SQL Project

Group:- Lyft It
Thao Nguyen(W1630222), Surbhi Zambad(W1628802), Anjali Gupta(W1628563), Amey Darwhekar(W 1609959), Shilpi Kumari(W1628645)

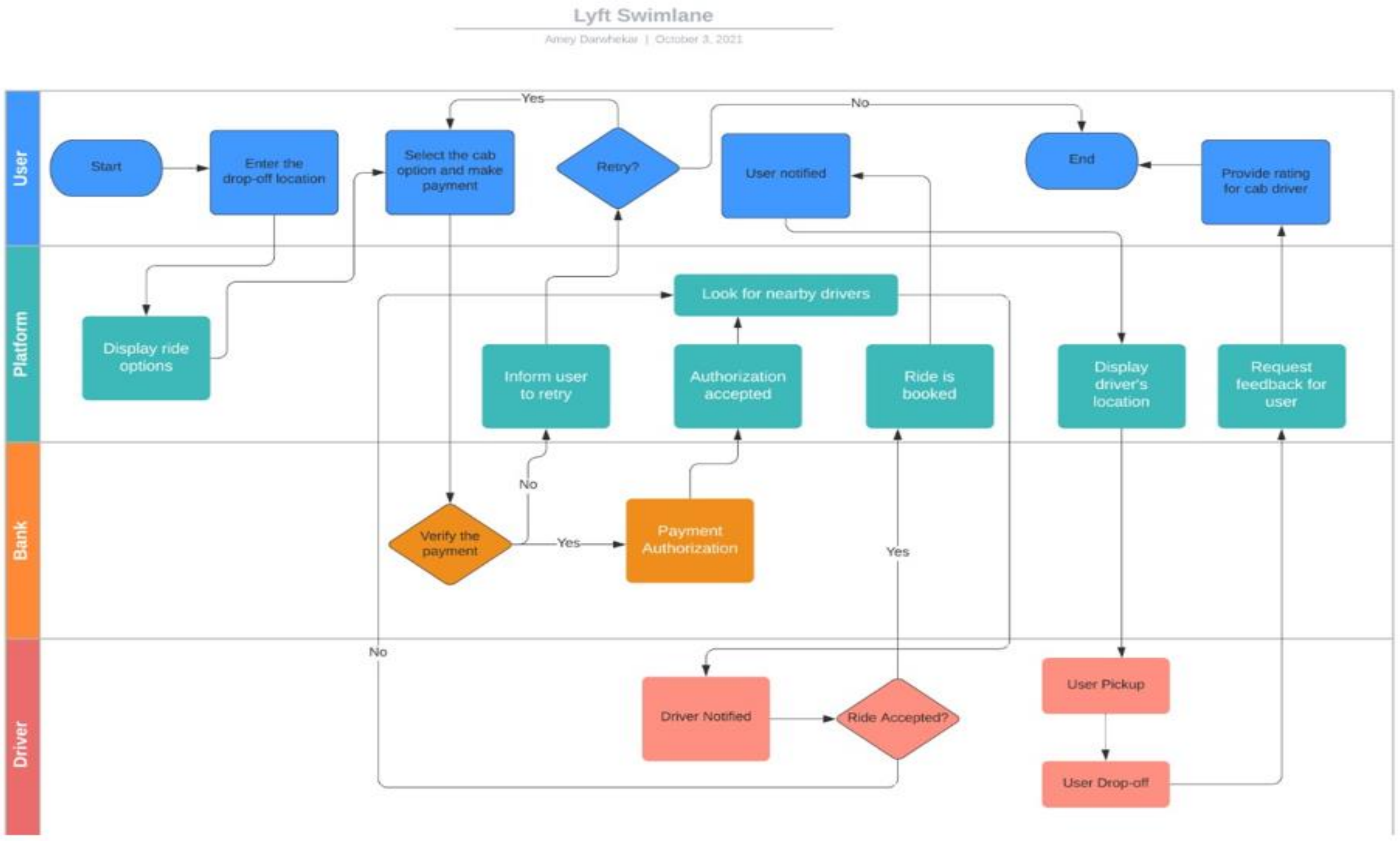
Agenda

Business Case Scenario:- Booking a lyft

It is Saturday night after a long week of exams. Everyone is going out to celebrate the end of it. Shilpi has made plan to go bowling with her friends too. 7:30 pm, she login into her Lyft app to book a ride from Littleton street to the bowling alley on Alameda street. She enters the pick-up and drop-off locations. Payment detail is already saved on the app for her account. The app shows many ride options with their respective pickup time and price. Shilpi chooses the standard sedan option and money is automatically deducts from the card she saved on the app. 2 minutes later, she receives a text on her phone that a driver named Tom is on the way to pick her up in sedan 6TRJ244. Lyft app allows her to see the driver's picture, contact number, rating, and his real time location.

At 7:40 pm, she receives another text saying her previous ride was cancelled and her money will be refunded into the account within 24 hours. She books another ride in the same process. The new driver arrives at 7:45 pm. Shilpi gets in and she can track the ride from the app. When the car is near Alameda, she receives a notification telling her that her destination is coming up. Few minutes later, the driver drops her off and Shilpi gets notified by the app that ride is completed. Finally, Shilpi rates the driver and the ride.

Swim Lane Diagram



Entity-Relationship Diagram

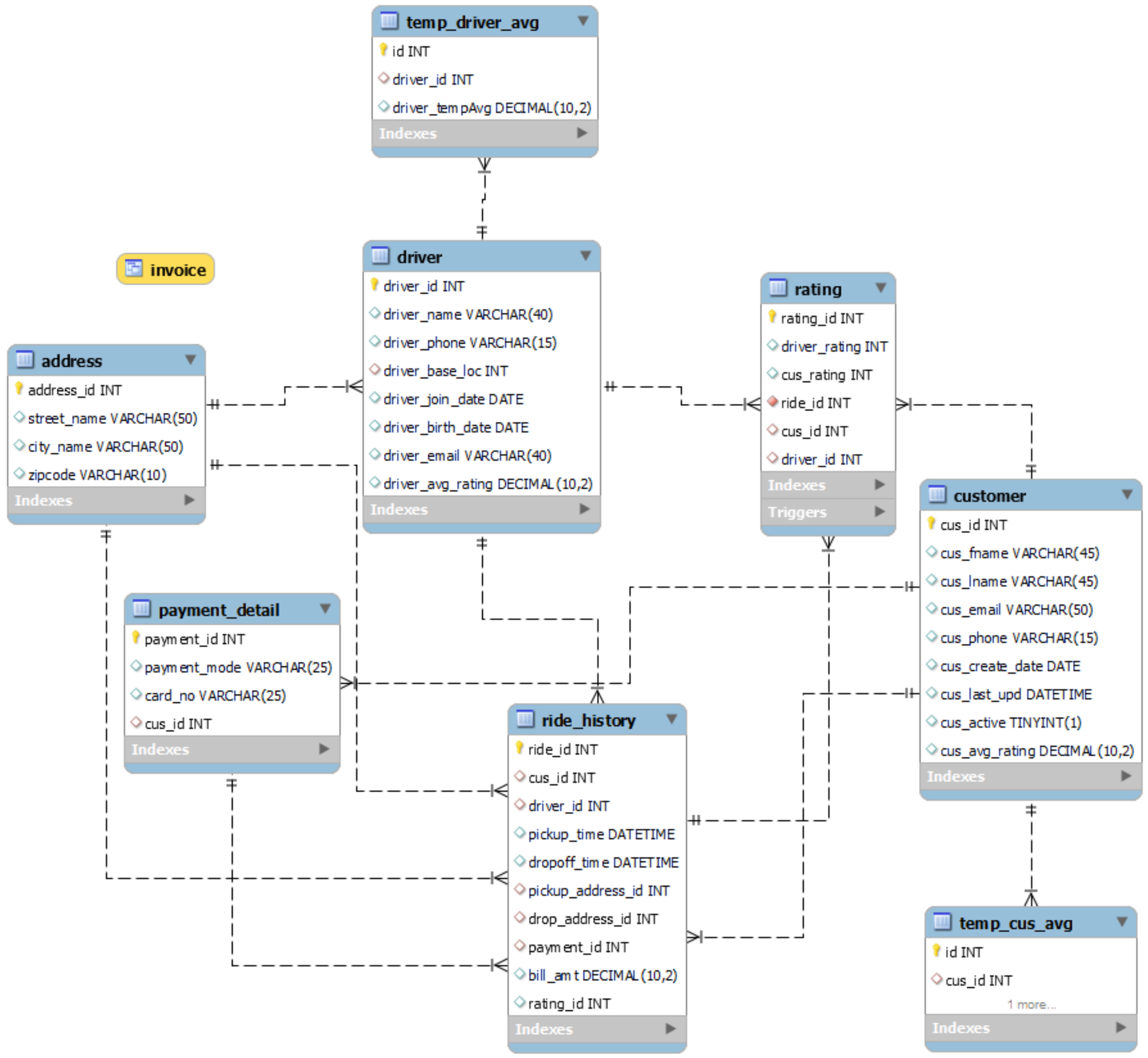
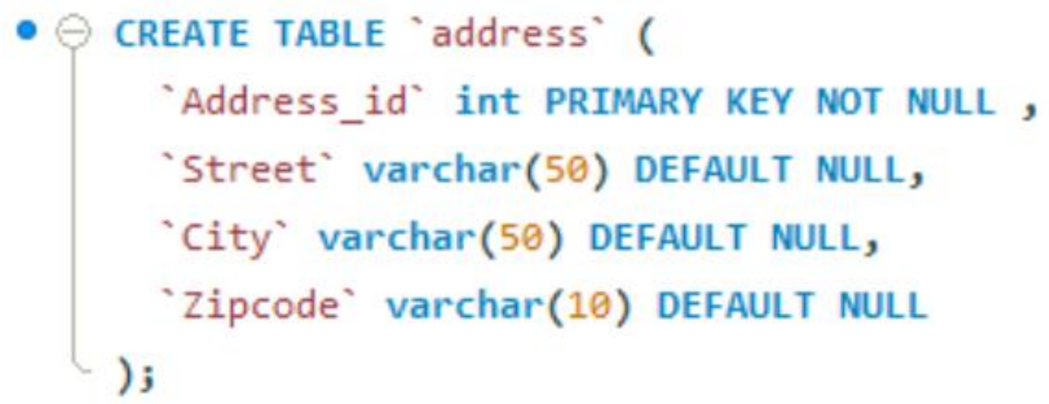


Table Creation – Customer Table

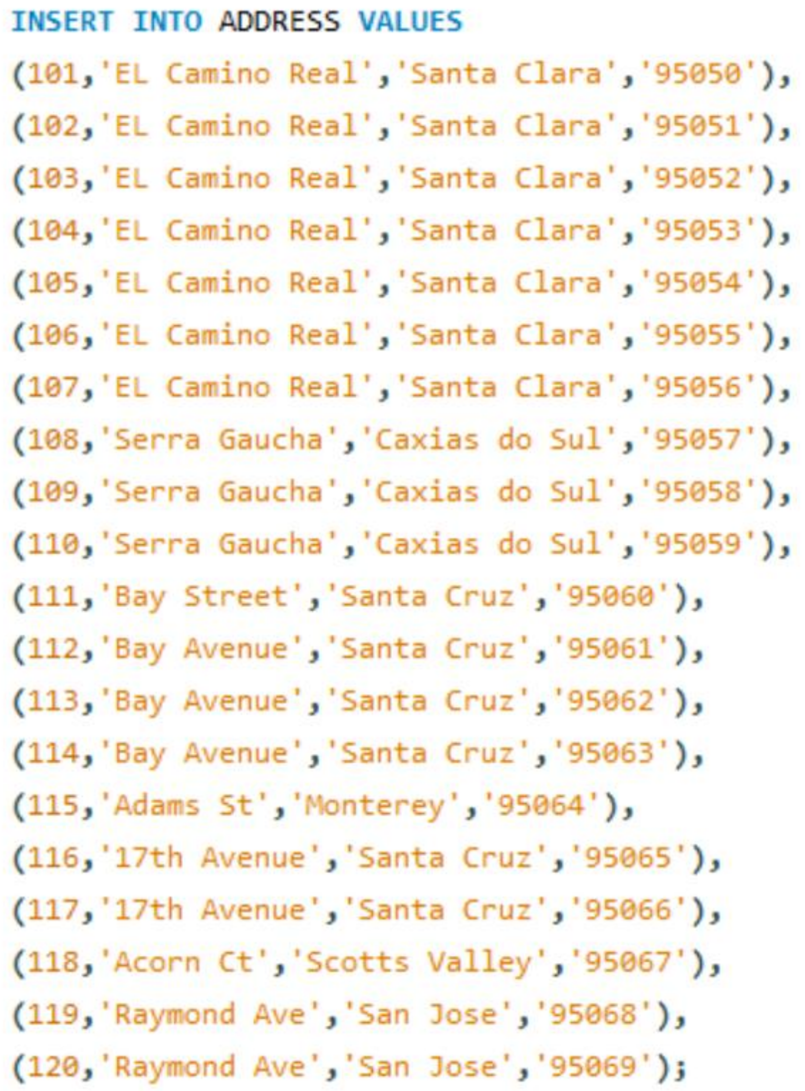
```sql
CREATE TABLE `customer` (
  `cus_id` int NOT NULL,
  `cus_fname` varchar(45) DEFAULT NULL,
  `cus_lname` varchar(45) DEFAULT NULL,
  `cus_email` varchar(50) DEFAULT NULL,
  `cus_phone` varchar(15) DEFAULT NULL,
  `cus_create_date` date DEFAULT NULL,
  `cus_last_upd` datetime DEFAULT NULL,
  `cus_active` tinyint(1) DEFAULT NULL,
  `cus_avg_rating` DECIMAL,
  PRIMARY KEY (`cus_id`)
);
```

```sql
INSERT INTO customer
VALUES (1, 'Mary', 'Smith', 'marysm@lyftcustomer.com', '123-456-7890', '2019-01-08', '2019-05-23', true,null),
(2, 'Patricia', 'Johnson', 'patjohn@lyftcustomer.com', '456-123-7890', '2019-05-24', '2019-05-23', true,null),
(3, 'John', 'Smith', 'johnsm@lyftcustomer.com', '890-123-7890', '2008-08-07', '2019-05-23', true,null),
(4, 'Maria', 'Jones', 'mariajo@lyftcustomer.com', '765-123-7890', '2019-05-23', '2019-07-23', true,null),
(5, 'Barbara', 'Brown', 'barbarabr@lyftcustomer.com', '765-789-7890', '2020-05-23', '2019-08-23', true,null),
(6, 'Madison', 'Peter', 'maddiepeter@lyftcustomer.com', '763-123-7890', '2020-04-23', '2019-08-23', true,null),
(7, 'Daniel', 'Miller', 'danmiller@lyftcustomer.com', '765-123-8908', '2019-08-08', '2019-05-23', true,null),
(8, 'John', 'Jones', 'johnjo@lyftcustomer.com', '765-999-7890', '2019-08-08', '2019-05-23', true,null),
(9,'Susan', 'William', 'susanwilliam@lyftcustomer.com', '503-123-7890', '2019-05-23', '2019-05-23', true,null),
(10, 'Sarah', 'Johnson', 'sarahjohnson@lyftcustomer.com', '408-099-7890', '2019-05-23', '2019-05-23', true,null),
(11, 'Susan', 'William', 'susanwilliam@lyftcustomer.com', '503-123-7890', '2019-05-23', '2019-05-23', true,null),
(12, 'Kate', 'Wilson', 'katewil@lyftcustomer.com', '503-123-3478', '2019-05-23', '2019-05-23', true,null),
(13, 'Peter', 'William', 'peterwil@lyftcustomer.com', '980-123-3462','2019-05-23', '2019-05-23', true,null),
(14, 'Andy', 'Simon', 'andysimon@lyftcustomer.com', '648-873-7890', '2019-05-23', '2019-05-23',true,null),
(15, 'Simon', 'Davis', 'simondavis@lyftcustomer.com', '503-123-3721','2019-05-23', '2019-05-23', true,null),
(16, 'Jennifer', 'Davison', 'jendav@lyftcustomer.com', '789-342-7890', '2019-05-23', '2019-05-23', true,null),
(17, 'Jacob', 'Jones', 'jjones@lyftcustomer.com', '403-988-3421', '2019-05-23', '2019-05-23', true,null),
(18, 'Tina', 'Jackson', 'tinajac@lyftcustomer.com', '203-643-3421','2019-05-23', '2019-05-23', true,null),
(19, 'Isabelle', 'Johnson', 'isabellej@lyftcustomer.com', '879-463-7890', '2019-05-23', '2019-05-23',true,null),
(20, 'Jackson', 'White', 'jackwhite@lyftcustomer.com', '404-987-3479','2019-05-23', '2019-05-23', false,null);
```

Table Creation – Address Table

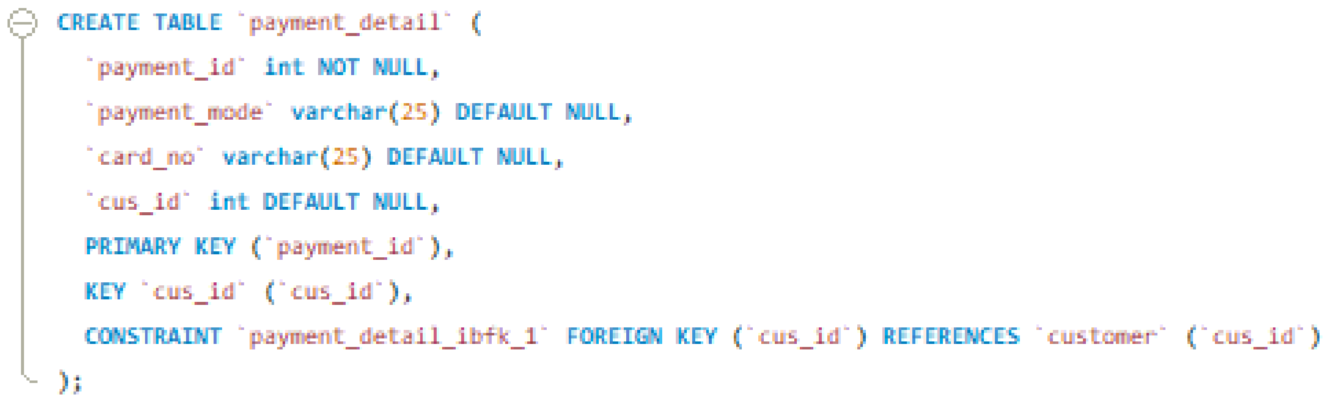


```sql
CREATE TABLE `address` (
  `Address_id` int PRIMARY KEY NOT NULL ,
  `Street` varchar(50) DEFAULT NULL,
  `City` varchar(50) DEFAULT NULL,
  `Zipcode` varchar(10) DEFAULT NULL
);
```

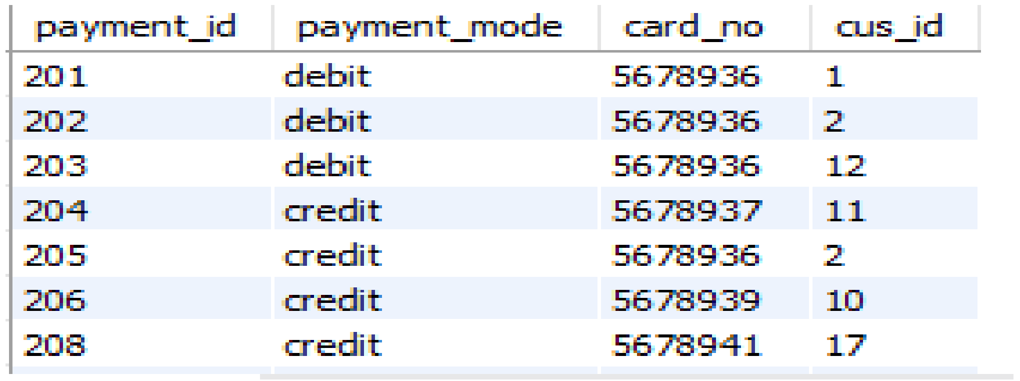


```sql
INSERT INTO ADDRESS VALUES
(101,'EL Camino Real','Santa Clara','95050'),
(102,'EL Camino Real','Santa Clara','95051'),
(103,'EL Camino Real','Santa Clara','95052'),
(104,'EL Camino Real','Santa Clara','95053'),
(105,'EL Camino Real','Santa Clara','95054'),
(106,'EL Camino Real','Santa Clara','95055'),
(107,'EL Camino Real','Santa Clara','95056'),
(108,'Serra Gaucha','Caxias do Sul','95057'),
(109,'Serra Gaucha','Caxias do Sul','95058'),
(110,'Serra Gaucha','Caxias do Sul','95059'),
(111,'Bay Street','Santa Cruz','95060'),
(112,'Bay Avenue','Santa Cruz','95061'),
(113,'Bay Avenue','Santa Cruz','95062'),
(114,'Bay Avenue','Santa Cruz','95063'),
(115,'Adams St','Monterey','95064'),
(116,'17th Avenue','Santa Cruz','95065'),
(117,'17th Avenue','Santa Cruz','95066'),
(118,'Acorn Ct','Scotts Valley','95067'),
(119,'Raymond Ave','San Jose','95068'),
(120,'Raymond Ave','San Jose','95069');
```

Table Creation – Payment Detail Table

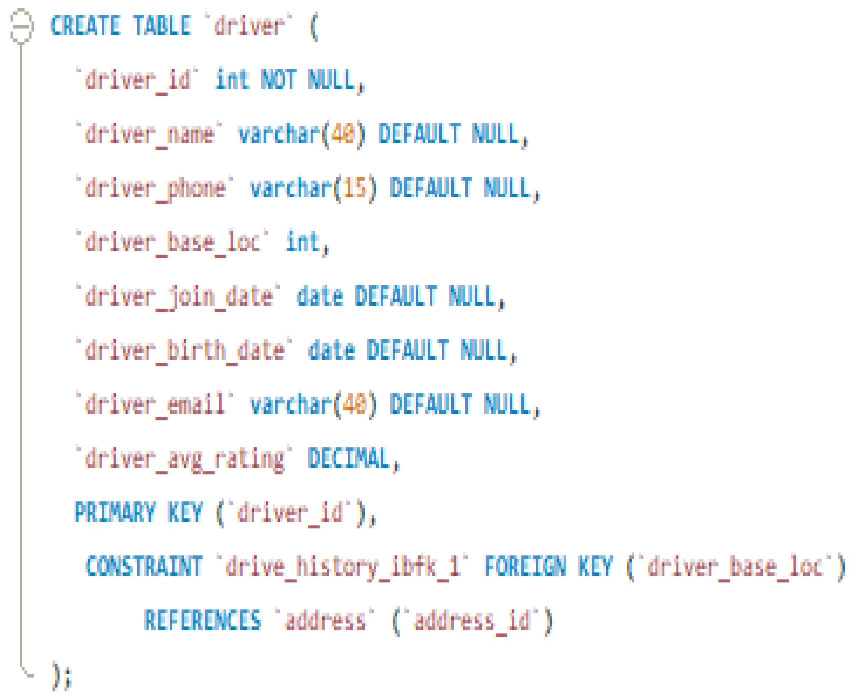


```
CREATE TABLE `payment_detail` (
  `payment_id` int NOT NULL,
  `payment_mode` varchar(25) DEFAULT NULL,
  `card_no` varchar(25) DEFAULT NULL,
  `cus_id` int DEFAULT NULL,
  PRIMARY KEY (`payment_id`),
  KEY `cus_id` (`cus_id`),
  CONSTRAINT `payment_detail_ibfk_1` FOREIGN KEY (`cus_id`) REFERENCES `customer` (`cus_id`)
);
```

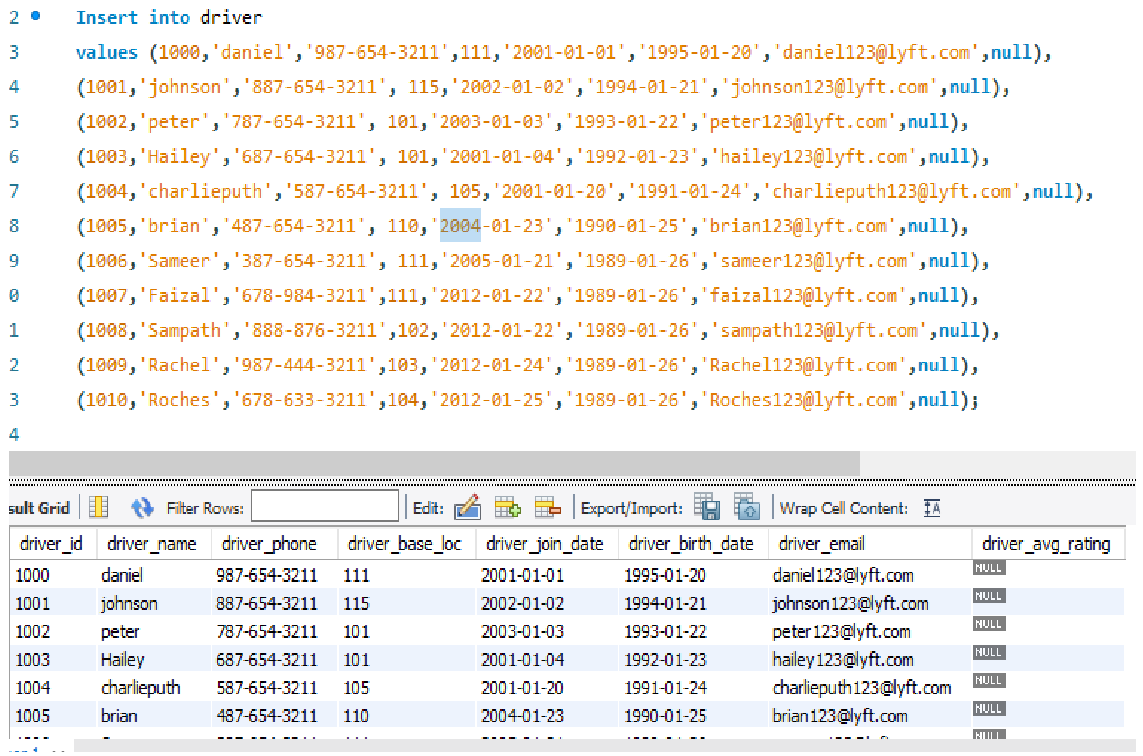


payment_id	payment_mode	card_no	cus_id
201	debit	5678936	1
202	debit	5678936	2
203	debit	5678936	12
204	credit	5678937	11
205	credit	5678936	2
206	credit	5678939	10
208	credit	5678941	17

Table Creation – Driver Table



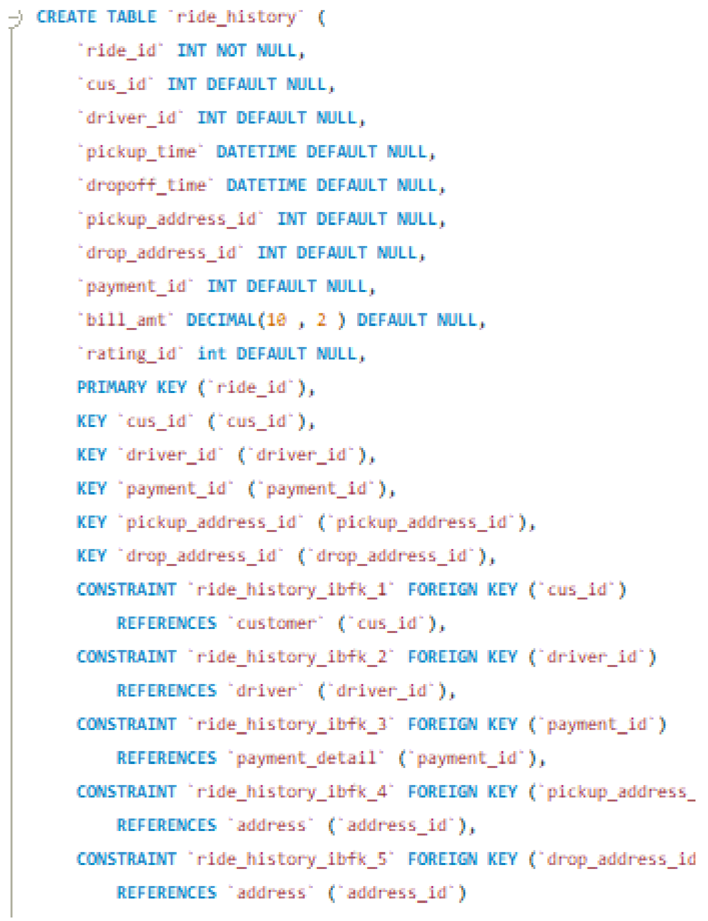
```
CREATE TABLE `driver` (
  `driver_id` int NOT NULL,
  `driver_name` varchar(40) DEFAULT NULL,
  `driver_phone` varchar(15) DEFAULT NULL,
  `driver_base_loc` int,
  `driver_join_date` date DEFAULT NULL,
  `driver_birth_date` date DEFAULT NULL,
  `driver_email` varchar(40) DEFAULT NULL,
  `driver_avg_rating` DECIMAL,
  PRIMARY KEY (`driver_id`),
   CONSTRAINT `drive_history_ibfk_1` FOREIGN KEY (`driver_base_loc`)
       REFERENCES `address` (`address_id`)
);
```



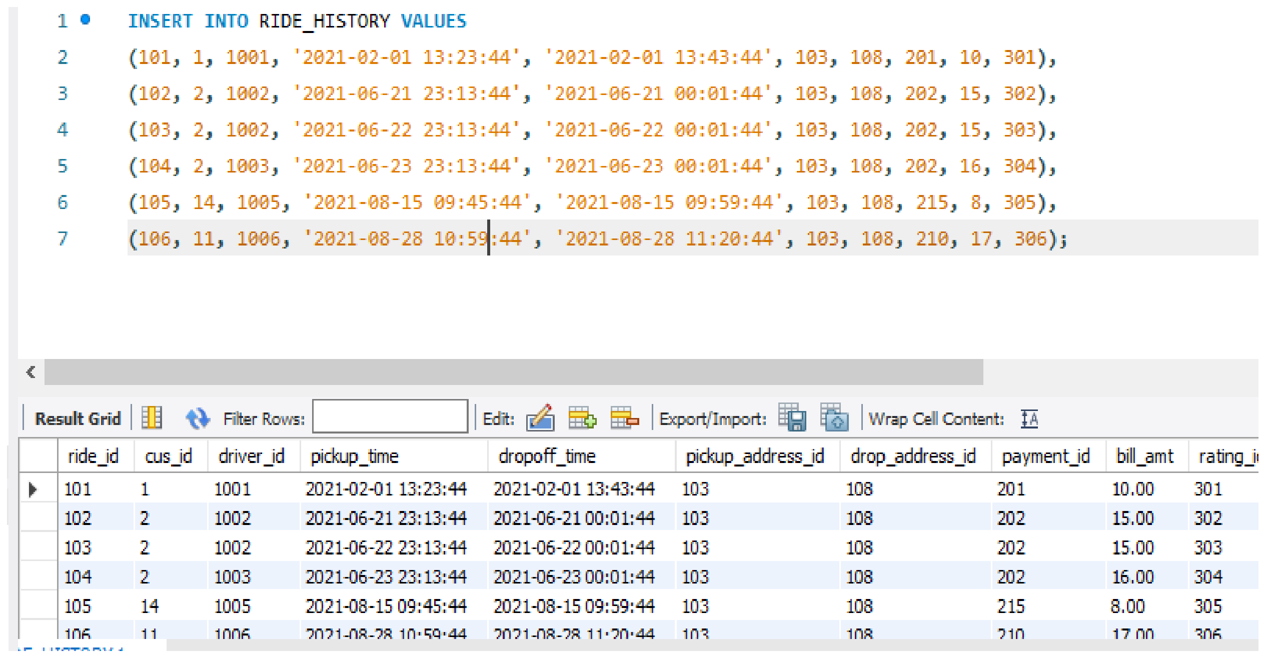
```
Insert into driver
values (1000,'daniel','987-654-3211',111,'2001-01-01','1995-01-20','daniel123@lyft.com',null),
(1001,'johnson','887-654-3211', 115,'2002-01-02','1994-01-21','johnson123@lyft.com',null),
(1002,'peter','787-654-3211', 101,'2003-01-03','1993-01-22','peter123@lyft.com',null),
(1003,'Hailey','687-654-3211', 101,'2001-01-04','1992-01-23','hailey123@lyft.com',null),
(1004,'charlieputh','587-654-3211', 105,'2001-01-20','1991-01-24','charlieputh123@lyft.com',null),
(1005,'brian','487-654-3211', 110,'2004-01-23','1990-01-25','brian123@lyft.com',null),
(1006,'Sameer','387-654-3211', 111,'2005-01-21','1989-01-26','sameer123@lyft.com',null),
(1007,'Faizal','678-984-3211',111,'2012-01-22','1989-01-26','faizal123@lyft.com',null),
(1008,'Sampath','888-876-3211',102,'2012-01-22','1989-01-26','sampath123@lyft.com',null),
(1009,'Rachel','987-444-3211',103,'2012-01-24','1989-01-26','Rachel123@lyft.com',null),
(1010,'Roches','678-633-3211',104,'2012-01-25','1989-01-26','Roches123@lyft.com',null);
```

driver_id	driver_name	driver_phone	driver_base_loc	driver_join_date	driver_birth_date	driver_email	driver_avg_rating
1000	daniel	987-654-3211	111	2001-01-01	1995-01-20	daniel123@lyft.com	NULL
1001	johnson	887-654-3211	115	2002-01-02	1994-01-21	johnson123@lyft.com	NULL
1002	peter	787-654-3211	101	2003-01-03	1993-01-22	peter123@lyft.com	NULL
1003	Hailey	687-654-3211	101	2001-01-04	1992-01-23	hailey123@lyft.com	NULL
1004	charlieputh	587-654-3211	105	2001-01-20	1991-01-24	charlieputh123@lyft.com	NULL
1005	brian	487-654-3211	110	2004-01-23	1990-01-25	brian123@lyft.com	NULL

Table Creation – Ride History Table



```sql
CREATE TABLE `ride_history` (
    `ride_id` INT NOT NULL,
    `cus_id` INT DEFAULT NULL,
    `driver_id` INT DEFAULT NULL,
    `pickup_time` DATETIME DEFAULT NULL,
    `dropoff_time` DATETIME DEFAULT NULL,
    `pickup_address_id` INT DEFAULT NULL,
    `drop_address_id` INT DEFAULT NULL,
    `payment_id` INT DEFAULT NULL,
    `bill_amt` DECIMAL(10 , 2 ) DEFAULT NULL,
    `rating_id` int DEFAULT NULL,
    PRIMARY KEY (`ride_id`),
    KEY `cus_id` (`cus_id`),
    KEY `driver_id` (`driver_id`),
    KEY `payment_id` (`payment_id`),
    KEY `pickup_address_id` (`pickup_address_id`),
    KEY `drop_address_id` (`drop_address_id`),
    CONSTRAINT `ride_history_ibfk_1` FOREIGN KEY (`cus_id`)
        REFERENCES `customer` (`cus_id`),
    CONSTRAINT `ride_history_ibfk_2` FOREIGN KEY (`driver_id`)
        REFERENCES `driver` (`driver_id`),
    CONSTRAINT `ride_history_ibfk_3` FOREIGN KEY (`payment_id`)
        REFERENCES `payment_detail` (`payment_id`),
    CONSTRAINT `ride_history_ibfk_4` FOREIGN KEY (`pickup_address_
        REFERENCES `address` (`address_id`),
    CONSTRAINT `ride_history_ibfk_5` FOREIGN KEY (`drop_address_id
        REFERENCES `address` (`address_id`)
```

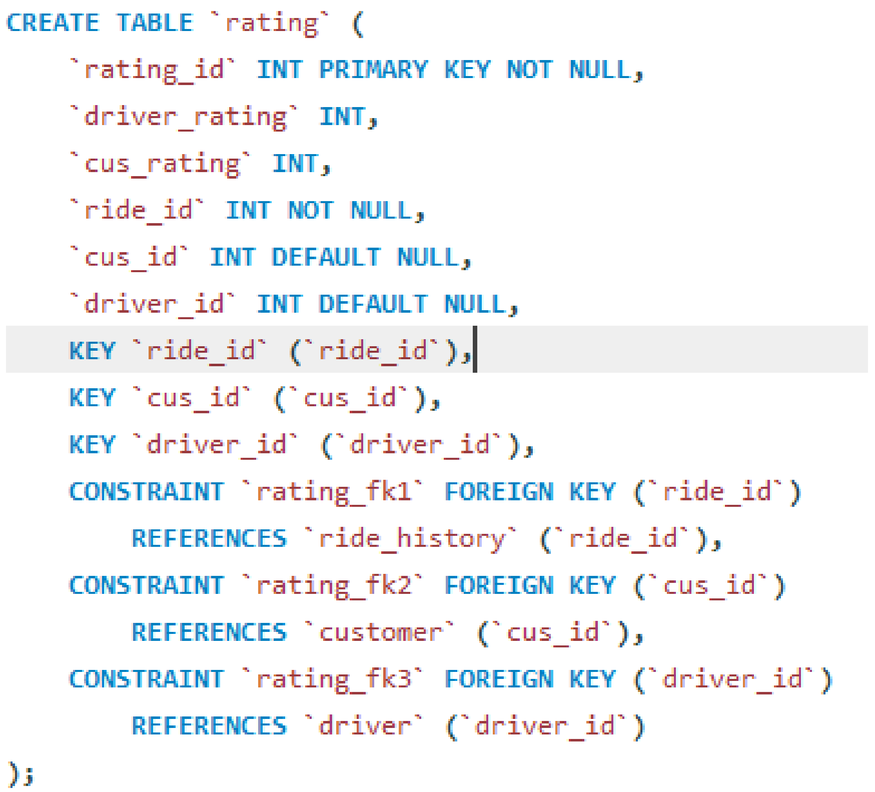


```sql
INSERT INTO RIDE_HISTORY VALUES
(101, 1, 1001, '2021-02-01 13:23:44', '2021-02-01 13:43:44', 103, 108, 201, 10, 301),
(102, 2, 1002, '2021-06-21 23:13:44', '2021-06-21 00:01:44', 103, 108, 202, 15, 302),
(103, 2, 1002, '2021-06-22 23:13:44', '2021-06-22 00:01:44', 103, 108, 202, 15, 303),
(104, 2, 1003, '2021-06-23 23:13:44', '2021-06-23 00:01:44', 103, 108, 202, 16, 304),
(105, 14, 1005, '2021-08-15 09:45:44', '2021-08-15 09:59:44', 103, 108, 215, 8, 305),
(106, 11, 1006, '2021-08-28 10:59:44', '2021-08-28 11:20:44', 103, 108, 210, 17, 306);
```

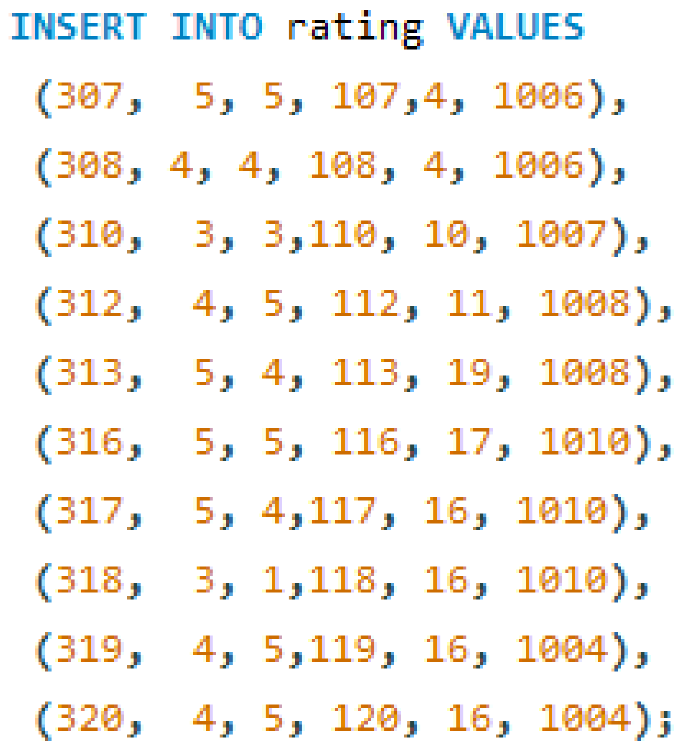
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

```
ride_id  cus_id  driver_id  pickup_time          dropoff_time         pickup_address_id  drop_address_id  payment_id  bill_amt  rating_i
101      1       1001       2021-02-01 13:23:44  2021-02-01 13:43:44  103                108              201         10.00     301
102      2       1002       2021-06-21 23:13:44  2021-06-21 00:01:44  103                108              202         15.00     302
103      2       1002       2021-06-22 23:13:44  2021-06-22 00:01:44  103                108              202         15.00     303
104      2       1003       2021-06-23 23:13:44  2021-06-23 00:01:44  103                108              202         16.00     304
105      14      1005       2021-08-15 09:45:44  2021-08-15 09:59:44  103                108              215         8.00      305
106      11      1006       2021-08-28 10:59:44  2021-08-28 11:20:44  103                108              210         17.00     306
```

Table Creation – Rating Table



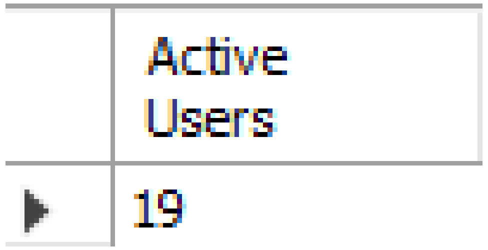
```
CREATE TABLE `rating` (
    `rating_id` INT PRIMARY KEY NOT NULL,
    `driver_rating` INT,
    `cus_rating` INT,
    `ride_id` INT NOT NULL,
    `cus_id` INT DEFAULT NULL,
    `driver_id` INT DEFAULT NULL,
    KEY `ride_id` (`ride_id`),
    KEY `cus_id` (`cus_id`),
    KEY `driver_id` (`driver_id`),
    CONSTRAINT `rating_fk1` FOREIGN KEY (`ride_id`)
        REFERENCES `ride_history` (`ride_id`),
    CONSTRAINT `rating_fk2` FOREIGN KEY (`cus_id`)
        REFERENCES `customer` (`cus_id`),
    CONSTRAINT `rating_fk3` FOREIGN KEY (`driver_id`)
        REFERENCES `driver` (`driver_id`)
);
```



```
INSERT INTO rating VALUES
 (307,  5, 5, 107,4, 1006),
 (308, 4, 4, 108, 4, 1006),
 (310,  3, 3,110, 10, 1007),
 (312,  4, 5, 112, 11, 1008),
 (313,  5, 4, 113, 19, 1008),
 (316,  5, 5, 116, 17, 1010),
 (317,  5, 4,117, 16, 1010),
 (318,  3, 1,118, 16, 1010),
 (319,  4, 5,119, 16, 1004),
 (320,  4, 5, 120, 16, 1004);
```

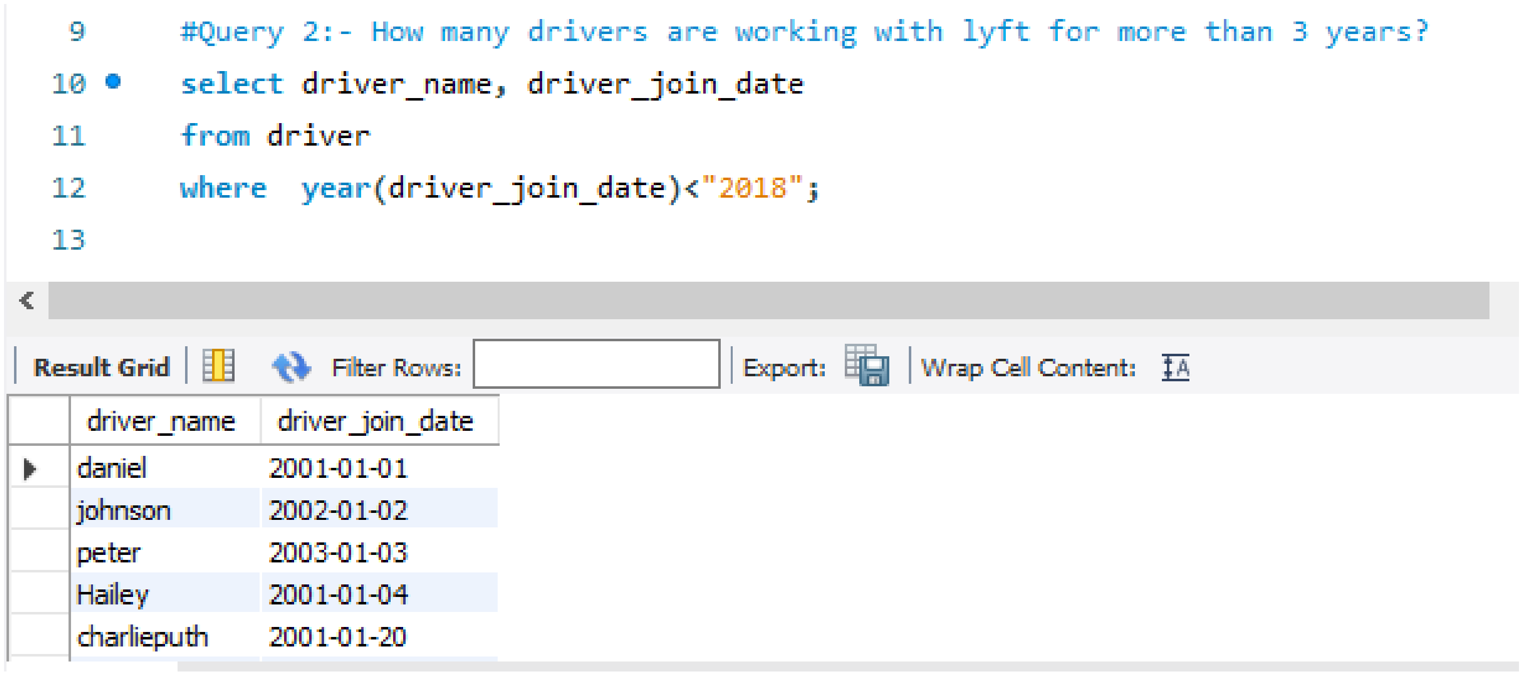
SQL Query

```
#Query 1:- Counting how many customers are active right now?
select count(*) as "Active Users"
from customer
where cus_active = 1;
```



	Active Users
▶	19

SQL Query

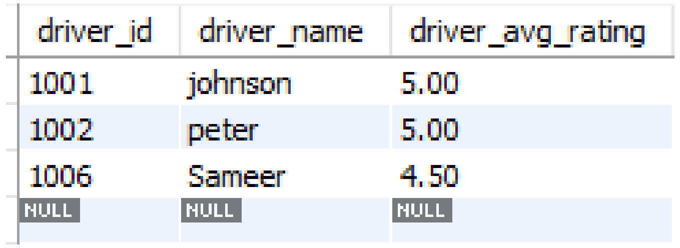


```
#Query 2:- How many drivers are working with lyft for more than 3 years?
select driver_name, driver_join_date
from driver
where  year(driver_join_date)<"2018";
```

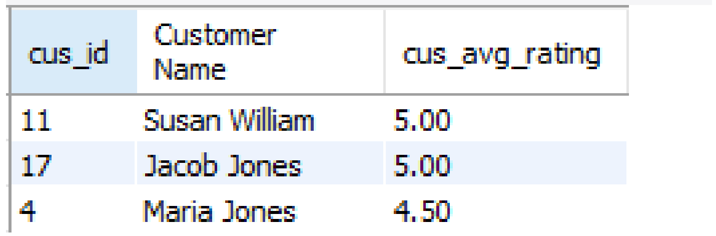
driver_name	driver_join_date
daniel	2001-01-01
johnson	2002-01-02
peter	2003-01-03
Hailey	2001-01-04
charlieputh	2001-01-20

SQL Query

```
#Query 3A:- Finding out top 5 drivers with highest average rating.
select driver_id, driver_name, driver_avg_rating
from driver
order by driver_avg_rating desc
limit 3;

#Query 3B:- Finding the 3 highest rated customers.
select cus_id, CONCAT(cus_fname ,' ', cus_lname) AS 'Customer Name', cus_avg_rating
from customer
order by cus_avg_rating desc
limit 3;
```

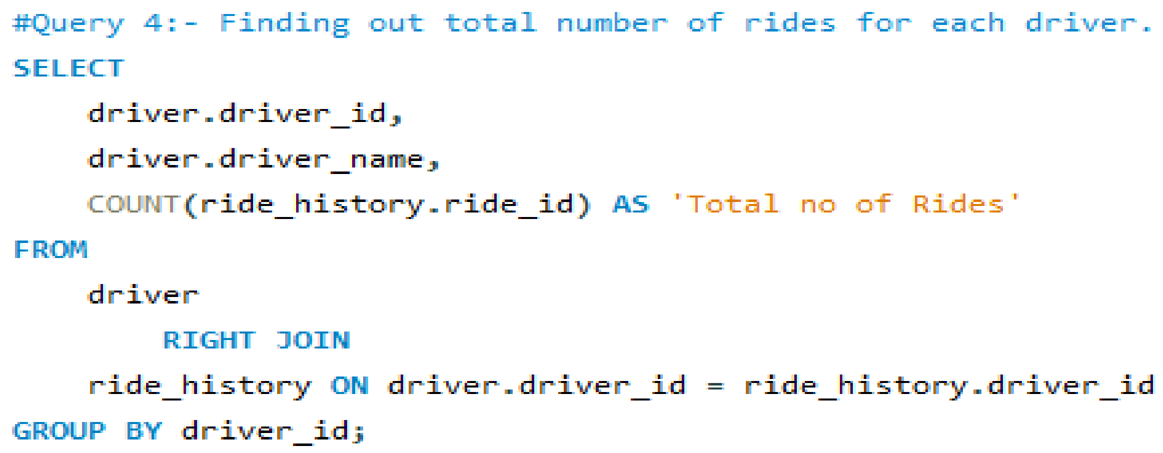


driver_id	driver_name	driver_avg_rating
1001	johnson	5.00
1002	peter	5.00
1006	Sameer	4.50
NULL	NULL	NULL

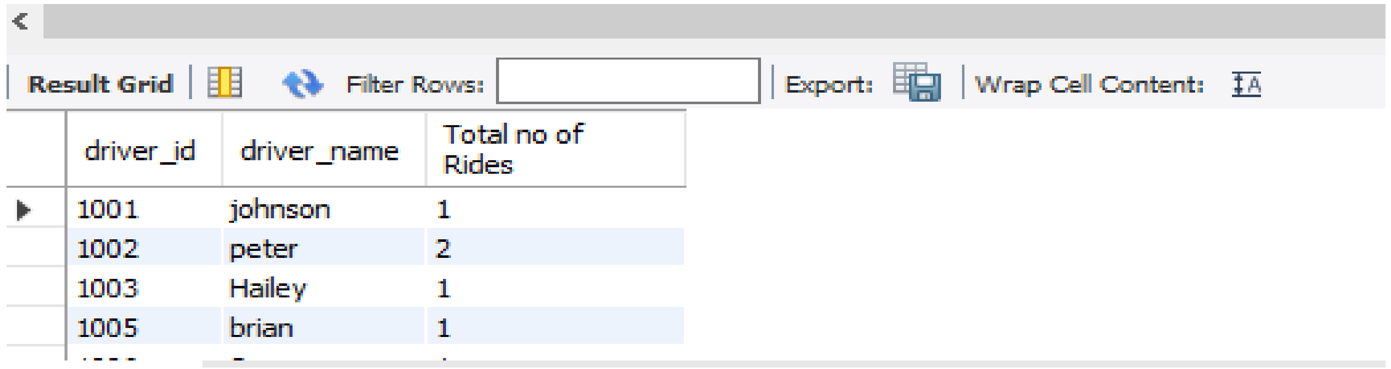


cus_id	Customer Name	cus_avg_rating
11	Susan William	5.00
17	Jacob Jones	5.00
4	Maria Jones	4.50

SQL Query



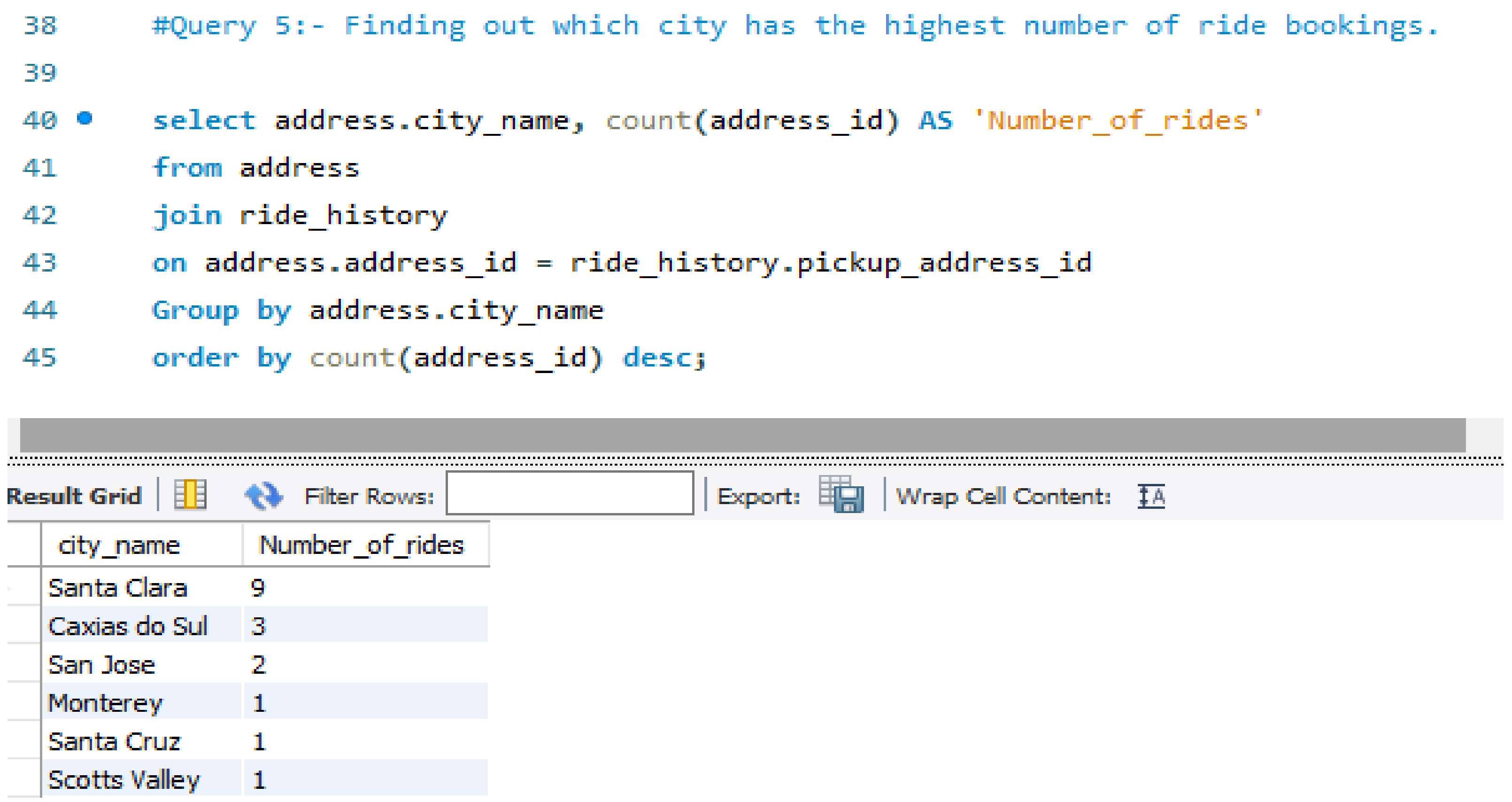
```
#Query 4:- Finding out total number of rides for each driver.
SELECT
    driver.driver_id,
    driver.driver_name,
    COUNT(ride_history.ride_id) AS 'Total no of Rides'
FROM
    driver
        RIGHT JOIN
    ride_history ON driver.driver_id = ride_history.driver_id
GROUP BY driver_id;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

driver_id	driver_name	Total no of Rides
1001	johnson	1
1002	peter	2
1003	Hailey	1
1005	brian	1

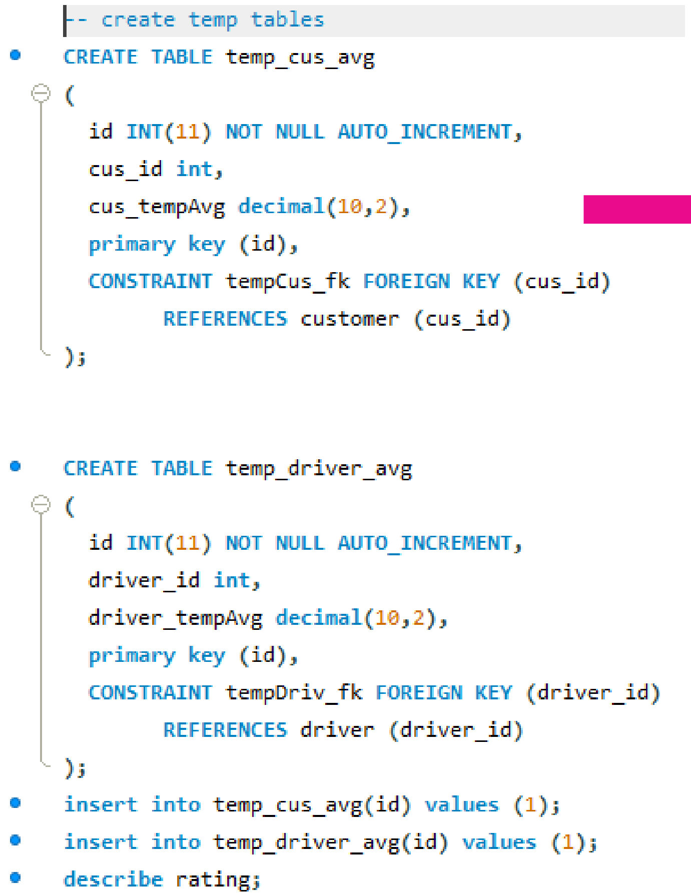
SQL Query



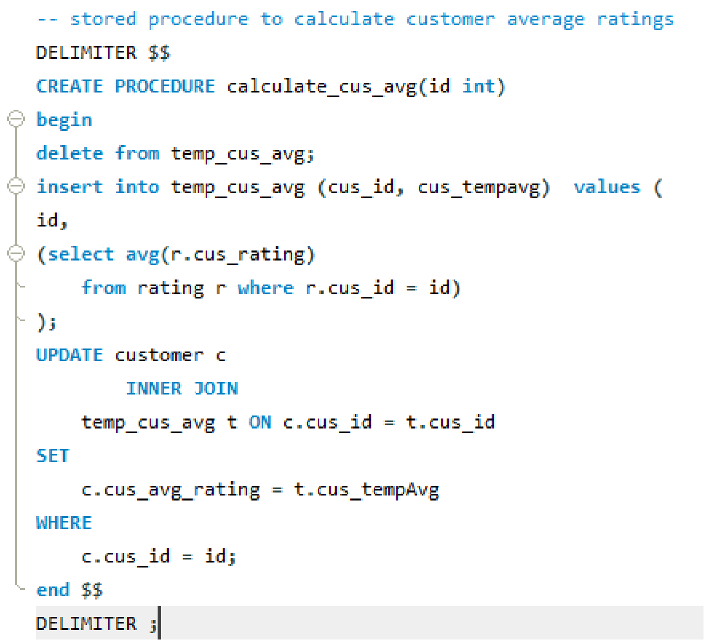
```
#Query 5:- Finding out which city has the highest number of ride bookings.

select address.city_name, count(address_id) AS 'Number_of_rides'
from address
join ride_history
on address.address_id = ride_history.pickup_address_id
Group by address.city_name
order by count(address_id) desc;
```

city_name	Number_of_rides
Santa Clara	9
Caxias do Sul	3
San Jose	2
Monterey	1
Santa Cruz	1
Scotts Valley	1

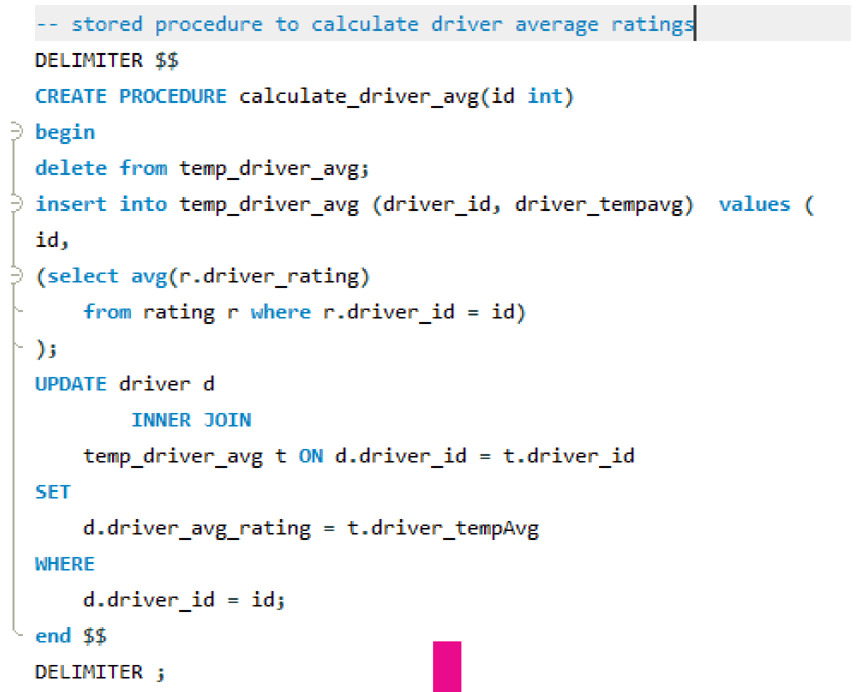
Calculating average ratings



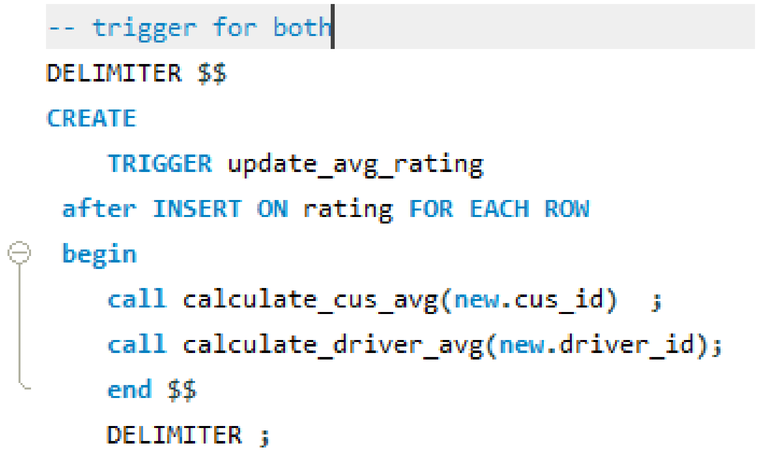
```sql
-- create temp tables
CREATE TABLE temp_cus_avg
(
  id INT(11) NOT NULL AUTO_INCREMENT,
  cus_id int,
  cus_tempAvg decimal(10,2),
  primary key (id),
  CONSTRAINT tempCus_fk FOREIGN KEY (cus_id)
        REFERENCES customer (cus_id)
);

CREATE TABLE temp_driver_avg
(
  id INT(11) NOT NULL AUTO_INCREMENT,
  driver_id int,
  driver_tempAvg decimal(10,2),
  primary key (id),
  CONSTRAINT tempDriv_fk FOREIGN KEY (driver_id)
        REFERENCES driver (driver_id)
);
insert into temp_cus_avg(id) values (1);
insert into temp_driver_avg(id) values (1);
describe rating;
```



```sql
-- stored procedure to calculate customer average ratings
DELIMITER $$
CREATE PROCEDURE calculate_cus_avg(id int)
begin
delete from temp_cus_avg;
insert into temp_cus_avg (cus_id, cus_tempavg)  values (
id,
(select avg(r.cus_rating)
    from rating r where r.cus_id = id)
);
UPDATE customer c
        INNER JOIN
    temp_cus_avg t ON c.cus_id = t.cus_id
SET
    c.cus_avg_rating = t.cus_tempAvg
WHERE
    c.cus_id = id;
end $$
DELIMITER ;
```

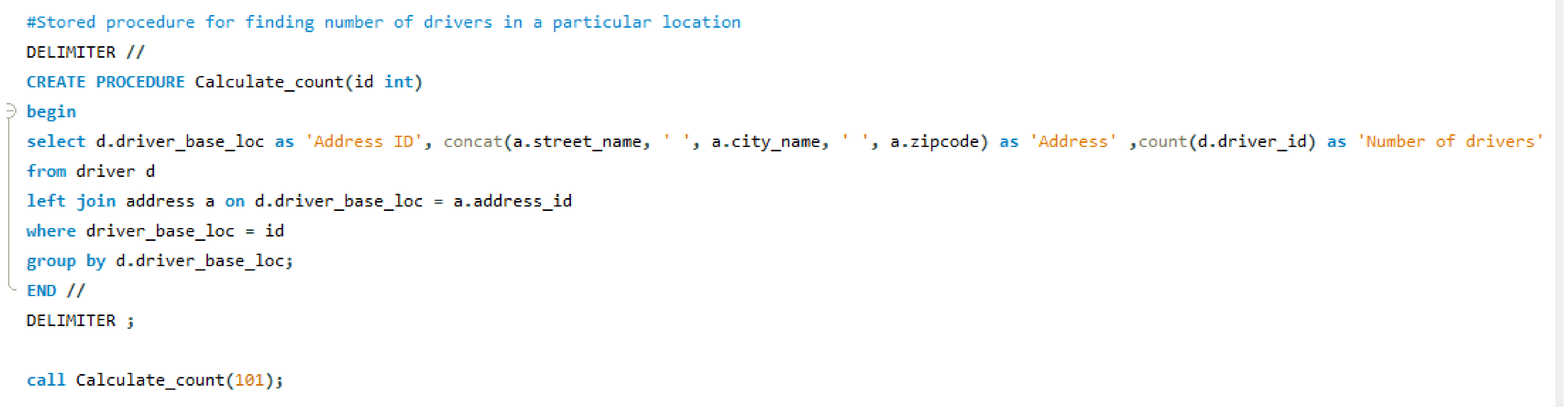


```sql
-- stored procedure to calculate driver average ratings
DELIMITER $$
CREATE PROCEDURE calculate_driver_avg(id int)
begin
delete from temp_driver_avg;
insert into temp_driver_avg (driver_id, driver_tempavg)  values (
id,
(select avg(r.driver_rating)
    from rating r where r.driver_id = id)
);
UPDATE driver d
        INNER JOIN
    temp_driver_avg t ON d.driver_id = t.driver_id
SET
    d.driver_avg_rating = t.driver_tempAvg
WHERE
    d.driver_id = id;
end $$
DELIMITER ;
```



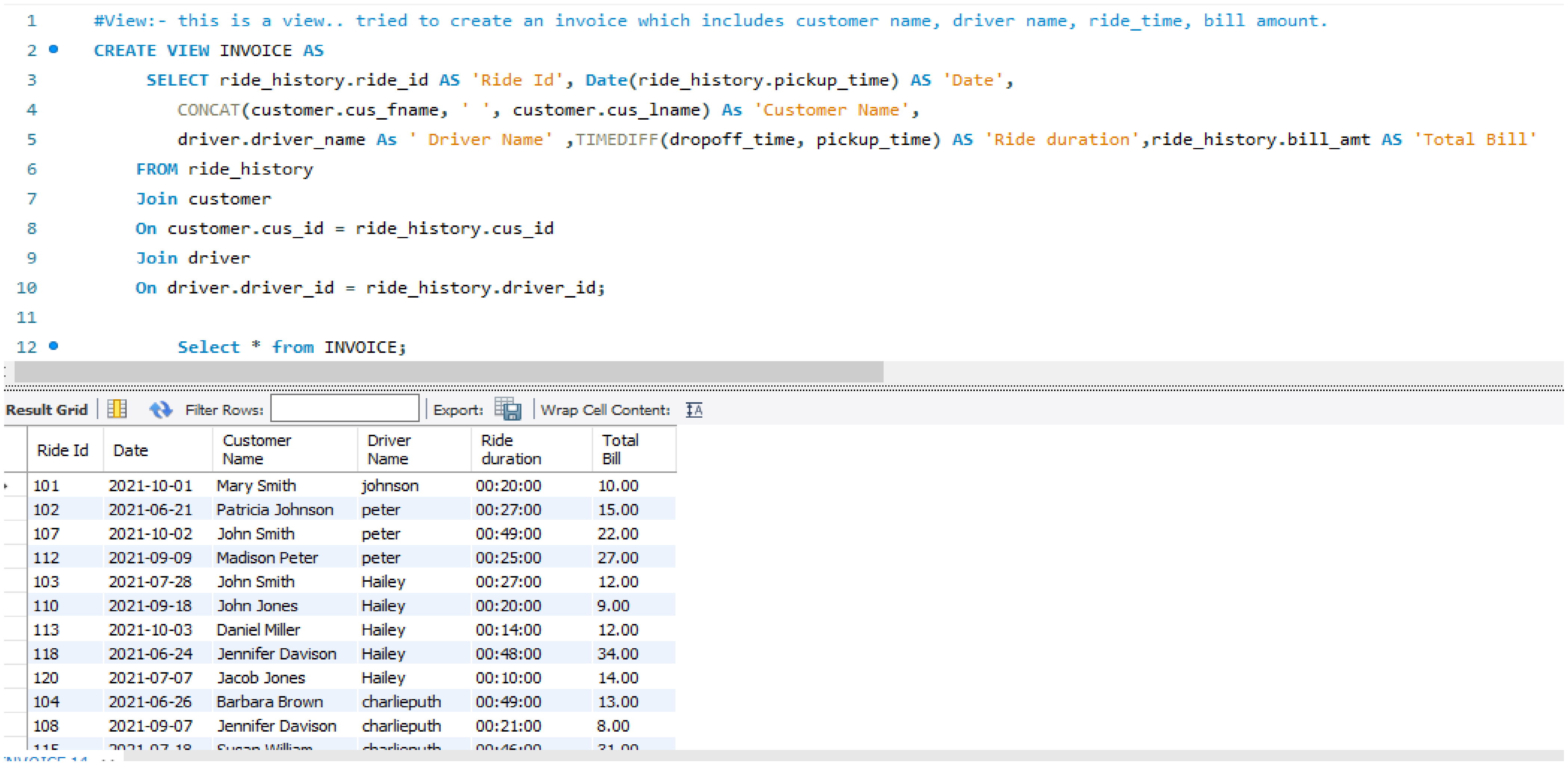
```sql
-- trigger for both
DELIMITER $$
CREATE
    TRIGGER update_avg_rating
 after INSERT ON rating FOR EACH ROW
 begin
    call calculate_cus_avg(new.cus_id)  ;
    call calculate_driver_avg(new.driver_id);
    end $$
    DELIMITER ;
```

SQL Stored Procedure



```
#Stored procedure for finding number of drivers in a particular location
DELIMITER //
CREATE PROCEDURE Calculate_count(id int)
begin
select d.driver_base_loc as 'Address ID', concat(a.street_name, ' ', a.city_name, ' ', a.zipcode) as 'Address' ,count(d.driver_id) as 'Number of drivers'
from driver d
left join address a on d.driver_base_loc = a.address_id
where driver_base_loc = id
group by d.driver_base_loc;
END //
DELIMITER ;

call Calculate_count(101);
```

SQL Views



```sql
#View:- this is a view.. tried to create an invoice which includes customer name, driver name, ride_time, bill amount.
CREATE VIEW INVOICE AS
     SELECT ride_history.ride_id AS 'Ride Id', Date(ride_history.pickup_time) AS 'Date',
        CONCAT(customer.cus_fname, ' ', customer.cus_lname) As 'Customer Name',
        driver.driver_name As ' Driver Name' ,TIMEDIFF(dropoff_time, pickup_time) AS 'Ride duration',ride_history.bill_amt AS 'Total Bill'
    FROM ride_history
    Join customer
    On customer.cus_id = ride_history.cus_id
    Join driver
    On driver.driver_id = ride_history.driver_id;

        Select * from INVOICE;
```

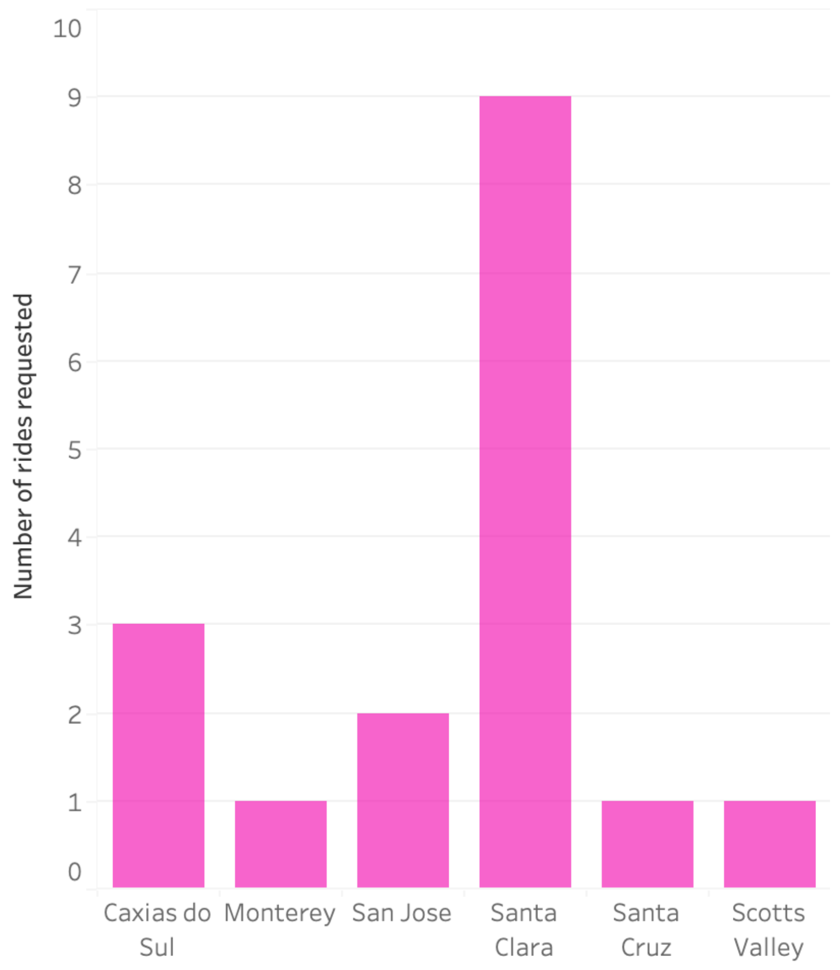
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Ride Id	Date	Customer Name	Driver Name	Ride duration	Total Bill
101	2021-10-01	Mary Smith	johnson	00:20:00	10.00
102	2021-06-21	Patricia Johnson	peter	00:27:00	15.00
107	2021-10-02	John Smith	peter	00:49:00	22.00
112	2021-09-09	Madison Peter	peter	00:25:00	27.00
103	2021-07-28	John Smith	Hailey	00:27:00	12.00
110	2021-09-18	John Jones	Hailey	00:20:00	9.00
113	2021-10-03	Daniel Miller	Hailey	00:14:00	12.00
118	2021-06-24	Jennifer Davison	Hailey	00:48:00	34.00
120	2021-07-07	Jacob Jones	Hailey	00:10:00	14.00
104	2021-06-26	Barbara Brown	charlieputh	00:49:00	13.00
108	2021-09-07	Jennifer Davison	charlieputh	00:21:00	8.00
115	2021-07-18	Susan William	charlieputh	00:46:00	31.00

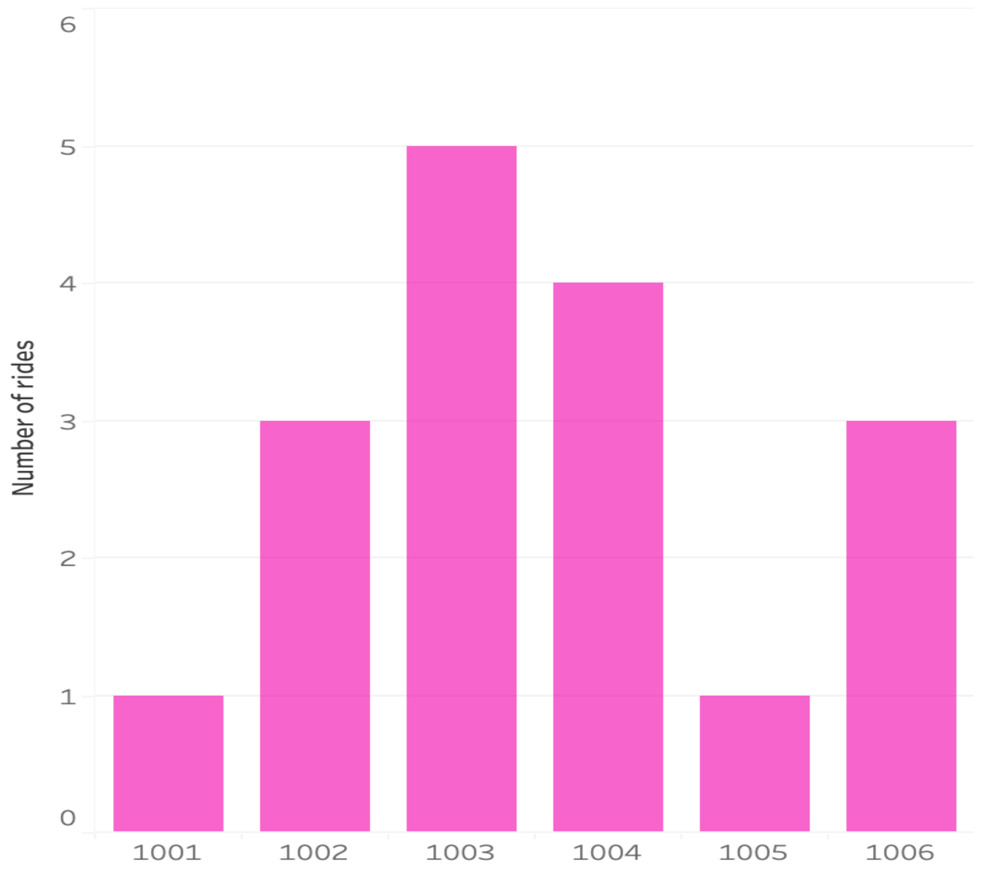
INVOICE 14

Visualization-Rides

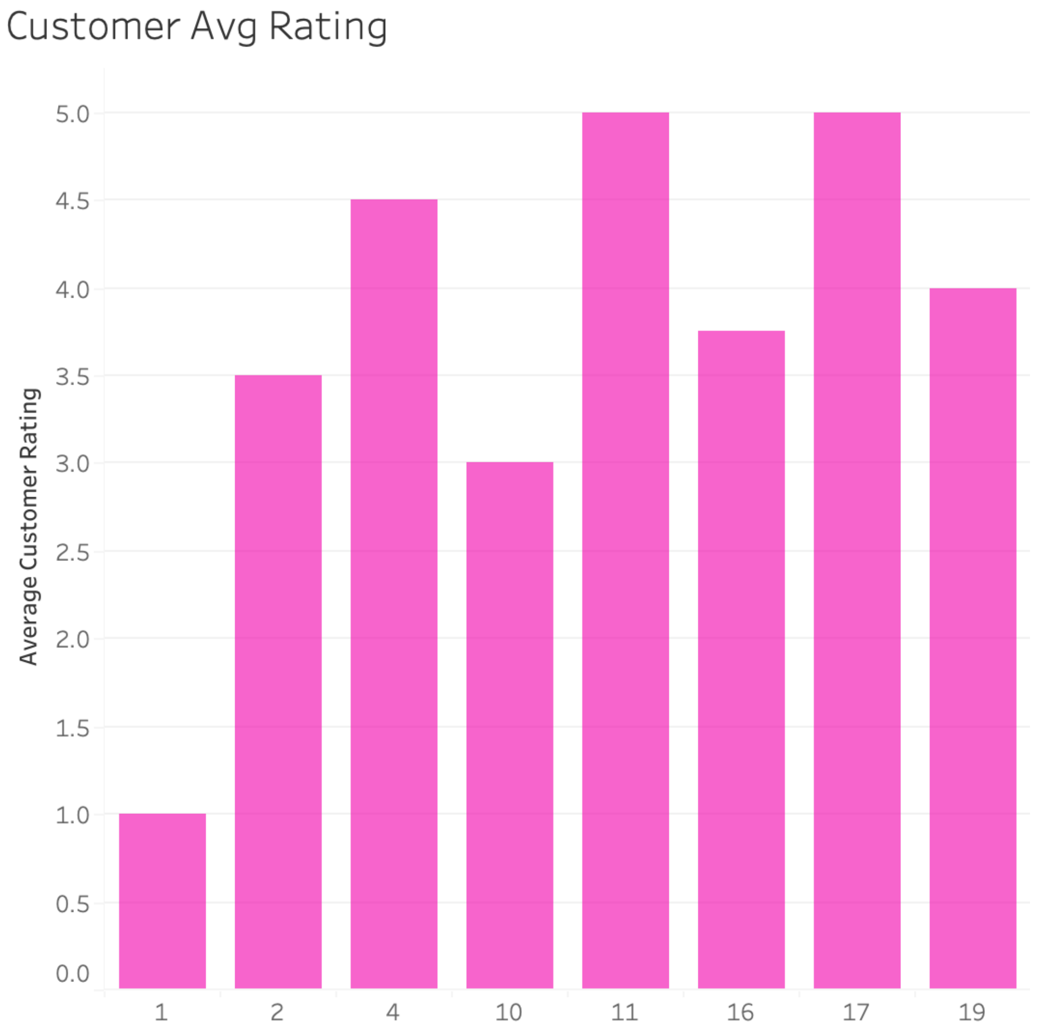
City wise ride distribution

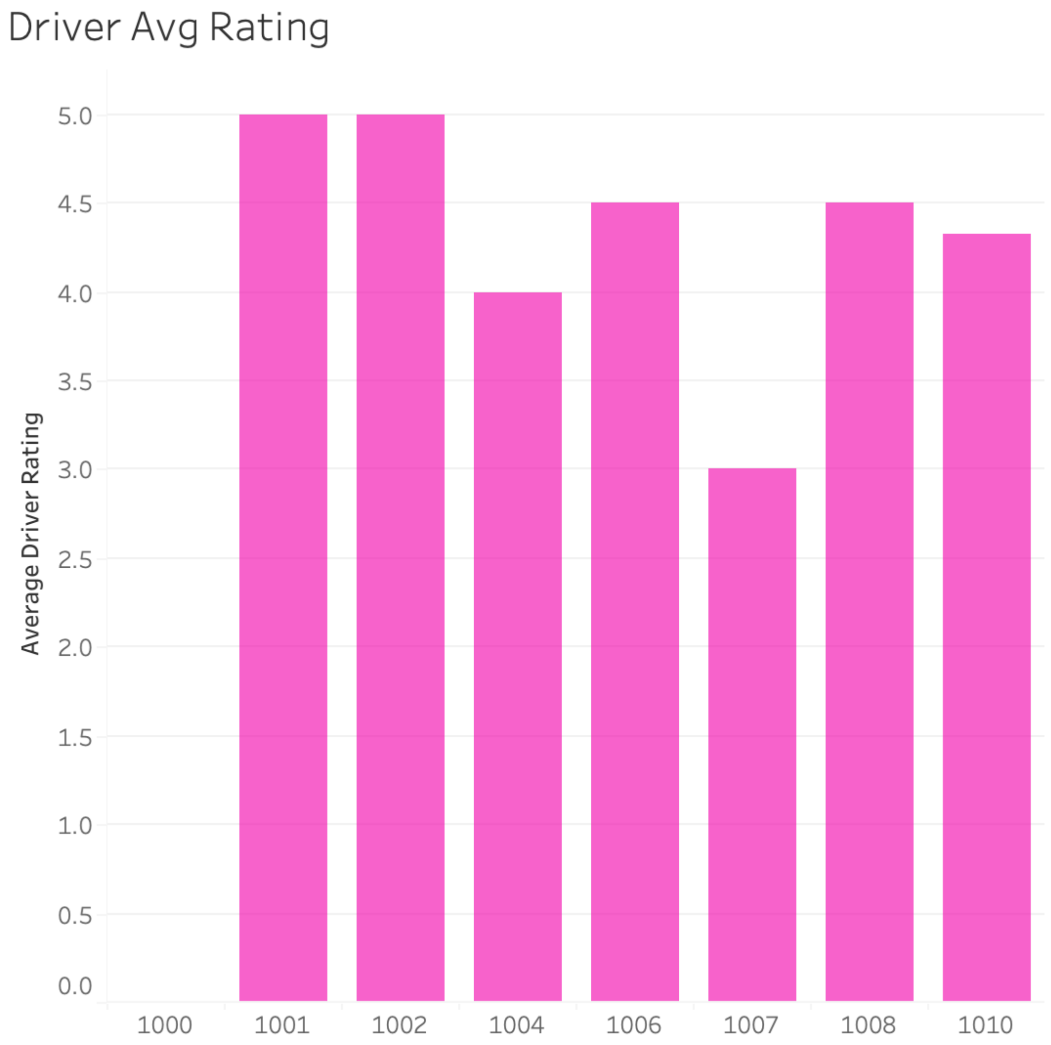


Rides per driver



Visualization





Lessons learned and conclusion

Thank You