DATABASE MANAGEMENT SYSTEM PROJECT

Ayush Prusty(2020427)
Hemang dahiya(2020435)
Sidhant(2020475)
Vasu Khanna(2020483)

STAKEHOLDERS:

1. Customers
2. Drivers
3. Booking team
4. Customercare team

SCOPE OF THE PROJECT:

This database can be used to create a company that would help the customers get easy transportation facilities from anywhere. This also helps in generating employment for more and more people who know driving and own a car.

RELATIONS

1. Customer reviews customer care
2. Customer Gives Money
3. Driver Receives Money
4. Driver Owned Car
5. Customer Booked Car

RELATIONSHIPS WITH ATTRIBUTES:

Transaction(Booking ID, Modes)
Booked (Price, Booking ID, PickUP location, DropOff location, Driver ID, Estimated Drop time, PickUP time, Status)
Reviews(Booking ID, Review)

RELATIONAL SCHEMA:

Customer Care (State, Helpline 1, Helpline2)
Driver (DriverID, DriverName, CarID, Phone Number, Email, Driver Rating)
Car (CarID, Type, Model, Capacity, DriverID)
Customer (Customer ID, Customer Rating, Customer Phone Number, Customer email, Phone Number, First Name, Last Name)

Weak Entity:

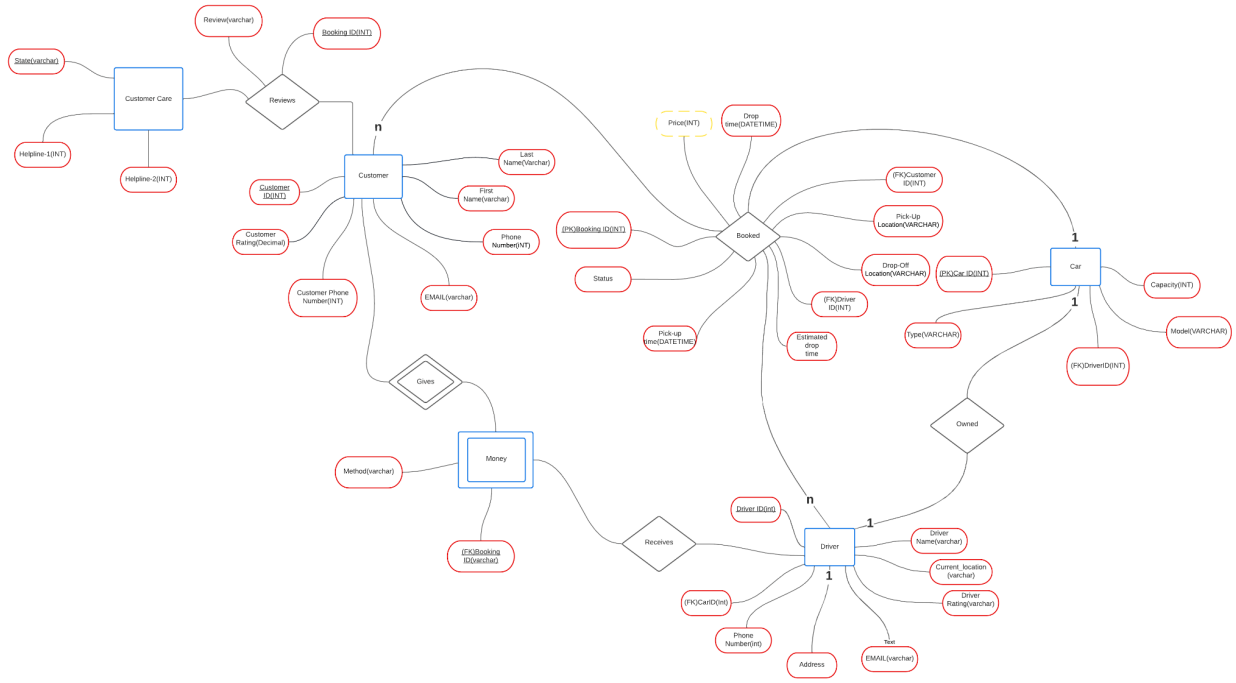
The Money Table can be identified as a weak entity as we know weak entity cannot be uniquely identified by its attributes. In the Money Table which consists of the Transaction method and its details where we cannot identify it alone, we use booking_id(FK).

Ternary Relationship:
We have a ternary relation name Booking between Driver, Customer and Car as Customer books the Car of the drive to ride to reach the location. Where customers can see the details of the driver and it's car accordingly. Where booking consist of attributes (PK)booking_id, (FK)Driver_id, (FK)customer_id and all the details regarding the ride.

Sufficient and Valid constraints in DDL:

1. Phone numbers should not be of character type.
2. Emails should end with @(Gmail/Yahoo/etc)l.com.
3. Pick-up Time and Drop-off Time should be in DATETIME Format.
4. State names should be valid and real.
5. Phone numbers should be of 10 digits.
6. In Booking status should be of type Dropped off, Pick-Up Done, Yet to be Picked, Dropped, Cancelled.

ER Diagram:

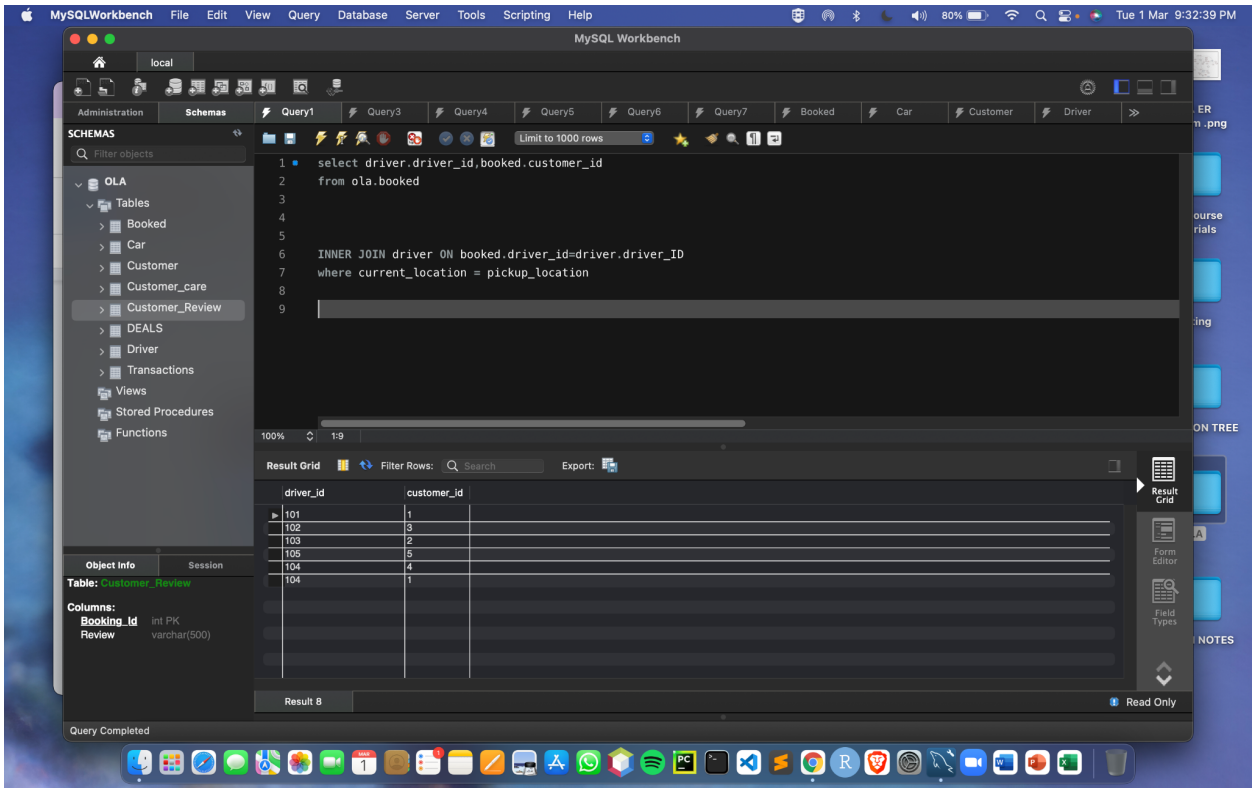

Review(varchar)
Booking ID(INT)
State(varchar)
Customer Care
Reviews
Helpline-1(INT)
Helpline-2(INT)
n
Customer
Last Name(Varchar)
Customer ID(INT)
First Name(varchar)
Customer Rating(Decimal)
Phone Number(INT)
Customer Phone Number(INT)
EMAIL(varchar)
Price(INT)
Drop time(DATETIME)
(FK)Customer ID(INT)
Pick-Up Location(VARCHAR)
[PK]Booking ID(INT)
Booked
Drop-Off Location(VARCHAR)
[PK]Car ID(INT)
Car
1
Status
(FK)Driver ID(INT)
Capacity(INT)
Pick-up time(DATETIME)
Estimated drop time
Type(VARCHAR)
Model(VARCHAR)
1
(FK)DriverID(INT)
Gives
Owned
Money
Method(varchar)
Driver ID(int)
n
1
Receives
Driver
Driver Name(varchar)
(FK)Booking ID(varchar)
Current_location (varchar)
1
(FK)CarID(int)
Driver Rating(varchar)
Phone Number(int)
Address
Text
EMAIL(varchar)

Query 1

Select the driver whose current location matches the customer's pick-up location and show his id. (and then modify the booked table) [store driver current location]

Ans.

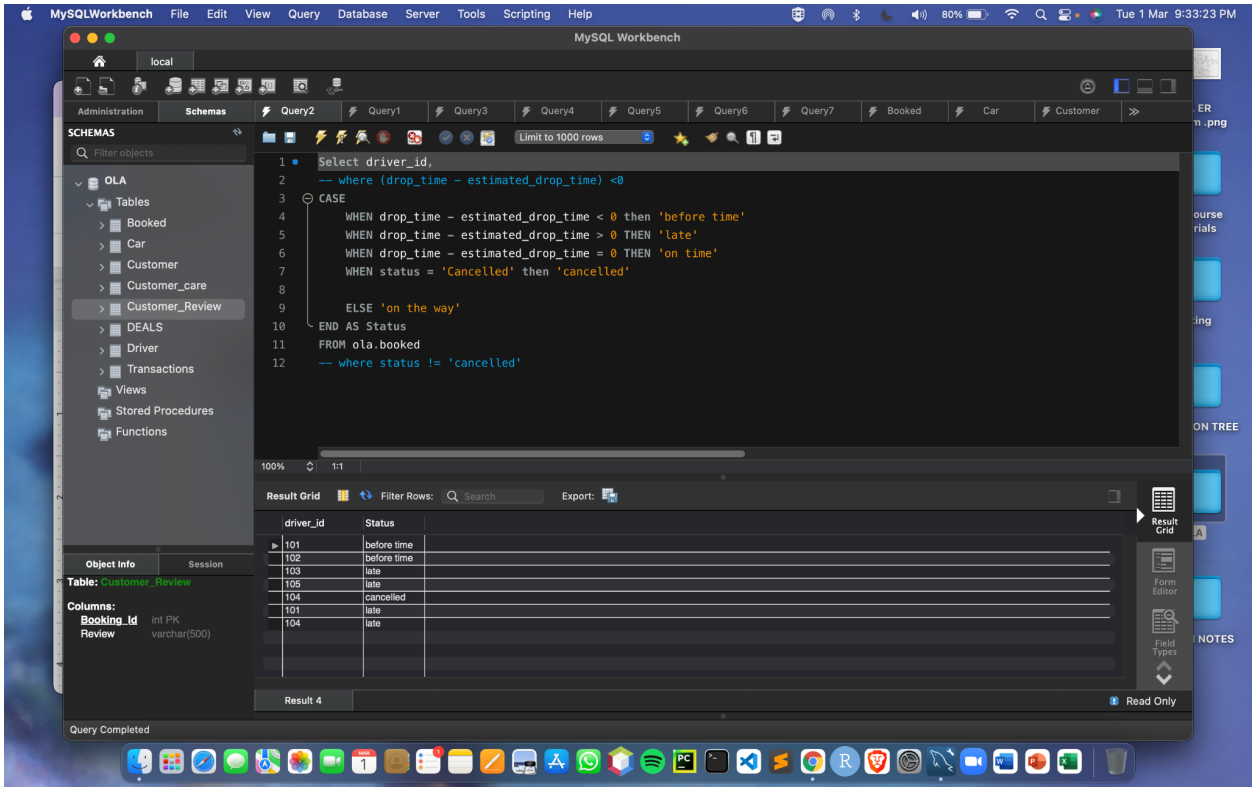
```
select driver.driver_id,booked.customer_id
from ola.booked
INNER JOIN driver ON booked.driver_id=driver.driver_ID
where current_location = pickup_location
```



Query 2
Show the status of each driver whether he dropped the customer on/before time. [Compare the drop time and the estimated drop time]
Ans.

```
Select driver_id,
-- where (drop_time - estimated_drop_time) <0
CASE
        WHEN drop_time - estimated_drop_time < 0 then 'before time'
   WHEN drop_time - estimated_drop_time > 0 THEN 'late'
   WHEN drop_time - estimated_drop_time = 0 THEN 'on time'
   WHEN status = 'Cancelled' then 'cancelled'

   ELSE 'on the way.'
END AS Status
FROM ola.booked
-- where status != 'cancelled'
```

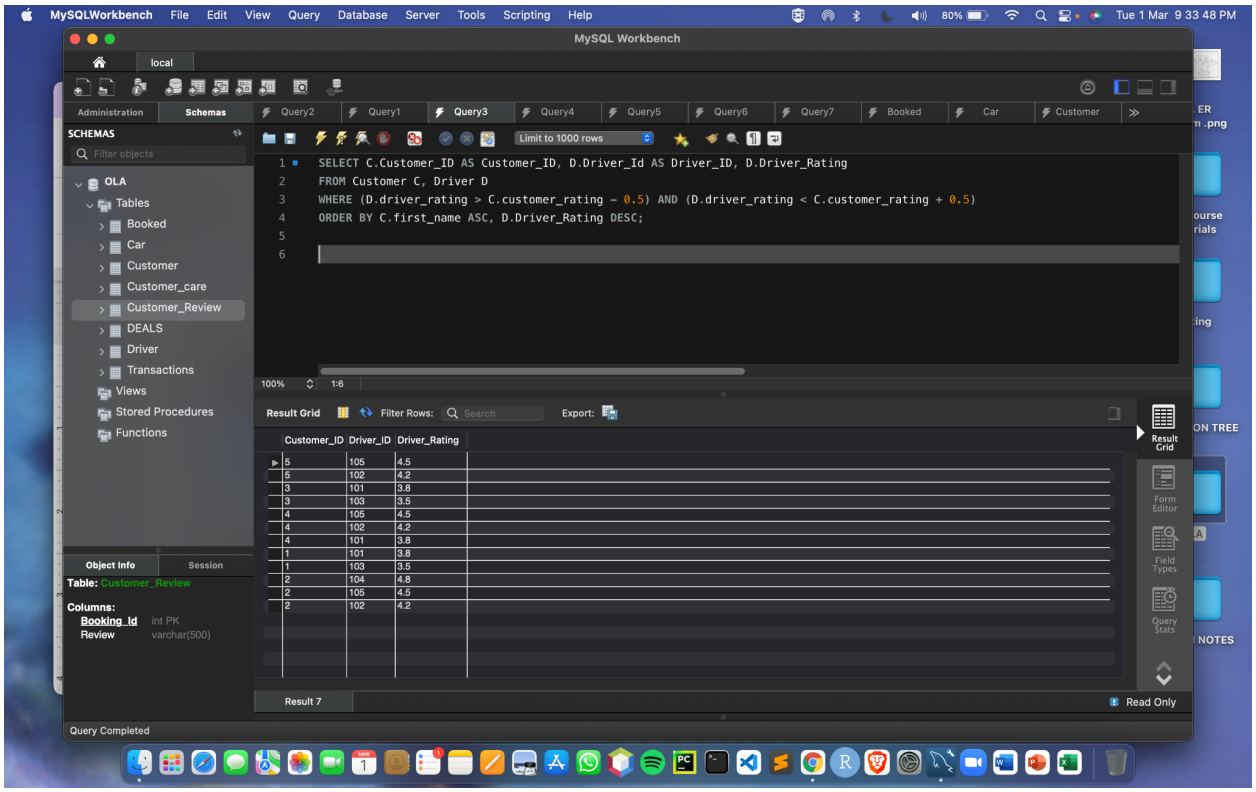


Query 3

Select the Compatible Drivers for the Customers according to their Ratings(Need to Add the Rating Column in customer)

Ans.

```
SELECT C.Customer_ID AS Customer_ID, D.Driver_Id AS Driver_ID, D.Driver_Rating
FROM Customer C, Driver D
WHERE (D.driver_rating > C.customer_rating - 0.5) AND (D.driver_rating < C.customer_rating + 0.5)
ORDER BY C.first_name ASC, D.Driver_Rating DESC;
```

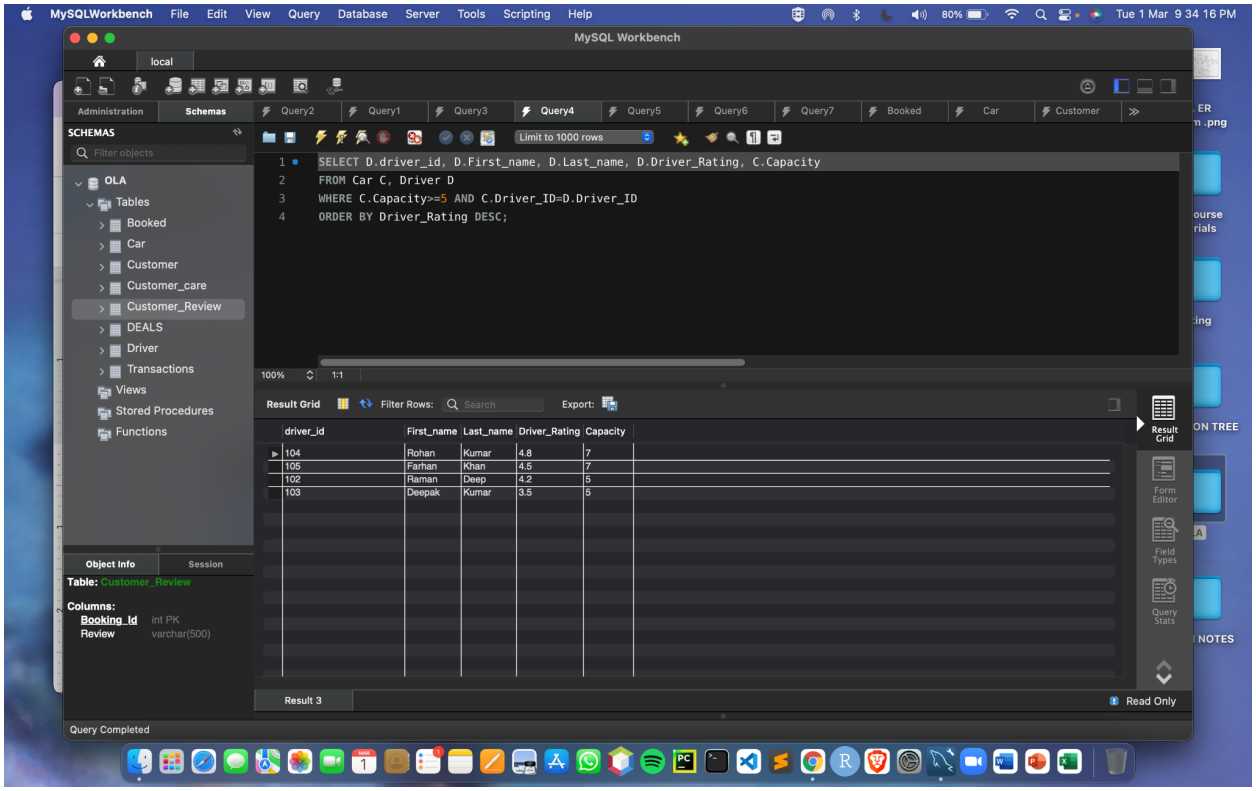


Query 4

Tell the names of the drivers who have the Specific car type or Capacity.

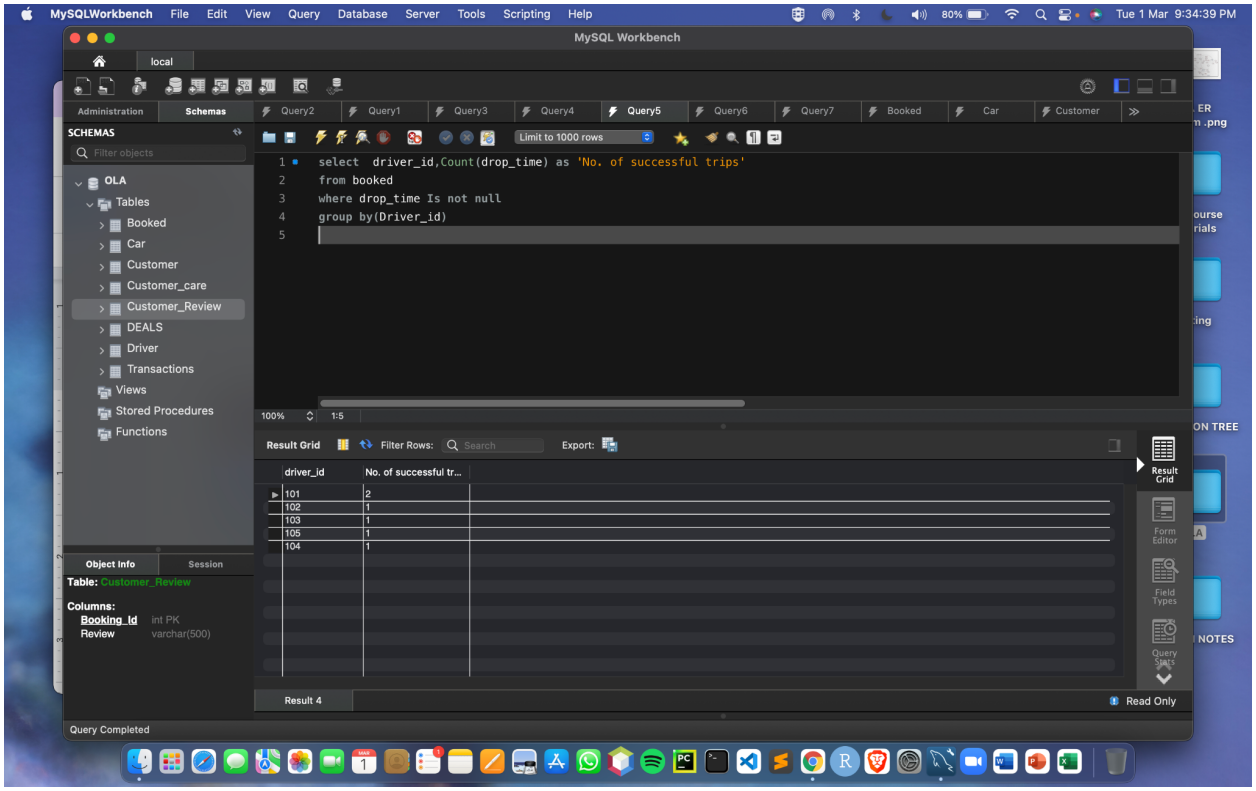
Ans.

```
SELECT D.driver_id, D.First_name, D.Last_name, D.Driver_Rating, C.Capacity, C.Model
FROM Car C, Drivers D
WHERE C.Capacity>=5 AND C.Driver_ID=D.Driver_ID
ORDER BY Driver_Rating DESC;
```



Query 5
Select the no. of successful trips of each driver
[Increase data in booking id, so that same driver has more than one trip]
Ans.

```
select  driver_id,Count(drop_time) as 'No. of successful trips'
from booked
where drop_time Is not null
group by(Driver_id)
```

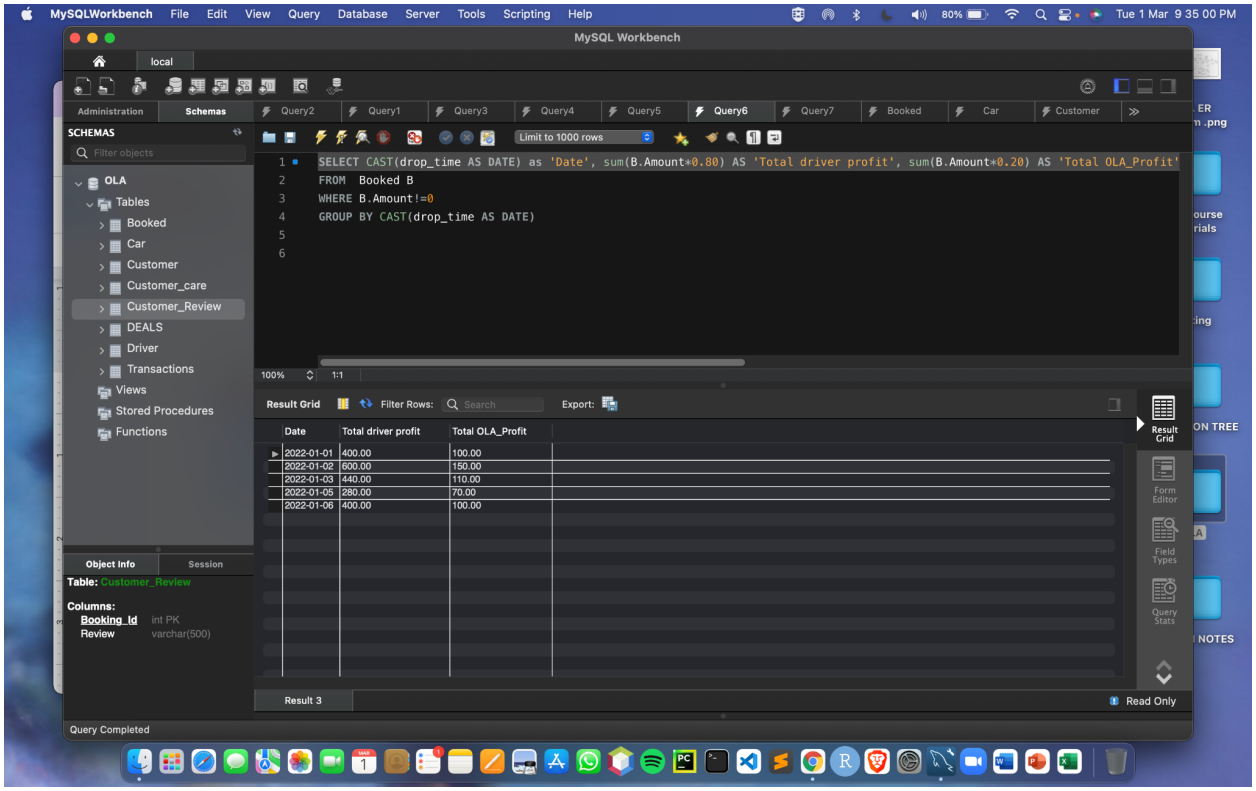


Query 6

Give the total profit for each date.

Ans.

```
SELECT CAST(drop_time AS DATE) as 'Date', sum(B.Amount*0.80) AS 'Total driver profit',
sum(B.Amount*0.20) AS 'Total OLA_Profit'
FROM  Booked B
WHERE B.Amount!=0
GROUP BY CAST(drop_time AS DATE)
```

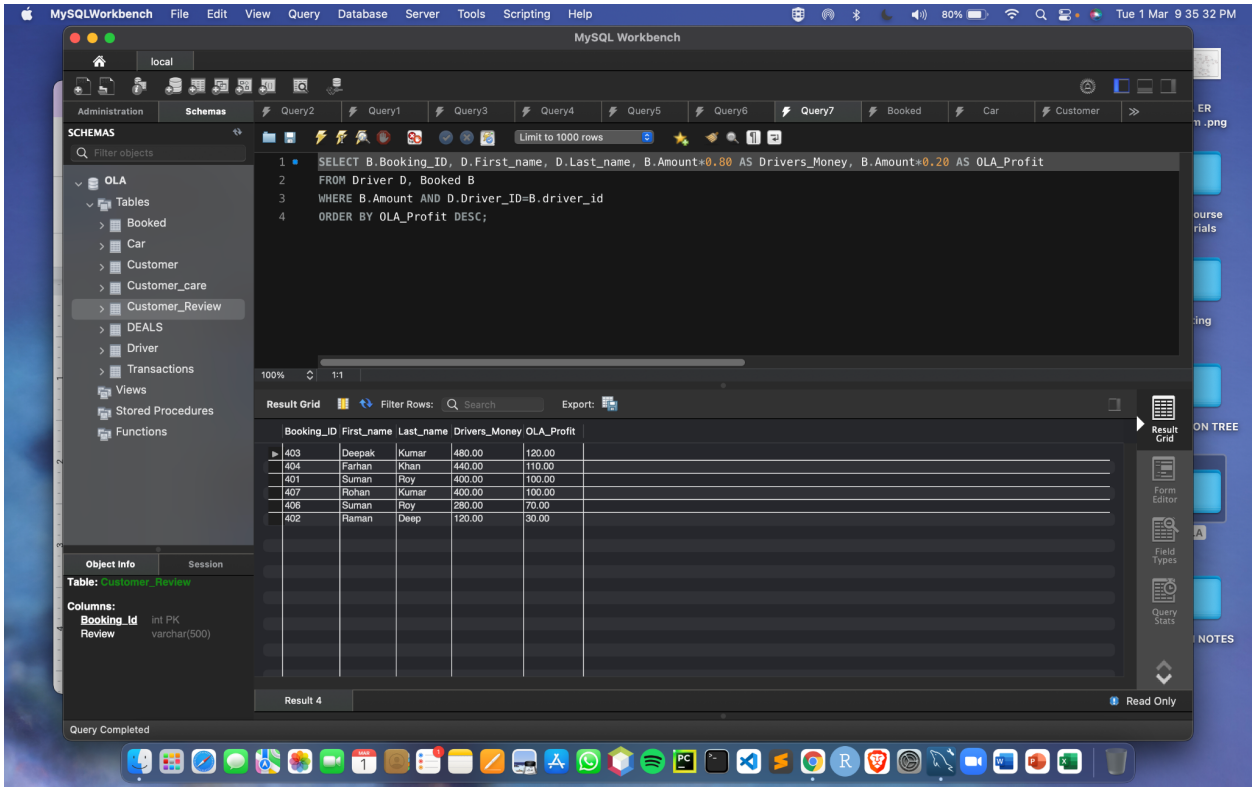


Query 7

Amount of Money Driver Received Giving 20% cut to the Online Cab Service.

Ans.

```
SELECT B.Booking_ID, D.First_name, D.Last_name, B.Amount*0.80 AS Drivers_Money,
B.Amount*0.20 AS OLA_Profit
FROM Drivers D, Booked B
WHERE B.Amount AND D.Driver_ID=B.driver_id
ORDER BY OLA_Profit DESC;
```

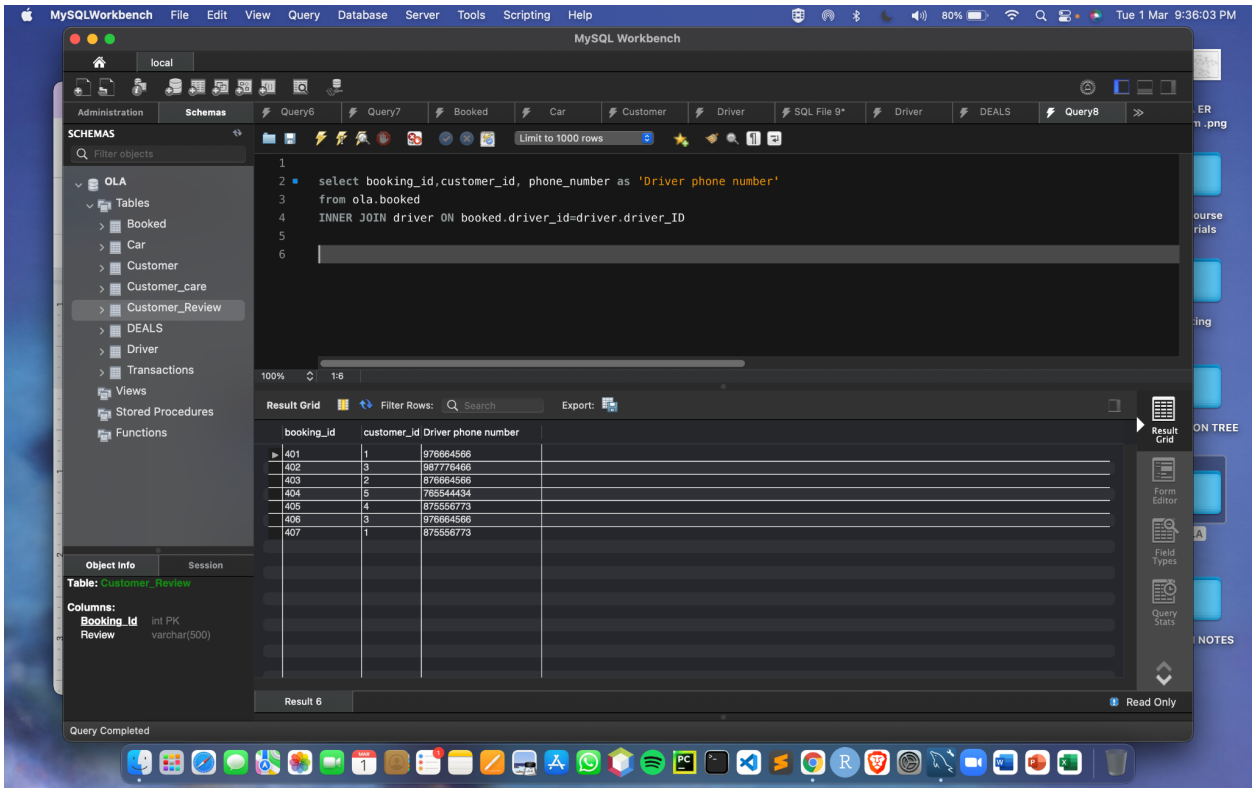


Query 8

Show phone numbers of the drivers which the customer has to dial for his booking

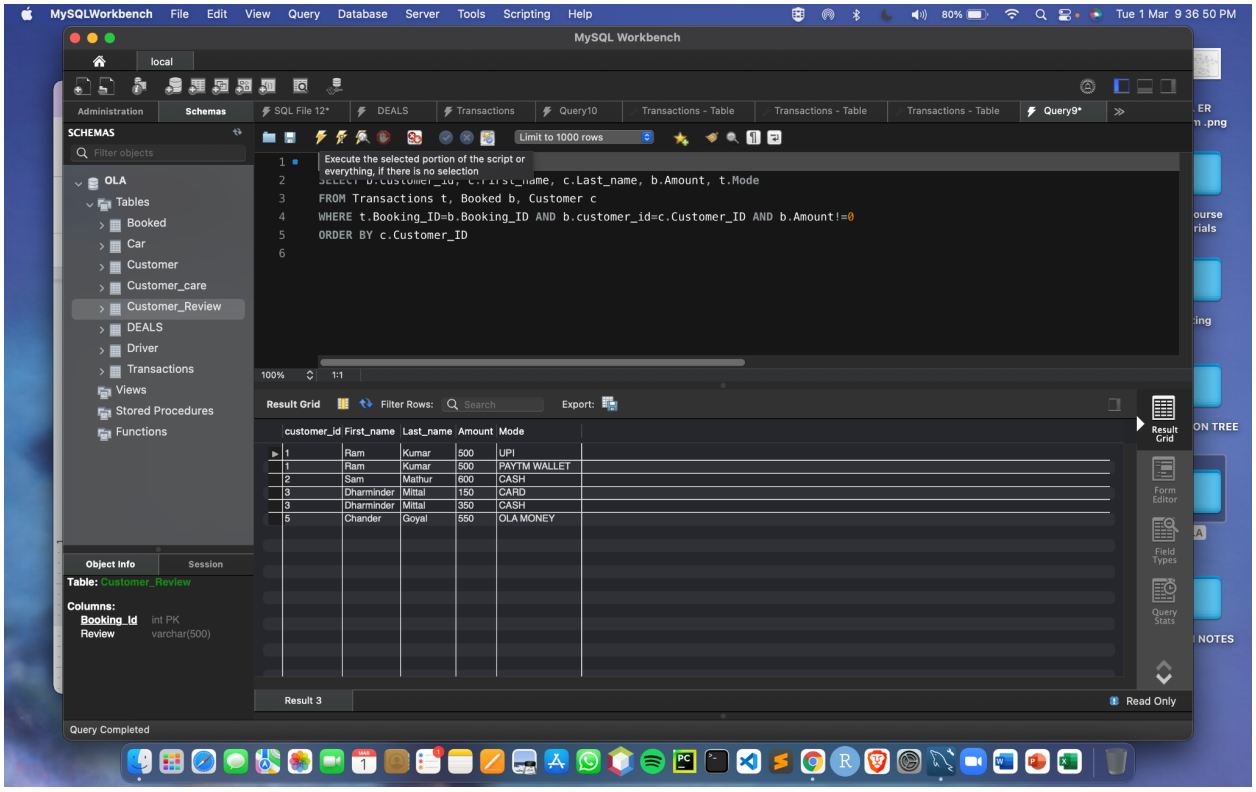
Ans.

```
select booking_id,customer_id, phone_number as 'Driver phone number'
from ola.booked
INNER JOIN driver ON booked.driver_id=driver.driver_ID
```



Query 9
Show which customer has used which mode of transaction to pay their trip
Ans.

```
CREATE view Name_and_method AS
SELECT b.customer_id, c.First_name, c.Last_name, b.Amount, t.Mode
FROM Transaction t, Booked b, Customer c
WHERE t.Booking_ID=b.Booking_ID AND b.customer_id=c.Customer_ID AND b.Amount!=0
ORDER BY c.Customer_ID
```



Query 10
Show the money received by each mode of payment
Ans.

```
create view mode_amount as
select mode,sum(amount)
from booked,transactions
where booked.booking_id= transactions.booking_id and mode is not null
group by(mode)
```