Rental Car System
Group 11

Aditya Gorakh Velapurkar
Sahil Sushil Mahajan

Database Management for Analytics

Why do we need to perform analytics on database created for a Rental Car Services Company?

To answer questions like-

1) To find the present inventory of cars in particular locations.
2) Which category of cars are more in demand?
3) Which type of subscription has more response from the clients?
4) Which payment method is most prefered by the clients?
5) To find who are high spending customers, to which discount can be offered.

Enhanced Entity-Relationship Diagram:

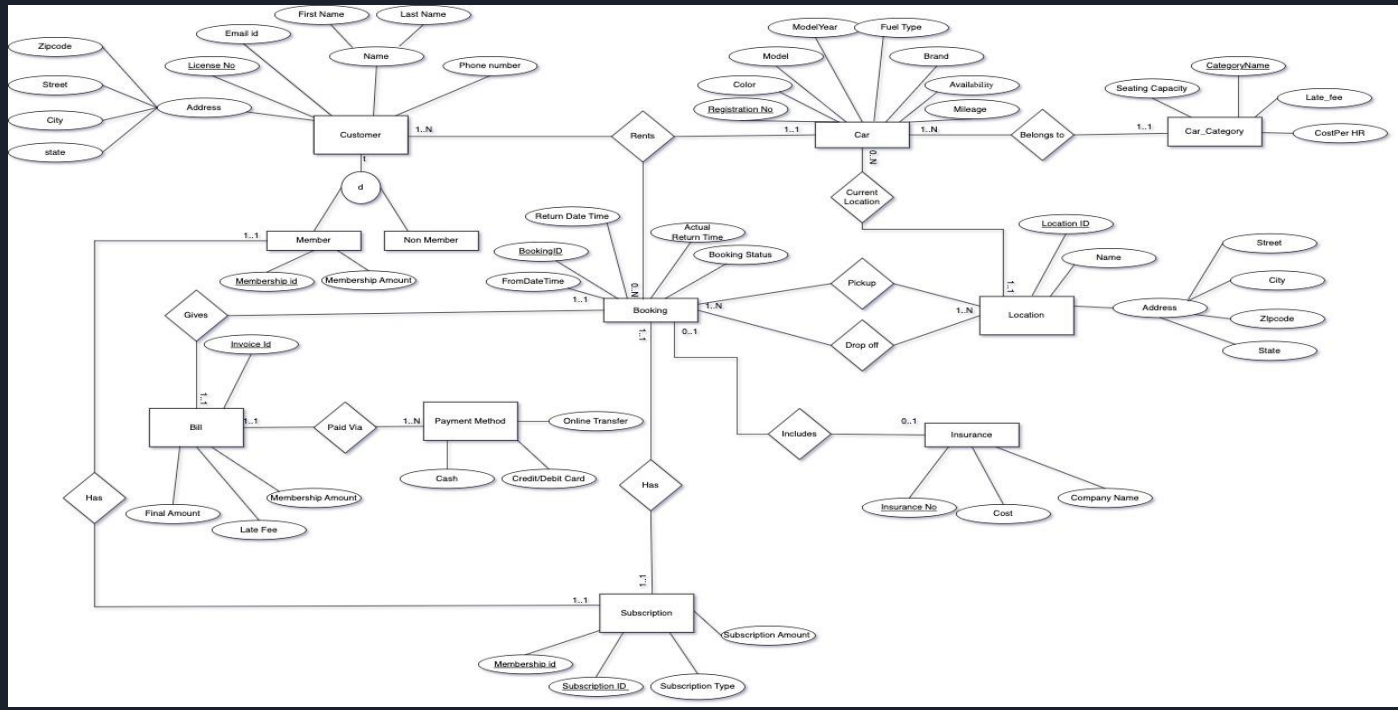
Assumptions:

1) A Customer can Be a member or not.

2) A member must be a Subscriber.

3)Car must belong to only One Car category

4) Pickup and Drop off Location can be different.



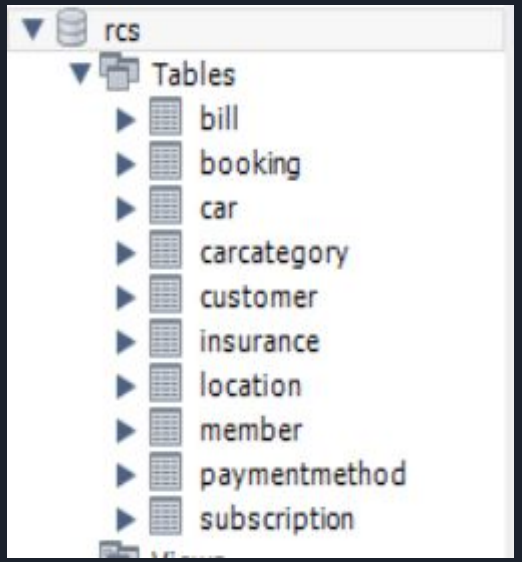
Database created in MySQL Workbench

Database Named “rcs” (Rental Car System)

10 Tables for each entity created:

Eg: Bill Table

InvoiceID	SubscriptionID	MembershipAmount	LateFee	FinalAmount	bookingid	payment_type
5001	106	150	45	215.00	100183	cash
5002	117	150	40	440.00	101947	credit card
5003	133	150	50	300.00	105448	debit card
5004	169	50	15	115.00	110306	cash
5005	178	50	55	125.00	115549	cash
5006	108	200	20	320.00	117673	credit card
5007	125	150	20	220.00	117788	credit card
5008	129	150	55	225.00	127126	debit card



SQL Analytical Queries

1)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
 ( 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'));
```

Output:

	firstname	lastname
▶	Ellsworth	Okuneva
	Josiah	Fadel

The marketing team can approach these customers with a special discounted weekend package.

SQL Analytical Queries

2) (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;
```

Output:

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
220.00	213.40
225.00	218.25

Connecting to Database using Python string

Connecting and performing Sql queries for Visualizations:

```
import mysql.connector
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

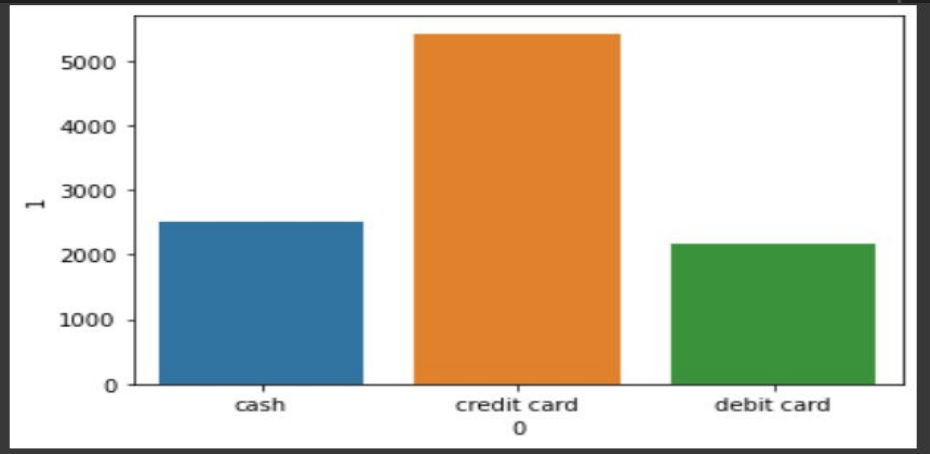
```
#Connecting my python notebook to mysql server
con = mysql.connector.connect(host='localhost',database='rcs', user='root',passwd='OmSai404')

mycur = con.cursor()
```

Query to find revenue via each Payment method.

```
query = 'Select payment_type, sum(FinalAmount) from bill group by payment_type;'
mycur.execute(query)
data=mycur.fetchall()
df=pd.DataFrame(data)
df
```

	0	1
0	cash	2495.0
1	credit card	5420.0
2	debit card	2180.0

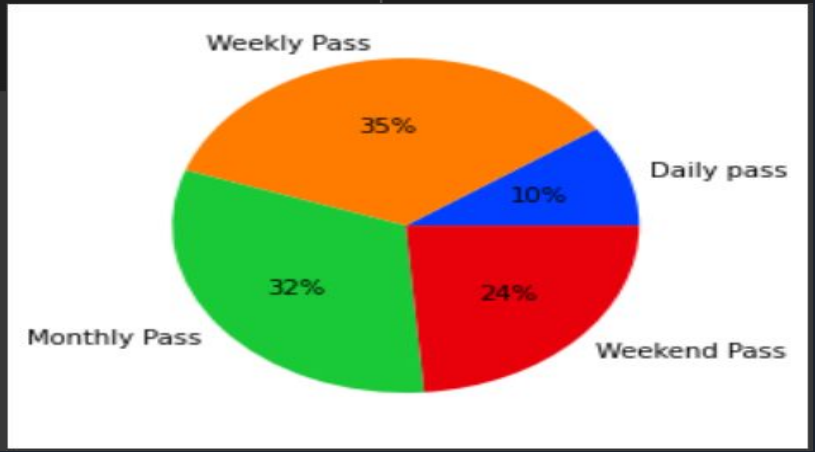


This Data can be used to tie up with credit card companies.

Query to Identify which Subscription has best response

```
query4 = 'select sum(SubscriptionAmount), SubscriptionType from subscription group by SubscriptionType;'
mycur.execute(query4)
data4=mycur.fetchall()
df4=pd.DataFrame(data4)
df4
```

	0	1
0	310	daily
1	1100	weekly
2	1000	monthly
3	750	weekend



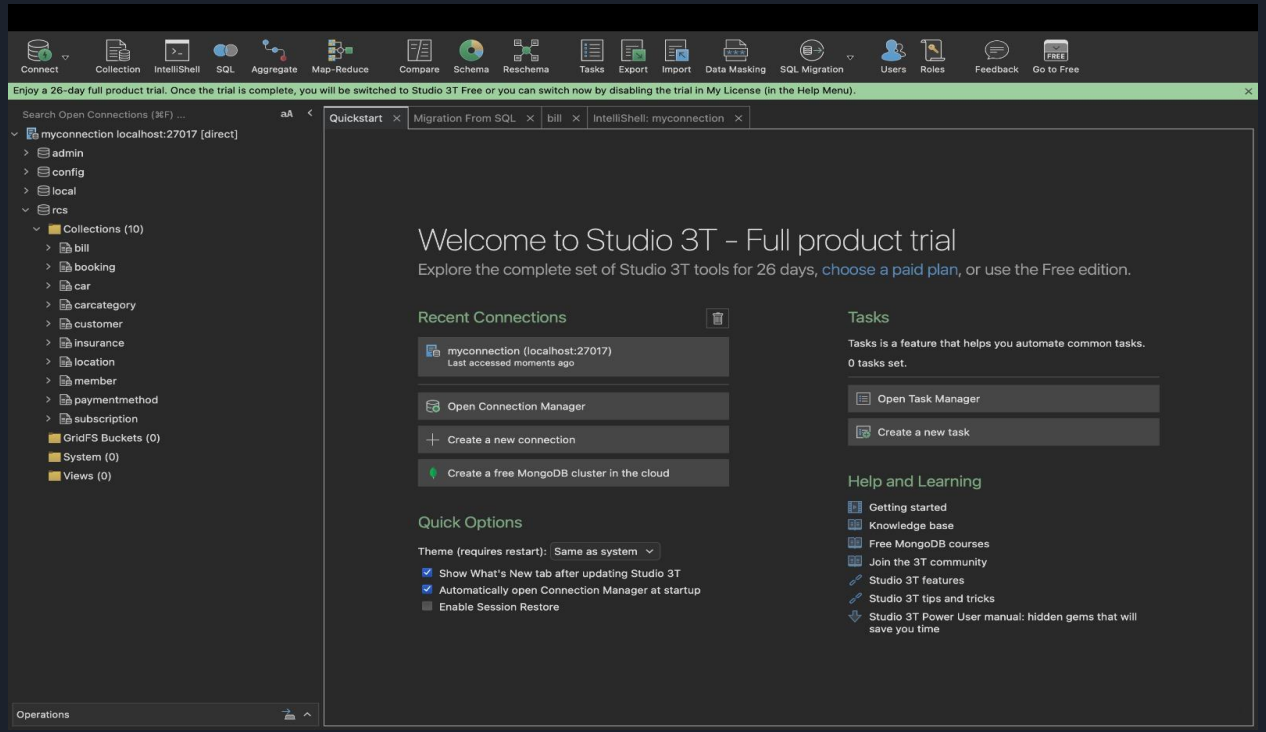
This data can be used to revise pricing for subscription passes.

No-SQL MongoDB

Installation commands on mac-
brew tap mongodb/brew
brew install
mongodb-community@6.0

To run MongoDB as a service-
brew services start
mongodb-community@6.0

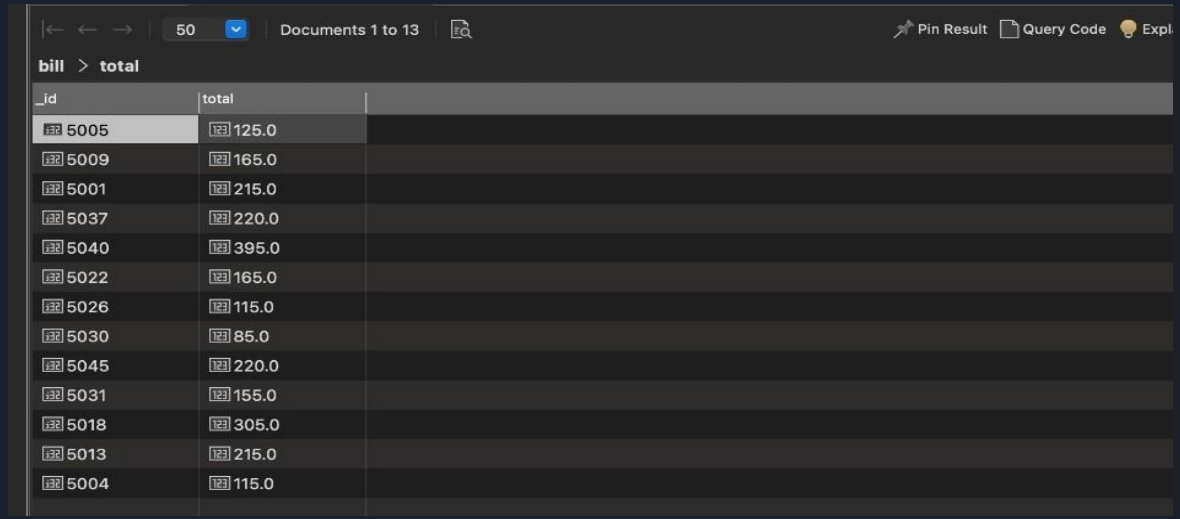
To stop MongoDB-
brew services stop
mongodb-community@6.0



Analytical Queries in MongoDB -1

Using aggregate function to find Invoice IDs and Final amount, who paid Bill by cash.

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

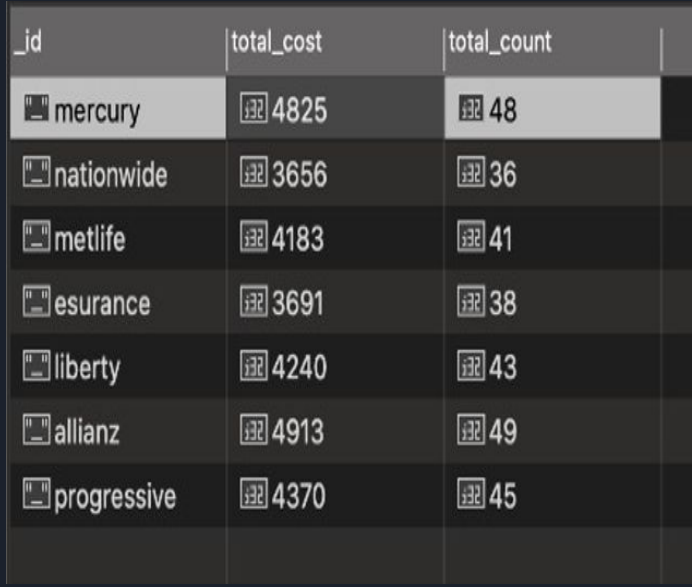


MongoDB Query -2

2) Query to find total amount earned by insurance companies

With the total count of customers catered.

```
db.insurance.aggregate([{$group: {
    _id: "$CompanyName",
    total_cost: {$sum: "$Cost"}, total_count: {$count: {}}
}}  ]);
```



_id	total_cost	total_count
mercury	4825	48
nationwide	3656	36
metlife	4183	41
esurance	3691	38
liberty	4240	43
allianz	4913	49
progressive	4370	45

Future scope:

- Using Apriori machine learning algorithm we can provide auto suggestions to the customers with customised subscription package based on their historical purchases.

- We can provide suggestions of cars which are often rented by the customer previously for which they have provided good feedback.

THANK YOU