University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: Tuesday, October 15th, 2024

Group Number: 32

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Martin Janzen	44630812	g4i8s	martinezjan2014@gmail.com
Armen Henry Bagdasarov	20302303	c7v1f	armenhb@student.ubc.ca
Borhan Rahmani Nejad	43553015	t3p1c	borhan200153@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

2. A brief (~2-3 sentences) summary of your project.
Many of your TAs are managing multiple projects so this will help them remember details about your project.

Our project is a port management software. This system will help port managers handle the logistics of incoming ships and cargo, creating an efficient supply chain with easy retrieval, customs processes, and oversight. Our database will contain many port-related entities such as trade agreements, cargo, warehouses, and more.

3. The ER diagram you are basing your item #3 (below) on.
This ER diagram may be the same as your milestone 1 submission or it might be different. If you have made changes from the version submitted in milestone 1, attach a note indicating what changes have been made and why.

Our project mentor has suggested no significant changes, but below are the changes we decided to implement for clarity of our diagram and fluidity of our system as a whole.

- Removed "Flag" attribute from all entities
- Changed "Begins At" relationship so that there are many Shipping Routes to one Country
- While our ER Diagram has some attributes with names such as "Name" and "# of Docked Ships", for the rest of the document we use terms such as "CountryName" and "DockedShips" for clarity.

PLEASE SEE LAST PAGE FOR THE UPDATED ER DIAGRAM! WE COULDN'T FIT IT ON TO THIS PAGE!

4. The schema derived from your ER diagram (above).
For the translation of the ER diagram to the relational model, follow the same instructions as in your lectures. The process should be reasonably straightforward. For each table:
a. List the table definition (e.g., Table1(attr1: domain1, attr2: domain2, ...)). Make sure to include the domains for each attribute.
b. Specify the primary key (PK), candidate key, (CK) foreign keys (FK), and other constraints (e.g., not null, unique, etc.) that the table must maintain.

Entity Tables:
Company(CEO: varchar, Name: varchar, Industry: varchar, YearlyRevenue: float, CountryName: varchar)

- PK = CEO + Name
- CK = CEO + Name
- FK = CountryName (references Country)

Tariff (TradeAgreement: varchar, AffectedGoods: varchar, ForeignName: varchar, HomeName: varchar, EnactmentDate: date, TariffRate: float)

- PK = TradeAgreement
- CK = TradeAgreement
- FK = ForeignName, HomeName

Country(Name: varchar, Population: Integer, GDP: float, Government: varchar)

- PK = Name
- CK = Name
- FK = none

Ship (Owner: varchar, ShipName: varchar, ShipSize: float, Capacity: float, ShippingRouteName: varchar, DockedAtPortAddress: varchar)

- PK = Owner+Name
- CK = Owner+Name
- FK = ShippingRouteName (references ShippingRoute), DockedAtPortAddress (references Port)

Shipment Container (GoodType: varchar, ContainerSize: float, Weight: float, TrackingNumber: integer (not null, unique), ShipOwner: varchar (not null), ShipName: varchar, PortAddress: varchar (unique), WarehouseSection: integer, TradeAgreement: varchar, CompanyName: varchar, CompanyCEO: varchar)

- PK = TrackingNumber
- CK = TrackingNumber
- FK = Owner+Name (references Ship), PortAddress+WarehouseSection (references Warehouse), TradeAgreement (references Tariff), CompanyName+CEO (references Company)

Warehouse (PortAddress: varchar (unique, not null), Section: integer, NumContainers: integer, Capacity: integer)

- PK = PortAddress+Section
- CK = PortAddress+Section
- FK = PortAddress (references Port)

Port(Address: varchar (unique, not null), Name: varchar, NumWorkers: Integer, NumDockedShips: Integer, CountryName: varchar)

- PK = Address
- CK = Address
- FK = CountryName (references Country)

ForeignCountry(Name: varchar (unique, not null), Population: Integer, GDP: float, Government: varchar, DockingFee: float)

- PK = Name
- CK = Name
- FK = Name (references Country)

HomeCountry(Name: varchar (unique, not null), Population: Integer, GDP: float, Government: varchar, DockingFee: float) - not null, unique

- PK = Name
- CK = Name
- FK = Name (references Country)

ShippingRoute(ShippingRouteName: varchar (unique, not null), Length: float, AnnualVolumeOfGoods: float, OriginCountryName: varchar (not null), TerminalPortAddress: varchar (not null)

- PK = Name
- CK = Name
- FK = OriginCountryName (references Country), TerminalPortAddress (references Port)

Note:

Since we happen to have no many-to-many relationships in our system, we do not need any separate tables for specific relationships. All information can be captured by adding attributes from existing relationships to entity tables. This makes for a cleaner, more concise implementation of our system.

5. Functional Dependencies (FDs)

a. Identify the functional dependencies in your relations, including the ones involving all candidate keys (including the primary key). PKs and CKs are considered functional dependencies and should be included in the list of FDs. You do not need to include trivial FDs such as A → A. Note: In your list of FDs, there must be some kind of valid FD other than those identified by a PK or CK. If you observe that no relations have FDs other than the PK and CK(s), then you will have to intentionally add some (meaningful) attributes to show valid FDs. We want you to get a good normalization exercise. Your design must go through a normalization process. You do not need to have a non-PK/CK FD for each relation but be reasonable. If your TA feels that some non-PK/CK FDs have been omitted, your grade will be adjusted accordingly.

Company
CompanyCEO, CompanyName → Industry, YearlyRevenue, CountryName

Tariff
TradeAgreement → AffectedGoods, ForeignName, HomeName, EnactmentDate, TariffRate
EnactmentDate, TariffRate, HomeName, ForeignName → TradeAgreement

Country
CountryName → Population, GDP, Government

Ship
Owner, ShipName → ShipSize, Capacity, ShippingRouteName, DockedAtPortAddress
ShipSize → Capacity

Shipment Container
TrackingNumber → ContainerSize, Weight, GoodType
ShipOwner, ShipName → PortAddress, WarehouseSection
TrackingNumber, PortAddress → TradeAgreement, CompanyName, CompanyCEO

Warehouse
PortAddress, Section → NumContainters, Capacity

Port
Address → NumWorkers, Name, NumDockedShips, CountryName

ForeignCountry
ForeignName → Population, GDP, Government, DockingFee

HomeCountry
HomeName → Population, GDP, Government, TariffRate

ShippingRoute
Name → Length, AnnualVolumeOfGoods, OriginCountryName, TerminalPortAddress
OriginCountryName, TerminalPortAddress → Name, Length

6. Normalization

a. Normalize each of your tables to be in 3NF or BCNF. Give the list of tables, their primary keys, their candidate keys, and their foreign keys after normalization. You should show the steps taken for the decomposition in a manner similar to that done in class. Should there be errors, and no work is shown, no partial credit can be awarded without steps shown. The format should be the same as Step 3, with tables listed similar to Table1(attr1:domain1, attr2:domain2, ...). ALL Tables must be listed, not only the ones post normalization.

Company**:** This table is already in BCNF because the primary key (CEO+Name) determines all other attributes. The normalized table is below.
FDs:

- CompanyCEO, CompanyName → Industry, YearlyRevenue, CountryName

Company(CEO: varchar, Name: varchar, Industry: varchar, YearlyRevenue: float, CountryName: varchar)

- PK = CEO + Name
- CK = CEO + Name
- FK = CountryName (references Country)

Tariff: Although this table is in 3NF, it violates BCNF. In the first FD, TradeAgreement is a primary key so it is by definition also a superkey. The second FD is problematic because X (the four attributes on the left) is not a superkey in the relation X→b.
FDs:

- TradeAgreement → AffectedGoods, ForeignName, HomeName, EnactmentDate, TariffRate
- EnactmentDate, TariffRate, HomeName, ForeignName → TradeAgreement

Decomposition:
{TradeAgreement+} = {TradeAgreement, AffectedGoods, ForeignName, HomeName, EnactmentDate, TariffRate}
{EnactmentDate, TariffRate, HomeName, ForeignName+} = {TradeAgreement, AffectedGoods, ForeignName, HomeName, EnactmentDate, TariffRate}

Decompose the Tariff relation on EnactmentDate, TariffRate, HomeName, ForeignName → TradeAgreement

LHS	Middle	RHS
AffectedGoods	EnactmentDate, TariffRate, HomeName, ForeignName	TradeAgreement

Tariff1(EnactmentDate: date, TariffRate: float, HomeName: varchar, ForeignName: varchar, TradeAgreement: varchar)

- PK: TradeAgreement
- CK: TradeAgreement
- FK: ForeignName (references Foreign Country), HomeName (references Home Country)

Tariff2(EnactmentDate: date, TariffRate: float, HomeName: varchar, ForeignName: varchar, AffectedGoods: varchar)

- PK: EnactmentDate, TariffRate, HomeName, ForeignName
- CK: EnactmentDate, TariffRate, HomeName, ForeignName
- FK: ForeignName (references Foreign Country), HomeName (references Home Country)

Country: This table is already in BCNF because the only FD has the key in the X side of the X→b relationship. Thus, the Country table remains unchanged.
FDs:

- CountryName → Population, GDP, Government

Country(Name: varchar, Population: Integer, GDP: float, Government: varchar, portAddress: varchar)
PK = Name
CK = Name
FK = none

Ship: This table violates BCNF. The first functional dependency is fine, since the X in X→b is a superkey for the relation. The second FD violates BCNF because ShipSize is not a superkey.
FDs:

- Owner, ShipName → ShipSize, Capacity, ShippingRouteName, DockedAtPortAddress
- ShipSize → Capacity

Decomposition:

Owner, ShipName+ {Owner, ShipName, ShipSize, Capacity, ShippingRouteName, DockedAtPortAddress}
ShipSize+ {ShipSize, Capacity}

LHS	Middle	RHS
Owner, ShipName, ShippingRouteName, DockedAtPortAddress	ShipSize	Capacity

Ship1(Owner: varchar, ShipName, ShippingRouteName: varchar, DockedAtPortAddress: varchar)

- PK: Owner, ShipName
- CK: Owner, ShipName
- FK: ShippingRouteName (references ShippingRoute), DockedAtPortAddress (references Port)

Ship2(ShipSize, Capacity) ← two entity relationship, therefore BCNF
- PK: ShipSize
- CK: ShipSize
- FK: None

Shipment Container: This table violates BCNF because ShipOwner and ShipName is not a superkey for the relation. We can note that this relation would also violate 3NF because PortAddress and WarehouseSection together are not part of any key.
Key: TrackingNumber
FDs:
- TrackingNumber → ContainerSize, Weight, GoodType, TradeAgreement, CompanyName, CompanyCEO
- ShipOwner, ShipName → PortAddress, WarehouseSection

Decompose Shipment Container on FD: ShipOwner, ShipName → PortAddress, WarehouseSection

LHS	Middle	RHS
GoodType, ContainerSize, Weight, TrackingNumber, TradeAgreement, CompanyName, CompanyCEO	ShipOwner, ShipName	PortAddress, WarehouseSection

ShipmentContainer1(ShipOwner: varchar (not null), ShipName: varchar, PortAddress: varchar (unique), WarehouseSection: integer)
- PK: ShipOwner+ShipName
- CK: ShipOwner+ShipName
- FK: ShipOwner (references Ship), ShipName (references Ship), PortAddress (references Port), WarehouseSection (references Warehouse)

ShipmentContainer2: (ShipOwner: varchar (not null), ShipName: varchar, GoodType: varchar, ContainerSize: float, Weight: float, TrackingNumber: integer, TradeAgreement: varchar, CompanyName: varchar, CompanyCEO: varchar)
- PK: TrackingNumber
- CK: TrackingNumber
- FK: ShipOwner (references Ship), ShipName (references Ship), PortAddress (references Port), WarehouseSection (references Warehouse)

Warehouse: This table does not violate BCNF or 3NF. PortAddress and Section is a key for the relation, and hence is a superkey. The normalized Warehouse table will remain unchanged.
FDs:
- PortAddress, Section → NumContainters, Capacity

Warehouse (PortAddress: varchar (unique, not null), Section: integer, NumContainers: integer, Capacity: integer)

- PK = PortAddress+Section
- CK = PortAddress+Section
- FK = PortAddress (references Port)

Port: This table does not violate BCNF or 3NF. Address is a key for the relation, and hence is a superkey. The normalized Port table will remain unchanged.
FDs:

- Address → NumWorkers, Name, NumDockedShips, CountryName

Port(Address: varchar (unique, not null), Name: varchar, NumWorkers: Integer, NumDockedShips: Integer, CountryName: varchar)

- PK = Address
- CK = Address
- FK = CountryName (references Country)

ForeignCountry: This table is already in BCNF because the only FD has the key (ForeignName) in the X side of the X→b relationship. Thus, the normalized ForeignCountry table remains unchanged.
FDs:

- ForeignName → Population, GDP, Government, DockingFee

ForeignCountry(Name: varchar (unique, not null), Population: Integer, GDP: float, Government: varchar, DockingFee: float)

- PK = Name
- CK = Name
- FK = Name (references Country)

HomeCountry: This table is already in BCNF because the only FD has the key (ForeignName) in the X side of the X→b relationship. Thus, the HomeCountry table remains unchanged.
FDs:

- HomeName → Population, GDP, Government, TariffRate

HomeCountry(Name: varchar (unique, not null), Population: Integer, GDP: float, Government: varchar, DockingFee: float) - not null, unique

- PK = Name
- CK = Name
- FK = Name (references Country)

ShippingRoute
This violates BCNF as in the FD OriginCountryName, TerminalPortAddress → Length, the left hand side does not contain a key and hence is not a superkey.
FDs:

- ShippingRouteName → Length, AnnualVolumeOfGoods, OriginCountryName, TerminalPortAddress
- OriginCountryName, TerminalPortAddress → ShippingRouteName, Length

Decomposition:
ShippingRouteName+ {ShippingRouteName, Length, AnnualVolumeOfGoods, OriginCountryName, TerminalPortAddress}
OriginCountryName, TerminalPortAddress+ {ShippingRouteName, Length}

Decomposing Original Relation on FD: OriginCountryName, TerminalPortAddress → ShippingRouteName, Length

LHS	Middle	RHS
AnnualVolumeOfGoods	OriginCountryName, TerminalPortAddress	ShippingRouteName, Length

ShippingRoute1(AnnualVolumeOfGoods: float, OriginCountryName: varchar (not null), TerminalPortAddress: varchar (not null))
PK - OriginCountry, TerminalPortAddress
CK - OriginCountry, TerminalPortAddress
FK - OriginCountryName (references Country), TerminalPortAddress (references Port)

ShippingRoute2(OriginCountryName: varchar (not null), TerminalPortAddress: varchar (not null), ShippingRouteName: varchar (unique, not null), Length: float)
PK - ShippingRouteName
CK - ShippingRouteName
FK - OriginCountryName (references Country), TerminalPortAddress (references Port)

As we have OriginCountryName and TerminalPortAddress in both relations, and the functional dependency [OriginCountryName, TerminalPortAddress → Name, Length] allows us to determine ShippingRouteName, these attributes (OriginCountryName and TerminalPortAddress) form a key in the first relation. Thus they act as superkeys and satisfy BCNF.

7. The SQL DDL statements required to create all the tables from item #6.

The statements should use the appropriate foreign keys, primary keys, UNIQUE constraints, etc. Unless you know that you will always have exactly x characters for a given character, it is better to use the VARCHAR data type as opposed to a CHAR(Y). For example, UBC courses always use four characters to represent which department offers a course. In that case, you will want to use CHAR(4) for the department attribute in your SQL DDL statement. If you are trying to represent the name of a UBC course, you will want to use VARCHAR as the number of characters in a course name can vary greatly.

```
CREATE TABLE Country (
  Name          VARCHAR UNIQUE,
  Population    INT,
  Government    VARCHAR,
  PortAddress   VARCHAR NOT NULL,
```

```
  PRIMARY KEY (Name),
  FOREIGN KEY (PortAddress) REFERENCES
                    Port(Address)
                    ON DELETE CASCADE
                    ON UPDATE CASCADE);

CREATE TABLE Warehouse (
  PortAddress      VARCHAR NOT NULL,
  Section          INT,
  NumContainers INT,
  Capacity          INT,
  PRIMARY KEY (PortAddress, Section),
  UNIQUE (PortAddress),
  FOREIGN KEY (PortAddress) REFERENCES
                    Port(Address)
                    ON DELETE CASCADE
                    ON UPDATE CASCADE);

CREATE TABLE Port (
  Address          VARCHAR NOT NULL,
  NumWorkers    INT,
  DockedShips    INT,
  CountryName   VARCHAR UNIQUE,
  PRIMARY KEY (Address),
  FOREIGN KEY (CountryName) REFERENCES
                    Country(Name)
                    ON DELETE CASCADE
                    ON UPDATE CASCADE);

CREATE TABLE  Tariff1 (
  TradeAgreement VARCHAR NOT NULL,
  TariffRate         FLOAT,
  HomeName       VARCHAR,
  ForeignName    VARCHAR,
  EnactmentDate  DATE,
  PRIMARY KEY (TradeAgreement),
  FOREIGN KEY (ForeignName) REFERENCES
                    ForeignCountry(Name)
                    ON DELETE CASCADE
                    ON UPDATE CASCADE,
  FOREIGN KEY (HomeName) REFERENCES
                    HomeCountry(Name)
                    ON DELETE CASCADE
                    ON UPDATE CASCADE);
```

```
CREATE TABLE  Tariff2 (
   TariffRate          FLOAT,
   AffectedGoods   VARCHAR,
   HomeName        VARCHAR,
   ForeignName     VARCHAR,
   EnactmentDate   DATE,
   PRIMARY KEY (EnactmentDate, TariffRate, HomeName, ForeignName),
   FOREIGN KEY (ForeignName) REFERENCES
                        ForeignCountry(Name)
                        ON DELETE CASCADE
                        ON UPDATE CASCADE,
   FOREIGN KEY (HomeName) REFERENCES
                         HomeCountry(Name)
                        ON DELETE CASCADE
                        ON UPDATE CASCADE);

CREATE TABLE Shipping Route1 (
   AnnualVolumeOfGoods FLOAT
   OriginCountryName       VARCHAR NOT NULL,
   TerminalPortAddress      VARCHAR NOT NULL,
   PRIMARY KEY (OriginCountryName, TerminalPortAddress),
   FOREIGN KEY (OriginCountryName) REFERENCES
                       ForeignCountry(Name)
                       ON DELETE CASCADE
                       ON UPDATE CASCADE,
   FOREIGN KEY (TerminalPortAddress) REFERENCES
                        Port(Address)
                       ON DELETE CASCADE
                       ON UPDATE CASCADE);

CREATE TABLE Shipping Route2 (
   Name                        VARCHAR UNIQUE NOT NULL,
   Length                      FLOAT,
   OriginCountryName      VARCHAR NOT NULL,
   TerminalPortAddress     VARCHAR NOT NULL,
   PRIMARY KEY (Name),
   FOREIGN KEY (OriginCountryName) REFERENCES
                               ForeignCountry(Name)
                               ON DELETE CASCADE
                               ON UPDATE CASCADE,
   FOREIGN KEY (TerminalPortAddress) REFERENCES
                               Port(Address),
                              ON DELETE CASCADE
```

```
                    ON UPDATE CASCADE);



CREATE TABLE Ship1(
        Owner VARCHAR NOT NULL,
        ShipName VARCHAR NOT NULL,
        ShippingRouteName VARCHAR,
        DockedAtPortAddress VARCHAR,
        PRIMARY KEY (Owner, ShipName),
        FOREIGN KEY (ShippingRouteName) REFERENCES ShippingRoute(ShippingRouteName),
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY (DockedAtPortAddress) REFERENCES Port(Address)
                ON DELETE CASCADE
                ON UPDATE CASCADE);

CREATE TABLE Ship2(
        ShipSize FLOAT NOT NULL,
        Capacity FLOAT,
        PRIMARY KEY (ShipSize),
                ON DELETE CASCADE
                ON UPDATE CASCADE);

CREATE TABLE HomeCountry(
        Name VARCHAR NOT NULL UNIQUE,
        Population INTEGER,
        GDP FLOAT,
        Government VARCHAR,
        DockingFee FLOAT,
        PRIMARY KEY (Name),
        FOREIGN KEY (Name) REFERENCES Country(Name)
                ON DELETE CASCADE
                ON UPDATE CASCADE);

CREATE TABLE ForeignCountry(
        Name VARCHAR NOT NULL UNIQUE,
        Population INTEGER,
        GDP FLOAT,
        Government VARCHAR,
        DockingFee FLOAT,
        PRIMARY KEY (Name),
        FOREIGN KEY (Name) REFERENCES Country(Name)
```

```
                ON DELETE CASCADE
                ON UPDATE CASCADE);

CREATE TABLE ShipmentContainer1(
        ShipOwner VARCHAR NOT NULL,
        ShipName VARCHAR NOT NULL,
        PortAddress VARCHAR UNIQUE,
        WarehouseSection INTEGER,
        PRIMARY KEY (ShipOwner, ShipName),
        FOREIGN KEY (ShipOwner, ShipName) REFERENCES Ship(Owner, ShipName),
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY (PortAddress) REFERENCES Port(Address)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY (WarehouseSection) REFERENCES Warehouse(WarehouseSection)
                ON DELETE CASCADE
                ON UPDATE CASCADE);

CREATE TABLE ShipmentContainer2(
        ShipOwner VARCHAR,
        ShipName VARCHAR,
        GoodType VARCHAR,
        ContainerSize FLOAT,
        Weight FLOAT,
        TrackingNumber INTEGER NOT NULL,
        TradeAgreement VARCHAR,
        CompanyName VARCHAR,
        CompanyCEO VARCHAR,
        PRIMARY KEY (TrackingNumber),
        FOREIGN KEY (ShipOwner, ShipName) REFERENCES
                        Ship(Owner, ShipName)
                        ON DELETE CASCADE
                        ON UPDATE CASCADE,
        FOREIGN KEY (TrackingNumber) REFERENCES
                        ShipmentContainer1(TrackingNumber),
                        ON DELETE CASCADE
                        ON UPDATE CASCADE,
        FOREIGN KEY (TradeAgreement) REFERENCES
                        Tariff(TradeAgreement),
                        ON DELETE CASCADE
                        ON UPDATE CASCADE,
        FOREIGN KEY (CompanyName, CompanyCEO) REFERENCES
                        Company(Name, CEO)
```

```
                ON DELETE CASCADE
                ON UPDATE CASCADE);

CREATE TABLE Company(
        CEO VARCHAR NOT NULL,
        Name VARCHAR NOT NULL,
        Industry VARCHAR,
        YearlyRevenue FLOAT,
        CountryName VARCHAR,
        PRIMARY KEY (CEO, Name),
        FOREIGN KEY (CountryName) REFERENCES Country(Name)
                ON DELETE CASCADE
                ON UPDATE CASCADE);
```

8. INSERT statements to populate each table with at least 5 tuples.

You will likely want to have more than 5 tuples so that you can have meaningful queries later. Note: Be consistent with the names used in your ER diagram, schema, and FDs. Make a note if the name has been intentionally changed.

COUNTRY:

```
INSERT INTO Country
        (Name, Population, Government, GDP, PortAddress)
VALUES
        ("Canada", 38930000, "Liberal Party - Justin Trudeau", "999 Canada Pl, Vancouver, BC V6C 3T4"),
        ("United States", 333300000, "Democratic Party - Joe Biden", "Signal St, San Pedro, CA 90731, United States"),
        ("China", 1412000000, "Chinese Communist Party - Xi Jinping", "Shengsi County, Zhoushan, China, 202461"),
        ("Japan", 125100000, "Liberal Democratic Party - Shigeru Ishiba", "4-chōme-8 Ariake, Koto City, Tokyo 135-0063, Japan"),
        ("Netherlands", 177000000, "Independant - Dick Schoof", "Wilhelminakade 909, 3072 AP Rotterdam, Netherlands");
```

WAREHOUSE:

```
INSERT INTO Warehouse
        (Section, NumContainers, Capacity, PortAddress)
VALUES
        (1, 90, 100, "999 Canada Pl, Vancouver, BC V6C 3T4"),
        (2, 200, 300, "999 Canada Pl, Vancouver, BC V6C 3T4"),
        (3, 200, 200, "999 Canada Pl, Vancouver, BC V6C 3T4"),
        (1, 631, 1000, "Shengsi County, Zhoushan, China, 202461"),
        (9, 10, 220, "Wilhelminakade 909, 3072 AP Rotterdam, Netherlands");
```

PORT:

```
INSERT INTO Port
        (PortAddress, NumWorkers, DockedShips, CountryName)
VALUES
        ("999 Canada Pl, Vancouver, BC V6C 3T4", 523, 53, "Canada"),
        ("Shengsi County, Zhoushan, China, 202461", 13546, 123, "China"),
        ("Wilhelminakade 909, 3072 AP Rotterdam, Netherlands", 1270, 225, "Netherlands"),
        ("Signal St, San Pedro, CA 90731, United States", 1230, 67, "United States"),
        ("4-chōme-8 Ariake, Koto City, Tokyo 135-0063, Japan", 30000, 44, "Japan");
```

TARIFF1:

```
INSERT INTO Tariff 1
        (TradeAgreement, TariffRate, HomeName, ForeignName, EnactmentDate)
VALUES
        ("China-USA Agreement", 12, "China", "USA", 2024-01-15),
        ("Canada-China Agreement", 9, "Canada", "China", 2020-10-25),
        ("Canada-Netherlands Agreement", 8, "Canada", "Netherlands", 2008-06-12),
        ("Canada-USA Agreement", 5, "Canada", "USA", 2020-01-30),
        ("Canada-Japan Agreement", 6, "Canada", "Japan", 1998-04-09);
```

TARIFF2:

```
INSERT INTO Tariff 2
        (TariffRate, AffectedGoods, HomeName, ForeignName, EnactmentDate)
VALUES
        (12, "Solar Panels",  "China", "USA", 2024-01-15),
        (9, "Lumber",  "Canada", "China", 2020-10-25),
        (8, "Maple Syrup", "Canada", "Netherlands", 2008-06-12),
        (5, "Oil",  "Canada", "USA", 2020-01-30),
        (6, "Wheat",  "Canada", "Japan", 1998-04-09);
```

SHIPPING ROUTE1:

```
INSERT INTO Shipping Route 1
        (AnnualVolumeOfGoods, OriginCountryName, TerminalPortAddress)
VALUES
        (12000,  "Canada", "Signal St, San Pedro, CA 90731, United States"),
        (45000,  "United States", "999 Canada Pl, Vancouver, BC V6C 3T4"),
        (80000,  "China", "999 Canada Pl, Vancouver, BC V6C 3T4"),
        (20000,  "Netherlands", "999 Canada Pl, Vancouver, BC V6C 3T4"),
        (60000,  "Japan", "999 Canada Pl, Vancouver, BC V6C 3T4");
```

SHIPPING ROUTE2:

```
INSERT INTO Shipping Route 2
        (Name, Length, OriginCountryName, TerminalPortAddress)
```

VALUES
("Great Circle", 4078, "Japan", "999 Canada Pl, Vancouver, BC V6C 3T4"),
("PANZ Seattle Loop", 1319 "United States", "999 Canada Pl, Vancouver, BC V6C 3T4"),
("Trans-Pacific Route", 7838, "China", "999 Canada Pl, Vancouver, BC V6C 3T4"),
("Rotterdam-Vancouver", 11564, "Netherlands", "999 Canada Pl, Vancouver, BC V6C 3T4"),
("PANZ Seattle Loop", 1319, "Canada", "Signal St, San Pedro, CA 90731, United States");

SHIP1:

INSERT INTO Ship1
(Owner, ShipName, ShippingRouteName, DockedAtPortAddress)
VALUES
('Maersk', 'Ocean Breeze', 'Great Circle', '999 Canada Pl, Vancouver, BC V6C 3T4'),
('Mediterranean Shipping Company', 'Seawolf', 'PANZ Seattle Loop', 'Shengsi County, Zhoushan, China, 202461'),
('Atlantic Trade', 'Blue Horizon', 'Trans-Pacific Route'', 'Wilhelminakade 909, 3072 AP Rotterdam, Netherlands'),
('Pacific Vessels', 'Tidal Wave', 'Rotterdam-Vancouver', Signal 'St, San Pedro, CA 90731, United States'),
('Maritime Enterprises', 'Northern Star', 'PANZ Seattle Loop'', '4-chōme-8 Ariake, Koto City, Tokyo 135-0063, Japan');

SHIP2:

INSERT INTO Ship2
(ShipSize, Capacity)
VALUES
(100.5, 500.0),
(150.75, 800.0),
(200.0, 1200.0),
(175.4, 950.0),
(225.6, 1400.0);

HOMECOUNTRY:

INSERT INTO HomeCountry
(Name, Population, GDP, Government, DockingFee)
VALUES
('Canada', 38000000, 1643.5, 'Liberal Party - Justin Trudeau', 500.0),
('USA', 331000000, 21137.0, 'Democratic Party - Joe Biden', 600.0),
('China', 83000000, 4381.0, 'Chinese Communist Party - Xi Jinping', 550.0),
('Japan', 125800000, 5150.0, 'Liberal Democratic Party - Shigeru Ishiba', 580.0),
('Netherlands', 25600000, 1390.0, 'Independant - Dick Schoof', 470.0);

FOREIGNCOUNTRY:

INSERT INTO ForeignCountry
(Name, Population, GDP, Government, DockingFee)
VALUES

('Canada', 38000000, 1643.5, ‘Liberal Party - Justin Trudeau’, 500.0),
('Russia', 146000000, 1680.0, ‘United Russia - Vladimir Putin’, 620.0),
('India', 1390000000, 2875.0, ‘Bharatiya Janata Party - Narendra Modi’, 580.0),
('Brazil', 213000000, 1505.0, ‘Workers Party - Luiz Inácio Lula da Silva’, 490.0),
('UK', 67000000, 3031.0, ‘Conservative Party - Rishi Sunak’, 550.0);

SHIPMENTCONTAINER1:

```
INSERT INTO ShipmentContainer1
    (ShipOwner, ShipName, PortAddress, WarehouseSection)
VALUES
    ('John Shipping Co.', 'Ocean Breeze', '999 Canada Pl, Vancouver, BC V6C 3T4', 1),
    ('Global Shipping Ltd.', 'Seawolf', ‘Shengsi County, Zhoushan, China, 202461’, 2),
    ('Atlantic Trade', 'Blue Horizon', ‘Wilhelminakade 909, 3072 AP Rotterdam, Netherlands’, 3),
    ('Pacific Vessels', 'Tidal Wave', ‘Signal St, San Pedro, CA 90731, United States’, 4),
    ('Maritime Enterprises', 'Northern Star', ‘4-chōme-8 Ariake, Koto City, Tokyo 135-0063, Japan’,
    9);
```

SHIPMENTCONTAINER2:

```
INSERT INTO ShipmentContainer2
    (ShipOwner, ShipName, GoodType, ContainerSize, Weight, TrackingNumber, TradeAgreement,
    CompanyName, CompanyCEO)
VALUES
    ('John Shipping Co.', 'Ocean Breeze', 'Electronics', 45.0, 300.0, 1001, ‘China-USA Agreement’,
    'TechCo', 'Alice Johnson'),
    ('Global Shipping Ltd.', 'Seawolf', 'Automobiles', 50.0, 450.0, 1002, ‘Canada-China Agreement’,
    'AutoInc', 'Bob Smith'),
    ('Atlantic Trade', 'Blue Horizon', 'Textiles', 30.0, 200.0, 1003, ‘Canada-Netherlands Agreement’,
    'TextileCorp', 'Charlie Williams'),
    ('Pacific Vessels', 'Tidal Wave', 'Furniture', 60.0, 500.0, 1004, ‘Canada-USA Agreement’,
    'FurnitureMakers', 'David Brown'),
    ('Maritime Enterprises', 'Northern Star', 'Machinery', 55.0, 400.0, 1005, ‘Canada-Japan
    Agreement’, 'HeavyMachines', 'Eve Davis');
```

COMPANY:

```
INSERT INTO Company
    (CEO, Name, Industry, YearlyRevenue, CountryName)
VALUES
    ('Alice Johnson', 'TechCo', 'Technology', 5000.0, 'Canada'),
    ('Bob Smith', 'AutoInc', 'Automobile', 3200.0, 'USA'),
    ('Charlie Williams', 'TextileCorp', 'Textiles', 1800.0, China),
    ('David Brown', 'FurnitureMakers', 'Furniture', 2500.0, Netherlands),
    ('Eve Davis', 'HeavyMachines', 'Machinery', 3500.0, 'Japan');
```

Updated ER Diagram: