CPSC 304

Milestone #2

Date: October 21st, 2022

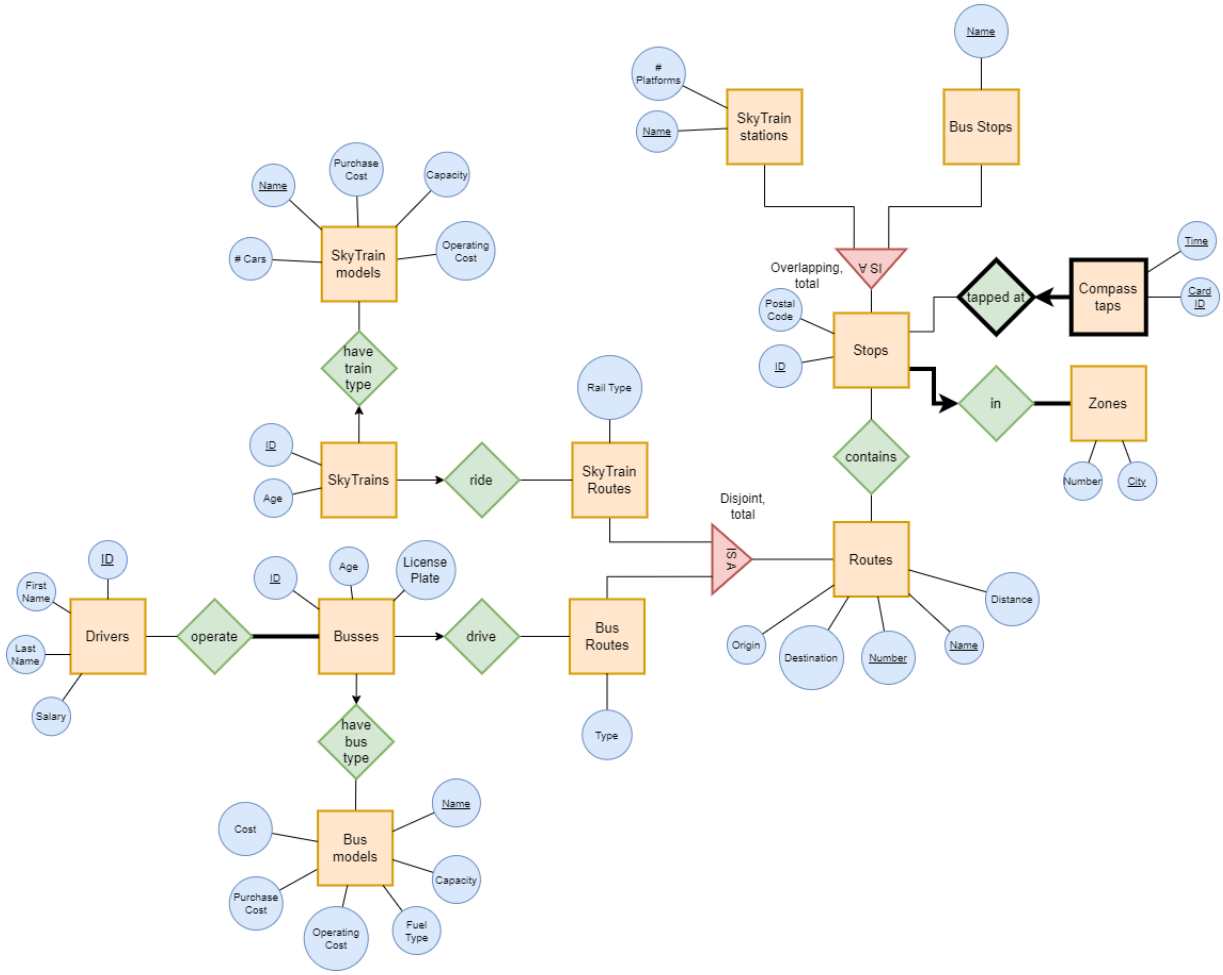
Group Number: 72

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Tom Henley	40054793	u6i2b	tomhenleyyy@gmail.com
Aviva Mei	74065350	j7x9t	aviivameii@gmail.com
Emily Chu	26625426	s3y2b	im.mlechu@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

ER Diagram



[Link for viewing of finer details (e.g. underlined attributes)]

Changes made since Milestone 1:

- From TA feedback, the relationship between Route and Stop is now many-to-many. The participation constraint has also been removed since a route or a stop may be temporarily disconnected from the system (e.g. construction).
- From TA feedback, “time” has been added to the compass tap key.
- From TA feedback, the relationship between Bus and Bus Model has been changed from ISA to a one-to-many relationship. A similar change to the above was made for Skytrain and SkyTrain Models.
- Bus model attribute “Type” is now “Fuel Type”.
- For both the bus model and Skytrain model, the “cost” attribute has been split into two attributes: “operating cost” and “purchase cost”.
- Driver now has more attributes: (ID, First Name, Last Name, Salary) instead of just ID and name.
- Removed number attribute from bus stop (now only name)
- Zone now has the attribute “city” instead of “fare”, and “name” was renamed to “number”

Unrefined schema from ER Diagram

Primary keys are underlined.
Foreign keys are in bold. (check SQL DDL section for the tables they are referencing).

- Bus(ID: CHAR(10), LicensePlate: CHAR(6), **Model: CHAR(20), RouteName: CHAR(50), RouteNumber: CHAR(3)**)
- BusModel(Name: CHAR(20), Cost: INT, Capacity: INT, FuelType: CHAR(6), Cost: INT, PurchaseCost: INT, OperatingCost: INT)
- Driver(ID: CHAR(8), FirstName: CHAR(30), LastName: CHAR(30))
- Skytrain(ID: CHAR(3), **Model: CHAR(20), RouteName: CHAR(50), RouteNumber: CHAR(3)**)
- SkytrainModel(Name: CHAR(20), Cost: INT, Capacity: INT, Cars: INT, PurchaseCost: INT, OperatingCost: INT)
- Route(Name: CHAR(50), Number: CHAR(3), **Origin: CHAR(10)**, **Destination: CHAR(10)**, Distance: REAL, BusRouteType: CHAR(20), RailType: CHAR(20))
- SkytrainStation(**StopID: CHAR(10)**, Name: CHAR(50), Platforms: INT)
- BusStop(**StopID: CHAR(10)**, Name: CHAR(50))
- Stop(ID: CHAR(10), Postcode: CHAR(6), **City: CHAR(20)**)
- Zone(Number: INT, City: CHAR(20))
- CompassTap(Card ID: CHAR(20), Time: INT, **Stop: CHAR(50)**)
- DriverAssignment(**DriverID: CHAR(8)**, **BusID: CHAR(10)**)
- AvailableStop(**Stop: CHAR(20)**, **RouteName: CHAR(50)**, **RouteNumber: CHAR(3)**)

Functional dependencies

- Bus
  - ID → LicensePlate, Model, RouteName, RouteNumber
- BusModel
  - Name → Cost, Capacity, FuelType, PurchaseCost, OperatingCost
  - PurchaseCost, OperatingCost → Cost
- Driver
  - ID → FirstName, LastName
- Skytrain

    - ID → Model, RouteName, RouteNumber
- SkytrainModel
    - Name → Cost, Capacity, Cars, PurchaseCost, OperatingCost
    - PurchaseCost, OperatingCost → Cost
- Route
    - Name, Number → Origin, Destination, Distance, BusRouteType, RailType
    - Number → BusRouteType
- SkytrainStation
    - Name, StopID → Platforms
- BusStop
    - (No non-trivial dependencies)
- Stop
    - ID → Postcode, City
- Zone
    - City → Number
- CompassTap
    - (No non-trivial dependencies)
- DriverAssignment
    - (No non-trivial dependencies)
- AvailableStop
    - (No non-trivial dependencies)

Normalization

Tables already in 3NF:

Bus(ID: CHAR(10), LicensePlate: CHAR(6), **Model: CHAR(20), RouteName: CHAR(50), Route Number: CHAR(3)**)

- FD: ID → License Plate, Model, RouteName, RouteNumber
- Minimal Key: {ID}

Driver(ID: CHAR(8), FirstName: CHAR(30), LastName: CHAR(30))

- FD: ID → FirstName, LastName
- Minimal Key: {ID}

Skytrain(ID: CHAR(3), **Model: CHAR(20), RouteName: CHAR(50), RouteNumber: CHAR(3)**)

- FD: ID → Model, RouteName, RouteNumber
- Minimal Key: {ID}

SkytrainStation(**StopID: CHAR(10)**, Name: CHAR(50), Platforms: INT)

- FD: Name, Stop ID → Platforms
- Minimal Key: {Name, StopID}

BusStop(**StopID: CHAR(10),** Name: CHAR(50))
- Minimal Key: {Name, StopID}

Stop(ID: CHAR(10), Postcode: CHAR(6), **City: CHAR(20)**)
- FD: ID → Postcode, City
- Minimal Key: {ID}

Zone(Number: INT, City: CHAR(20))
- FD: City → Number
- Minimal Key: {City}

CompassTap(CardID: CHAR(20), Time: INT, **Stop: CHAR(10)**)
- Minimal Key: {CardID, Time, Stop}

DriverAssignment(**DriverID: CHAR(8)**, **BusID: CHAR(10)**)
- Minimal Key: {DriverID, BusID}

AvailableStop(**Stop: CHAR(10)**, **RouteName: CHAR(50)**, **RouteNumber: CHAR(3)**)
- Minimal Key: {Stop, RouteName, RouteNumber}

Tables not already in 3NF

Determining if BusModel is in 3NF:

BusModel(Name: CHAR(20), Cost: INT, Capacity: INT, FuelType: CHAR(6), PurchaseCost: INT, OperatingCost: INT)

BusModel has FDs:
- Name → Cost, Capacity, FuelType, PurchaseCost, OperatingCost
- PurchaseCost, OperatingCost → Cost

{Name}+ = {Name, Cost, Capacity, FuelType, PurchaseCost, OperatingCost}
{PurchaseCost, OperatingCost}+ = {PurchaseCost, OperatingCost, Cost}

BusModel has **minimal key: {Name}**

PurchaseCost, OperatingCost → Cost **violates 3NF**

Decomposing into 3NF using a minimal cover and 3NF synthesis:

Minimal Cover of FDs:
- Name → Capacity
- Name → FuelType

- Name → PurchaseCost
- Name → OperatingCost
- PurchaseCost, OperatingCost → Cost

(Difference from original FDs: Name → Cost was removed)

Merging right hand side of minimal cover so we form smaller relations that are still lossless and dependency preserving:

- Name → Capacity, FuelType, PurchaseCost, OperatingCost
- PurchaseCost, OperatingCost → Cost

The final set of relations:

- BusModel1(Name: CHAR(20), Capacity: INT, FuelType: CHAR(6), **PurchaseCost: INT**, **OperatingCost: INT**)
- BusModel2(PurchaseCost, OperatingCost, Cost)

Determining if SkytrainModel is in 3NF:

SkytrainModel(Name: CHAR(20), Cost: INT, Capacity: INT, Cars: INT, PurchaseCost: INT, OperatingCost: INT)

SkytrainModel has FDs:

- Name → Cost, Capacity, Cars, PurchaseCost, OperatingCost
- PurchaseCost, OperatingCost → Cost

{Name}+ = {Name, Cost, Capacity, Cars, PurchaseCost, OperatingCost}
{PurchaseCost, OperatingCost}+ = {PurchaseCost, OperatingCost, Cost}

SkytrainModel has **minimal key: {Name}**

PurchaseCost, OperatingCost → Cost **violates 3NF**

Decomposing into 3NF using a minimal cover and 3NF synthesis:

Minimal Cover of FDs:

- Name → Capacity
- Name → Cars
- Name → PurchaseCost
- Name → OperatingCost
- PurchaseCost, OperatingCost → Cost

(Difference from original FDs: Name → Cost was removed)

Merging right hand side of minimal cover so we form smaller relations that are still lossless and dependency preserving:

- Name → Capacity, Cars, PurchaseCost, OperatingCost
- PurchaseCost, OperatingCost → Cost

The final set of relations:

- SkytrainModel1(Name: CHAR(20), Capacity: INT, Cars: INT, **PurchaseCost: INT**, **OperatingCost: INT**)
- SkytrainModel2(PurchaseCost: INT, OperatingCost: INT, Cost)

Determining if Route is in 3NF:

Route(Name: CHAR(50), Number: CHAR(3), **Origin: CHAR(10)**, **Destination: CHAR(10)**, Distance: REAL, BusRouteType: CHAR(20), RailType: CHAR(20))

Route has FDs:

- Name, Number → Origin, Destination, Distance, BusRouteType, RailType
- Number → BusRouteType

{Name, Number}+ = {Name, Number, Origin, Destination, Distance, BusRouteType, RailType}
{Number}+ = {Number, BusRouteType}

Route has **minimal key: {Name, Number}**

Number → BusRouteType **violates 3NF**

Decomposing into 3NF using a minimal cover and 3NF synthesis:

Minimal Cover of FDs:

- Name, Number → Origin
- Name, Number → Destination
- Name, Number → RailType
- Name, Number → Distance
- Number → BusRouteType

(Difference from original FDs: Name, Number → BusRouteType was removed)

Merging right hand side of minimal cover so we form smaller relations that are still lossless and dependency preserving:

- Name, Number → Origin, Destination, Distance, RailType
- Number → BusRouteType

The final set of relations:

- Route1(Name: CHAR(50), **Number: CHAR(3), Origin: CHAR(10)**, **Destination: CHAR(10)**, Distance: REAL, RailType: CHAR(20))
- Route2(Number:CHAR(3), BusRouteType: CHAR(20))

SQL DDL statements

```
CREATE TABLE Bus(
      id CHAR(10),
      license_plate CHAR(6),
      model CHAR(20),
      route_name CHAR(50),
      route_number CHAR(3),
      PRIMARY KEY (id),
      UNIQUE (license_plate),
      FOREIGN KEY (model) REFERENCES BusModel1 (name)
            ON DELETE SET NULL
            ON UPDATE CASCADE,
      FOREIGN KEY (route_name, route_number) REFERENCES Route1 (name, number)
            ON DELETE CASCADE
            ON UPDATE CASCADE
);

CREATE TABLE Driver(
      id CHAR(8),
      first_name CHAR(30),
      last_name CHAR(30),
      PRIMARY KEY (id)
);

CREATE TABLE SkyTrain(
      id CHAR(3),
      route_name CHAR(50),
      route_number CHAR(3)
      model CHAR(20),
      PRIMARY KEY (id),
      FOREIGN KEY (model) REFERENCES SkytrainModel1 (name)
            ON DELETE SET NULL
            ON UPDATE CASCADE,
      FOREIGN KEY (route_name, route_number) REFERENCES Route1 (name, number)
            ON DELETE CASCADE
```

```
            ON UPDATE CASCADE

);

CREATE TABLE SkyTrainStation(
      stopid CHAR(10),
      name CHAR(50),
      platforms INT,
      PRIMARY KEY (stopid, name),
      FOREIGN KEY (stopid) REFERENCES Stop (id)
            ON DELETE CASCADE
            ON UPDATE CASCADE
);

CREATE TABLE BusStop(
      stopid CHAR(10),
      name CHAR(50),
      PRIMARY KEY (stopid, name),
      FOREIGN KEY (stopid) REFERENCES Stop (id)
            ON DELETE CASCADE
            ON UPDATE CASCADE
);

CREATE TABLE Stop(
      id CHAR(10),
      postcode CHAR(6),
      city CHAR(20),
      PRIMARY KEY (id),
      FOREIGN KEY (city) REFERENCES Zone (city)
            ON DELETE CASCADE
            ON UPDATE CASCADE
);

CREATE TABLE Zone(
      number INT,
      city CHAR(20),
      PRIMARY KEY (city),
);

CREATE TABLE CompassTap(
      card_id CHAR(20),
      time INT,
      stop CHAR(10),
      PRIMARY KEY (card_id, time, stop),
```

```
      FOREIGN KEY (stop) REFERENCES Stop (id)
            ON DELETE CASCADE
            ON UPDATE CASCADE
);

CREATE TABLE DriverAssignment(
      driver_id CHAR(8),
      bus_id CHAR(10),
      PRIMARY KEY (driver_id, bus_id),
      FOREIGN KEY (driver_id) REFERENCES Driver (id)
            ON DELETE CASCADE
            ON UPDATE CASCADE,
      FOREIGN KEY (bus_id) REFERENCES Bus (id)
            ON DELETE CASCADE
            ON UPDATE CASCADE
);

CREATE TABLE AvailableStop(
      stop CHAR(10),
      route_name CHAR(50),
      route_number CHAR(3)
      PRIMARY KEY (stop, route_name, route_number),
      FOREIGN KEY (stop) REFERENCES Stop (id)
            ON DELETE CASCADE
            ON UPDATE CASCADE,
      FOREIGN KEY (route_name, route_number) REFERENCES Route1 (name, number)
            ON DELETE CASCADE
            ON UPDATE CASCADE
);

CREATE TABLE BusModel1(
      name CHAR(20),
      capacity INT,
      fuel_type CHAR(5),
      purchase_cost INT,
      operating_cost INT,
      PRIMARY KEY (name),
      FOREIGN KEY (purchasing_cost, operating_cost) REFERENCES BusModel2
                  (purchasing_cost, operating_cost)
);

CREATE TABLE BusModel2(
      purchase_cost INT,
      operating_cost INT,
```

```
      cost INT,
      PRIMARY KEY (purchase_cost, operating_cost)
);

CREATE TABLE SkyTrainModel1(
      name CHAR(20),
      capacity INT,
      cars INT,
      purchase_cost INT,
      operating_cost INT,
      PRIMARY KEY (name),
      FOREIGN KEY (purchasing_cost, operating_cost) REFERENCES SkytrainModel2
                  (purchasing_cost, operating_cost)
);

CREATE TABLE SkyTrainModel2(
      purchase_cost INT,
      operating_cost INT,
      cost INT,
      PRIMARY KEY (purchase_cost, operating_cost)
);

CREATE TABLE Route1(
      name CHAR(50),
      number CHAR(3),
      origin CHAR(10),
      destination CHAR(10),
      rail_type CHAR(20),
      distance REAL,
      PRIMARY KEY (name, number),
      FOREIGN KEY (origin) REFERENCES Stop (id),
      FOREIGN KEY (destination) REFERENCES Stop (id),
      FOREIGN KEY (number) REFERENCES Route2(number)
);

CREATE TABLE Route2(
      number CHAR(3),
      bus_route_type CHAR(20),
      PRIMARY KEY (number)
);
```

SQL insert statements

```
INSERT INTO Bus
VALUES ('18022', 'NG5745', 'NFI XDE60', '41st Ave', 'R4'),
       ('9409', 'BY4048', 'Nova Bus LFS HEV', 'Brentwood Station/UBC', '25'),
       ('19027', 'NN9907', 'NFI XDE60', 'Marine Dr', 'R2'),
       ('16047', 'LG4805', 'NFI XN40', 'White Rock Centre/Willowbrook',
'531'),
       ('9660', 'KX3049', 'Nova Bus LFS', '29th Avenue Stn/Arbutus', '16');

INSERT INTO Driver
VALUES ('28903460', 'Derek', 'Lee'),
       ('90020439', 'Cynthia', 'Newman'),
       ('18834306', 'Evan', 'Holmes'),
       ('44947234', 'Lan', 'Duong'),
       ('63057871', 'Aman', 'Shah');

INSERT INTO SkyTrain
VALUES ('111', 'Canada Line', 'CL', 'Hyundai Rotem EMU'),
       ('337', 'Expo Line', 'EL', 'Bombardier ART Mark II'),
       ('52', 'Expo Line', 'EL', 'Bombardier ICTS Mark I'),
       ('144', 'Millennium Line', 'ML', 'Bombardier ICTS Mark I'),
       ('219', 'Canada Line', 'CL', 'Hyundai Rotem EMU');

INSERT INTO SkyTrainStation
VALUES ('BR', 'Bridgeport', 2),
       ('ST', 'Stadium-Chinatown', 3),
       ('CO', 'Columbia', 2),
       ('LG', 'Langara-49th Avenue', 2),
       ('BW', 'Commercial-Broadway', 5);

INSERT INTO BusStop
VALUES ('50904', 'Westbound E Broadway @ Rupert St'),
       ('50590', 'Westbound W 4th Ave @ Alma St'),
       ('61304', 'Eastbound Lougheed Hwy @ Harris Rd'),
       ('50545', 'Southbound Granville St @ Angus Dr'),
       ('50136', 'Oakridge-41st Ave Station @ Bay 2');

INSERT INTO Stop
VALUES ('BW', 'V5N4B9', 'Vancouver'),
       ('61304', 'V3Y2J4', 'Pitt Meadows'),
       ('ST', 'V6B2L3', 'Vancouver'),
       ('50136', 'V5Z0E3', 'Vancouver'),
       ('56474', 'V6X3M2', 'Richmond');
```

```
INSERT INTO Zone
VALUES (1, ‘Vancouver’),
       (2, ‘Richmond’),
       (3, ‘Maple Ridge’),
       (2, ‘Burnaby’),
       (3, ‘Surrey’);

INSERT INTO CompassTap
VALUES ('53426693990928605252', 35263, '61304'),
       ('32761835554193258185', 32007, 'LG'),
       ('29700392471242162660', 71191, '50545'),
       ('11327375108351674279', 80104, 'LG'),
       ('32761835554193258185', 67420, 'BR');



INSERT INTO DriverAssignment
VALUES ('44947234', '9660'),
       ('63057871', '16047'),
       ('90020439', '18022'),
       ('18834306', '19027'),
       ('28903460', '9409');

INSERT INTO AvailableStop
VALUES ('50136', '41st Ave', 'R4'),
       ('61772', 'Marine Dr', 'R2'),
       ('51862', 'Burquitlam Station/SFU', '143'),
       ('50627', 'Robson/Downtown', '5'),
       ('LG', 'Canada Line', 'CL');

INSERT INTO BusModel1
VALUES ('NFI XN40', 70, 'N Gas', 500000, 143),
 ('NFI XDE60', 120, 'D-E', 1200000, 522),
 ('Nova Bus LFS', 70, 'Diesel', 700000, 433),
 ('Nova Bus LFS HEV', 70, 'D-E', 700000, 302),
 ('Nova Bus LFSe', 70, 'B-E', 400000, 92);

INSERT INTO BusModel2
VALUES (500000, 143, 19000),
       (1200000, 522, 1269000),
       (700000, 433, 760000),
       (700000, 302, 740000),
       (400000, 92, 412000);
```

```
INSERT INTO SkyTrainModel1
VALUES ('Bombardier ICTS Mark I', 80, 6, 2450000, 2000),
       ('Hyundai Rotem EMU', 200, 2, 3250000, 500),
       ('Bombardier ART Mark II', 130, 4, 3300000, 1125),
       ('Bombardier ART Mark III', 135, 4, 3500000, 1000),
       ('Alstom Mark V', 160, 3500000, 750);

INSERT INTO SkyTrainModel2
VALUES (2450000, 2000, 3180000),
       (3500000, 1000, 3865000),
       (3250000, 500, 3432500),
       (3300000, 1125, 3660625),
       (3500000, 750, 3773750);



INSERT INTO Route1
VALUES ('Commercial-Broadway Station/UBC', '99', 'BW', '60358', NULL),
       ('Canada Line', 'CL', 'WF', 'RB', 'Conventional'),
       ('29th Avenue Stn/Arbutus', '16', '51061', '59042', NULL),
       ('Expo Line', 'EL', 'WF', 'PW', 'Linear Induction'),
       ('Brentwood Station/UBC', '25', '60516', '60358', NULL);

INSERT INTO Route2
VALUES ('106', 'Diesel'),
       ('6', 'Trolley'),
       ('99', 'B-Line'),
       ('4', 'Trolley'),
       ('R4', 'RapidBus');
```