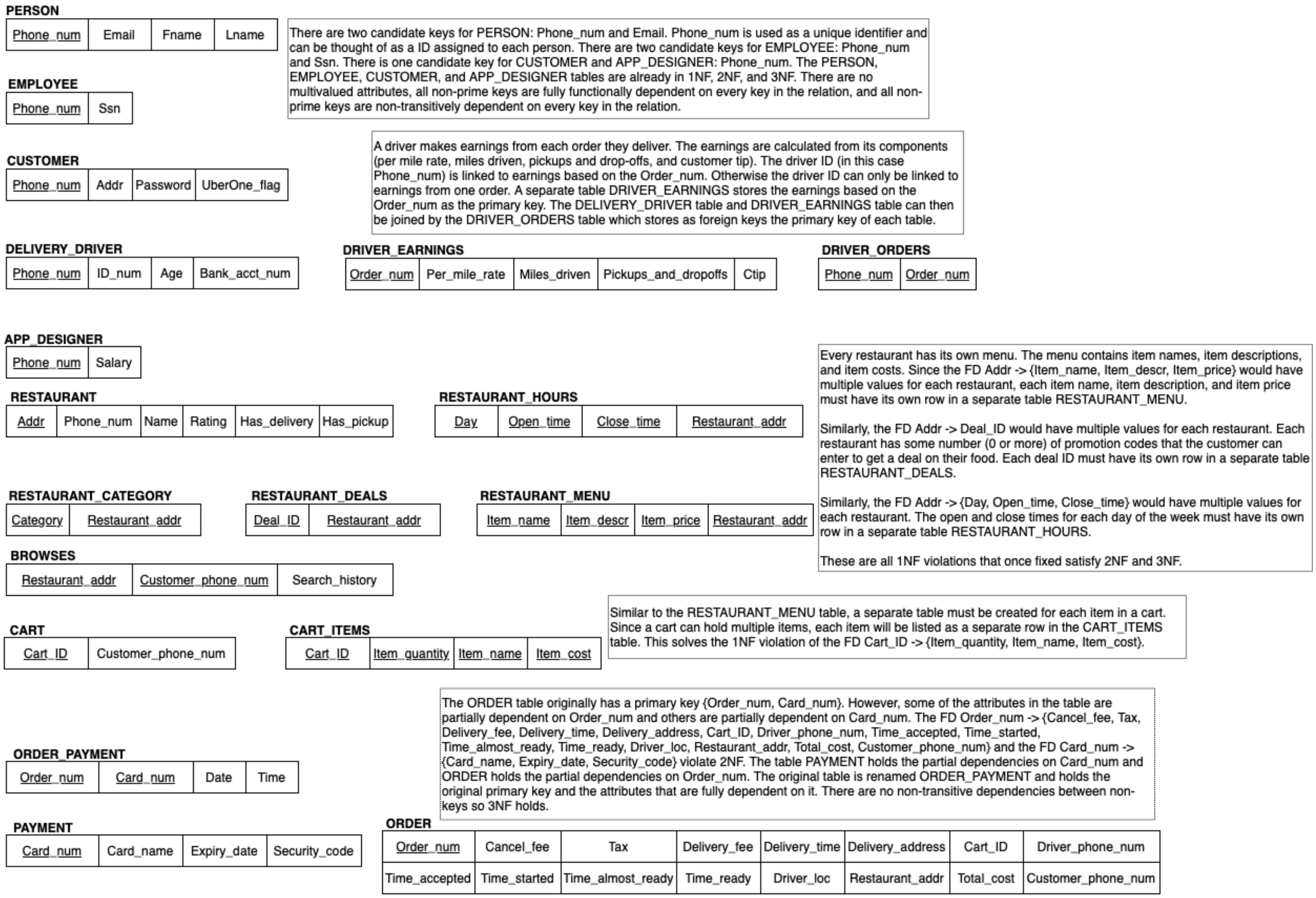
Ishwari Joshi
Database Design – Final Project

Normalized Tables in 3NF:



PERSON

Phone_num	Email	Fname	Lname

There are two candidate keys for PERSON: Phone_num and Email. Phone_num is used as a unique identifier and can be thought of as a ID assigned to each person. There are two candidate keys for EMPLOYEE: Phone_num and Ssn. There is one candidate key for CUSTOMER and APP_DESIGNER: Phone_num. The PERSON, EMPLOYEE, CUSTOMER, and APP_DESIGNER tables are already in 1NF, 2NF, and 3NF. There are no multivalued attributes, all non-prime keys are fully functionally dependent on every key in the relation, and all non-prime keys are non-transitively dependent on every key in the relation.

EMPLOYEE

Phone_num	Ssn

CUSTOMER

Phone_num	Addr	Password	UberOne_flag

A driver makes earnings from each order they deliver. The earnings are calculated from its components (per mile rate, miles driven, pickups and drop-offs, and customer tip). The driver ID (in this case Phone_num) is linked to earnings based on the Order_num. Otherwise the driver ID can only be linked to earnings from one order. A separate table DRIVER_EARNINGS stores the earnings based on the Order_num as the primary key. The DELIVERY_DRIVER table and DRIVER_EARNINGS table can then be joined by the DRIVER_ORDERS table which stores as foreign keys the primary key of each table.

DELIVERY_DRIVER

Phone_num	ID_num	Age	Bank_acct_num

DRIVER_EARNINGS

Order_num	Per_mile_rate	Miles_driven	Pickups_and_dropoffs	Ctip

DRIVER_ORDERS

Phone_num	Order_num

APP_DESIGNER

Phone_num	Salary

RESTAURANT

Addr	Phone_num	Name	Rating	Has_delivery	Has_pickup

RESTAURANT_HOURS

Day	Open_time	Close_time	Restaurant_addr

Every restaurant has its own menu. The menu contains item names, item descriptions, and item costs. Since the FD Addr -> {Item_name, Item_descr, Item_price} would have multiple values for each restaurant, each item name, item description, and item price must have its own row in a separate table RESTAURANT_MENU.

Similarly, the FD Addr -> Deal_ID would have multiple values for each restaurant. Each restaurant has some number (0 or more) of promotion codes that the customer can enter to get a deal on their food. Each deal ID must have its own row in a separate table RESTAURANT_DEALS.

Similarly, the FD Addr -> {Day, Open_time, Close_time} would have multiple values for each restaurant. The open and close times for each day of the week must have its own row in a separate table RESTAURANT_HOURS.

These are all 1NF violations that once fixed satisfy 2NF and 3NF.

RESTAURANT_CATEGORY

Category	Restaurant_addr

RESTAURANT_DEALS

Deal_ID	Restaurant_addr

RESTAURANT_MENU

Item_name	Item_descr	Item_price	Restaurant_addr

BROWSES

Restaurant_addr	Customer_phone_num	Search_history

Similar to the RESTAURANT_MENU table, a separate table must be created for each item in a cart. Since a cart can hold multiple items, each item will be listed as a separate row in the CART_ITEMS table. This solves the 1NF violation of the FD Cart_ID -> {Item_quantity, Item_name, Item_cost}.

CART

Cart_ID	Customer_phone_num

CART_ITEMS

Cart_ID	Item_quantity	Item_name	Item_cost

The ORDER table originally has a primary key {Order_num, Card_num}. However, some of the attributes in the table are partially dependent on Order_num and others are partially dependent on Card_num. The FD Order_num -> {Cancel_fee, Tax, Delivery_fee, Delivery_time, Delivery_address, Cart_ID, Driver_phone_num, Time_accepted, Time_started, Time_almost_ready, Time_ready, Driver_loc, Restaurant_addr, Total_cost, Customer_phone_num} and the FD Card_num -> {Card_name, Expiry_date, Security_code} violate 2NF. The table PAYMENT holds the partial dependencies on Card_num and ORDER holds the partial dependencies on Order_num. The original table is renamed ORDER_PAYMENT and holds the original primary key and the attributes that are fully dependent on it. There are no non-transitive dependencies between non-keys so 3NF holds.

ORDER_PAYMENT

Order_num	Card_num	Date	Time

PAYMENT

Card_num	Card_name	Expiry_date	Security_code

ORDER

Order_num	Cancel_fee	Tax	Delivery_fee	Delivery_time	Delivery_address	Cart_ID	Driver_phone_num
Time_accepted	Time_started	Time_almost_ready	Time_ready	Driver_loc	Restaurant_addr	Total_cost	Customer_phone_num

TRANSPORTATION_METHOD

Registration_no	Speed	Car_license_plate_no	Scooter_license_plate_no	Car_flag	Scooter_flag	Bicycle_flag	Foot_flag

REGISTERS

Driver_phone_num	Registration_no

CAR

Car_license_plate_no	Car_make	Car_model	Car_color	Car_year	Car_num_doors

SCOOTER

Scooter_license_plate_no	Scooter_make	Scooter_model	Scooter_color	Scooter_year	Scooter_cc

Registration_no is the primary key in the TRANSPORTATION_METHOD relation. The FD Car_license_plate_no -> {Car_make, Car_model, Car_color, Car_year, Car_num_doors} has dependencies between non-keys. Similarly, the FD Scooter_license_plate_no -> {Scooter_make, Scooter_model, Scooter_color, Scooter_year, Scooter_cc} has dependencies between non-keys. The attributes dependent on Car_license_plate_no are pulled out into a separate table CAR and the attributes dependent on Scooter_license_plate_no are pulled out into another table SCOOTER. The tables satisfy 3NF with no non-key to non-key dependencies.

REQUEST

Request_ID	Driver_loc	Restaurant_addr	Customer_addr

REQUEST_DETAILS_FOR_DRIVER

Driver_loc	Restaurant_addr	Customer_addr	Total_delivery_miles	Total_delivery_time	Payment_minus_Ctip

RECEIVES

Driver_phone_num	Request_ID	Is_driver_online

Each request has a primary key: Request_ID. The FD {Driver_loc, Restaurant_addr, Customer_addr} -> {Total_delivery_miles, Total_delivery_time, Payment_minus_Ctip} is not satisfied in 3NF and a new table REQUEST_DETAILS_FOR_DRIVER is created for this non-key to non-key dependency. 3NF does not allow the transitive dependency between the FD Request_ID -> {Driver_loc, Restaurant_addr, Customer_addr} and {Driver_loc, Restaurant_addr, Customer_addr} -> {Total_delivery_miles, Total_delivery_time, Payment_minus_Ctip} which two separate tables resolves.

CUSTOMER_SERVICE

Phone_num

CONTACTS

Customer_phone_num	Customer_service_phone_num

COMMENTS_ON

Customer_phone_num	Driver_phone_num	Restaurant_addr

COMMENTS_ON has a primary key: {Customer_phone_num, Driver_phone_num, Restaurant_addr}. Since ratings and feedback are between two entities at a time (Customer and Restaurant, Customer and Driver, Restaurant and Driver), the non-prime keys are partially dependent on part of the key which violates 2NF. The respective attributes are pulled out in separate tables with the partial keys they are dependent on. A table with the full primary key is kept as well even though no attributes are fully dependent on it.

COMMENTS_ON_C_AND_R

Customer_phone_num	Restaurant_addr	R_rating_from_C	Feedback_C_to_R

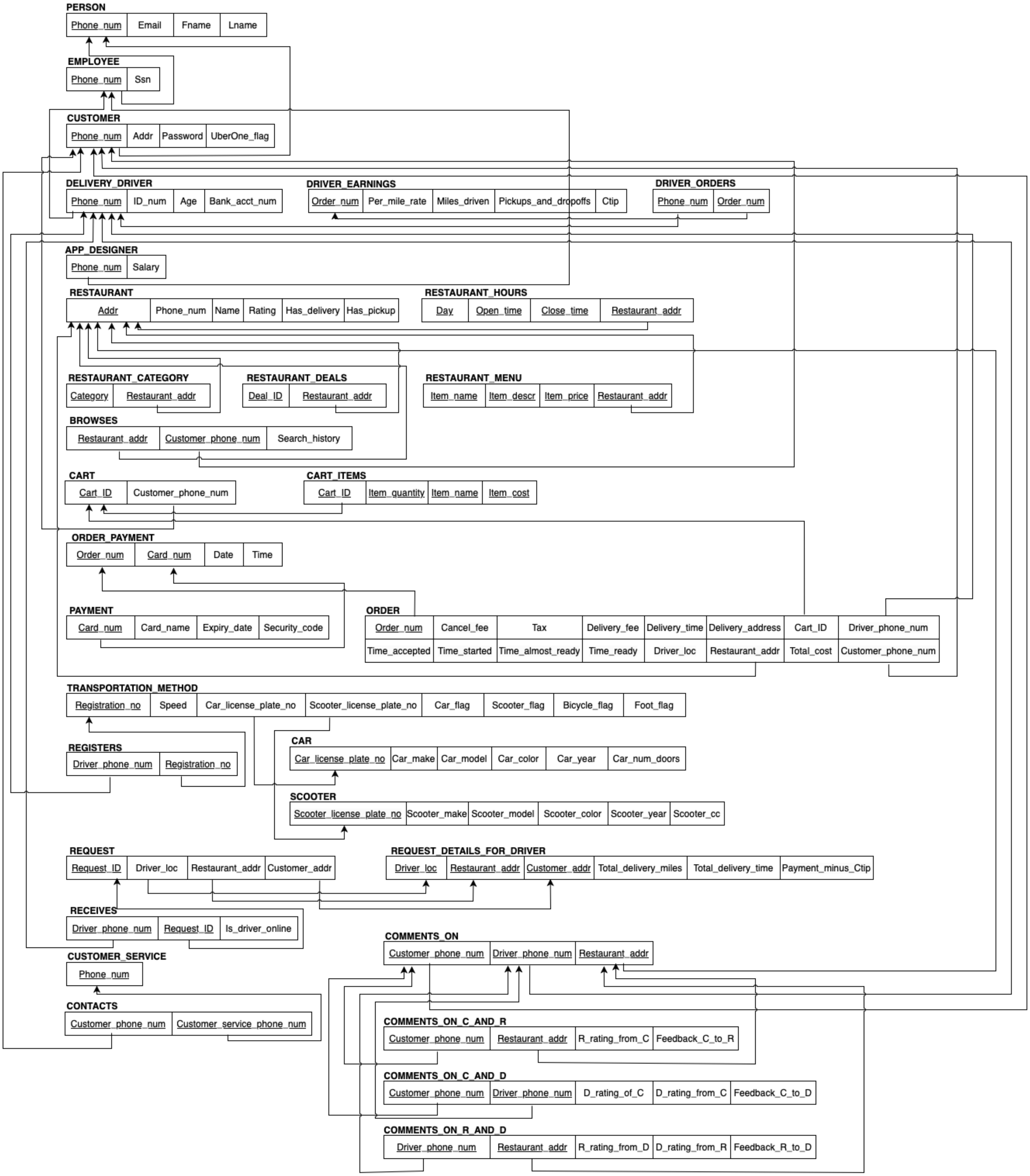
COMMENTS_ON_C_AND_D

Customer_phone_num	Driver_phone_num	D_rating_of_C	D_rating_from_C	Feedback_C_to_D

COMMENTS_ON_R_AND_D

Driver_phone_num	Restaurant_addr	R_rating_from_D	D_rating_from_R	Feedback_R_to_D

Final Relational Schema after Normalization:



PERSON

Phone_num	Email	Fname	Lname

EMPLOYEE

Phone_num	Ssn

CUSTOMER

Phone_num	Addr	Password	UberOne_flag

DELIVERY_DRIVER

Phone_num	ID_num	Age	Bank_acct_num

DRIVER_EARNINGS

Order_num	Per_mile_rate	Miles_driven	Pickups_and_dropoffs	Ctip

DRIVER_ORDERS

Phone_num	Order_num

APP_DESIGNER

Phone_num	Salary

RESTAURANT

Addr	Phone_num	Name	Rating	Has_delivery	Has_pickup

RESTAURANT_HOURS

Day	Open_time	Close_time	Restaurant_addr

RESTAURANT_CATEGORY

Category	Restaurant_addr

RESTAURANT_DEALS

Deal_ID	Restaurant_addr

RESTAURANT_MENU

Item_name	Item_descr	Item_price	Restaurant_addr

BROWSES

Restaurant_addr	Customer_phone_num	Search_history

CART

Cart_ID	Customer_phone_num

CART_ITEMS

Cart_ID	Item_quantity	Item_name	Item_cost

ORDER_PAYMENT

Order_num	Card_num	Date	Time

PAYMENT

Card_num	Card_name	Expiry_date	Security_code

ORDER

Order_num	Cancel_fee	Tax	Delivery_fee	Delivery_time	Delivery_address	Cart_ID	Driver_phone_num
Time_accepted	Time_started	Time_almost_ready	Time_ready	Driver_loc	Restaurant_addr	Total_cost	Customer_phone_num

TRANSPORTATION_METHOD

Registration_no	Speed	Car_license_plate_no	Scooter_license_plate_no	Car_flag	Scooter_flag	Bicycle_flag	Foot_flag

REGISTERS

Driver_phone_num	Registration_no

CAR

Car_license_plate_no	Car_make	Car_model	Car_color	Car_year	Car_num_doors

SCOOTER

Scooter_license_plate_no	Scooter_make	Scooter_model	Scooter_color	Scooter_year	Scooter_cc

REQUEST

Request_ID	Driver_loc	Restaurant_addr	Customer_addr

REQUEST_DETAILS_FOR_DRIVER

Driver_loc	Restaurant_addr	Customer_addr	Total_delivery_miles	Total_delivery_time	Payment_minus_Ctip

RECEIVES

Driver_phone_num	Request_ID	Is_driver_online

CUSTOMER_SERVICE

Phone_num

CONTACTS

Customer_phone_num	Customer_service_phone_num

COMMENTS_ON

Customer_phone_num	Driver_phone_num	Restaurant_addr

COMMENTS_ON_C_AND_R

Customer_phone_num	Restaurant_addr	R_rating_from_C	Feedback_C_to_R

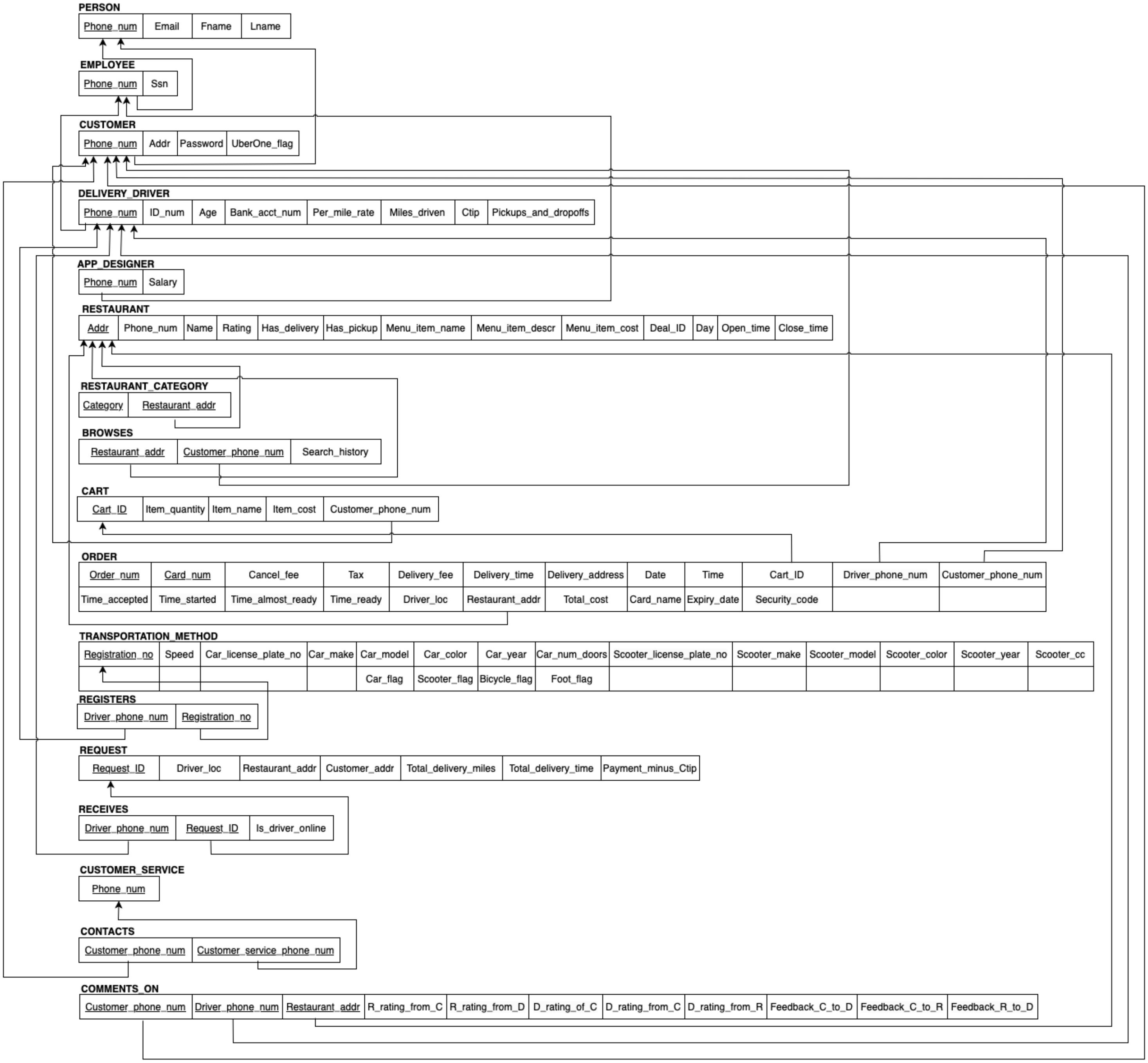
COMMENTS_ON_C_AND_D

Customer_phone_num	Driver_phone_num	D_rating_of_C	D_rating_from_C	Feedback_C_to_D

COMMENTS_ON_R_AND_D

Driver_phone_num	Restaurant_addr	R_rating_from_D	D_rating_from_R	Feedback_R_to_D

Updated Relational Schema:



PERSON

Phone_num	Email	Fname	Lname

EMPLOYEE

Phone_num	Ssn

CUSTOMER

Phone_num	Addr	Password	UberOne_flag

DELIVERY_DRIVER

Phone_num	ID_num	Age	Bank_acct_num	Per_mile_rate	Miles_driven	Ctip	Pickups_and_dropoffs

APP_DESIGNER

Phone_num	Salary

RESTAURANT

Addr	Phone_num	Name	Rating	Has_delivery	Has_pickup	Menu_item_name	Menu_item_descr	Menu_item_cost	Deal_ID	Day	Open_time	Close_time

RESTAURANT_CATEGORY

Category	Restaurant_addr

BROWSES

Restaurant_addr	Customer_phone_num	Search_history

CART

Cart_ID	Item_quantity	Item_name	Item_cost	Customer_phone_num

ORDER

Order_num	Card_num	Cancel_fee	Tax	Delivery_fee	Delivery_time	Delivery_address	Date	Time	Cart_ID	Driver_phone_num	Customer_phone_num
Time_accepted	Time_started	Time_almost_ready	Time_ready	Driver_loc	Restaurant_addr	Total_cost	Card_name	Expiry_date	Security_code		

TRANSPORTATION_METHOD

Registration_no	Speed	Car_license_plate_no	Car_make	Car_model	Car_color	Car_year	Car_num_doors	Scooter_license_plate_no	Scooter_make	Scooter_model	Scooter_color	Scooter_year	Scooter_cc
				Car_flag	Scooter_flag	Bicycle_flag	Foot_flag						

REGISTERS

Driver_phone_num	Registration_no

REQUEST

Request_ID	Driver_loc	Restaurant_addr	Customer_addr	Total_delivery_miles	Total_delivery_time	Payment_minus_Ctip

RECEIVES

Driver_phone_num	Request_ID	Is_driver_online

CUSTOMER_SERVICE

Phone_num

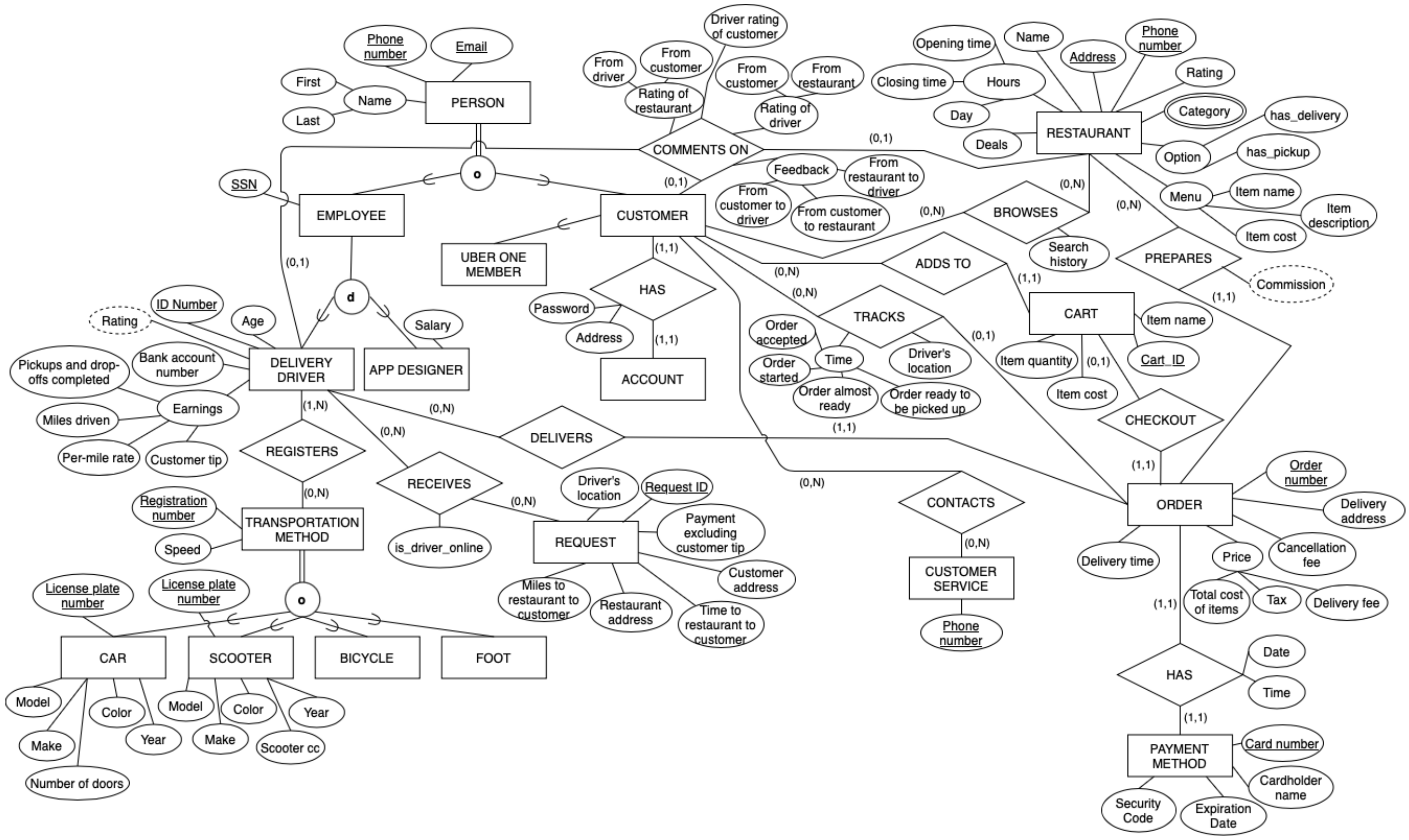
CONTACTS

Customer_phone_num	Customer_service_phone_num

COMMENTS_ON

Customer_phone_num	Driver_phone_num	Restaurant_addr	R_rating_from_C	R_rating_from_D	D_rating_of_C	D_rating_from_C	D_rating_from_R	Feedback_C_to_D	Feedback_C_to_R	Feedback_R_to_D

ER Diagram:



Data Requirements

The data requirements for an Uber Eats database system are summarized as follows:

- Uber Eats is an online food ordering and delivery platform that connects customers with delivery drivers and nearby restaurants.
- Each person has their first name, last name, their unique phone number, and unique email stored in the system. A person can be an employee, a customer, or both. In addition, each customer has to have an account on the app in which they set a password and save their address, and each employee has a unique SSN.
- A customer can be an Uber One member. A customer is not required to sign up for Uber One membership.
- An employee can be a delivery driver or an app designer. Each delivery driver has a ID number (For delivery by car and scooter, the driver has a driver's license number. For delivery by bike and foot, the driver has a government-issued ID number), age, rating, bank account number, and earnings. Earnings can be broken down into pickups and drop-offs completed, a per-mile rate, miles driven, and customer tip. Each app designer has a salary.
- A delivery driver has to register for whichever transportation methods they wish to use when signing up to be a driver in Uber's Driver app. Each transportation method

is given a unique registration number. These transportation methods include by car, scooter, bicycle, or on foot. The car or scooter's model, make, color, year, and unique license plate number must be registered. In addition, the number of doors on a car and the cc's of a scooter are required. Through the app, Uber Eats records driving speed to identify if the driver is violating the speed limit.

- Restaurants are organized by category, which is the type of cuisine. A restaurant can belong to more than one category. Each restaurant has a name, unique address, unique phone number, hours (day, opening time, and closing time), rating, and menu (item name, item description, and item cost). Some restaurants offer special deals in which the customer enters a deal ID (can be thought of as a promotion code), and some restaurants have both a delivery and a pickup option (in which the customer goes to pick up the order).
- Customers can browse participating restaurants in the app, and their search history is recorded.
- A customer adds items to a cart. A single customer can add items to multiple carts at the same time. Each cart can be added to by one customer. A cart has a unique cart ID, item quantities, item names, and item costs.
- Each order is placed by checking out from a cart. Items will remain in a cart until the order is placed. An order has a unique order number. The customer will see their delivery address, an estimated delivery time, and the price of the order including total cost of the items, tax, and delivery fee. If the customer cancels their order, they will be charged a cancellation fee.
- Each order is paid for by entering the unique card number, cardholder name, expiration date, and security code. The payment method used is linked to the date and time the order is placed.
- The restaurant prepares orders and pays commission on the orders to Uber Eats.
- The delivery driver receives delivery requests in the Driver app. The driver can turn delivery requests on and off as they wish. When the driver goes online in the app, the app will show available requests with a unique request ID, the driver's location, restaurant address, customer address, total delivery miles (miles from the driver's location to the restaurant and then to the customer address), total delivery time (time from the driver's location to the restaurant and then to the customer address), and how much the driver will get paid for the order excluding customer tip.
- The delivery driver delivers orders. Each order is assigned to one driver.
- An order can be tracked in the app by the customer. The time the order is accepted, when its preparation is started, when it is almost ready, when it is ready to be picked up by the delivery driver, and the driver's progress from the restaurant to the customer's address on the map (their current location) can be seen.

- After adding items to as many carts as desired, each cart can be checked out to place an order. In this way, a customer can place and track as many orders as they want.
- After the customer receives their order, they will get a notification from Uber Eats asking them to rate their driver and the restaurant and to leave them both feedback. The driver can rate the restaurant and the customer once they drop off the order. The restaurant can rate the driver and leave feedback. The driver's overall rating is based on the average ratings received from restaurants and customers.
- A customer can contact customer service, which is identified by a phone number. Multiple representatives may have to be contacted to solve the customer's issue.