Rental Equipment service management

Milestone: Project Report

Group 9

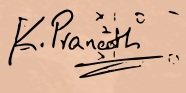
Praneeth Korukonda
Henna Shah

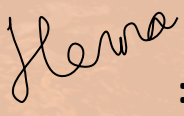
(857)693-9571(Praneeth)
(617)686-5378 (Henna)

korukonda.p@northeasten.edu
shah.henn@northeastern.edu

Percentage of Effort Contributed by Praneeth: 100%

Percentage of Effort Contributed by Henna: 100%

Signature of Praneeth Korukonda: 

Signature of Henna Shah  **:**

Submission Date: 12/10/2022

Table of Contents

Problem statement use cases:

Introduction:

The market size of the U.S. construction sector was valued at around 1.6 trillion U.S. dollars in 2021 and it is expected to increase further in the next year. Rentals. Inc is a rental company, which rents out industrial and construction equipment such as scissor lifts, boom lifts, warehouse forklifts, mini excavators etc.

Rental equipment companies run and manage complex businesses and business processes. They work with some of the heaviest and most complicated equipment in the world as an equipment rental dealer. A service and rental management tool that gives you visibility and control over every piece of equipment you sell, rent or service. If you are doing equipment rental, you demand a lot of key customers effectively. Therefore, such companies must manage equipment purchasing, sale, customer accounts, transaction, and lease details more effectively.

This project aims to build a database management system for industrial and commercial equipment rental company, Rental Inc.

Theory:

Rentals Inc. don't have proper data management for their rented equipment, which can easily lead to lot of confusion in the fleet management, this can cause a great deal of damage to their company in terms of revenue as there will be differences in supply and demand of that particular equipment, to improve their business, we aim to design a data base system which can store all the data related to that particular equipment including the customer details

All the orders are provided with a unique id, which can help the company manage the fleet management and keep track of the equipment which has been rented, these orders are often done through online with the help of 3rd party apps or direct calls to the rental Inc. here we plan to record the equipment details, rental details, Transaction details, customer account details, location details, security deposit details, service request details.

Equipment details provides us the details regarding status of the equipment, original equipment cost, equipment ids and transaction details provides insights into rental revenue of the business.

Customer account details provides customer id, name, age, address, phone number, city, state etc. with the help of this data, the company can keep the track of customer buying behavior, as it is the foundation of the marketing process, each time a customer makes a purchase, an order number is generated together with the related equipment id, equipment type, order date, quantity, and price. With the help of this database management system the rental company can utilize the recorded data to perform numerous analysis, which can in turn help in strategizing different marketing methods to promote the company and increase their rental revenue.

Some of the problems which can be answered with the help of this database system:

1- Find the area where the profitability is high
2- The equipment which is generating the most rental revenue?
3- OEC (original equipment cost) vs revenue generated, is this business profitable?
5- Which equipment has the highest demand?

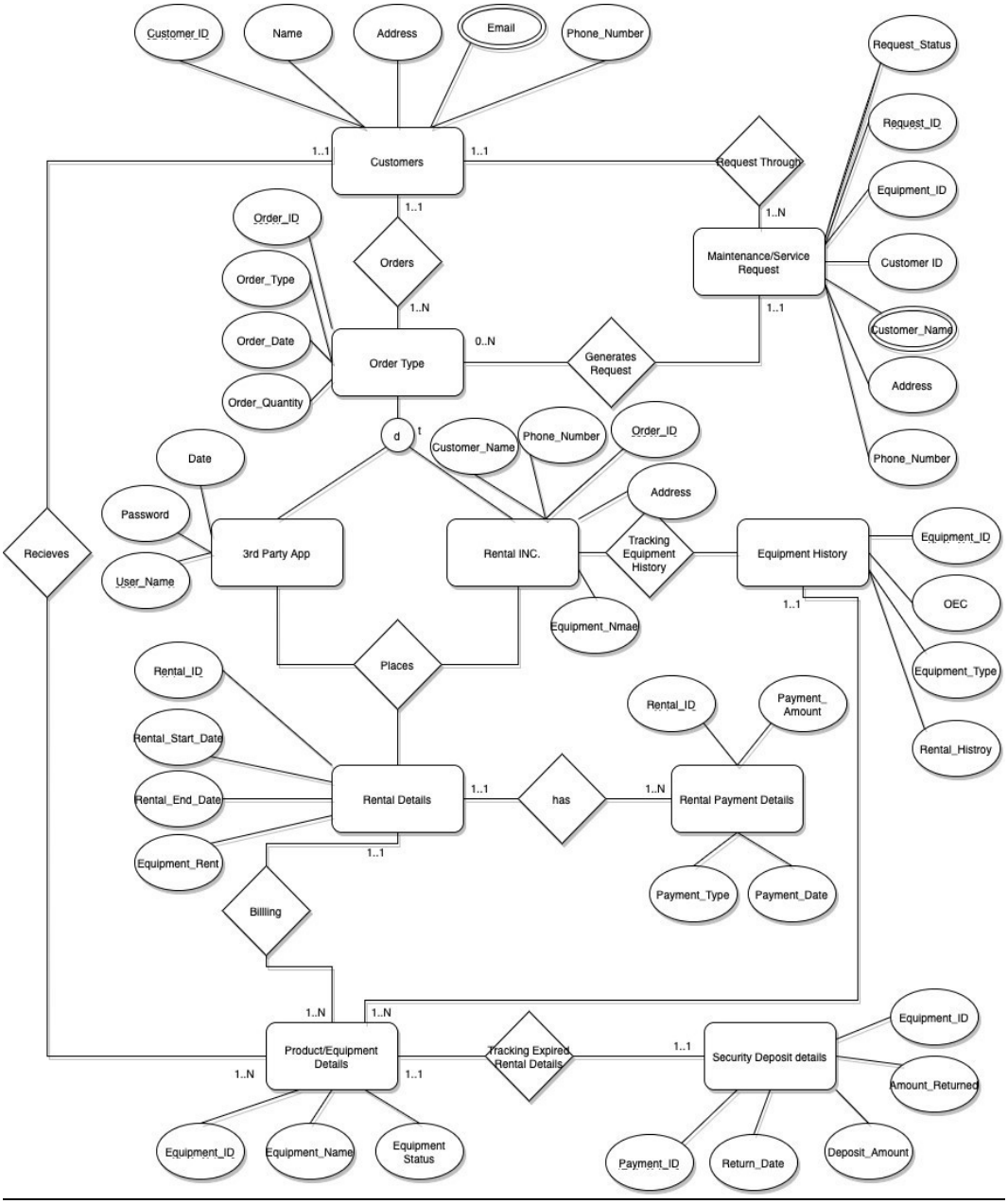
Some of the Tables which can be used while designing the database system:

1. Product/Equipment Details - equip name, type, status, quantity, rental history.
2. Rental details – Rental Id, name, rental start date, rental end date, rental_amount.
3. Rental payment details - payment id, Return Date, Deposit Amount, Amount returned, Equipment ID
4. security deposit details - payment id, return date, security deposit details, amount returned after rent (. This helps in keeping track of expired leases and repayment of security deposits which were taken by the customers during the start of lease.
5. Customers - customer id, name, ph. no, address, city, state.
6. Order type – order Id, order Type, order Date, order Quantity
7. 3rd party app - Date, Username, Password
8. Rental INC. (rental company) – Customer Name, Phone number, Address, Equipment Name
9. Equipment history– Equipment ID, OEC, Equipment type, rental history.
10. Location- (COUNTRY) – location id, state, city, county, country

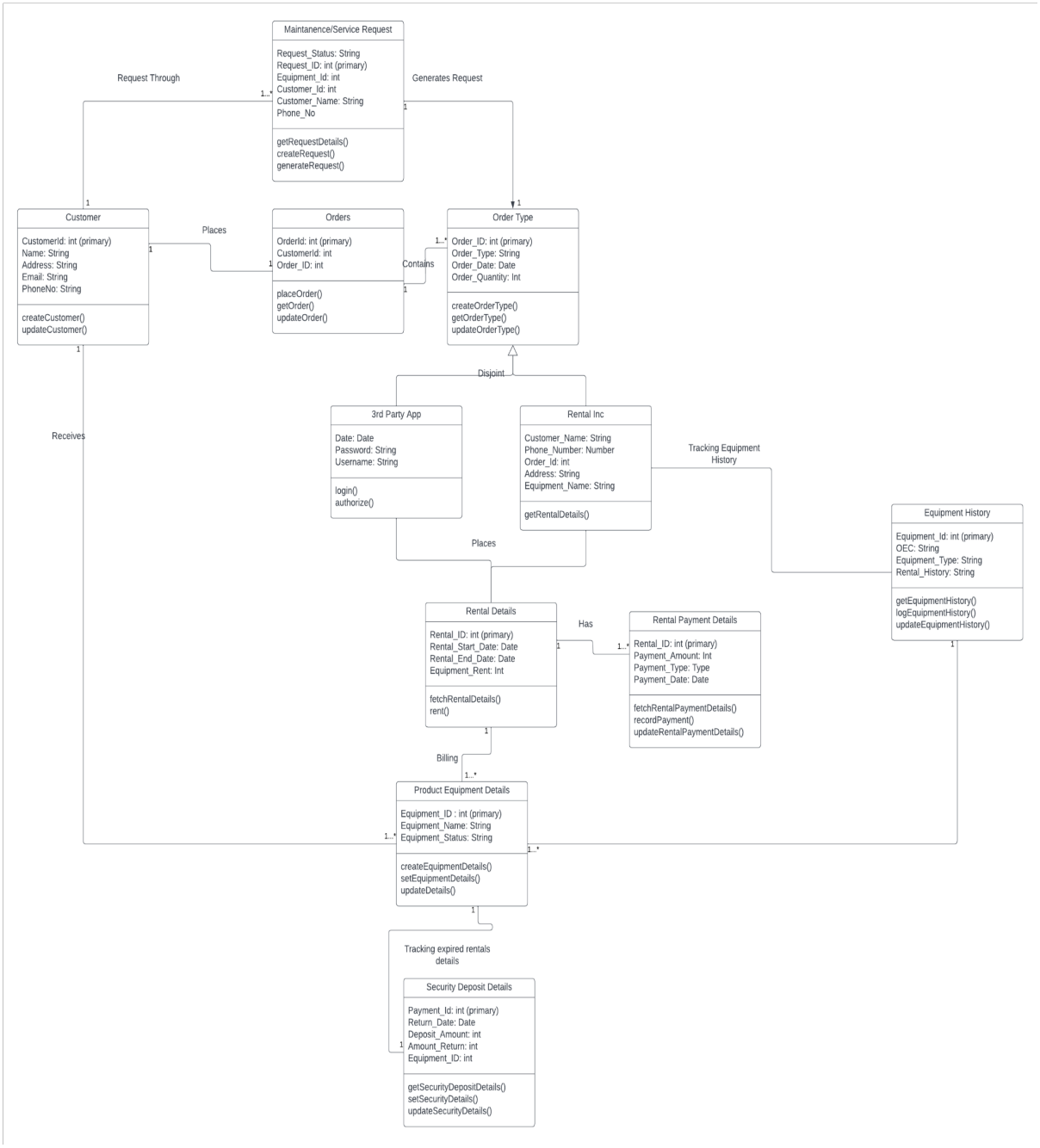
11. Maintenance / Service request table – request id, equipment ID, Customer ID, Customer Name, Address, request status

II. Conceptual Data Modeling

EER Diagram



UML Diagram


Maintanence/Service Request
Request_Status: String
Request_ID: int (primary)
Equipment_Id: int
Customer_Id: int
Customer_Name: String
Phone_No
getRequestDetails()
createRequest()
generateRequest()
Request Through
Generates Request
Customer
CustomerId: int (primary)
Name: String
Address: String
Email: String
PhoneNo: String
createCustomer()
updateCustomer()
Places
Orders
OrderId: int (primary)
CustomerId: int
Order_ID: int
placeOrder()
getOrder()
updateOrder()
Contains
Order Type
Order_ID: int (primary)
Order_Type: String
Order_Date: Date
Order_Quantity: Int
createOrderType()
getOrderType()
updateOrderType()
Disjoint
Receives
3rd Party App
Date: Date
Password: String
Username: String
login()
authorize()
Rental Inc
Customer_Name: String
Phone_Number: Number
Order_Id: int
Address: String
Equipment_Name: String
getRentalDetails()
Tracking Equipment History
Equipment History
Equipment_Id: int (primary)
OEC: String
Equipment_Type: String
Rental_History: String
getEquipmentHistory()
logEquipmentHistory()
updateEquipmentHistory()
Places
Rental Details
Rental_ID: int (primary)
Rental_Start_Date: Date
Rental_End_Date: Date
Equipment_Rent: Int
fetchRentalDetails()
rent()
Has
Rental Payment Details
Rental_ID: int (primary)
Payment_Amount: Int
Payment_Type: Type
Payment_Date: Date
fetchRentalPaymentDetails()
recordPayment()
updateRentalPaymentDetails()
Billing
Product Equipment Details
Equipment_ID : int (primary)
Equipment_Name: String
Equipment_Status: String
createEquipmentDetails()
setEquipmentDetails()
updateDetails()
Tracking expired rentals details
Security Deposit Details
Payment_Id: int (primary)
Return_Date: Date
Deposit_Amount: int
Amount_Return: int
Equipment_ID: int
getSecurityDepositDetails()
setSecurityDetails()
updateSecurityDetails()

III. Mapping Conceptual Model to Relational Model

Customers (Customer_ID, Name, Address, Phone_Number)
Primarykey: Customer ID
Foreign key: Email_ID, Equipment ID. (NOT NULL NOT ALLOWED)

Email (Email_ID, Name, Email_id 1, Email_id 2)
Primary key: Email_ID
Foreign Key: NA

Name (Customer_ID, First_Name, Last_Name)
Foreign Key: Customer_ID
Primary Key:NA

Maintainance/Service History (Request_ID, request_Status, Address, Phone_Number, Customer_ID, Order_ID)
Primary Key: Request_ID
Foreign Key: Customer_ID, Order ID. (NOT NULL NOT ALLOWED)

Order_Type (Order_ID, Order_Type, Order_Date, Order_Quantity, Customer ID)
Primary Key: Order_ID
Foreign Key: Customer ID. (NOT NULL NOT ALLOWED)

3rd Party App (Order_ID, Date, Password, User_Name)
Foreign Key: Order_ID
Primary Key:NA

Equipment History (Original Equipment Cost, Equipment_Type, Rental History, EquipmentID, Order_ID)
Foreign Keys: EquipmentID, Order_ID
Primary Key: NA

Rental INC (Order_ID, Customer_Name, Phone_Number, Address, Equipment Name)
Foreign Key: Order_ID
Primary Key: NA

Rental details (Rental_ID, Rental_Start_Date, Rental_End_Date, Equipment_Rent, Order_ID) Primary Key: Rental_ID
Foreign Key: Order_ID. (NOT NULL NOT ALLOWED)

Rental Payment Details (Payment_Amount, Payment_Type, Payment_Date, Rentail_ID)
Foreign Key: Rental_ID (NOT NULL NOT ALLOWED)
Primary Key: NA

Product/equipment details (Equipment ID, Equipment_Name, Equipment_Status, Rental_ID, Customer ID)
Primary Key: Equipment ID
Foreign Keys: Rental_ID, Customer ID (NOT NULL NOT ALLOWED)

Security Deposit details(Payment ID, Return_Date, Deposit_Amount, Amount_Returned, Equipment ID)
Primary Key: Payment ID
Foreign Key: Equipment ID (NOT NULL NOT ALLOWED)

IV. Implementation of Relation Model via MySQL and NoSQL

MySQL Implementation:

The database was created in MySQL and the following queries were performed:

1) Query to find the type which is being preferred by the customer to order the equipment

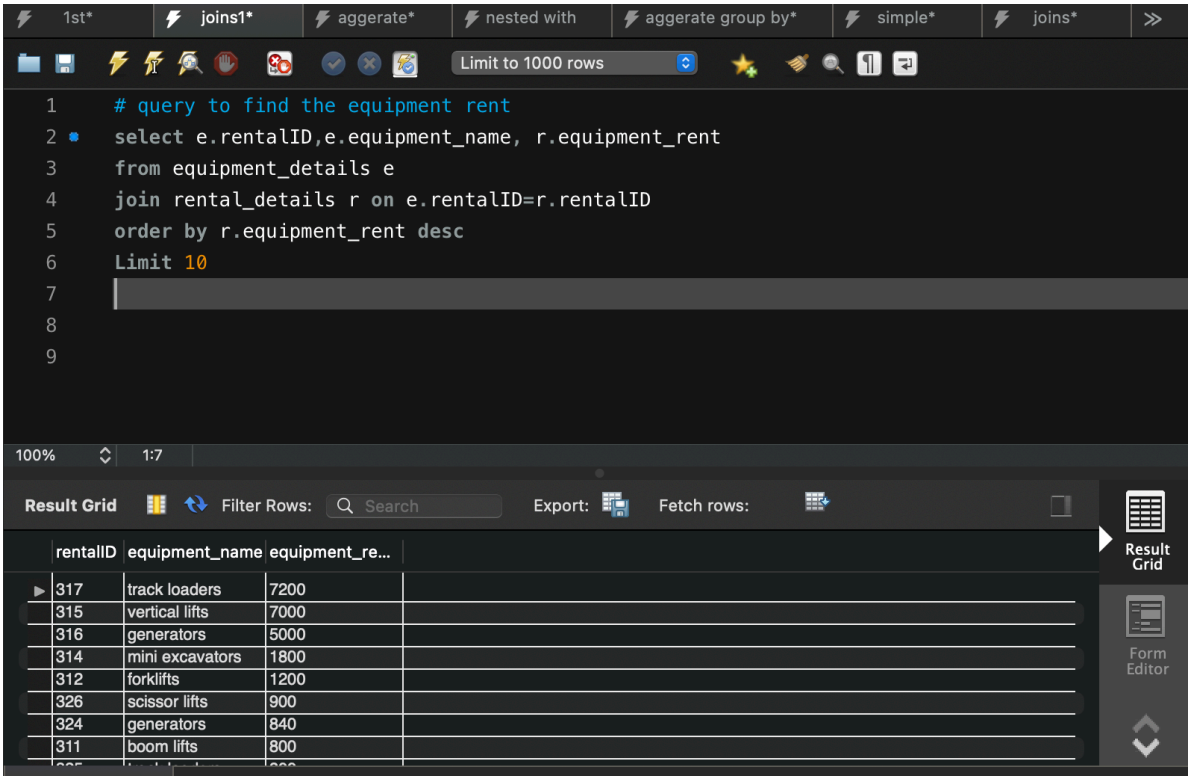
```
1    # to find the type order which is being most used
2 •  select order_type, count('orderID')
3    from orderdetails
4    group by 1
```

100% 11:4

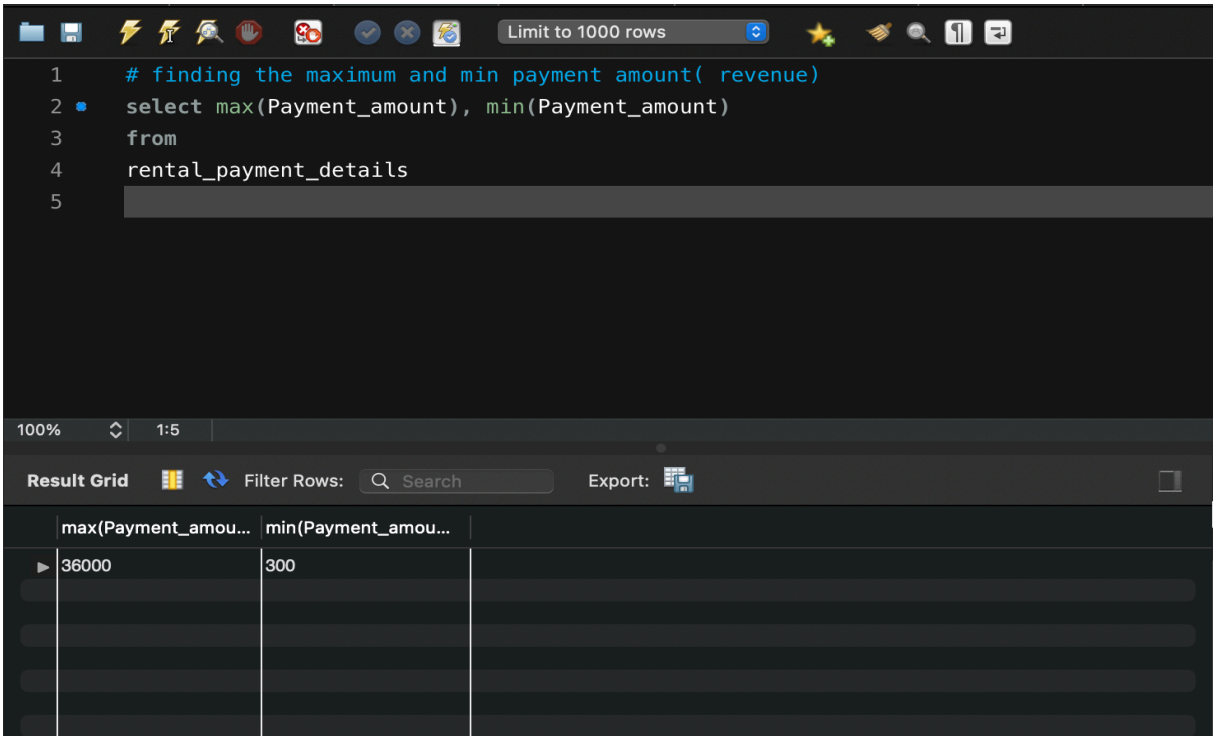
Result Grid Filter Rows: Search Export:

order_type	count('orderI...
3rd party app	12
Rental_INC	8

2) Query to find the rent of each equipments available at rental inc



3) Query to find the minimum and maximum payment amount



4) Query to find the method which has the highest revenue

```
# Payment method that has the highest revenue
with a as (select Payment_Type, sum(Payment_Amount) as Total_Amt
from rental_payment_details group by Payment_Type)
select * from
a where Total_Amt = (select max(Total_Amt) from a)
```

Result Grid

Payment_Type	Total_Amt
credit	180750

5) Query to find the area which has more number of customers

```
# query to find area where the profitibility is high
select state, count(customerID)
from customers
group by 1
order by count(customerID) desc;
```

Result Grid

state	count(customerI...
TX	6
CA	6
MA	4
GA	3
MD	1

6) query to find the equipments which are available for rent

```
1    # query to find the equipments which are avaliable
2    select equipment_name, equipment_status
3    from equipment_details
4    where equipment_Status = 'available'
5
```

100% 1:5

Result Grid | Filter Rows: Search | Export:

equipment_name	equipment_stat...
boom lifts	Available
mini excavators	Available
vertical lifts	Available
track loaders	Available
scissor lifts	Available
boom lifts	Available
mini excavators	Available
scissor lifts	Available

7) query to find the customer who is requesting for service

Limit to 1000 rows

```
1    # query to find the customer who is requesting for service
2    select c.name, h.customerID, h.requestStatus,h.state
3    from service_request h
4    left join customers c on h.customerid= c.customerid
5    where h.RequestStatus = 'requested'
6
7
```

100% 1:7

Result Grid | Filter Rows: Search | Export:

name	customerID	requestStat...	state
Jhon kennedy	201	requested	MA
cristano ronaldo	212	requested	TX
chris hemstorth	218	requested	TX
joe biden	220	requested	MA

8) query to find the customer who rented the most expensive equipment

```
# query to find the customer who rented the most expensive equipment
select * from customers where equipment_ID =
(select equipmentID
from rental_details
 where Equipment_Rent =
(select max(Equipment_Rent) from rental_details))
```

100% 1:7

Result Grid Filter Rows: Search Export:

CustomerID	equipment_ID	Name	city	state	Phonenumber	EmailID
208	1008	Curtis Lowe	dallas	TX	860 4675 311	Curtis Lowe@gmail.com

9) query to get equipments which has the rent above the average rent

```
#query to get equipments which has the rent above the average rent
select e.equipmentID, e.equipment_name, r.equipment_rent
from equipment_details e
join rental_details r on e.EquipmentID= r.EquipmentID
where Equipment_Rent > ALL
(select avg(Equipment_Rent) as avv_equipmentrent
from rental_details);
```

100% 1:8

Result Grid Filter Rows: Search Export:

equipmentID	equipment_name	equipment_re...
1005	mini excavators	1800
1006	vertical lifts	7000
1007	generators	5000
1008	track loaders	7200

10) query to find the rent of each equipment

```
# query to find the rent of each equipment
select e.rentalID,e.equipment_name, r.equipment_rent
from equipment_details e
join rental_details r on e.rentalID=r.rentalID
order by r.equipment_rent desc
Limit 10
```

100% 1:7

Result Grid | Filter Rows: Search | Export: | Fetch rows:

rentalID	equipment_name	equipment_re...
317	track loaders	7200
315	vertical lifts	7000
316	generators	5000
314	mini excavators	1800
312	forklifts	1200
326	scissor lifts	900
324	generators	840
311	boom lifts	800

11) Top 5 priced Equipments using correlated query

```
# Top 5 priced Equipments using correlated query
select a.equipmentID from rental_details a
where 5 > (select count(*) from rental_details b
where  a.equipmentID!=b.equipmentID
and a.Equipment_Rent<b.Equipment_rent)
```

100% 1:7

Result Grid | Filter Rows: Search | Export:

equipmentID
1003
1005
1006
1007
1008

12) Query to find customers who ordered the equipment with rent more than 200

```
# customers who ordered the equipment with rent more than 200
select e.Equipment_Name, c.Name,c.Phonenumber,c.EmailID
from equipment_details e join customers c
on e.EquipmentID = c.equipment_ID
where exists(
select * from rental_details rd
where rd.Equipment_Rent > 200)
```

100% 1:9

Result Grid Filter Rows: Search Export:

Equipment_Name	Name	Phonenumber	EmailID
scissor lifts	Jhon kennedy	857 4989 321	Jhon kennedy@gmail.com
boom lifts	Mike tyson	857 4675 311	Mike tyson@gmail.com
forklifts	Tommy ginger	858 4989 321	Tommy ginger@gmail.com
track loaders	Margarette Hammes	858 4675 311	Margarette Hammes@gmail.com
mini excavators	Jeanie Fahey	859 4989 321	Jeanie Fahey@gmail.com
vertical lifts	Brando Kshlerin	859 4675 311	Brando Kshlerin@gmail.com
generators	Donato Schoen	860 4989 321	Donato Schoen@gmail.com
track loaders	Curtis Lowe	860 4675 311	Curtis Lowe@gmail.com

No SQL Implementation

Three tables(Equipment Details,Order details, Equipment histroy) have been created in mongo db playground. The following are the queries which were performed

Query 1: Finding all the equipment which has been rented out

```
db.equipment_details1.find({'Equipment_Status' : 'rented'}).pretty()
```

Run

Reset

Click here to reset the database to its initial state (all your changes will be lost).

Tips

Enter MongoDB Javascript commands in the text area. Pressing "Run" will present the result of the MongoDB shell output. Try `db.getCollectionNames();` to see defined collections and `db.COLLECTIONAME.find();` to retrieve a list of documents inside the given collection. See the MongoDB reference for useful commands.

Result


```
{
        "_id" : ObjectId("638b99597b07687302f5d061"),
        "EquipmentID" : 1001,
        "Equipment_Name" : "scissor lifts",
        "Equipment_Status" : "rented",
        "CustomerID" : 201,
        "rentalID" : 310
}
{
        "_id" : ObjectId("638b9a2549fc8b5e519e2275"),
        "EquipmentID" : 1003,
        "Equipment_Name" : "forklifts",
        "Equipment_Status" : "rented",
        "CustomerID" : 203,
        "rentalID" : 312
}
{
        "_id" : ObjectId("638b9a2549fc8b5e519e2276"),
        "EquipmentID" : 1004,
        "Equipment_Name" : "track loaders",
        "Equipment_Status" : "rented",
        "CustomerID" : 204,
        "rentalID" : 313
}
{
        "_id" : ObjectId("638b9a2549fc8b5e519e2279"),
        "EquipmentID" : 1007,
        "Equipment_Name" : "generators",
        "Equipment_Status" : "rented",
        "CustomerID" : 207,
        "rentalID" : 316
}
{
        "_id" : ObjectId("638b9a2549fc8b5e519e227c"),
        "EquipmentID" : 1010,
        "Equipment_Name" : "boom lifts",
        "Equipment_Status" : "rented",
        "CustomerID" : 210,
        "rentalID" : 319
}
```

Query 2: Finding the method used by the customer to book the equipment

Write your statement below and press "Run" to see the result.

```
db.orderdetails1.aggregate(
[{$group:{_id:"$Order_type", total_orders_from:{$sum:"$Order_Quantity"}}}, {$sort:{total_orders_from:1}}
]);
```

Run

Result

```
{ "_id" : "rental_inc", "total_orders_from" : 23 }
{ "_id" : "3rd party app", "total_orders_from" : 38 }
```

Query 3: Finding the equipment's which have their original equipment cost greater than 25000

Write your statement below and press "Run" to see the result.

```
db.equipment_histroy1.find({'OEC':{$gt:25000}}).pretty()
```

Run

Result


```
{
        "_id" : ObjectId("638bb912b482d152a8851c81"),
        "OEC" : 50000,
        "Rental_Histroy" : "rented",
        "Equipment_ID" : 1002,
        "Equipment Name" : "boom lifts"
}
{
        "_id" : ObjectId("638bb912b482d152a8851c82"),
        "OEC" : 40000,
        "Rental_Histroy" : "rented",
        "Equipment_ID" : 1003,
        "Equipment Name" : "fork lifts"
}
{
        "_id" : ObjectId("638bb912b482d152a8851c88"),
        "OEC" : 50000,
        "Rental_Histroy" : "rented",
        "Equipment_ID" : 1009,
        "Equipment Name" : "mini excavators"
}
```

V. Database Access via Python

The database is accessed using Python and visualization of analyzed data is shown below. The connection of MySQL to Python is done using mysql.connector, followed by cursor.excecute to run and fetchall from query, followed by converting the list into a data frame using pandas library and using matplotlib to plot the graphs for the analytics.

```
pip install mysql-connector-python
```

```
Requirement already satisfied: mysql-connector-python in ./opt/anaconda3/lib/python3.9/site-packages (8.0.31)
Requirement already satisfied: protobuf<=3.20.1,>=3.11.0 in ./opt/anaconda3/lib/python3.9/site-packages (from mysql
-connector-python) (3.20.1)
Note: you may need to restart the kernel to use updated packages.
```

```
import mysql.connector
from mysql.connector import Error
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

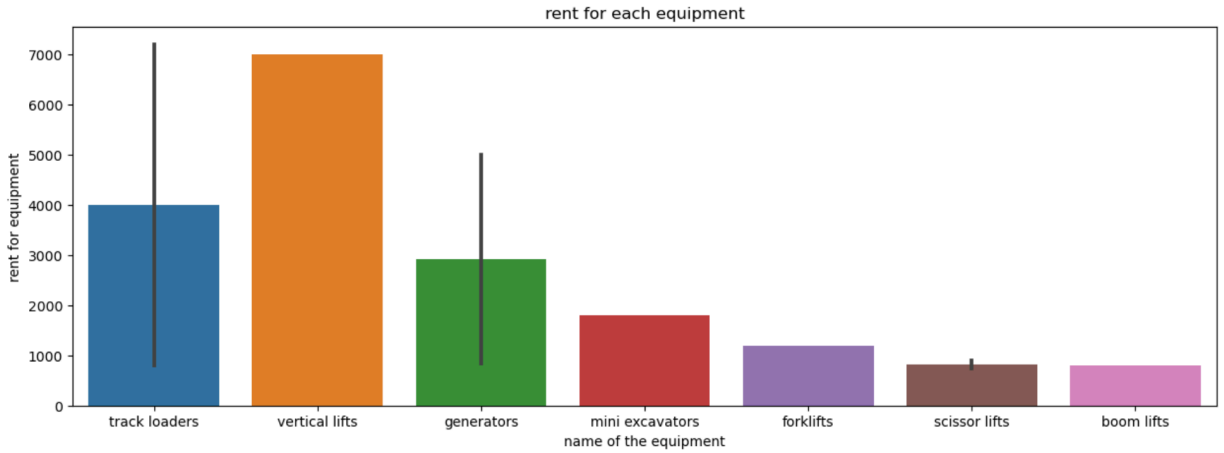
```
connection = mysql.connector.connect(user='root', password='admin1234',
                                     host='localhost',
                                     database='rental_inc')
```

```
sql_select_Query = "show tables"
cursor = connection.cursor()
cursor.execute(sql_select_Query)
records = cursor.fetchall()
for x in records:
  print(x)
```

```
('customers',)
('EMAIL',)
('equipment_details',)
('equipment_histroy',)
('NAME',)
('orderdetails',)
('rental_details',)
('rental_inc',)
('rental_payment_details',)
('security_deposit_details',)
('service_request',)
('third_party_app',)
```

Some analytics which were done through python:

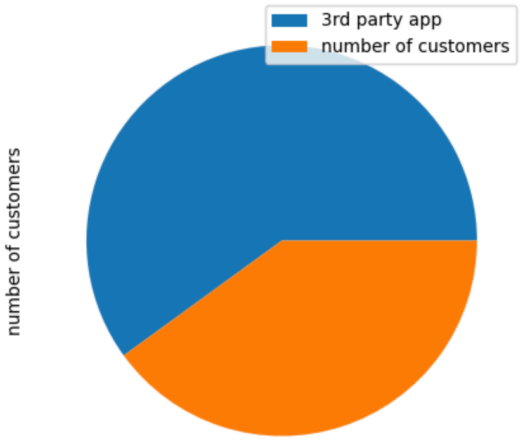
1) Rent of each equipment available at Rental INC



2) **Type of method preffered by the customer to book the equipment**

```
plot = df_typ.plot.pie(y='number of customers', figsize=(5, 5),labels=None)
plot.legend(['3rd party app', 'number of customers'])
```

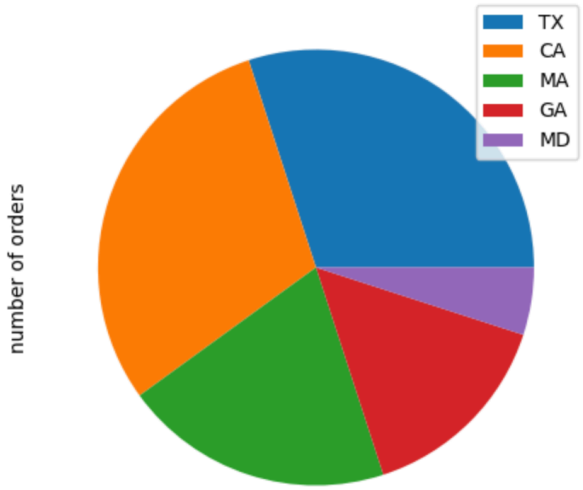
```
<matplotlib.legend.Legend at 0x7f9d014bf460>
```



3) To find the state which has most number of orders

```
plot = df_st.plot.pie(y='number of orders', figsize=(5, 5),labels=None)
plot.legend(['TX', 'CA', 'MA', 'GA','MD'])
```

```
<matplotlib.legend.Legend at 0x7f9d1848b130>
```



VI. Summary and Recommendation

The Rental_INC database designed on MySQL is an industry ready relational database that can be implemented in any rental equipment industry. It can be helpful in lot of cost savings and provides great analytical capabilities, small part of which is shown in this report utilizing Python.

By building the centralized Rental database system we should be able to optimize by making sure only required equipment's are available at the required place or make it available at the required time. Let's consider a scenario where a customer orders an equipment and the customer representative investigates the databases and confirms whether the equipment is available or not, in case if the equipment is not available the customer representative can request it from the nearest branch and deliver it to the customer within committed time.