

DATABASE MANAGEMENT SYSTEM PROJECT

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

STAKEHOLDERS:

1. Customers
2. Drivers
3. Booking team
4. Customer care 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)

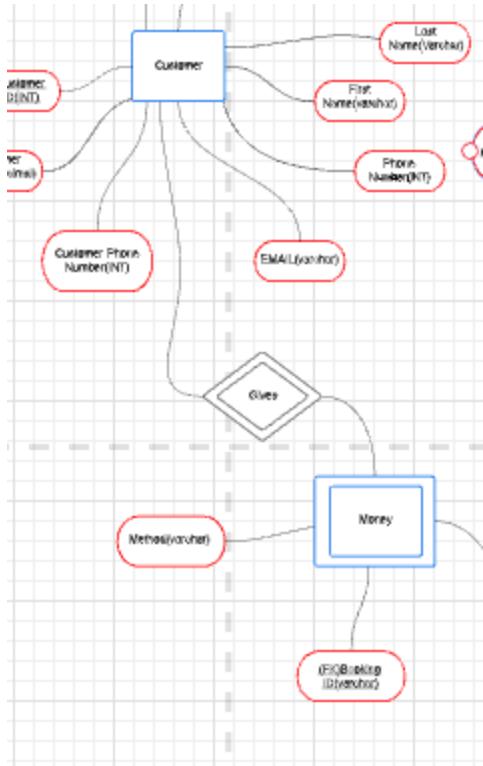
Driver (DriverID, DriverName, CarID, Phone Number, Email, Driver Rating)

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).



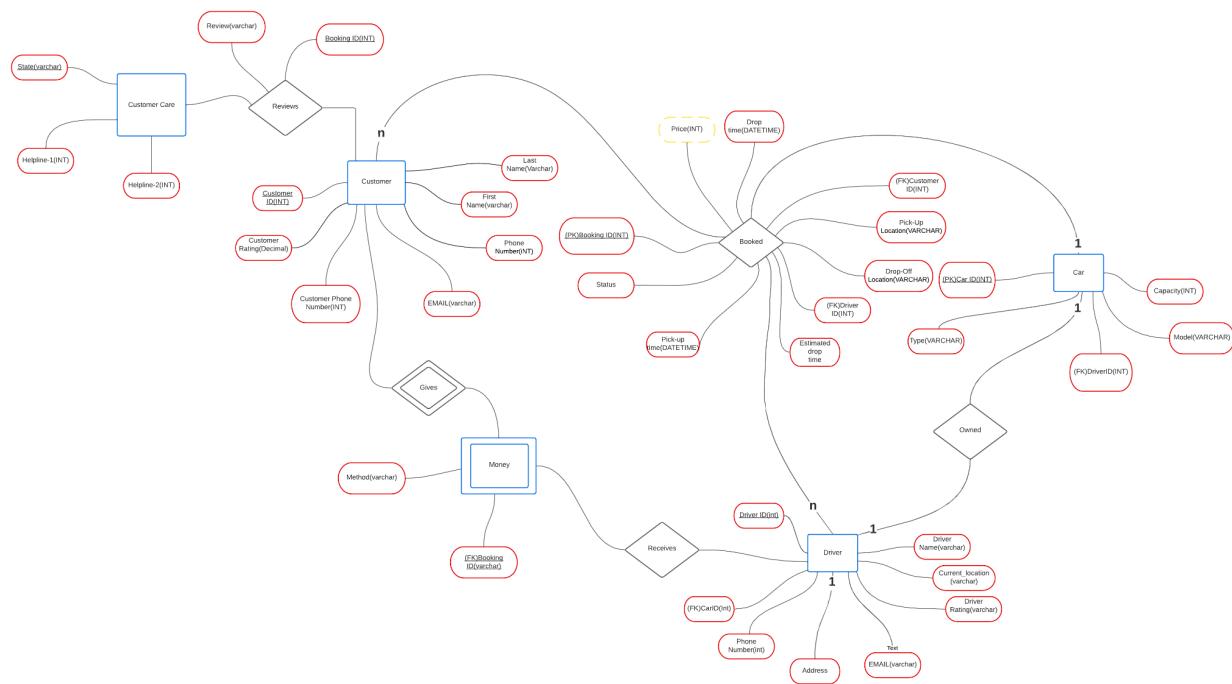
Ternary Relationship:

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).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:



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
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The main window displays a SQL editor with the following query:

```
1 • select driver.driver_id,booked.customer_id  
2 from ola.booked  
3  
4  
5  
6 INNER JOIN driver ON booked.driver_id=driver.driver_ID  
7 where current_location = pickup_location  
8  
9
```

The results grid shows the following data:

driver_id	customer_id
101	1
102	3
103	2
105	5
104	4
104	1

The status bar at the bottom indicates "Result 8" and "Read Only".

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'

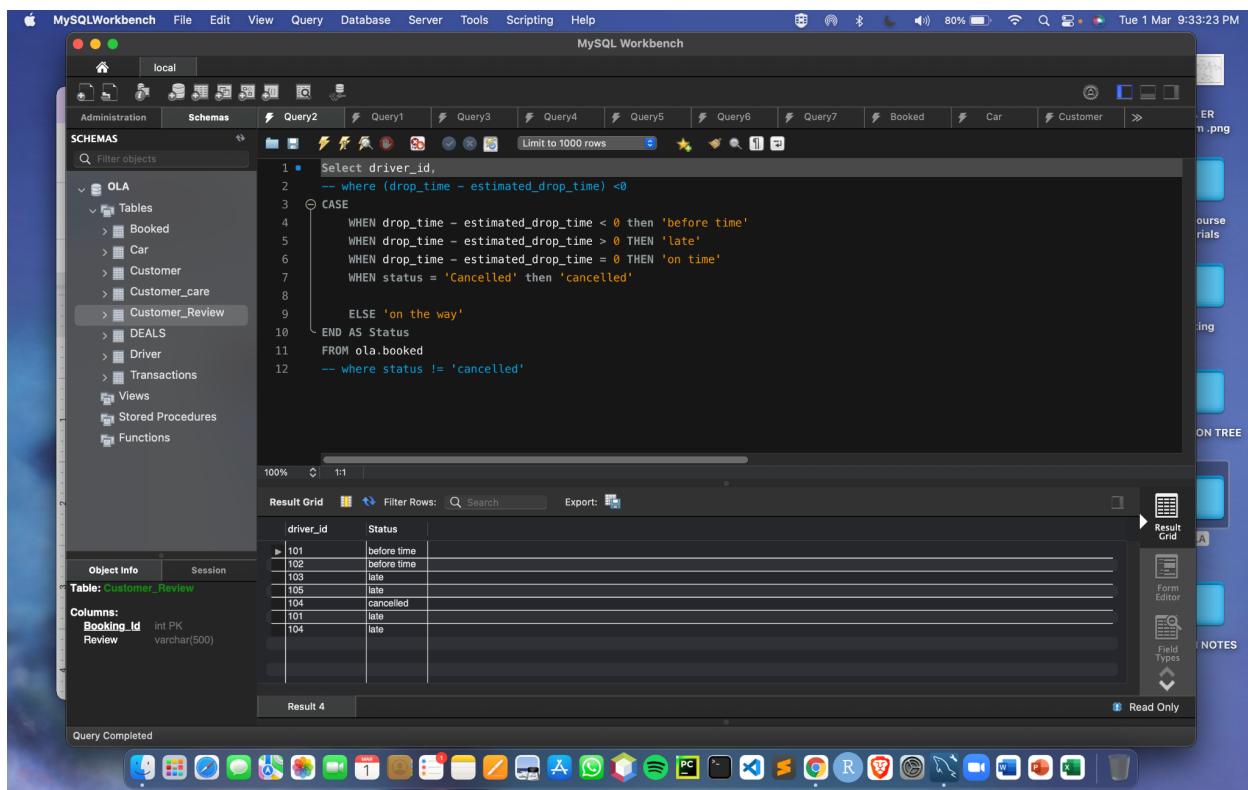
WHEN status = 'Cancelled' then 'cancelled'

ELSE 'on the way.'

END AS Status

FROM ola.booked

-- where status != 'cancelled'



The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```
1 • Select driver_id,
2   -- where (drop_time - estimated_drop_time) <0
3   CASE
4     WHEN drop_time - estimated_drop_time < 0 then 'before time'
5     WHEN drop_time - estimated_drop_time > 0 THEN 'late'
6     WHEN drop_time - estimated_drop_time = 0 THEN 'on time'
7     WHEN status = 'Cancelled' then 'cancelled'
8
9     ELSE 'on the way'
10   END AS Status
11   FROM ola.booked
12   -- where status != 'cancelled'
```

The result grid shows the following data:

driver_id	Status
101	before time
102	before time
103	late
105	late
104	cancelled
101	late
104	late

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;
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The query editor window contains the SQL code for Query 3. The results grid displays a table with three columns: Customer_ID, Driver_ID, and Driver_Rating. The data consists of 10 rows, each containing a unique combination of Customer_ID and Driver_ID, along with their corresponding Driver_Rating values.

Customer_ID	Driver_ID	Driver_Rating
5	105	4.5
5	102	4.2
3	101	3.8
3	103	3.5
4	105	4.5
4	102	4.2
4	101	3.8
1	101	3.8
1	103	3.5
2	104	4.8
2	105	4.5
2	102	4.2

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;
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The main window displays a query editor with the following SQL code:

```
1 • SELECT D.driver_id, D.First_name, D.Last_name, D.Driver_Rating, C.Capacity  
2   FROM Car C, Driver D  
3   WHERE C.Capacity>=5 AND C.Driver_ID=D.Driver_ID  
4   ORDER BY Driver_Rating DESC;
```

The results grid shows the following data:

driver_id	First_name	Last_name	Driver_Rating	Capacity
104	Rohan	Kumar	4.8	7
105	Farhan	Khan	4.5	7
102	Raman	Deep	4.2	5
103	Deepak	Kumar	3.5	5

The status bar at the bottom indicates "Query Completed".

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)
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The query editor window displays the SQL code for Query 5. The results grid shows the output of the query, which is a table with two columns: 'driver_id' and 'No. of successful tr...'. The data in the table is:

driver_id	No. of successful tr...
101	2
102	1
103	1
105	1
104	1

The status bar at the bottom indicates "Query Completed".

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)
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The query editor window contains the SQL code for Query 6. The results grid displays the output, which is a table with three columns: Date, Total driver profit, and Total OLA_Profit. The data shows profits for five different dates in January 2022.

Date	Total driver profit	Total OLA_Profit
2022-01-01	400.00	100.00
2022-01-02	600.00	150.00
2022-01-03	440.00	110.00
2022-01-05	280.00	70.00
2022-01-06	400.00	100.00

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;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema with the 'OLA' database selected, specifically the 'Customer_Review' table. The main pane contains the SQL query:

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

The results are shown in a 'Result Grid' table:

Booking_ID	First_name	Last_name	Drivers_Money	OLA_Profit
403	Despak	Kumar	480.00	120.00
404	Farhan	Khan	440.00	110.00
401	Suman	Roy	400.00	100.00
407	Rohan	Kumar	400.00	100.00
406	Suman	Roy	280.00	70.00
402	Raman	Deep	120.00	30.00

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
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The query editor window displays the following SQL code:

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

The results grid shows the output of the query:

booking_id	customer_id	Driver phone number
401	1	97664566
402	3	98777466
403	2	87664566
404	5	78554434
405	4	87556773
406	3	97664566
407	1	87556773

The status bar at the bottom indicates "Query Completed".

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
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQL Workbench" and the status bar shows the date and time as "Tue 1 Mar 9 36 50 PM". The main window displays a SQL editor with the following query:

```
1 Execute the selected portion of the script or  
2 entire file if there is no selection  
3 SELECT b.customer_id, c.First_name, c.Last_name, b.Amount, t.Mode  
4 FROM Transaction t, Booked b, Customer c  
5 WHERE t.Booking_ID=b.Booking_ID AND b.customer_id=c.Customer_ID AND b.Amount!=0  
6 ORDER BY c.Customer_ID
```

Below the editor is a "Result Grid" pane showing the output of the query:

	customer_id	First_name	Last_name	Amount	Mode
1	1	Ram	Kumar	500	UPI
1	1	Ram	Kumar	500	PAYTM WALLET
2	2	Sem	Methur	600	CASH
3	3	Dharminder	Mittal	150	CARD
3	3	Dharminder	Mittal	350	CASH
5	5	Chander	Goyal	550	OLA MONEY

The status bar at the bottom indicates "Result 3" and "Read Only". The MySQL Workbench interface includes a sidebar with "Administration", "Schemas", and "Tables" sections, and a central workspace with various toolbars and panes.

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)
```

The screenshot shows the MySQL Workbench interface on a Mac OS X desktop. The title bar reads "MySQL Workbench". The main window has several tabs at the top: Administration, Schemas, SQL File 12*, DEALS, Transactions, Query10*, Transactions - Table, Transactions - Table, Transactions - Table, and Query9*. The "SQL File 12*" tab is active, displaying the query code provided above. Below the tabs is a toolbar with various icons. The left sidebar shows the database schema with tables like OLA, Books, Car, Customer, Customer_care, Customer_Review, DEALS, Driver, Transactions, Views, and Stored Procedures. The "Customer_Review" table is currently selected. The main pane displays the results of the query in a grid format:

mode	sum(amount)
UPI	500
CARD	150
CASH	950
CLA MONEY	550
PAYTM WALLET	500

At the bottom of the results pane, it says "Result 14" and "Read Only". The status bar at the bottom of the screen shows the date and time as "Tue 1 Mar 9 37 12 PM".