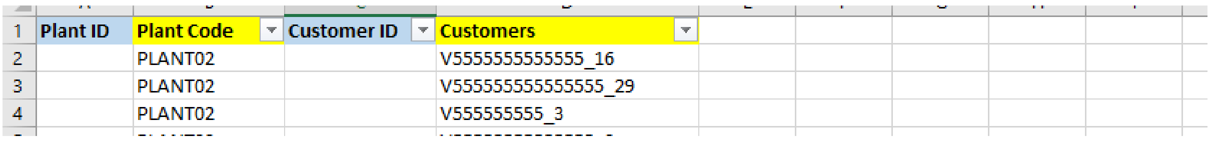
- Plant Code to Plant ID



	Plant ID	Plant Code	Product ID
1	Plant ID	Plant Code	Product ID
2		PLANT15	1698815
3		PLANT17	1664419
4		PLANT17	1664426
5		PLANT17	1672826

VMI Customers:

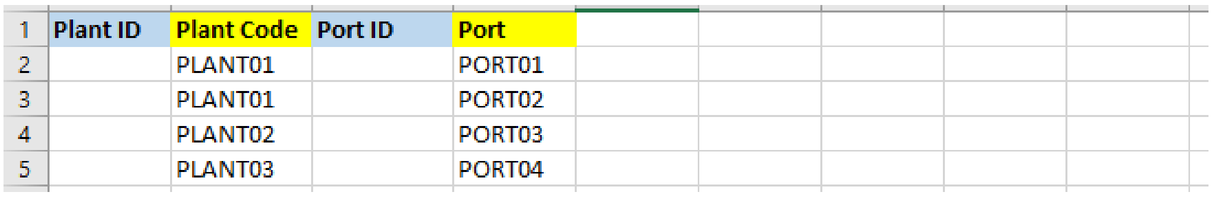
- Plant Code to Plant ID
- Customers to Customers ID



	Plant ID	Plant Code	Customer ID	Customers
1	Plant ID	Plant Code	Customer ID	Customers
2		PLANT02		V5555555555555_16
3		PLANT02		V555555555555555_29
4		PLANT02		V555555555_3

Plant Ports:

- Plant Code to Plant ID
- Port to Port ID



	Plant ID	Plant Code	Port ID	Port
1	Plant ID	Plant Code	Port ID	Port
2		PLANT01		PORT01
3		PLANT01		PORT02
4		PLANT02		PORT03
5		PLANT03		PORT04

3. **Third Normal Form (3NF) –** The order list table was divided into order and shipments tables, in which the latter was further decomposed creating a new table called Routes.

Orders Table:

- Order ID
- Order Date
- Customer ID
- Product ID
- Unity Quantity
- Weight

Shipments Table:

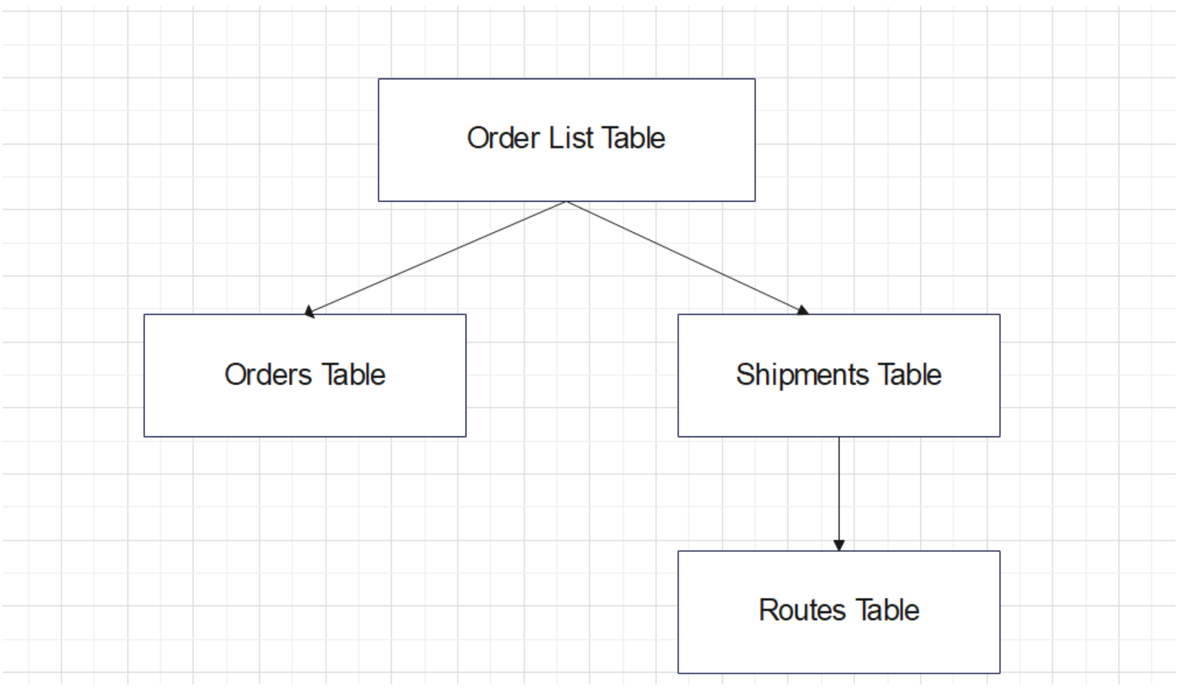
- Order ID
- Route ID
- Plant Code ID

- Service Level ID
- Ship Ahead Day Count
- Ship Late Day Count

Routes Table:

- Route ID
- Carrier ID
- Origin Port ID
- Destination Port ID

Illustration

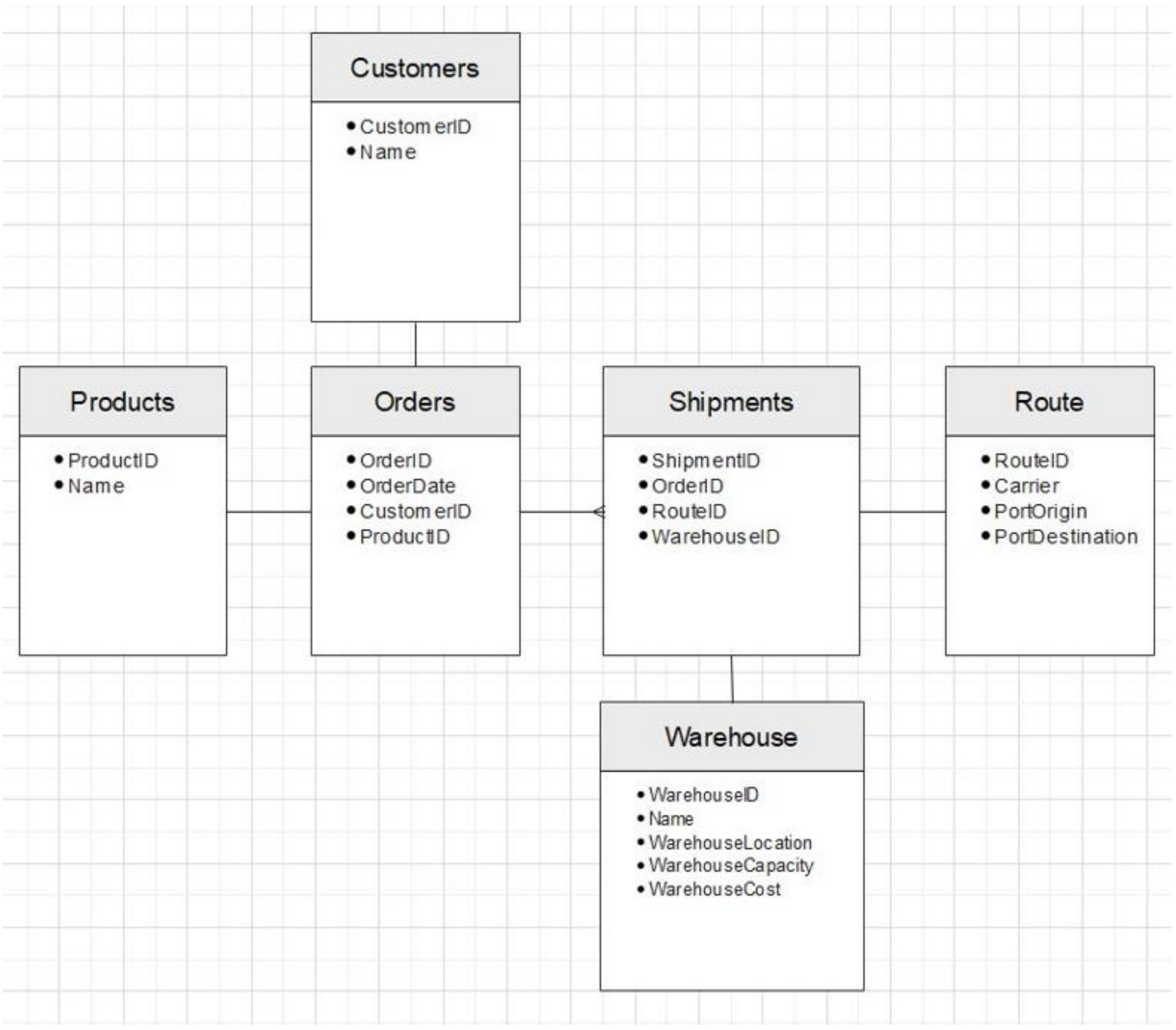


I replaced the Carrier ID, Origin_Port_ID, and Destination_Port_ID to optimize the Freight Rates table with the Route ID as the primary column.

Freight Rate Table:

- Route ID
- Service Level ID
- Mode ID
- Minimum Weight Quantity (min_wgh_qty)
- Maximum Weight Quantity (max_wgh_qty)
- Minimum Cost (minimum cost)
- Rate
- TPT (tpt_day_

d. Non- Relational Model for Supply Chain Logistics



e. Documentation

I extracted five variables using the same data and converted them to entities. These are the following:

- Customers – purchase and consume products (attributes: customer ID and name)
- Products – for this scenario, it is a tangible item sold by the company to meet customer needs (attributes: product ID and name)
- Orders – an act of purchasing made by the customer (attributes: order ID, order date, customer ID, product ID)
- Shipments – the collection of products transported from one location to another (attributes: shipment ID, order ID, route ID, warehouse ID)
- Route – a path of shipments (attributes: route ID, carrier, port origin, and port destination)

In a non-relational data model, the relationship can be established by embedding, denormalization, and referencing. It does not necessarily mean we cannot see any relationship. It is just being handled differently from the traditional way (relational), making it more flexible to use. In this scenario, the database store all the data that can stand alone.

Chapter 4: Task 2 – Database Implementation

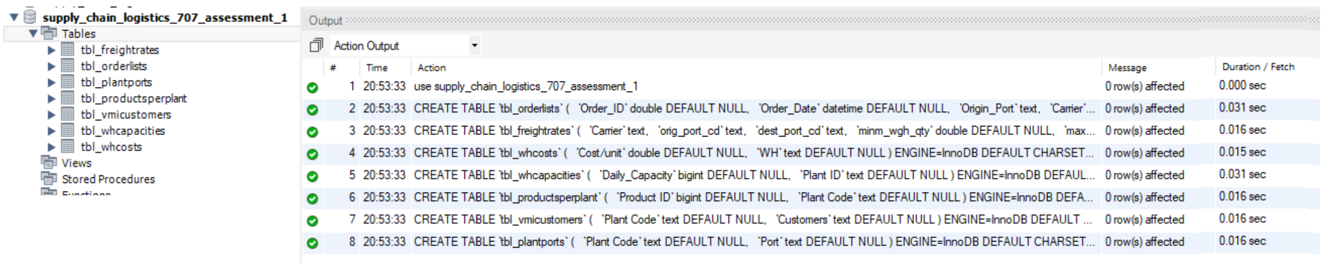
A. Relational – using MySQL database

a. Supply Chain Logistics Table

```
-- Command that selects the database to use for all Create statement
use supply_chain_logistics_707_assessment_1;

-- For Order list worksheet
CREATE TABLE `tbl_orderlists` (
  `Order_ID` BIGINT NOT NULL AUTO_INCREMENT,
  `Order_Date` datetime DEFAULT NULL,
  `Origin_Port` text,
  `Carrier` text,
  `TPT` bigint DEFAULT NULL,
  `Service_Level` text,
  `Ship_ahead_day_count` bigint DEFAULT NULL,
  `Ship_Late_Day_count` bigint DEFAULT NULL,
  `Customer` text,
  `Product_ID` bigint DEFAULT NULL,
  `Plant_Code` text,
  `Destination_Port` text,
  `Unit_quantity` bigint DEFAULT NULL,
  `Weight` double DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- For Freight Rates worksheet
CREATE TABLE `tbl_freightrates` (
  `Carrier` text,
  `orig_port_cd` text,
  `dest_port_cd` text,
  `minm_wgh_qty` double DEFAULT NULL,
  `max_wgh_qty` double DEFAULT NULL,
  `svc_cd` text,
  `minimum_cost` double DEFAULT NULL,
  `rate` double DEFAULT NULL,
  `mode_dsc` text,
  `tpt_day_cnt` bigint DEFAULT NULL,
  `Carrier_type` text
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- For Warehouse Costs worksheet
CREATE TABLE `tbl_whcosts` (
  `Cost/unit` double DEFAULT NULL,
  `WH` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- For Warehouse Capacities worksheet
CREATE TABLE `tbl_whcapacities` (
  `Daily_Capacity` bigint DEFAULT NULL,
  `Plant Code` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- For Products Per Plant worksheet
CREATE TABLE `tbl_productsperplant` (
  `Product_ID` bigint DEFAULT NULL,
  `Plant Code` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
-- For VMI Customers worksheet
CREATE TABLE `tbl_vmicustomers` (
  `Plant Code` text DEFAULT NULL,
  `Customers` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- For Plant Ports worksheet
CREATE TABLE `tbl_plantports` (
  `Plant Code` text DEFAULT NULL,
  `Port` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Output:



```
▼ supply_chain_logistics_707_assessment_1
  ▼ Tables
    ▶ tbl_freightrates
    ▶ tbl_orderlists
    ▶ tbl_plantports
    ▶ tbl_productsperplant
    ▶ tbl_vmicustomers
    ▶ tbl_whcapacities
    ▶ tbl_whcosts
    Views
    Stored Procedures
```

Output
Action Output

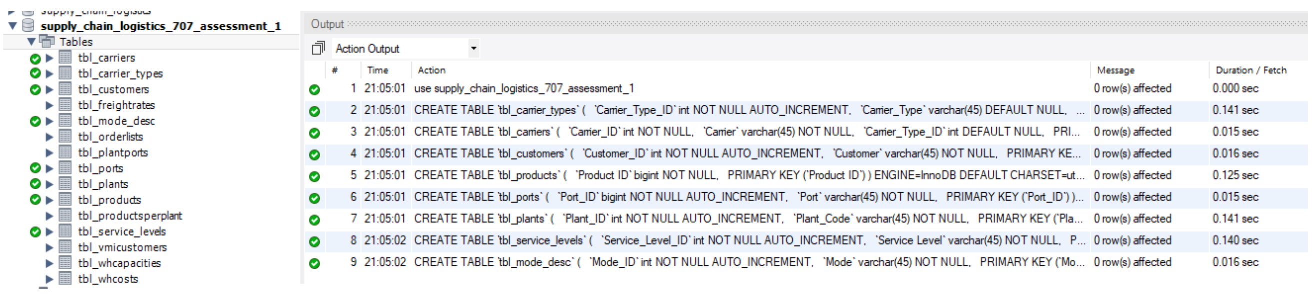
	#	Time	Action	Message	Duration / Fetch
✓	1	20:53:33	use supply_chain_logistics_707_assessment_1	0 row(s) affected	0.000 sec
✓	2	20:53:33	CREATE TABLE `tbl_orderlists` (`Order_ID` double DEFAULT NULL, `Order_Date` datetime DEFAULT NULL, `Origin_Port` text, `Carrier`...	0 row(s) affected	0.031 sec
✓	3	20:53:33	CREATE TABLE `tbl_freightrates` (`Carrier` text, `orig_port_cd` text, `dest_port_cd` text, `minm_wgh_qty` double DEFAULT NULL, `max...	0 row(s) affected	0.016 sec
✓	4	20:53:33	CREATE TABLE `tbl_whcosts` (`Cost/unit` double DEFAULT NULL, `WH` text DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET...	0 row(s) affected	0.015 sec
✓	5	20:53:33	CREATE TABLE `tbl_whcapacities` (`Daily_Capacity` bigint DEFAULT NULL, `Plant ID` text DEFAULT NULL) ENGINE=InnoDB DEFAUL...	0 row(s) affected	0.031 sec
✓	6	20:53:33	CREATE TABLE `tbl_productsperplant` (`Product ID` bigint DEFAULT NULL, `Plant Code` text DEFAULT NULL) ENGINE=InnoDB DEFA...	0 row(s) affected	0.016 sec
✓	7	20:53:33	CREATE TABLE `tbl_vmicustomers` (`Plant Code` text DEFAULT NULL, `Customers` text DEFAULT NULL) ENGINE=InnoDB DEFAULT ...	0 row(s) affected	0.016 sec
✓	8	20:53:33	CREATE TABLE `tbl_plantports` (`Plant Code` text DEFAULT NULL, `Port` text DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET...	0 row(s) affected	0.016 sec

Create Independent Variables extracted from Supply Chain Logistics datasets.

```
-- Command that selects the database to use for all Create statement
use supply_chain_logistics_707_db;

-- Table for the list of Carrier Types
CREATE TABLE `tbl_carrier_types` (
  `Carrier_Type_ID` int NOT NULL AUTO_INCREMENT,
  `Carrier_Type` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`Carrier_Type_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- Table for the list of Carriers
CREATE TABLE `tbl_carriers` (
  `Carrier_ID` int NOT NULL,
  `Carrier` varchar(45) NOT NULL,
  `Carrier_Type_ID` int DEFAULT NULL,
  PRIMARY KEY (`Carrier_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- Table for the list of Customers
CREATE TABLE `tbl_customers` (
  `Customer_ID` int NOT NULL AUTO_INCREMENT,
  `Customer` varchar(45) NOT NULL,
  PRIMARY KEY (`Customer_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- Table for the list of Products
CREATE TABLE `tbl_products` (
  `Product_ID` bigint NOT NULL AUTO_INCREMENT,
  `Product` varchar(45) NOT NULL,
  PRIMARY KEY (`Product ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
-- Table for the list of Ports
CREATE TABLE `tbl_ports` (
  `Port_ID` int NOT NULL AUTO_INCREMENT,
  `Port` varchar(45) NOT NULL,
  PRIMARY KEY (`Port_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- Table for the list of Plants (Warehouse)
CREATE TABLE `tbl_plants` (
  `Plant_ID` int NOT NULL AUTO_INCREMENT,
  `Plant_Code` varchar(45) NOT NULL,
  PRIMARY KEY (`Plant_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- Table for the list of Service Levels
CREATE TABLE `tbl_service_levels` (
  `Service_Level_ID` int NOT NULL AUTO_INCREMENT,
  `Service Level` varchar(45) NOT NULL,
  PRIMARY KEY (`Service_Level_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- Table for the list of Modes
CREATE TABLE `tbl_mode_desc` (
  `Mode_ID` int NOT NULL AUTO_INCREMENT,
  `Mode` varchar(45) NOT NULL,
  PRIMARY KEY (`Mode_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Output:



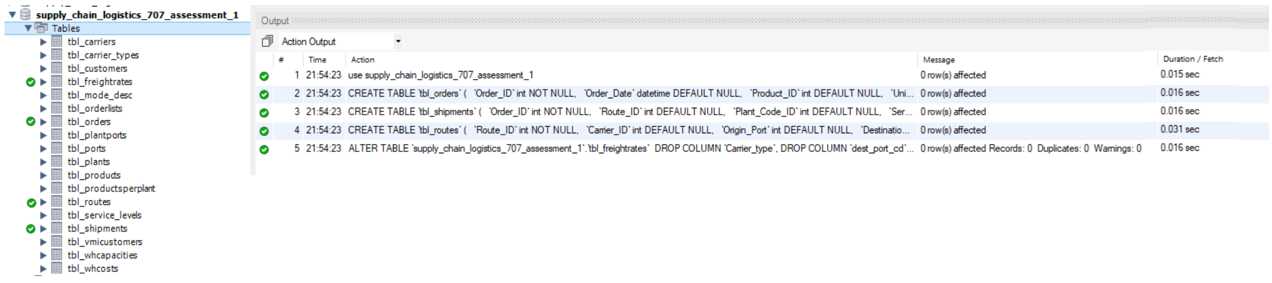
```
supply_chain_logistics_707_assessment_1
  Tables
    tbl_carriers
    tbl_carrier_types
    tbl_customers
    tbl_freightrates
    tbl_mode_desc
    tbl_orderlists
    tbl_plantports
    tbl_ports
    tbl_plants
    tbl_products
    tbl_productsperplant
    tbl_service_levels
    tbl_vmicustomers
    tbl_whcapacities
    tbl_whcosts
```

```
Output
Action Output
#  Time      Action                                                                                                                                Message            Duration / Fetch
1  21:05:01  use supply_chain_logistics_707_assessment_1                                                                                            0 row(s) affected  0.000 sec
2  21:05:01  CREATE TABLE `tbl_carrier_types` (  `Carrier_Type_ID` int NOT NULL AUTO_INCREMENT,  `Carrier_Type` varchar(45) DEFAULT NULL,  ...     0 row(s) affected  0.141 sec
3  21:05:01  CREATE TABLE `tbl_carriers` (  `Carrier_ID` int NOT NULL,  `Carrier` varchar(45) NOT NULL,  `Carrier_Type_ID` int DEFAULT NULL,  PRI...  0 row(s) affected  0.015 sec
4  21:05:01  CREATE TABLE `tbl_customers` (  `Customer_ID` int NOT NULL AUTO_INCREMENT,  `Customer` varchar(45) NOT NULL,  PRIMARY KE...            0 row(s) affected  0.016 sec
5  21:05:01  CREATE TABLE `tbl_products` (  `Product ID` bigint NOT NULL,  PRIMARY KEY (`Product ID`) ) ENGINE=InnoDB DEFAULT CHARSET=ut...        0 row(s) affected  0.125 sec
6  21:05:01  CREATE TABLE `tbl_ports` (  `Port_ID` bigint NOT NULL AUTO_INCREMENT,  `Port` varchar(45) NOT NULL,  PRIMARY KEY (`Port_ID`) )...     0 row(s) affected  0.015 sec
7  21:05:01  CREATE TABLE `tbl_plants` (  `Plant_ID` int NOT NULL AUTO_INCREMENT,  `Plant_Code` varchar(45) NOT NULL,  PRIMARY KEY (`Pla...         0 row(s) affected  0.141 sec
8  21:05:02  CREATE TABLE `tbl_service_levels` (  `Service_Level_ID` int NOT NULL AUTO_INCREMENT,  `Service Level` varchar(45) NOT NULL,  P...     0 row(s) affected  0.140 sec
9  21:05:02  CREATE TABLE `tbl_mode_desc` (  `Mode_ID` int NOT NULL AUTO_INCREMENT,  `Mode` varchar(45) NOT NULL,  PRIMARY KEY (`Mo...            0 row(s) affected  0.016 sec
```

Align with Normalization, Decompose tbl_orderlist and create three columns.

```
-- Command that selects the database to use for all Create statement
use supply_chain_logistics_db;

-- Extracted from tbl_orderlist.
-- Order table only needs the detail pertaining to orders like order id, date,
product ordered, quantity, weight and the customer who ordered
CREATE TABLE `tbl_orders` (
  `Order_ID` int NOT NULL AUTO_INCREMENT,
  `Order_Date` date DEFAULT NULL,
  `Product_ID` BIGINT DEFAULT NULL,
  `Unit_Quantity` decimal(10,0) DEFAULT NULL,
  `Weight` decimal(10,0) DEFAULT NULL,
  `Customer_ID` int DEFAULT NULL,
  PRIMARY KEY (`Order_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
-- Extracted from tbl_orderlist
-- Shipments table only needs to know which order to ship, which route to take,
what plant (warehouse) to use, the service level, the day counts
CREATE TABLE `tbl_shipments` (
 `Shipment_ID` int NOT NULL AUTO_INCREMENT,
  `Order_ID` int NOT NULL,
  `Route_ID` int DEFAULT NULL,
  `Plant_ID` int DEFAULT NULL,
  `Service_Level_ID` int DEFAULT NULL,
  `Ship_ahead_day_count` int DEFAULT NULL,
  `Ship_late_day_count` int DEFAULT NULL,
 PRIMARY KEY (`Shipment_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- Extracted from tbl_shipments
-- Extracted Carrier, Origin Port, and Destination Port as this causes
redundancy in the Shipment table. When an existing route changes, it should
affect the shipments related to it.
CREATE TABLE `tbl_routes` (
  `Route_ID` int NOT NULL AUTO_INCREMENT,
  `Carrier_ID` int DEFAULT NULL,
  `Origin_Port` int DEFAULT NULL,
  `Destination_Port` int DEFAULT NULL,
  PRIMARY KEY (`Route_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- Extracted from tbl_shipments
-- Extracted Carrier, Origin Port, and Destination Port as this causes
redundancy in the Shipment table. When an existing route changes, it should
affect the shipments related to it.
ALTER TABLE `tbl_freightrates`
DROP COLUMN `Carrier_type`,
DROP COLUMN `dest_port_cd`,
DROP COLUMN `orig_port_cd`,
DROP COLUMN `Carrier`,
CHANGE COLUMN `svc_cd` `svc_cd` INT NULL DEFAULT NULL ,
CHANGE COLUMN `mode_dsc` `mode_dsc` INT NULL DEFAULT NULL ,
ADD COLUMN `RouteID` INT NULL AFTER `tpt_day_cnt`;
```



```
▼ supply_chain_logistics_707_assessment_1
  ▼ Tables
      ▶ tbl_carriers
      ▶ tbl_carrier_types
      ▶ tbl_customers
    ✓ ▶ tbl_freightrates
      ▶ tbl_mode_desc
      ▶ tbl_orderlists
    ✓ ▶ tbl_orders
      ▶ tbl_plantports
      ▶ tbl_ports
      ▶ tbl_plants
      ▶ tbl_products
      ▶ tbl_productsperplant
    ✓ ▶ tbl_routes
      ▶ tbl_service_levels
    ✓ ▶ tbl_shipments
      ▶ tbl_vmicustomers
      ▶ tbl_whcapacities
      ▶ tbl_whcosts
```

Output

Action Output

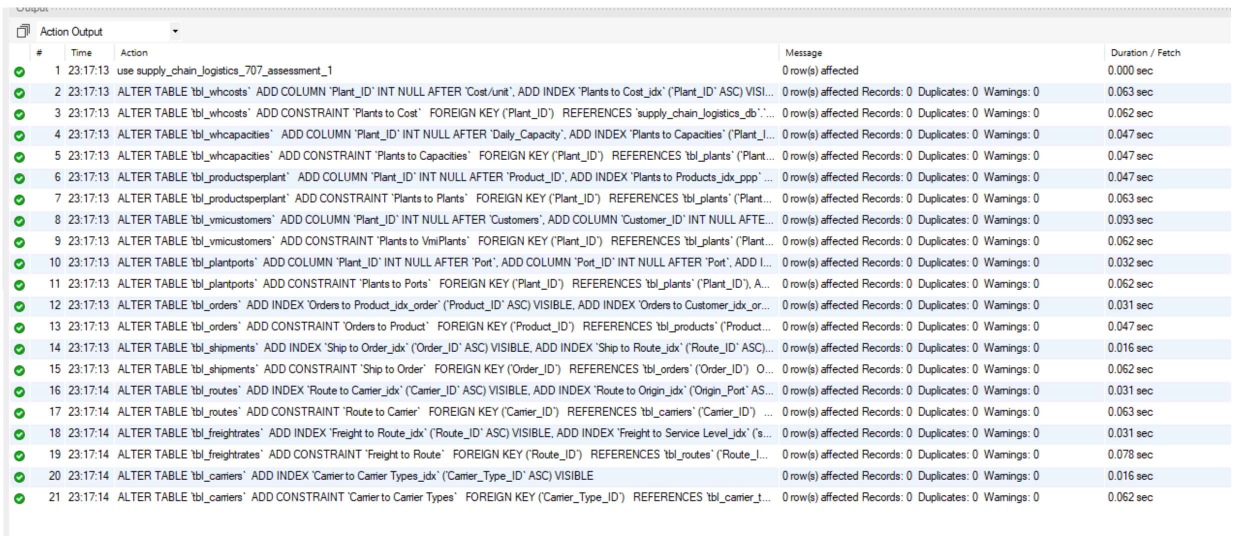
	#	Time	Action	Message	Duration / Fetch
✓	1	21:54:23	use supply_chain_logistics_707_assessment_1	0 row(s) affected	0.015 sec
✓	2	21:54:23	CREATE TABLE `tbl_orders` (`Order_ID` int NOT NULL, `Order_Date` datetime DEFAULT NULL, `Product_ID` int DEFAULT NULL, `Uni...	0 row(s) affected	0.016 sec
✓	3	21:54:23	CREATE TABLE `tbl_shipments` (`Order_ID` int NOT NULL, `Route_ID` int DEFAULT NULL, `Plant_Code_ID` int DEFAULT NULL, `Ser...	0 row(s) affected	0.016 sec
✓	4	21:54:23	CREATE TABLE `tbl_routes` (`Route_ID` int NOT NULL, `Carrier_ID` int DEFAULT NULL, `Origin_Port` int DEFAULT NULL, `Destinatio...	0 row(s) affected	0.031 sec
✓	5	21:54:23	ALTER TABLE `supply_chain_logistics_707_assessment_1`.`tbl_freightrates` DROP COLUMN `Carrier_type`, DROP COLUMN `dest_port_cd`...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec

Alter tables and add Constraints.

```
-- Command that selects the database to use for all Alter statement
use supply_chain_logistics_707_assessment_1;

ALTER TABLE `tbl_whcosts`
ADD COLUMN `Plant_ID` INT NULL AFTER `Cost/unit`,
ADD INDEX `Plants to Cost_idx` (`Plant_ID` ASC) VISIBLE;

ALTER TABLE `tbl_whcosts`
ADD CONSTRAINT `Plants to Cost`
```

```
  FOREIGN KEY (`Plant_ID`)

  REFERENCES `supply_chain_logistics_db`.`tbl_plants` (`Plant_ID`);

ALTER TABLE `tbl_whcosts`

DROP COLUMN `WH`;

ALTER TABLE `tbl_whcapacities`
ADD COLUMN `Plant_ID` INT NULL AFTER `Daily_Capacity`,
ADD INDEX `Plants to Capacities` (`Plant_ID` ASC) VISIBLE;

ALTER TABLE `tbl_whcapacities`
ADD CONSTRAINT `Plants to Capacities`
  FOREIGN KEY (`Plant_ID`)

  REFERENCES `tbl_plants` (`Plant_ID`);

ALTER TABLE `tbl_whcapacities`

DROP COLUMN `Plant Code`;


ALTER TABLE `tbl_productsperplant`
ADD COLUMN `Plant_ID` INT NULL AFTER `Product_ID`,
ADD INDEX `Plants to Products_idx_ppp` (`Plant_ID` ASC) VISIBLE,
ADD INDEX `Products to Products_idx_ppp` (`Product_ID` ASC) VISIBLE;

ALTER TABLE `tbl_productsperplant`
ADD CONSTRAINT `Plants to Plants`
  FOREIGN KEY (`Plant_ID`)
  REFERENCES `tbl_plants` (`Plant_ID`),
ADD CONSTRAINT `Products to Products`
  FOREIGN KEY (`Product_ID`)

  REFERENCES `tbl_products` (`Product_ID`);

ALTER TABLE `tbl_productsperplant`

DROP COLUMN `Plant Code`;

ALTER TABLE `tbl_vmicustomers`
ADD COLUMN `Plant_ID` INT NULL AFTER `Customers`,
ADD COLUMN `Customer_ID` INT NULL AFTER `Plant_ID`,
ADD INDEX `Plants to VmiPlants_idx_vmicustomers` (`Plant_ID` ASC) VISIBLE,
ADD INDEX `Customers to VmiCustomers_idx_vmicustomers` (`Customer_ID` ASC)
VISIBLE;

ALTER TABLE `tbl_vmicustomers`
ADD CONSTRAINT `Plants to VmiPlants`
  FOREIGN KEY (`Plant_ID`)
  REFERENCES `tbl_plants` (`Plant_ID`),
ADD CONSTRAINT `Customers to VmiCustomers`
  FOREIGN KEY (`Customer_ID`)

  REFERENCES `tbl_customers` (`Customer_ID`);

ALTER TABLE `tbl_vmicustomers`

DROP COLUMN `Customers`,

DROP COLUMN `Plant Code`;
```

```
ALTER TABLE `tbl_plantports`
ADD COLUMN `Plant_ID` INT NULL AFTER `Port`,
ADD COLUMN `Port_ID` INT NULL AFTER `Port`,
ADD INDEX `Plants to Ports_idx_pp` (`Plant_ID` ASC) VISIBLE;


ALTER TABLE `tbl_plantports`
ADD CONSTRAINT `Plants to Ports`
  FOREIGN KEY (`Plant_ID`)
  REFERENCES `tbl_plants` (`Plant_ID`),
ADD CONSTRAINT `Ports to Ports`
  FOREIGN KEY (`Port_ID`)

  REFERENCES `tbl_ports` (`Port_ID`);

ALTER TABLE `tbl_plantports`

DROP COLUMN `Port`,

DROP COLUMN `Plant Code`;

ALTER TABLE `tbl_orders`
ADD INDEX `Orders to Product_idx_order` (`Product_ID` ASC) VISIBLE,
ADD INDEX `Orders to Customer_idx_order` (`Customer_ID` ASC) VISIBLE;

ALTER TABLE `tbl_orders`
ADD CONSTRAINT `Orders to Product`
  FOREIGN KEY (`Product_ID`)
  REFERENCES `tbl_products` (`Product_ID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `Orders to Customer`
  FOREIGN KEY (`Customer_ID`)
  REFERENCES `tbl_customers` (`Customer_ID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION;

ALTER TABLE `tbl_shipments`
ADD INDEX `Ship to Order_idx` (`Order_ID` ASC) VISIBLE,
ADD INDEX `Ship to Route_idx` (`Route_ID` ASC) VISIBLE,
ADD INDEX `Ship to Plant_idx` (`Plant_ID` ASC) VISIBLE,
ADD INDEX `Ship to Service Level_idx` (`Service_Level_ID` ASC) VISIBLE;

ALTER TABLE `tbl_shipments`
ADD CONSTRAINT `Ship to Order`
  FOREIGN KEY (`Order_ID`)
  REFERENCES `tbl_orders` (`Order_ID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `Ship to Route`
  FOREIGN KEY (`Route_ID`)
  REFERENCES `tbl_routes` (`Route_ID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `Ship to Service Level`
  FOREIGN KEY (`Service_Level_ID`)
  REFERENCES `tbl_service_levels` (`Service_Level_ID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `Ship to Plant`
  FOREIGN KEY (`Plant_ID`)
  REFERENCES `tbl_plants` (`Plant_ID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION;


ALTER TABLE `tbl_routes`
ADD INDEX `Route to Carrier_idx` (`Carrier_ID` ASC) VISIBLE,
ADD INDEX `Route to Origin_idx` (`Origin_Port` ASC) VISIBLE,
ADD INDEX `Route to Destination_idx` (`Destination_Port` ASC) VISIBLE;

ALTER TABLE `tbl_routes`
ADD CONSTRAINT `Route to Carrier`
  FOREIGN KEY (`Carrier_ID`)
  REFERENCES `tbl_carriers` (`Carrier_ID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `Route to Origin`
  FOREIGN KEY (`Origin_Port`)
  REFERENCES `tbl_ports` (`Port_ID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `Route to Destination`
  FOREIGN KEY (`Destination_Port`)
  REFERENCES `tbl_ports` (`Port_ID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION;

ALTER TABLE `tbl_freightrates`
ADD INDEX `Freight to Route_idx` (`Route_ID` ASC) VISIBLE,
ADD INDEX `Freight to Service Level_idx` (`svc_cd` ASC) VISIBLE,
ADD INDEX `Freight to Mode_idx` (`mode_dsc` ASC) VISIBLE;

ALTER TABLE `tbl_freightrates`
ADD CONSTRAINT `Freight to Route`
  FOREIGN KEY (`Route_ID`)
  REFERENCES `tbl_routes` (`Route_ID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `Freight to Service Level`
  FOREIGN KEY (`svc_cd`)
  REFERENCES `tbl_service_levels` (`Service_Level_ID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `Freight to Mode`
  FOREIGN KEY (`mode_dsc`)
  REFERENCES `tbl_mode_desc` (`Mode_ID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION;

ALTER TABLE `tbl_carriers`
ADD INDEX `Carrier to Carrier Types_idx` (`Carrier_Type_ID` ASC) VISIBLE;

ALTER TABLE `tbl_carriers`
ADD CONSTRAINT `Carrier to Carrier Types`
  FOREIGN KEY (`Carrier_Type_ID`)
  REFERENCES `tbl_carrier_types` (`Carrier_Type_ID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION;
```



```
Action Output
#   Time      Action                                                                                                                                              Message                                                              Duration / Fetch
1   23:17:13  use supply_chain_logistics_707_assessment_1                                                                                                         0 row(s) affected                                                    0.000 sec
2   23:17:13  ALTER TABLE `tbl_whcosts` ADD COLUMN `Plant_ID` INT NULL AFTER `Cost/unit`, ADD INDEX `Plants to Cost_idx` (`Plant_ID` ASC) VISI...                 0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.063 sec
3   23:17:13  ALTER TABLE `tbl_whcosts` ADD CONSTRAINT `Plants to Cost`  FOREIGN KEY (`Plant_ID`)  REFERENCES `supply_chain_logistics_db`.`...                   0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.062 sec
4   23:17:13  ALTER TABLE `tbl_whcapacities`  ADD COLUMN `Plant_ID` INT NULL AFTER `Daily_Capacity`, ADD INDEX `Plants to Capacities` (`Plant_I...               0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.047 sec
5   23:17:13  ALTER TABLE `tbl_whcapacities` ADD CONSTRAINT `Plants to Capacities`  FOREIGN KEY (`Plant_ID`)  REFERENCES `tbl_plants` (`Plant...                  0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.047 sec
6   23:17:13  ALTER TABLE `tbl_productsperplant`  ADD COLUMN `Plant_ID` INT NULL AFTER `Product_ID`, ADD INDEX `Plants to Products_idx_ppp` ...                  0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.047 sec
7   23:17:13  ALTER TABLE `tbl_productsperplant` ADD CONSTRAINT `Plants to Plants`  FOREIGN KEY (`Plant_ID`)  REFERENCES `tbl_plants` (`Plant...                  0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.063 sec
8   23:17:13  ALTER TABLE `tbl_vmicustomers` ADD COLUMN `Plant_ID` INT NULL AFTER `Customers`, ADD COLUMN `Customer_ID` INT NULL AFTE...                          0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.093 sec
9   23:17:13  ALTER TABLE `tbl_vmicustomers` ADD CONSTRAINT `Plants to VmiPlants`  FOREIGN KEY (`Plant_ID`)  REFERENCES `tbl_plants` (`Plant...                   0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.062 sec
10  23:17:13  ALTER TABLE `tbl_plantports` ADD COLUMN `Plant_ID` INT NULL AFTER `Port`, ADD COLUMN `Port_ID` INT NULL AFTER `Port`, ADD I...                     0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.032 sec
11  23:17:13  ALTER TABLE `tbl_plantports` ADD CONSTRAINT `Plants to Ports`  FOREIGN KEY (`Plant_ID`)  REFERENCES `tbl_plants` (`Plant_ID`), A...                 0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.062 sec
12  23:17:13  ALTER TABLE `tbl_orders` ADD INDEX `Orders to Product_idx_order` (`Product_ID` ASC) VISIBLE, ADD INDEX `Orders to Customer_idx_or...                0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.031 sec
13  23:17:13  ALTER TABLE `tbl_orders` ADD CONSTRAINT `Orders to Product`  FOREIGN KEY (`Product_ID`)  REFERENCES `tbl_products` (`Product...                    0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.047 sec
14  23:17:13  ALTER TABLE `tbl_shipments` ADD INDEX `Ship to Order_idx` (`Order_ID` ASC) VISIBLE, ADD INDEX `Ship to Route_idx` (`Route_ID` ASC)...                0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.016 sec
15  23:17:13  ALTER TABLE `tbl_shipments` ADD CONSTRAINT `Ship to Order`  FOREIGN KEY (`Order_ID`)  REFERENCES `tbl_orders` (`Order_ID`)  O...                    0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.062 sec
16  23:17:14  ALTER TABLE `tbl_routes` ADD INDEX `Route to Carrier_idx` (`Carrier_ID` ASC) VISIBLE, ADD INDEX `Route to Origin_idx` (`Origin_Port` AS...           0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.031 sec
17  23:17:14  ALTER TABLE `tbl_routes` ADD CONSTRAINT `Route to Carrier`  FOREIGN KEY (`Carrier_ID`)  REFERENCES `tbl_carriers` (`Carrier_ID`)  ...              0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.063 sec
18  23:17:14  ALTER TABLE `tbl_freightrates` ADD INDEX `Freight to Route_idx` (`Route_ID` ASC) VISIBLE, ADD INDEX `Freight to Service Level_idx` (`s...          0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.031 sec
19  23:17:14  ALTER TABLE `tbl_freightrates` ADD CONSTRAINT `Freight to Route`  FOREIGN KEY (`Route_ID`)  REFERENCES `tbl_routes` (`Route_I...                   0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.078 sec
20  23:17:14  ALTER TABLE `tbl_carriers` ADD INDEX `Carrier to Carrier Types_idx` (`Carrier_Type_ID` ASC) VISIBLE                                                  0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.016 sec
21  23:17:14  ALTER TABLE `tbl_carriers` ADD CONSTRAINT `Carrier to Carrier Types`  FOREIGN KEY (`Carrier_Type_ID`)  REFERENCES `tbl_carrier_t...                 0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0             0.062 sec
```

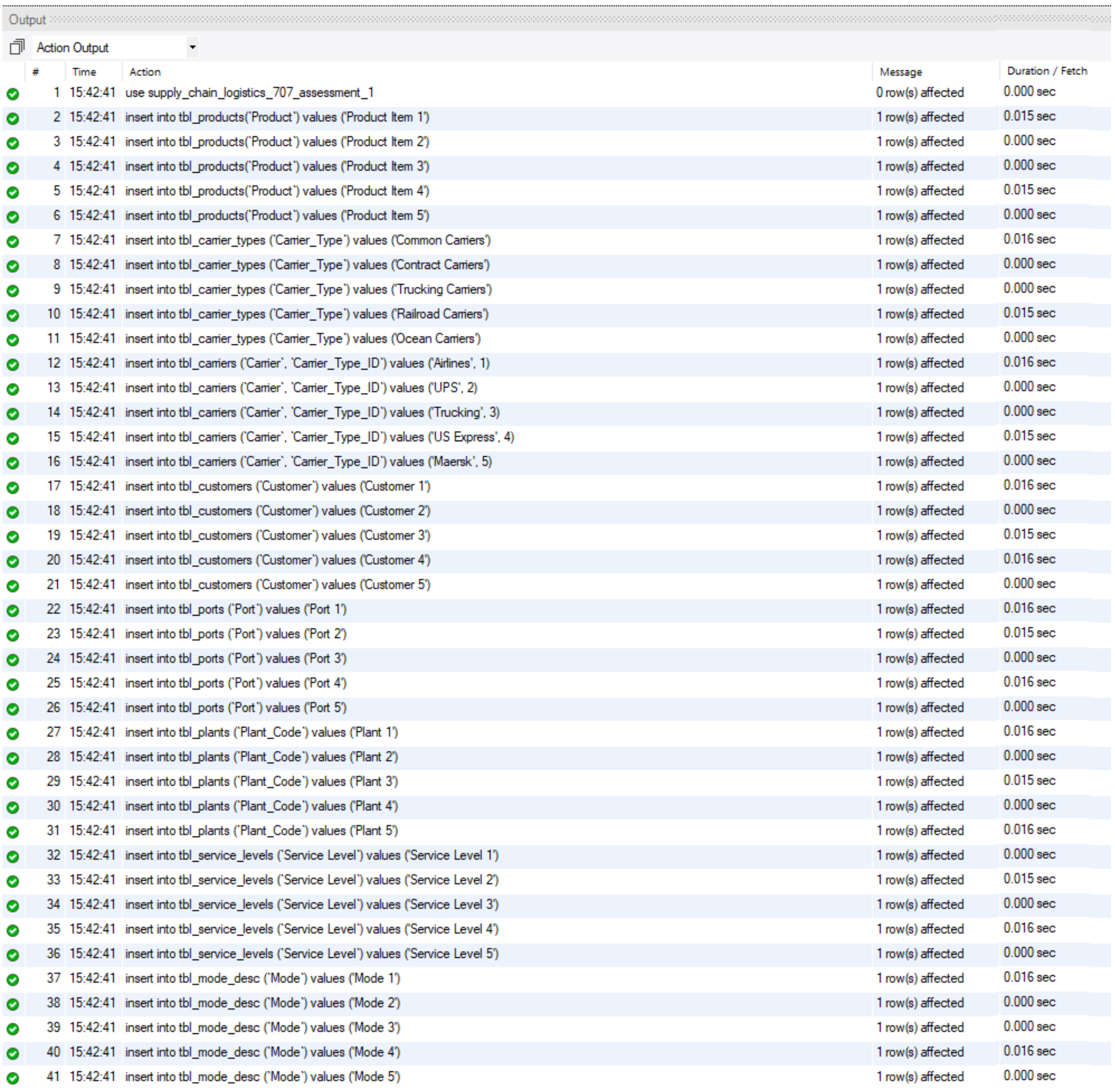
b. Inserting records

Inserting records into the independent tables

```sql
use supply_chain_logistics_707_assessment_1;

-- Inserting 5 products
insert into tbl_products(`Product`) values ('Product Item 1');
insert into tbl_products(`Product`) values ('Product Item 2');
insert into tbl_products(`Product`) values ('Product Item 3');
insert into tbl_products(`Product`) values ('Product Item 4');
insert into tbl_products(`Product`) values ('Product Item 5');

-- Inserting 5 Carrier Types
insert into tbl_carrier_types (`Carrier_Type`) values ('Common Carriers');
insert into tbl_carrier_types (`Carrier_Type`) values ('Contract Carriers');
insert into tbl_carrier_types (`Carrier_Type`) values ('Trucking Carriers');
insert into tbl_carrier_types (`Carrier_Type`) values ('Railroad Carriers');
insert into tbl_carrier_types (`Carrier_Type`) values ('Ocean Carriers');

-- Inserting 5 Carriers using the Carrier Types
insert into tbl_carriers (`Carrier`, `Carrier_Type_ID`) values ('Airlines', 1);
insert into tbl_carriers (`Carrier`, `Carrier_Type_ID`) values ('UPS', 2);
insert into tbl_carriers (`Carrier`, `Carrier_Type_ID`) values ('Trucking', 3);
insert into tbl_carriers (`Carrier`, `Carrier_Type_ID`) values ('US Express', 4);
insert into tbl_carriers (`Carrier`, `Carrier_Type_ID`) values ('Maersk', 5);
```

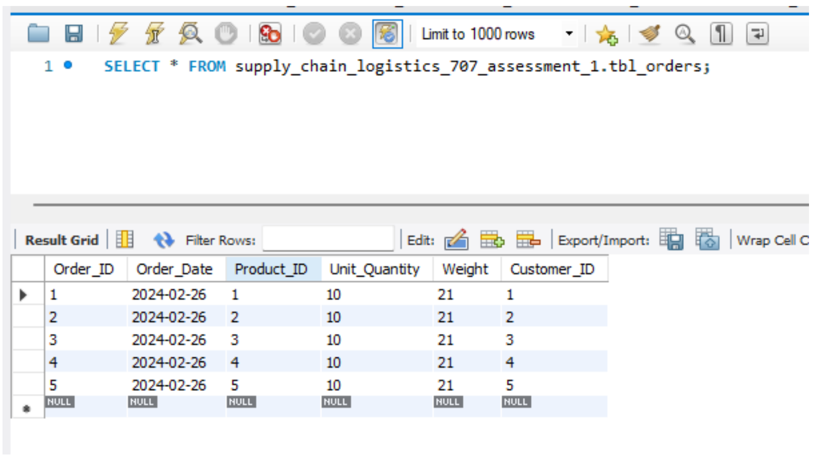
```sql
-- Inserting 5 Customers
insert into tbl_customers (`Customer`) values ('Customer 1');
insert into tbl_customers (`Customer`) values ('Customer 2');
insert into tbl_customers (`Customer`) values ('Customer 3');
insert into tbl_customers (`Customer`) values ('Customer 4');
insert into tbl_customers (`Customer`) values ('Customer 5');

-- Inserting 5 Ports
insert into tbl_ports (`Port`) values ('Port 1');
insert into tbl_ports (`Port`) values ('Port 2');
insert into tbl_ports (`Port`) values ('Port 3');
insert into tbl_ports (`Port`) values ('Port 4');
insert into tbl_ports (`Port`) values ('Port 5');

-- Inserting 5 Plants
insert into tbl_plants (`Plant_Code`) values ('Plant 1');
insert into tbl_plants (`Plant_Code`) values ('Plant 2');
insert into tbl_plants (`Plant_Code`) values ('Plant 3');
insert into tbl_plants (`Plant_Code`) values ('Plant 4');
insert into tbl_plants (`Plant_Code`) values ('Plant 5');

-- Inserting 5 Service Levels
insert into tbl_service_levels (`Service Level`) values ('Service Level 1');
insert into tbl_service_levels (`Service Level`) values ('Service Level 2');
insert into tbl_service_levels (`Service Level`) values ('Service Level 3');
insert into tbl_service_levels (`Service Level`) values ('Service Level 4');
insert into tbl_service_levels (`Service Level`) values ('Service Level 5');

-- Inserting 5 Modes
insert into tbl_mode_desc (`Mode`) values ('Mode 1');
insert into tbl_mode_desc (`Mode`) values ('Mode 2');
insert into tbl_mode_desc (`Mode`) values ('Mode 3');
insert into tbl_mode_desc (`Mode`) values ('Mode 4');
insert into tbl_mode_desc (`Mode`) values ('Mode 5');
```

Output



```
Output
Action Output
#   Time      Action                                                                              Message             Duration / Fetch
1   15:42:41  use supply_chain_logistics_707_assessment_1                                         0 row(s) affected   0.000 sec
2   15:42:41  insert into tbl_products(`Product`) values ('Product Item 1')                       1 row(s) affected   0.015 sec
3   15:42:41  insert into tbl_products(`Product`) values ('Product Item 2')                       1 row(s) affected   0.000 sec
4   15:42:41  insert into tbl_products(`Product`) values ('Product Item 3')                       1 row(s) affected   0.000 sec
5   15:42:41  insert into tbl_products(`Product`) values ('Product Item 4')                       1 row(s) affected   0.015 sec
6   15:42:41  insert into tbl_products(`Product`) values ('Product Item 5')                       1 row(s) affected   0.000 sec
7   15:42:41  insert into tbl_carrier_types (`Carrier_Type`) values ('Common Carriers')           1 row(s) affected   0.016 sec
8   15:42:41  insert into tbl_carrier_types (`Carrier_Type`) values ('Contract Carriers')         1 row(s) affected   0.000 sec
9   15:42:41  insert into tbl_carrier_types (`Carrier_Type`) values ('Trucking Carriers')         1 row(s) affected   0.000 sec
10  15:42:41  insert into tbl_carrier_types (`Carrier_Type`) values ('Railroad Carriers')          1 row(s) affected   0.015 sec
11  15:42:41  insert into tbl_carrier_types (`Carrier_Type`) values ('Ocean Carriers')            1 row(s) affected   0.000 sec
12  15:42:41  insert into tbl_carriers (`Carrier`, `Carrier_Type_ID`) values ('Airlines', 1)      1 row(s) affected   0.016 sec
13  15:42:41  insert into tbl_carriers (`Carrier`, `Carrier_Type_ID`) values ('UPS', 2)           1 row(s) affected   0.000 sec
14  15:42:41  insert into tbl_carriers (`Carrier`, `Carrier_Type_ID`) values ('Trucking', 3)      1 row(s) affected   0.000 sec
15  15:42:41  insert into tbl_carriers (`Carrier`, `Carrier_Type_ID`) values ('US Express', 4)    1 row(s) affected   0.015 sec
16  15:42:41  insert into tbl_carriers (`Carrier`, `Carrier_Type_ID`) values ('Maersk', 5)        1 row(s) affected   0.000 sec
17  15:42:41  insert into tbl_customers (`Customer`) values ('Customer 1')                        1 row(s) affected   0.016 sec
18  15:42:41  insert into tbl_customers (`Customer`) values ('Customer 2')                        1 row(s) affected   0.000 sec
19  15:42:41  insert into tbl_customers (`Customer`) values ('Customer 3')                        1 row(s) affected   0.015 sec
20  15:42:41  insert into tbl_customers (`Customer`) values ('Customer 4')                        1 row(s) affected   0.016 sec
21  15:42:41  insert into tbl_customers (`Customer`) values ('Customer 5')                        1 row(s) affected   0.000 sec
22  15:42:41  insert into tbl_ports (`Port`) values ('Port 1')                                    1 row(s) affected   0.016 sec
23  15:42:41  insert into tbl_ports (`Port`) values ('Port 2')                                    1 row(s) affected   0.015 sec
24  15:42:41  insert into tbl_ports (`Port`) values ('Port 3')                                    1 row(s) affected   0.000 sec
25  15:42:41  insert into tbl_ports (`Port`) values ('Port 4')                                    1 row(s) affected   0.016 sec
26  15:42:41  insert into tbl_ports (`Port`) values ('Port 5')                                    1 row(s) affected   0.000 sec
27  15:42:41  insert into tbl_plants (`Plant_Code`) values ('Plant 1')                            1 row(s) affected   0.016 sec
28  15:42:41  insert into tbl_plants (`Plant_Code`) values ('Plant 2')                            1 row(s) affected   0.000 sec
29  15:42:41  insert into tbl_plants (`Plant_Code`) values ('Plant 3')                            1 row(s) affected   0.015 sec
30  15:42:41  insert into tbl_plants (`Plant_Code`) values ('Plant 4')                            1 row(s) affected   0.000 sec
31  15:42:41  insert into tbl_plants (`Plant_Code`) values ('Plant 5')                            1 row(s) affected   0.016 sec
32  15:42:41  insert into tbl_service_levels (`Service Level`) values ('Service Level 1')         1 row(s) affected   0.000 sec
33  15:42:41  insert into tbl_service_levels (`Service Level`) values ('Service Level 2')         1 row(s) affected   0.015 sec
34  15:42:41  insert into tbl_service_levels (`Service Level`) values ('Service Level 3')         1 row(s) affected   0.000 sec
35  15:42:41  insert into tbl_service_levels (`Service Level`) values ('Service Level 4')         1 row(s) affected   0.016 sec
36  15:42:41  insert into tbl_service_levels (`Service Level`) values ('Service Level 5')         1 row(s) affected   0.000 sec
37  15:42:41  insert into tbl_mode_desc (`Mode`) values ('Mode 1')                                1 row(s) affected   0.016 sec
38  15:42:41  insert into tbl_mode_desc (`Mode`) values ('Mode 2')                                1 row(s) affected   0.000 sec
39  15:42:41  insert into tbl_mode_desc (`Mode`) values ('Mode 3')                                1 row(s) affected   0.000 sec
40  15:42:41  insert into tbl_mode_desc (`Mode`) values ('Mode 4')                                1 row(s) affected   0.016 sec
41  15:42:41  insert into tbl_mode_desc (`Mode`) values ('Mode 5')                                1 row(s) affected   0.000 sec
```

```
-- Inserting 5 Orders
use supply_chain_logistics_707_assessment_1;
INSERT INTO tbl_orders (`Order_Date`, `Product_ID`, `Unit_Quantity`,
`Weight`, `Customer_ID`)
values ('2024-02-26', 1, 10, 20.5, 1);
INSERT INTO tbl_orders (`Order_Date`, `Product_ID`, `Unit_Quantity`,
`Weight`, `Customer_ID`)
values ('2024-02-26', 2, 10, 20.5, 2);
INSERT INTO tbl_orders (`Order_Date`, `Product_ID`, `Unit_Quantity`,
`Weight`, `Customer_ID`)
values ('2024-02-26', 3, 10, 20.5, 3);
INSERT INTO tbl_orders (`Order_Date`, `Product_ID`, `Unit_Quantity`,
`Weight`, `Customer_ID`)
values ('2024-02-26', 4, 10, 20.5, 4);
INSERT INTO tbl_orders (`Order_Date`, `Product_ID`, `Unit_Quantity`,
`Weight`, `Customer_ID`)
values ('2024-02-26', 5, 10, 20.5, 5);
```

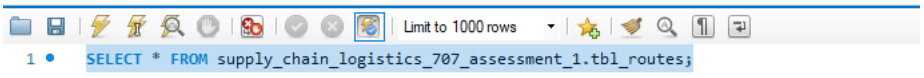


```
SELECT * FROM supply_chain_logistics_707_assessment_1.tbl_orders;
```

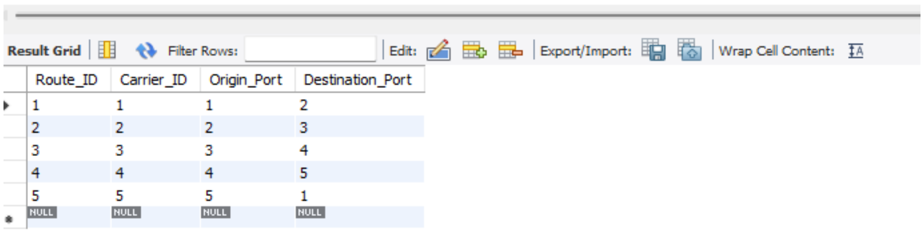
Order_ID	Order_Date	Product_ID	Unit_Quantity	Weight	Customer_ID
1	2024-02-26	1	10	21	1
2	2024-02-26	2	10	21	2
3	2024-02-26	3	10	21	3
4	2024-02-26	4	10	21	4
5	2024-02-26	5	10	21	5
NULL	NULL	NULL	NULL	NULL	NULL

```
-- Inserting 5 Routes
use supply_chain_logistics_707_assessment_1;
INSERT INTO tbl_routes (`Carrier_ID`, `Origin_Port`, `Destination_Port`)
values (1, 1, 2);
INSERT INTO tbl_routes (`Carrier_ID`, `Origin_Port`, `Destination_Port`)
values (2, 2, 3);
INSERT INTO tbl_routes (`Carrier_ID`, `Origin_Port`, `Destination_Port`)
values (3, 3, 4);
INSERT INTO tbl_routes (`Carrier_ID`, `Origin_Port`, `Destination_Port`)
values (4, 4, 5);
```

```
INSERT INTO tbl_routes (`Carrier_ID`, `Origin_Port`, `Destination_Port`)
values (5, 5, 1);
```

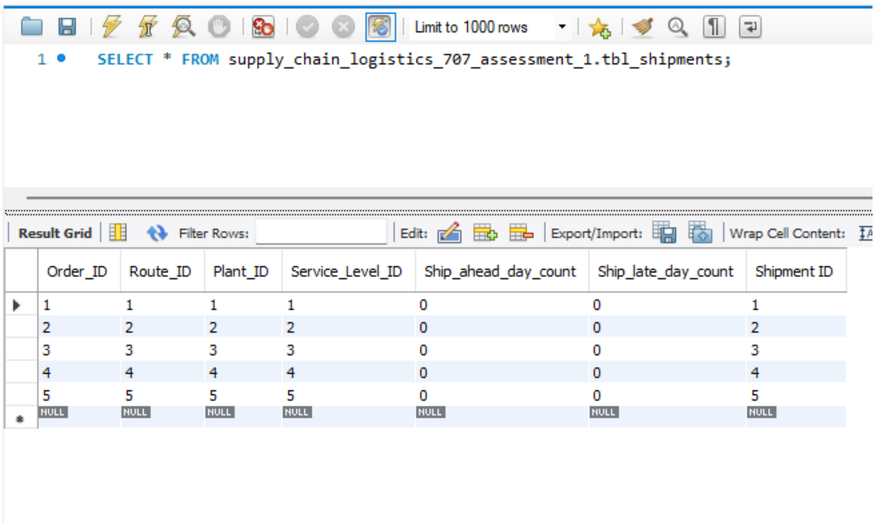


```
SELECT * FROM supply_chain_logistics_707_assessment_1.tbl_routes;
```



Route_ID	Carrier_ID	Origin_Port	Destination_Port
1	1	1	2
2	2	2	3
3	3	3	4
4	4	4	5
5	5	5	1
NULL	NULL	NULL	NULL

```
-- Inserting 5 Shipments
use supply_chain_logistics_707_assessment_1;
INSERT INTO tbl_shipments (`Order_ID`, `Route_ID`, `Plant_ID`,
`Service_Level_ID`, `Ship_ahead_day_count`, `Ship_late_day_count`)
values (1, 1, 1, 1, 0, 0);
INSERT INTO tbl_shipments (`Order_ID`, `Route_ID`, `Plant_ID`,
`Service_Level_ID`, `Ship_ahead_day_count`, `Ship_late_day_count`)
values (2, 2, 2, 2, 0, 0);
INSERT INTO tbl_shipments (`Order_ID`, `Route_ID`, `Plant_ID`,
`Service_Level_ID`, `Ship_ahead_day_count`, `Ship_late_day_count`)
values (3, 3, 3, 3, 0, 0);
INSERT INTO tbl_shipments (`Order_ID`, `Route_ID`, `Plant_ID`,
`Service_Level_ID`, `Ship_ahead_day_count`, `Ship_late_day_count`)
values (4, 4, 4, 4, 0, 0);
INSERT INTO tbl_shipments (`Order_ID`, `Route_ID`, `Plant_ID`,
`Service_Level_ID`, `Ship_ahead_day_count`, `Ship_late_day_count`)
values (5, 5, 5, 5, 0, 0);
```



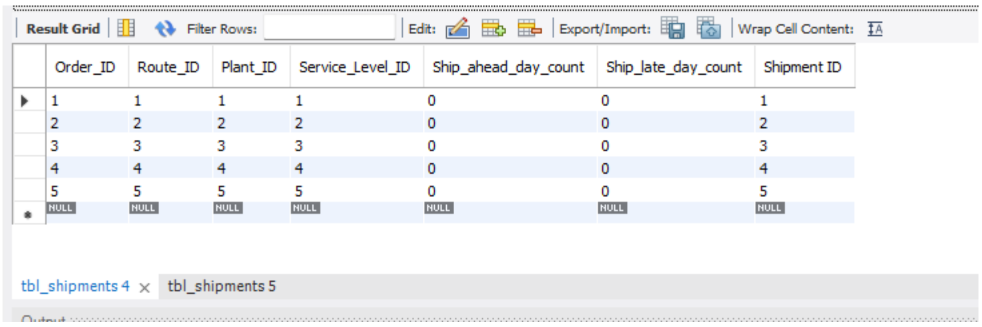
```
SELECT * FROM supply_chain_logistics_707_assessment_1.tbl_shipments;
```

Order_ID	Route_ID	Plant_ID	Service_Level_ID	Ship_ahead_day_count	Ship_late_day_count	Shipment ID
1	1	1	1	0	0	1
2	2	2	2	0	0	2
3	3	3	3	0	0	3
4	4	4	4	0	0	4
5	5	5	5	0	0	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

c. Updates

```
-- Update the Shipment Order so that it goes to Route_ID 1
-- Use Order_ID 2
use supply_chain_logistics_707_assessment_1;
SELECT * FROM tbl_shipments;
Update tbl_shipments
Set Route_ID = 1
Where Order_ID = 2;
SELECT * FROM tbl_shipments;
```

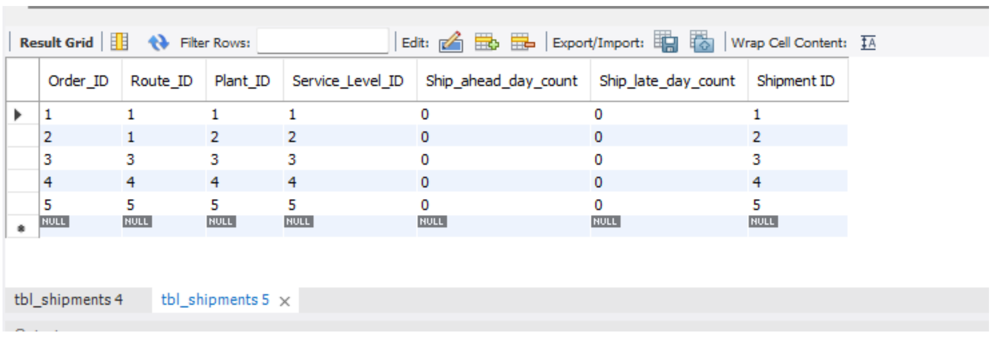
Before



Order_ID	Route_ID	Plant_ID	Service_Level_ID	Ship_ahead_day_count	Ship_late_day_count	Shipment ID
1	1	1	1	0	0	1
2	2	2	2	0	0	2
3	3	3	3	0	0	3
4	4	4	4	0	0	4
5	5	5	5	0	0	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

tbl_shipments 4 × tbl_shipments 5

After

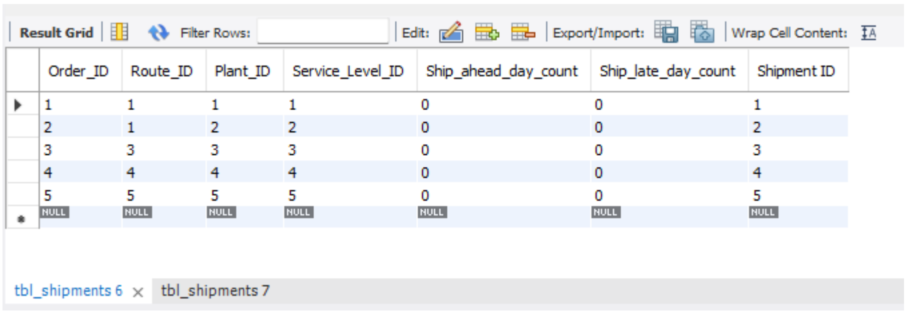


Order_ID	Route_ID	Plant_ID	Service_Level_ID	Ship_ahead_day_count	Ship_late_day_count	Shipment ID
1	1	1	1	0	0	1
2	1	2	2	0	0	2
3	3	3	3	0	0	3
4	4	4	4	0	0	4
5	5	5	5	0	0	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

tbl_shipments 4 tbl_shipments 5 ×

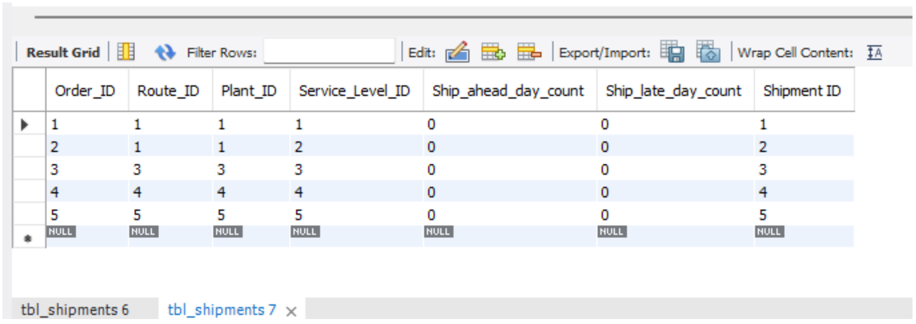
```sql
-- Update Shipment 2 to use Plant ID 1
use supply_chain_logistics_707_assessment_1;
SELECT * FROM tbl_shipments;
Update tbl_shipments
Set Plant_ID = 1
Where `Shipment ID` = 2;
SELECT * FROM tbl_shipments;
```

Before



Order_ID	Route_ID	Plant_ID	Service_Level_ID	Ship_ahead_day_count	Ship_late_day_count	Shipment ID
1	1	1	1	0	0	1
2	1	2	2	0	0	2
3	3	3	3	0	0	3
4	4	4	4	0	0	4
5	5	5	5	0	0	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

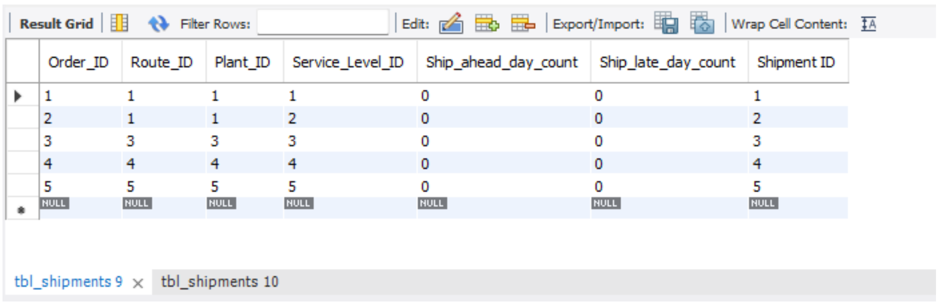
After



Order_ID	Route_ID	Plant_ID	Service_Level_ID	Ship_ahead_day_count	Ship_late_day_count	Shipment ID
1	1	1	1	0	0	1
2	1	1	2	0	0	2
3	3	3	3	0	0	3
4	4	4	4	0	0	4
5	5	5	5	0	0	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```sql
-- Update all Shipments to add one count to the Ship_late_day_count
use supply_chain_logistics_707_assessment_1;
-- Temporarily disable safe update mode for the current session
SET SQL_SAFE_UPDATES = 0;
SELECT * FROM tbl_shipments;
Update tbl_shipments
Set Ship_late_day_count = Ship_late_day_count + 1;
SELECT * FROM tbl_shipments;
```

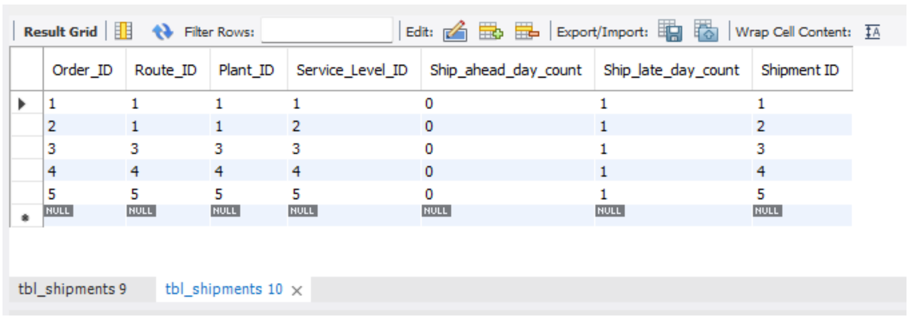
Before



Order_ID	Route_ID	Plant_ID	Service_Level_ID	Ship_ahead_day_count	Ship_late_day_count	Shipment ID
1	1	1	1	0	0	1
2	1	1	2	0	0	2
3	3	3	3	0	0	3
4	4	4	4	0	0	4
5	5	5	5	0	0	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

tbl_shipments 9 tbl_shipments 10

After



Order_ID	Route_ID	Plant_ID	Service_Level_ID	Ship_ahead_day_count	Ship_late_day_count	Shipment ID
1	1	1	1	0	1	1
2	1	1	2	0	1	2
3	3	3	3	0	1	3
4	4	4	4	0	1	4
5	5	5	5	0	1	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

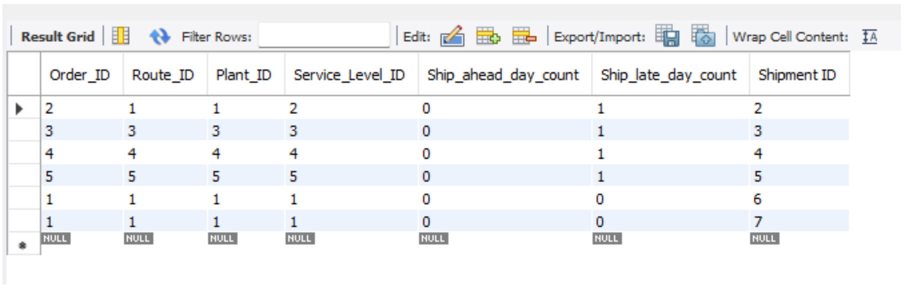
tbl_shipments 9 tbl_shipments 10

d. Removing duplicate records.

** To be able to perform the task, I duplicated one entry.*

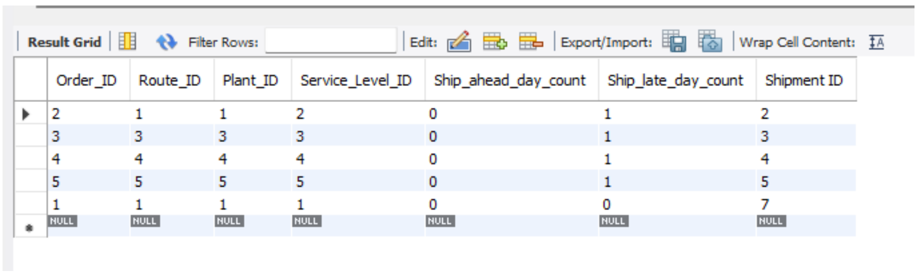
```
-- Insert Duplicate order into the tbl_shipments
use supply_chain_logistics_707_assessment_1;
INSERT INTO tbl_shipments (`Order_ID`, `Route_ID`, `Plant_ID`,
`Service_Level_ID`, `Ship_ahead_day_count`, `Ship_late_day_count`)
values (1, 1, 1, 1, 0, 0);
Select * from tbl_shipments;
Delete s1 from tbl_shipments s1 inner join tbl_shipments s2 where s1.Order_ID
= s2.Order_ID and s1.`Shipment ID` < s2.`Shipment ID`;
Select * from tbl_shipments;
```

Before



Order_ID	Route_ID	Plant_ID	Service_Level_ID	Ship_ahead_day_count	Ship_late_day_count	Shipment ID
2	1	1	2	0	1	2
3	3	3	3	0	1	3
4	4	4	4	0	1	4
5	5	5	5	0	1	5
1	1	1	1	0	0	6
1	1	1	1	0	0	7
NULL	NULL	NULL	NULL	NULL	NULL	NULL

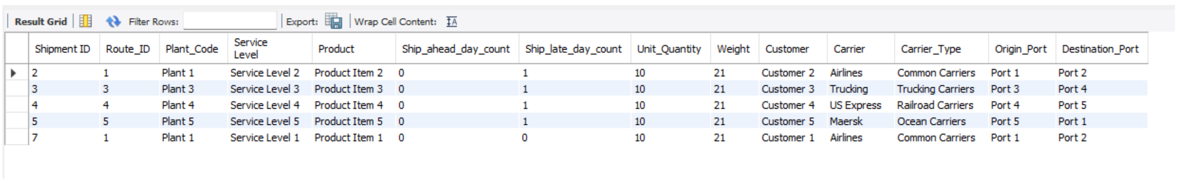
After



Order_ID	Route_ID	Plant_ID	Service_Level_ID	Ship_ahead_day_count	Ship_late_day_count	Shipment ID
2	1	1	2	0	1	2
3	3	3	3	0	1	3
4	4	4	4	0	1	4
5	5	5	5	0	1	5
1	1	1	1	0	0	7
NULL	NULL	NULL	NULL	NULL	NULL	NULL

e. Table joins to retrieve records for data aggregation

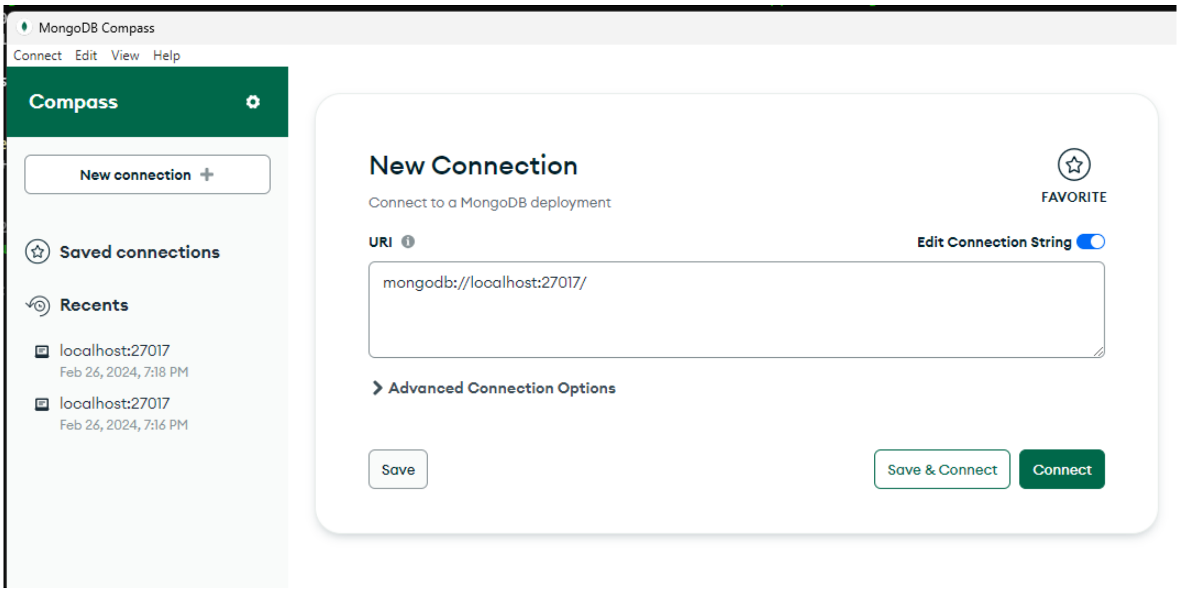
```
-- Join Shipment, Order, Route, Plants, Service Levels, Products, Customers, Ports, Carriers, and Carrier Types

Select s.`Shipment ID`, r.Route_ID, p.Plant_Code, sl.`Service Level`, pt.Product, s.Ship_ahead_day_count, s.Ship_late_day_count, o.Unit_Quantity, o.Weight, c.Customer,

cc.Carrier, ct.Carrier_Type, po.Port as Origin_Port, po2.Port as Destination_Port

from tbl_shipments s left join tbl_orders o on s.Order_ID = o.Order_ID

left join tbl_routes r on s.Route_ID = r.Route_ID

left join tbl_plants p on s.Plant_ID = p.Plant_ID

left join tbl_service_levels sl on s.Service_Level_ID = sl.Service_Level_ID

left join tbl_products pt on o.Product_ID = pt.Product_ID

left join tbl_customers c on o.Customer_ID = c.Customer_ID

left join tbl_ports po on r.Origin_Port = po.Port_ID

left join tbl_ports po2 on r.Destination_Port = po2.Port_ID

left join tbl_carriers cc on r.Carrier_ID = cc.Carrier_ID

left join tbl_carrier_types ct on cc.Carrier_Type_ID = ct.Carrier_Type_ID;
```



Shipment ID	Route_ID	Plant_Code	Service Level	Product	Ship_ahead_day_count	Ship_late_day_count	Unit_Quantity	Weight	Customer	Carrier	Carrier_Type	Origin_Port	Destination_Port
2	1	Plant 1	Service Level 2	Product Item 2	0	1	10	21	Customer 2	Airlines	Common Carriers	Port 1	Port 2
3	3	Plant 3	Service Level 3	Product Item 3	0	1	10	21	Customer 3	Trucking	Trucking Carriers	Port 3	Port 4
4	4	Plant 4	Service Level 4	Product Item 4	0	1	10	21	Customer 4	US Express	Railroad Carriers	Port 4	Port 5
5	5	Plant 5	Service Level 5	Product Item 5	0	1	10	21	Customer 5	Maersk	Ocean Carriers	Port 5	Port 1
7	1	Plant 1	Service Level 1	Product Item 1	0	0	10	21	Customer 1	Airlines	Common Carriers	Port 1	Port 2

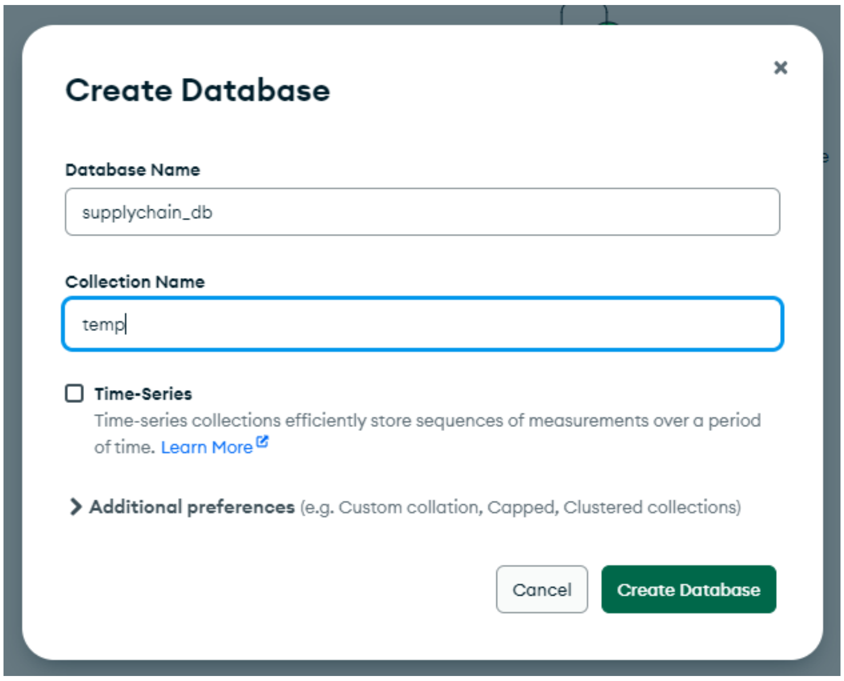
B. Non-Relational – using MongoDB Compass

Connect to MongoDB using MongoDB Compass



mongodb://localhost:27017/

create a new Database called "supplychain_db" and a "temp" collection name.

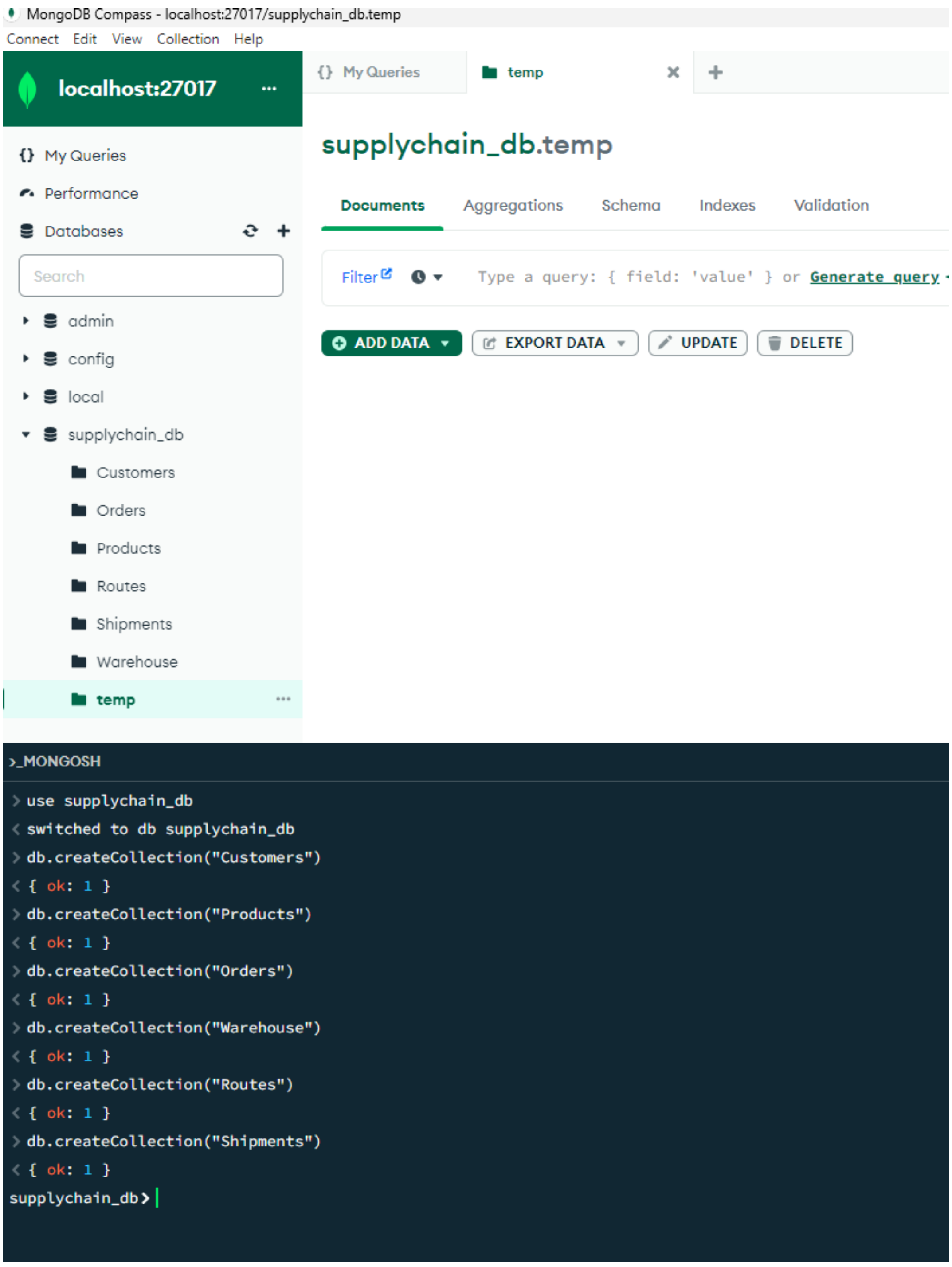

Create Database
Time-Series
Time-series collections efficiently store sequences of measurements over a period of time. Learn More
Additional preferences (e.g. Custom collation, Capped, Clustered collections)
Cancel
Create Database

Database Name: supplychain_db

Collection Name: temp

a. Collection Creation

- Using the mongosh feature of Mongodb compass, we are going to create new collections under the database "supplychain_db"
- Using the command db.createCollection([collection_name]), Create collections for Customers, Products, Orders, Warehouse, Routes and Shipments

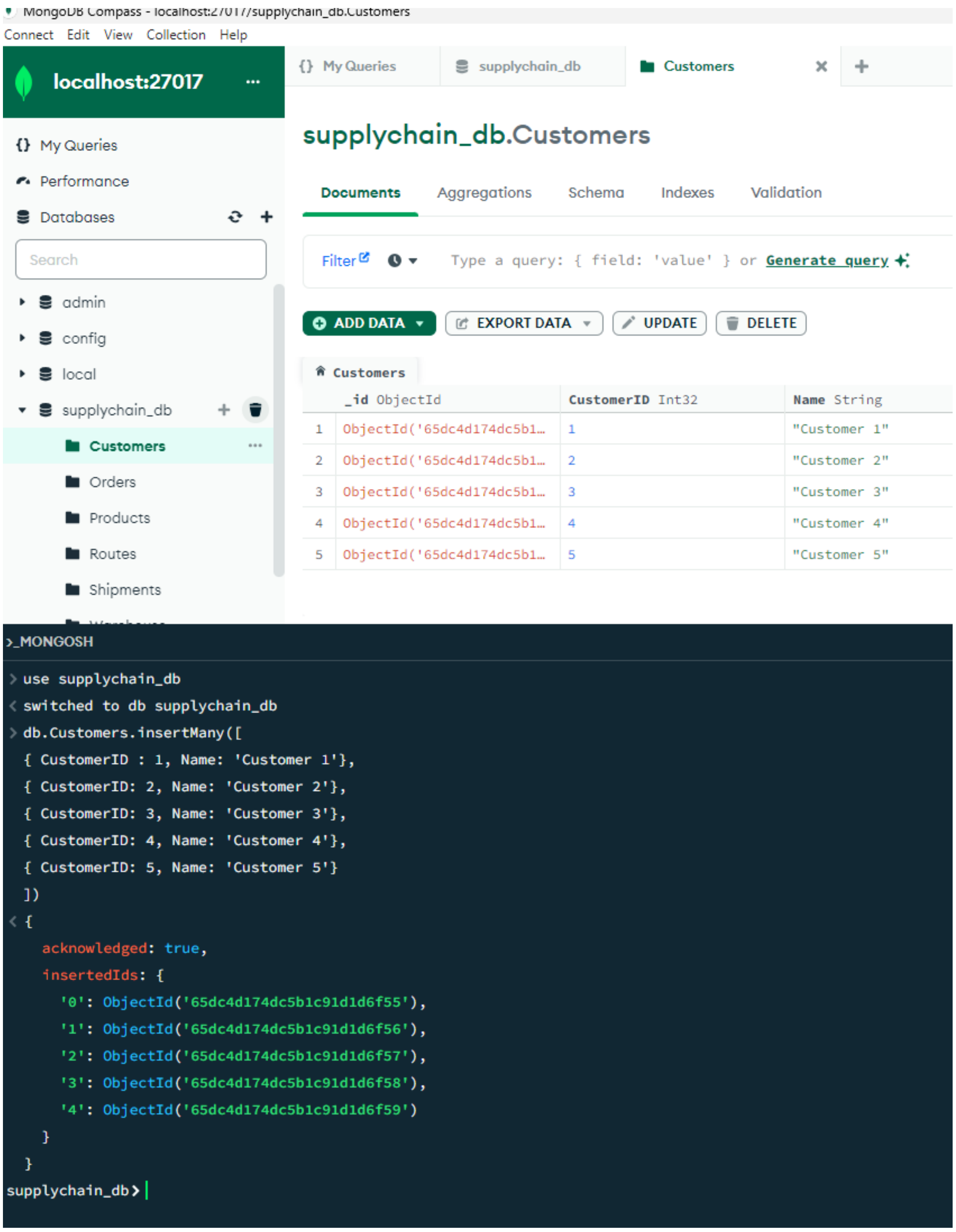


```
> use supplychain_db
< switched to db supplychain_db
> db.createCollection("Customers")
< { ok: 1 }
> db.createCollection("Products")
< { ok: 1 }
> db.createCollection("Orders")
< { ok: 1 }
> db.createCollection("Warehouse")
< { ok: 1 }
> db.createCollection("Routes")
< { ok: 1 }
> db.createCollection("Shipments")
< { ok: 1 }
supplychain_db>
```

b. Document Insertion

- Using db.[collection_name].insertMany or db.[collection_name].insertOne, insert 5 sample documents for each entity collection.

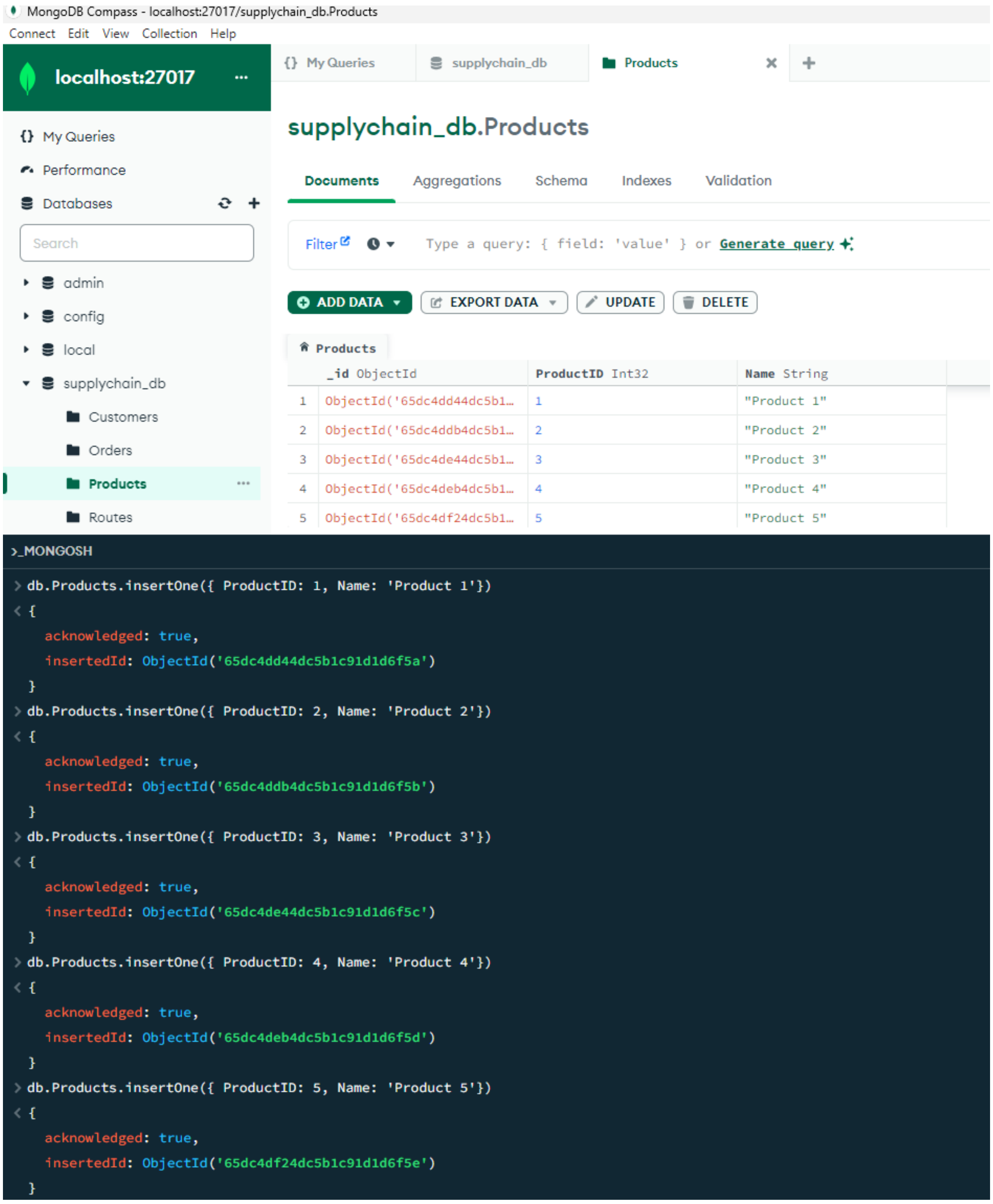
Customers



	_id ObjectId	CustomerID Int32	Name String
1	ObjectId('65dc4d174dc5b1…	1	"Customer 1"
2	ObjectId('65dc4d174dc5b1…	2	"Customer 2"
3	ObjectId('65dc4d174dc5b1…	3	"Customer 3"
4	ObjectId('65dc4d174dc5b1…	4	"Customer 4"
5	ObjectId('65dc4d174dc5b1…	5	"Customer 5"

```
>_MONGOSH
> use supplychain_db
< switched to db supplychain_db
> db.Customers.insertMany([
  { CustomerID : 1, Name: 'Customer 1'},
  { CustomerID: 2, Name: 'Customer 2'},
  { CustomerID: 3, Name: 'Customer 3'},
  { CustomerID: 4, Name: 'Customer 4'},
  { CustomerID: 5, Name: 'Customer 5'}
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('65dc4d174dc5b1c91d1d6f55'),
      '1': ObjectId('65dc4d174dc5b1c91d1d6f56'),
      '2': ObjectId('65dc4d174dc5b1c91d1d6f57'),
      '3': ObjectId('65dc4d174dc5b1c91d1d6f58'),
      '4': ObjectId('65dc4d174dc5b1c91d1d6f59')
    }
  }
supplychain_db>
```

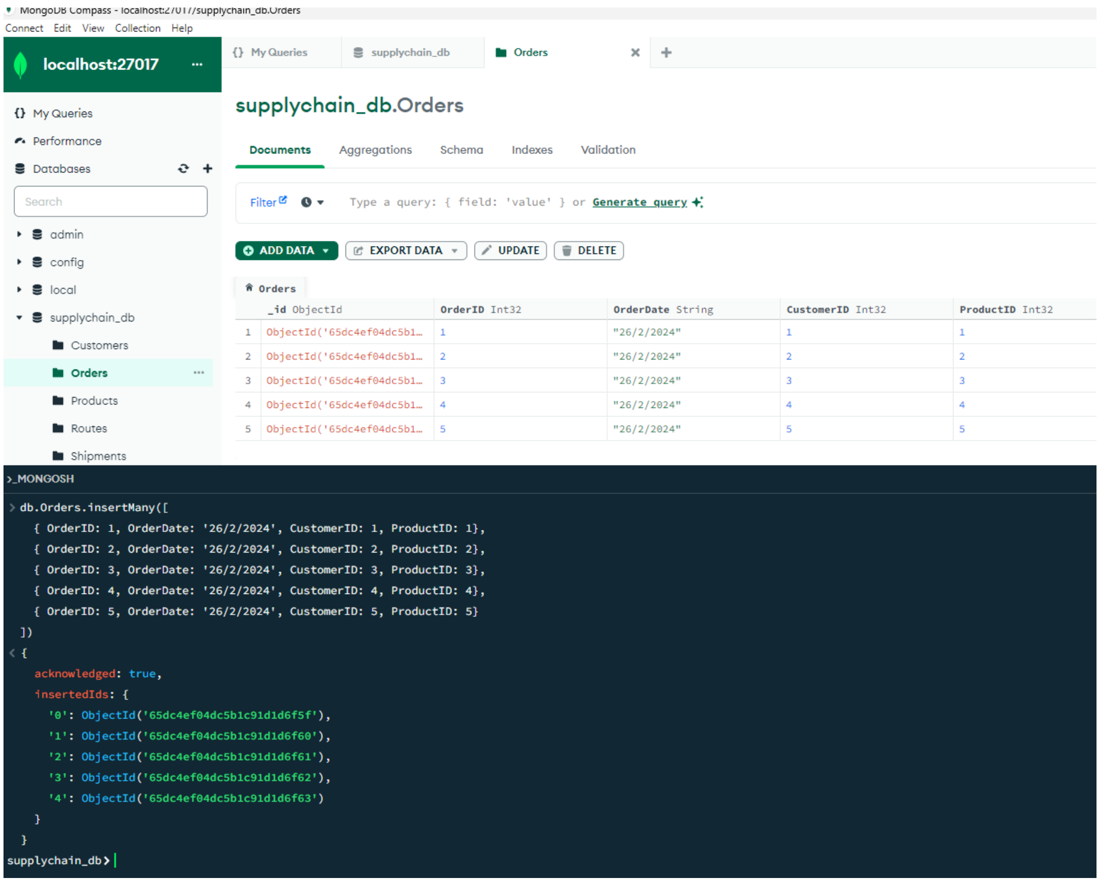
Products



	_id ObjectId	ProductID Int32	Name String
1	ObjectId('65dc4dd44dc5b1…	1	"Product 1"
2	ObjectId('65dc4ddb4dc5b1…	2	"Product 2"
3	ObjectId('65dc4de44dc5b1…	3	"Product 3"
4	ObjectId('65dc4deb4dc5b1…	4	"Product 4"
5	ObjectId('65dc4df24dc5b1…	5	"Product 5"

```
> db.Products.insertOne({ ProductID: 1, Name: 'Product 1'})
< {
    acknowledged: true,
    insertedId: ObjectId('65dc4dd44dc5b1c91d1d6f5a')
  }
> db.Products.insertOne({ ProductID: 2, Name: 'Product 2'})
< {
    acknowledged: true,
    insertedId: ObjectId('65dc4ddb4dc5b1c91d1d6f5b')
  }
> db.Products.insertOne({ ProductID: 3, Name: 'Product 3'})
< {
    acknowledged: true,
    insertedId: ObjectId('65dc4de44dc5b1c91d1d6f5c')
  }
> db.Products.insertOne({ ProductID: 4, Name: 'Product 4'})
< {
    acknowledged: true,
    insertedId: ObjectId('65dc4deb4dc5b1c91d1d6f5d')
  }
> db.Products.insertOne({ ProductID: 5, Name: 'Product 5'})
< {
    acknowledged: true,
    insertedId: ObjectId('65dc4df24dc5b1c91d1d6f5e')
  }
```

Orders

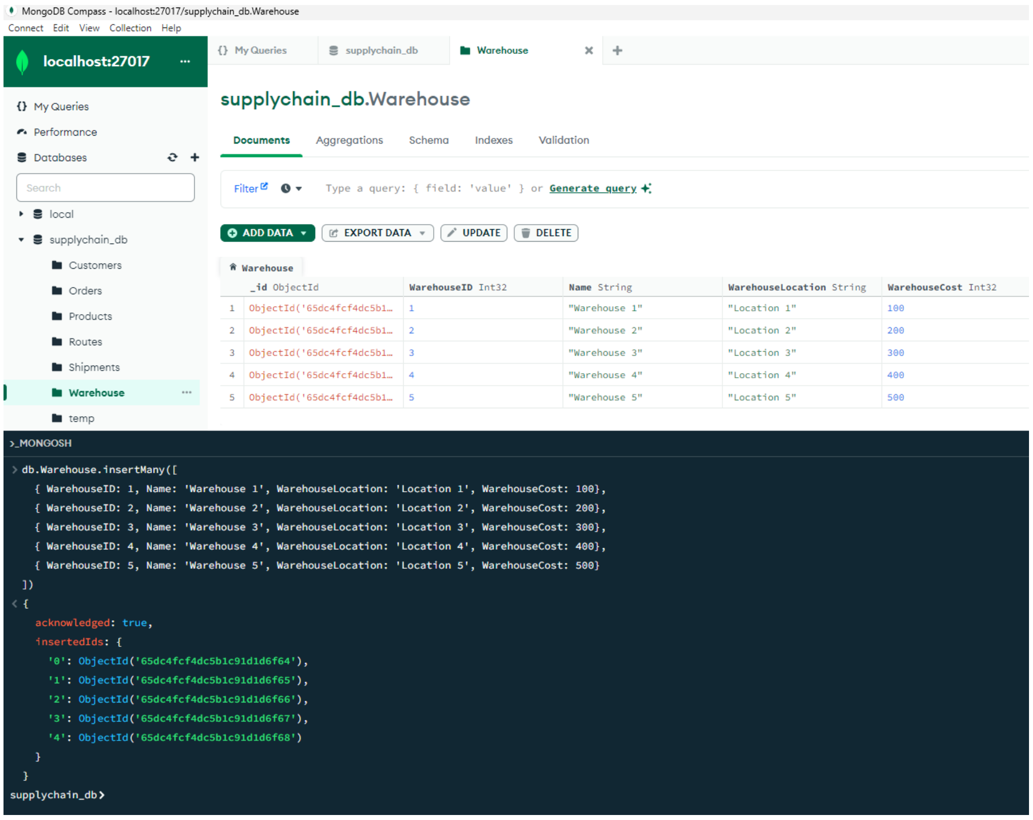

MongoDB Compass - localhost:27017/supplychain_db.Orders
Connect Edit View Collection Help
localhost:27017
My Queries
Performance
Databases
Search
admin
config
local
supplychain_db
Customers
Orders
Products
Routes
Shipments
supplychain_db.Orders
Documents Aggregations Schema Indexes Validation
Filter
Type a query: { field: 'value' } or Generate query
ADD DATA
EXPORT DATA
UPDATE
DELETE
>_MONGOSH

Orders

	_id ObjectId	OrderID Int32	OrderDate String	CustomerID Int32	ProductID Int32
1	ObjectId('65dc4ef04dc5b1…	1	"26/2/2024"	1	1
2	ObjectId('65dc4ef04dc5b1…	2	"26/2/2024"	2	2
3	ObjectId('65dc4ef04dc5b1…	3	"26/2/2024"	3	3
4	ObjectId('65dc4ef04dc5b1…	4	"26/2/2024"	4	4
5	ObjectId('65dc4ef04dc5b1…	5	"26/2/2024"	5	5

```
> db.Orders.insertMany([
    { OrderID: 1, OrderDate: '26/2/2024', CustomerID: 1, ProductID: 1},
    { OrderID: 2, OrderDate: '26/2/2024', CustomerID: 2, ProductID: 2},
    { OrderID: 3, OrderDate: '26/2/2024', CustomerID: 3, ProductID: 3},
    { OrderID: 4, OrderDate: '26/2/2024', CustomerID: 4, ProductID: 4},
    { OrderID: 5, OrderDate: '26/2/2024', CustomerID: 5, ProductID: 5}
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('65dc4ef04dc5b1c91d1d6f5f'),
      '1': ObjectId('65dc4ef04dc5b1c91d1d6f60'),
      '2': ObjectId('65dc4ef04dc5b1c91d1d6f61'),
      '3': ObjectId('65dc4ef04dc5b1c91d1d6f62'),
      '4': ObjectId('65dc4ef04dc5b1c91d1d6f63')
    }
  }
supplychain_db>
```

Warehouse

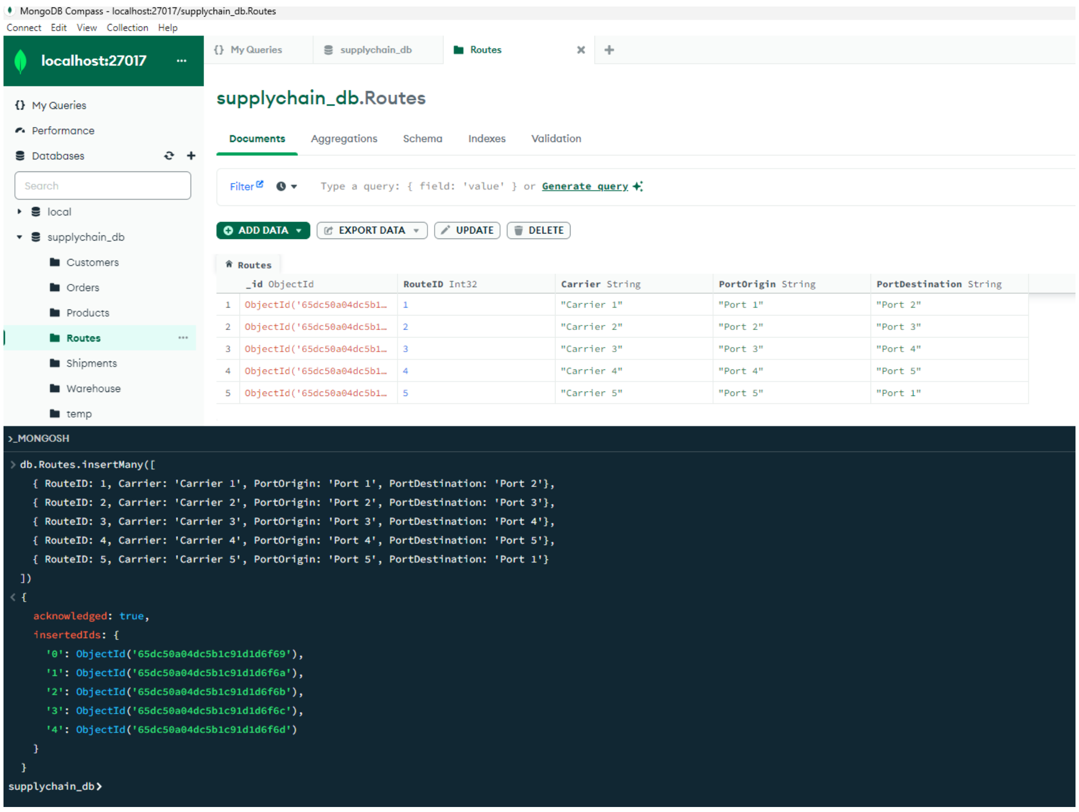


Warehouse

	_id ObjectId	WarehouseID Int32	Name String	WarehouseLocation String	WarehouseCost Int32
1	ObjectId('65dc4fcf4dc5b1…	1	"Warehouse 1"	"Location 1"	100
2	ObjectId('65dc4fcf4dc5b1…	2	"Warehouse 2"	"Location 2"	200
3	ObjectId('65dc4fcf4dc5b1…	3	"Warehouse 3"	"Location 3"	300
4	ObjectId('65dc4fcf4dc5b1…	4	"Warehouse 4"	"Location 4"	400
5	ObjectId('65dc4fcf4dc5b1…	5	"Warehouse 5"	"Location 5"	500

```
> db.Warehouse.insertMany([
    { WarehouseID: 1, Name: 'Warehouse 1', WarehouseLocation: 'Location 1', WarehouseCost: 100},
    { WarehouseID: 2, Name: 'Warehouse 2', WarehouseLocation: 'Location 2', WarehouseCost: 200},
    { WarehouseID: 3, Name: 'Warehouse 3', WarehouseLocation: 'Location 3', WarehouseCost: 300},
    { WarehouseID: 4, Name: 'Warehouse 4', WarehouseLocation: 'Location 4', WarehouseCost: 400},
    { WarehouseID: 5, Name: 'Warehouse 5', WarehouseLocation: 'Location 5', WarehouseCost: 500}
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('65dc4fcf4dc5b1c91d1d6f64'),
      '1': ObjectId('65dc4fcf4dc5b1c91d1d6f65'),
      '2': ObjectId('65dc4fcf4dc5b1c91d1d6f66'),
      '3': ObjectId('65dc4fcf4dc5b1c91d1d6f67'),
      '4': ObjectId('65dc4fcf4dc5b1c91d1d6f68')
    }
  }
supplychain_db>
```

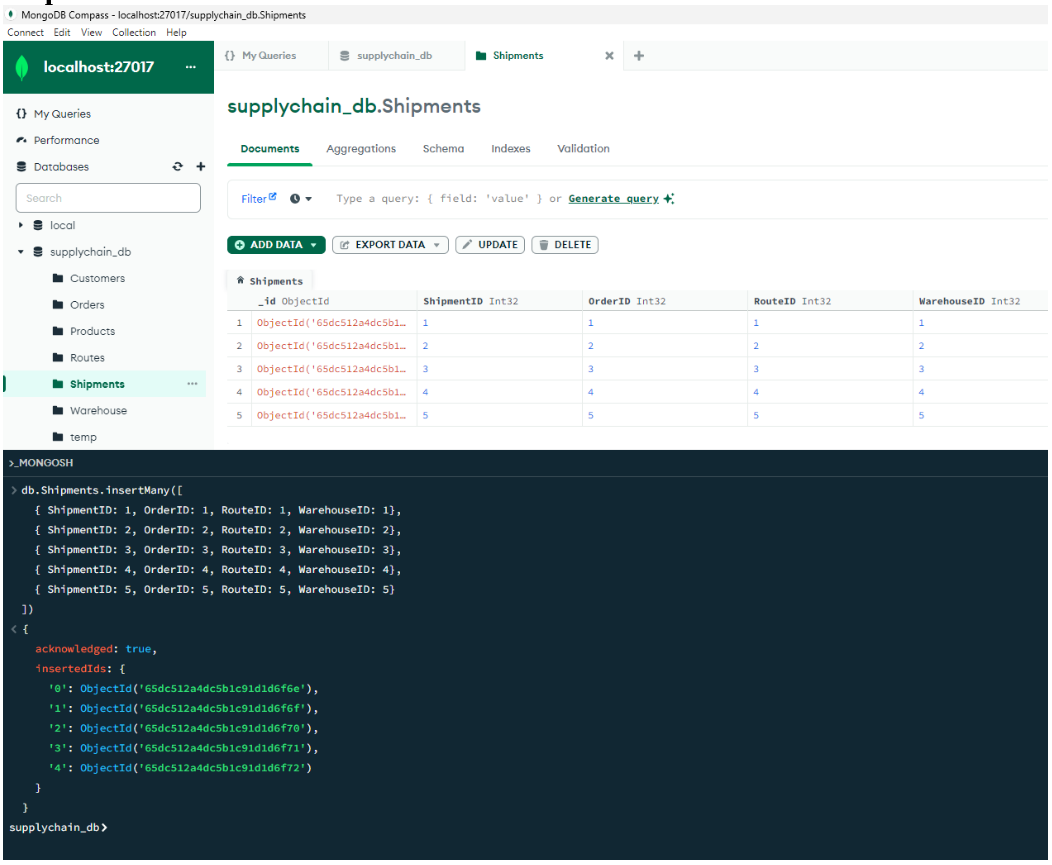
Routes



	_id ObjectId	RouteID Int32	Carrier String	PortOrigin String	PortDestination String
1	ObjectId('65dc50a04dc5b1…	1	"Carrier 1"	"Port 1"	"Port 2"
2	ObjectId('65dc50a04dc5b1…	2	"Carrier 2"	"Port 2"	"Port 3"
3	ObjectId('65dc50a04dc5b1…	3	"Carrier 3"	"Port 3"	"Port 4"
4	ObjectId('65dc50a04dc5b1…	4	"Carrier 4"	"Port 4"	"Port 5"
5	ObjectId('65dc50a04dc5b1…	5	"Carrier 5"	"Port 5"	"Port 1"

```
>_MONGOSH
> db.Routes.insertMany([
    { RouteID: 1, Carrier: 'Carrier 1', PortOrigin: 'Port 1', PortDestination: 'Port 2'},
    { RouteID: 2, Carrier: 'Carrier 2', PortOrigin: 'Port 2', PortDestination: 'Port 3'},
    { RouteID: 3, Carrier: 'Carrier 3', PortOrigin: 'Port 3', PortDestination: 'Port 4'},
    { RouteID: 4, Carrier: 'Carrier 4', PortOrigin: 'Port 4', PortDestination: 'Port 5'},
    { RouteID: 5, Carrier: 'Carrier 5', PortOrigin: 'Port 5', PortDestination: 'Port 1'}
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('65dc50a04dc5b1c91d1d6f69'),
      '1': ObjectId('65dc50a04dc5b1c91d1d6f6a'),
      '2': ObjectId('65dc50a04dc5b1c91d1d6f6b'),
      '3': ObjectId('65dc50a04dc5b1c91d1d6f6c'),
      '4': ObjectId('65dc50a04dc5b1c91d1d6f6d')
    }
  }
supplychain_db>
```

Shipments



	_id ObjectId	ShipmentID Int32	OrderID Int32	RouteID Int32	WarehouseID Int32
1	ObjectId('65dc512a4dc5b1…	1	1	1	1
2	ObjectId('65dc512a4dc5b1…	2	2	2	2
3	ObjectId('65dc512a4dc5b1…	3	3	3	3
4	ObjectId('65dc512a4dc5b1…	4	4	4	4
5	ObjectId('65dc512a4dc5b1…	5	5	5	5

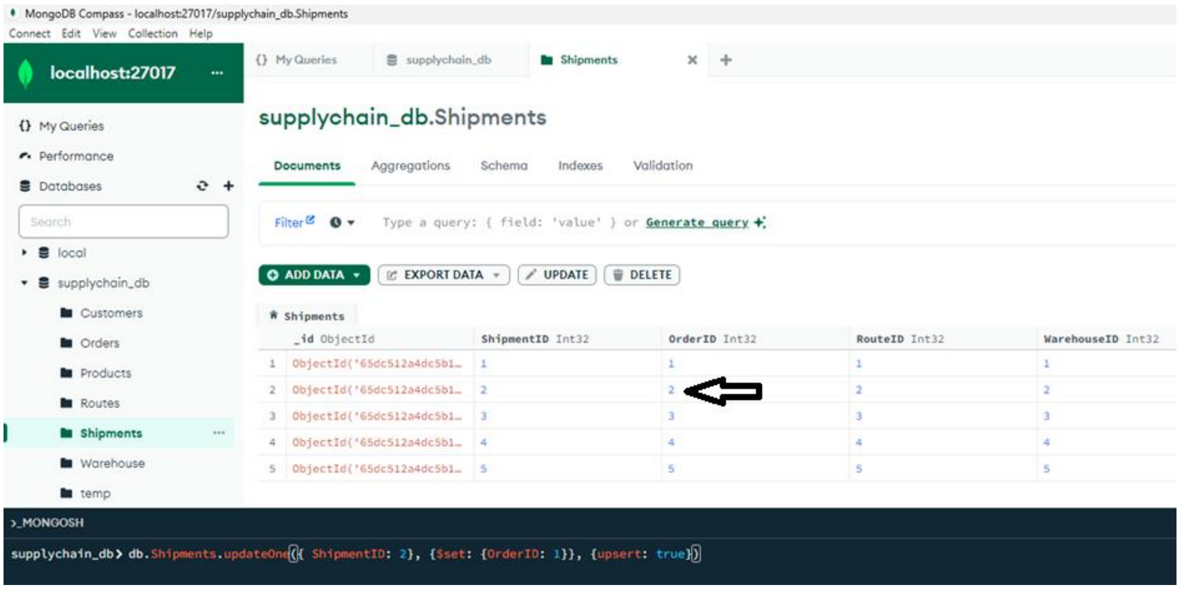
```
>_MONGOSH
> db.Shipments.insertMany([
    { ShipmentID: 1, OrderID: 1, RouteID: 1, WarehouseID: 1},
    { ShipmentID: 2, OrderID: 2, RouteID: 2, WarehouseID: 2},
    { ShipmentID: 3, OrderID: 3, RouteID: 3, WarehouseID: 3},
    { ShipmentID: 4, OrderID: 4, RouteID: 4, WarehouseID: 4},
    { ShipmentID: 5, OrderID: 5, RouteID: 5, WarehouseID: 5}
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('65dc512a4dc5b1c91d1d6f6e'),
      '1': ObjectId('65dc512a4dc5b1c91d1d6f6f'),
      '2': ObjectId('65dc512a4dc5b1c91d1d6f70'),
      '3': ObjectId('65dc512a4dc5b1c91d1d6f71'),
      '4': ObjectId('65dc512a4dc5b1c91d1d6f72')
    }
  }
supplychain_db>
```

c. Update Queries

- Using db.[collection_name].updateOne, update atleast 3 sample documents for each entity collection.

Updating the Order ID field in the Shipments Collection

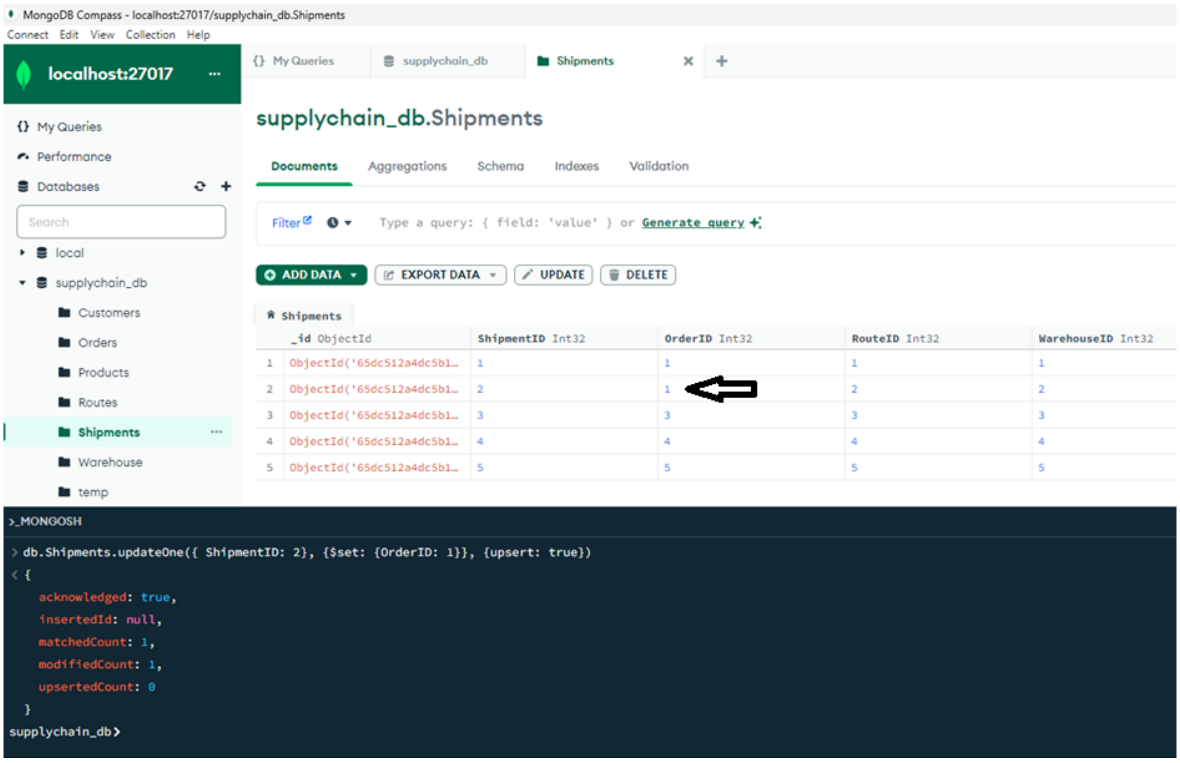
Before



	_id ObjectId	ShipmentID Int32	OrderID Int32	RouteID Int32	WarehouseID Int32
1	ObjectId('65dc512a4dc5b1…	1	1	1	1
2	ObjectId('65dc512a4dc5b1…	2	2	2	2
3	ObjectId('65dc512a4dc5b1…	3	3	3	3
4	ObjectId('65dc512a4dc5b1…	4	4	4	4
5	ObjectId('65dc512a4dc5b1…	5	5	5	5

```
>_MONGOSH
supplychain_db> db.Shipments.updateOne({ ShipmentID: 2}, {$set: {OrderID: 1}}, {upsert: true})
```

After

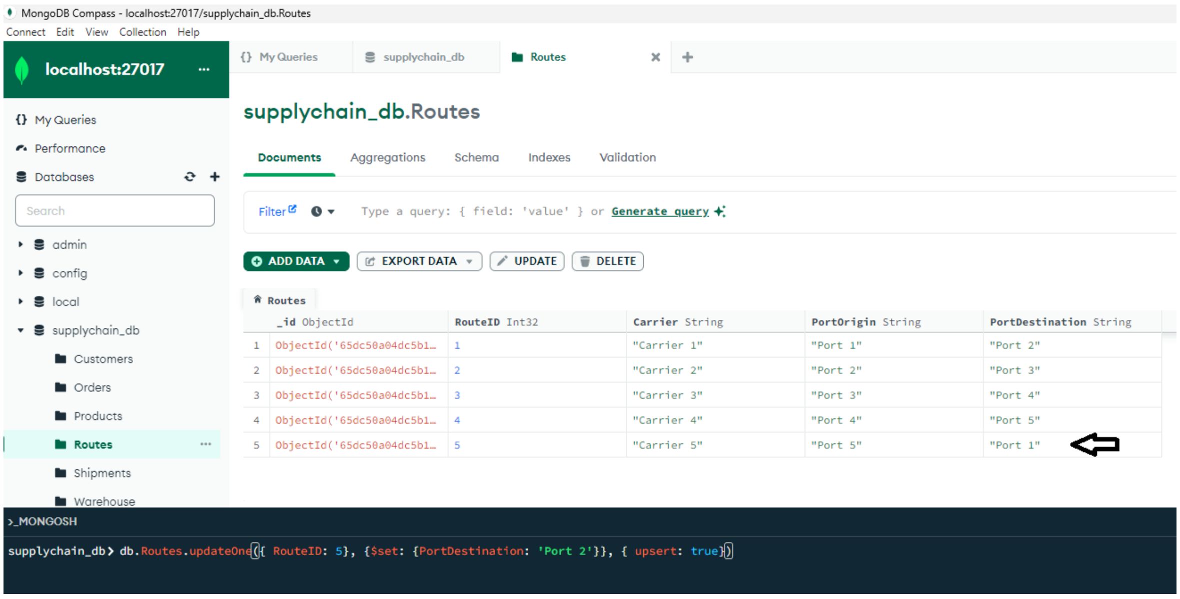


	_id ObjectId	ShipmentID Int32	OrderID Int32	RouteID Int32	WarehouseID Int32
1	ObjectId('65dc512a4dc5b1…	1	1	1	1
2	ObjectId('65dc512a4dc5b1…	2	1	2	2
3	ObjectId('65dc512a4dc5b1…	3	3	3	3
4	ObjectId('65dc512a4dc5b1…	4	4	4	4
5	ObjectId('65dc512a4dc5b1…	5	5	5	5

```
>_MONGOSH
> db.Shipments.updateOne({ ShipmentID: 2}, {$set: {OrderID: 1}}, {upsert: true})
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
supplychain_db>
```

Updating Port Destination in the Routes Collection

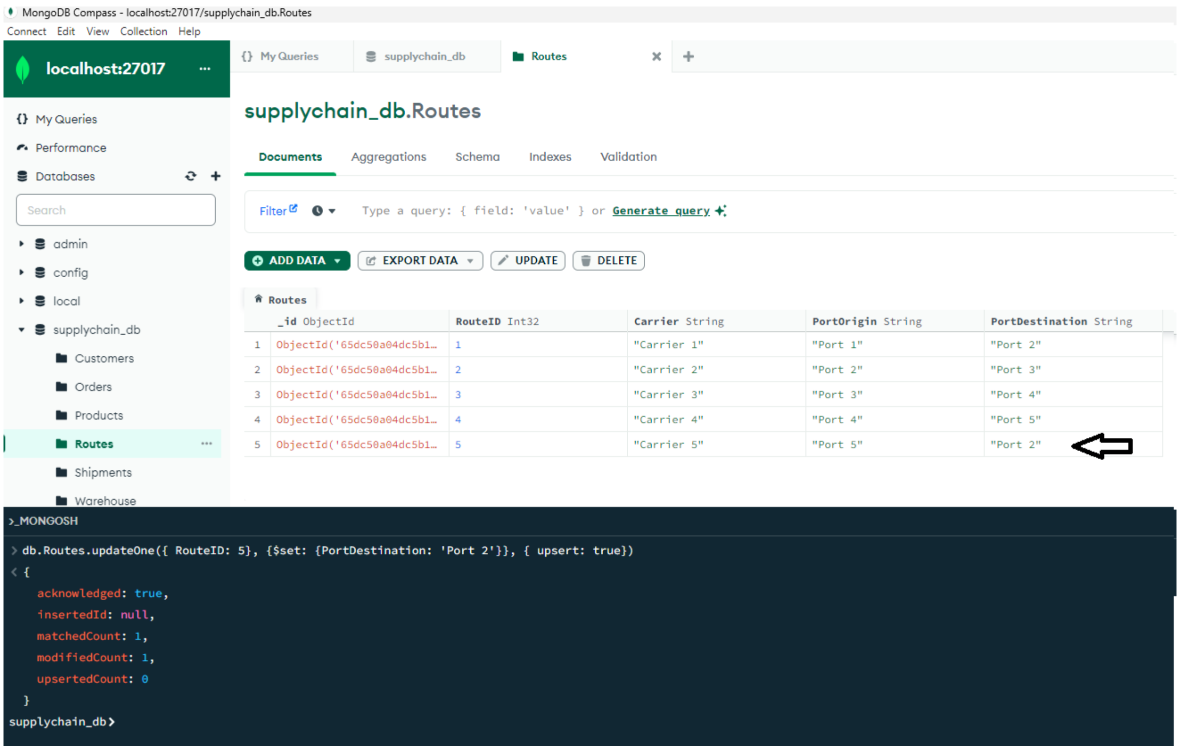
Before



	_id ObjectId	RouteID Int32	Carrier String	PortOrigin String	PortDestination String
1	ObjectId('65dc50a04dc5b1…	1	"Carrier 1"	"Port 1"	"Port 2"
2	ObjectId('65dc50a04dc5b1…	2	"Carrier 2"	"Port 2"	"Port 3"
3	ObjectId('65dc50a04dc5b1…	3	"Carrier 3"	"Port 3"	"Port 4"
4	ObjectId('65dc50a04dc5b1…	4	"Carrier 4"	"Port 4"	"Port 5"
5	ObjectId('65dc50a04dc5b1…	5	"Carrier 5"	"Port 5"	"Port 1"

```
supplychain_db> db.Routes.updateOne({ RouteID: 5}, {$set: {PortDestination: 'Port 2'}}, { upsert: true})
```

After

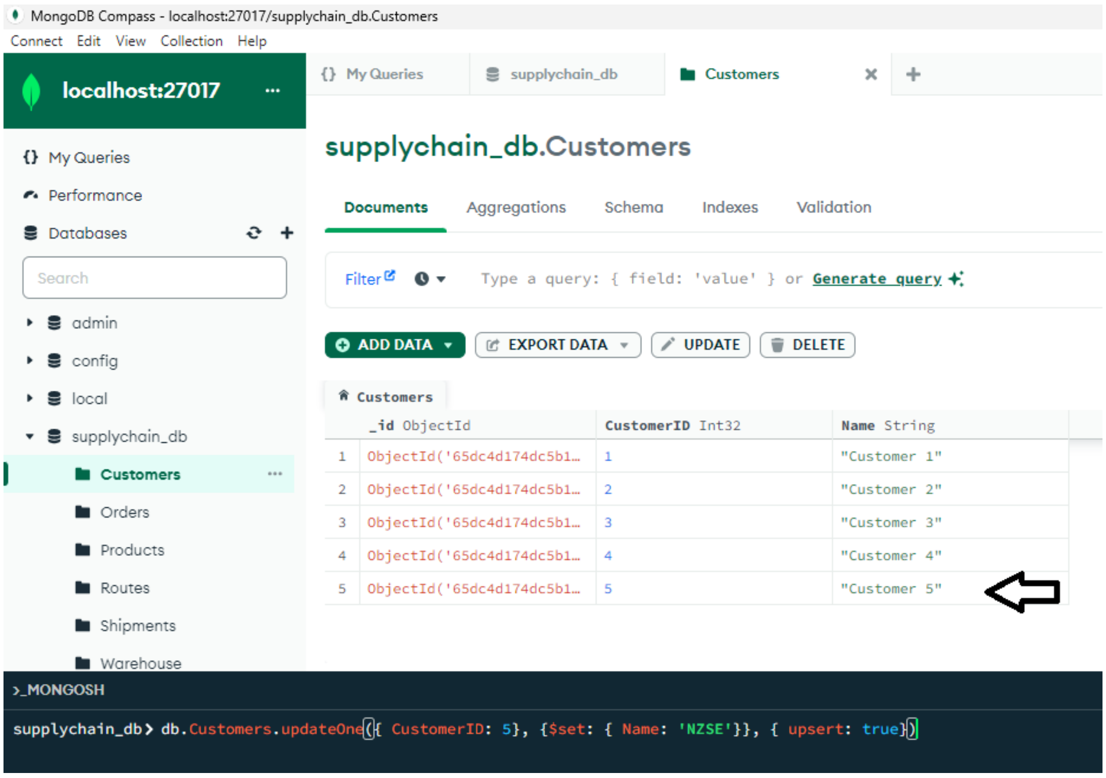


	_id ObjectId	RouteID Int32	Carrier String	PortOrigin String	PortDestination String
1	ObjectId('65dc50a04dc5b1…	1	"Carrier 1"	"Port 1"	"Port 2"
2	ObjectId('65dc50a04dc5b1…	2	"Carrier 2"	"Port 2"	"Port 3"
3	ObjectId('65dc50a04dc5b1…	3	"Carrier 3"	"Port 3"	"Port 4"
4	ObjectId('65dc50a04dc5b1…	4	"Carrier 4"	"Port 4"	"Port 5"
5	ObjectId('65dc50a04dc5b1…	5	"Carrier 5"	"Port 5"	"Port 2"

```
> db.Routes.updateOne({ RouteID: 5}, {$set: {PortDestination: 'Port 2'}}, { upsert: true})
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
supplychain_db>
```

Updating Customer Name in Customers Collection

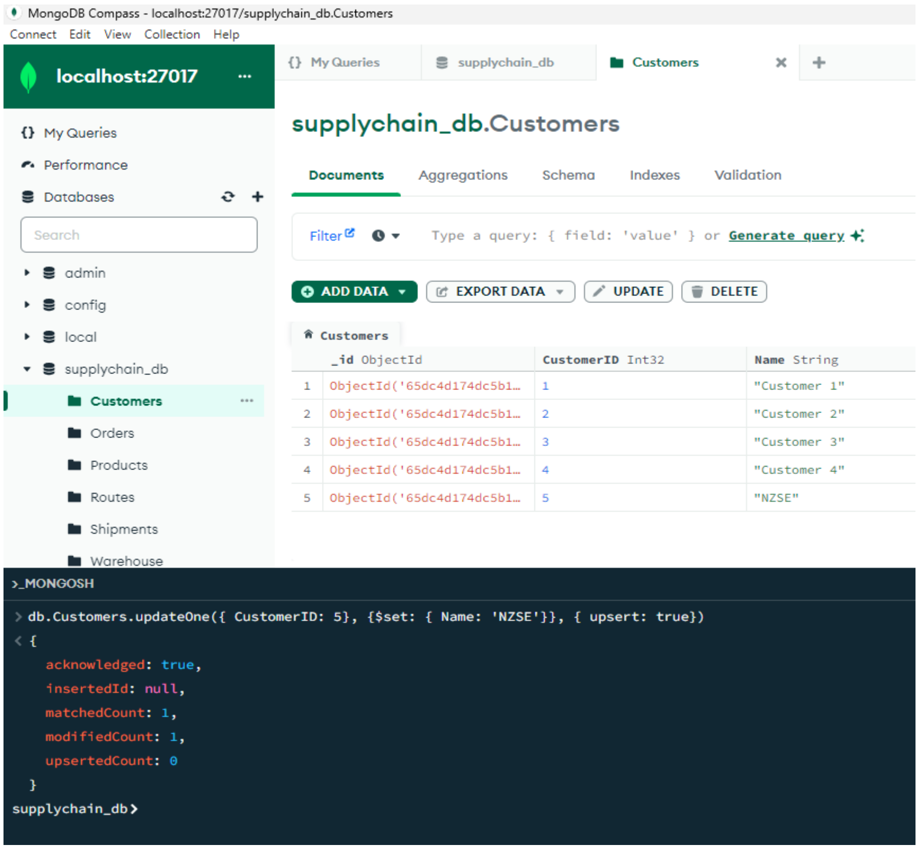
Before



	_id ObjectId	CustomerID Int32	Name String
1	ObjectId('65dc4d174dc5b1…	1	"Customer 1"
2	ObjectId('65dc4d174dc5b1…	2	"Customer 2"
3	ObjectId('65dc4d174dc5b1…	3	"Customer 3"
4	ObjectId('65dc4d174dc5b1…	4	"Customer 4"
5	ObjectId('65dc4d174dc5b1…	5	"Customer 5"

```
supplychain_db> db.Customers.updateOne({ CustomerID: 5}, {$set: { Name: 'NZSE'}}, { upsert: true})
```

After



	_id ObjectId	CustomerID Int32	Name String
1	ObjectId('65dc4d174dc5b1…	1	"Customer 1"
2	ObjectId('65dc4d174dc5b1…	2	"Customer 2"
3	ObjectId('65dc4d174dc5b1…	3	"Customer 3"
4	ObjectId('65dc4d174dc5b1…	4	"Customer 4"
5	ObjectId('65dc4d174dc5b1…	5	"NZSE"

```
> db.Customers.updateOne({ CustomerID: 5}, {$set: { Name: 'NZSE'}}, { upsert: true})
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
supplychain_db>
```

Chapter 5: Documentation

I used the Brunel University of London web page data for relational and non-relational data models in this project. The reason for using the same data is to show the flexibility of the two models. Initially, the datasets were structured as non-relational. All the worksheets or tables can stand alone in the presence of the attributes on the entity forming one document. The data was then converted into relational by assigning the primary and foreign keys to establish the relationships.

The raw data is downloaded from the website in an Excel file, and normalization is manually done by assigning the primary and foreign keys and creating and converting the necessary tables to facilitate the establishment of the relationships for the relational data model. To perform the required tasks, I used the MySQL database because I've seen a lot of YouTube tutorials about this, and I believe it is commonly used in the business. In the database creation, I used the dataset's original structure, excluding its 9,215 data records in the order list. The reason is to see the required tasks efficiently in the database's output section.

Using the same datasets in the non-relational model, I extracted some variables to create a collection of documents in the database. I used MongoDB for non-relational due to my familiarity with the database.

The challenge in doing this assessment is the short period of time to study the two data models and the databases. As a beginner with no experience using the database, it is hard to quickly remember all the lessons about it. To address this challenge, I used the lecture notes and materials from our Tutors, did research about the topics, and watched YouTube tutorials.

References

DatabaseTown. (n.d.) Relational Vs Non Relational Database (Key Differences). [Online] [Accessed on 21st of February 2024]

https://databasetown.com/relational-vs-non-relational-database/

Kalganova, Tatiana & Dzalbs, Ivars. (2019,December 14). *Supply Chain Logistics Problem Dataset*.Brunel University London. *https://brunel.figshare.com/articles/dataset/Supply_Chain_Logistics_Problem_Dataset/7558679*

Pawlan, David. (n.d.). Relational vs. Non-Relational Database: Pros & Cons. Aloa Blogger. https://aloa.co/blog/relational-vs-non-relational-database-pros-cons#:~:text=A%20relational%20database%20structures%20data,ability%20to%20store%20complex%20structures.

Simplilearn.(2023, June 19). *What is a Database? Everything You Need to Know*.[Online] [Online] [Accessed on 21st of February 2024] https://www.simplilearn.com/tutorials/dbms-tutorial/what-is-a-database