**EHubWash**

**Milestone: Project Report**

# **Group 2**

# Sneha Amin

# Kaajal Shah

+1 (857) 398 5938

+1 (857) 396-6019

# [amin.sn@northeastern.edu](mailto:amin.sn@northeastern.edu)

# [shah.kaa@northeastern.edu](mailto:shah.kaa@northeastern.edu)

**Percentage of Effort Contributed by Student1:** 50%

**Percentage of Effort Contributed by Student1:** 50%

**Signature of Student 1:** *Sneha Amin*

**Signature of Student 2:** *Kaajal Shah*

Submission Date:\_**04/21/2024**

**Executive Summary**

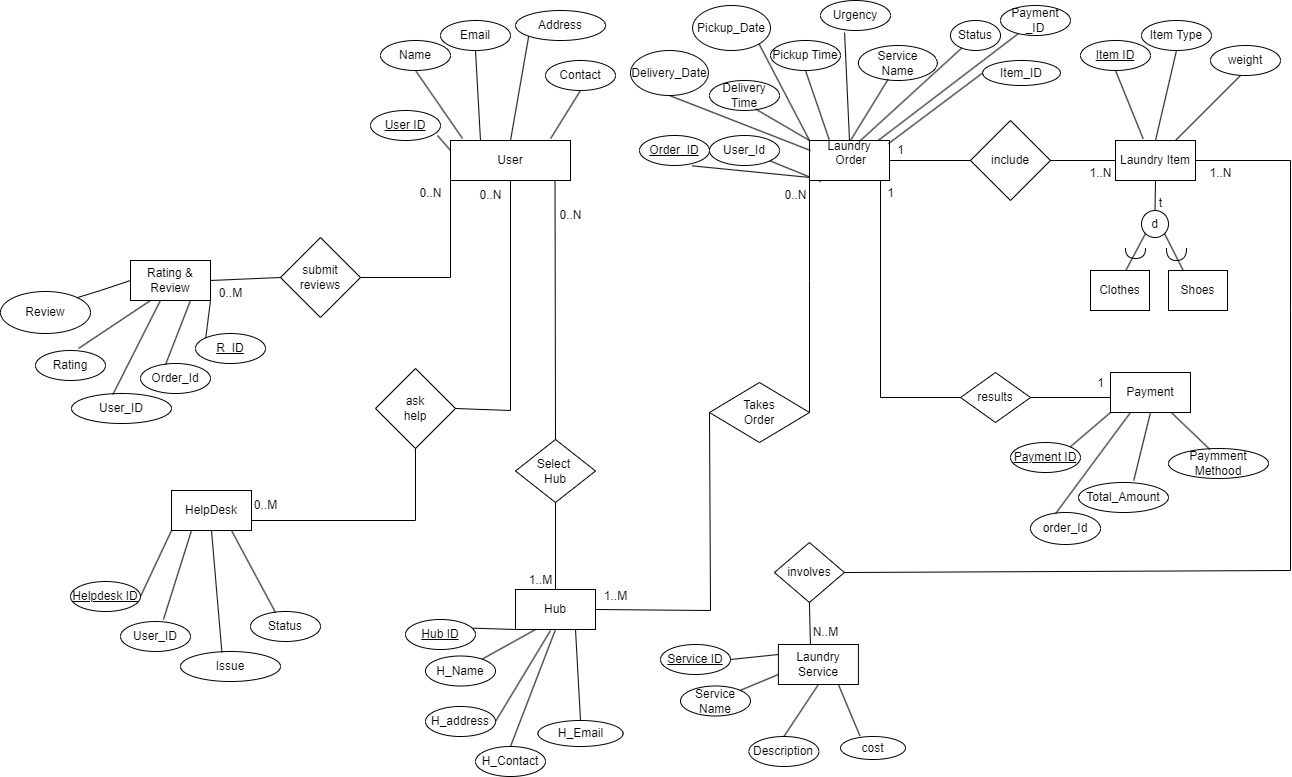
EWash Hub is a premier pick-and-drop laundry service committed to providing convenience and top-notch quality to our clientele. Recognizing the demands of contemporary lifestyles, our service is tailored to seamlessly integrate with the busy schedules of our customers. Our offerings include prompt pickup of laundry within 48 hours, ensuring it is returned freshly cleaned and pressed. For urgent requirements, we provide an express service guaranteeing delivery within 6 hours for a nominal additional fee. The cornerstone of our service lies in our user-friendly mobile application, designed to streamline the laundry process and cater to the unique preferences of each user. By creating an account, users can personalize their experience, specifying service types, detergent preferences, and any special instructions for handling their laundry. Our scheduling feature allows users to book laundry pickups at their convenience, selecting preferred time slots for both pickup and delivery. Users have access to a wide array of laundry services through our application, ranging from washing and drying to steam pressing and shoe cleaning. Tailored services are available for various garment types, ensuring specific care for regular clothes, party attire, coats, delicate items, woolens, and shoes. Additionally, our application features a robust customer support system, including helpdesk support, chat assistance, and a FAQs section for addressing queries promptly. Users can also rate and review our services, contributing to the maintenance of our quality standards. Transparent billing, detailed invoices, and multiple payment options, including credit/debit cards and digital wallets, further enhance the user experience, ensuring a seamless and hassle-free laundry service.

**Introduction:**

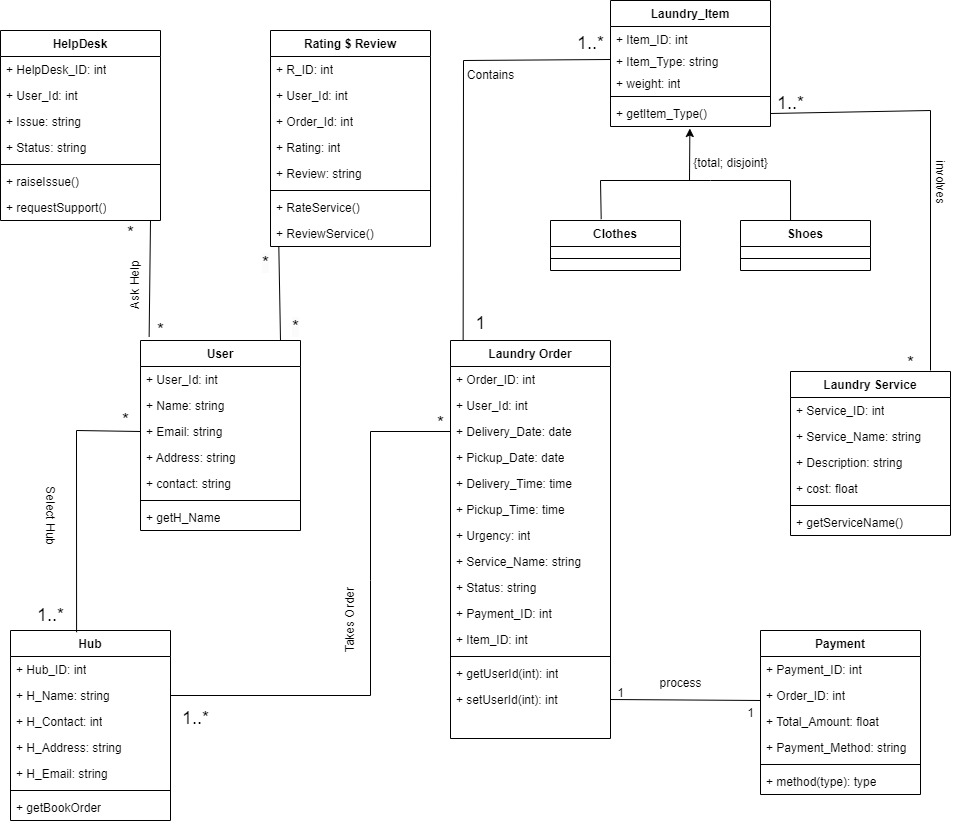
EWash Hub is a premium pick-and-drop laundry service aimed at providing convenience and quality to our customers. We understand the hustle of modern life and the need for a reliable laundry service that fits seamlessly into your busy schedule. Our services include picking up your laundry within 48 hours and delivering it back to you freshly cleaned and pressed. For those with urgent needs, we offer an express service where your laundry will be taken care of within 6 hours, for a nominal additional fee.

* EWash Hub offers a user-friendly mobile application designed to make the laundry process seamless, convenient, and tailored to your needs.
* Users can create an account with EWash Hub by providing essential details such as name, address, and contact information.
* Profile customization allows users to set preferences, including preferred service types, detergent choices, and any specific instructions for handling their laundry.
* Users can schedule laundry pickups at their convenience using the app.
* Select a preferred time slot for both pickup and delivery, accommodating your busy schedule. Choose from a variety of laundry services, including washing, drying, steam pressing, and shoe cleaning.
* Select specific services for different types of items such as regular clothes, party attire, coats, delicate items, woolens, and shoes.
* For urgent needs, users can opt for the express service, ensuring laundry is picked up and delivered within 6 hours (additional fee applies). Track the status of your laundry order in real-time.
* Receive notifications at each stage, from pickup to cleaning, pressing, and delivery. Seamless payment options through the app, including credit/debit cards or digital wallets.
* Transparent billing with detailed invoices, including a breakdown of charges based on the weight of clothes and pairs of shoes.

**EER**

****

**UML**

****

**Relational Model**

Primary Key, *Foreign Key*

**User** (UserID, Name, Email, Address, Contact)​

**Laundry Order** (OrderID, UserID, delivery date, delivery time, pickup date, pickup time, urgency, servicename, status, payment id, item id)​

**Laundry Item** (ItemID, Item type, weight, *OrderID*)​

**Clothes** (ItemID)

​

**Shoes** (ItemID)​

**Payment** (PaymentID, *OrderID*, total amount, payment method)​

**Laundry service** (serviceID, service name, description, cost)​

**Hub** (HubID, hub name, hub add, hub contact, hub email)​

**Helpdesk** (HelpdeskID, UserID, issue, status)​

**Rating and review** (RID, OrderID, UserID, rating, review)​

**Takes Order** (*OrderID, HubID*)​

**Involves**(*ItemID, ServiceID*)​

**Select Hub** (*UserID, HubID*)​

**Ask Help**(*UserID, HelpdeskID*)​

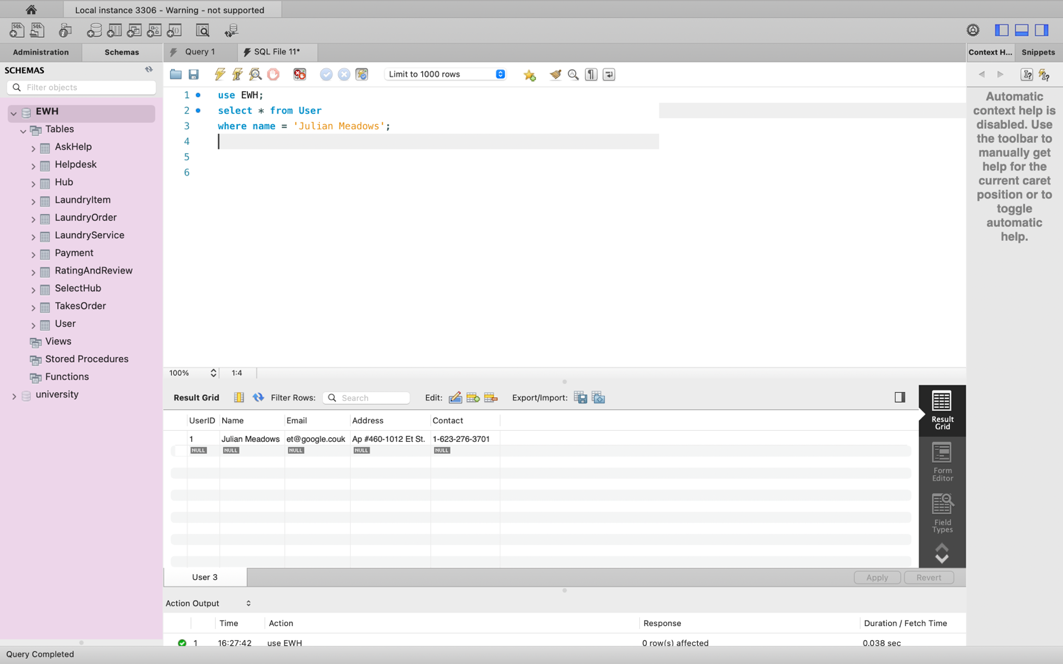
**Submit Reviews** (*RID, UserID*)​

**Implementation in MySQL**

The database was created in MySQL and the following queries were performed: **1. Retrieve all information about the user named "Julian Meadows" from User table.**

select \* from User​

where name = 'Julian Meadows';

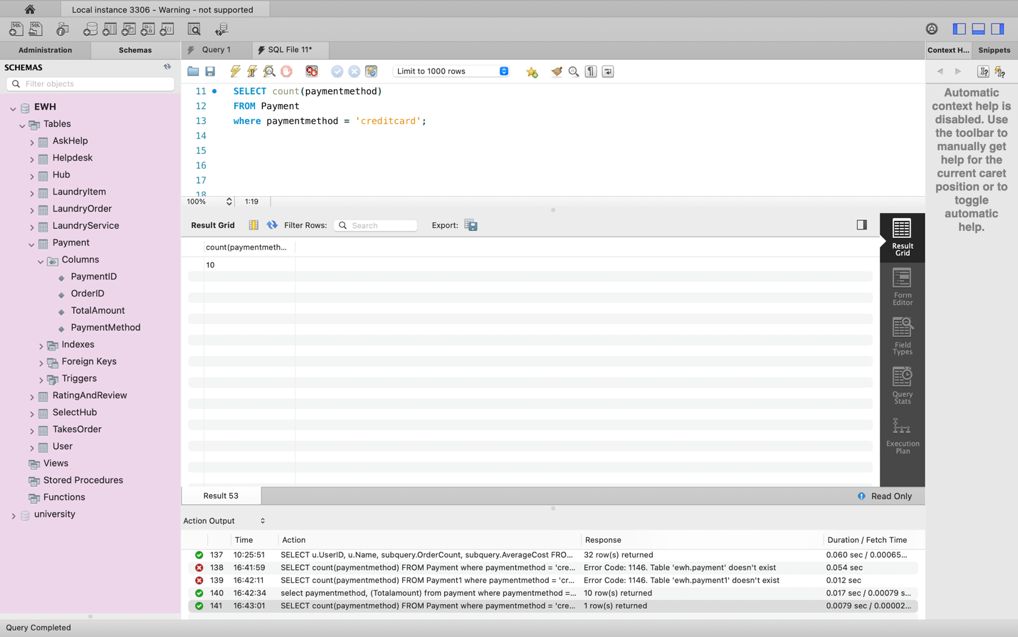


**2. retrieving the count of rows from the Payment table where the paymentmethod is 'creditcard'.**

SELECT count(paymentmethod)

FROM Payment

where paymentmethod = 'creditcard';



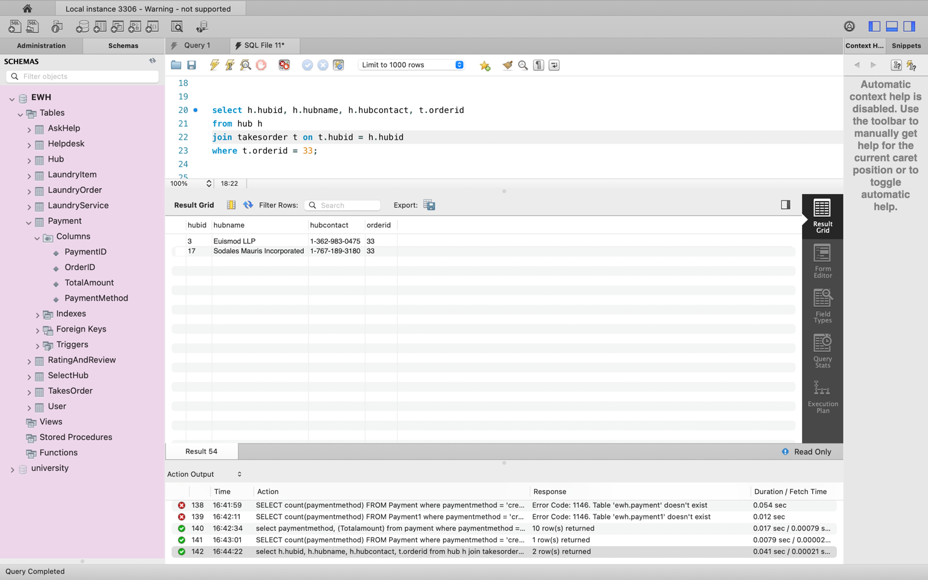
**3. retrieve hubid, hubname, hubcontact, orderid from table hub and takesorder for orderid 33.**

select h.hubid, h.hubname, h.hubcontact, t.orderid​

from hub h​

join takesorder t on t.hubid = h.hubid​

where t.orderid = 33;​



**4. Select Name and UserID from the User table where the UserID is 37 who asked for help.​**

SELECT Name, UserID ​

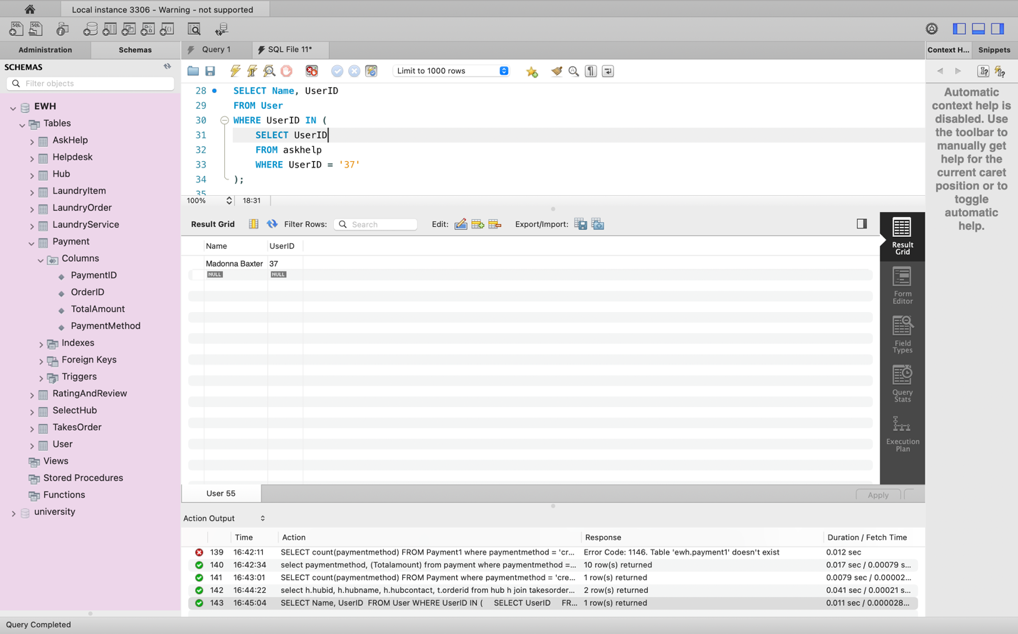
FROM User​

WHERE UserID IN (​

    SELECT UserID​

    FROM askhelp​

    WHERE UserID = '37');​



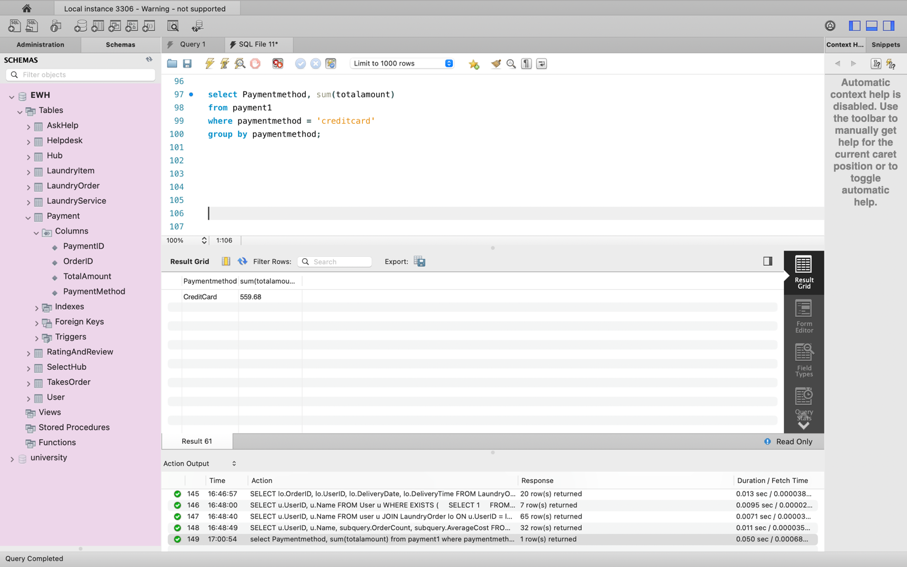
**5. Select Paymentmethod and the sum of totalamount from payment1 where paymentmethod is 'creditcard'​**

select Paymentmethod, sum(totalamount)​

from payment1​

where paymentmethod = 'creditcard'​

group by paymentmethod = '37');​



**6. Select the UserID and Name from the User for those users who have placed orders in the Laundry.​**

SELECT u.UserID, u.Name​

FROM User u​

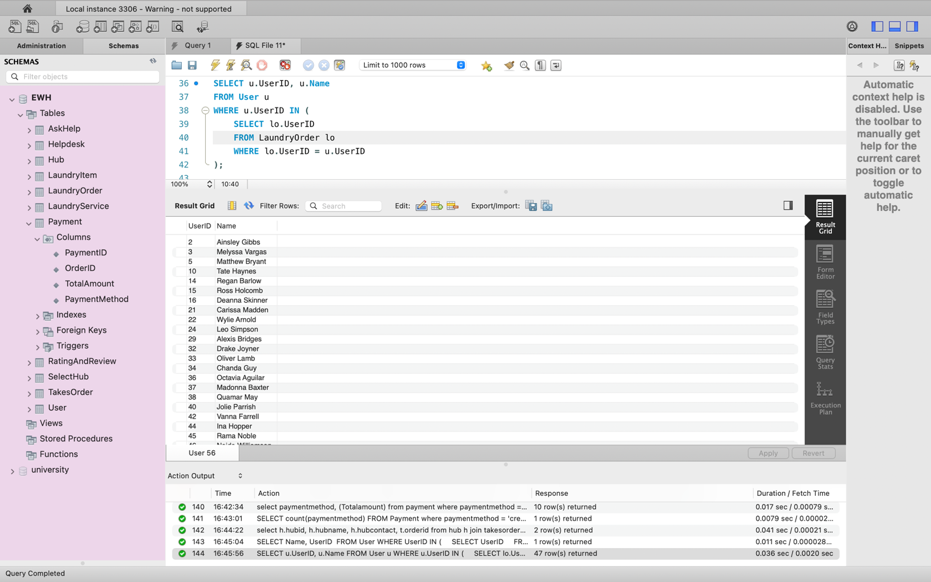
WHERE u.UserID IN (​

    SELECT lo.UserID​

    FROM LaundryOrder lo​

    WHERE lo.UserID = u.UserID​

);​

​

**7. Select the OrderID, UserID, DeliveryDate, and DeliveryTime from the LaundryOrder table for orders marked as 'urgent'. ​**

SELECT lo.OrderID, lo.UserID, lo.DeliveryDate, lo.DeliveryTime​

FROM LaundryOrder lo​

WHERE lo.Urgency = 'urgent'​

AND ( SELECT COUNT(\*)​

    FROM LaundryOrder lo2​

    WHERE lo2.OrderID = lo.OrderID​

    AND lo2.PickupDate = lo.DeliveryDate​

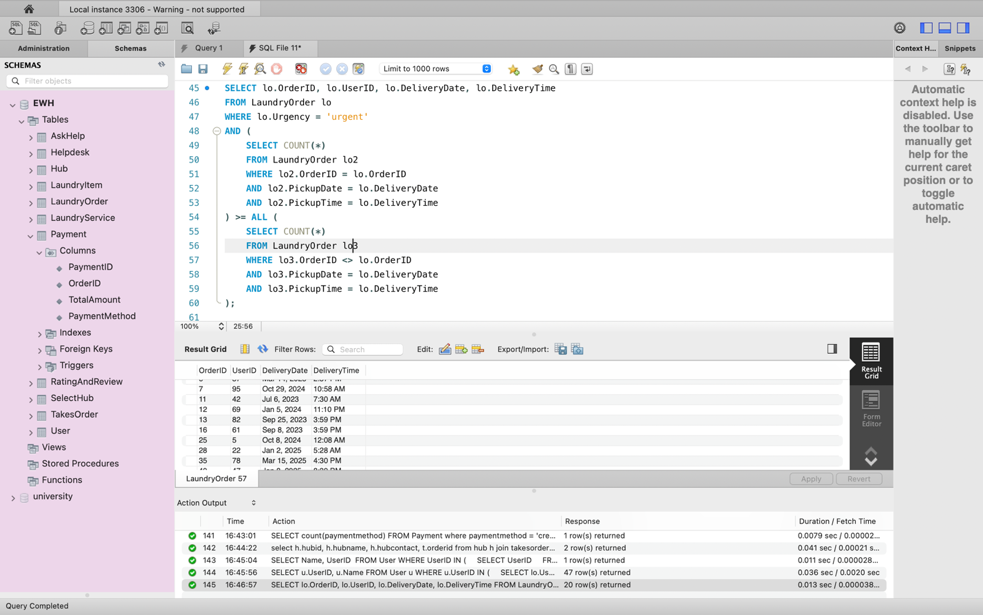
    AND lo2.PickupTime = lo.DeliveryTime​ ) >= ALL (​  SELECT COUNT(\*)​

    FROM LaundryOrder lo3​

    WHERE lo3.OrderID <> lo.OrderID​

    AND lo3.PickupDate = lo.DeliveryDate​

    AND lo3.PickupTime = lo.DeliveryTime​ );​

​

**8. Select UserID and Name for users who have completed laundry orders but have not taken any orders.​**

SELECT u.UserID, u.Name​

FROM User u​

WHERE EXISTS (​

    SELECT 1​

    FROM LaundryOrder lo​

    WHERE lo.UserID = u.UserID​

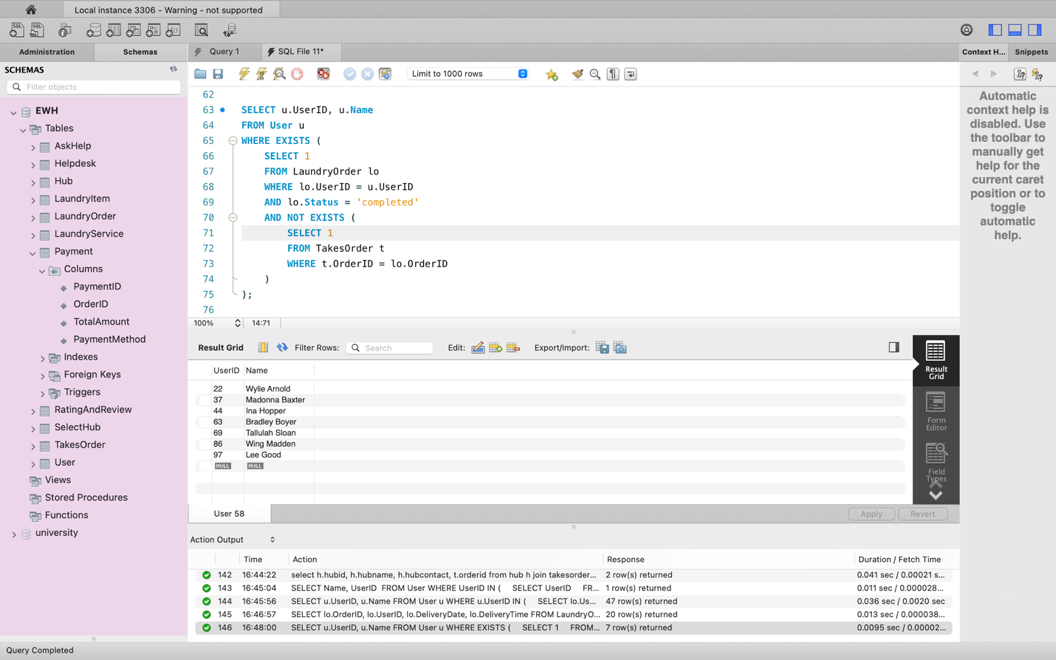
    AND lo.Status = 'completed'​

    AND NOT EXISTS (​

        SELECT 1​

        FROM TakesOrder t​

        WHERE t.OrderID = lo.OrderID));​



​

**9. Select UserID and Name of users who have placed laundry orders or requested help from the helpdesk.​**

SELECT u.UserID, u.Name​

FROM user u​

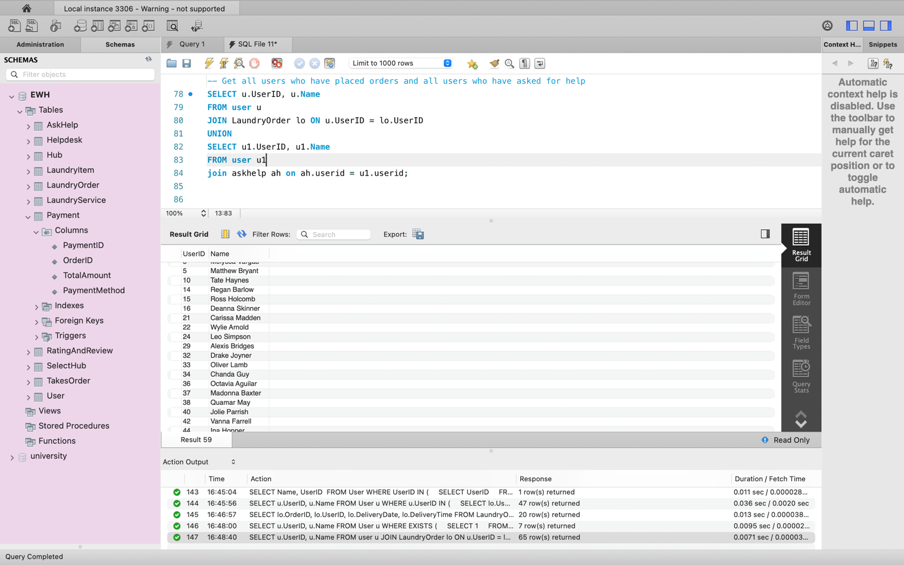
JOIN LaundryOrder lo ON u.UserID = lo.UserID​

UNION​

SELECT u1.UserID, u1.Name ​

FROM user u1​

join askhelp ah on ah.userid = u1.userid;​



​**10. Select UserID, Name, OrderCount, and AverageCost of users who have placed laundry orders.​**

SELECT u.UserID, u.Name, subquery.OrderCount, subquery.AverageCost​

FROM User u​

JOIN (​

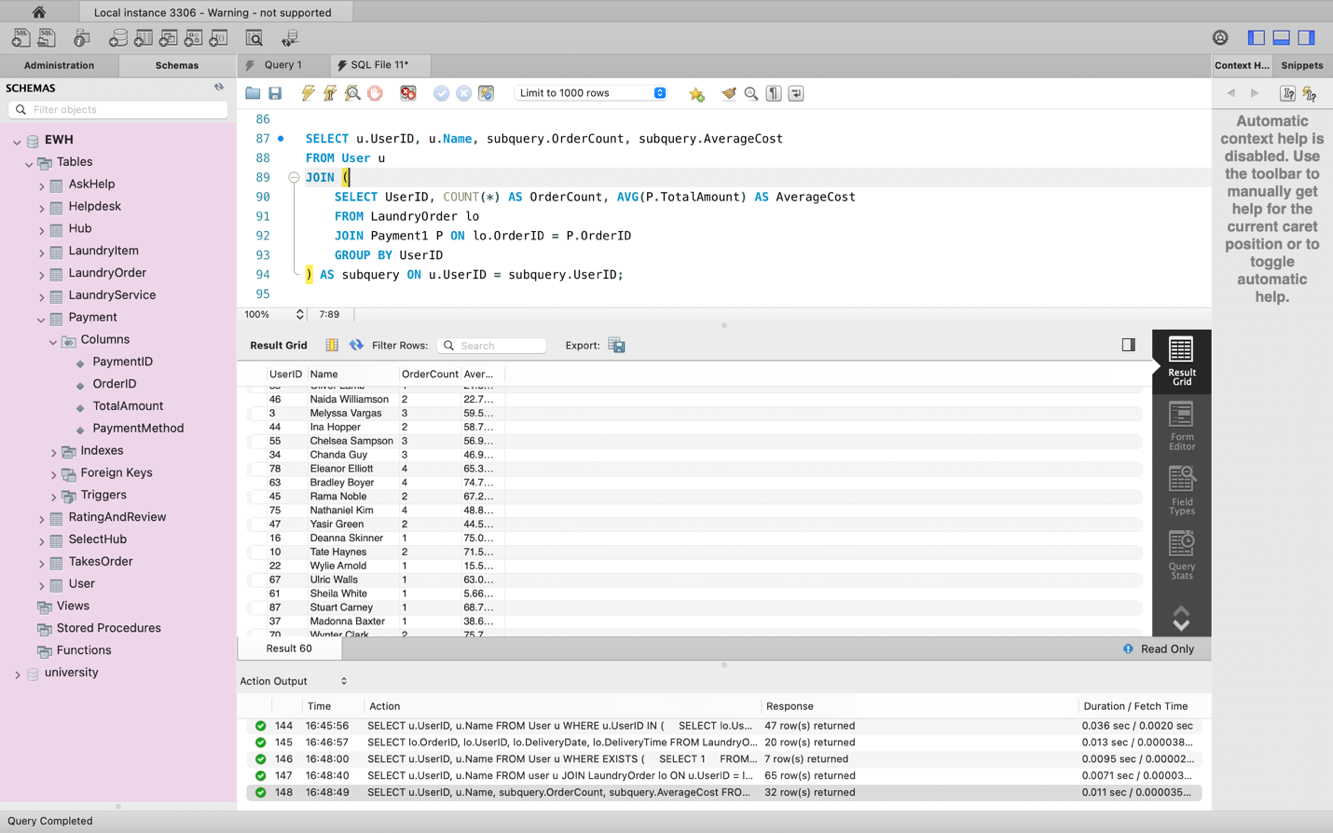
    SELECT UserID, COUNT(\*) AS OrderCount, AVG(P.TotalAmount) AS AverageCost​

    FROM LaundryOrder lo​

    JOIN Payment1 P ON lo.OrderID = P.OrderID​

    GROUP BY UserID​

) AS subquery ON u.UserID = subquery.UserID;​

****

**Implementation in Python**

import mysql.connector

#

from mysql.connector import Error

#

try:

connection = mysql.connector.connect(host='localhost',

database='ehubwash',

user='root',

password='root',

auth\_plugin = 'mysql\_native\_password')

if connection.is\_connected():

db\_Info = connection.get\_server\_info()

print("Connected to MySQL Server version ", db\_Info)

cursor = connection.cursor()

cursor.execute("select database();")

record = cursor.fetchone()

print("Your connected to database: ", record)

#

sql\_select\_Query = "select Name from user;"

cursor = connection.cursor()

cursor.execute(sql\_select\_Query)

records = cursor.fetchall()

print("Names of all the user :\n")

for row in records:

print('Names =',row[0],"\n")

#

except Error as e:

print("Error while connecting to MySQL", e)

finally:

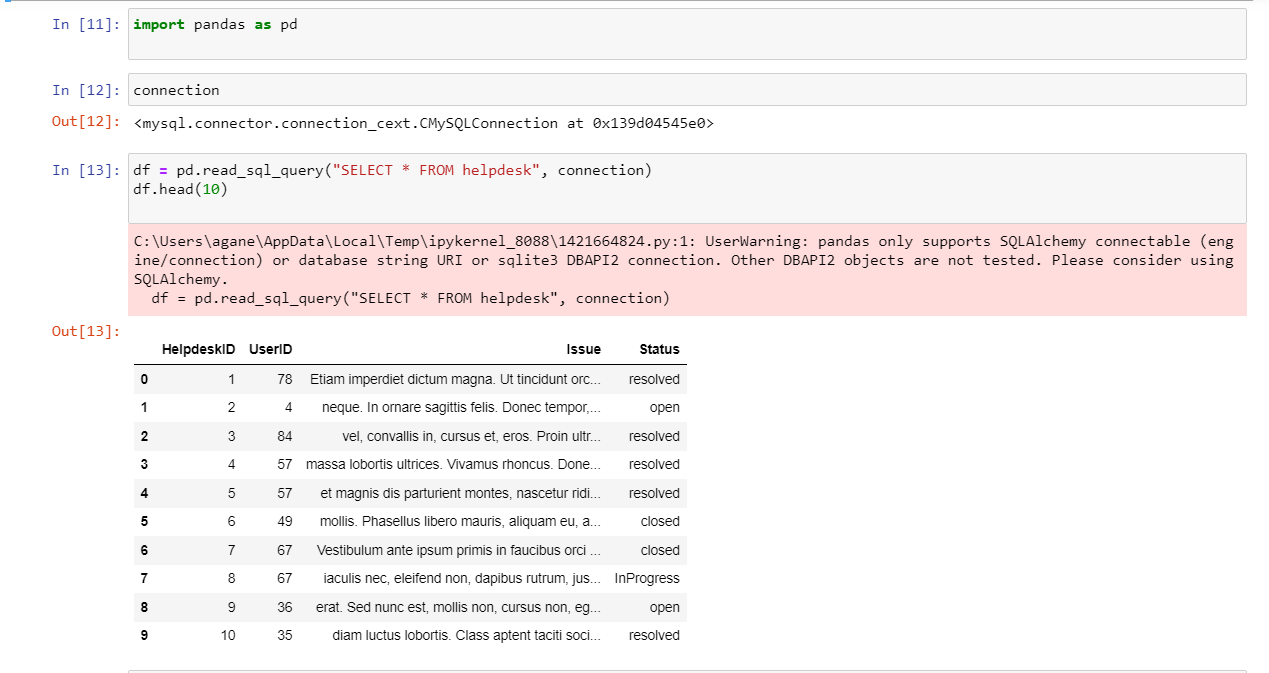
if (connection.is\_connected()):

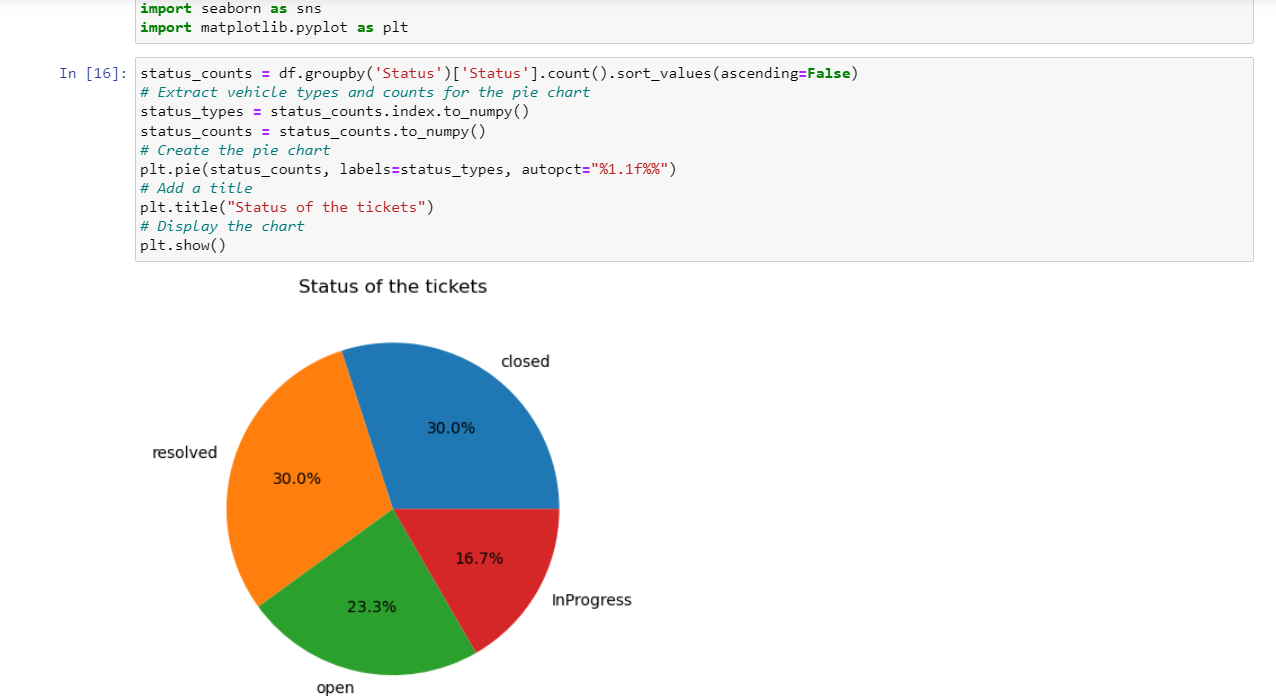
#cursor.close()

#connection.close()

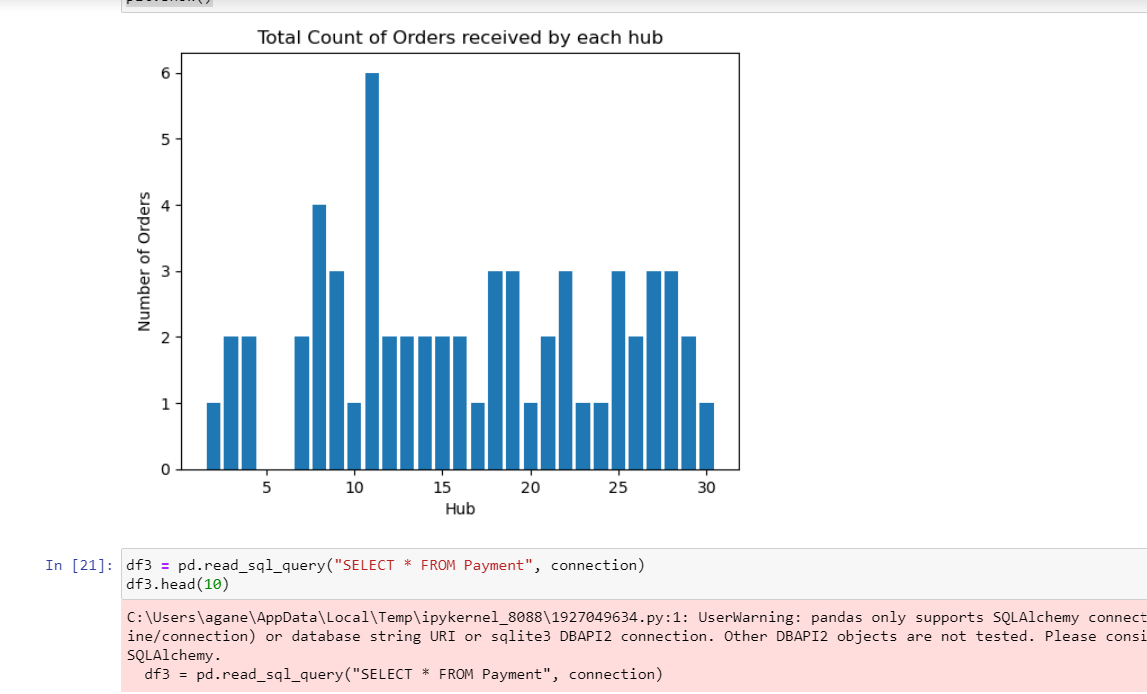
print("MySQL connection is connected")

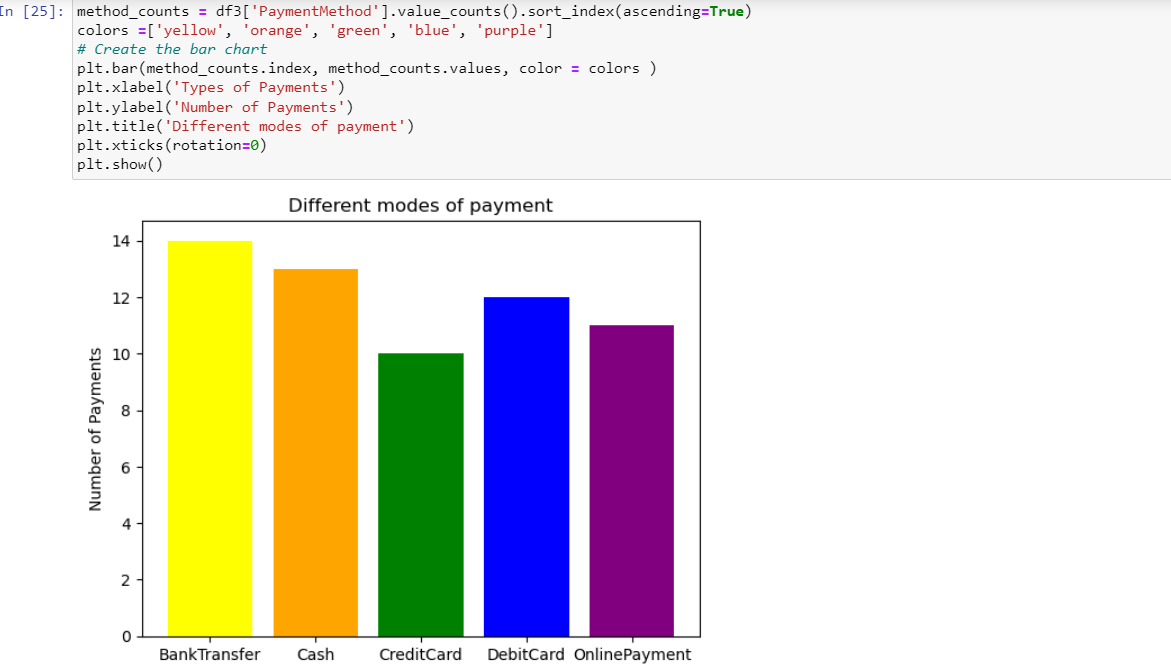
****

****

****

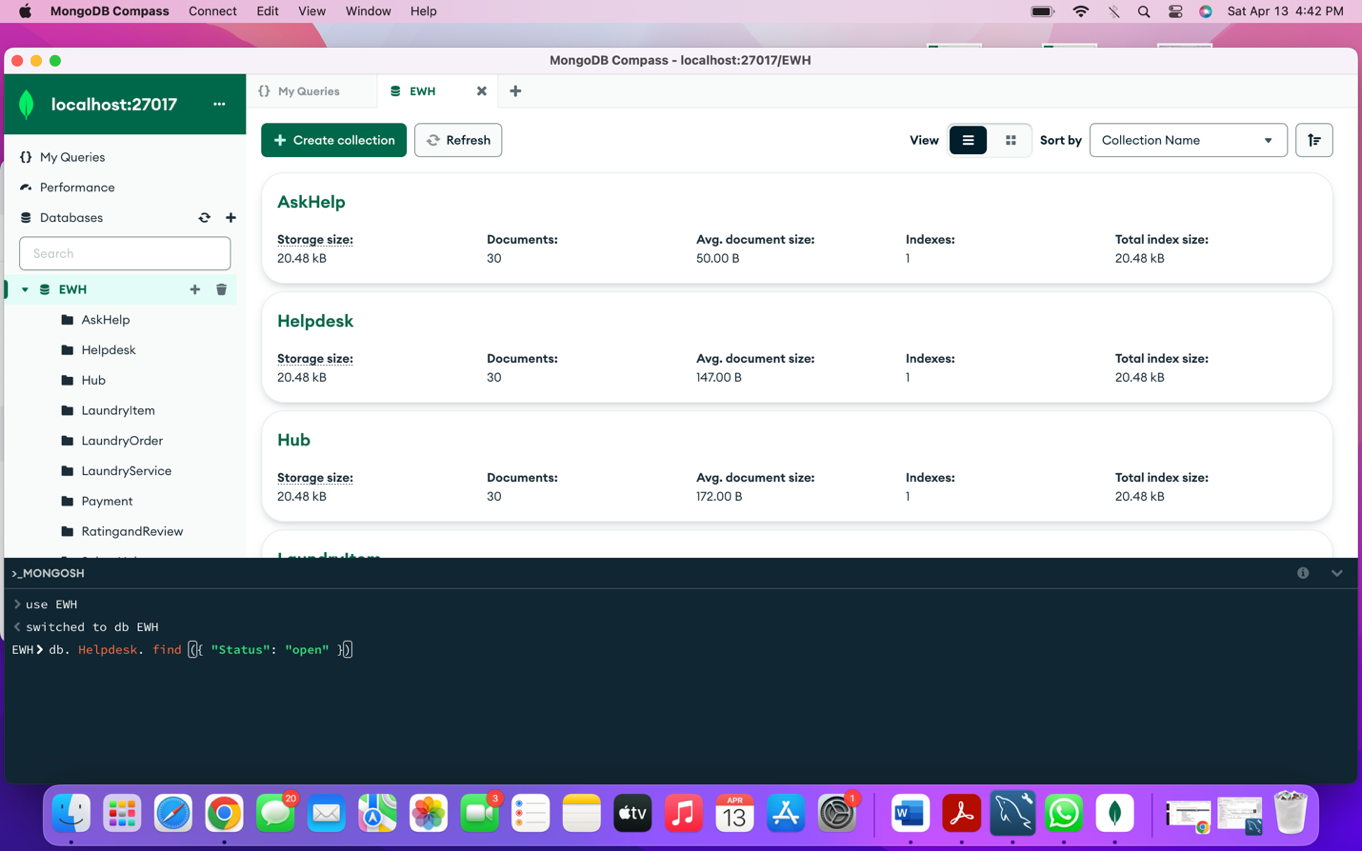
****

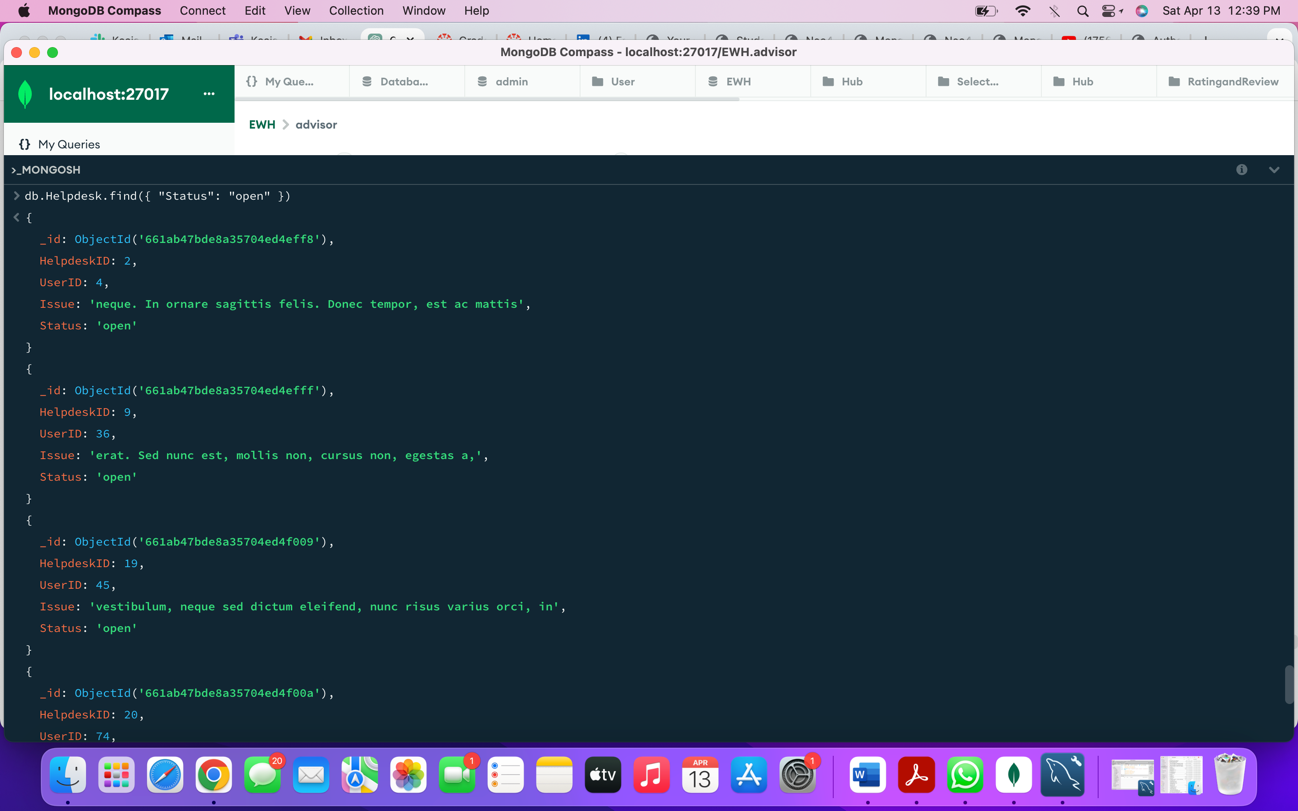
****

****

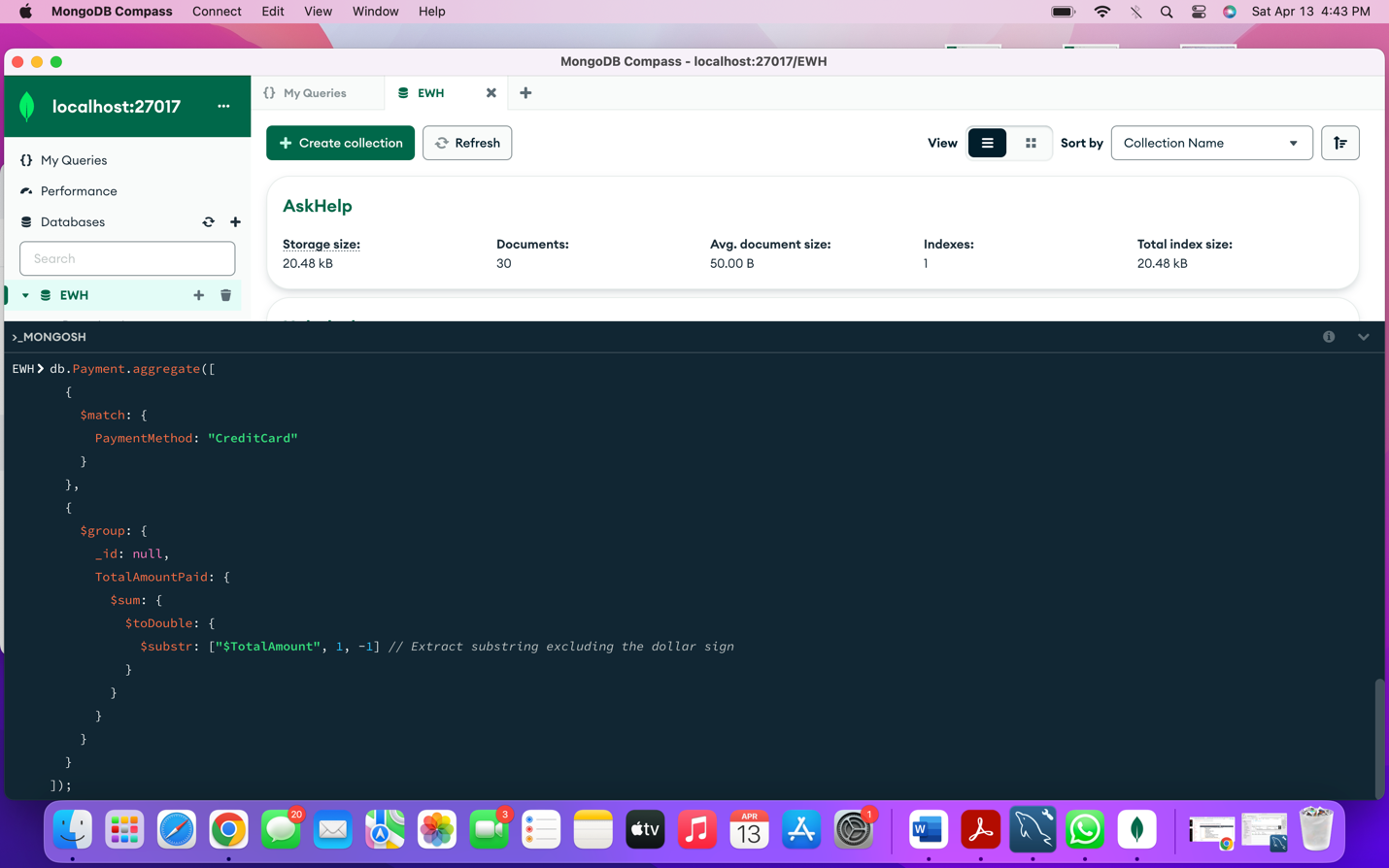
**Implementation in NoSQL**

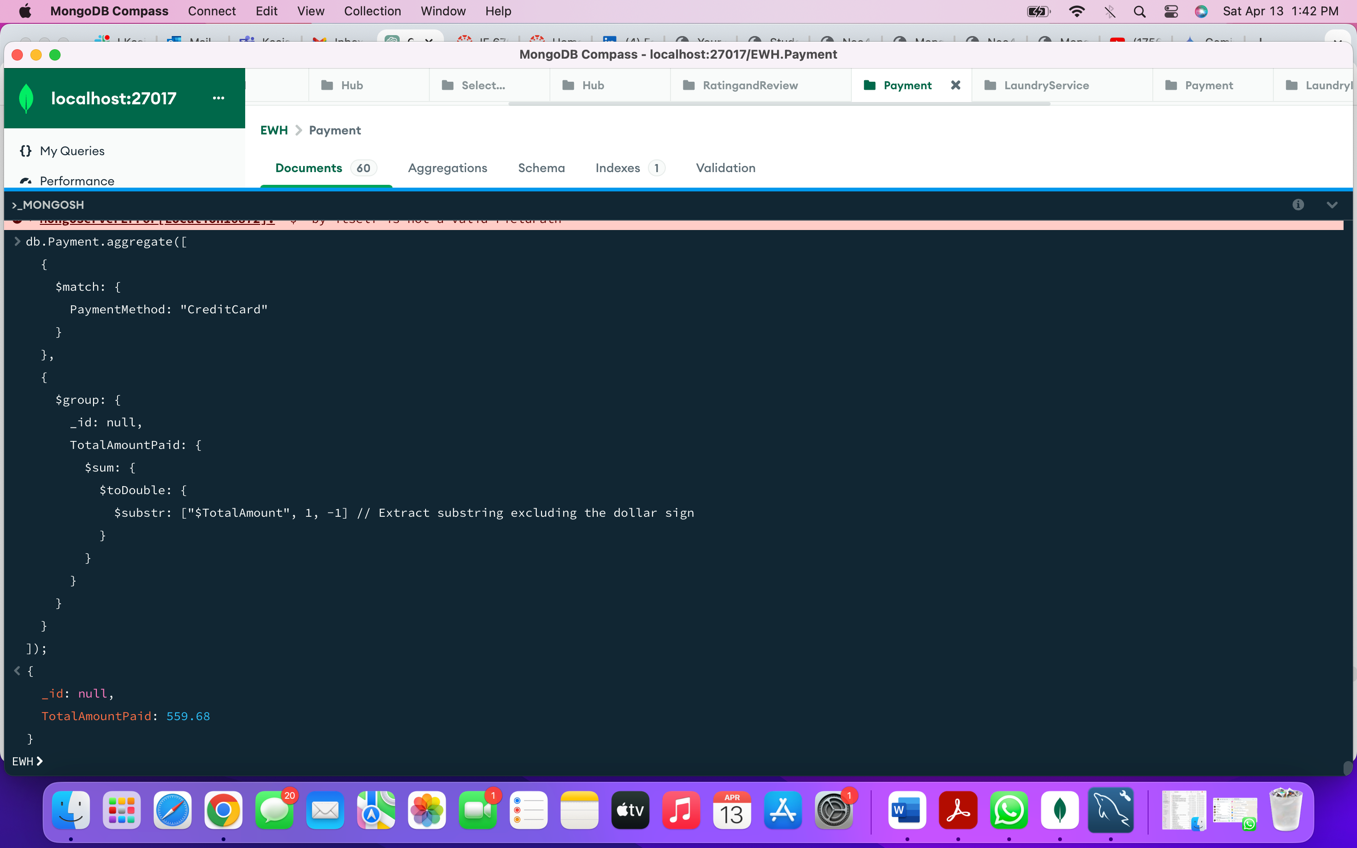
1. **Write a query to find out the open status of ticket in the Helpdesk.**

****

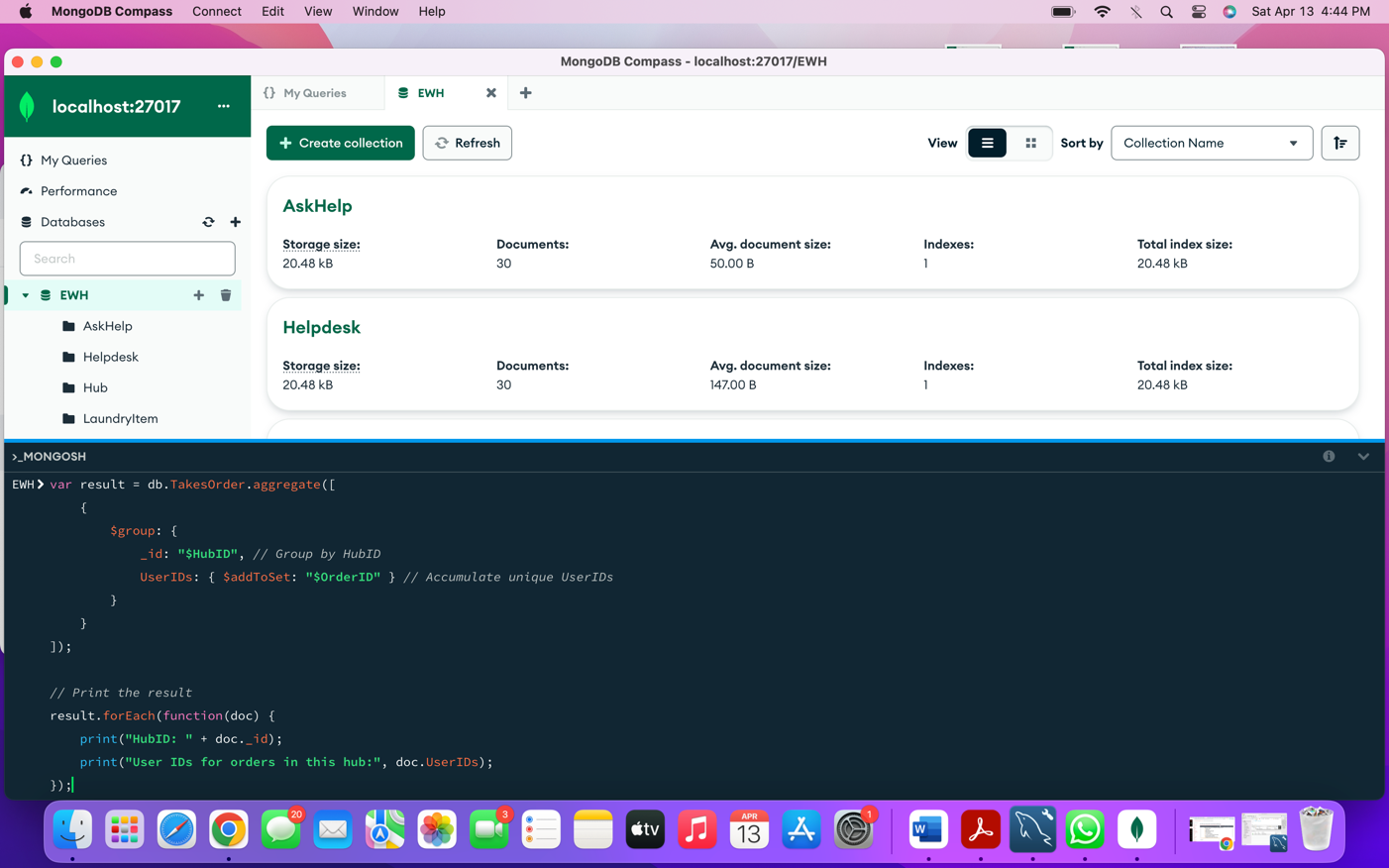
****

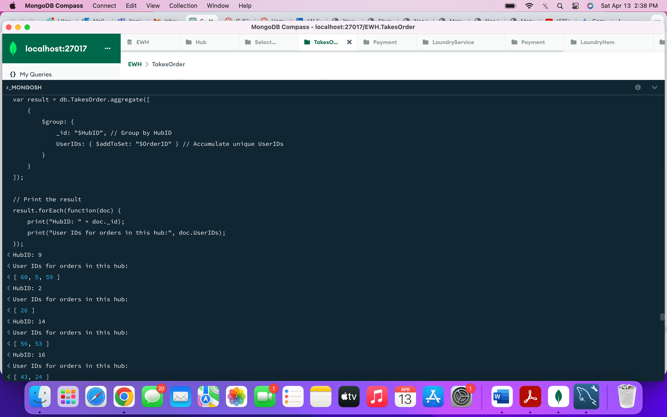
1. **Write a query to calculate the Total Amount paid using Credit Card.**

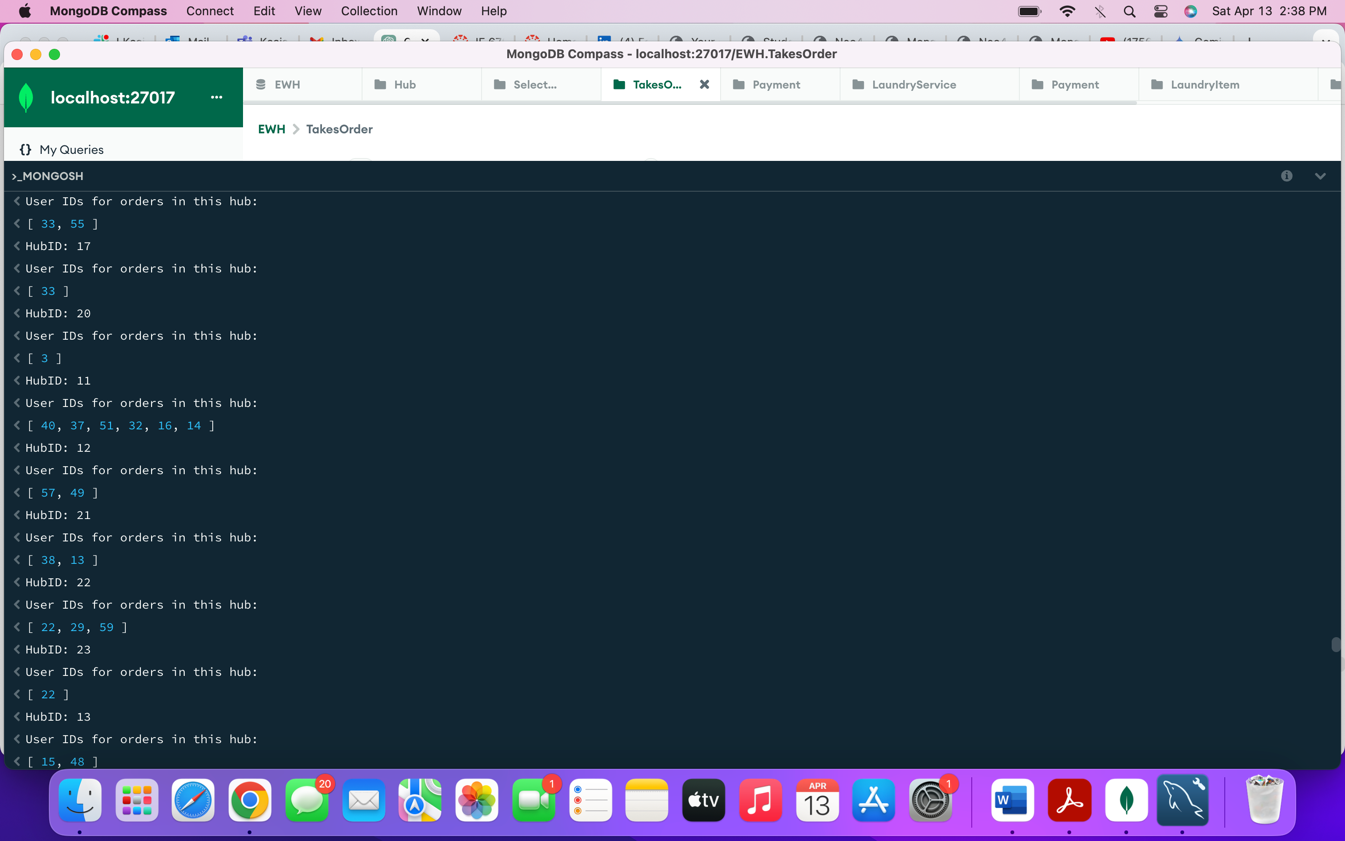
****

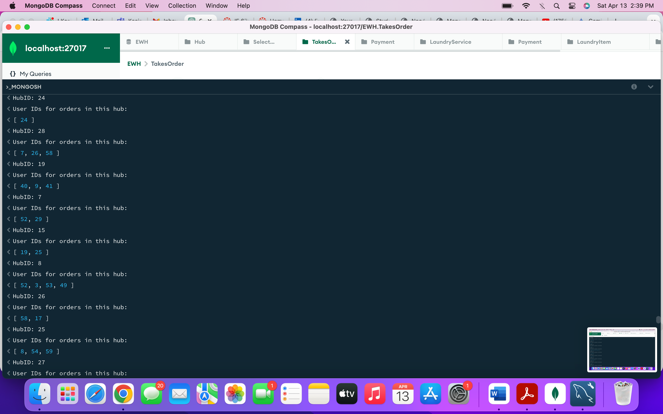
****

1. **Write to query to print User ID’s who has ordered from the same Hub.**

****

****

****

****

**Summary:**

1. Learned to implement the backbone of the application which is database.
2. It was hard in the beginning, we considered to store ever small attribute(details) but it was not possible we had to understand and prioritize the attributes which plays important role
3. First step was conceptual model, we designed our concepts into EER model which felt great achievement
4. Second step converting it to UML helped us understand what function each table has to perform, it made us thought more about the process.
5. In the third step we understood the importance of normalization. It helps us to connect table in more efficient way.
6. While implementing SQL, we realized how important are the first three steps, those are like foundation of database even a single relationship missing can mess up the entire DB.
7. Python Implementation gives the sense how the application runs and how we can draw different insights from the data extracted from DB.
8. NoSQL helped to understand how we can extract complex relational SQL query with ease using NoSQL.