# IST769 Homework Submission #9

## Basic Information

Your Name: Thulasi Ram Ruppa Krishnan   
Your SUID: 460746269  
Your Email: truppakr@syr.edu  
Date Due: 05-28-2020   
Homework #: HW9

## Instructions

For each answer, please include your answer as text, and any screenshot(s) which demonstrate your answer was executed. Most importantly, make sure to include evidence your answer is correct. This will most likely be a screenshot. If you had issues, problems, or had to make assumptions include them in your answer.

# Exercises

Complete each of the following exercises. If you are unsure how to accomplish the task, please consult the coursework videos where there are explanations and demos.

1. Design your own scenario for which a Cassandra table would be a good solution. Make sure to explain the scenario and the specific characteristics of the scenario which would make Cassandra a good fit. Make sure to follow a query first approach and justify how the partition and cluster keys should be setup.

**Scenario**

Let’s assume an online shopping site where a shopping cart (order) is created by each user and it has all the items (products) added by the user for purchase. A user will have many orders over a period. There will be large amount of orders placed by the users and it needs to be scaled horizontally for writes. As a user, I will be interested only in the orders that I have placed, and I don’t need to see the orders from others.

In this case, the user and the orders placed by each user is denormalized. A single user and the orders placed by that user should be maintained on the same node, so that the retrieval of information for a given user is faster. All user’s data can be distributed across multiple nodes to scale horizontally.

Partitioning

* Based on user
* User places one order at a time

Clustered Index

* Items added to an order will be in the sort order in which it is added

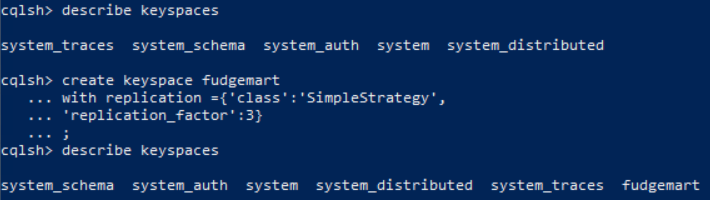
What to show to the User?

* User information
* Order Information
* Item Details
* All IDs to retrieve

Design

* user\_id (Partition Key)
* User Information: {email, name, address}
* order\_id
* item\_id
* Item Info: (item name, item price, item quantity, item amount)
* Order Date
* Order amount
* Last updated (Cluster Key)
* Primary Key (user\_id, last\_updated)

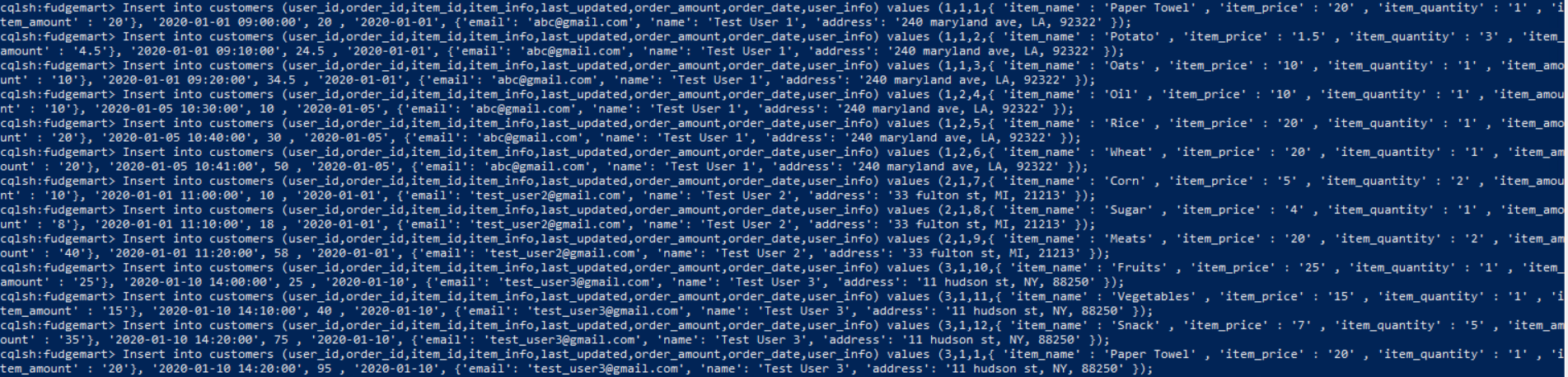
1. Create your Cassandra table in CQL based on your scenario from the previous exercise. You should define the columns and data types to suit your scenario in addition to configuring the partition and cluster keys.

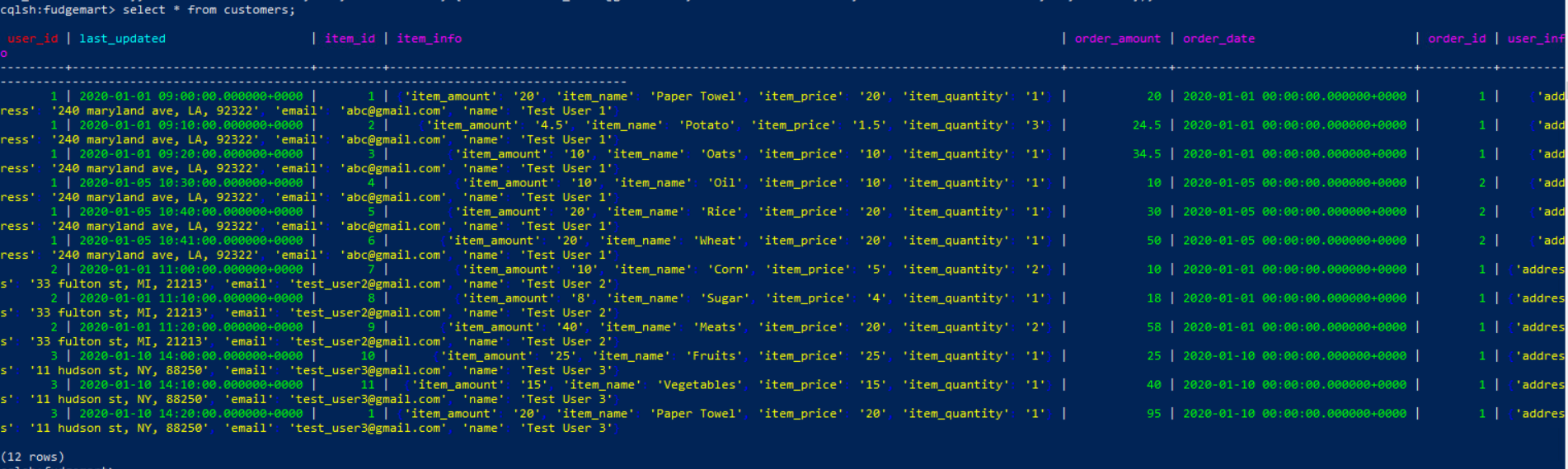




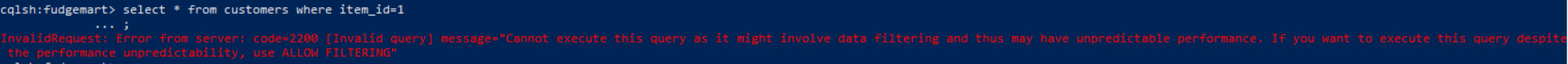


1. Write CQL statements to add data to your table. Add at least 9 records consisting of 3 different partition and cluster keys

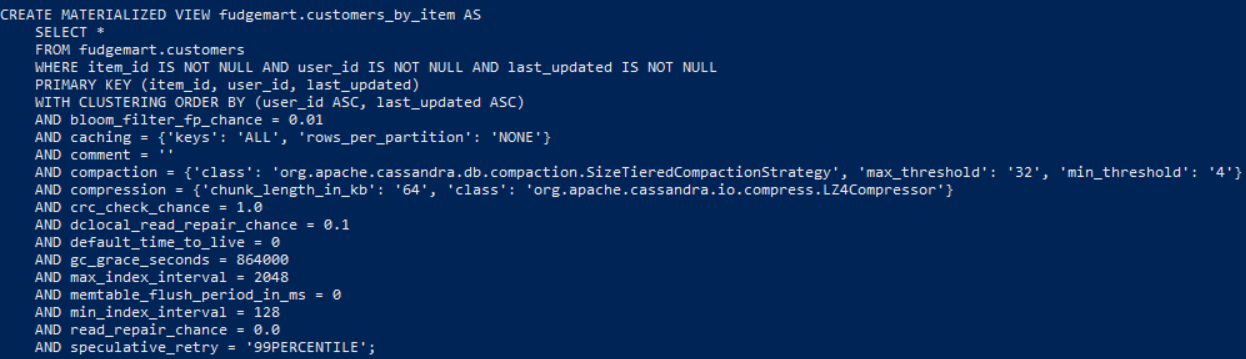


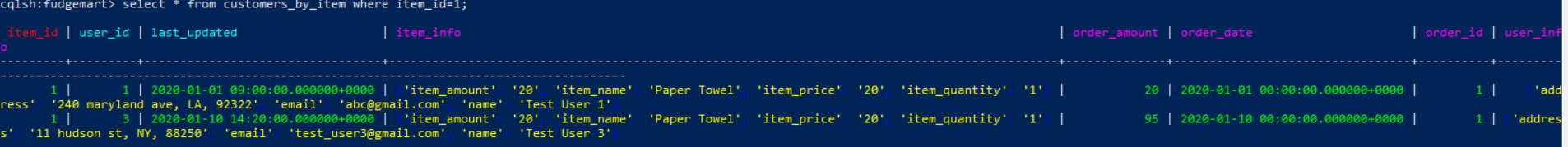


1. Write a CQL statement to create an index or materialized view on your table so that you can set a different partition key to prevent ALLOW FILTERING. Then write a CQL SELECT statement to demonstrate it works as designed.



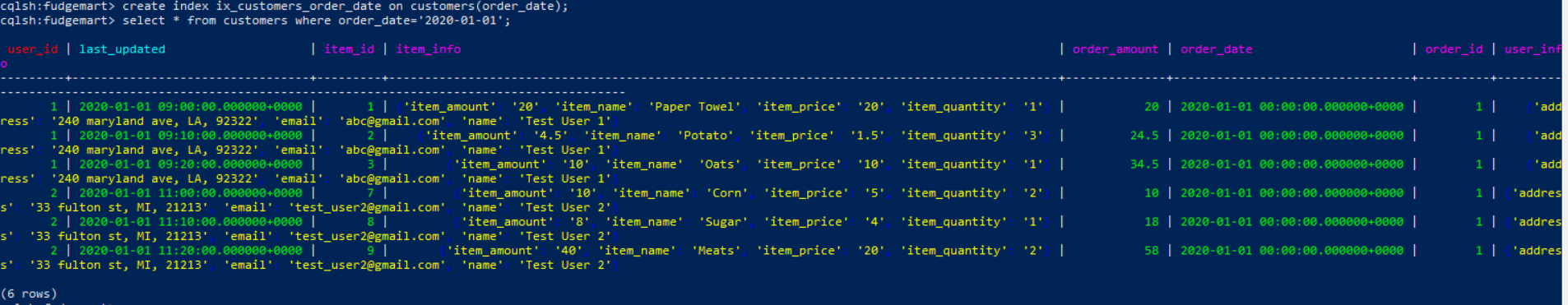


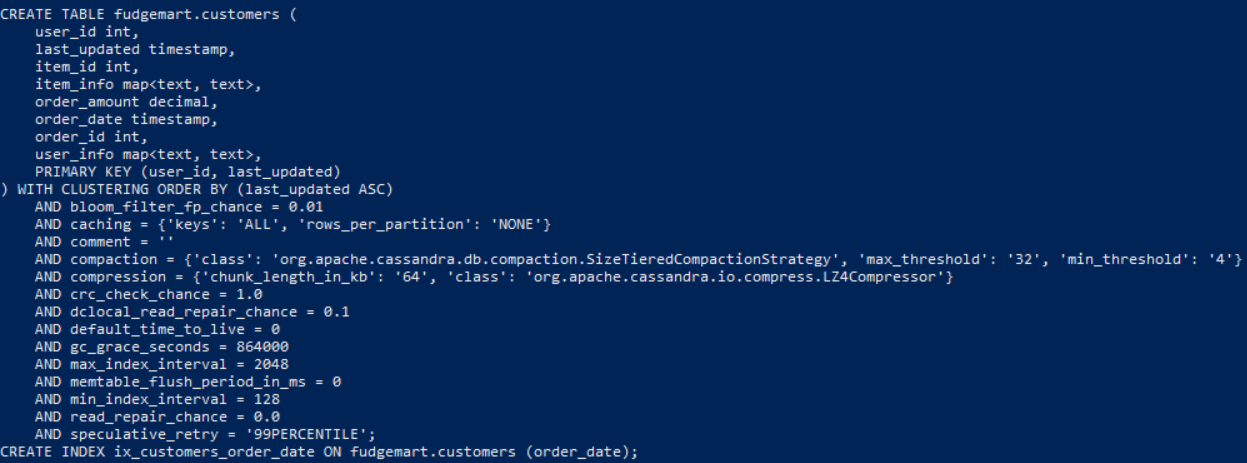




1. Write a CQL statement to create an index or materialized view on your table so that you can set a different cluster key to prevent ALLOW FILTERING. Then write a CQL SELECT statement to demonstrate it works as designed.







# Tear-Down

When you are finished with the homework you should stop the environment:

1. From the terminal window where you typed docker-compose up -d type in the following:  
   docker-compose stop