

CPSC 304 Project Cover Page

Milestone #: 2

Date: Feb 28, 2023

Group Number: 94

| Name | Student Number | CS Alias (Userid) | Preferred E-mail Address |
|----------------|----------------|-------------------|---------------------------|
| Hansen Dan | 84620178 | u0c4h | hdan2580@gmail.com |
| Celine Liu | 20153755 | t3z6w | zijingliu2021@outlook.com |
| Bhavye Thukral | 80045370 | t3l0m | bhavyeth@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

Summary

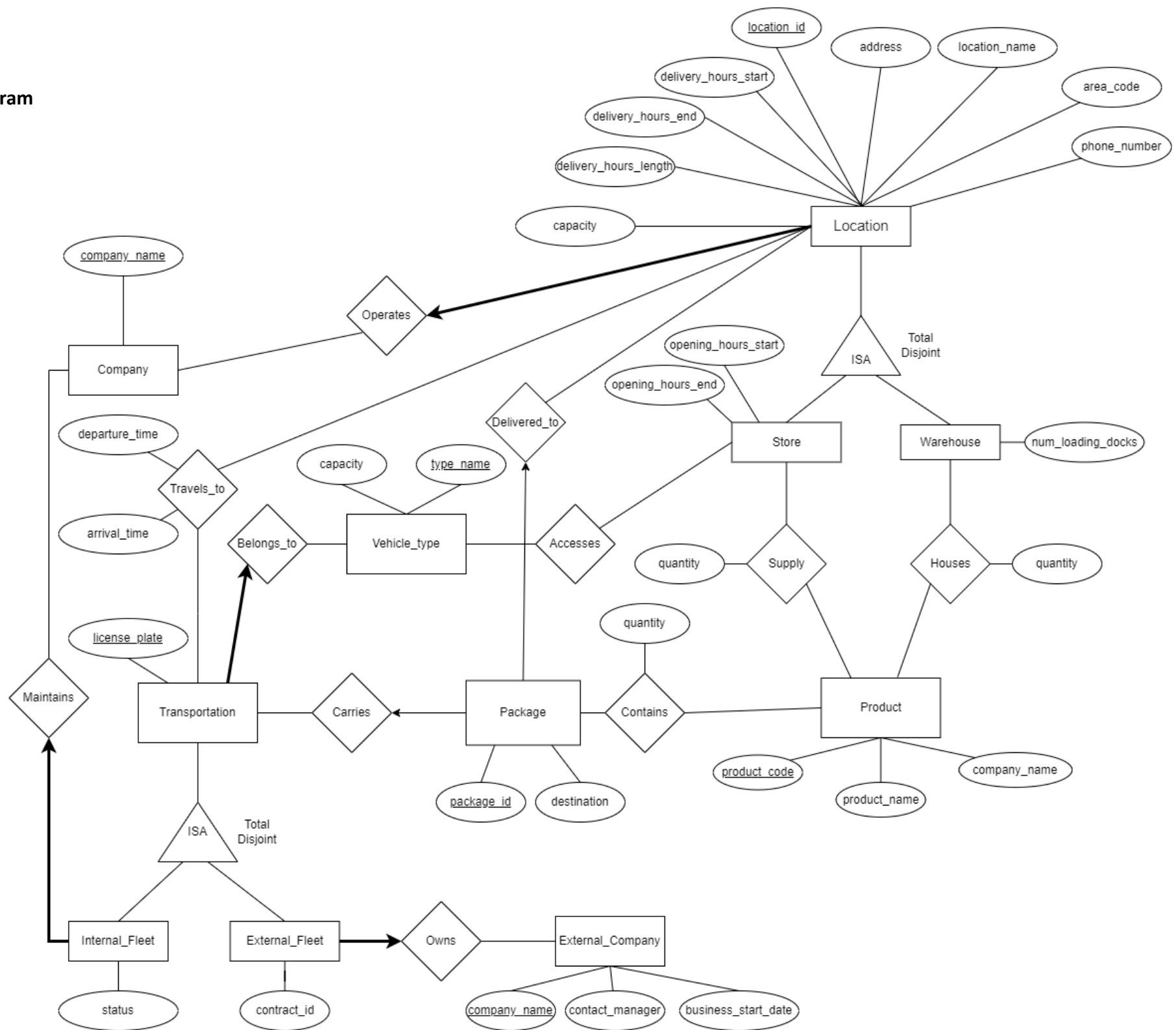
This application is designed as an inventory tracking system that tracks the status of each package and the quantity of the products in stock. Given a package, a user can view the products it contains and the exact location within the company's internal distribution network, including whether it is currently housed at a location or is in transit between two locations.

ER Diagram

[The ER Diagram is on the following page. Changes made since Module 1 are listed below:]

- Unbolded the lines of the ISA relationships and added the "disjoint, total" text boxes beside the ISA to convey the constraints more clearly.
- Adjusted the attributes of the Transportation ISA subclasses.
- To tackle the confusion of "one item/product can only be stored at a single store location", we combined the brand and the item entity into a product entity. Inside the product entity, a tuple represents a general type of product with information of product_code, product_name, and company_name. Therefore, each store and warehouse can have a certain amount of products.
- We changed the relations from Product to Stores and Locations into many-to-many relationships because each location can have multiple products, for example, a store can have multiple dried mangos produced by company A, and a Product can appear in multiple Locations.
- We changed the relation between Product and Package from one-to-many to many-to-many because different packages can contain the same kind of product. For example, package A and package B both contain dried mango produced by company A. We also added a quantity attribute to the Contains relationship to record the amount of a product inside a package.
- Added the Vehicle_type entity with the attributes of type_name and capacity.
- Added the Accesses relation between Vehicle_type and the Store so that we can know which type of vehicle can access a store, replacing the vehicle_access attribute of Store
- Added the Belongs to the relation between Transportation and Vehicle_type, replacing the vehicle_type attribute of Transportation
- Delivered_to is only populated when a Package arrives at a Location, Stock is only populated when a Package is loaded onto Transportation. The total participation constraints on packages were relaxed.
 - By relaxing the participation constraints on Package and revising the relationships between Package, Transportation, and Location, this scheme should be sufficient to capture package information without a Trips entity.
- Added the destination attribute of a Package to represent its final intended destination.
- Added additional attributes to External Company

ER Diagram



Schema

Legend:

Underline: primary key

Bold: foreign key

Note:

For both ISAs, we have decided to use one table for each entity, with excess relationships removed.

- Product(product_code: integer, product_name, company_name: char[30])
 - Candidate Keys: product_code
 - The following must not be NULL:
 - product_code, product_name, company_name
- Supplies(product_code: integer, location_id: integer, quantity: integer)
 - Candidate Keys: (product_code, location_id)
 - The following must not be NULL: product_code, location_id
- Houses(product_code: integer, location_id: integer, quantity: integer)
 - Candidate Keys: (product_code, location_id)
 - The following must not be NULL: product_code, location_id
- Contains(product_code: integer, package_id: integer, quantity: integer)
 - Candidate Keys: (product_code, package_id)
 - The following must not be NULL: product_code, package_id
- Location(location_id: integer, address: char[100], location_name: char[30], area_code: integer, phone_number: integer, delivery_hours_start: time, delivery_hours_end: time, delivery_hours_length: integer, capacity: integer, **company_name**: char[30])
 - Candidate Keys: location_id; (address, location_name)
 - Foreign Keys: **company_name** REFERENCES Company
 - The following must not be NULL
 - Location_id, address, location_name, company_name, capacity
- Store(location_id: integer, opening_hours_start: time, opening_hours_end: time)
 - Candidate Keys: location_id
 - Foreign Keys: location_id REFERENCES Location
 - The following must not be NULL
 - location_id
- Warehouse(location_id: integer, num_loading_docks: integer)
 - Candidate Keys: location_id
 - Foreign Keys: location_id REFERENCES Location
 - The following must not be NULL

■ location_id

- Package(package_id, destination: integer, **location_id**: integer, **license_plate**: char[6])
 - Candidate Keys: package_id
 - Foreign Keys: **location_id** REFERENCES Location; **license_plate** REFERENCES Transportation
 - The following attributes must not be null
 - package_id, location_id, destination
- Company(company_name: char[30])
 - Candidate Keys: company_name
- Vehicle_type(type_name: char[30], capacity: integer)
 - Candidate Keys: type_name
- Accesses(type_name: char[30], **location_id**: integer)
 - Candidate Keys: (type_name, **location_id**)
 - Foreign Keys: **location_id** REFERENCES Locations
- Transportation(license_plate: char[6], **type_name**: char[30])
 - Candidate Keys: license_plate
 - Foreign Keys: **type_name** REFERENCES Vehicle_type
- Internal_Fleet(**license_plate**: char[6], status: char[10], **company_name**: char[30])
 - Candidate Keys: license_plate
 - Foreign Keys: **license_plate** REFERENCES Transportation; **company_name** REFERENCES Company
- External_Fleet(**license_plate**: char[6], contract_id: integer, **company_name**: char[30])
 - Candidate Keys: license_plate
 - Foreign Keys: **license_plate** REFERENCES Transportation; **company_name** REFERENCES Company
- External_Company(company_name: char[30], contact_manager: char[20], business_start_date: date, **license_plate**: char[6])
 - Candidate Keys: company_name
 - Foreign Keys: **license_plate** REFERENCES Transportation
- Travels_to(license_plate: char[6], location_id: integer, departure_time: datetime, arrival_time: datetime)
 - Candidate Keys: (license_plate, location_id)

Functional Dependencies

- Product
 - $\text{product_code} \rightarrow \text{product_name}$
 - $\text{product_code} \rightarrow \text{company_name}$
- Supplies
 - $\text{product_code}, \text{location_id} \rightarrow \text{quantity}$
- Houses
 - $\text{product_code}, \text{location_id} \rightarrow \text{quantity}$
- Contains
 - $\text{package_id}, \text{product_code} \rightarrow \text{quantity}$
- Location
 - $\text{address} \rightarrow \text{area_code}$
 - $\text{delivery_hours_start}, \text{delivery_hours_end} \rightarrow \text{delivery_hours_length}$
 - $\text{location_id} \rightarrow \text{address}$
 - $\text{location_id} \rightarrow \text{location_name}$
 - $\text{location_id} \rightarrow \text{area_code}$
 - $\text{location_id} \rightarrow \text{phone_number}$
 - $\text{location_id} \rightarrow \text{delivery_hours_start}$
 - $\text{location_id} \rightarrow \text{delivery_hours_end}$
 - $\text{location_id} \rightarrow \text{delivery_hours_length}$
 - $\text{location_id} \rightarrow \text{capacity}$
 - $\text{address}, \text{location_name} \rightarrow \text{location_id}$
 - $\text{address}, \text{location_name} \rightarrow \text{area_code}$
 - $\text{address}, \text{location_name} \rightarrow \text{phone_number}$
 - $\text{address}, \text{location_name} \rightarrow \text{delivery_hours_start}$
 - $\text{address}, \text{location_name} \rightarrow \text{delivery_hours_end}$
 - $\text{address}, \text{location_name} \rightarrow \text{delivery_hours_length}$
 - $\text{address}, \text{location_name} \rightarrow \text{capacity}$
- Store
 - $\text{location_id} \rightarrow \text{opening_hours_start}$
 - $\text{location_id} \rightarrow \text{opening_hours_end}$
- Warehouse
 - $\text{location_id} \rightarrow \text{num_loading_docks}$
- Package
 - $\text{package_id} \rightarrow \text{destination}$
 - $\text{package_id} \rightarrow \text{location_id}$
 - $\text{package_id} \rightarrow \text{license_plate}$

- Vehicle_type
 - type_name → capacity
- Internal_Fleet
 - license_plate → status
 - license_plate → company_name
- External_Fleet
 - License_plate → contract_id
 - License_plate → company_name
- External_Company
 - company_name → contact_manager
 - company_name → business_start_date
 - company_name → Licence_plate
- Transportation
 - license_plate → type_name
- Travels_to
 - license_plate, location_id → departure_time
 - license_plate, location_id → arrival_time

Normalization

- Product, already in BCNF
- Supplies, already in BCNF
- Houses, already in BCNF
- Contains, already in BCNF
- Store, already in BCNF
- Warehouse, already in BCNF
- Company, already in BCNF
- Vehicle_type, already in BCNF
- Access, already in BCNF
- Internal_Fleet, already in BCNF
- External_Fleet, already in BCNF
- External_Company, already in BCNF
- Transportation, already in BCNF
- Travels_to, already in BCNF
- Location, not in BCNF, so we use decomposition to normalize Location into BCNF form
 - Closures
 - $\text{address}^+ = \{ \text{area_code} \}$

- $(\text{delivery_hours_start}, \text{delivery_hours_end})^+ = \{\text{delivery_hours_start}, \text{delivery_hours_end}, \text{delivery_hours_length}\}$
 - $\text{location_id}^+ = \{\text{location_id}, \text{address}, \text{location_name}, \text{area_code}, \text{phone_number}, \text{delivery_hours_start}, \text{delivery_hours_end}, \text{delivery_hours_length}, \text{capacity}\}$
 - $(\text{address}, \text{location_name})^+ = \{\text{location_id}, \text{address}, \text{location_name}, \text{area_code}, \text{phone_number}, \text{delivery_hours_start}, \text{delivery_hours_end}, \text{delivery_hours_length}, \text{capacity}\}$
- Step1: $\text{address} \rightarrow \text{area_code}$ violates the BCNF, decompose:
 R1(address, area_code), R2(address, location_id, location_name, phone_number, delivery_hours_start, delivery_hours_end, delivery_hours_length, capacity, **company_name**)
- Step2: $\text{delivery_hours_start}, \text{delivery_hours_end} \rightarrow \text{delivery_hours_length}$ violates BCNF, so decompose
 R3(delivery_hours_start, delivery_hours_end, delivery_hours_length),
 R4(address, location_id, location_name, phone_number, delivery_hours_start, delivery_hours_end, capacity, **company_name**)
- Final answer: R1(address, area_code), R3(delivery_hours_start, delivery_hours_end, delivery_hours_length), R4(address, location_id, location_name, phone_number, delivery_hours_start, delivery_hours_end, capacity, **company_name**)
- R1:
 - PK: address
 - CK: address
- R3:
 - PK: delivery_hours_start, delivery_hours_end
 - CK: delivery_hours_start, delivery_hours_end
- R4:
 - PK: location_id
 - CK: location_id, (address, location_name);
 - FK: **company_name**

SQL DDL Statements

The SQL DDL statements required to create all the tables from item #5. The statements should use the appropriate foreign keys, primary keys, UNIQUE constraints, etc.


```
CREATE TABLE Product (  
    product_code INTEGER PRIMARY KEY,  
    product_name CHAR(20) NOT NULL,  
    company_name CHAR(30) NOT NULL,  
);
```

```
CREATE TABLE Supplies (  
    product_code INTEGER NOT NULL,  
    location_id INTEGER NOT NULL,  
    quantity INTEGER,  
    PRIMARY KEY(product_code, location_id)  
);
```

```
CREATE TABLE Houses (  
    product_code INTEGER NOT NULL,  
    location_id INTEGER NOT NULL,  
    quantity INTEGER,  
    PRIMARY KEY(product_code, location_id)  
);
```

```
CREATE TABLE Contains(  
    product_code INTEGER NOT NULL,  
    package_id INTEGER NOT NULL,  
    quantity INTEGER,  
    PRIMARY KEY(product_code, package_id)  
);
```

```
CREATE TABLE Locations_R1(  
    address CHAR(100) PRIMARY KEY,  
    area_code INTEGER,  
);
```

```
CREATE TABLE Locations_R3(  
    delivery_hours_start TIME,  
    delivery_hours_end TIME,  
    delivery_hours_length INTEGER,  
    PRIMARY KEY (delivery_hours_start, delivery_hours_end)  
);
```

```
CREATE TABLE Locations_R4 (  
    address CHAR(100) NOT NULL,  
    location_id INT,  
    location_name CHAR(30) NOT NULL,  
    phone_number INT,  
    delivery_hours_start TIME,  
    delivery_hours_end TIME,
```

```
        company_name CHAR(30) NOT NULL,  
        capacity INT NOT NULL,  
        PRIMARY KEY (location_id),  
        UNIQUE (address, location_name)  
        FOREIGN KEY (company_name) REFERENCES Company  
  
);
```

```
CREATE TABLE Store(  
    location_id INTEGER PRIMARY KEY,  
    opening_hours_start TIME,  
    opening_hours_end TIME,  
    FOREIGN KEY (location_id) REFERENCES Location_R4  
);
```

```
CREATE TABLE Warehouse(  
    location_id INTEGER PRIMARY KEY,  
    num_loading_docks INTEGER,  
    FOREIGN KEY (location_id) REFERENCES Locations_R4  
);
```

```
CREATE TABLE Package(  
    package_id INTEGER PRIMARY KEY,  
    destination INTEGER NOT NULL,  
    location_id INTEGER,  
    license_plate CHAR(6),  
    FOREIGN KEY (location_id) REFERENCE Location_R4,  
    FOREIGN KEY (license_plate) REFERENCE Transportation  
);
```

```
CREATE TABLE Company (  
    company_name CHAR(30) PRIMARY KEY  
);
```

```
CREATE TABLE Vehicle_type(  
    type_name CHAR(30) PRIMARY KEY,  
    capacity INTEGER,  
    FOREIGN KEY (location_id) REFERENCES location_R4  
);
```

```
CREATE TABLE Transportation (  
    license_plate CHAR(6),  
    type_name CHAR(30),  
    PRIMARY KEY (license_plate),
```

```
        FOREIGN KEY (type_name) REFERENCES Vehicle_type
    );
```

```
CREATE TABLE Accesses(
    type_name CHAR(30),
    location_id INTEGER,
    PRIMARY KEY (type_name, location_id),
    FOREIGN KEY (location_id) REFERENCES Location_R4
);
```

```
CREATE TABLE Internal_Fleet (
    license_plate CHAR(6),
    status CHAR(16),
    company_name CHAR(30),
    PRIMARY KEY (license_plate),
    FOREIGN KEY(company_name) REFERENCES Company,
    FOREIGN KEY(license_plate) REFERENCES Transportation
);
```

```
CREATE TABLE External_Fleet (
    license_plate CHAR(6),
    contract_id INT,
    company_name CHAR(30),
    PRIMARY KEY (license_plate),
    FOREIGN KEY(company_name) REFERENCES Company,
    FOREIGN KEY(license_plate) REFERENCES Transportation
);
```

```
CREATE TABLE External_Company (
    company_name CHAR(30),
    contact_manager CHAR(20),
    business_start_date DATE,
    PRIMARY KEY (company_name)
);
```

```
CREATE TABLE Travels_to (
    license_plate CHAR(6),
    location_id INT,
    departure_time DATETIME,
    arrival_time DATETIME,
    PRIMARY KEY (license_plate, location_id)
);
```

INSERT Statements

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 on.

```
INSERT INTO Product
VALUES (1, "Dried Mangoes", "Mango Republic");
INSERT INTO Product
VALUES (2, "Mango Juice", "Mango Republic");
INSERT INTO Product
VALUES (3, "Canned Peas", "Canton Canning Company");
INSERT INTO Product
VALUES (4, "Canned Tomatoes", "Canton Canning Company");
INSERT INTO Product
VALUES (5, "Eraser", "Resare Limited");
```

```
INSERT INTO Supplies
VALUES (1, 103, 12);
INSERT INTO Supplies
VALUES (1, 104, 16);
INSERT INTO Supplies
VALUES (1, 104, 8);
INSERT INTO Supplies
VALUES (2, 103, 2);
INSERT INTO Supplies
VALUES (4, 105, 12);
```

```
INSERT INTO Houses
VALUES (1, 226, 50);
INSERT INTO Houses
VALUES (2, 226, 64);
INSERT INTO Houses
VALUES (1, 223, 81);
INSERT INTO Houses
VALUES (2, 213, 100);
INSERT INTO Houses
VALUES (1, 215, 10);
```

```
INSERT INTO Contains
VALUES (1, 1101, 20);
INSERT INTO Contains
VALUES (2, 1101, 12);
INSERT INTO Contains
VALUES (1, 1102, 10);
INSERT INTO Contains
VALUES (3, 1103, 20);
```

INSERT INTO Contains
VALUES (1, 1103, 10);

INSERT INTO Locations_R1
VALUES ("10800 170 Street, Surrey, BC", 604);
INSERT INTO Locations_R1
VALUES ("57098 E Bakerview Road", 360);
INSERT INTO Locations_R1
VALUES ("6400 Macdonald Street, Vancouver, BC", 604);
INSERT INTO Locations_R1
VALUES ("4900 Minoru Boulevard, Richmond, BC", 604);
INSERT INTO Locations_R1
VALUES ("3600 28 Avenue, Delta, BC", 604);
INSERT INTO Locations_R1
VALUES ("10900 180 Street, Surrey, BC", 604);
INSERT INTO Locations_R1
VALUES ("2700 King Road, Abbotsford, BC", 604);
INSERT INTO Locations_R1
VALUES ("5080 Pacific Street, Bellingham, WA", 360);
INSERT INTO Locations_R1
VALUES ("9000 Steveston Highway, Richmond, BC", 604);
INSERT INTO Locations_R1
VALUES ("300 Low level Rd, North Vancouver, BC", 604);

INSERT INTO Locations_R3
VALUES (18:00, 1:00, 7);
INSERT INTO Locations_R3
VALUES (19:00, 4:00, 9);
INSERT INTO Locations_R3
VALUES (18:00, 2:00, 8);
INSERT INTO Locations_R3
VALUES (6:00, 18:00, 12);
INSERT INTO Locations_R3
VALUES (4:00, 18:00, 14);

INSERT INTO Locations_R4
VALUES (101, "10800 170 Street, Surrey, BC", "AAA Market Surrey", 5556666, 18:00, 1:00, 1000, "AAA Canada");
INSERT INTO Location
VALUES (102, "57098 E Bakerview Road", Bellingham, WA", "AAA Market Bellingham", 7779999, 19:00, 4:00, 1500, "AAA US");
INSERT INTO Location
VALUES (103, "6400 Macdonald Street, Vancouver, BC", "AAA Market Vancouver", 6663333, 18:00, 2:00, 2000, "AAA Canada");
INSERT INTO Location
VALUES (104, "4900 Minoru Boulevard, Richmond, BC", "AAA Market Richmond", 6662222, 18:00, 2:00,, 2000, "AAA Canada");

INSERT INTO Location

VALUES (105, "3600 28 Avenue, Delta, BC", "AAA Market Delta", 6661111, 18:00, 1:00, 1000, "AAA Canada")

INSERT INTO Locations

VALUES (223, "10900 180 Street, Surrey, BC", "AAA Warehouse 23", 5557777, 6:00, 18:00, 15000, "AAA North America")

INSERT INTO Location

VALUES (226, "2700 King Road, Abbotsford, BC", "AAA Warehouse 26", 5558888, 6:00, 18:00, 15000, "AAA North America")

INSERT INTO Location

VALUES (213, "5080 Pacific Street, Bellingham, WA", "AAA Warehouse 13", 1112222, 6:00, 18:00, 15000, "AAA North America")

INSERT INTO Location

VALUES (215, "9000 Steveston Highway, Richmond, BC", "AAA Warehouse 15", 5553333, 6:00, 18:00, 15000, "AAA North America")

INSERT INTO Location

VALUES (218, "300 Low level Rd, North Vancouver, BC", "AAA Warehouse 18", 5552222, 4:00, 18:00, 7000, "AAA North America")

);

INSERT INTO Store

VALUES (101, 10:00, 18:00);

INSERT INTO Store

VALUES (102, 10:00, 19:00);

INSERT INTO Store

VALUES (103, 10:00, 18:00);

INSERT INTO Store

VALUES (104, 10:00, 18:00);

INSERT INTO Store

VALUES (105, 10:00, 18:00);

INSERT INTO Warehouse

VALUES (223, 6);

INSERT INTO Warehouse

VALUES (226, 7);

INSERT INTO Warehouse

VALUES (213, 10);

INSERT INTO Warehouse

VALUES (215, 4);

INSERT INTO Warehouse

VALUES (218, 12);

VALUES (1101, 103, 226, NULL);

INSERT INTO Package

VALUES (1102, 104, 223, NULL);

INSERT INTO Package

VALUES (1103, 103, NULL, "ABC 148");

```
INSERT INTO Package
VALUES (1104, 101, NULL, "CDE 789");
INSERT INTO Package
VALUES (1105, 213, NULL, "BCD 259");
```

```
INSERT INTO Company
VALUES ("AAA Canada");
INSERT INTO Company
VALUES ("AAA US");
INSERT INTO Company
VALUES ("AAA North America");
INSERT INTO Company
VALUES ("AAA Global");
INSERT INTO Company
VALUES ("AAA Quebec");
```

```
INSERT INTO Vehicle_type
VALUES ("box_truck", 100);
INSERT INTO Vehicle_type
VALUES ("18_wheel", 500);
INSERT INTO Vehicle_type
VALUES ("pickup", 50);
INSERT INTO Vehicle_type
VALUES ("van", 75);
INSERT INTO Vehicle_type
VALUES ("flatbed", 200);
```

```
INSERT INTO Transportation
VALUES ("ABC 148", "box_truck");
INSERT INTO Transportation
VALUES ("CDE 789", "box_truck");
INSERT INTO Transportation
VALUES ("DEF 232", "box_truck");
INSERT INTO Transportation
VALUES ("FGH 124", "box_truck");
INSERT INTO Transportation
VALUES ("GHI 616", "box_truck");
INSERT INTO Transportation
VALUES ("BCD 259", "18_wheel");
INSERT INTO Transportation
VALUES ("EFG 343", "18_wheel");
INSERT INTO Transportation
VALUES ("HIJ 120", "18_wheel");
INSERT INTO Transportation
VALUES ("IJK 987", "18_wheel");
INSERT INTO Transportation
```

```
VALUES("JKL 676", "18_wheel");
```

```
INSERT INTO Accesses
```

```
VALUES ("box_truck", 101);
```

```
INSERT INTO Accesses
```

```
VALUES ("18_wheel", 101);
```

```
INSERT INTO Accesses
```

```
VALUES ("box_truck", 102);
```

```
INSERT INTO Accesses
```

```
VALUES ("box_truck", 103);
```

```
INSERT INTO Accesses
```

```
VALUES ("box_truck", 104);
```

```
INSERT INTO Accesses
```

```
VALUES ("box_truck", 105);
```

```
INSERT INTO Accesses
```

```
VALUES ("18_wheel", 105);
```

```
INSERT INTO Internal_Fleet
```

```
VALUES ("ABC 148", "delivery", "AAA North America");
```

```
INSERT INTO Internal_Fleet
```

```
VALUES ("CDE 789", "delivery", "AAA North America");
```

```
INSERT INTO Internal_Fleet
```

```
VALUES ("DEF 232", "maintenance", "AAA North America");
```

```
INSERT INTO Internal_Fleet
```

```
VALUES ("FGH 124", "maintenance", "AAA North America");
```

```
INSERT INTO Internal_Fleet
```

```
VALUES ("GHI 616", "standby", "AAA North America");
```

```
INSERT INTO External_Fleet
```

```
VALUES ("BCD 259", "18_wheel", "YYY Logistics");
```

```
INSERT INTO External_Fleet
```

```
VALUES ("EFG 343", "18_wheel", "YYY Logistics");
```

```
INSERT INTO External_Fleet
```

```
VALUES ("HIJ 120", "18_wheel", "YYY Logistics");
```

```
INSERT INTO External_Fleet
```

```
VALUES("IJK 987", "18_wheel", "RSE Global Transport");
```

```
INSERT INTO External_Fleet
```

```
VALUES("JKL 676", "18_wheel", "LHD");
```

```
INSERT INTO External_Company
```

```
VALUES ("FedUp", "6044440044", August 17, 2018);
```

```
INSERT INTO External_Company
```

```
VALUES ("LHD", "6045550055", August 18, 2019);
```

```
INSERT INTO External_Company
```

```
VALUES ("YYY Logistics", "6047770077", January 29, 2013);
```

```
INSERT INTO External_Company
```



```
VALUES ("RSE Global Transport", "6042220022", February 10, 2014);  
INSERT INTO External_Company  
VALUES ("Global Leap", "6041110011", February 29, 2020);
```

```
INSERT INTO Travels_to  
VALUES ("ABC 148", 103, 2023-02-20 12:00:00, 2023-06-20 12:00:00);  
INSERT INTO Travels_to  
VALUES ("CDE 789", 101, 2023-02-20 12:00:00, 2023-06-20 12:00:00);  
INSERT INTO Travels_to  
VALUES ("GHI 616", 103, 2023-06-20 12:00:00, 2023-06-22 12:00:00);  
INSERT INTO Travels_to  
VALUES ("BCD 259", 213, 2023-02-24 12:00:00, 2023-05-20 12:00:00);  
INSERT INTO Travels_to  
VALUES("IJK 987", 215, 2023-05-10 12:00:00, 2023-05-12 12:00:00);
```