

CPSC 304 Project Cover Page

Milestone #: 2

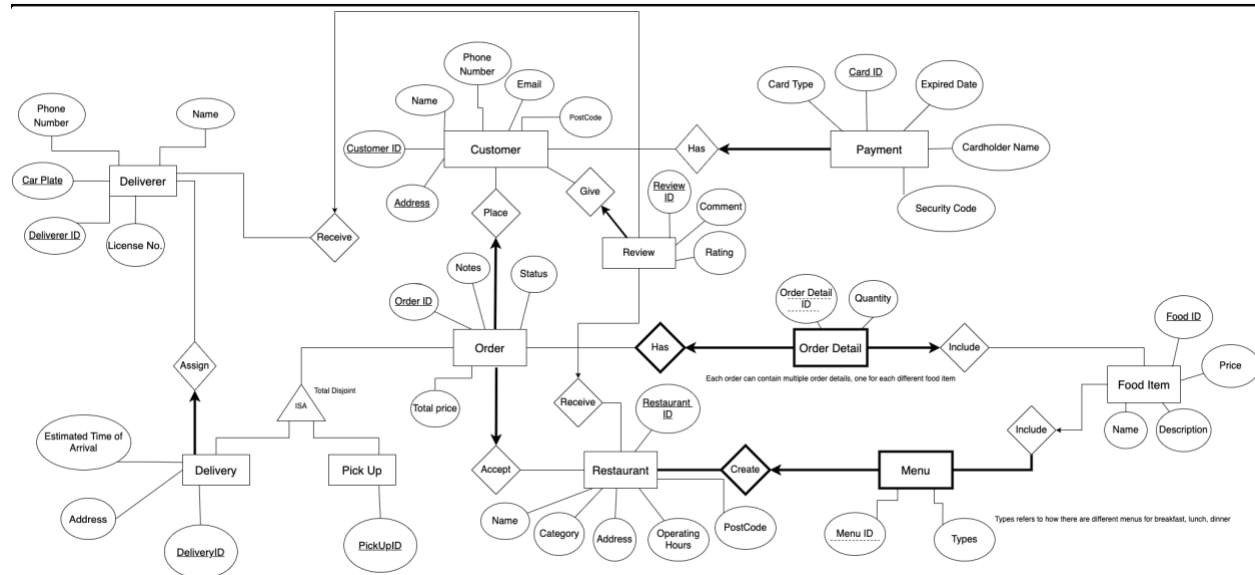
Date: March 3rd, 2021

Group Number: 25

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Annabelle Wang	83085613	c1s2b	awctw160@gmail.com
Yixin Qian	88174206	t0o1b	qyx1118@outlook.com
Iris Xu	72124175	o9f2b	xiynxiyn019@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 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



Corrections in the Diagram:

- Made Car Plate a primary key because a deliverer could have more than one car under their account
- Made OrderDetail to FoodItem a M:1 total participation relationship
- Moved attribute of assign, Estimated Time of Arrival, to be the attribute of Delivery
- Made Address and CustomerID a primary key because a customer could have more than one address
- Added PostCode as an attribute for Customer

Part 3: Schema

Customer(CustomerID: INT,
Address: VARCHAR(100),
Name: VARCHAR(50),
PhoneNumber: INT (candidate key),
Email: VARCHAR(50) (candidate key),
PostCode: CHAR(6))

Order(OrderID: INT,
RestaurantID: INT (RestaurantID can't be null),
CustomerID: INT (CustomerID can't be null),
Notes: VARCHAR(200),
Status: CHAR(10),
TotalPrice: DECIMAL)

Delivery(OrderID: INT,
DelivererID: INT (DelivererID can't be null),
DeliveryID: INT,
ETA: datetime)

PickUp(OrderID: INT,
PickUpID: INT)

Deliverer(DelivererID: INT,
LicenseNum: CHAR(10) ,
CarPlate: CHAR(10),
PhoneNumber: INT,
Name: VARCHAR(50))

Restaurant(RestaurantID: INT,
Name: CHAR(20),
Category: CHAR(20),
Address: VARCHAR(50),
PostCode: CHAR(6),
OperatingHours: INT)

Menu(MenuID: INT,
RestaurantID: INT,
Type: CHAR(20))

FoodItem(FoodID: INT,
MenuID: INT (MenuID can't be null),
Price: DECIMAL,
Description: VARCHAR(200),
Name: VARCHAR(50))

OrderDetail(OrderDetailID: INT,
OrderID: INT,
FoodID: INT (FoodID can't be null),
Quantity: INT)

Review(ReviewID: INT,
CustomerID: INT (CustomerID can't be null),
DelivererID: INT (DelivererID can't be null),
RestaurantID: INT (RestaurantID can't be null),
Comment: VARCHAR(1024),
Rating: INT)

Payment(CardID: INT,
CustomerID: INT (CustomerID can't be null),
CardType: CHAR(10),
ExpiredDate: INT,
CardHolderName: VARCHAR(50),
SecurityCode: INT)

Part 4: Functional Dependencies

Customer: Address → PostCode

CustomerID → Name, PhoneNumber, Email

PhoneNumber, Email → CustomerID

Order: OrderID → CustomerID, Address, RestaurantID, Notes, Status, TotalPrice

Delivery: DeliveryID → OrderID, CarPlate, DelivererID, Address, EstimatedTimeofArrival

Deliverer: DelivererID, CarPlate → LicenseNo, PhoneNum, Name

LicenseNo → DelivererID, Name

Restaurant: RestaurantID → Name, Category, Address, Operating Hours, PostCode

Menu: MenuID → RestaurantID, Types

FoodItem: FoodID → MenuID, Price, Description, Name

OrderDetail: OrderDetailID → OrderID, FoodID, Quantity

Review: ReviewID → CustomerID, Address, Comment, Rating, DelivererID,
RestaurantID

Payment: CardID → CustomerID, Address, CardType, ExpiredDate, CardHolderName,
SecurityCode

Part 5: Normalization

Customer:

LHS: Address

Middle: CustomerID, PhoneNumber, Email

RHS: Name, PostCode

Keys:

[Address, CustomerID]

[Address, PhoneNumber, Email]

University of British Columbia, Vancouver

Department of Computer Science

LHS is not a superkey so not in BCNF.

Address → PostCode

CustomerID → Name

CustomerID → PhoneNumber

CustomerID → Email

PhoneNumber, Email → CustomerID

Can't minimize.

Can't remove any FDs.

Address → PostCode violates 3NF. Decompose.

R1(Address, PostCode), R2(Address, CustomerID, PhoneNumber, Email, Name)

CustomerID → Name violates BCNF, so decompose.

R3(CustomerID, Name), R4(Address, CustomerID, PhoneNumber, Email)

CustomerID → PhoneNumber violates BCNF, so decompose.

R5(CustomerID, PhoneNumber), R6(Address, CustomerID, Email)

CustomerID → Email violates BCNF, so decompose.

R6(CustomerID, Email), R7(Address, CustomerID)

Therefore, R1, R3, R5, R6, R7 are now in 3NF.

Not all FDs are preserved, PhoneNumber, Email → CustomerID was lost so add

R8(PhoneNumber, Email, CustomerID).

Don't need R5, R6 since they are redundant.

So R1, R3, R7, R8

R1(Address, PostCode)

R3(CustomerID, Name)

R7(Address, CustomerID)

R8(PhoneNumber, Email, CustomerID)

Deliverer:

LHS: CarPlate

Middle: License, DelivererID

RHS: PhoneNum, Name

Keys:

University of British Columbia, Vancouver

Department of Computer Science

[CarPlate, DelivererID]

[CarPlate, LicenseNo]

LHS is not a superkey so not BCNF.

DelivererID, CarPlate → LicenseNo, PhoneNum, Name

LicenseNo → DelivererID

LicenseNo → Name

No need to minimize.

Can't remove any FDs.

LicenseNo → Name violates 3NF. Decompose.

R1(LicenseNo, Name), R2(LicenseNo, CarPlate, DelivererID, PhoneNum)

LicenseNo → DelivererID violates BCNF. Decompose.

R3(LicenseNo, DelivererID), R4(LicenseNo, CarPlate, PhoneNum)

Therefore R1, R3, R4 are now in 3NF.

Not all FDs are preserved, DelivererID, CarPlate → LicenseNo, PhoneNum, Name was lost so add R5(DelivererID, CarPlate, LicenseNo, PhoneNum, Name).

R1, R3, R4 become redundant so don't need them.

So R5 is the new relation.

R5(DelivererID, CarPlate, LicenseNo, PhoneNum, Name)

Part 6: SQL DDL

```
CREATE TABLE Customer(  
    CustomerID INT AUTO_INCREMENT ,  
    Address VARCHAR(100) NOT NULL ,  
    Name VARCHAR(50) ,  
    PhoneNumber INT UNIQUE,  
    Email VARCHAR(50) UNIQUE ,  
    PostCode CHAR(6) ,  
    PRIMARY KEY (CustomerID,Address)  
);
```

```
CREATE TABLE Order(  
    OrderID INT AUTO_INCREMENT ,  
    RestaurantID INT NOT NULL ,
```

University of British Columbia, Vancouver

Department of Computer Science

```
CustomerID INT NOT NULL ,  
Notes VARCHAR(200) ,  
Status VARCHAR(32) NOT NULL ,  
TotalPrice DECIMAL(10,2) NOT NULL ,  
PRIMARY KEY (OrderID) ,  
FOREIGN KEY (RestaurantID) REFERENCES Restaurant (RestaurantID) ,  
FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID)  
);
```

```
CREATE TABLE Delivery(  
OrderID INT ,  
DeliveryID INT AUTO_INCREMENT ,  
DelivererID INT NOT NULL ,  
ETA DATETIME ,  
PRIMARY KEY (OrderID,DeliveryID) ,  
FOREIGN KEY (OrderID) REFERENCES Order (OrderID) ,  
FOREIGN KEY (DelivererID) REFERENCES Deliverer (DelivererID)  
);
```

```
CREATE TABLE PickUp(  
OrderID INT ,  
PickUpID INT AUTO_INCREMENT ,  
PRIMARY KEY (OrderID,PickUpID) ,  
FOREIGN KEY (OrderID) REFERENCES Order (OrderID) ,  
);
```

```
CREATE TABLE Deliverer(  
DelivererID INT AUTO_INCREMENT ,  
LicenseNum CHAR(10) ,  
CarPlate CHAR(10) ,  
PhoneNumber INT ,  
Name VARCHAR(50) ,  
PRIMARY KEY (DelivererID,CarPlate) ,  
FOREIGN KEY (ReviewID) REFERENCES Review (ReviewID) ,  
);
```

```
CREATE TABLE Restaurant(  
RestaurantID INT AUTO_INCREMENT ,  
Name VARCHAR(50) ,  
Category VARCHAR(20) ,  
Address VARCHAR(50) ,  
PostCode CHAR(6) ,  
OperatingHours INT ,  
PRIMARY KEY (RestaurantID) ,  
);
```

University of British Columbia, Vancouver

Department of Computer Science

```
CREATE TABLE Menu(  
    MenuID INT AUTO_INCREMENT ,  
    RestaurantID INT ,  
    Type CHAR(20) ,  
    PRIMARY KEY (MenuID, RestaurantID) ,  
    FOREIGN KEY (RestaurantID) REFERENCES Restaurant (RestaurantID) ON  
DELETE CASCADE,  
);
```

```
CREATE TABLE FoodItem(  
    FoodID INT AUTO_INCREMENT ,  
    MenuID INT NOT NULL ,  
    Price DECIMAL(10,2) ,  
    Name VARCHAR(50) ,  
    Description VARCHAR(200) ,  
    PRIMARY KEY (FoodID) ,  
    FOREIGN KEY (MenuID) REFERENCES Menu (MenuID) ,  
);
```

```
CREATE TABLE OrderDetail(  
    OrderDetailID INT AUTO_INCREMENT ,  
    OrderID INT ,  
    FoodID INT NOT NULL ,  
    Quantity INT NOT NULL ,  
    PRIMARY KEY (OrderDetailID, OrderID) ,  
    FOREIGN KEY (OrderID) REFERENCES Order (OrderID) ON DELETE CASCADE ,  
    FOREIGN KEY (FoodID) REFERENCES FoodItem (FoodID) ,  
);
```

```
CREATE TABLE Review(  
    ReviewID INT AUTO_INCREMENT ,  
    CustomerID INT NOT NULL ,  
    RestaurantID INT NOT NULL ,  
    DelivererID INT ,  
    Comment VARCHAR(1024) ,  
    Rating INT ,  
    PRIMARY KEY (ReviewID) ,  
    FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID) ,  
    FOREIGN KEY (RestaurantID) REFERENCES Restaurant (RestaurantID) ,  
    FOREIGN KEY (DelivererID) REFERENCES Deliverer (DelivererID) ,  
);
```

```
CREATE TABLE Payment(  
    CardID VARCHAR(32),  
    CustomerID VARCHAR(32) NOT NULL ,  
    CardType CHAR(10) ,
```


University of British Columbia, Vancouver

Department of Computer Science

```
ExpiredDate INT ,
CardHolderName VARCHAR(50) ,
SecurityCode VARCHAR(32) ,
PRIMARY KEY (CardID) ,
FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID) ON DELETE
CASCADE,
);
```

Part 7: Example Tuples

Customer:

CustomerID	Address	Name	PhoneNumber	Email	PostCode
1	abc ave	John	123	a@gmail.com	V6T1Z4
2	aa street	John	234	b@gmail.com	V6T1Z4
3	bb ave	Oscar	345	c@gmail.com	V6T1Z4
4	cc ave	Mike	456	d@gmail.com	V3N0G9
5	dd ave	Maggie	567	e@gmail.com	V6M2W5

Order:

OrderID	RestaurantID	CustomerID	Notes	Status	TotalPrice
1	1	1	null	InProgress	20.05
2	1	2	something	InProgress	65.90
3	1	3	null	InProgress	40.80
4	4	4	null	Complete	32.78
5	5	5	leave at the front door	Complete	14.93

Delivery:

OrderID	DeliveryID	DelivererID	ETA
1	1	1	12:30pm
4	2	2	18:00pm

University of British Columbia, Vancouver

Department of Computer Science

7	3	3	17:00pm
8	4	4	16:00p
9	5	5	15:00pm

PickUp:

OrderID	PickupID
2	1
3	2
5	3
6	4
10	5

Deliverer:

DelivererID	LicenseNum	CarPlate	PhoneNumber	Name
1	11112222	A11B11	1234	Jack
1	11112222	A22B22	1234	Jack
1	11112222	A33B33	1234	Jack
2	33334444	A44B44	3456	Emily
3	55556666	A55B55	6789	George

Restaurant:

RestaurantID	Name	Category	Address	PostCode	OperatingHrs
1	A	hot pot	123 ave	V7M2W3	8
2	B	hot pot	234 ave	V7M2W3	8
3	C	hot pot	345 ave	V7M2W1	8
4	D	pizza	456 ave	V5M2W3	6
5	E	pizza	678 ave	V7M2W8	6

University of British Columbia, Vancouver
Department of Computer Science

Menu:

MenuID	RestaurantID	Type
1	1	breakfast
2	1	lunch
3	2	brunch
4	2	dinner
5	3	dessert

FoodItem:

FoodID	MenuID	Price	Name	Description
1	1	10.00	sandwich	absolutely tasty
2	1	15.00	instant noodle	hmm
3	2	18.50	egg benedict	hhhh
4	3	10.79	chocolate	ahhhh
5	3	4.00	milk tea	oh yeah

OrderDetail:

OrderDetailID	OrderID	FoodID	Quantity
1	1	1	3
2	1	2	5
3	1	3	3
4	2	4	1
5	2	4	2

Review:

ReviewID	CustomerID	RestaurantID	DelivererID	Comment	Rating
1	1	1	null	nice	3

University of British Columbia, Vancouver

Department of Computer Science

2	2	1	null	great	4
3	3	1	1	excellent	5
4	4	4	2	fantastic	5
5	5	5	3	disappointing	1

Payment:

CardID	CustomerID	CardType	ExpiredDate	CardHolder	SecurityCode
1111	1	visa	0827	a	123
2222	1	visa	0322	b	145
3333	2	visa	1025	c	667
4444	3	mastercard	0823	d	464
5555	4	mastercard	0628	e	256

Part 8: Example Queries

- Insertion: Add a customer to the customer list including information Customer ID, Address, Name, Phone Number, Email, Post Code
- Update: Update a customer's Phone Number
- Delete: Delete a customer from the customer list
 - This also indicates deleting all the Orders, Payments, Reviews associated with the customer
- Selection: Select all the Restaurants from the Restaurant list that are categorized as dessert
- Projection: Choose the Name tuple from the Restaurant list
- Join (Natural): Join Order list with Restaurant list that order from Restaurant A
 - Find all the Orders from Restaurant A in the Restaurant list
- Division: Find all the Orders from Customer A that order from Restaurant A, Order/Restaurant