University of British Columbia, Vancouver

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2
Date: March 1st, 2024

Group Number: 38

Name	Student Number	CS Alias	Email Address
Esther Hsueh	55621346	e9x8k	snow113@students.cs.ubc.ca
Reuben Sinha	48327316	x4y2b	reuben.sinha@gmail.com
Rhi Ann Ng	10455657	h5w2b	12rhiann@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.

Project Summary

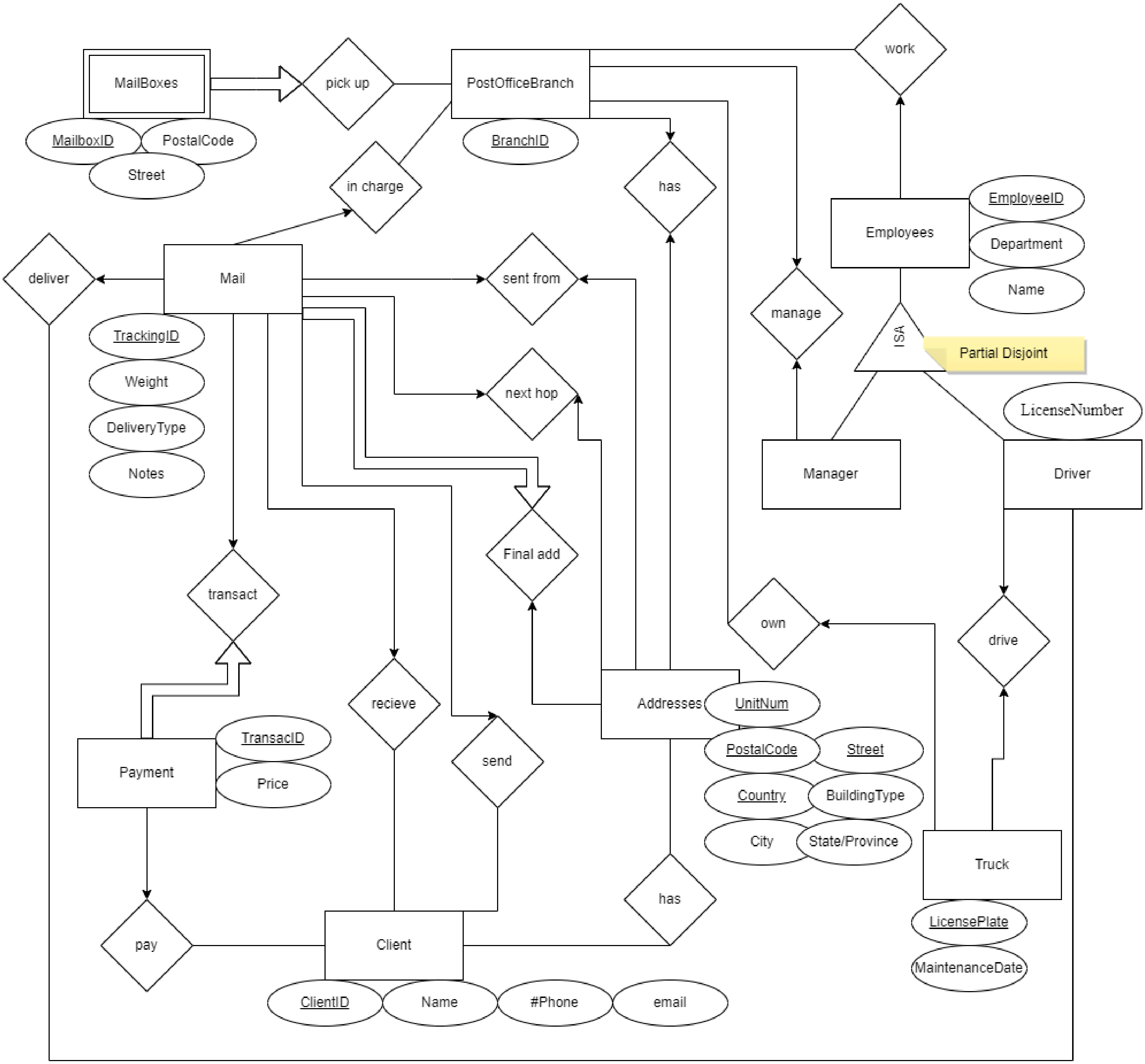
Our project is to make an application for a Post Office Mail Management System. The System is designed to enhance the efficiency and reliability of mail delivery services by providing real-time tracking and monitoring capabilities for various types of mail items. The system serves as a comprehensive platform for postal workers, clients, and administrative staff to monitor the progress, location, and the value of mail shipment throughout the delivery process.

The entities include Mail, PostOfficeBranch, Employees, Manager, Driver, Address, Clients, MailBoxes, Payment, Truck.

ER Diagram

Changes made to ERD from milestone 1

- Add ISA constraint - partial and disjoint
- Add an attribute, LicenseNumber, to Driver
- Add attributes to Address
- Add an attribute, Street, to MailBoxes
- Thick Lines: Payment to transact; Mailbox to Pick up; Mail to final add.


MailBoxes
pick up
PostOfficeBranch
work
MailboxID
PostalCode
Street
BranchID
in charge
has
Employees
EmployeeID
Department
Name
deliver
Mail
sent from
manage
ISA
Partial Disjoint
TrackingID
Weight
DeliveryType
Notes
next hop
LicenseNumber
Manager
Driver
Final add
transact
own
drive
recieve
Addresses
UnitNum
send
Payment
TransacID
Price
PostalCode
Street
Country
BuildingType
City
State/Province
Truck
has
pay
Client
LicensePlate
MaintenanceDate
ClientID
Name
#Phone
email

Schema

Mail (TrackingID : int, **CurrentBranch**: int, **SenderUnitNum:** varchar, **SenderStreet:** varchar, **SenderPostCode:** varchar, **SenderCountry**: varchar, **NextUnitNum**: varchar, **NextStreet**: varchar, **NextPostCode:** varchar, **NextCountry**: varchar, **FinalUnitNum**: varchar, **FinalStreet**: varchar, **FinalPostcode**: varchar, **FinalCountry**: varchar, **Sender**: int, **Receiver**: int, Weight: float, DeliveryType: varchar, Notes: varchar)
NOT NULL: TrackingID, CurrentBranch, NextUnitNum, NextStreet, NextPostCode, FinalUnitNum, FinalStreet, FinalPostCode, Receiver, DeliveryType
UNIQUE: TrackingID
CANDIDATE KEYS: TrackingID

PostOfficeBranch (BranchID: int, **UnitNum**: varchar, **Street**: varchar, **PostalCode**: varchar, **BranchManager**: int)
NOT NULL: BranchID, UnitNum, Street, PostalCode, BranchManager
UNIQUE: BranchID, (UnitNum + Street + PostalCode), BranchManager
CANDIDATE KEYS: BranchID, (UnitNum + Street + PostalCode), BranchManager

Employees(EmployeeID: int, Name : varchar, **WorksAtBranch**: int, Department: char)
NOT NULL: EmployeeID, Name, WorksAtBranch
UNIQUE: EmployeeID
CANDIDATE KEYS: EmployeeID

Manager(EmployeeID: int, **ManagesBranch**: int)
NOT NULL: EmployeeID, ManagesBranch
UNIQUE: EmployeeID, ManagesBranch
CANDIDATE KEYS: EmployeeID, ManagesBranch

Driver(EmployeeID: int, LicenseNumber: int)
NOT NULL: EmployeeID, LicenseNumber
UNIQUE: EmployeeID, LicenseNumber

CANDIDATE KEYS: EmployeeID, LicenseNumber

Address (UnitNum: varchar, Street: varchar, City: varchar, State/Province: varchar, PostalCode: varchar, Country: varchar, BuildingType: char, **Resident:** int, **LocalPostOffice**: int)
NOT NULL: UnitNum, Street, City, State/Province, PostalCode, Country, LocalPostOffice
UNIQUE: (UnitNum + Street + PostalCode + Country)
CANDIDATE KEYS: (UnitNum + Street + PostalCode + Country)

Clients (ClientID: int, Name : varchar, **ClientUnitNum**: varchar, **ClientStreet**: varchar, **ClientPostCode**: varchar, **ClientCountry**: varchar, PhoneNumber: int, Email: varchar)
NOT NULL: ClientID, Name, (Email, PhoneNumber)
UNIQUE: ClientID
CANDIDATE KEYS: ClientID

MailBoxes (MailBoxID : int, **PostOfficeBranch** : int, Street: varchar, PostalCode: varchar)
NOT NULL: MailBoxID, PostOfficeBranch, Street, PostalCode
UNIQUE: MailBoxID
CANDIDATE KEYS: MailBoxID

Payment (TransactionID: int, **Payer**: int, **Mail:** int, Price: float)
NOT NULL: TransactionID, Payer, Mail, Price
UNIQUE: TransactionID
CANDIDATE KEYS: TransactionID

Truck (LicensePlate: varchar, **Branch**: int, MaintenanceDate: datetime)
NOT NULL: LicensePlate, Branch, MaintenanceDate
UNIQUE: LicensePlate
CANDIDATE KEYS: LicensePlate

**** Underlined attributes are primary keys and bolded attributes are foreign keys. (X + Y) means AND, (X,Y) means OR.**

Functional Dependencies

- Mail
  - TrackingID → CurrentBranch, SenderUnitNum, SenderStreet, SenderPostCode, SenderCountry, NextUnitNum, NextStreet, NextPostCode, NextCountry, FinalUnitNum, FinalStreet, FinalPostcode, FinalCountry, Sender, Receiver, Weight, DeliveryType, Notes

- PostOfficeBranch
  - BranchID → UnitNum, Street, PostalCode, BranchManager
  - UnitNum, Street, PostalCode → BranchID, BranchManager
  - BranchManager → BranchID, UnitNum, StreetNum, PostalCode

- Employees
  - EmployeeID → Name, WorksAtBranch, Department

- Manager
  - EmployeeID → ManagesBranch
  - ManagesBranch → EmployeeID

- Driver
  - EmployeeID → LicenseNumber
  - LicenseNumber → EmployeeID

- Address
  - UnitNum, Street, PostalCode, Country → City, State/Province, BuildingType, Resident, LocalPostOffice
  - City, State/Province → Country
  - PostalCode, Country → City, State/Province
- Clients
  - ClientID → Name, ClientUnitNum, ClientStreet, ClientPostCode, PhoneNumber, Email, ClientCountry, PhoneNumber

- MailBoxes
  - MailBoxID → PostOfficeBranch, Street, PostalCode

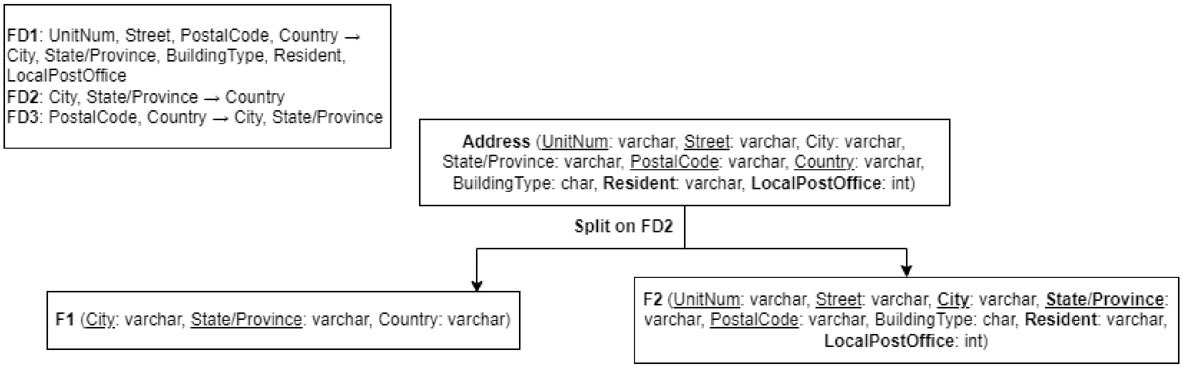
- Payment

- TransactionID → Payer, Mail, Price

- Truck
  - LicensePlate → Branch, MaintenanceDate

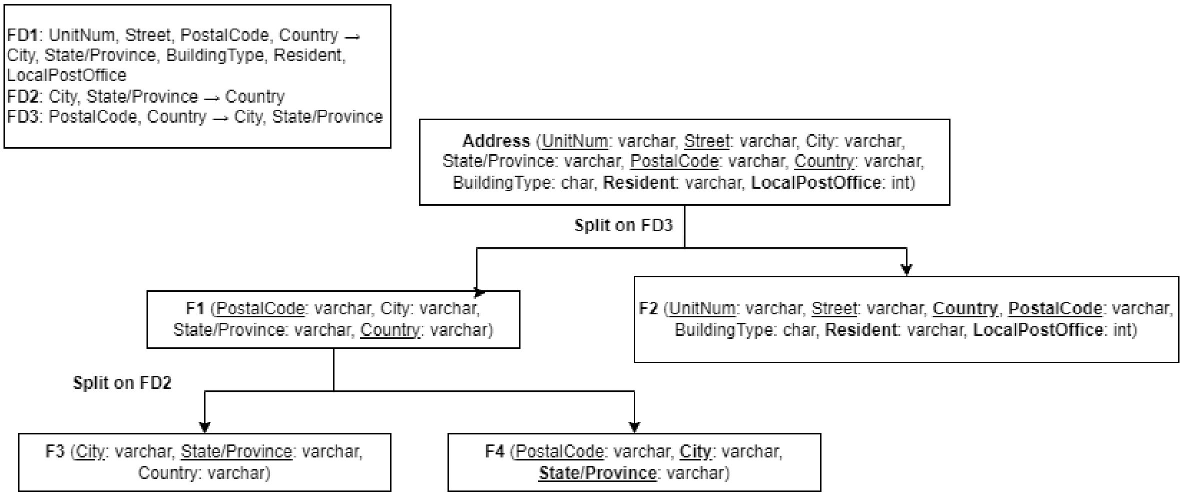
Normalization

- Address
  - It is not in BCNF and needs to be normalized.
  - **Attempt 1:**


FD1: UnitNum, Street, PostalCode, Country → City, State/Province, BuildingType, Resident, LocalPostOffice
FD2: City, State/Province → Country
FD3: PostalCode, Country → City, State/Province
Address (UnitNum: varchar, Street: varchar, City: varchar, State/Province: varchar, PostalCode: varchar, Country: varchar, BuildingType: char, Resident: varchar, LocalPostOffice: int)
Split on FD2
F1 (City: varchar, State/Province: varchar, Country: varchar)
F2 (UnitNum: varchar, Street: varchar, City: varchar, State/Province: varchar, PostalCode: varchar, BuildingType: char, Resident: varchar, LocalPostOffice: int)

Violates FD 1 and FD3

  - **Attempt 2:**


FD1: UnitNum, Street, PostalCode, Country → City, State/Province, BuildingType, Resident, LocalPostOffice
FD2: City, State/Province → Country
FD3: PostalCode, Country → City, State/Province
Address (UnitNum: varchar, Street: varchar, City: varchar, State/Province: varchar, PostalCode: varchar, Country: varchar, BuildingType: char, Resident: varchar, LocalPostOffice: int)
Split on FD3
F1 (PostalCode: varchar, City: varchar, State/Province: varchar, Country: varchar)
F2 (UnitNum: varchar, Street: varchar, Country, PostalCode: varchar, BuildingType: char, Resident: varchar, LocalPostOffice: int)
Split on FD2
F3 (City: varchar, State/Province: varchar, Country: varchar)
F4 (PostalCode: varchar, City: varchar, State/Province: varchar)

Violates FD3

  - BCNF violates our functional dependencies, hence we will be normalizing to 3NF.

  - **Address** (UnitNum: varchar, Street: varchar, **PostalCode**: varchar, **Country**: varchar, BuildingType: char, **Resident:** int, **LocalPostOffice**: int)

        - NOT NULL: UnitNum, Street, PostalCode, Country, LocalPostOffice
        - UNIQUE: (UnitNum + Street + PostalCode + Country)
        - CANDIDATE KEYS: (UnitNum + Street + PostalCode + Country)
    - **PostCode** (PostalCode: varchar, Country: varchar, City: varchar, State/Province: varchar)
        - NOT NULL: City, State/Province, PostalCode, Country
        - UNIQUE: (PostalCode + Country), (City + Province)
        - CANDIDATE KEYS: (PostalCode + Country), (Postal Code + City + Province),

- All other relations are in BCNF, so they do not need to be normalized.

SQL DDL

```
-- Table for Mail
CREATE TABLE Mail (
  TrackingID INT PRIMARY KEY,
  CurrentBranch INT NOT NULL,
  SenderUnitNum VARCHAR(20),
  SenderStreet VARCHAR(20),
  SenderPostCode VARCHAR(20),
  SenderCountry VARCHAR(20),
  NextUnitNum VARCHAR(20) NOT NULL,
  NextStreet VARCHAR(20) NOT NULL,
  NextPostCode VARCHAR(20) NOT NULL,
  FinalUnitNum VARCHAR(20) NOT NULL,
  FinalStreet VARCHAR(20) NOT NULL,
  FinalPostCode VARCHAR(20) NOT NULL,
  FinalCountry VARCHAR(20),
  Sender INT,
  Receiver INT NOT NULL,
  Weight FLOAT,
  DeliveryType VARCHAR(20) NOT NULL,
  Notes VARCHAR(200),
  FOREIGN KEY (CurrentBranch) REFERENCES PostOfficeBranch,
  FOREIGN KEY (SenderUnitNum, SenderStreet) REFERENCES Address(UnitNum, Street),
  FOREIGN KEY (SenderPostCode, SenderCountry) REFERENCES PostCode(PostalCode, Country),
  FOREIGN KEY (NextUnitNum, NextStreet) REFERENCES Address(UnitNum, Street),
  FOREIGN KEY (NextPostCode, NextCountry) REFERENCES PostCode(PostalCode, Country),
  FOREIGN KEY (FinalUnitNum, FinalStreet) REFERENCES Address(UnitNum, Street),
  FOREIGN KEY (FinalPostalCode, FinalCountry) REFERENCES PostCode(PostalCode, Country),
  FOREIGN KEY (Sender) REFERENCES Clients(ClientID),
  FOREIGN KEY (Receiver) REFERENCES Clients(ClientID)
);

-- Table for PostOfficeBranch
CREATE TABLE PostOfficeBranch (
  BranchID INT PRIMARY KEY,
  UnitNum VARCHAR(20) NOT NULL,
  Street VARCHAR(20) NOT NULL,
  PostalCode VARCHAR(20) NOT NULL,
  BranchManager INT NOT NULL UNIQUE,
  FOREIGN KEY (UnitNum, Street) REFERENCES Address,
  FOREIGN KEY (PostalCode)  REFERENCES PostCode,
  FOREIGN KEY (BranchManager) REFERENCES Manager(EmployeeID),
);

-- Table for Employees
CREATE TABLE Employees (
  EmployeeID INT PRIMARY KEY,
  Name VARCHAR(20) NOT NULL,
  WorksAtBranch INT NOT NULL,
  Department VARCHAR(20),
  FOREIGN KEY (WorksAtBranch) REFERENCES PostOfficeBranch(BranchID),
);

-- Table for Manager
CREATE TABLE Manager (
  EmployeeID INT PRIMARY KEY,
  ManagesBranch INT NOT NULL UNIQUE,
  FOREIGN KEY (ManagesBranch) REFERENCES PostOfficeBranch(BranchID),
  FOREIGN KEY (EmployeeID) REFERENCES Employees ON DELETE CASCADE
);

-- Table for Driver
CREATE TABLE Driver (
  EmployeeID INT PRIMARY KEY,
  LicenseNumber INT UNIQUE,
  FOREIGN KEY (EmployeeID) REFERENCES Employees ON DELETE CASCADE
);

-- Table for Address
CREATE TABLE Address (
  UnitNum VARCHAR(20),
  Street VARCHAR(20),
  PostalCode VARCHAR(20),
  Country VARCHAR(20),
  BuildingType VARCHAR(20),
  Resident INT,
  LocalPostOffice INT NOT NULL,
  PRIMARY KEY (UnitNum, Street, PostalCode, Country),
  FOREIGN KEY (PostCode, Country) REFERENCES PostCode(PostalCode, Country),
  FOREIGN KEY (Resident) REFERENCES Client(ClientID),
  FOREIGN KEY (LocalPostOffice) REFERENCES PostOfficeBranch(BranchID)
);

-- Table for PostCode
CREATE TABLE PostCode (
  PostalCode VARCHAR(20),
  Country VARCHAR(20),
  City VARCHAR(20) NOT NULL,
  State/Province VARCHAR(20) NOT NULL,
  PRIMARY KEY (PostalCode, Country),
);

-- Table for Clients
CREATE TABLE Clients (
  ClientID INT PRIMARY KEY,
  Name VARCHAR(20),
  ClientUnitNum VARCHAR(20),
  ClientStreet VARCHAR(20),
  ClientPostCode VARCHAR(20),
  ClientCountry VARCHAR(20),
  PhoneNumber INT,
  Email VARCHAR(20),
  FOREIGN KEY (ClientPostCode, ClientCountry) REFERENCES PostCode(PostalCode,
Country),
```

```sql
   FOREIGN KEY (ClientUnitNum, ClientStreet) REFERENCES Address(UnitNum, Street),
);

-- Table for MailBoxes
CREATE TABLE MailBoxes (
  MailBoxID INT PRIMARY KEY,
  PostOfficeBranch INT NOT NULL,
  Street VARCHAR(20) NOT NULL,
  PostalCode VARCHAR(20) NOT NULL
  FOREIGN KEY (PostOfficeBranch) REFERENCES PostOfficeBranch(BranchID) ON DELETE
CASCADE
);

-- Table for Payment
CREATE TABLE Payment(
  TransactionID INT PRIMARY KEY,
  Payer INT NOT NULL,
  Mail INT NOT NULL,
  Price FLOAT NOT NULL,
  FOREIGN KEY (Payer) REFERENCES Clients(ClientID),
  FOREIGN KEY (Mail) REFERENCES Mail(TrackingID)
);

-- Table for Truck
CREATE TABLE Truck(
  LicensePlate VARCHAR(20) PRIMARY KEY,
  Branch INT NOT NULL,
  MaintenanceDate DATE NOT NULL,
  FOREIGN KEY (Branch) REFERENCES PostOfficeBranch(BranchID)
);
```

Populating Tables

```sql
-- Inserting tuples into the Mail table
INSERT INTO Mail (TrackingID, CurrentBranch, SenderUnitNum, SenderStreet, SenderPostCode,
SenderCountry, NextUnitNum, NextStreet, NextPostCode, NextCountry, FinalUnitNum,
FinalStreet, FinalPostcode, FinalCountry, Sender, Receiver, Weight, DeliveryType, Notes)
VALUES
```

```
(1111, 1, '2-202', 'Broadway Ave', 'B2B 2B2', 'Canada', '404', 'Oak St', 'D4D 4D4', 'Canada',
'Canada', '404', 'Oak St', 'D4D 4D4', 'Canada', 2222, 4444, 1.2, 'Standard', 'Fragile')
(2222, 2, NULL, NULL, NULL, NULL, '505', 'Maple Ave', 'E5E 5E5', 'Canada', '101', 'Main St',
'A1A 1A1', 'Canada', NULL, 1111, 0.8, 'Standard', NULL)
(3333, 3, '505', 'Maple Ave', 'E5E 5E5', 'Canada', '202', 'Broadway Ave', 'B2B 2B2', 'Canada', '101',
'Main St', 'A1A 1A1', 'Canada', NULL, 1111, NULL, 'Express', NULL)
(4444, 4, NULL, NULL, NULL, NULL, '101', 'Main St', 'A1A 1A1', 'Canada', '303', 'Elm St', 'C3C
3C3', 'Canada', 2222, 3333, NULL, 'Standard', NULL)
(5555, 5, '404', 'Oak St', 'D4D 4D4', 'Canada', '303', 'Elm St', 'C3C 3C3', 'Canada', '202', 'Broadway
Ave', 'B2B 2B2', 'Canada', 4444, 2222, 2.2, 'Standard', 'Fragile');

-- Inserting tuples into the PostOfficeBranch table
INSERT INTO PostOfficeBranch (BranchID, UnitNum, Street, PostalCode, BranchManager)
VALUES
(1, '101', 'Main Street', 'A1A 1A1', 101),
(2, '5-505', 'Maple Ave', 'E5E 5E5',, 102),
(3, '2-202', 'Broadway', 'B2B 2B2', 103),
(4, '3-303', 'Elm Street', "C3C 3C3', 104),
(5, '404', 'Oak St', 'D4D 4D4', 105);

-- Inserting tuples into the Employees table
INSERT INTO Employees (EmployeeID, Name, WorksAtBranch, Department)
VALUES
(1, 'Alice', 1, 'Delivery'),
(2, 'Bob', 2, 'Delivery'),
(3, 'Charlie', 3, 'Delivery'),
(4, 'David', 4, 'Delivery'),
(5, 'Emma', 5, 'Delivery' ),
(6, 'Rina', 1, 'Customer Service'),
(7, 'Kevin', 2, 'Customer Service'),
(8, 'Charlie', 3, 'Customer Service'),
(9, 'Sarah', 4, 'Customer Service'),
(10, 'Sean', 5, 'Customer Service'),
(101, 'Kayla', 1, NULL),
(102, 'John', 2, NULL),
(103, 'Maria', 3, NULL),
(104, 'Lily', 4, NULL),
(105, 'Penny', 5, NULL)
```

```
-- Inserting tuples into the Manager table
INSERT INTO Manager VALUES
(101, 1),
(102, 2),
(103, 3),
(104, 4),
(105, 5)

-- Inserting tuples into the Driver table
INSERT INTO Driver (EmployeeID, LicenseNumber)
VALUES
(1, 123456),
(2, 789012),
(3, 345678),
(4, 901234),
(5, 567890);

-- Inserting tuples into the Address table
INSERT INTO Address (UnitNum, Street, PostalCode, Country, BuildingType, Resident,
LocalPostOffice)
VALUES
('101', 'Main St', 'A1A 1A1', 'Canada', 'House', 1, 1),
('2-202', 'Broadway Ave', 'B2B 2B2', 'Canada', 'Apartment', 2, 2),
('3-303', 'Elm St', 'C3C 3C3', 'Canada', 'Condo', 3, 3),
('404', 'Oak St', 'D4D 4D4', 'Canada', 'House', 4, 4),
('5-505', 'Maple Ave', 'E5E 5E5', 'Canada', 'Apartment', 5, 5);

-- Inserting tuples into the PostCode table
INSERT INTO PostCode (PostCode, Country, City, State/Province)
VALUES
('A1A 1A1', 'Canada', 'Vancouver', 'BC')
('B2B 2B2', 'Canada', 'Toronto', 'ON")
('C3C 3C3', 'Canada', 'Ottawa", 'ON')
('D4D 4D4', 'Canada', 'Montreal', 'QC')
('E5E 5E5', 'Canada', 'Edmonton', 'AB')

-- Inserting tuples into the Clients table
INSERT INTO Clients (ClientID, Name, ClientUnitNum, ClientStreet, ClientPostCode,
ClientCountry, PhoneNumber, Email)
VALUES
```

```
(1111, 'John Doe', '101', 'Main St', 'A1A 1A1', 'Canada', 1234567890, 'john@example.com'),
(2222, 'Jane Smith', '202', 'Broadway Ave', 'B2B 2B2', 'Canada', 2345678901, 'jane@example.com'),
(3333, 'Alice Johnson', '303', 'Elm St', 'C3C 3C3', 'Canada', 3456789012, 'alice@example.com'),
(4444, 'Bob Brown', '404', 'Oak St', 'D4D 4D4', 'Canada', 4567890123, 'bob@example.com'),
(5555, 'Emily Davis', '505', 'Maple Ave', 'E5E 5E5', 'Canada', 5678901234, 'emily@example.com');



-- Inserting tuples into the MailBoxes table
INSERT INTO MailBoxes
VALUES
(1, 1, 'Mountain St', 'A1A 1A1'),
(2, 1, 'Burrard St', 'A1A 1A1'),
(3, 3, 'Aspen Ave', 'C3C 3C3'),
(4, 4, 'Kirkland Drive', 'D4D 4D4'),
(5, 5, 'Lenovo Rd', 'E5E 5E5')

-- Inserting tuples into the Payment table
INSERT INTO Payment (TransactionID, Payer, Mail, Price)
VALUES
(1, 101, 1111, 10.99),
(2, 102, 2222, 15.75),
(3, 103, 3333, 20.50),
(4, 104, 4444, 12.25),
(5, 105, 5555, 18.99);

-- Inserting tuples into the Truck table
INSERT INTO Truck
VALUES
('NE866W', 1, 2023-11-01),
('VJ3N8K', 2, 2024-01-02)
('2FAST4U', 3, 2023-07-23),
('L87NE2', 4, 2023-09-15),
('PK3N84', 5, 2023-12-10)
```