Use Case Study Report

Group no: 11

Student Names: Sahil Sushil Mahajan and Aditya Gorakh Velapurkar

Executive Summary:

In this study, we have created a rental car system. We have chosen Rental Car System because this business is in huge demand and many car rental businesses are struggling with inventory management issues as well as customer retention and also are unable to increase their total revenue because of huge competition. As per the latest studies, the car rental business is projected to grow by 4.6% CAGR (Compounded Annual Growth Return). Our goal is to address the problems the car rental businesses are facing therefore, we have performed analytical queries on the rental car databases using MySQL and MongoDB to gather meaningful insights which will eventually help the business in better decision-making.

The database has been modeled taking into consideration various present car rental websites which are market leaders (eg. Zipcar, Boston Car Rental). The entities and the relationships are designed such as the data looks like real-life data of current business. The data used in this system is demo data. The project started with the construction of the ERDDs and the UML diagram, followed by the relational model. The relational model has been implemented in SQL data using MySQL workbench, and the same model has also been implemented in NoSQL using MongoDB. The database has been connected to Python, and SQL queries are implemented in python to perform visualizations of analytical queries.

The achievements of this project are that by performing analytics we came up with different ideas to increase the overall revenue of the business and also increase customer retention by offering loyalty programs to the customers and discounts to the customers who are the highest spenders.

I.Introduction:

One of the biggest challenges faced by rental car businesses currently is inventory management and how to increase customer retention. The rental car businesses were struggling to survive during the Covid-19 pandemic but now as the revenge tourism wave is in full swing and during the holiday season there is a huge demand for rental cars, the businesses are unable to provide satisfactory services or make the most out of the opportunities, therefore we have chosen Rental car system as the topic of our study. We aim to draw meaningful insights from the data by performing different analytical queries on the car rental database so that this analytical data can be used in better decision-making by the management of the businesses. The goal of our study is to apply the concepts of MySQL and MongoDB we learned in our Database Management for analytics class on the Rental Car System database and help the Rental car business improve their inventory management and customer retention and eventually help grow their revenue over a period of time.

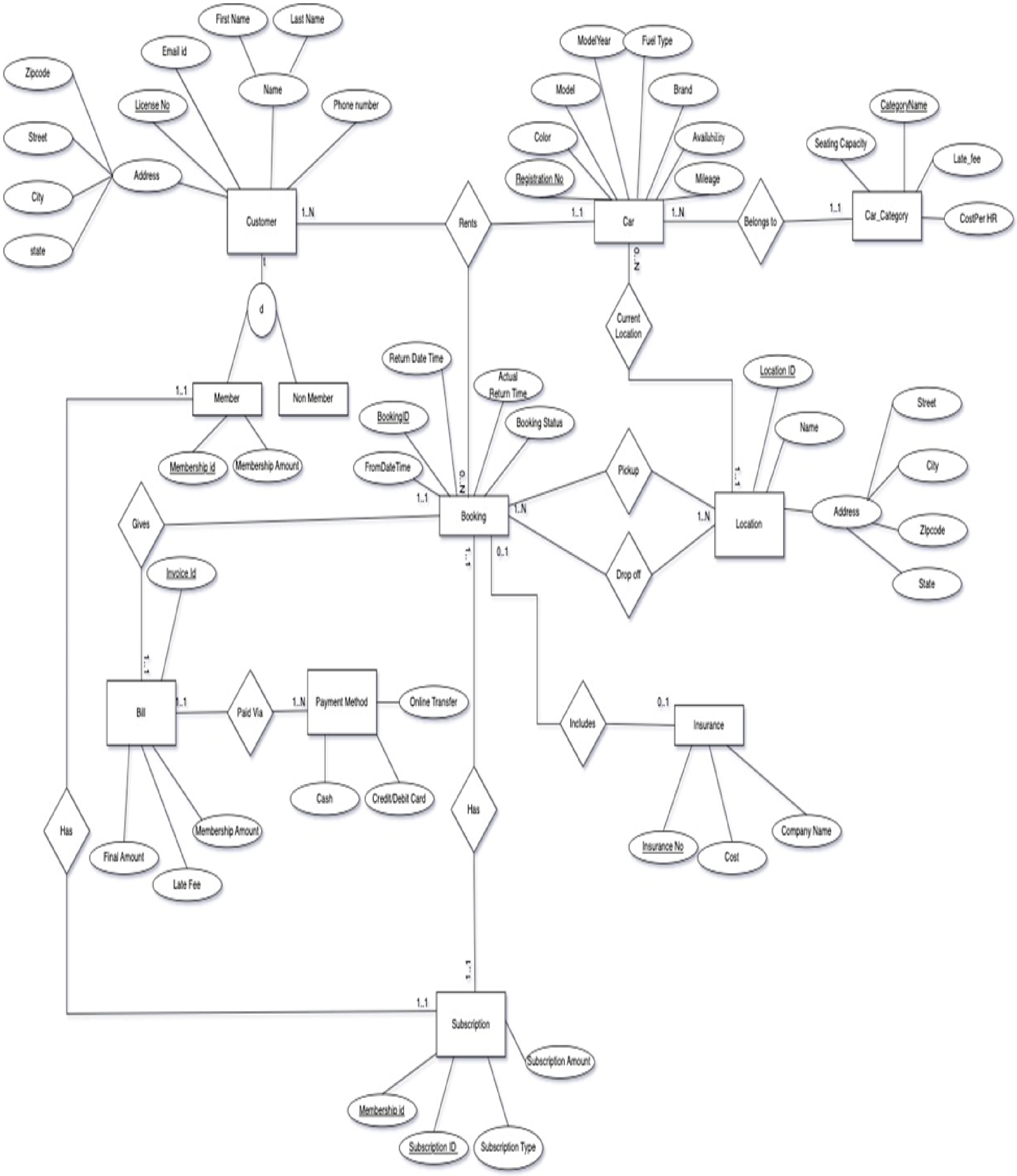
The requirement for this use case study project is the software required to create the database MySQL workbench and Studio 3T (for MongoDB), to perform Visualizations we connected the MYSQL server with Python in Jupyter Notebook.

In our system, Customers can rent a car based on different brands and models. A car may belong to one (exactly one) vehicle category (compact, economy, convertible, etc.). Each category is described by a unique Category Name label (e.g. Sedan, SUV, Hatchback, etc.). Our system allows customers to have different pick-up and drop-off locations and will impose a late fee if the rental car is returned beyond the return date and time. The Booking details will also have insurance details which will be chosen by the customer. The Customer can be a member or non-member and a member must have a subscription, we have 4 types of subscriptions, (daily, weekly, weekend, and monthly).

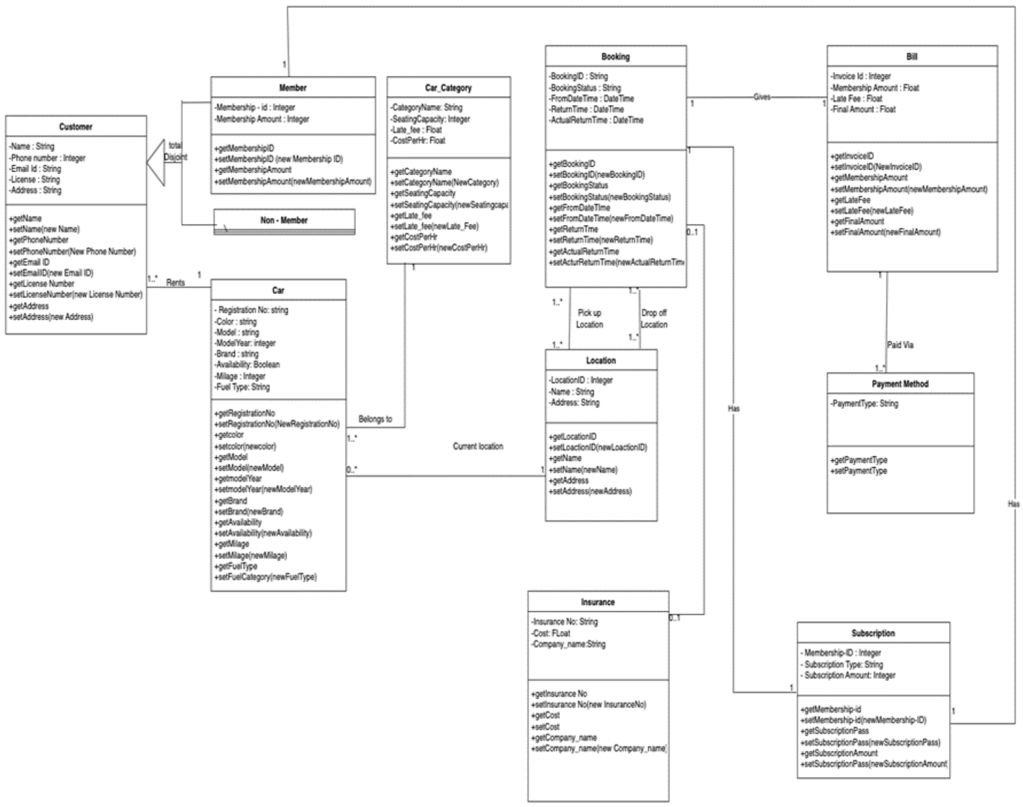
The Rental Car Business is in huge demand nowadays and businesses are looking for a way to improve their efficiency our Database system analytics will be very helpful for businesses to implement the changes in their business and incorporate the required changes by using the analyzed data. We have created the data taking into consideration the real businesses (eg Zipcar, Boston Car rental).

II. Conceptual Data Modeling

A. EER Model



B. UML Diagram



III. Mapping Conceptual Model to Relational Model:

Customer(License No, Phone Number, Email id, Street, City, State, Zipcode)

License No -> Primary Key not null

Member(MembershipID,MembershipAmount)

MembershipID-> Primary key not null.

Car (Registration No, Mileage, Model, Color, Availability, FuelType,ModelYear,CategoryName*,LocationID*)

Registration No -> Primary Key not null

CategoryName-> Foreign Key refers CarCategory

LocationID-> Foreign Key refers Location

Car-Model(Registration No, Model, Model Year)

Registration no -> Primary key Not Null

Customer-Car(License No, Registration No)

License No -> Primary Key not null

Registration No -> Primary Key not null

CarCategory (Category Name, Seating Capacity, Late fee, CostPer Hr)

Category Name -> Primary Key Not Null

Car-CarCategory(Registration No, CategoryName*)

Registration no -> Primary key Not Null

Category Name -> Foreign key refers from the table Car Category

Payment Method (Credit Card, Debit Card, Cash)

Booking (Booking ID, Booking status, FromDate Time, ReturnDate, Amount, Actual Return Time,PickupLocation*,DropoffLocation*,LocationID*,InsuranceNo*)

Booking ID -> Primary Key Not Null

PickupLocation*,DropoffLocation*,LocationID*-> Foreign key refers Location

InsuranceNo-> Foreign key refers Insurance

Insurance(Insurance No, Cost, Company Name)

Insurance No -> Primary Key Not Null

Location(Location ID, Name, Street, City, Zipcode, State)

Location ID -> Primary Key Not Null

Customer-Booking(License No , Booking ID*)

License No -> Primary Key not null

Booking ID -> Foreign key refers from the table Booking

Bill(InvoiceID,SubscriptionID*,MembershipAmount*,LateFee*,FinalAmount,BookingID*,Payment_type*)

InvoiceID-> Primary Key

SubscriptionID->Foreign Key refers subscription

MembershipAmount-> Foreign Key refers Member

LateFee-> Foreign Key refers CarCategory

Booking-Bill(Booking ID*, Invoice ID*)

Booking ID -> Foreign key Not Null refers from the table Booking

Insurance No -> Foreign key Not Null refers from the table Insurance

Bill – Payment Model(Invoice ID*, Credit Card, Debit Card, Cash)

Invoice ID -> Foreign key not null refers refers from the table Bill

Booking-Insurance(Booking ID, Insurance No)

Booking ID -> Primary Key Not Null refers from the table Booking

Insurance No -> Primary Key Not Null refers from the table Insurance

Subscription(Subscription ID, Membership ID*, Subscription Type,Subscription Amount)

Subscription ID -> Primary Key Not Null

Membership Id -> Foreign key Not Null refers from the table Members

Relations Normalized till 3.5 NF :

Rental Car System (CustomerLicense No , MembershipID*, InvoiceID*, InsuranceNO*,

Location ID*)

MembershipID -> Foreign key refers to Memebership Entity.(NOT NULL)

InvoiceID -> Foreign key refers to Bill Entity. (NOT NULL)

InsuranceNO*-> Foreign key refers to Insurance Entity. (NOT NULL)

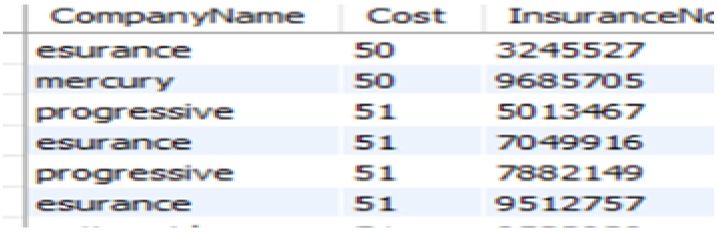
Location ID* -> Foreign key refers to Location Entity. (NOT NULL)

IV. Implementation of Relation Model via MySQL and NoSQL:

#Query 1 (finding Insurance company name and InsuranceNo where cost of insurance is less than 100)

select distinct(CompanyName), Cost,

InsuranceNo from Insurance

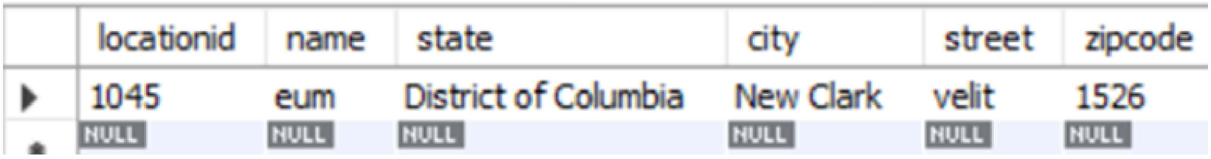


CompanyName	Cost	InsuranceN
esurance	50	3245527
mercury	50	9685705
progressive	51	5013467
esurance	51	7049916
progressive	51	7882149
esurance	51	9512757

where cost <=100 order by Cost;

#Query 2 (find location where city is New Clark and zipcode =1526)

select * from location where city = 'New Clark' and zipcode = 1526;

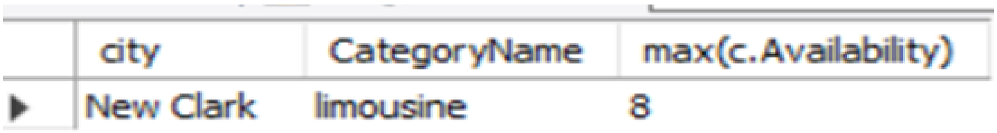


	locationid	name	state	city	street	zipcode
▶	1045	eum	District of Columbia	New Clark	velit	1526
*	NULL	NULL	NULL	NULL	NULL	NULL

#Query 3 (Find city name which has maximum number of limousine)

select distinct (l.city), c.CategoryName, max(c.Availability)

from car c, location l , carcategory cc where c.CategoryName = 'limousine' order by l.city;



	city	CategoryName	max(c.Availability)
▶	New Clark	limousine	8

#Query 4 (Find membershipID and SubscriptionID where subscription type is daily)

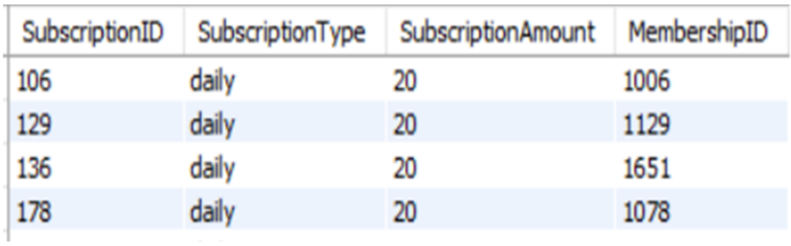
select s.SubscriptionID, s.SubscriptionType,

s.SubscriptionAmount, m.MembershipID

from subscription s, member m

where m.MembershipID=s.MembershipID

and s.SubscriptionType='daily'order by s.SubscriptionID



SubscriptionID	SubscriptionType	SubscriptionAmount	MembershipID
106	daily	20	1006
129	daily	20	1129
136	daily	20	1651
178	daily	20	1078

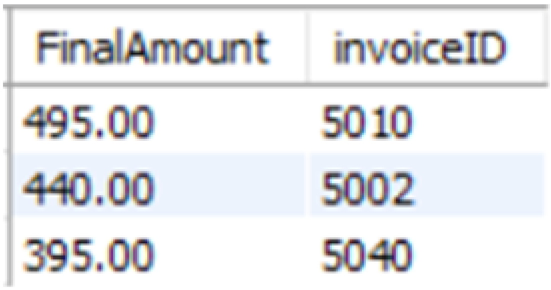
#Query 5 (correlated query to find out top 3 highest amount of invoices and their invoice numbers)

select b1.FinalAmount from bill b1 where 3>

(select count(*) from bill b2

where b2.FinalAmount>b1.FinalAmount)

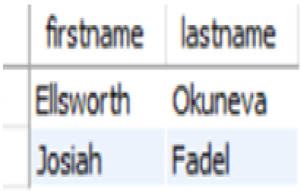
order by FinalAmount DESC;



FinalAmount	invoiceID
495.00	5010
440.00	5002
395.00	5040

#Query 6 (nested query to identify firstname and last name of customers who took subscription of type weekend and monthly.

select firstname,lastname from customer where Licenseno IN

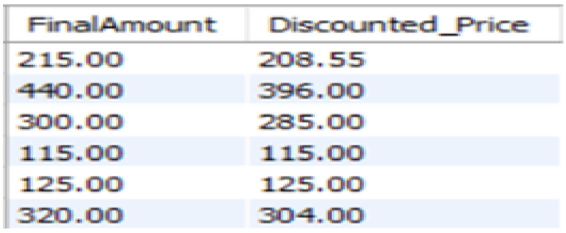


firstname	lastname
Ellsworth	Okuneva
Josiah	Fadel

```
 ( select Licenseno from member where MembershipID IN

 (select MembershipID from subscription where SubscriptionType = 'weekend'))

 AND Licenseno IN( select Licenseno from member where MembershipID IN

 (select MembershipID from subscription where SubscriptionType = 'monthly'));
```

#query 7 (Query to provide discount according to final amount range, returning final discounted price)

```
select b.FinalAmount,

CASE when (b.FinalAmount>=450 and b.FinalAmount<=500) then (b.FinalAmount*0.85)

                when (b.FinalAmount>=400 and b.FinalAmount<=450) then (b.FinalAmount*0.90)

                when (b.FinalAmount>=300 and b.FinalAmount<=400) then (b.FinalAmount*0.95)

       when (b.FinalAmount>=200 and b.FinalAmount<=300) then (b.FinalAmount*0.97)

       else b.FinalAmount

       END AS Discounted_Price

from bill b;
```



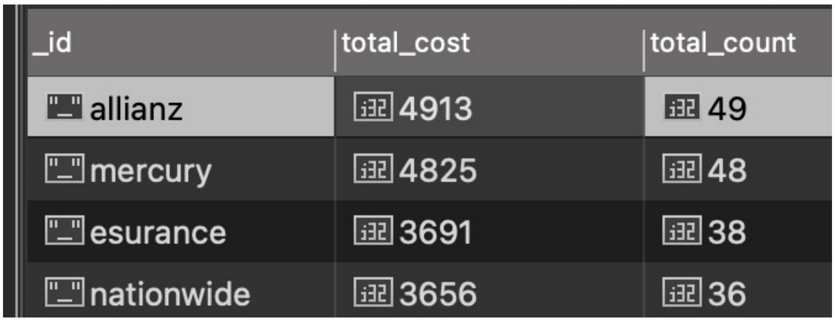
FinalAmount	Discounted_Price
215.00	208.55
440.00	396.00
300.00	285.00
115.00	115.00
125.00	125.00
320.00	304.00

B. Implementation of the Relational Model in NoSQL

The database was constructed in MongoDB and the following queries were done. The results for the Mongo DB queries are available in the appendix

#Query 1:Displaying Each insurance company revenue generatede and no. of customers bought

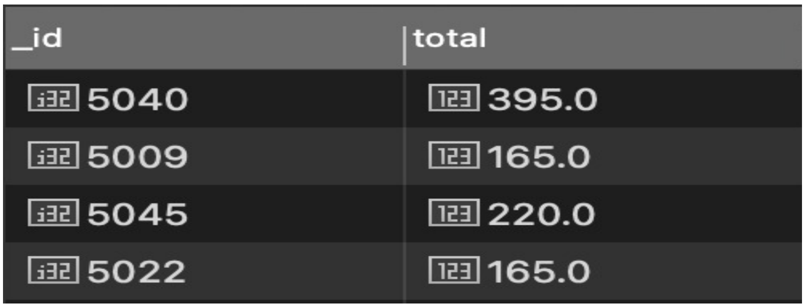
```
db.insurance.aggregate([{$group: {

   _id: "$CompanyName",

   total_cost: {$sum: "$Cost"}, total_count:
{$count: {}}}}  ]);
```



_id	total_cost	total_count
allianz	4913	49
mercury	4825	48
esurance	3691	38
nationwide	3656	36

#Query2:Display the Invoice ID who paid by cash and the final amout paid by each person

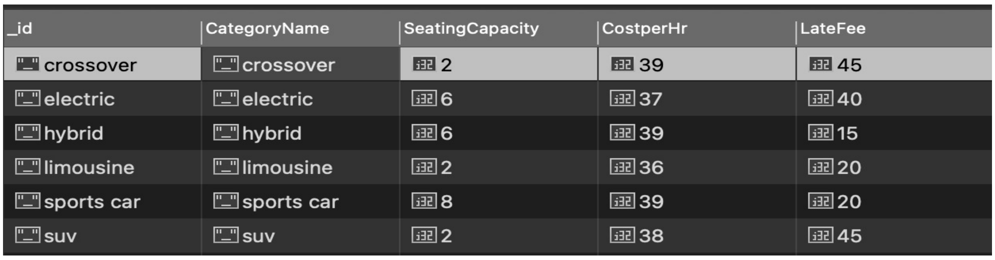
```
db.bill.aggregate([{$match: {payment_type: "cash"}}, {$group: {_id: "$InvoiceID",
total:{$sum:"$FinalAmount"}} }]);
```



_id	total
5040	395.0
5009	165.0
5045	220.0
5022	165.0

#Query 3: find car category name which has seating capacity 2 and their cost per hour is in between 30 and 40

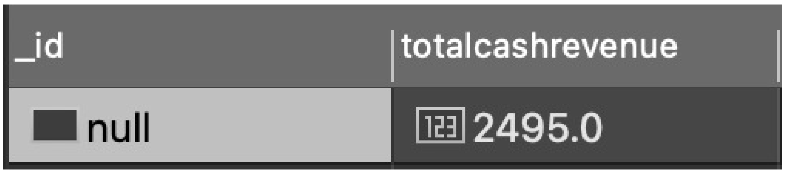
db.carcategory.find({CostperHr: {$gt:30, $lte:40}});



_id	CategoryName	SeatingCapacity	CostperHr	LateFee
crossover	crossover	2	39	45
electric	electric	6	37	40
hybrid	hybrid	6	39	15
limousine	limousine	2	36	20
sports car	sports car	8	39	20
suv	suv	2	38	45

#Query4: #Total revenue earned from cash

db.bill.aggregate([{$match: {payment_type: "cash"}}, {$group: {_id:null, totalcashrevenue:{$sum:"$FinalAmount"} }}]);



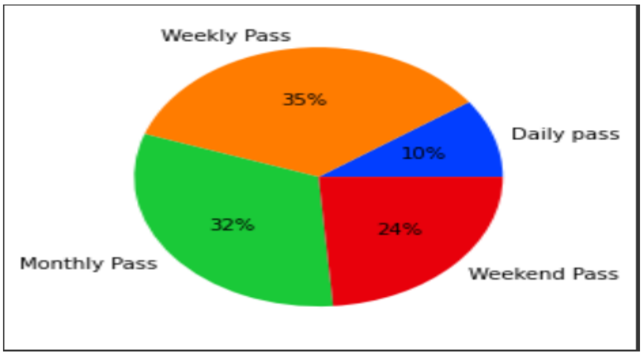
_id	totalcashrevenue
null	2495.0

V. Database Access via Python

With the help of mysql.connector, Python accesses the database, followed by running and fetching the results with cursor.execute,converting the list into a dataframe using the pandas library, and plotting the data frame graphs with pyplot from matplotlib.

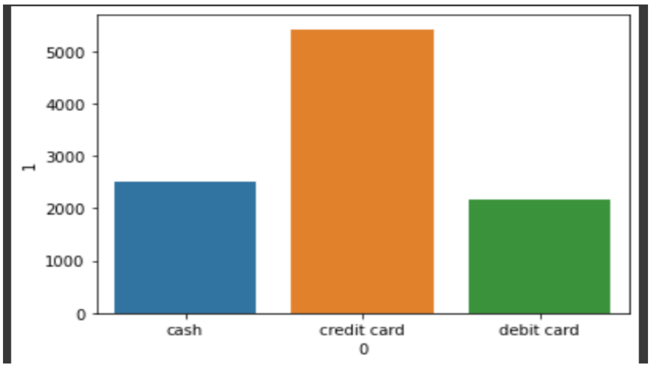
1) Which Subscription pass has the highest demand.

If we increase price of weekly pass gradually then revenue will increase exponentially.



2) Which payment method is used the most:

This data can be used to tie up with credit card company.

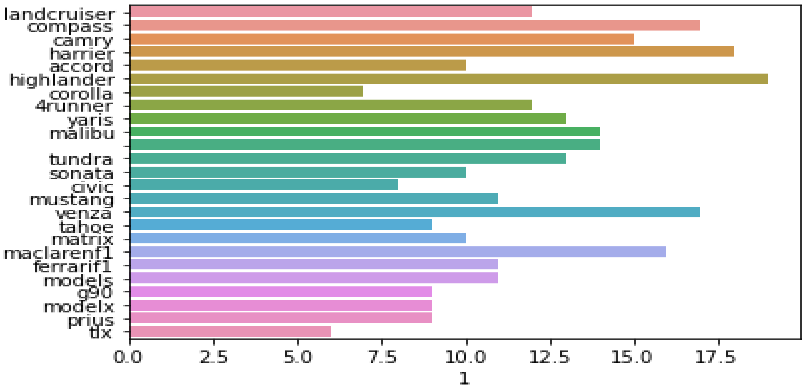


3) Histogram showing car models

And their availability.

This data can be used to make decision of

Future purchase.



VI. Summary and Recommendation:

The Database is ready to use and the Rental car business can easily use the analytics provided to take business decisions. By including more effective data and current client preferences, we can make this project better. Additionally, we were only able to perform a limited number of studies, which we could have easily done with data from the real world because the data we used was generated at random using "filldb".

Future plans for this system include the ability to offer users auto-suggestions for automobiles and subscription packages based on their prior purchases utilizing the Apriori Machine Learning algorithm.

In conclusion, we have succeeded in maintaining a trustworthy database that is available to Business users.