

Logic used for Assignment – Databases

Table of Contents

Logic used for Assignment – Databases.....	1
Task-01: Data Loading – Logic.....	2
Task-2: Queries – Logic	3
Query-01	3
Query-02	3
Query-03	3
Query-04	3
Query-05	4
APPENDIX – Sequence Diagram (Method Call Flow)	5
APPENDIX – AWS Screenshots	6

Task-01: Data Loading – Logic

DriverManager from Java SQL package is used to establish connection with the Upgrad AWS RDS MySQL Database URL, using provided credentials.

MongoDB Java driver package is used to create client instance for establishing connection to AWS MongoDB NoSQL database, created in AWS Cloud using shared AWS account.

[Refer Appendix for AWS screenshots for VPC, EC2, Security group configurations & mongo db installation for reference.]

Created mongoDB “pgcdata” with collections named as “products”.

Upon establishing connection with AWS RDS MySQL DB & AWS EC2 MongoDB, part of initialization steps, all the documents under “products” collection were deleted. Though this operation isn’t mentioned as an explicit requirement part of the assignment, this is done to ensure that query results remain consistent upon multiple executions of the assignment application.

Same cleanup operation shall be done as last operation of this assignment application. But since it is possible for the assignment evaluators to verify the mongoDB collection from console, currently this cleanup operation at end of application execution is disabled/commented.

Considering the limited scope of applications evaluation rubrics, following assumptions were made –

1. MySQL table will contain all the fields of type VARCHAR / Strings.
2. List of records shall not be huge in size, yet restricted fetch size to 10, considering given MySQL dataset tables were having only 5 records.
3. Havent employed any other 3rd party libraries for ResultSet to JSON to MongoDB Import based data migration. Used simple methods only.

Created static stub methods to query individual tables & refactored common methods for better code reuse. Below methods are created for same purpose, which are called from main() method of Driver class –

1. importRDSVarcharTablesToMongoDb() - Method to import multiple RDS SQL table with only VARCHAR (String) fields to Mongo DB, uses below method.
2. insertVarcharTableToMongoDb() - Method to import records fetched by given sql query on a RDS SQL table with only VARCHAR (String) fields by adding a new category field referring to product category to Mongo DB. This method currently fetches records by paging 10 records at once. This shall be set as per needs. Uses below method.
3. getDocumentList() - Method to construct Mongo Document object for each records. This method uses to fetch individual fields of each records as present part of returned upon querying SQL table with only VARCHAR/String fields.

Refer [Appendix – Sequence Diagram](#) for method call flow.

Task-2: Queries – Logic

Query-01: Show the complete product inventory with common attributes. (Print the data only for the common attributes which are there for all three product categories)

Logic:

1. Build Query to find all documents
2. add (optionally) projection to include only the common fields & get MongoClient.
3. Iterate cursor & Call PrintHelper.printSingleCommonAttributes() to display the common attributes on the Screen.

Additionally, usage of Projections here seems to be optional, as the PrintHelper.printSingleCommonAttributes() method is hardcoded to print the common attributes only, irrespective of whether other non-common / non-null columns are present or not part of the fetched result mongo cursor.

Query-02: Get the information about the first five products under the category 'Mobiles'.

Logic:

1. Build Query to find all documents with category as "Mobile"
2. Limit records to 5
3. get MongoClient on returned documents
4. Iterate cursor & call respective PrintHelper.printAllAttributes() method to display output as required.

Query-03: Get the products ordered by category without their auto-generated '_id' column in MongoDB.

Logic:

1. Build Query to find all documents
2. Use Projections to exclude ID field &
3. Sort documents by \$Category field in descending order
4. Get MongoClient on returned documents
5. Iterate cursor & call respective PrintHelper.printAllAttributes() method to display output as required.

Query-04: Get the product count for every category.

Logic:

1. Build Query to aggregate documents, by grouping documents by "\$Category"

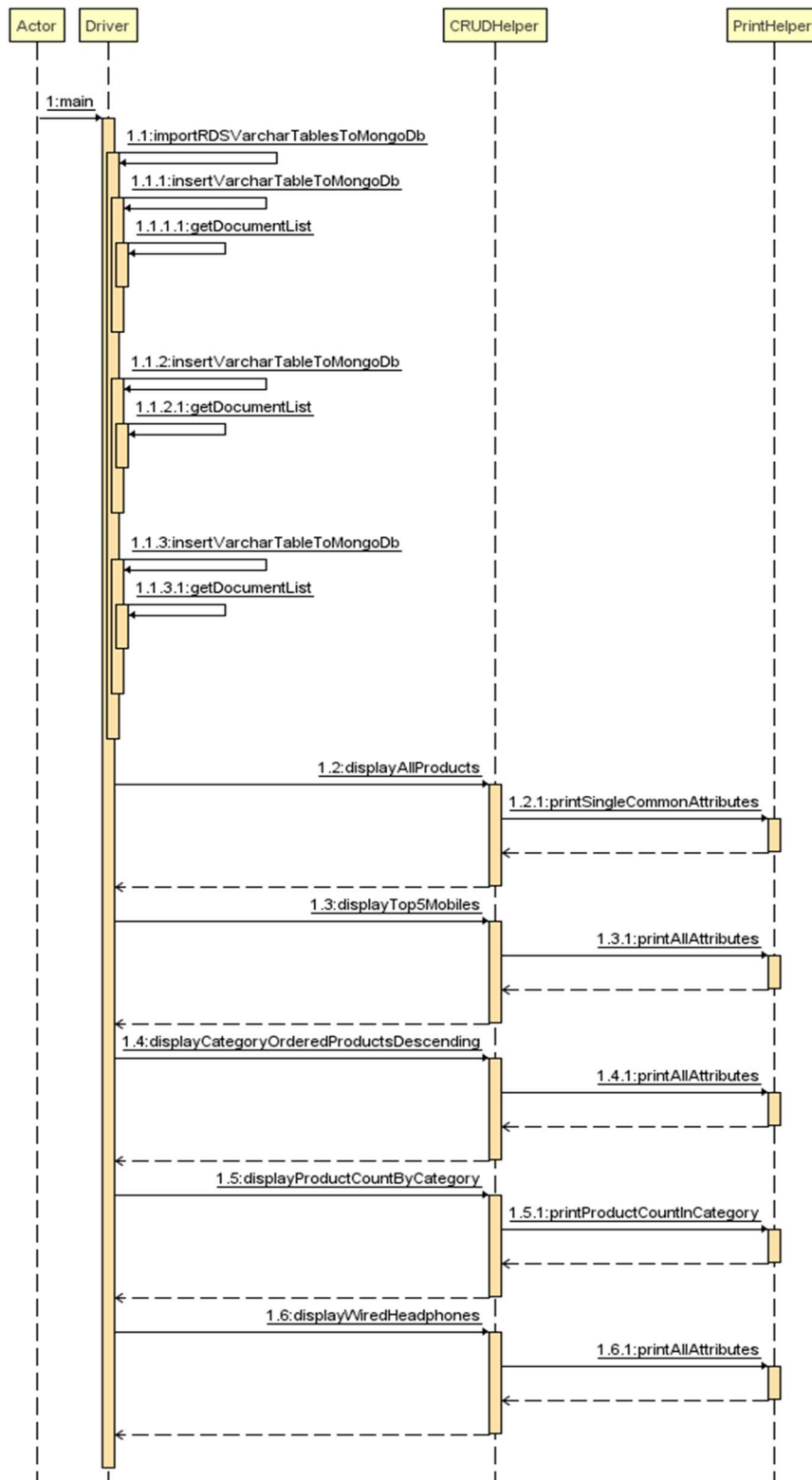
2. Accumulate the number of records using Accumulator.sum to keep adding 1 for every record grouped by "\$Category" field
3. Sort the list & get MongoClient on returned documents
4. Iterate cursor & call respective PrintHelper.printProductCountInCategory() method to display output as required.

Query-05: Get the information about all the wired headphones.

Logic:

1. Create BasicDBObject to define criterias, such that "Category" key contains value "Headphone" & "ConnectorType" key contains value "Wired"
2. Build query to find all documents matching criteria set by BasicDBObject
3. Get MongoClient on returned documents.
4. Iterate cursor & call respective PrintHelper.printAllAttributes() method to display output as required.

APPENDIX – Sequence Diagram (Method Call Flow)



APPENDIX – AWS Screenshots

1. VPC Creation

Subscription Details | Nuvepro x VPC Management Console x +

console.aws.amazon.com/vpc/home?region=us-east-1#wizardFullpagePublicOnly

aws Services Search for services, features, marketplace products, and docs [Alt+S]

upgradmuthukuma @ 9638-1310-0165 N Virginia Support

Step 2: VPC with a Single Public Subnet

IPv4 CIDR block:* 10.0.0.0/16 (65531 IP addresses available)

IPv6 CIDR block: ☒ No IPv6 CIDR Block
☐ Amazon provided IPv6 CIDR block
☐ IPv6 CIDR block owned by me

VPC name: muthu_upgrad_assignment

Public subnet's IPv4 CIDR:* 10.0.0.0/24 (251 IP addresses available)

Availability Zone:* No Preference

Subnet name: Public subnet

You can add more subnets after AWS creates the VPC.

Service endpoints

Add Endpoint

Enable DNS hostnames:* ☒ Yes ☐ No

Hardware tenancy:* Default

66% Enabling DNS...

Cancel and Exit Back Create VPC

Feedback English (US) © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

2. Configuring Security Group

Subscription Details | Nuvepro

VPC Management Console

EC2 Management Console

console.aws.amazon.com/vpc/home?region=us-east-1#CreateSecurityGroup:

Services

Search for services, features, marketplace products, and docs

upgradmuthukuma @ 8638-1310-0165

N. Virginia

Support

VPC > Security Groups > Create security group

Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name [Info](#)
muthu_assignment_mongo_db_security_group
Name cannot be edited after creation.

Description [Info](#)
muthu_assignment_mongo_db_security_group

VPC [Info](#)
vpc-0dd6826407e599c9c (muthu_upgrad_assignment_vpc)

Inbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
All TCP	TCP	0 - 65535	My IP <input type="text"/> 49.37.160.130/32	For all inbound	Delete
Custom TCP	TCP	27017	My IP <input type="text"/> 49.37.160.130/32	mongo connection port	Delete
<div>Add rule</div>					

Feedback

English (US)

© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use

Cookie preferences

3. Creating EC2

The screenshot shows the AWS Management Console interface for creating an EC2 instance. The page is titled "Step 7: Review Instance Launch" and includes a progress bar at the top with steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review.

AMI Details

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0533f2ba8a1995cf9

Free tier eligible

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is a...

Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

Security Groups

Security Group ID	Name	Description
sg-0cd10f01b14721bf6	muthu_assignment_mongo_db_security_group	muthu_assignment_mongo_db_security_group

All selected security groups inbound rules

Type	Protocol	Port Range	Source	Description
All TCP	TCP	0 - 65535	49.37.160.130/32	For all inbound
Custom TCP Rule	TCP	27017	49.37.160.130/32	mongo connection p...

Instance Details

Number of instances: 1

Purchasing option: On demand

Network: vpc-0dd6826407e599c9c

Subnet: subnet-0f812394811868ad4

[Cancel](#) [Previous](#) [Launch](#)

Feedback English (US) © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

4. Connecting to EC2 from putty & setting up mongo DB

```
ec2-user@ip-10-0-0-43:~  
mongodb-org-tools.x86_64 0:4.4.4-1.amzn2  
  
Complete!  
[ec2-user@ip-10-0-0-43 ~]$ sudo vi /etc/mongod.conf  
[ec2-user@ip-10-0-0-43 ~]$ sudo service mongod start  
Redirecting to /bin/systemctl start mongod.service  
[ec2-user@ip-10-0-0-43 ~]$ mongo  
MongoDB shell version v4.4.4  
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb  
Implicit session: session { "id" : UUID("304828b6-5bda-44aa-ad4b-c76ff47da760") }  
MongoDB server version: 4.4.4  
Welcome to the MongoDB shell.  
For interactive help, type "help".  
For more comprehensive documentation, see  
    https://docs.mongodb.com/  
Questions? Try the MongoDB Developer Community Forums  
    https://community.mongodb.com  
---  
The server generated these startup warnings when booting:  
    2021-03-28T07:29:09.555+00:00: Access control is not enabled for the database.  
    Read and write access to data and configuration is unrestricted  
---  
---  
    Enable MongoDB's free cloud-based monitoring service, which will then receive and display  
    metrics about your deployment (disk utilization, CPU, operation statistics, etc).  
  
    The monitoring data will be available on a MongoDB website with a unique URL accessible to you  
    and anyone you share the URL with. MongoDB may use this information to make product  
    improvements and to suggest MongoDB products and deployment options to you.  
  
    To enable free monitoring, run the following command: db.enableFreeMonitoring()  
    To permanently disable this reminder, run the following command: db.disableFreeMonitoring()  
---  
> show dbs  
admin    0.000GB  
config   0.000GB  
local    0.000GB  
>
```

5. Running EC2 instance with mongo DB installed.

Instance details | EC2 Management console

console.aws.amazon.com/ec2/v2/home?region=us-east-1#InstanceDetails:instanceId=i-092cf04d484a48f68

Services

Search for services, features, marketplace products, and docs

upgrademuthukuma@8658-1310-0165

N. Virginia

Support

New EC2 Experience

EC2 Dashboard

Events

Tags

Limits

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Scheduled Instances

Capacity Reservations

Images

AMIs

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

Load Balancing

Load Balancers

EC2 > Instances > i-092cf04d484a48f68

Instance summary for i-092cf04d484a48f68 (muthu_upgrad_assignment_mongo_db_ec2)

Updated less than a minute ago

Instance ID

i-092cf04d484a48f68 (muthu_upgrad_assignment_mongo_db_ec2)

Instance state

Running

Instance type

t2.micro

AWS Compute Optimizer finding

Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Public IPv4 address

54.237.144.181 | open address

Public IPv4 DNS

ec2-34-237-144-181.compute-1.amazonaws.com | open address

Elastic IP addresses

-

IAM Role

-

Private IPv4 addresses

10.0.0.43

Private IPv4 DNS

ip-10-0-0-43.ec2.internal

VPC ID

vpc-0dd6826407e599c9c (muthu_upgrad_assignment_vpc) |

Subnet ID

subnet-0f812394811868ed4 (Public subnet) |

Details

Security

Networking

Storage

Status checks

Monitoring

Tags

▼ Security details

IAM Role

-

Owner ID

865813100165

Launch time

Mon Mar 29 2021 10:30:17 GMT+0530 (India Standard Time)

Security groups

sg-0cd10f01b14721bf6 (muthu_assignment_mongo_db_security_group) |

▼ Inbound rules

Filter rules

< 1 >

Port range	Protocol	Source	Security groups
All	TCP	49.37.167.118/32	muthu_assignment_mongo_db_security_group
27017	TCP	49.37.167.118/32	muthu_assignment_mongo_db_security_group

▼ Outbound rules

Feedback

English (US)

© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use

Cookie preferences