University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project - Transit System

Milestone #: 2

Date: October 8th, 2024

Group Number: 49

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Dayshaun Lee	76084086	o4h1m	lee.dayshaun@gmail.com
Minh Vu	33077769	p5n0o	minhvuams@gmail.com
Khue Do	70790423	k3h7x	khuedothi@gmail.com

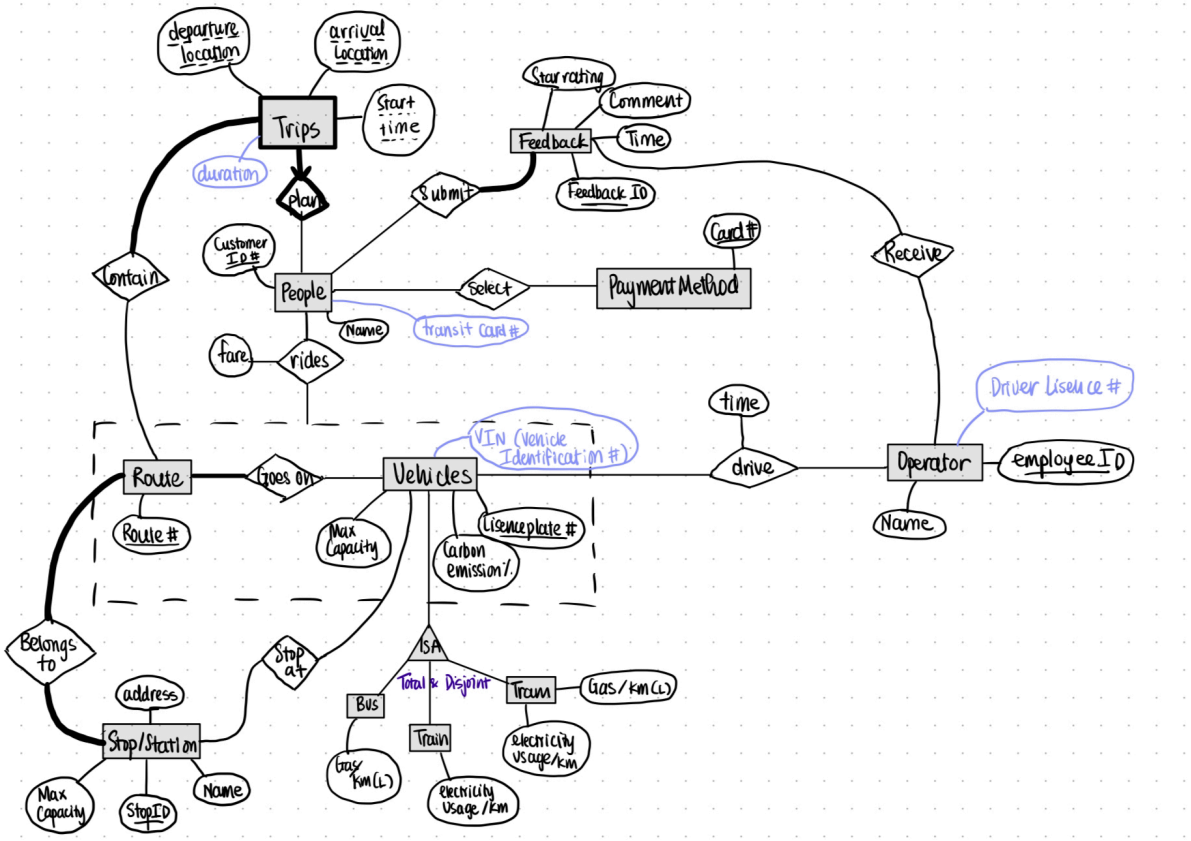
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address,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

University of British Columbia, Vancouver
Department of Computer Science

I. Project Summary

This project aims to optimize the Vancouver public transportation system by providing real-time scheduling data, operational vehicles and routes information, user feedback and environmental impact tracking through an Oracle database. It will manage bus and train schedules, vehicle maintenance, and ridership data to improve efficiency and wait times for customers, at the same time meet the city's sustainability goals.



II. Entity-Relationship Diagram

Changes made:

- Reorganized diagram to make it look neater, keeping the same as Milestone 1 diagram with addition of a few new changes below
- added startTime in trips as a partial key to indicate that people can make the same trip at different times
- changed relationship between Vehicles and Operator to be many-to-many to indicate that an operator may drive different vehicles at different times
- added PaymentMethod entity
- added additional attributes (in blue)

- Comment "it would be better to combine the rides and containment relationships into a single relationship between Trips and the aggregation, as a passenger's trip involves riding a vehicle that operates on a route" was not implemented because people can sometimes just hop on a bus without planning a trip. A Trip here is when you plan out your bus routes on Google Maps. You can also just take one bus anywhere. The "fare" attribute here is per ride per bus, not for the whole Trip. If you take the bus/train that is outside a specific zone it may have a different fare and cost more. We wanted to keep this "fare" attribute as is so decided to keep to this design but thanks for the suggestion!
- Removed availableShifts because it is a list
- changed Rides relationship between People and Vehicles to be People and aggregate to specify that people only ride on vehicles travelling a certain route
- ISA for Vehicles is total and disjoint
- Changed attributed for Train and Tram since Trains are fully electric and Trams are hybrid. This makes it easier to measure their carbonEmission later on.

III. Schema derived from ER diagram

NOTE: All entity, attribute, relationship names from the ER diagram have been made consistent using camelcase naming format for easier transition to code. Entity and Relationship names are in the format "EntityName" and attribute names are in the format "attributeName#"

Route(Route# (PK): Integer)

StopAt(lisencePlate#(FK, PK): Char(6), stopID (FK, PK): Integer)
GoesOn(route# (PK, FK): Integer, lisencePlate# (PK, FK): Char(6))
BelongsTo(route# (PK, FK): Integer, stopID (PK, FK): Integer)
*cannot fully model total participation constraint
Contain(route# (PK, FK): Integer, departureLocation (PK, FK): VARCHAR, arrivalLocation (PK, FK): VARCHAR, startTime (PK, FK): date, customerID (PK, FK): Integer)

TripsPlan(departureLocation (PK): Char(15), arrival location (PK): Char(15), startTime (PK): Date, customerID (FK, PK): Integer, duration: Integer)
*words in orange are supposed to be dash-underlined
*duration is in minutes

Feedback(feedbackID (PK):Integer, starRating:Integer, comment:Char(120), time:Date)
People(customerID (PK): Integer, name: VARCHAR, transitCard# (CK, UNIQUE): Integer)
Rides(customerID (PK, FK): Integer, lisencePlate# (PK, FK): Char(6), route# (PK, FK): integer, fare: Decimal(3,2))
Submit(customerID (PK, FK): Integer, feedbackID (PK, FK): Integer)
Receive(feedbackID (PK, FK): Integer, employeeID (PK, FK): Integer)

Stops(stopID (PK): Integer, address(CK): Char(20), maxCapacity: Integer, name (CK): Char(20))
Operator(employeeID (PK): Integer, driverLisence#(CK): Integer, name: Char(20))

Drive(lisencePlate# (PK, FK): Char(6), employeeID (PK, FK): Integer, time: Date)

Vehicles(lisencePlate# (PK): Char(6), capacity: Integer, carbonEmission: Decimal(5,2), startTime: Date, VIN(CK): Integer)
*note that VIN stands for "Vehicle Identification Number"

Bus(lisencePlate# (PK): Char(6), gas/km: Decimal(5,2), maxCapacity: Integer, carbonEmission: Decimal(5,2), VIN(CK): Integer)
Train(lisencePlate# (PK): Char(6), electricityUsage/km: Decimal(5,2), maxCapacity: Integer, carbonEmission: Decimal(5,2), VIN(CK): Integer)
Tram(lisencePlate# (PK): Char(6), electricityUsage/km: Decimal(5,2), gas/km:Decimal(5,2), maxCapacity: Integer, carbonEmission: Decimal(5,2), VIN(CK):Integer)
*note that VIN stands for "Vehicle Identification Number"

PaymentMethod(card# (PK): integer)
Select(customerID# (PK, FK): integer, card# (PK, FK): integer)

IV. Functional Dependencies

Entity/Relationship	**Functional Dependencies**
Route	NA
StopAt	licensePlate# -> stopID stopID -> licensePlate#
GoesOn	lisencePlate# ⟶ route# route# -> licensePlate#
BelongsTo	stopID ⟶ route# route# -> stopID
Contain	NA
TripsPlan	customerID, departureLocation, arrivalLocation, startTime ⟶ duration departureLocation, arrivalLocation, startTime ⟶ duration
Feedback	feedbackID ⟶ starRating, comment, time
People	customerID ⟶ name, transitCard#

	transitCard# → name, customerID
Rides	customerID, lisencePlate#, route# → fare
Submit	feedbackID → customerID customerID -> feedbackID
Receive	feedbackID → employeeID (should this be a many to one relationship)
Stops	name → stopID, address, maxCapacity stopID → name, address, maxCapacity address → name, maxCapacity
Operator	employeeID → driverLisence#, name driverLisence# → employeeID, name
Drive	lisencePlate#, employeeID → time
Vehicles	lisencePlate# → capacity, carbonEmission, startTime, VIN VIN → lisencePlate
Bus	lisencePlate → gas/km, maxCapacity, carbonEmission, VIN VIN → lisencePlate gas/km → carbonEmission
Train	lisencePlate → maxCapacity, carbonEmission, VIN VIN → lisencePlate electricity/km → carbonEmission
Tram	lisencePlate → electricityUsage, maxCapacity, carbonEmission, VIN VIN → lisencePlate gas/km, electricityUsage/km → carbonEmission
PaymentMethod	NA
Select	NA

V Normalization (BCNF)

TripsPlan(departureLocation (PK, CK): Char(15), arrival location (PK, CK): Char(15), startTime (PK, CK): Date, customerID (FK, PK): Integer, duration: Integer)

- {customerID, departureLocation, arrivalLocation, startTime}+ = {customerID, departureLocation, arrivalLocation, startTime, duration}

- {departureLocation, arrivalLocation, startTime}+ = {departureLocation, arrivalLocation, startTime, duration}
- decompose on departureLocation, arrivalLocation, startTime -> duration
- TripsPlan1(departureLocation (PK):VARCHAR, arrivalLocation (PK): VARCHAR, startTime(PK): date, duration: integer)
- TripsPlan2(customerID (PK): integer, departureLocation (PK, FK): VARCHAR, arrivalLocation PK, (PK, FK): VARCHAR, startTime (PK, FK): date)

Note: Can put into minimal cover for decomposition to 3NF

Stops(stopID (PK): Integer, address(CK): Char(20), Max capacity: Integer, name (CK): Char(20))

- {name}+ → {name, stopID, address, maxCapacity}
- {stopID}+ → {stopID, name, address, maxCapacity}
- {address}+ → {address, name, maxCapacity}
- decompose on address → name, maxCapacity
- Stops1(address (PK, FK): VARCHAR, name: VARCHAR, maxCapacity: integer)
- Stops2(stopID (PK): integer, address (PK, FK): VARCHAR)

Bus(lisencePlate# (PK): Char(6), gas/km: Decimal(5,2), VIN: Integer)

- {lisencePlate#}+ → {licensePlate#, gas/km, maxCapacity, carbonEmission}
- {VIN}+ →{VIN, licensePlate#, gas/km, maxCapacity, carbonEmission}
- {gas/km}+ → {gas/km, carbonEmission}
- Decompose on gas/km → carbonEmission
- Bus1(gas/km (PK, FK): Decimal(5, 2), carbonEmission: Decimal(5, 2))
- Bus2(gas/km (FK): Decimal(5, 2), maxCapacity: Integer, lisensePlate# (PK): char(6), VIN: Integer)

Tram(lisencePlate# (PK): Char(6), electricityUsage/km: Decimal(5,2), gas/km:Decimal(5,2), VIN: Integer)

- {lisencePlate#}+ → {lisencePlate#, electricityUsage/km, maxCapacity, carbonEmission}
- {VIN}+ → {VIN, lisencePlate#, electricityUsage/km, maxCapacity, carbonEmission}
- {electricityUsage}+ → {electricityUsage, carbonEmission}
- Decompose on electricityUsage → carbonEmission
- Tram1(electricityUsage/km (PK, FK): Decimal(5,2), carbonEmission: Decimal(5,2))
- Tram2(lisencePlate# (PK): Char(6), maxCapacity: Integer, electricityUsage: Decimal(5,2), VIN: Integer)

Train(lisencePlate# (PK): Char(6), electricityUsage/km: Decimal(5,2), VIN: Integer)

- {lisencePlate#}+ → {licensePlate#, electricity/km, maxCapacity, carbonEmission}
- {VIN}+ → {VIN, licensePlate#, electricity/km, maxCapacity, carbonEmission}
- {electricity/km}+ → {electricity/km, carbonEmission}
- Decompose on electricity/km → carbonEmission
- Train1(electricity/km (PK, FK): Decimal(5, 2), carbonEmission: Decimal(5, 2))

- Train2(electricity/km (FK): Decimal(5, 2), maxCapacity: Integer, licensePlate# (PK): char(6), VIN: Integer)

Schema after Normalization

Route(Route# (PK): Integer)

StopAt(lisencePlate#(FK, PK, CK): Char(6), stopID (FK, PK, CK): Integer)
GoesOn(route# (PK, FK, CK): Integer, lisencePlate# (PK, FK, CK): Char(6))
BelongsTo(route# (PK, FK, CK): Integer, stopID (PK, FK, CK): Integer)
*cannot fully model total participation constraint
Contain(route# (PK, CK, FK): Integer, departureLocation (PK, FK, CK): VARCHAR, arrivalLocation (PK, FK, CK): VARCHAR, startTime (PK, FK, CK): date, customerID (PK, FK, CK): Integer)

TripsPlan1(departureLocation (PK):VARCHAR, arrivalLocation (PK): VARCHAR, startTime(PK): date, duration: integer)
TripsPlan2(customerID (PK): integer, departureLocation (FK): VARCHAR, arrivalLocation (FK): VARCHAR, startTime (FK): date)

Feedback(feedbackID (PK):Integer, starRating:Integer, comment:Char(120), time:Date)
People(customerID (PK, CK): Integer, name: VARCHAR, transitCard# (CK, UNIQUE): Integer)
Rides(customerID (PK, CK, FK): Integer, lisencePlate# (PK, CK, FK): Char(6), route# (PK, CK, FK): integer, fare: Decimal(3,2))
Submit(customerID (PK, FK, CK): Integer, feedbackID (PK, FK, CK): Integer)
Receive(feedbackID (PK, FK, CK): Integer, employeeID (PK, FK, CK): Integer)
Stops1(address (PK, FK): VARCHAR, name: VARCHAR, maxCapacity: integer)
Stops2(stopID (PK): integer, address (PK, FK): VARCHAR)
Operator(employeeID (PK, CK): Integer, driverLisence#(CK): Integer, name: Char(20))

Drive(lisencePlate# (PK, FK): Char(6), employeeID (PK) (FK): Integer, time: Date)

Vehicles(lisencePlate# (PK): Char(6), capacity: Integer, carbonEmission: Decimal(5,2), startTime: Date, VIN(CK): Integer)
*note that VIN stands for “Vehicle Identification Number”
Bus1(gas/km (PK, FK): Decimal(5, 2), carbonEmission: Decimal(5, 2))
Bus2(gas/km (FK): Decimal(5, 2), maxCapacity: integer, licensePlate# (PK): char(6), VIN: Integer)
Train1(electricity/km (PK, FK): Decimal(5, 2), carbonEmission: Decimal(5, 2))
Train2(electricity/km (FK): Decimal(5, 2), maxCapacity: Integer, licensePlate# (PK): char(6), VIN: Integer)
Tram1(electricityUsage/km (PK, FK): Decimal(5,2), carbonEmission: Decimal(5,2))
Tram2(lisencePlate#: Char(6), maxCapacity: Integer, electricityUsage(PK, FK): Decimal(5,2), VIN: Integer)
*note that VIN stands for “Vehicle Identification Number”

PaymentMethod(card# (PK, CK): integer)
Select(customerID# (PK, CK, FK): integer, card# (PK, CK, FK): integer)

VII. SQL DDL

*cannot represent foreign keys to Vehicles since we represent the ISA with method 3 (no table for superclass)

```
CREATE TABLE Route (
route# int PRIMARY KEY
)

CREATE TABLE StopAt (
licencePlate# char(6),
stopID int,
PRIMARY KEY (licencePlate#, stopID),
FOREIGN KEY (stopID) REFERENCES Stops1 ON DELETE CASCADE)

CREATE TABLE BelongsTo (
route# int,
stopID int,
PRIMARY KEY (route#, stopID),
FOREIGN KEY (route#) REFERENCES Route ON DELETE CASCADE,
FOREIGN KEY (stopID) REFERENCES Stop1, Stop2 ON DELETE CASCADE)
*cannot fully express participation constraint

CREATE TABLE Contains (
route# int,
departureLocation VARCHAR,
arrivalLocation VARCHAR,
startTime date,
customerID int,
PRIMARY KEY (route#, departureLocation, arrivalLocation, startTime, customerID),
FOREIGN KEY (route#) REFERENCES Route ON DELETE CASCADE,
FOREIGN KEY (departureLocation, arrivalLocation, startTime) REFERENCES TripsPlan1 ON
DELETE CASCADE,
FOREIGN KEY (departureLocation, arrivalLocation, startTime) REFERENCES TripsPlan2 ON
DELETE CASCADE)

CREATE TABLE TripsPlan1 (
departureLocation VARCHAR,
arrivalLocation VARCHAR,
startTime date,
```

duration int,
PRIMARY KEY (departureLocation, arrivalLocation, startTime),
FOREIGN KEY (departureLocation, arrivalLocation, startTime) REFERENCES TripsPlan2 ON DELETE CASCADE)

CREATE TABLE TripsPlan2 (
departureLocation VARCHAR,
arrivalLocation VARCHAR,
startTime date,
customerID int,
PRIMARY KEY (customerID, arrivalLocation, startTime, departureLocation),
FOREIGN KEY (departureLocation, arrivalLocation, startTime) REFERENCES TripsPlan1 ON DELETE CASCADE,
FOREIGN KEY (customerID) REFERENCES Customers ON DELETE CASCADE)

CREATE TABLE GoesOn (
route# int,
licensePlate# char(6),
PRIMARY KEY (route#, licensePlate#),
FOREIGN KEY (route#) REFERENCES Route ON DELETE CASCADE)

CREATE TABLE Feedback (
feedbackID int PRIMARY KEY,
starRating int,
comment char(120),
time date)

CREATE TABLE People (
customerID int PRIMARY KEY,
name VARCHAR,
transitCard# int UNIQUE)

CREATE TABLE Rides (
customerID int,
route# int,
licencePlate# int,
fare Decimal(3,2),
PRIMARY KEY (customerID, route#, licensePlate#),
FOREIGN KEY (customerID) REFERENCES People ON DELETE CASCADE,
FOREIGN KEY (route#) REFERENCES Route ON DELETE CASCADE)

CREATE TABLE Submit (
customerID int,

```
feedbackID int,
PRIMARY KEY (customerID, feedbackID),
FOREIGN KEY (customerID) REFERENCES Customers ON DELETE CASCADE,
FOREIGN KEY (feedbackID) REFERENCES Feedback ON DELETE CASCADE)

CREATE TABLE Receive (
feedbackID int,
employeeID int,
PRIMARY KEY (feedbackID, employeeID),
FOREIGN KEY (feedbackID) REFERENCES Feedback ON DELETE CASCADE,
FOREIGN KEY (employeeID) REFERENCES Employee ON DELETE CASCADE)

CREATE TABLE Stops1 (
stopID int PRIMARY KEY,
address char(20),
FOREIGN KEY (address) REFERENCES Stops2 ON DELETE CASCADE)

CREATE TABLE Stops2 (
address char(20) PRIMARY KEY,
FOREIGN KEY (address) REFERENCES Stops1 ON DELETE CASCADE)

CREATE TABLE Operator (
employeeID int PRIMARY KEY,
driverLicense# int,
name VARCHAR)

CREATE TABLE Drive (
licencePlate# char(6),
employeeID int,
time date,
PRIMARY KEY (licencePlate#, employeeID),
FOREIGN KEY (employeeID) REFERENCES Operator ON DELETE CASCADE)

CREATE TABLE Vehicles (
licencePlate# char(6),
capacity int,
carbonEmission Decimal(5,2),
VIN int UNIQUE)

CREATE TABLE Bus1 (
gas/km Decimal(5, 2) PRIMARY KEY,
carbonEmission Decimal(5, 2),
FOREIGN KEY (gas/km) REFERENCES Bus2 ON DELETE CASCADE)
```

```
CREATE TABLE Bus2 (
gas/km Decimal(5, 2),
maxCapacity integer,
licensePlate# PRIMARY KEY,
VIN integer,
FOREIGN KEY (gas/km) REFERENCES Bus1 ON DELETE CASCADE)

CREATE TABLE Train1 (
electricity/km Decimal(5, 2) PRIMARY KEY,
carbonEmission Decimal(5, 2),
FOREIGN KEY (electricity/km) REFERENCES Train2 ON DELETE CASCADE)

CREATE TABLE Train2 (
electricity/km Decimal(5, 2),
maxCapacity Integer,
licensePlate# char(6) PRIMARY KEY,
VIN integer,
FOREIGN KEY (electricity/km) REFERENCES Train1 ON DELETE CASCADE)

CREATE TABLE Tram1(
electricityUsage/km Decimal(5, 2) PRIMARY KEY,
carbonEmission Decimal(5, 2),
FOREIGN KEY (electricityUsage/km) REFERENCES Tram2 ON DELETE CASCADE)

CREATE TABLE Tram2(
licensePlate# PRIMARY KEY,
electricityUsage/km Decimal(5, 2),
maxCapacity integer,
VIN integer,
FOREIGN KEY (electricityUsage/km) REFERENCES Tram1 ON DELETE CASCADE)

CREATE TABLE PaymentMethod (
card# int PRIMARY KEY)

CREATE TABLE Select (
card# int,
customerID# int,
PRIMARY KEY (card#, customerID#),
FOREIGN KEY (card#) REFERENCES PaymentMethod ON DELETE CASCADE,
FOREIGN KEY (customerID#) REFERENCES People ON DELETE CASCADE)
```

VIII Insert Statements

```
INSERT INTO Route (route#) VALUES (99), (84), (1), (352), (68);

INSERT INTO StopAt (licencePlate#, stopID)
VALUES ('ABC123', 101), ('QPC485', 305),
    ('JD9876', 7), ('CD5678', 104),
    ('EF9012', 105);

INSERT INTO GoesOn(route#, lisencePlate#)
VALUES (‘99’, ‘ABC123’),
(‘84’, 'QPC485'),
(‘352’, 'CD5678'),
(‘1’, 'EF9012'),
(‘68’, 'JD9876')

INSERT INTO BelongsTo (route#, stopID)
VALUES (99, 101), (84, 305),
    (352, 7), (1, 104),
    (68, 105);

INSERT INTO Contains (route#, departureLocation, arrivalLocation, startTime)
VALUES (99, 'C', 'D', '2024-09-10 20:10:00 '),
    (1, 'Metrotown', 'UBC', '2024-10-15 08:00:00'),
    (84, 'VCC-Clark', 'UBC', '2024-10-15 12:00:00'),
    (352, 'A', 'B', '2024-10-18 17:30:30'),
    (68, 'H', 'I', '2024-10-19 23:00:00');

INSERT INTO TripsPlan1 (departureLocation, arrivalLocation, startTime, duration)
VALUES ('C', 'D', '2024-09-10 20:10:00', 60),
    ('Metrotown', 'UBC', '2024-10-15 08:00:00', 70),
    ('VCC-Clark', 'UBC', '2024-10-15 12:00:00', 45),
    ('A', 'B', '2024-10-18 17:30:30', 8),
    ('H', 'I', '2024-10-19 23:00:00', 12);


INSERT INTO TripsPlan2 (departureLocation, arrivalLocation, startTime, customerID)
VALUES ('C', 'D', '2024-09-10 20:10:00', 101),
    ('Metrotown', 'UBC', '2024-10-15 08:00:00', 102),
    ('VCC-Clark', 'UBC', '2024-10-15 12:00:00', 103),
    ('A', 'B', '2024-10-18 17:30:30', 104),
    ('H', 'I', '2024-10-19 23:00:00', 105);
```

```
INSERT INTO Feedback (feedbackID, starRating, comment, time)
VALUES (1, 5, 'Good service', '2024-10-01 10:00:00'),
    (2, 3, 'Okay ride', '2024-10-02 12:15:00'),
    (3, 1, 'Never riding again', '2024-10-03 14:30:00'),
    (4, 4, 'Smooth ride', '2024-10-05 16:45:00'),
    (5, 2, 'Late bus', '2024-10-06 18:00:00');

INSERT INTO People (customerID, name, transitCard#)
VALUES (1,'Michael Jackson', 12345),
    (2, 'John Smith', 67890),
    (3, 'First Last', 11223),
    (4, 'Apple Orange', 33445),
    (5, 'Name Name', 55667);

INSERT INTO Rides (customerID, route#, licencePlate#, fare) VALUES (1, 99, 'ABC123', 2.50), (2,
1, 'CD5678', 3.00), (3, 84, 'QPC485', 2.75), (4, 352, 'JD9876', 4.50), (5, 68, 'EF9012', 3.25);

INSERT INTO Submit (customerID, feedbackID)
VALUES (1, 1),
    (2, 2),
    (3, 3),
    (4, 4),
    (5, 5);

INSERT INTO Receive (feedbackID, employeeID)
VALUES (1, 5),
    (2, 1),
    (3, 2),
    (4, 4),
    (5, 3);

INSERT INTO Stops (stopID, address, maxCapacity, name)
VALUES (101, '123 Main St', 50, 'Waterfront'),
    (305, '456 Elm St', 60, 'UBC'),
    (7, '789 Oak St', 40, 'Production-Way'),
    (104, '321 Pine St', 70, 'VCC-Clark', '104, 105'),
    (105, '654 Cedar St', 30, 'Main Street', '105');

INSERT INTO Operator (employeeID, driverLicense#, name)
VALUES (1, 123456, 'Adam'),
    (4, 111111, 'Bonnie'),
    (3, 121212, 'Cameron'),
    (2, 323232, 'Dane'),
```

```
    (5, 198237, 'Erica');

INSERT INTO Drive (licencePlate#, employeeID, time)
VALUES ('ABC123', 3, '2024-09-10 20:10:00'),
    ('CD5678', 2, '2024-10-15 08:00:00'),
    ('QPC485', 1, '2024-10-15 12:00:00'),
    ('JD9876', 4, '2024-10-18 17:30:30'),
    ('EF9012', 5, '2024-10-19 23:00:00');

CREATE TABLE Vehicles (
licencePlateNumber char(6),
capacity int,
carbonEmission Decimal(5,2),
startTime date,
VIN int)

INSERT INTO Bus1 (gas/km, carbonEmission)
VALUES (8.50, 5.75),
(10.50, 6.25),
(7.50, 6.00),
(6.00, 3.00),
(10.00, 12.00);

INSERT INTO Bus2 (gas/km, carbonEmission, licensePlate#, VIN)
VALUES (8.50, 60, ‘ABC123’, 123456),
(10.50, 50, 'CD5678', 234567),
(7.50, 40, 'JD9876', 345678),
(6.00, 30, 'EF9012', 234677),
(10.00, 100, 'QPC485', 098765);

INSERT INTO Train1 (electricityUsage/km, carbonEmission)
VALUES (3.50, 4.00),
(5.50, 4.25),
(6.00, 5.00),
(7.00, 6.25),
(8.25, 10.00);

INSERT INTO Train2 (licensePlateNumber, electricityUsage/km, maxCapacity, VIN)
VALUES (‘SA3512’, 3.50, 20, 864734),
(‘FB3451’, 5.50, 25, 323455),
(‘FC4574’, 6.00, 30, 345734),
(‘QD2564’, 7.00, 35, 785684),
(‘PE1235’, 8.25, 40, 323497);
```

```
INSERT INTO Tram1 (electricityUsage/km, carbonEmission)
VALUES (3.50, 4.00),
(5.50, 4.25),
(6.00, 5.00),
(7.00, 6.25),
(8.25, 10.00);

INSERT INTO Tram2 (licensePlate#, electricityUsage/km, maxCapacity, VIN)
VALUES ('SD3512', 3.50, 20, 112233),
('FG3451', 5.50, 25, 445566),
('FG4574', 6.00, 30, 778899),
('QW2564', 7.00, 35, 101112),
('PL1235', 8.25, 40, 121314);

INSERT INTO PaymentMethod (card#)
VALUES (12345), (67890), (11223), (33445), (55667);

INSERT INTO Select (card#, customerID#)
VALUES (12345, 101),
    (67890, 102),
    (11223, 103),
    (33445, 104),
    (55667, 105);
```