IS6420 DATABASE THEORY AND DESIGN

UBER COMPANY DATABASE

Group 13

Aidan Tarufelli, Andy Charles, Bobbie Mason, Xuan Qin

Uber

Additional Submission Items:

Presentation

https://youtu.be/O9BuPbclAFQ

Slide Deck

https://drive.google.com/file/d/1F18Mdq_VHqXsASy_TLgqgCFa9UsRI4rr/view?usp=sharing

Uber

Contents

Additional Submission Items: 1
- Presentation 1
- Slide Deck 1

Contents 2

Executive Summary 4

An Uber Backstory 5
- The Meteoric Rise of Rideshares 5

Vision and Objectives 5

Products/Services 6

How does Uber use Transactional Databases? 6

Prioritized Requirements 7

Conceptual Model 8
- Conceptual Model 8
- Conceptual Model Description 9

Logical Model 10
- Logical Model 10
- Logical Model Description 10

Physical Model 11
- Physical Model 11
- Physical Model Description 11
- Description of Sample Data Used 12

Current and Future Features 12
- Currently Functional 12
- Could Be Supported, Yet to Be Developed 13
- Future Features 13
  - Rating System 13
  - Drivers as Passengers 13
  - ETA for Pickup 13
  - ETA for Arrival 14
  - Map View of Driver en Route 14
  - Map View While Riding 14
  - Multiple Stored Payment Methods 14

Apple Pay, Google Pay, Samsung Pay, Venmo, PayPal Etc 14
Scheduled Rides 14
Fare splitting 15

Conclusion 16

SQL Statements 17
Create Table Driver 17
Create Table Passenger 17
Create Table Ride 18

Insert Statements 18
Insert Into Driver 18
Insert Into Passenger 19
Insert Into Ride 20

References 22

Appendix 23
Course Material Used 23
Ideal Requirements Updated 24
Group Contribution Table 25

Uber

Executive Summary

Transportation has long been a massive industry for consumers. Specifically the concept of a taxi service has been around since the late 1800's. Consumers will continue to demand goods and services with more effortless delivery and on their time table.

With the advent of the smartphone, new industries have exploded in the last decade that exist solely within an app that runs on a device that fits in a pocket. As with any industry that has existed for the last century, so many changes have come about in ways that we may have never thought possible. Taxi services were in no way immune to these new forces in the marketplace that have been drivers for innovation and change. With services continually becoming more and more subject to fitting into the space of the culture of convenience, it is no surprise that services such as Uber, Lyft, Zimride, and so many other ridesharing services have come about.

For the purposes of this project, we are going to limit ourselves on the types of functionality that we are implementing. Our aim is to successfully implement the data stores that are associated with driver's and riders interacting with the app in relation to taking a trip using the services of Uber.

Our goals that we have listed as critical to our project involve us creating 6 tables that allow us to store all of the data that we are looking to capture. These include a table for riders, drivers, trips that have been taken, passenger ratings, driver ratings, and ride requests. Our team was able to use these tables to create the transactional database system such that we were able to create our tables, populate them with data, and verify that they would suit the needs of Uber.

Overall, this project was a great success for our team and showcases what we are capable of when we collaborate on complex issues. Should Uber need our assistance in the future, we would gladly accept as it has been a pleasure to work with all teams and divisions of the business.

Uber

An Uber Backstory

The Meteoric Rise of Rideshares

Uber is an exceptionally popular ridesharing app that first hit the market in 2010 in the City of San Francisco in the form of a smartphone app. The co-founders Travis Kalanick and Garrett Camp were struck with the idea in late 2008 when they were unable to get a ride when they needed one on a late night excursion. As they stood waiting for a ride on a snowy night in December, they saw a need in the market that they could fill. Their concept was to create a service that would allow you to request a ride at the tap of a button and have it come straight to you.

Just a decade later, Uber is one of the largest transportation companies offering services that range from their original ridesharing idea, food delivery, freight delivery, and corporate accounts. Their rise to the top is nothing short of incredible, but shows how a good idea and innovative thinking can take you on a journey all the way to the top.

U B E R

"Uber is evolving the way the world moves. By seamlessly connecting riders to drivers through our apps, we make cities more accessible, opening up more possibilities for riders and more business for drivers."

Vision and Objectives

Uber's vision is to provide a brand new, innovative way of transporting. It's all about changing the way people are traveling, and connecting to each other. It strives to provide the best network and platform to provide the most flexible, reliable and convenient way to connect the drivers and passengers. It gives literally everyone an

opportunity to become a Uber driver and make additional income based on their own schedule, and it provides the most economical, personalized taxi drop off service to passages. This also brings down the cost and makes cab services widely available everywhere.

Products/Services

Uber's product and services are largely online, cloud-computing based. It runs on a complex algorithm and provides the connection between a driver and a passenger. It also integrates the GPS technology to instantly match the driver and the requesting passenger. It has the algorithm that can find and pair the most adjacent driver with the closest distance so the passenger can get a ride as soon as possible. Uber's software also allows the destination to be typed in the request so it's a door-to-door service. Uber's product also provides transparency about fare cost, it will show you right at the spot to tell the passenger about the mileage and fare for each trip. So there is no need to haggle about the price and any discrepancy. Uber also makes the passenger provide their credit card information when they sign up at the Uber App, so the trip will be automatically paid when it's done and the driver will be paid accordingly. So overall, Uber provides a platform that allows the pairing of drivers and passengers, a system to manage the trip, and the system to process the payments seamlessly.

How does Uber use Transactional Databases?

As with any modern organization, transactional databases are at the very heart of how Uber does business. From managing customer and driver information, to making sure that they are able to keep track of what kinds of people are using their services with things like ratings. The most prominent feature of Uber that most people know is their ridesharing. Users request rides on their mobile devices, drivers accept or reject the request, and the process begins.

Uber offers so many other services apart from its original such as food delivery, freight, medical transport, election day transport and more. For the purposes of this project, we are focusing on the original product that is the Uber rideshare.

Uber

Prioritized Requirements

For the scope of our project we limited our prioritized requirements to just the completed Uber trips operated under Uber.com. We determined three requirements to go live which are the following:

Category	Description	Priority
Service	Passengers must be able to initiate a trip request	1
Service	Drivers must be able to receive and authorize the request	1
Service	Passengers must be able to see driver information (name, vehicle model, rating, etc)	1
Service	Passengers must be able to cancel the ride request if they desire to do so	1
Service	Drivers should be able to pick up multiple riders at once	2
Service	Each passenger must have a driver on a Ride	1
Service	Each driver must have a passenger on a Ride	1
Payments	Passengers should be able to use paypal as payment	2
Payments	Passengers should be able to have multiple payment methods saved	3

*Note: please see the appendix for an update on completion of the ideal requirements.

In order to meet those requirements, we went through and started analyzing the organizational requirements and we believe we were able to narrow down our priorities to six specific tables which are as follows:

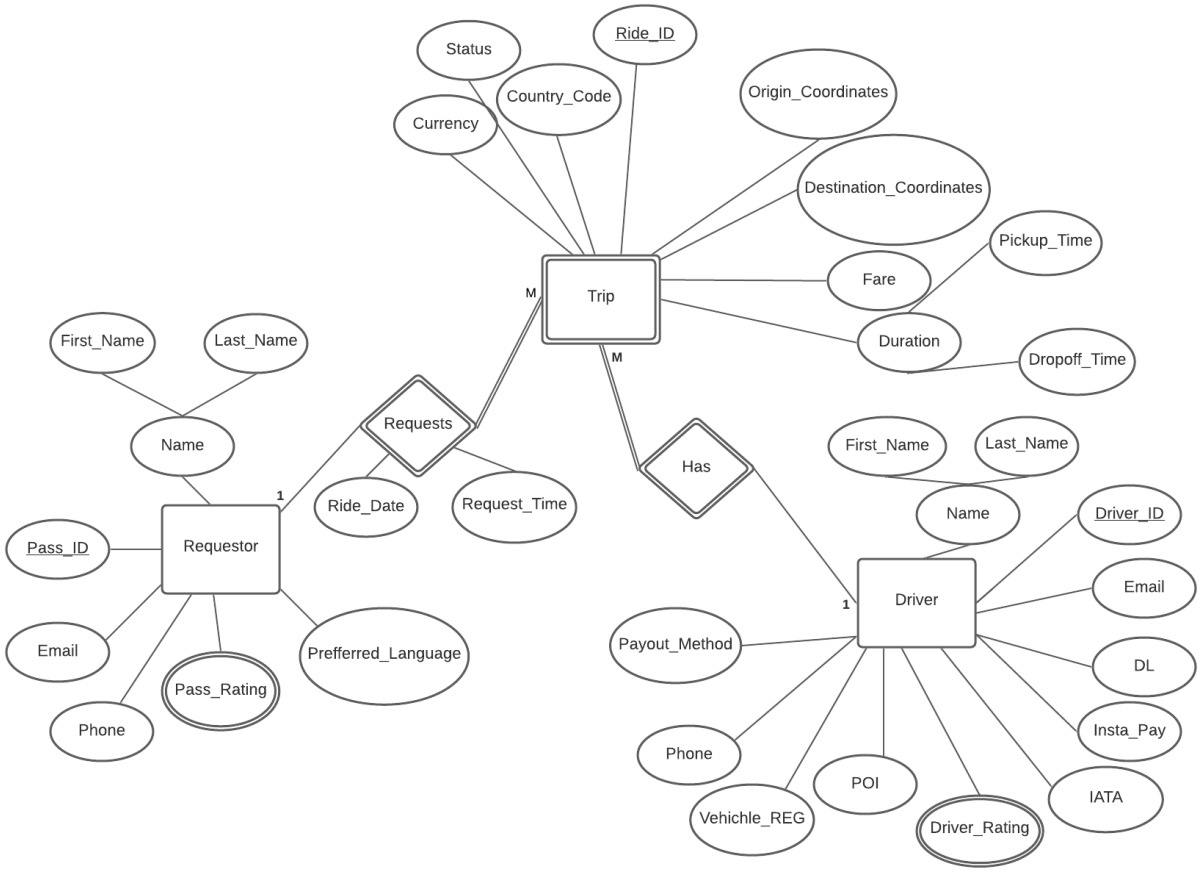
- Passenger
- Driver
- Passenger_Rating
- Driver_Rating

- Ride
- Ride_Request

These tables will help us to meet all of these requirements with ease as they are able to capture and store all of the data that we need in reference to how Uber operates on a day to day basis.

Conceptual Model

Conceptual Model



Conceptual Model Description

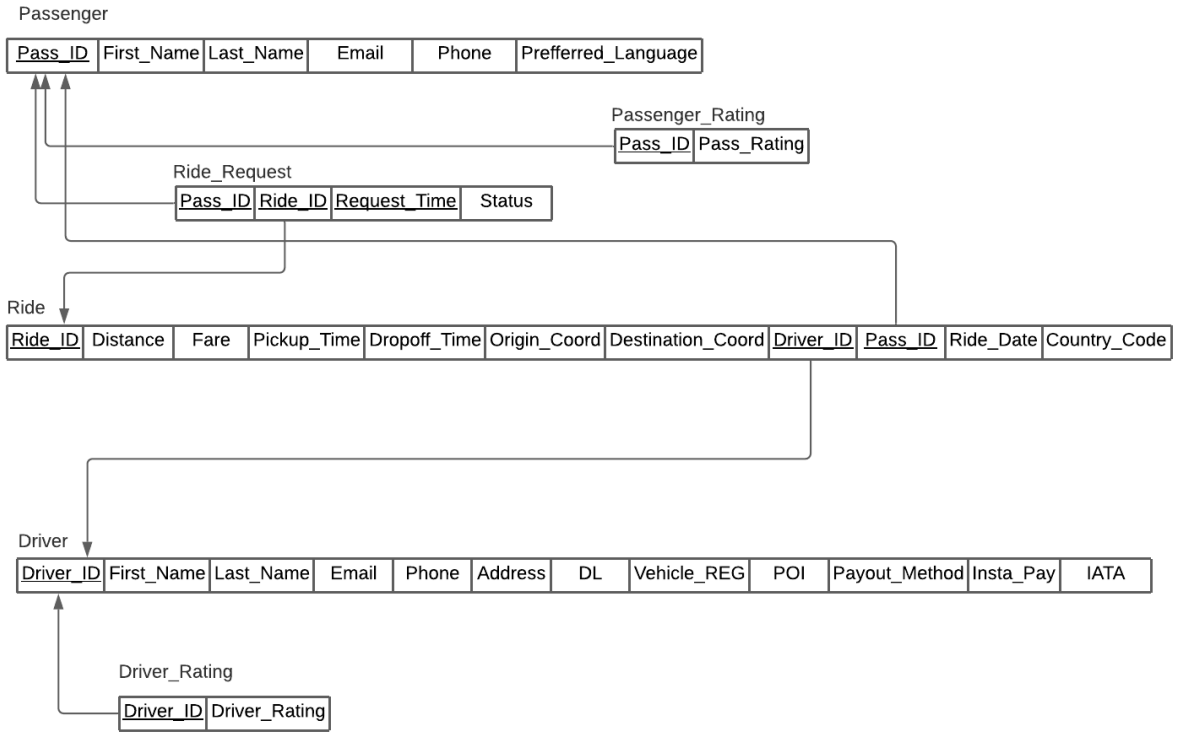
The diagram that is shown above is our conceptual model that we have created to represent the portion of the Uber transactional database that we are building. As with all things Uber, the act of transportation, in this case, the “Ride” is the linchpin holding this entire model together.
As we were building out this model, we found that the most interesting concept is that only partial participation exists. There is no requirement for participation for either drivers or passengers to exist in the system. However, considering that the service of Uber is to transport people from point A to point B, it is inevitable that rides will happen and we will have data that must be handled. Technically, there could be users or drivers who sign up for the app and input all of their information and then choose to do nothing with it.

Another interesting bit of information that was gleaned as we were doing our research to create this model is that every driver is associated with their nearest airport. This is not intuitive right away as many of us utilize Uber within our own city more frequently than anywhere else. But considering that Uber is the new taxi service, it only makes sense that a good bulk of fares for some drivers come from airports that are nearest them as travellers want an easy and convenient way to get to their destination.

We initially considered building out the rating system that exists with Uber as we were coming up with the concept of this model, but we found that it would be best to focus on the core features of the service. We agreed that building out the aggregating logic would be quite the task as some users of the service, drivers in particular would likely have thousands or tens of thousands of ratings at some point that need to be handled in such a way that they are more easily store and iterated through when calculating average rating or for the purposes of investigation into why a passengers experience was the way it was.

Logical Model

Logical Model



Logical Model Description

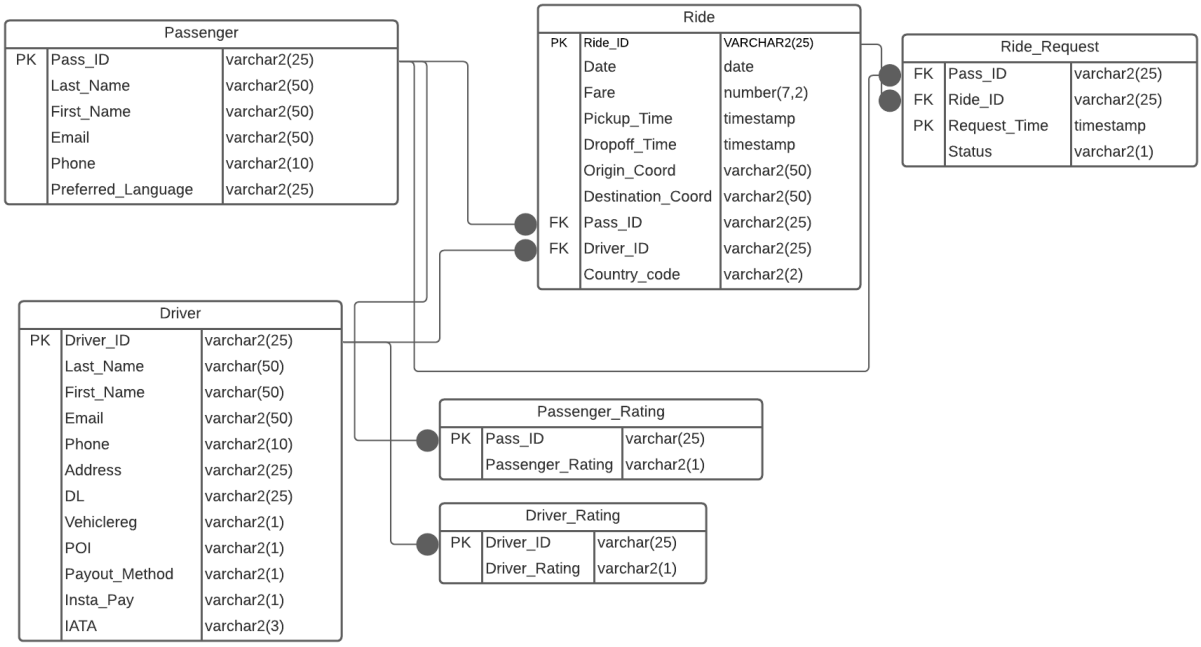
The diagram that is shown above is our logical model that was based on our conceptual model previously described. As you can see, we have handled the rating as it is a multivalued attribute on the backend even though another user would only ever see an aggregated result. All of the rating data must be stored so that an average rating can be calculated. The Ride table is interesting in that it is required for it to have a foreign key from Driver and from passenger in order for an entry to exist. These foreign keys help us to tie the data stored within to the users that are related to the occurrence.

The table that stores the most information is the driver table. This table encompasses all of the data that is required from a drive for Uber to allow them to work as an employee. This information is all critical to the safety and security of the organization. Without it, Uber could be held liable should one of their employees not meet these standards.

Uber

Physical Model

Physical Model



Physical Model Description

As we were building out the physical model, we relied heavily on the logical model that we described in the previous section. As you can see, we built out a module for each relation that we created in the logical model. This was the easy part of the physical model. What was more difficult was determining the data type and size for each attribute that we had decided to include. The app as it exists now is very well built and easy to navigate especially when you are setting up your account. We had to be certain that we were allowing enough space for names and other important information to be accurately captured.

In addition to learning how the app actually works, we tried to figure out if there would be any edge cases that we should be aware of. One of the big questions is how do we govern where users select as their pickup or dropoff locations? Uber is not necessarily available as a service in all areas so this would cause a breakdown in their business model if it was not properly handled. In the case of building out our version of this system, we made the assumption that users and drivers that are users of Uber understand the limitations of the app and are only making and fulfilling requests that operate within the confines of what is feasible for the service. Obviously there would

need to be some sort of indicator associated with regions that users are requesting services for that would indicate that Uber is not available. This was another feature that we were not quite sure how it would be implemented, but thought it would be cool to explore as a thought experiment. We did ensure, though, that we handled the instance of when a user or a driver decides to cancel or decline the request that has been sent through. We all make mistakes when we are trying to use a service that is new to us, so we felt it important to include.

Description of Sample Data Used

For this project, we generated our sample data ourselves utilizing free third-party service that quickly and effectively generates data that was suitable for the purposes of testing. There were a few kinks that we had to work out as we were generating this data as certain aspects of what we would like to capture are very unique. We have included some screenshots of SQL developer showcasing what our data looks like when inserted into a table and is then queried.

For the purposes of the service, we would have to implement restrictions on what types of information could be inserted into our tables as we would not want to leave ourselves open to any type of attack. More importantly, we would want to ensure that our users get the service that they expect from Uber. Good data can make for an exceptional product or experience and Uber is not an exception.

Current and Future Features

Currently Functional

At the close of this project, we feel that we have implemented our most priority objectives that we set out to accomplish. We wanted users to be able to input all of their required data regardless of if they were a passenger or a driver. We also wanted users to be able to request rides and have them be fulfilled by a trusted driver. By implementing the structure of this data as we have, passengers and drivers will be able to look back at their history. Granted, this will take some further effort on our part, but it is on the roadmap of what is possible in the near future should we build out this implementation further.

Could Be Supported, Yet to Be Developed

As mentioned in the previous section, the current database design could handle the looking up of historic rides for both passengers and drivers. We did not have time to implement such a feature as it was thought of much later in the process. We felt it was worth noting as all of this data is going to be held onto by Uber and they might as well make the most of it.

Future Features

As we were exploring the Uber app, we came across many features that they have already implemented that we believe would be beneficial. Obviously, Uber has been around for some time and built up these features over that period. Below are the features that we discovered:

- Rating System
- Drivers as Passengers
- ETA for pickup
- ETA for arrival
- Map view of driver en route
- Map view while riding
- Multiple stored payment methods
- Apple Pay, Google Pay, Samsung Pay, Venmo, PayPal Etc
- Scheduled rides
- Fare splitting

Rating System

Uber utilizes a rating system that allows passengers and drivers to rate one another. This has become a key metric for Uber to know how people are using their service and if they are being respectful. This would allow Uber to build a case against someone if needs be, or reward individuals who are standout

Drivers as Passengers

This feature is very self explanatory. Drivers should be able to utilize the services of Uber when they are not driving themselves. It would be nonsense to make them create a separate account for this purpose.

ETA for Pickup

As with everything these days, people are concerned about time. Time is the one thing that we can’t get back if it’s lost and people want to know if theirs is worth the investment. This feature is standard and is helpful for people to plan out their travels.

ETA for Arrival

As mentioned previously, time is ever important. Users should be able to see just how long it is going to take for them to get to their destination. Obviously those estimates could be off, but it is better than nothing as it can give users and those they are communicating with about their arrival more insights.

Map View of Driver en Route

This is another very standard feature that Uber has implemented to give users more insights as to where their driver is. It can also let users know just what is going on in the event that there is something happening that they need to be aware of. Let's be honest, we've all had a driver that takes the most interesting route to get to you and it's fun to watch.

Map View While Riding

This feature is another standard for Uber as users want to be able to know how they are getting from point A to point B. it also allows users to share their exact location with someone if they choose.

Multiple Stored Payment Methods

Most users these days have multiple forms of card payment that they may want to store on their account. Entering in their information repeatedly would be rather annoying and likely detract people from utilizing the service as much as Uber would like. This feature already exists and likely won't go anywhere anytime soon.

Apple Pay, Google Pay, Samsung Pay, Venmo, PayPal Etc

Soft wallets are becoming ever more common in the modern day of contactless payments. Users want these features as options because they don't want to worry about inputting their information on every app or service that they choose to use. Uber has already implemented this feature and it is quite a popular form of payment.

Scheduled Rides

Users having the ability to schedule their rides ahead of time takes away the stress of worrying about waiting for a vehicle to arrive. This means that they can more accurately plan out their day or their travels. Many people and companies like this aspect of Uber as they can know when arrivals and departures happen at airports and can set everything up in advance.

Fare splitting

The implementation of this feature was one that was highly anticipated by everyone that likes to go out with their friends. It can be exceptionally annoying to try to coordinate repayment to one person after a ride is over. Thus, fare splitting came about to make it easier for groups of friends to share the cost of their ride much more easily.

Conclusion

Uber is a very young company that has had explosive growth in just the few short years that it has been around. It was fascinating for us to analyze their systems as a group and really try to figure out how they would implement certain features. It also gave us the chance to explore some of the more obscure features that Uber has implemented over time. A company such as Uber has to be continually innovating and making sure that they are in tune with what their consumers want. Now more than ever, companies are having to focus on the consumer. In the day and age of reviews, customer satisfaction and delight are top of mind for any company and Uber is a great example of how to keep innovating to keep consumers satisfied. After completing this project as a group, we believe that Uber truly has longevity potential and will continue to be the first choice of people that are just looking for a ride, food, freight delivery, medical transportation, and so much more.

Uber

SQL Statements

Create Table Driver

```
drop table Driver;
create table Driver
(Driver_ID varchar2(25) not null
,First_Name  varchar2(50)
,Last_Name  varchar2(50)
,email varchar2(50) not null
,Phone          varchar2(10) not null
,Address        varchar2(25)
,DL     varchar2(25)
,VehicleReg  varchar2(1)
,POI    varchar2(1)
,Payout_Method      varchar2(1)
,Insta_Pay    varchar2(1)
,IATA  varchar2(3)
,Driver_Rating          varchar2(1)
,CONSTRAINT driver_pk PRIMARY KEY (driver_id))
;
```

Create Table Passenger

```
drop table Passenger;
create table Passenger
(Pass_ID       varchar2(25) not NULL
,first_name    varchar2(50)
,last_name    varchar2(50)
,Email varchar2(50)
,Phone          varchar2(10)
,Preferred_Language         varchar2(25)
,Pass_Rating varchar2(1)
,CONSTRAINT passenger_pk PRIMARY KEY (pass_id))
;
```

Create Table Ride

```
Drop table Ride;
create table Ride
(Ride_ID       varchar2(25) not null
,Driver_ID varchar2(25) not null
,Pass_ID       varchar2(25) Not NULL
,Ride_Date    date not null
,Currency      varchar2(50)
,Fare Number (7,2)
,Pickup_Time          timestamp
,Dropoff_Time         timestamp
,Origin_Coord         varchar2(50)
,Destination_Coord  varchar2(50)
,Country_Code         varchar2(2)
,Request_Time         timestamp not null
,Status          varchar2(1)
,CONSTRAINT Ride_pk PRIMARY KEY (Ride_id))
;
```

Insert Statements

Insert Into Driver

```
Insert into Driver values
('D9110077','Eydie','Coarser','EydieCoarser@invalidemail.com',6158472745,'0104
Warrior Junction',1444348488,0,0,1,0,'ACC',1);
Insert into Driver values
('D6092297','Arlena','OLennane','ArlenaOLennane@invalidemail.com',5491261724,'977
5 Farragut Street',5722904489,0,0,1,0,'ADL',3);
Insert into Driver values
('D5019061','Ennis','Creasey','EnnisCreasey@invalidemail.com',6559123036,'639
Center Court',7531445919,1,1,1,1,'MAD',2);
Insert into Driver values
('D4695965','Bendite','McEwen','BenditeMcEwen@invalidemail.com',2434410030,'0492
0 Briar Crest Circle',2608252289,1,1,2,1,'CAK',3);
Insert into Driver values
('D1740277','Saree','Woltering','SareeWoltering@invalidemail.com',9077277322,'684
Merchant Junction',965981163,0,0,1,0,'DWC',2);
```

```
Insert into Driver values
('D7687988','Barron','Einchcombe','BarronEinchcombe@invalidemail.com',8432973303,
'08872 Texas Pass',1119469778,1,1,2,1,'HOF',3);
Insert into Driver values
('D8558783','Erek','Eyres','ErekEyres@invalidemail.com',2566278291,'3 Kim
Point',2992856200,1,1,1,1,'ALB',2);
Insert into Driver values
('D5354254','Nariko','Hulse','NarikoHulse@invalidemail.com',2544350742,'114 West
Terrace',3387282954,0,0,1,0,'OAJ',4);
Insert into Driver values
('D6830425','Wilek','MacKaig','WilekMacKaig@invalidemail.com',1162193050,'82233
Mockingbird Road',3701483192,1,1,2,1,'ABQ',2);
SELECT * From Driver;
```

Insert Into Passenger

```
Insert into Passenger values
('P5016076','Bernetta','Derisly','BernettaDerisly@invalidemail.com',1151156032,'English
',3);
Insert into Passenger values
('P4376647','Mariele','Mealham','MarieleMealham@invalidemail.com',5390044856,'Fren
ch',5);
Insert into Passenger values
('P1756933','Orella','Bassingden','OrellaBassingden@invalidemail.com',1898509803,'G
erman',4);
Insert into Passenger values
('P8389673','Laurie','Branney','LaurieBranney@invalidemail.com',5684353699,'Spanish'
,2);
Insert into Passenger values
('P5689818','Bink','Jopp','BinkJopp@invalidemail.com',1672730381,'English',4);
Insert into Passenger values
('P4928545','Xymenes','Wenderoth','XymenesWenderoth@invalidemail.com',58750523
96,'French',2);
Insert into Passenger values
('P8031872','Leila','Stayt','LeilaStayt@invalidemail.com',8816281679,'German',3);
Insert into Passenger values
('P1209839','Maurie','Stubbes','MaurieStubbes@invalidemail.com',5051495209
        ,'Spanish',4);
Insert into Passenger values
('P1213404','Meade','Chaldecott','MeadeChaldecott@invalidemail.com',4753172421,'En
glish',3);
```

```
Insert into Passenger values
('P7366594','Horatio','Denford','HoratioDenford@invalidemail.com',6983850233,'French'
,1);
Insert into Passenger values
('P8967493','Kendra','Lemmon','KendraLemmon@invalidemail.com',9200943180,'Germ
an',5);
Insert into Passenger values
('P9090012','Benoite','Drillingcourt','BenoiteDrillingcourt@invalidemail.com',4450054443
,'Spanish',5);
SELECT * From Passenger;
```

Insert Into Ride

```
Insert into Ride values ('TW585619550','D9959367','P5016076',To_Date('2019-01-
01','YYYY-MM-DD'),'New Taiwan Dollar',5.56,TO_TIMESTAMP('2019-0101
03:58:45','YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2019-01-01 04:04:12','YYYY-
MM-DD HH24:MI:SS'),'137 Onsgard Center','1178 Mayfield
Parkway','TW',TO_TIMESTAMP('2019-01-01 04:15:12','YYYY-MM-DD
HH24:MI:SS'),1);
Insert into Ride values ('US349689962','D9110077','P4376647',To_Date('2019-04-
22','YYYY-MM-DD'),'US Dollar',13.85,'','6 Crowley Court','72 Warbler
Alley','US',TO_TIMESTAMP('2019-04-22 05:17:59','YYYY-MM-DD HH24:MI:SS'),0);
Insert into Ride values ('PK765173277','D6092297','P1756933',To_Date('2019-10-
16','YYYY-MM-DD'),'Pakistan Rupee',16.81,TO_TIMESTAMP('2019-10-16
19:21:13','YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2019-10-16 19:38:13','YYYY-
MM-DD HH24:MI:SS'),'71359 Hanover Place','42 Green Ridge
Center','PK',TO_TIMESTAMP('2019-10-16 19:15:46','YYYY-MM-DD HH24:MI:SS'),1);
Insert into Ride values ('DE568868871','D5019061','P8389673',To_Date('2019-12-
17','YYYY-MM-DD'),'Euro',3.76,TO_TIMESTAMP('2019-12-17 21:44:19','YYYY-MM-DD
HH24:MI:SS'),TO_TIMESTAMP('2019-12-17 21:57:19','YYYY-MM-DD HH24:MI:SS'),'3
Ronald Regan Hill','52546 Hauk Way','DE',TO_TIMESTAMP('2019-12-17
21:38:52','YYYY-MM-DD HH24:MI:SS'),1);
Insert into Ride values ('ZA803060985','D4695965','P5689818',To_Date('2020-08-
20','YYYY-MM-DD'),'Rand',16.03,TO_TIMESTAMP('2020-08-20 22:39:29','YYYY-MM-
DD HH24:MI:SS'),TO_TIMESTAMP('2020-08-20 22:50:29','YYYY-MM-DD
HH24:MI:SS'),'1 Fordem Center','19706 Tennessee
Terrace','ZA',TO_TIMESTAMP('2020-08-20 22:34:02','YYYY-MM-DD HH24:MI:SS'),1);
Insert into Ride values ('US302888235','D1740277','P4928545',To_Date('2019-04-
13','YYYY-MM-DD'),'US Dollar',14.52,TO_TIMESTAMP('2019-04-13 03:05:27','YYYY-
MM-DD HH24:MI:SS'),TO_TIMESTAMP('2019-04-13 03:19:27','YYYY-MM-DD
```

```
HH24:MI:SS'),'4 Nobel Lane','05 Kensington Street','US',TO_TIMESTAMP('2019-04-13
03:00:00','YYYY-MM-DD HH24:MI:SS'),1);
Insert into Ride values ('GB242895108','D7687988','P8031872',To_Date('2020-01-
03','YYYY-MM-DD'),'Pound Sterling',15.55,TO_TIMESTAMP('2020-01-03
05:15:53','YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2020-01-03 05:34:53','YYYY-
MM-DD HH24:MI:SS'),'9438 Green Center','201 Eggendart
Pass','GB',TO_TIMESTAMP('2020-01-03 05:10:26','YYYY-MM-DD HH24:MI:SS'),1);
Insert into Ride values ('US253180881','D8558783','P1209839',To_Date('2020-02-
20','YYYY-MM-DD'),'US Dollar',18.08,TO_TIMESTAMP('2020-02-20 05:50:13','YYYY-
MM-DD HH24:MI:SS'),TO_TIMESTAMP('2020-02-20 06:01:13','YYYY-MM-DD
HH24:MI:SS'),'130 Hollow Ridge Way','05709 Buena Vista
Crossing','US',TO_TIMESTAMP('2020-02-02 05:44:46','YYYY-MM-DD HH24:MI:SS'),1);
Insert into Ride values ('US111285367','D5354254','P1213404',To_Date('2020-07-
12','YYYY-MM-DD'),'US Dollar',11.52,'','65640 Ridgeway Parkway','4 Stone Corner
Drive','US',TO_TIMESTAMP('2020-07-12 16:43:16','YYYY-MM-DD HH24:MI:SS'),0);
Insert into Ride values ('US478915186','D6830425','P7366594',To_Date('2019-05-
23','YYYY-MM-DD'),'US Dollar',16.71,TO_TIMESTAMP('2019-05-23 19:00:01','YYYY-
MM-DD HH24:MI:SS'),TO_TIMESTAMP('2019-05-23 19:19:01','YYYY-MM-DD
HH24:MI:SS'),'9743 Vahlen Crossing','31926 Main
Terrace','US',TO_TIMESTAMP('2019-05-23 18:54:34','YYYY-MM-DD HH24:MI:SS'),1);
SELECT * From Ride;
```

References

All external information regarding Uber that was used in our report comes from https://www.uber.com/

Appendix

Course Material Used

User Requirements
For this section we utilized the first assignment we completed

- Learning what user requirements are
- Identifying user requirements to implement

Conceptual Model
For this section we used the first assignment we completed

- Determined how to create conceptual model
- Decide what types of relationships to have

Logical Model
For this section, we used the second assignment

- Turning our conceptual model into a logical model
- Deciding what our primary and foreign keys would be

Physical Model
For this section, we utilized the second lab

- Creating a physical model based on logical models
- Determining what data types to use based on what we know

SQL Statements
For this section, we utilized the first lab, the third lab, and the third assignment

- Setting up the database
- Populating our tables with data and updating it
- Querying our data as needed

Uber

Ideal Requirements Updated

Category	Description	Status
Service	Passengers must be able to initiate a trip request	Completed
Service	Drivers must be able to receive and authorize the request	Completed
Service	Passengers must be able to see driver information (name, vehicle model, rating, etc)	Completed
Service	Passengers must be able to cancel the ride request if they desire to do so	Completed
Service	Drivers should be able to pick up multiple riders at once	Future
Service	Each passenger must have a driver on a Ride	Completed
Service	Each driver must have a passenger on a Ride	Completed
Payments	Passengers should be able to use paypal as payment	Future
Payments	Passengers should be able to have multiple payment methods saved	Future

Uber

Group Contribution Table

Date	Team Member	Hours Spent	Description of Work	Additional Comments
9/21/2020	Aidan Tarufelli	1.5	Requirements Gathering	Working to gather business requirements from Uber app to determine what we wanted to implement.
9/26/2020	Aidan Tarufelli	2	Conceptual Model	Put together a rough conceptual model of what we wanted to base our implementation on
10/2/2020	Aidan Tarufelli	3	Conceptual & Logical Model	Solidified the conceptual model and what we wanted that to look like. Started working on implementing the logical model
10/6/2020	Aidan Tarufelli	1.5	Conceptual & Logical Model	Revised a few typos on conceptual model and finished up the logical model
10/12/2020	Aidan Tarufelli	2	Physical Model	Built out the first draft of the physical model
10/15/2020	Aidan Tarufelli	2	Report	Finalized all models and started working on the skeleton of our final report
10/15/2020	Xuan Qin	2	Write Report	Gathering information and doing research to write the vision/mission portion of report
10/15/2020	Full Group	1	Models	Finalized all models and agreed that we were good to continue to development
10/20/2020	Aidan Tarufelli	0.5	Data Generation	Created sample data for our tables that were to be implemented.

10/21/2020	Aidan Tarufelli	2	Final Report	Wrote the executive summary for our project and began work on description of models and other report info
10/24/2020	Aidan Tarufelli	0.5	Final Report	Worked on formatting and ensuring that we had all sections listed out to ensure that we covered all info
10/24/2020	Bobbie Mason	0.5	DatabaseDevelopment	Write queries to build tables
10/25/2020	Bobbie Mason	0.5	DatabaseDevelopment	Write queries to insert data into tables
10/26/2020	Bobbie Mason	2	DatabaseDevelopment	Work on data insertion
10/26/2020	Andy Charles	1.5	Presentation	Found presentation theme and created a structured format for the group to add in information. Added in initial information for introduction and company history.
10/27/2020	Bobbie Mason	1.5	DatabaseDevelopment	Manipulate Data for insertion into tables
10/28/2020	Aidan Tarufelli	3	Presentation	Worked to add in slides required for our presentation and make sure that information was cleaned up and presentable
10/28/2020	Bobbie Mason	2	DatabaseDevelopment	Manipulate Data for insertion into tables
10/30/2020	Bobbie Mason	0.5	DatabaseDevelopment	Manipulate Data for insertion into tables
10/30/2020	Aidan Tarufelli	2	Final Report	Adding in more information that is needed.
10/31/2020	Bobbie Mason	1.5	DatabaseDevelopment	Work on data insertion
10/31/2020	Aidan Tarufelli	0.5	Final Report	Added in SQL data and tables

11/1/2020	Aidan Tarufelli	2	Final Report	Finalizing the report and making sure that everything is formatted properly and presentable.