

CPSC 304 Project Cover Page

Milestone #: 2

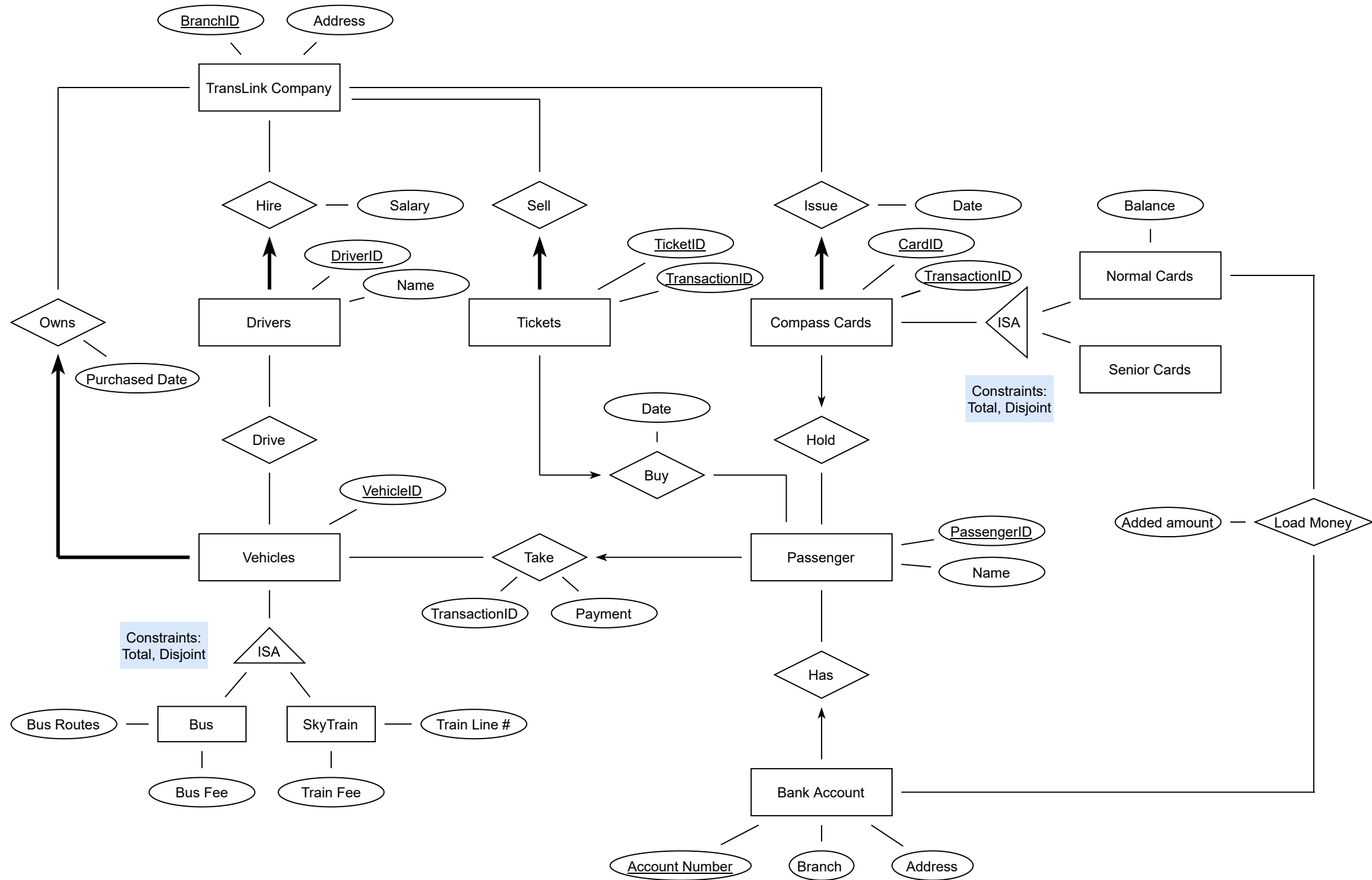
Date: 2021-10-25

Group Number: 45

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Kelvin Li	82433020	y9y1b	clay2pws@gmail.com
Jin Niu	73491458	b3w2b	dkiv9570@163.com
Mina Yang	88523360	c7x2b	yangybio@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



1. ER Diagram Comments (revised since last milestone):

1. Bus now has Bus Routes that determines Bus Fee, and SkyTrain now has TrainLine that determines Train Fee; these two attributes were added in order to fulfill FD requirements.
2. Bank Account now has an Address, which will be determined by its registered Branch, this additional FD is also added to fulfill the FD requirements.
3. The “TransactionID” in relationship “Take” is now corrected as an attribute. (We mistakenly underlined it in the ERD of previous milestone).
4. The ‘Buy’ relationship is now corrected as an M:1 relationship since it makes more sense.

2. Derived Schema:

vehicleOwns (VehicleID: int, **BranchID**: int, PurchasedDate: date)

BranchID not null.

Vehicles have to be either a bus or a Skytrain, not both and not neither. (Total & disjoint).

bus (VehicleID: int, BusRoutes: int, BusFee: int)

BusFee has default value: 1.

skyTrain (VehicleID: int, TrainLine: int, TrainFee: int)

TrainFee has default value: 3.

drivers(DriverID: int, Name: char(25), **BranchID**: int, Salary: int)

BranchID not null.

Salary has a default value: 2000.

drive (**DriverID**: int, VehicleID: int)

TranslinkCompany (BranchID: int, Address: char(30))

TicketSellBuy (TicketID: int, TransactionID: int, **PassengerID**: int, **BranchID**: int, Date: date)

BranchID is not null

PassengerTake (PassengerID : int, Name : char(10), TransactionID : int, Payment: int, VehicleID: int)

BankAccount (AccountNumber : int, Branch : int, Address : char(40), **PassengerID**: int)

compassCardIssueAndHold (TransactionID: int, CardID: int, Date: date, **BranchID**: int, **PassengerID**: int)

BranchID not null.

normalCard (TransactionID: int, CardID: int, Balance: int)

seniorCard (TransactionID: int, CardID: int)

loadMoney (AddedAmount: int, AccountNumber: int, CardID: int, TransactionID: int)

3. Functional Dependency:

vehicleOwns:

VehicleID ==> BranchID, PurchasedDate (primary key FD)

bus:

VehicleID ==> BusRoutes, BusFee (primary key FD)

BusRoutes ==> BusFee (Violate 3NF & BCNF, used in decomposition)

skyTrain:

VehicleID ==> TrainLine, TrainFee (primary key FD)

TrainLine ==> TrainFee (Violate 3NF & BCNF, used in decomposition)

drivers:

DriverID ==> Name, BranchID, Salary (primary key FD)

drive:

DriverID, VehicleID ==> DriverID, VehicleID (primary key FD)

TranslinkCompany:

BranchID ==> Address (Primary key FD)

TicketSellBuy:

TransactionID, CardID ==> Date, BranchID, PassengerID (primary key FD)

PassengerTake:

PassengerID, VehicleID, ==> Name, TransactionID, Payment (Primary key FD)

BankAccount :

AccountNumber \implies Branch, Address, PassengerID (Primary key FD)

Branch \implies Address (Violate 3NF & BCNF, used in decomposition)

compassCardIssueAndHold:

TransactionID, CardID \implies Date, BranchID, PassengerID (primary key FD)

normalCard:

TransactionID, CardID \implies Balance (primary key FD)

seniorCard:

TransactionID, CardID \implies TransactionID, CardID (primary key FD)

loadMoney:

AccountNumber, CardID, TransactionID \implies AddedAmount (primary key FD)

4. Decomposition for Normalization:

bus (VehicleID, BusRoutes, BusFee)

\implies [BusRoutes \Rightarrow BusFee] \implies

R1(VehicleID, BusRoutes),

R2(BusRoutes, BusFee)

skyTrain (VehicleID, TrainLine, TrainFee)

\implies [TrainLine \Rightarrow TrainFee] \implies

R1(VehicleID, TrainLine),

R2(TrainLine, TrainFee)

BankAccount(AccountNumber, Branch, Address, PassengerID)

\implies [Branch \Rightarrow Address] \implies

R1 (AccountNumber, Branch, PassengerID),

R2 (Branch, Address)

5. Normalization:

vehicleOwns (VehicleID: int, **BranchID**: int, PurchasedDate: date)

BranchID not null.

Vehicles have to be either a bus or a Skytrain, not both and not neither. (Total & disjoint).

busRoutes (**VehicleID**: int, BusRoutes: int)

busFee (BusRoutes: int, BusFee: int)

BusFee has default value: 1.

skyTrainLine (**VehicleID**: int, TrainLine: int)

skyTrainFee (TrainLine: int, TrainFee: int)

TrainFee has default value: 3.

drivers(DriverID: int, Name: char(25), **BranchID**: int, Salary: int)

BranchID not null.

Salary has a default value: 2000.

drive (**DriverID**: int, **VehicleID**: int)

TranslinkCompany (BranchID:int, Address: char(30))

TicketSellBuy (TicketID: int,TransactionID: int, **PassengerID**: int, **BranchID**:int, Date: date)

BranchID is not null

PassengerTake (PassengerID: int, Name : char(10), TransactionID : int, Payment: int, **VehicleID**:int)

PassengerAccount (AccountNumber: int, Branch : int, **PassengerID**:int)

BranchAddress (Branch: int, Address: char(40))

compassCardIssueAndHold (TransactionID: int, CardID: int, Date: date, **BranchID**: int, **PassengerID**: int)

BranchID not null.

normalCard (**TransactionID**: int, **CardID**: int, Balance: int)

seniorCard (TransactionID: int, CardID: int)

loadMoney (AddedAmount: int, AccountNumber: int, CardID: int, TransactionID: int)

6. SQL DDL:

```
CREATE TABLE vehicleOwns(  
    VehicleID integer PRIMARY KEY,  
    BranchID integer NOT NULL,  
    PurchasedDate date,  
    FOREIGN KEY(BranchID) REFERENCES TranslinkCompany  
        ON DELETE CASCADE  
        ON UPDATE CASCADE);
```

```
CREATE TABLE busRoutes (  
    VehicleID integer PRIMARY KEY,  
    BusRoutes integer,  
    FOREIGN KEY(VehicleID) REFERENCES vehicleOwns  
        ON DELETE CASCADE  
        ON UPDATE CASCADE);
```

```
CREATE TABLE busFee (  
    BusRoutes integer PRIMARY KEY,  
    BusFee integer DEFAULT 1);
```

```
CREATE TABLE skyTrainLine (  
    VehicleID integer PRIMARY KEY,
```

TrainLine integer,
FOREIGN KEY(VehicleID) REFERENCES vehicleOwns
ON DELETE CASCADE
ON UPDATE CASCADE);

CREATE TABLE skyTrainFee (
TrainLine integer PRIMARY KEY,
TrainFee integer DEFAULT 3);

CREATE TABLE drivers (
DriverID integer PRIMARY KEY,
Name char(25),
BranchID integer NOT NULL,
Salary integer DEFAULT 2000,
FOREIGN KEY(BranchID) REFERENCES TranslinkCompany
ON DELETE CASCADE
ON UPDATE CASCADE);

CREATE TABLE drive (
DriverID integer,
VehicleID integer,
PRIMARY KEY(DriverID, VehicleID),
FOREIGN KEY(DriverID) REFERENCES drivers
ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY(VehicleID) REFERENCES vehicleOwns

ON DELETE CASCADE
ON UPDATE CASCADE);

CREATE TABLE TranslinkCompany(
BranchID integer PRIMARY KEY,
Address char(30));

CREATE TABLE TicketSellBuy(
BranchID integer NOT NULL,
TicketID integer,
TransactionID integer,
PassengerID integer,
PRIMARY KEY(TicketID, TransactionID),
FOREIGN KEY(BranchID) REFERENCES TranslinkCompany
ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY(PassengerID) REFERENCES Passenger
ON DELETE CASCADE
ON UPDATE CASCADE);

CREATE TABLE PassengerTake(
PassengerID integer,
Name char(10),
TransactionID integer,
Payment integer,
VehicleID integer,

PRIMARY KEY(PassengerID, VehicleID),
FOREIGN KEY(VehicleID) REFERENCES Vehicles
ON DELETE CASCADE
ON UPDATE CASCADE);

CREATE TABLE PassengerAccount (
AccountNumber integer PRIMARY KEY,
Branch integer,
PassengerID integer,
FOREIGN KEY(PassengerID) REFERENCES Passenger
ON DELETE CASCADE
ON UPDATE CASCADE);

CREATE TABLE BranchAddress (
Branch integer PRIMARY KEY,
Address char(40));

CREATE TABLE compassCardIssueAndHold (
TransactionID integer,
CardID integer,
Date date,
BranchID integer NOT NULL,
PassengerID integer,
PRIMARY KEY(TransactionID, CardID),
FOREIGN KEY(BranchID) REFERENCES TranslinkCompany
ON DELETE CASCADE

ON UPDATE CASCADE,
FOREIGN KEY(PassengerID) REFERENCES Passenger);

```
CREATE TABLE normalCard (  
    TransactionID integer,  
    CardID integer,  
    Balance integer,  
    PRIMARY KEY(TransactionID, CardID),  
    FOREIGN KEY(TransactionID, CardID) REFERENCES compassCardIssueAndHold  
(TransactionID, CardID)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE );
```

```
CREATE TABLE seniorCard (  
    TransactionID integer,  
    CardID integer,  
    PRIMARY KEY(TransactionID, CardID),  
    FOREIGN KEY(TransactionID, CardID) REFERENCES compassCardIssueAndHold  
(TransactionID, CardID)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE );
```

```
CREATE TABLE loadMoney (  
    AddedAmount integer,  
    NormalCardID integer,  
    AccountNumber integer,  
    TransactionID integer,
```

PRIMARY KEY(AccountNumber, NormalCardID, TransactionID),

FOREIGN KEY(TransactionID, NormalCardID) REFERENCES normalCard
(TransactionID, CardID)

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY(AccountNumber) REFERENCES bankAccount

ON DELETE CASCADE

ON UPDATE CASCADE);

7. Populate Tables:

An example of insertion:

```
INSERT INTO busFee (BusRoutes, BusFee)
VALUES ('4', '1');
```

Other data insertions are represented in the following screenshots:

vehicleOwns			busRoutes		busFee			
VehicleID	BranchID	PurchasedDate	VehicleID	BusRoutes	BusRoutes	BusFee		
66611	1	2007/1/1	66611	4	4	1		
66612	1	2007/1/2	66612	14	14	1		
66613	1	2007/1/3	66613	99	99	2		
66614	1	2007/1/4	66614	41	41	1		
66615	2	2007/1/5	66615	33	33	1		
66616	2	2007/1/6						
66617	2	2007/1/7						
66618	3	2007/1/8						
66619	3	2007/1/9						
66620	3	2007/1/10						
skyTrainLine			skyTrainFee		drivers			
VehicleID	TrainLine		TrainLine	TrainFee	DriverID	Name	BranchID	Salary
66616	17		17	3	55511	Smith	1	2100
66617	27		27	3	55512	John	1	2200
66618	37		37	5	55513	Liz	1	2300
66619	47		47	5	55514	Thomas	2	2200
66620	57		57	5	55515	Beth	2	2000
drive								
DriverID	VehicleID							
55511	66611							
55511	66614							
55511	66615							
55512	66612							
55512	66615							
55513	66613							
55513	66615							
55514	66614							
55515	66615							

TranslinkCompany			TicketSell		
BranchID	Address		BranchID	TicketID	TransactionID
1	2222 King Rd		1	1	1
2	3980 SE Lemon rd		2	2	2
3	1111 SW Apple Dr		3	3	3
4	2233 B Dr		4	4	4
5	5577 C Dr		5	5	5
Buy					
TicketID	TransactionID	Date	PassengerID		
1	1	1/1/2007	1		
2	2	1/2/2007	2		
3	3	1/3/2007	3		
4	4	1/4/2007	4		
5	5	1/5/2007	5		
PassengerTake					
PassengerID	Name	TransactionID	Payment	VehicleID	
1	Tom	1	10	11	
2	Jack	2	10	22	
3	Mary	3	10	33	
4	David	4	10	44	
5	Ben	5	10	55	
PassengerAccount					
AccountNumber	Branch	PassengerID			
11001110		1			
11001111		2			
11001112		3			
11001113		4			
11001114		5			
BranchAddress					
Branch	Address				
	1 2200 King Rd				
	2 3900 SE Lemon rd				
	3 1100 SW Apple Dr				
	4 2200 B Dr				
	5 5500 C Dr				

Add a senior card into seniorCard database.

Add a Bank Account into the Bank Account database.
Add a load money with Bank Account and Normal card into LoadMoney database.

Deletion:

Delete a vehicle from the vehicleOwns database.
Delete a bus from the busRoutes database.
Delete a skytrain from the skyTrainLine database.
Delete a driver from the drivers database.
Delete a combination of driver and vehicle from the drive database.
Delete a combination of passenger and vehicle from the passengerTake database.
Delete a combination of ticket and passenger from the Buy database.
Delete a compass card with its TransLink company and its owner from the compassCardIssueAndHold database.
Delete a normal card from normalCard database.
Delete a senior card from seniorCard database.
Delete a Bank Account from the Bank Account database.

Delete a load money with Bank Account and Normal card from LoadMoney database.

Update:

Update a vehicle's BranchID in the vehicleOwns database.
Update a vehicle's purchased date in the vehicleOwns database.
Update a bus's registered route in the busRoutes database.
Update a bus's fee in the busRoutes database.
Update a skytrain's registered skytrain line in the skyTrainLine database.
Update a skytrain's fee in the skyTrainLine database.
Update a driver's Name in the drivers database.
Update a driver's BranchID in the drivers database.
Update a driver's Salary in the drivers database.
Update a compass card's card ID in the compassCardIssueAndHold database.
Update a compass card's Transaction ID in the compassCardIssueAndHold database.
Update a compass card's issue date in the compassCardIssueAndHold database.
Update a compass card's issue company branch ID in the compassCardIssueAndHold database.
Update a compass card's owner ID in the compassCardIssueAndHold database.
Update a normal card's Transaction ID in the normalCard database.
Update a normal card's card ID in the normalCard database.
Update a normal card's balance in the normalCard database.
Update a senior card's Transaction ID in the normalCard database.
Update a senior card's card ID in the normalCard database.
Update a bank account number which loads money to the card in the LoadMoney database.
Update a normal card ID which is loaded money into in the LoadMoney database.
Update the added amount of money in the LoadMoney database.
Update the BranchID of Translink Company in the TranslinkCompany database.
Update the Address of Translink Company in the TranslinkCompany database.
Update the BranchID of Translink Company in the TicketSell database.
Update the TicketID of Ticket in the TicketSell database.

Update the TransactionID of the ticket in the TicketSell database.
Update the PassengerID of passenger in the PassengerTake database.
Update the Name of passenger in the PassengerTake database.
Update the VehicleID of vehicle in the PassengerTake database.
Update the amount of payment of passenger in the PassengerTake database.
Update the AccounterNumber in the BankAccount database.
Update the Branch of the bank in the BankAccount database.
Update the Address of bank in the BankAccount database.
Update the PassengerID of a passenger in the BankAccount database.

Selection:

Select a vehicle from the vehicleOwns database.
Select a bus from the busRoutes database.
Select a bus route's fee from the busFee database.
Select a skytrain from the skyTrainLine database.
Select a skytrain line's fee from the skyTrainFee database.
Select a driver from the drivers database.
Select a combination of driver and vehicle from the drive database.
Select a compass card with its TransLink company and its owner from the compassCardIssueAndHold database.
Select a normal card from normalCard database.
Select a senior card from seniorCard database.
Select a load money with Bank Account and Normal card from LoadMoney database.
Select a BranchID of Translink Company in the TranslinkCompany database.
Select a Address of Translink Company in the TranslinkCompany database.
Select a BranchID of Translink Company in the TicketSell database.
Select a TicketID of Ticket in the TicketSell database.
Select a TransactionID of the ticket in the TicketSell database.
Select a PassengerID of passenger in the PassengerTake database.
Select a Name of passenger in the PassengerTake database.
Select a VehicleID of vehicle in the PassengerTake database.
Select a amount of payment of passenger in the PassengerTake database.
Select a AccounterNumber in the BankAccount database.
Select a Branch of the bank in the BankAccount database.
Select a Address of bank in the BankAccount database.
Select a PassengerID of a passenger in the BankAccount database.