**DATA MANAGEMENT**

(MIS- 6326)

**PROJECT**

UBER APP

Professor. Vivek Arora

Group 4:

Rony Mathew Joseph

Sumana Narayana Reddy

Dipsa Prusty

Anthony Sung

**TABLE OF CONTENTS**

* + Project Description
  + Tables
  + ERD
  + SQL Queries
  + Query Scenarios

**PROJECT DESCRIPTION**

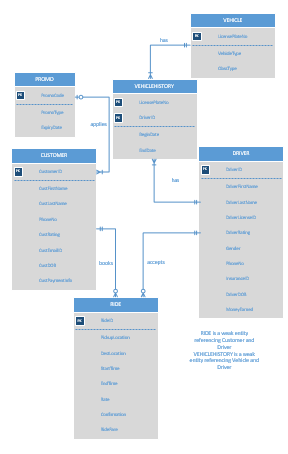
**Uber** is an online transportation network company whose primary product is the development and operation of its Uber software. Uber allows consumers to request car transportation via its mobile app. To support these functions, the database should contain information about customers, drivers, vehicles and the rides. The following list describes Uber’s primary database requirements in more detail.

* Uber drivers must be 21 years of age or older, have a current in-state Driver’s License, have in-state car insurance in the driver’s name, and a smart phone able to use the Uber App. Uber drivers’ full names are shown when customers request a ride.
* Uber drivers must have access to a vehicle that meet Uber’s specifications in at least one of its vehicle options (Black Car, Taxi, UberX, SUV, and LUX) with a registered license plate. Uber drivers may drive more than one vehicle.
* To request a ride, customers must have created an account through the mobile app and enter their first and last name, phone number, email address, and date of birth. After the customer has created an account, a unique customer ID will be created for every customer. Before requesting a ride, customers must also enter with a valid credit card and/or paypal account and select one as a default option.
* Promo Codes are available for a limited time and may be either a percentage off a ride or an amount off a ride. Customers may apply as many valid codes as they wish but one at a time,i.e, one code is valid for one ride.
* When customers need to book a ride, they will select a pick-up spot and destination. After accepting the rate and confirming the ride with a driver, the driver will pick up the customer and transport the customer(s) to the selected destination. The travel time and rate will be used to calculate the final fare, which will be charged to the customer’s account and the driver will be paid his or her amount. Drivers are also able to see how much money they have accumulated.
* Customers may select gender of driver. They may also select driverless cars as an option. Gender has three options of M, F, and R. In the future, may technologists believe driverless cars will be an option.
* Uber and other companies will likely offer promotions to get people to try driverless vehicles.

Since gender is an additional field to the driver entity, there were not significant changes to the ERD.

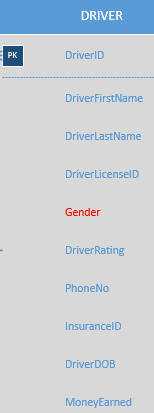
* Customers and Drivers rate one another after the ride is completed. Customers may book as many rides as needed.

**ERD DIAGRAM**



**ERD NEW FEATURE**- Choosing driver type

* We were inspired by Uber in India where female customers wanted to be able to drive safely with female drivers so we added Gender to the Driver Entity. And with this new feature of selecting type of driver, customer can choose the driver type which includes male, female driver and autonomous driverless vehicles. Having the option to choose is always a good thing for any private service. If you have different options, you can choose something that fits you better. This is a way to allow women passengers to choose women drivers without isolating them or making them the only ones to benefit.
* It will become cheaper for consumers to hail a driverless car than to own a personal vehicle. A car capable of handling most of the tasks of driving on its own. A self-driving car can carry someone from home to work, even take a package in the boot to be delivered to a client – all while a conventional car would be sitting in its owner’s car park. We believe that by providing people with more choices, we can help them make better decisions; and by using new technology we can improve safety for everyone on the road.



The entities are Customer, Driver, Vehicle, Promo and Ride. Ride is a weak entity referencing Customer and Driver.

* For the *applies* relationship, the customer will be able to apply one promo code (which may be a percentage or amount off a ride) at a time and many customers may have the same promo code. This implies a 1-M relationship between Promo and customer.
* For *drives* relationship between vehicle and driver, the driver may be associated with multiple vehicles and vice versa. Thus, there exists a M-N relationship between vehicle and driver.
* A *books* relation will exist when a customer wants a ride. A customer may or may not have booked a ride and a ride is associated with one customer at a time. This indicates a 1-M relation between customer and Ride.
* *Accepts* relation will exist when the driver accepts the request from the customer. A ride will be associated with one driver when he accepts the request from customer. Thus, there exists a 1-M relation between Driver and ride.

There exists a M-N relationship between Vehicle and Driver. We can add a history relationship between Vehicle and Driver as an independent change, which will keep a record of the drivers who have used that particular vehicle .

**TABLES**

The database consists of 6 tables. The tables are as follows:

1. **Customer details**: this table contains basic data related to the customers.

**Variables**: Cutomer\_id, first\_name, last\_name, email\_id, DOB, rating, payment\_info

1. **Driver details**: this table contains all relevant data about the drivers working for uber.

**Variables**: Driver\_id, first\_name, last\_name, driverlicense\_id, rating, phone\_no, insurance\_id, dob, money\_earned, gender, license\_plate\_no

1. **Promocode**: This table displays the promocode, expiry date, promo type.

**Variables**: Promocode, promotype, expiry\_date

1. **Ride**: This table shows all the information about the ride booked by the customer.

**Variables**: Ride\_id, pickup\_location, dest\_location, start\_time, end\_time, rate, distance, driver\_id, customer\_id, ride\_fare

1. **Vehicle**: This table contains details about the vehicle.

**Variables**: License\_plate\_no, vehicle\_type, class\_type

1. **Vehicle history**: it has the data regarding vehicle history.

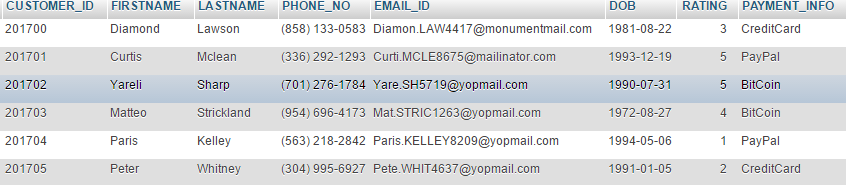
**Variables**: License\_plate\_no, driver\_id, begin\_date, end\_date

**SQL QUERIES**

* **Customer**- CREATE TABLE `uber`.`customer` ( `CUSTOMER ID` VARCHAR(10) NOT NULL , `FIRSTNAME` TEXT NOT NULL , `LASTNAME` TEXT NOT NULL , `PHONE NO` VARCHAR(15) NOT NULL , `EMAIL ID` VARCHAR(50) NOT NULL , `DOB` DATE NOT NULL , `RATING` FLOAT(3) NOT NULL , `PAYMENT INFO` VARCHAR(10) NOT NULL );

CONSTRAINT PKCustomerID PRIMARY KEY (‘CUSTOMER ID’))

CONSTRAINT FKPromoCode FOREIGN KEY(PROMOCODE) REFERENCES promocode)



* **Driver**- CREATE TABLE `uber`.`driver` ( `DRIVER\_ID` VARCHAR(10) NOT NULL , `FIRSTNAME` VARCHAR(15) NOT NULL , `LASTNAME` VARCHAR(15) NOT NULL , `DRIVERLICENSE\_ID` VARCHAR(20) NOT NULL , `RATING` INT(1) NOT NULL , `PHONE\_NO` VARCHAR(15) NOT NULL , `INSURANCE\_ID` VARCHAR(15) NOT NULL , `DOB` DATE NOT NULL , `MONEY\_EARNED` FLOAT(10) NOT NULL;

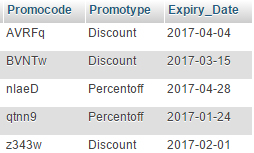
CONSTRAINT PKdriverID PRIMARY KEY (‘DRIVER\_ID’))

CONSTRAINT FKLP FOREIGN KEY(LICENSE\_PLATE\_NO) REFERENCES VEHICLE



* **Promocode**- CREATE TABLE `uber`.`promocode` ( `Promocode` VARCHAR(10) NOT NULL , `Promotype` VARCHAR(10) NOT NULL , `Expiry\_Date` DATE NOT NULL ;

CONSTRAINT PromoPK Primary Key (‘Promocode’) )



* **Ride**- CREATE TABLE `uber`.`RIDE` ( `RIDE\_ID` VARCHAR(15) NOT NULL , `PICKUP\_LOCATION` TEXT NOT NULL , `DEST\_LOCATION` TEXT NOT NULL , `START\_TIME` DATETIME NOT NULL , `END\_TIME` DATETIME NOT NULL , `RATE` FLOAT(5) NOT NULL , `RIDE\_FARE` DOUBLE(10) NOT NULL ); CONSTRAINT PKrideID PRIMARY KEY (‘RIDE\_ID’))

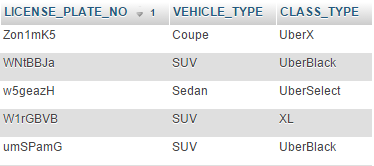
CONSTRAINT FKCustID FOREIGN KEY(CUSTOMER\_ID) REFERENCES customer;

****CONSTRAINT FKDriverID FOREIGN KEY(DRIVER\_ID) REFERENCES driver)

* **Vehicle**- CREATE TABLE `uber`.`Vehicle` ( `LICENSE\_PLATE\_NO` VARCHAR(10) NOT NULL ,

`VEHICLE\_TYPE` TEXT NOT NULL , `CLASS\_TYPE` TEXT NOT NULL;

CONSTRAINT PKLP PRIMARY KEY (‘LICENSE\_PLATE\_NO’))

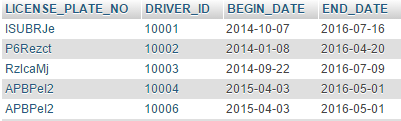


* **Vehicle\_history**-CREATE TABLE `uber`.`Vehicle\_history` ( `LICENSE\_PLATE\_NO` VARCHAR(10) NOT NULL , `DRIVER\_ID` VARCHAR(10) NOT NULL , `BEGIN\_DATE` DATE NOT NULL, `END\_DATE` DATE NOT NULL;

CONSTRAINT PKLP PRIMARY KEY (‘LICENSE\_PLATE\_NO’))

CONSTRAINT FKlicenseno FOREIGN KEY(LICENSE\_PLATE\_NO) REFERENCES vehicle

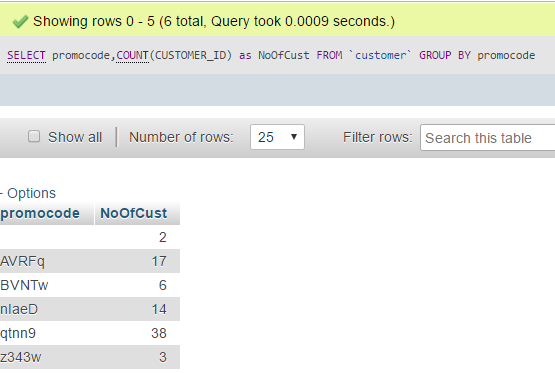
CONSTRAINT FKdriverID FOREIGN KEY(DRIVER\_ID) REFERENCES driver



**QUERY SCENARIOS**

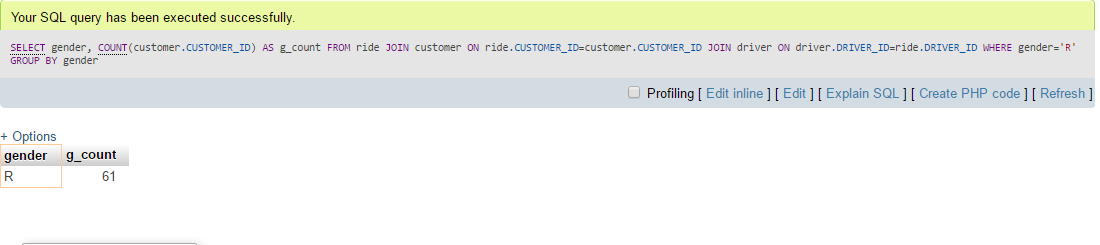
1. **Find how many customers used a particular promocode.**

SELECT promocode,COUNT(CUSTOMER\_ID) as NoOfCust FROM `customer` GROUP BY promocode



1. **Number of customers who prefer robot as drivers**

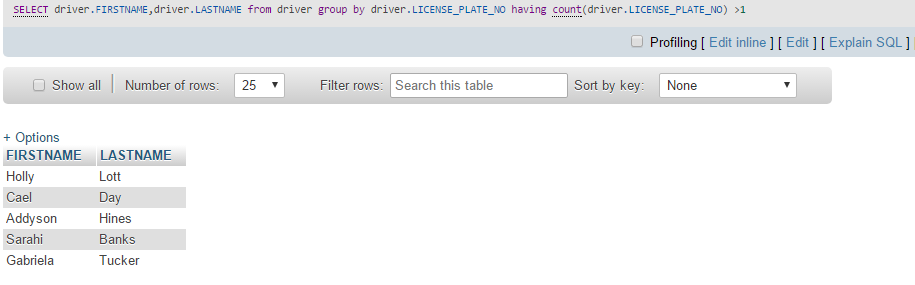
SELECT gender, COUNT(customer.CUSTOMER\_ID) AS g\_count FROM ride JOIN customer ON ride.CUSTOMER\_ID=customer.CUSTOMER\_ID JOIN driver ON driver.DRIVER\_ID=ride.DRIVER\_ID WHERE gender='R' GROUP BY gender



1. **Details of driver having more than one vehicle**

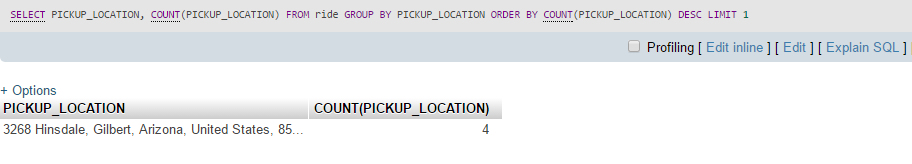
SELECT driver.FIRSTNAME,driver.LASTNAME from driver group by driver.LICENSE\_PLATE\_NO

having count(driver.LICENSE\_PLATE\_NO) >1



1. **Most popular pick-up location**

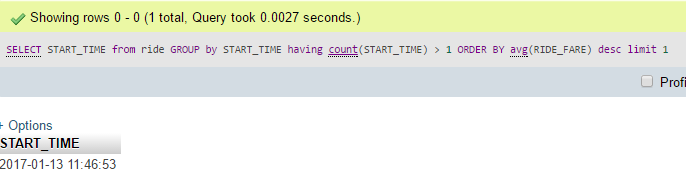
SELECT PICKUP\_LOCATION, COUNT(PICKUP\_LOCATION) FROM ride GROUP BY PICKUP\_LOCATION ORDER BY COUNT(PICKUP\_LOCATION) DESC LIMIT 1



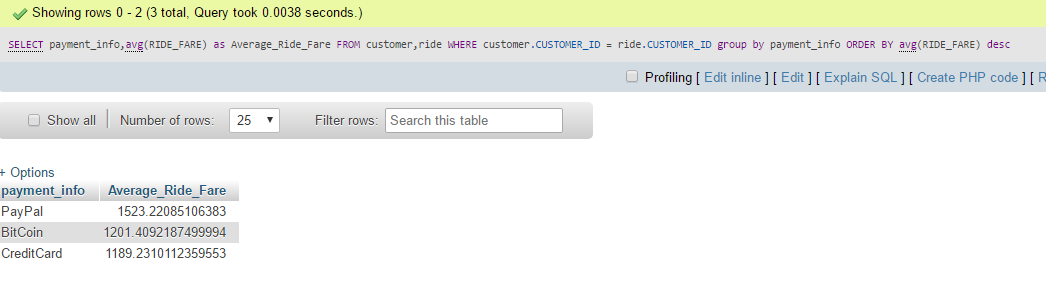
1. **Start time of ride which has highest ride fare with more than one start time**

SELECT START\_TIME FROM RIDE GROUP BY START\_TIME HAVING COUNT(START\_TIME) >1

ORDER BY avg(RIDE\_FARE) DESC limit 1



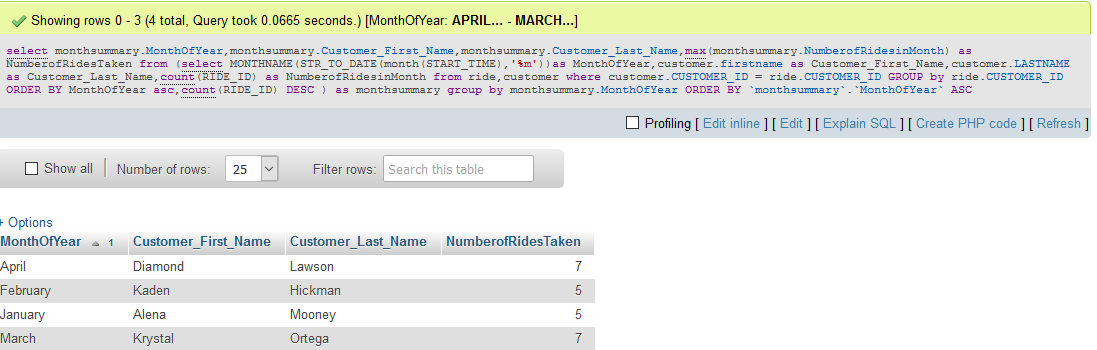
1. **Average ride fare per payment method**

SELECT payment\_info ,avg(RIDE\_FARE) FROM customer,ride WHERE customer.CUSTOMER\_ID ride.CUSTOMER\_ID GROUP BY payment\_info ORDER BY avg(RIDE\_FARE) DESC

1. **Customer details with highest number of rides per month.**

Select monthsummary.MonthOfYear, monthsummary.Customer\_First\_Name, monthsummary.Customer\_Last\_Name, max(monthsummary.NumberofRidesinMonth) as NumberofRidesTaken from (select MONTHNAME(STR\_TO\_DATE(month(START\_TIME),'%m')) as MonthOfYear,customer.firstname as Customer\_First\_Name,customer.LASTNAME as Customer\_Last\_Name,count(RIDE\_ID) as NumberofRidesinMonth from ride,customer where customer.CUSTOMER\_ID = ride.CUSTOMER\_ID GROUP by ride.CUSTOMER\_ID ORDER BY MonthOfYear asc,count(RIDE\_ID) DESC ) as monthsummary group by monthsummary.MonthOfYear

ORDER BY `monthsummary`.`MonthOfYear` ASC



1. **Customer and Driver details who have shared maximum number of rides**

SELECT driver.DRIVER\_ID ,driver.FIRSTNAME AS driver\_firstname , driver.LASTNAME AS driver\_lastname, customer.CUSTOMER\_ID,customer.FIRSTNAME AS customer\_firstname, customer.LASTNAME AS customer\_lastname , NoOfRides FROM customer , driver ,(

SELECT driver\_id,CUSTOMER\_ID ,COUNT(ride\_id) as NoOfRides from ride GROUP BY driver\_id,customer\_id

ORDER by noofrides DESC limit 1) AS tbl1 WHERE customer.CUSTOMER\_ID = tbl1.customer\_id AND driver.DRIVER\_ID = tbl1.driver\_id

