CS6650 Assignment3

1. Github repo url:

https://github.com/IreneZ723/CS6650Assignment3

2. Server Design:

2.1 LikeServlet:

This servlet handles HTTP POST requests related to album likes or dislikes. It is mapped to the "/review/*" URL pattern. The servlet uses RabbitMQ to send messages indicating album likes or dislikes. Messages are sent to the "review" queue.

Methods:

- init(): Initializes the servlet by creating a RabbitMQ connection factory.
- doPost(HttpServletRequest request, HttpServletResponse response): Handles HTTP POST requests, extracting like/dislike information from the request URL, updating the likes/dislikes, and sending a message to the RabbitMQ queue.
- performReview(String likeOrNot, String albumID): Implements the logic to update likes or dislikes for a given albumID.
- sendToQueue(String msg): Sends a message to the RabbitMQ queue.

2.2Consumer:

This class is responsible for consuming messages from the RabbitMQ queue and updating the database accordingly.

Methods:

- main(String[] argv): Initializes the necessary components, creates a pool of threads to consume messages, and shuts down the thread pool when done.
- consumeMessages(): Consumes messages from the RabbitMQ queue and updates the database.

Dependencies:

LikeDao: Handles database operations related to album likes and dislikes.
DBCPDataSource: Provides a DataSource for obtaining database connections.
ExecutorService: Manages a pool of threads for message consumption.

RabbitMQ Integration:

The consumer connects to the "review" queue in RabbitMQ.
Multiple threads are used to concurrently consume messages from the queue.
Each thread processes the received message by updating the database using LikeDao.

2.3 Relationships:

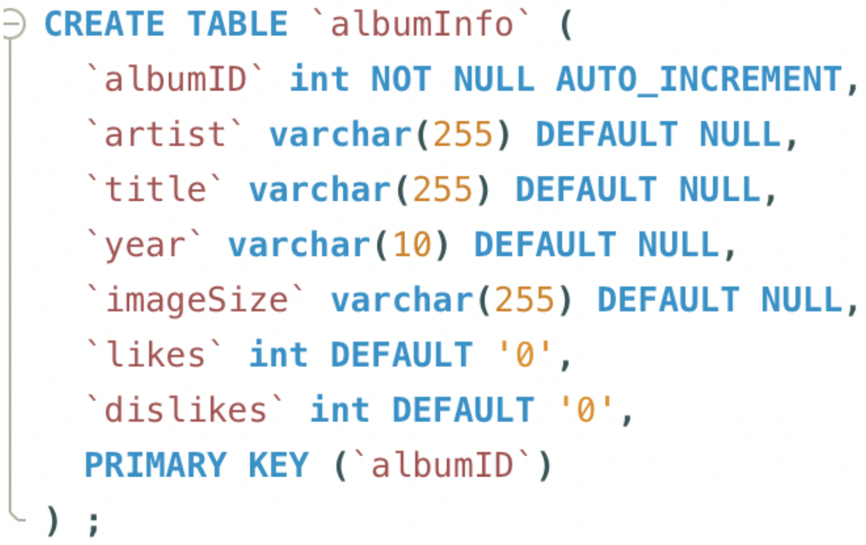
The LikeServlet and Consumer share a common RabbitMQ queue named "review".
When a user likes or dislikes an album through the servlet, a message is sent to the RabbitMQ queue.
The consumer, running in multiple threads, listens to the RabbitMQ queue, consumes messages, and updates the database accordingly using LikeDao.
The database connection is managed by DBCPDataSource.
Both applications use the RabbitMQ Java client library for interacting with RabbitMQ.
Exception handling is implemented for potential errors during initialization, message processing, and database operations.

Threading is employed in the consumer to achieve concurrent message processing.

2.4 Data Model Design

Add Two columns like and dislike to store the number of (dis)like received for particular albumID.

This create statement as follows:

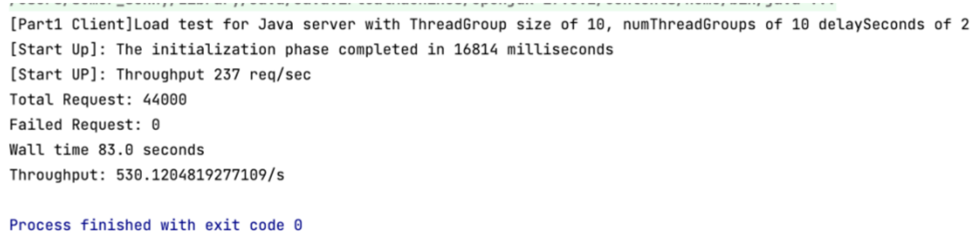


```
CREATE TABLE `albumInfo` (
  `albumID` int NOT NULL AUTO_INCREMENT,
  `artist` varchar(255) DEFAULT NULL,
  `title` varchar(255) DEFAULT NULL,
  `year` varchar(10) DEFAULT NULL,
  `imageSize` varchar(255) DEFAULT NULL,
  `likes` int DEFAULT '0',
  `dislikes` int DEFAULT '0',
  PRIMARY KEY (`albumID`)
) ;
```

3. Test result

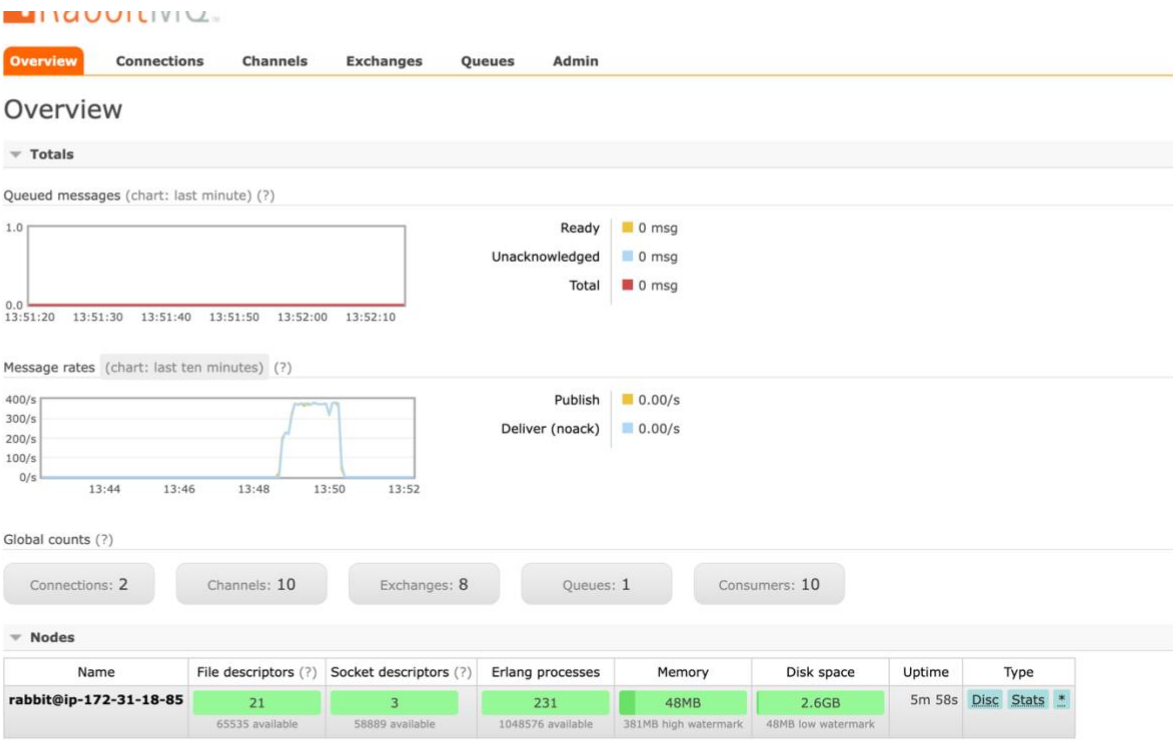
The throughputs are ‘roughly’ the same as assignment 2. In all tests, production_rate ≈ consumption_rate thus the queue length remains small. Also all message rate charts show an increase to a plateau like /¯¯¯¯\ and the plateau is less than a 1000.

3.1 threadGroupSize = 10, numThreadGroups = 10, delay = 2 consumer = 10



```
[Part1 Client]Load test for Java server with ThreadGroup size of 10, numThreadGroups of 10 delaySeconds of 2
[Start Up]: The initialization phase completed in 16814 milliseconds
[Start UP]: Throughput 237 req/sec
Total Request: 44000
Failed Request: 0
Wall time 83.0 seconds
Throughput: 530.1204819277109/s

Process finished with exit code 0
```



Overview

Totals

Queued messages (chart: last minute) (?)

Ready	🟨 0 msg
Unacknowledged	🟦 0 msg
Total	🟥 0 msg

Message rates (chart: last ten minutes) (?)

Publish	🟨 0.00/s
Deliver (noack)	🟦 0.00/s

Global counts (?)

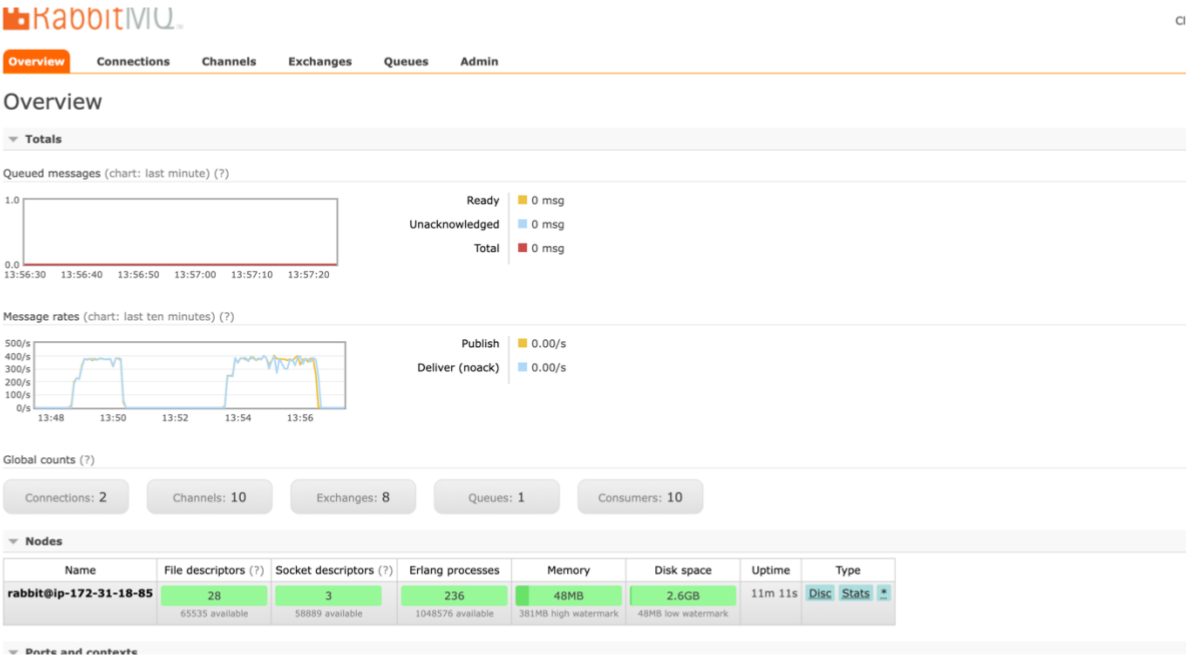
Connections: 2 Channels: 10 Exchanges: 8 Queues: 1 Consumers: 10

Nodes

Name	File descriptors (?)	Socket descriptors (?)	Erlang processes	Memory	Disk space	Uptime	Type
rabbit@ip-172-31-18-85	21 65535 available	3 58889 available	231 1048576 available	48MB 381MB high watermark	2.6GB 48MB low watermark	5m 58s	Disc Stats *

3.2 threadGroupSize = 10, numThreadGroups = 20, delay = 2 consumer = 10

```
[Part1 Client]Load test for Java server with ThreadGroup size of 10, numThreadGroups of 20 delaySeconds of 2
[Start Up]: The initialization phase completed in 13454 milliseconds
[Start UP]: Throughput 297 req/sec
Total Request: 84000
Failed Request: 0
Wall time 162.0 seconds
Throughput: 518.5185185185185/s
```



Overview

Totals

Queued messages (chart: last minute) (?)

Ready	🟨 0 msg
Unacknowledged	🟦 0 msg
Total	🟥 0 msg

Message rates (chart: last ten minutes) (?)

Publish	🟨 0.00/s
Deliver (noack)	🟦 0.00/s

Global counts (?)

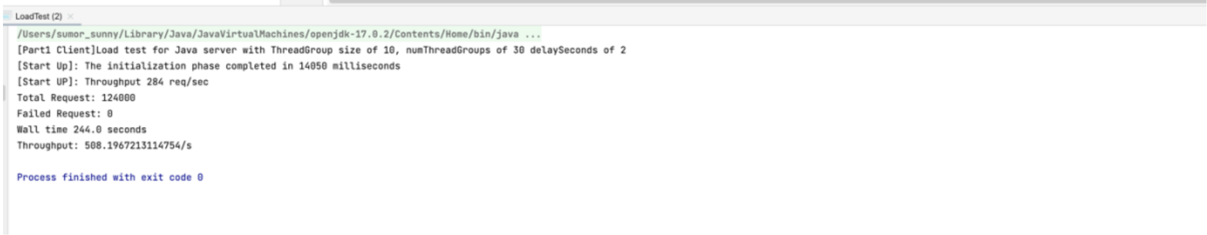
Connections: 2 Channels: 10 Exchanges: 8 Queues: 1 Consumers: 10

Nodes

Name	File descriptors (?)	Socket descriptors (?)	Erlang processes	Memory	Disk space	Uptime	Type
rabbit@ip-172-31-18-85	28 65535 available	3 58889 available	236 1048576 available	48MB 381MB high watermark	2.6GB 48MB low watermark	11m 11s	Disc Stats *

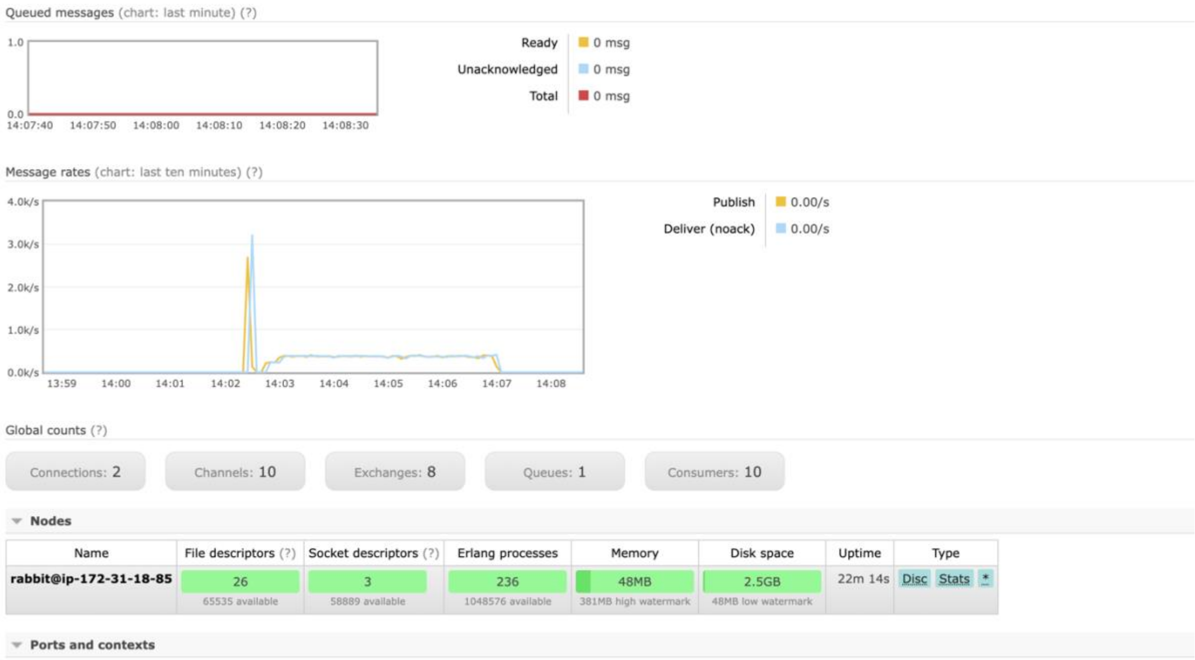
Ports and contexts

3.3 threadGroupSize = 10, numThreadGroups = 30, delay = 2 consumer = 10



```
/Users/sumor_sunny/Library/Java/JavaVirtualMachines/openjdk-17.0.2/Contents/Home/bin/java ...
[Part1 Client]Load test for Java server with ThreadGroup size of 10, numThreadGroups of 30 delaySeconds of 2
[Start Up]: The initialization phase completed in 14050 milliseconds
[Start UP]: Throughput 284 req/sec
Total Request: 124000
Failed Request: 0
Wall time 244.0 seconds
Throughput: 508.1967213114754/s

Process finished with exit code 0
```


Queued messages (chart: last minute) (?)

Ready	🟨 0 msg
Unacknowledged	🟦 0 msg
Total	🟥 0 msg

Message rates (chart: last ten minutes) (?)

Publish	🟨 0.00/s
Deliver (noack)	🟦 0.00/s

Global counts (?)

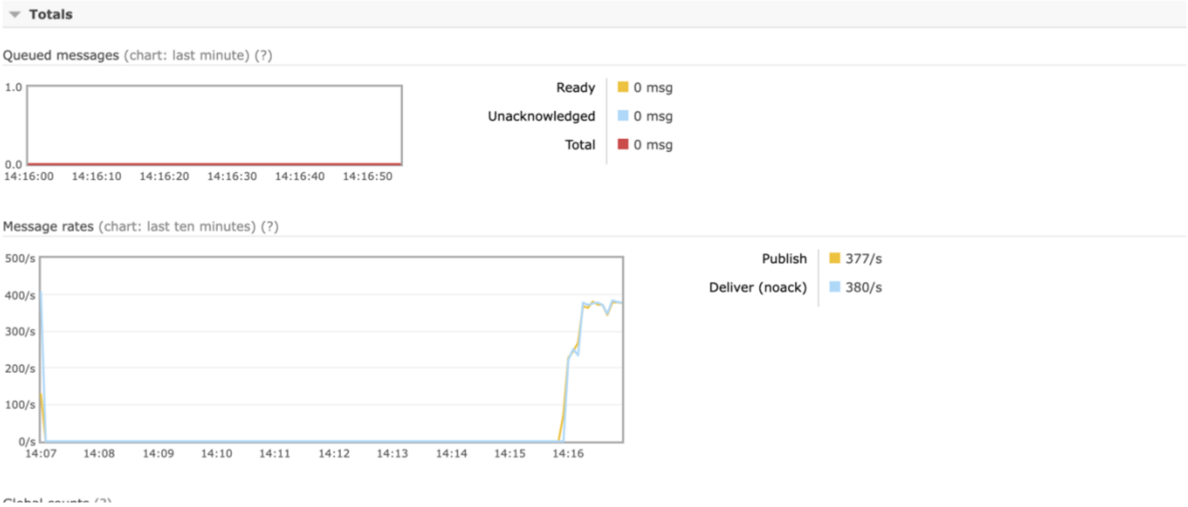
Connections: 2 | Channels: 10 | Exchanges: 8 | Queues: 1 | Consumers: 10

▾ Nodes

Name	File descriptors (?)	Socket descriptors (?)	Erlang processes	Memory	Disk space	Uptime	Type
rabbit@ip-172-31-18-85	26 65535 available	3 58889 available	236 1048576 available	48MB 381MB high watermark	2.5GB 48MB low watermark	22m 14s	Disc Stats *

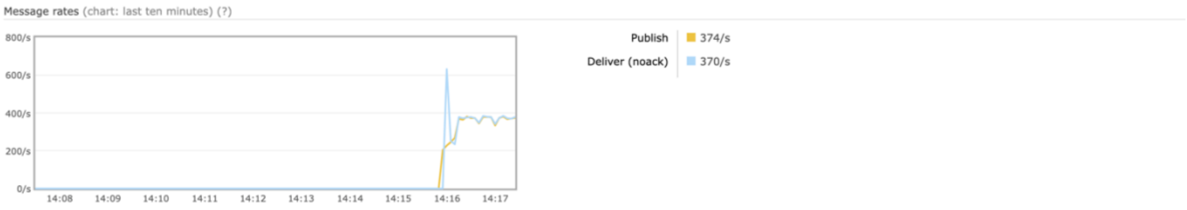
▾ Ports and contexts

3.4 threadGroupSize = 10, numThreadGroups = 10, delay = 2 consumer = 30

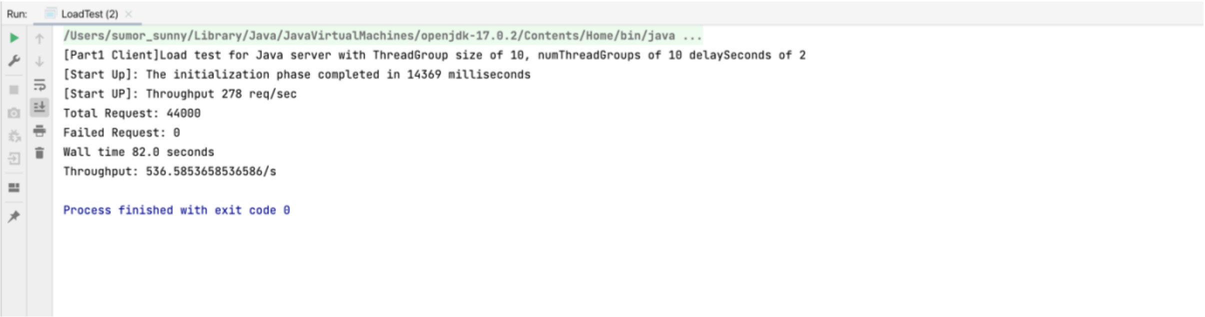


Ready	0 msg
Unacknowledged	0 msg
Total	0 msg

Publish	377/s
Deliver (noack)	380/s



Publish	374/s
Deliver (noack)	370/s

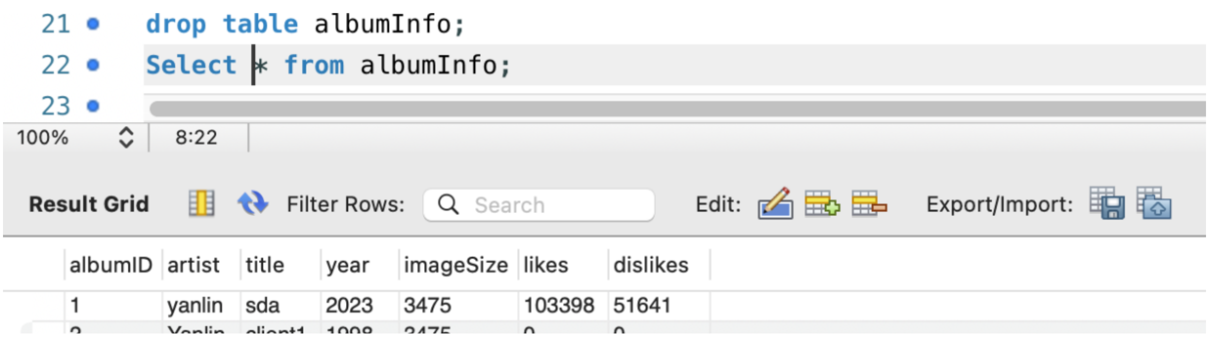


```
/Users/sumor_sunny/Library/Java/JavaVirtualMachines/openjdk-17.0.2/Contents/Home/bin/java ...
[Part1 Client]Load test for Java server with ThreadGroup size of 10, numThreadGroups of 10 delaySeconds of 2
[Start Up]: The initialization phase completed in 14369 milliseconds
[Start UP]: Throughput 278 req/sec
Total Request: 44000
Failed Request: 0
Wall time 82.0 seconds
Throughput: 536.5853658536586/s

Process finished with exit code 0
```

3.5 Database updated

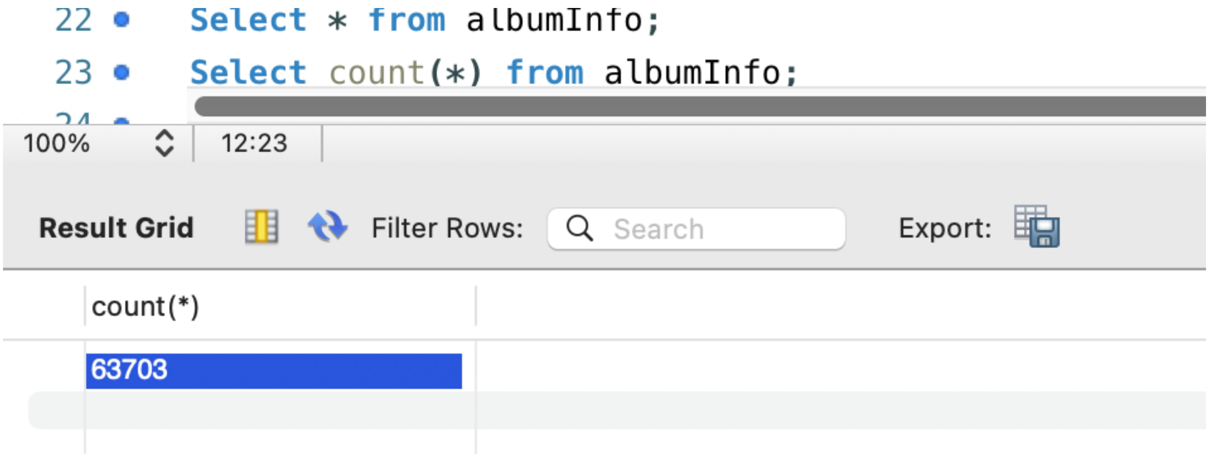
3.5.1 The number of dislike and like



```
drop table albumInfo;
Select * from albumInfo;
```

albumID	artist	title	year	imageSize	likes	dislikes
1	yanlin	sda	2023	3475	103398	51641
2	Yanlin	client1	1998	3475	0	0

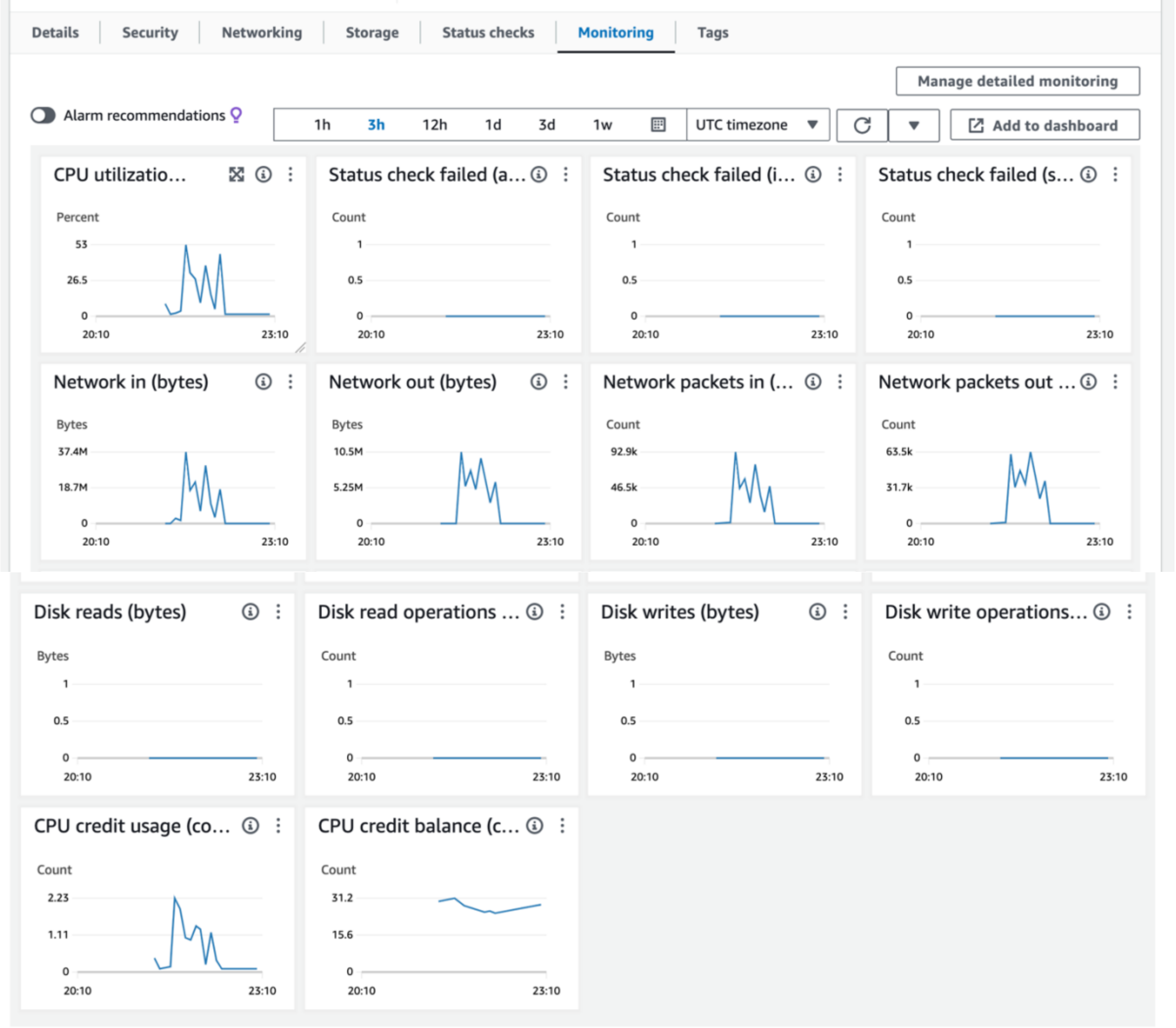
3.5.2 the number of albums



```
Select * from albumInfo;
Select count(*) from albumInfo;
```

count(*)
63703

3.6 Monitoring EC2 instance



Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Manage detailed monitoring
Alarm recommendations
1h
3h
12h
1d
3d
1w
UTC timezone
Add to dashboard
CPU utilizatio...
Percent
53
26.5
0
20:10
23:10
Status check failed (a...
Count
1
0.5
0
20:10
23:10
Status check failed (i...
Count
1
0.5
0
20:10
23:10
Status check failed (s...
Count
1
0.5
0
20:10
23:10
Network in (bytes)
Bytes
37.4M
18.7M
0
20:10
23:10
Network out (bytes)
Bytes
10.5M
5.25M
0
20:10
23:10
Network packets in (...
Count
92.9k
46.5k
0
20:10
23:10
Network packets out ...
Count
63.5k
31.7k
0
20:10
23:10
Disk reads (bytes)
Bytes
1
0.5
0
20:10
23:10
Disk read operations ...
Count
1
0.5
0
20:10
23:10
Disk writes (bytes)
Bytes
1
0.5
0
20:10
23:10
Disk write operations...
Count
1
0.5
0
20:10
23:10
CPU credit usage (co...
Count
2.23
1.11
0
20:10
23:10
CPU credit balance (c...
Count
31.2
15.6
0
20:10
23:10