Group#: S15
Members: Sophie Zhao, Asawari Kanitkar, Xiaorui Hu
67-262   Raja Sooriamurthi
Report on Uber Database

Uber's core mission:

Uber's three main company values has helped it become the successful company it is today. These values include being an owner, taking bold bets, and choosing the best idea. Uber implements "being an owner" by having hiring managers who go above and beyond just to find the perfect fit employees for their company. Uber implements its second core value by ensuring that they take risks and try new things. Lastly, Uber encourages its employees to always vocalize and follow up on new ideas.

Stakeholders:

Driver: a person who has been hired by Uber to drive their car around a specific area to provide rides for people who request them.

Passenger: a person who needs to be picked up at a certain location and dropped off at a certain location

Manager: a person who works on the backend of the system and oversees the operation of business

User Stories:

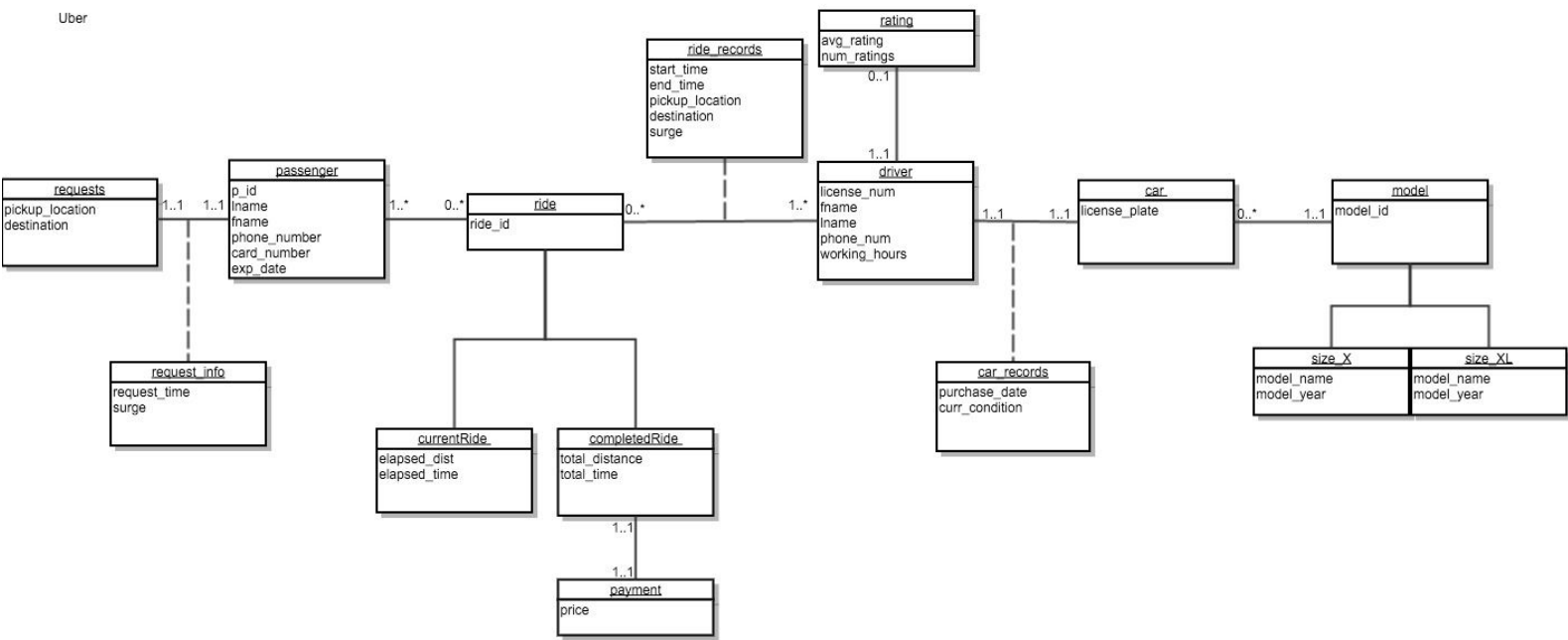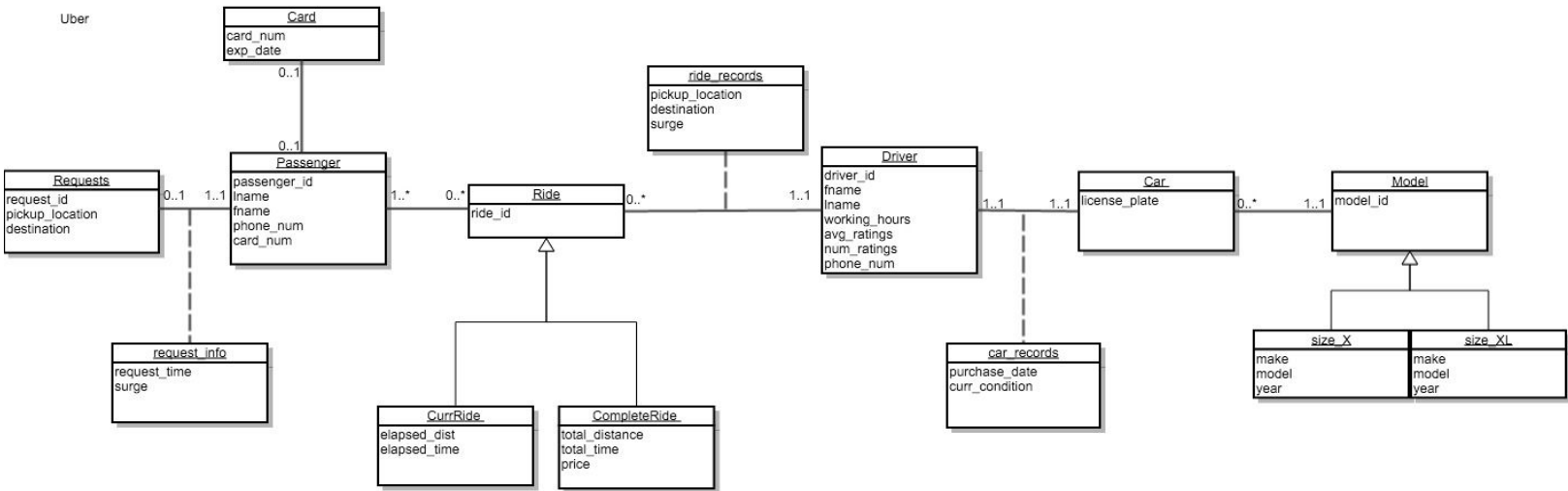| #	As a <role>	I want <goal>	So that <reasons>	Type Query
1	Driver	To know whether or not there will be a surge price to the requests	I am motivated to drive more when my pay is higher	simple
2	Manager	be able to pull out a driver's working hours	I can determine how long he has worked for the month	simple
3	Passenger	To be able to make a request for a ride	I can try to get a ride somewhere	simple
4	Passenger	To be able to view previous destinations	saves trouble and time of looking it up elsewhere	simple
5	Manager	To view condition of driver's car	to ensure quality customer service	simple
6	Passenger	To be able to rate the driver	I can validate the driver's service quality	complex #1
7	Driver	To be able to view all possible pick up locations	to decide which request is the most convenient for me	simple
8	Driver	To get access to passengers' phone number	I can contact passenger and notify upon arrival	simple
9	Driver	Accept a requested ride	I can begin a ride	complex #2
10	Driver	Complete a ride & view price for the ride	I can get paid and move on and take the next request	complex #3

Diagram of Initial Conceptual Model:

Uber

rating
avg_rating
num_ratings

ride_records
start_time
end_time
pickup_location
destination
surge

requests
pickup_location
destination

passenger
p_id
lname
fname
phone_number
card_number
exp_date

ride
ride_id

driver
license_num
fname
lname
phone_num
working_hours

car
license_plate

model
model_id

1..1 1..1   1..* 0..*   0..*   1..*   1..1   1..1   0..*   1..1   0..1   1..1

request_info
request_time
surge

currentRide
elapsed_dist
elapsed_time

completedRide
total_distance
total_time

car_records
purchase_date
curr_condition

size_X
model_name
model_year

size_XL
model_name
model_year

1..1
1..1

payment
price

Diagram of Final Conceptual Model:

Uber

Card
card_num
exp_date

0..1
0..1

ride_records
pickup_location
destination
surge

Requests
request_id
pickup_location
destination

Passenger
passenger_id
lname
fname
phone_num
card_num

Ride
ride_id

Driver
driver_id
fname
lname
working_hours
avg_ratings
num_ratings
phone_num

Car
license_plate

Model
model_id

0..1 1..1   1..* 0..*   0..*   1..1   1..1   1..1   0..*   1..1

request_info
request_time
surge

CurrRide
elapsed_dist
elapsed_time

CompleteRide
total_distance
total_time
price

car_records
purchase_date
curr_condition

size_X
make
model
year

size_XL
make
model
year

Relational Schema

KEY:
*Primary Key (Underlined)
Foreign Key (Bolded)

requests{pickup_location, destination, request_time, surge, p_id}
passenger{p_id, lname, fname, phone_number, card_number, exp_date}
currRide{ride_id, elapsed_dist, elapsed_time, start_time, end_time, pickup_location, destination, surge, **license_num, p_id**}
completeRide{ride_id, total_distance, total_time, start_time, end_time, pickup_location, destination, surge, **license_num, p_id**}
payment{price, ride_id}
driver{license_num, fname, lname, phone_num, working_hours}
rating{avg_rating, num_ratings, **license_num**}
car{purchase_date, curr_condition, license_plate, **model_id**}
size_X{model_id, model_name, model_year}
size_XL{model_id, model_name, model_year}

request_info is absorbed into requests
ride_records is absorbed into currRide and completeRide
car_records is absorbed into car

Functional Dependencies:

Requests: 2NF
p_id → Pickup_location
p_id → destination
p_id → request_time
request_time → surge **BAD DEPENDENCY**

Passenger: 2NF
p_id → lname
p_id → fname
p_id → phone_number
p_id → card_number
card_number → exp_date **BAD DEPENDENCY**

CurrRide: BCNF
ride_id → pickup_location
ride_id → destination
ride_id → surge
ride_id → p_id
ride_id → license_num

ride_id → elapsed_distance
ride_id → elapsed_time
ride_id → start_time
ride_id → end_time

CompleteRide: BCNF
ride_id → pickup_location
ride_id → destination
ride_id → surge
ride_id → total_distance
ride_id → total_time
ride_id → start_time
ride_id → end_time
ride_id → price
ride_id → p_id
ride_id → license_num

Driver: BCNF
license_num → fname
license_num → lname
license_num → phone_num
license_num → working_hours

Rating: BCNF
license_num → avg_rating
license_num → num_ratings

Car: BCNF
license_plate → purchase_date
license_plate → curr_condition
license_plate → model_id

size_X: BCNF
model_id → model_name
model_id → model_year

size_XL: BCNF
model_id → model_name
model_id → model_year

Normalized Schema:

<u>*Primary Key (Underlined)</u>
*Foreign Key (Bolded)

Passenger{<u>passenger_id</u>, fname, lname, phone_num, **card_num**}
Card{<u>card_num</u>, exp_date}
Requests{<u>request_id</u>, pickup_location, destination,request_time, surge,**passenger_id**}
CurrRide{<u>ride_id</u>, pickup_location, destination, surge, elapsed_distance, elapsed_time, **passenger_id, driver_id**}
CompleteRide{<u>ride_id</u>, pickup_location, destination, surge, total_distance, total_time, **passenger_id, driver_id,** price}
Driver{<u>driver_id</u>, fname, lname, phone_num, working_hours, avg_rating, num_ratings, **license_plate**}
Car{<u>license_plate</u>, purchase_date, curr_condition, **model_id**}
size_X{<u>model_id</u>, make, model, year}
size_XL{<u>model_id</u>, make, model, year}