

**Big Data Processing Technology**

**Final Project**

Project Name: Zanbil Log Data Analysis system

ID: 20812502011

Name: MD SEFATULLAH

Contents

[1. System Introduction 1](#_Toc104140155)

[2. Design and Development of Data Acquisition Module 2](#_Toc104140156)

[3. Design and Development of Data Preprocessing Module 3](#_Toc104140157)

[4. Design and Development of Data Warehouse 4](#_Toc104140158)

[5. Design and Development of Data Analysis Module 5](#_Toc104140159)

[6. Design and Development of Data Export Module 6](#_Toc104140160)

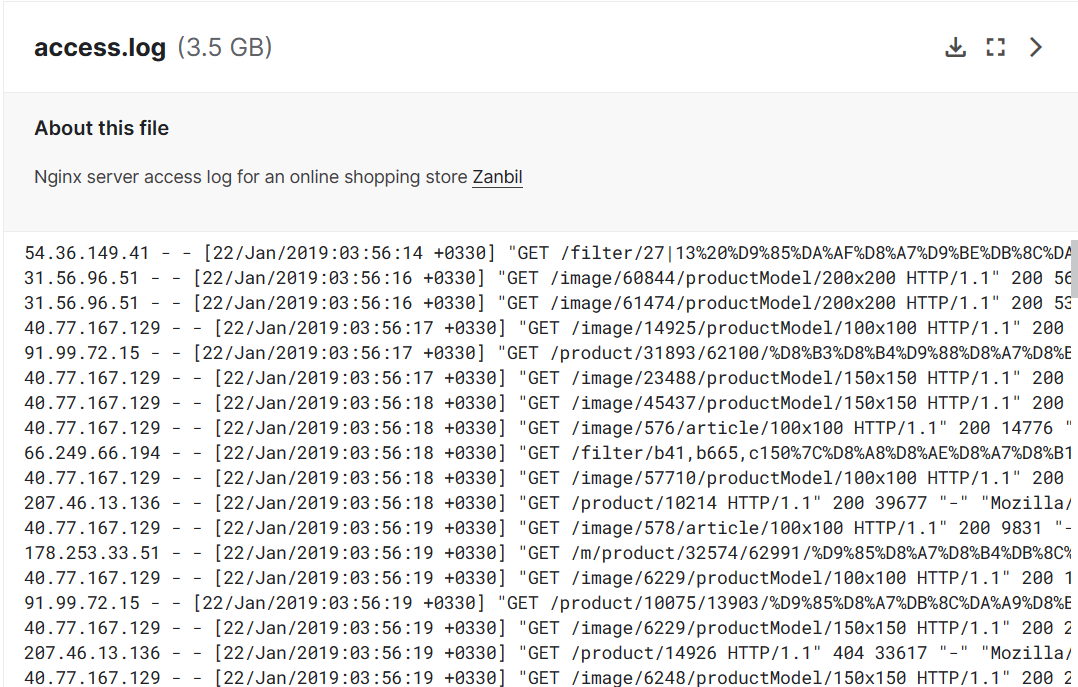
[7. Design and Development of Data Visualization Module 7](#_Toc104140161)

[8. System Characteristics and Conclusion 8](#_Toc104140162)

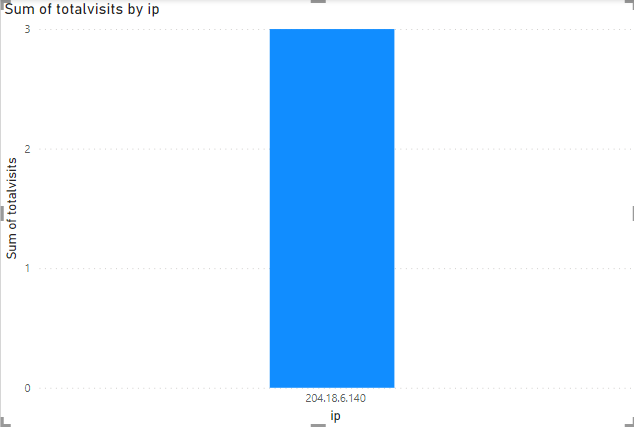
# System Introduction

# 1.1 System Background

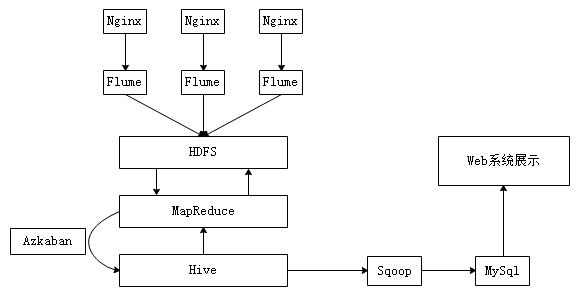
For my project, I leveraged various technologies from the Hadoop ecosystem to examine weblog data and develop a system for analyzing Zanbil log data. The dataset I utilized was obtained from the Web server access logs of the Kaggle website. These logs specifically originated from zanbil.ir, which is an e-commerce platform based in Iran.



Through the analysis of the dataset, I performed computations to identify the IP addresses present in the data and determined the IP address that was visited most frequently on the website. Employing the power of MapReduce technology, I efficiently processed the data, eliminating duplicate entries, and performed data cleansing and filtering operations



1.2 System Architecture Design



1. Collect the log files generated by the Nginx server into HDFS using Flume.

2. Create a MapReduce application to preprocess data in the format of the original log file.

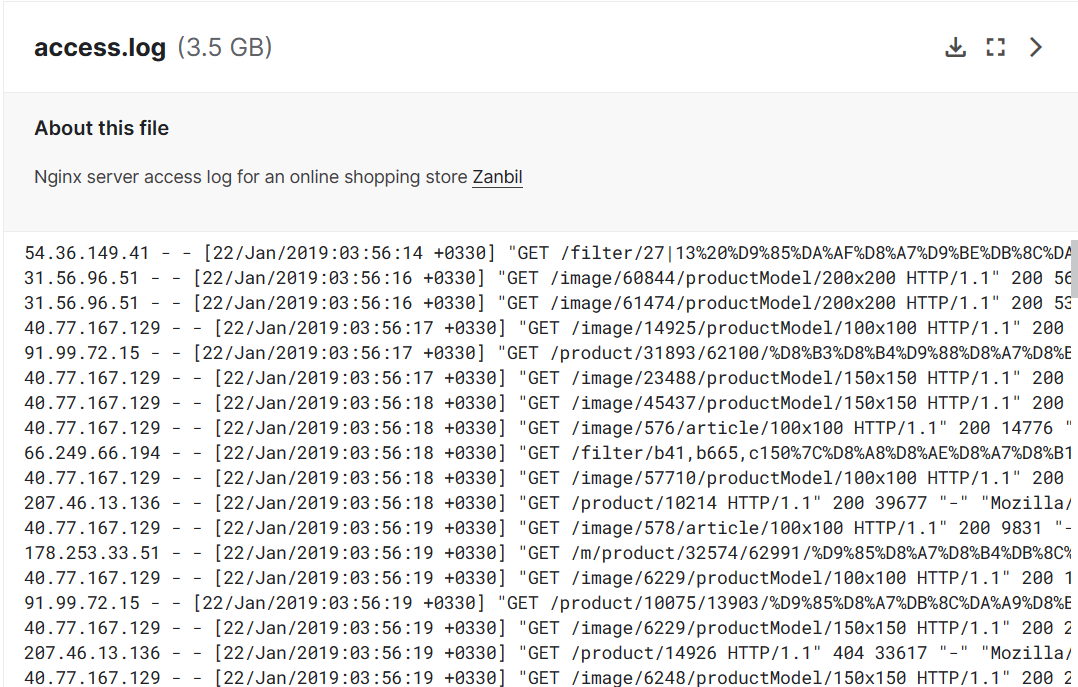
3. For the most crucial data analysis, use Hive.

4. Export the analysis findings to MySQL using Sqoop.

5. Data visualization via a web-based system

# Data Description

I obtained the Web Server access log data for my project from the Kaggle website, specifically using the access.log file provided in the dataset available at <www.kaggle.com/datasets/eliasdabbas/web-server-access-logs?select=access.log.> The collected data logs originated from zanbil.ir, an e-commerce platform based in Iran.



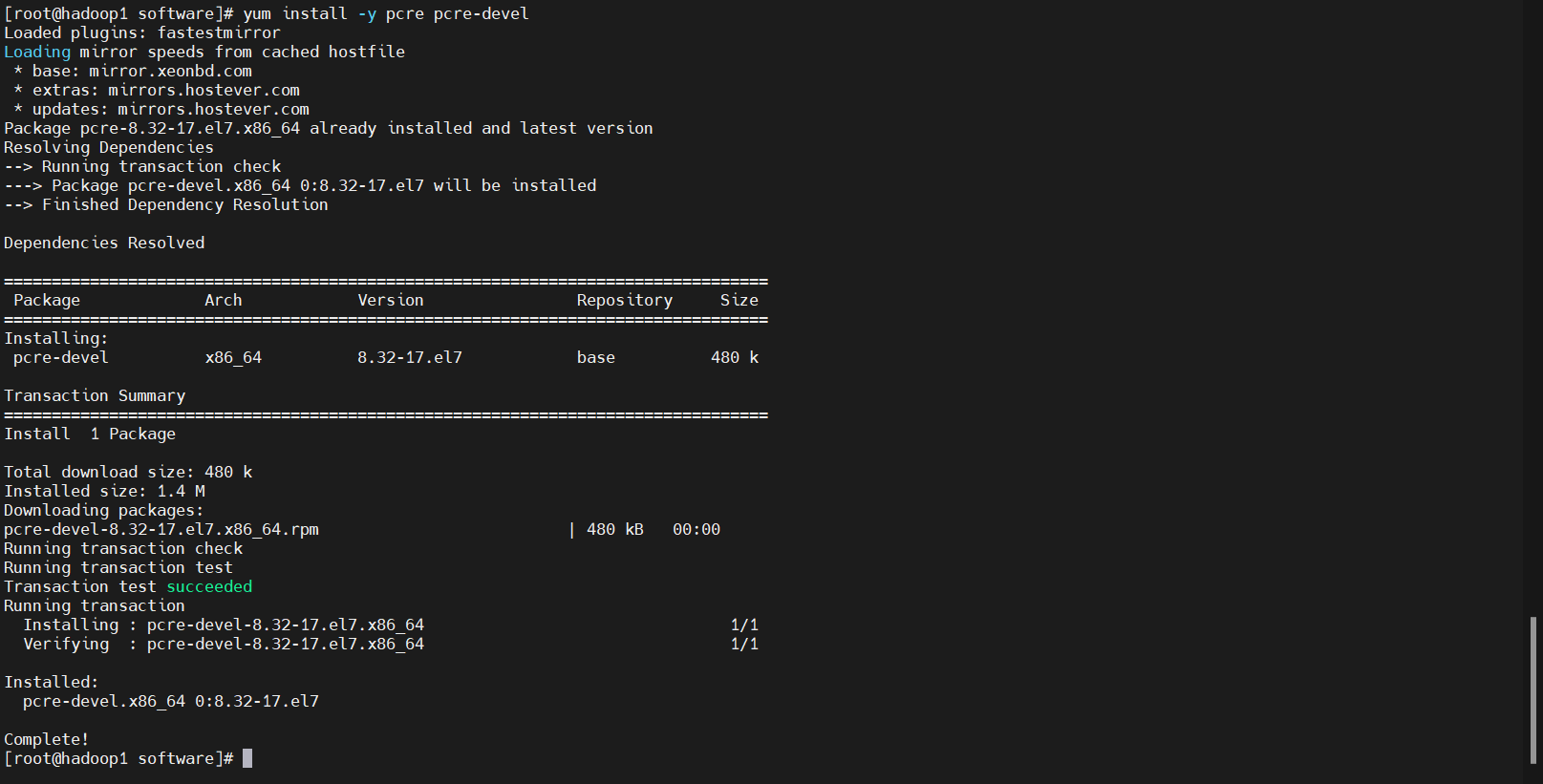
Nginx Server Installation:

At first I have to install c & c++ environment using following command,

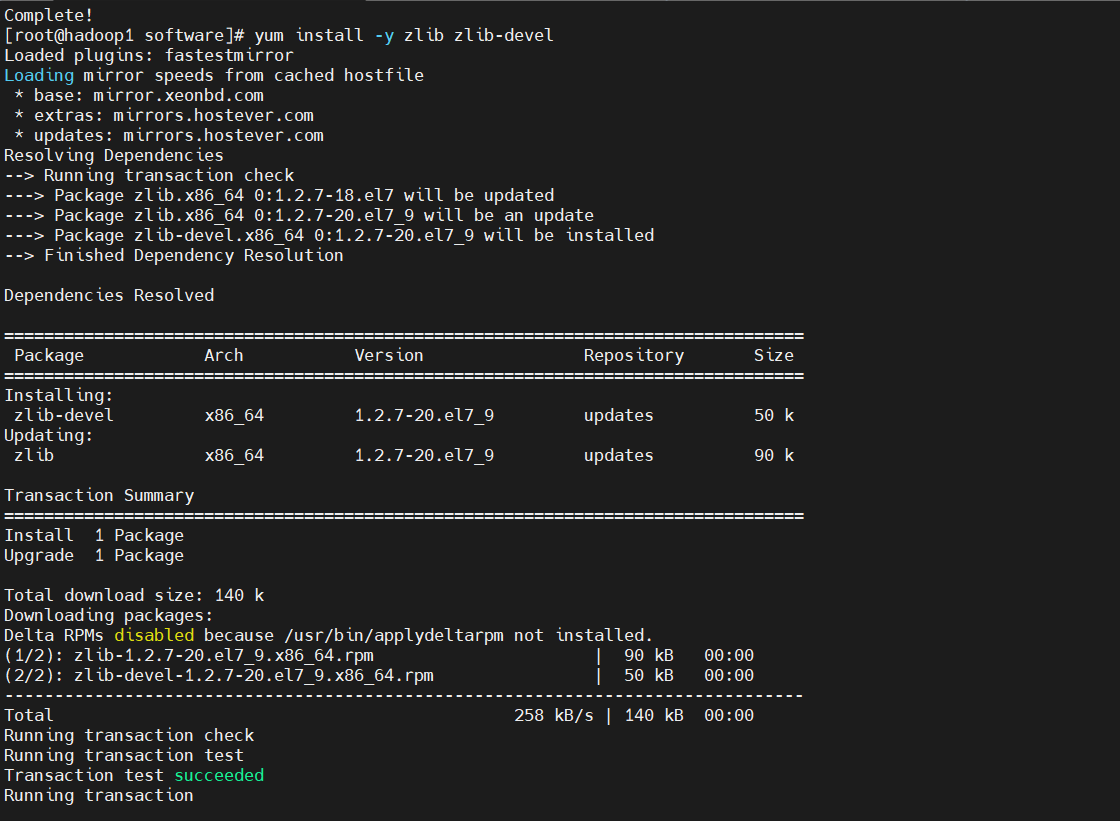
$ yum install gcc-c++

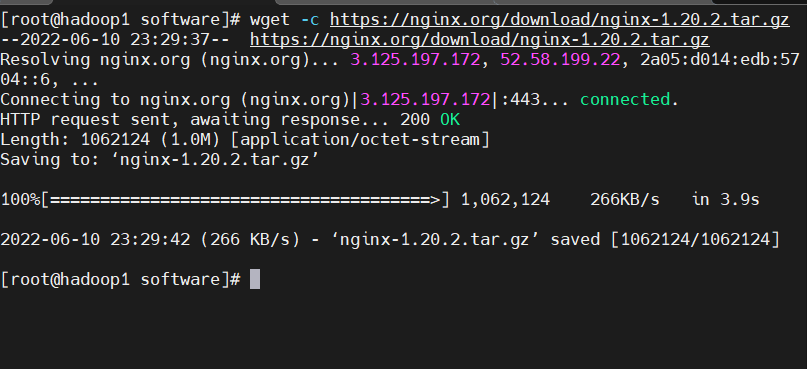
Then install pcre-devel, zilb & OpenSSL using following command:

$ yum install –y pcre pcre-devel



$ yum install –y zilb zilb-devel





$ yum install –y openssl openssl-devel

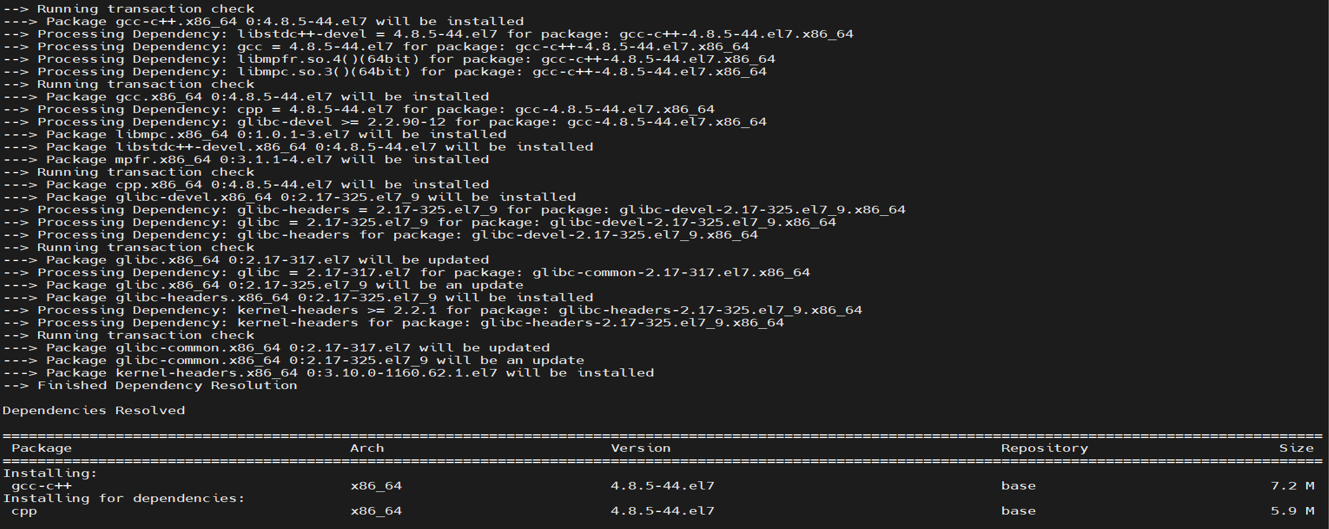
Now download and decompress nginx package using following command

$ yum –y install wget

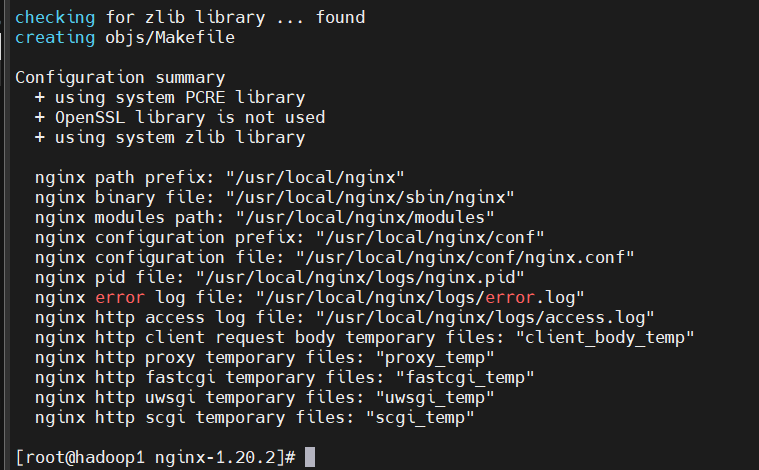
$ cd /export/software

$ wget -c <https://nginx.org/download/nginx-1.20.2.tar.gz>

$ tar -xzf nginx-1.20.2.tar.gz -C /export/servers/

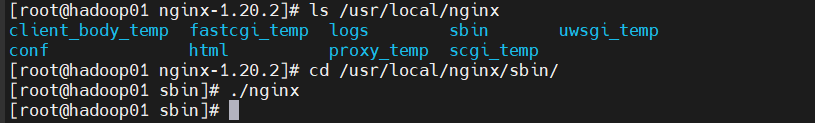


cd /export/servers/nginx-1.20.2/ &./configure it



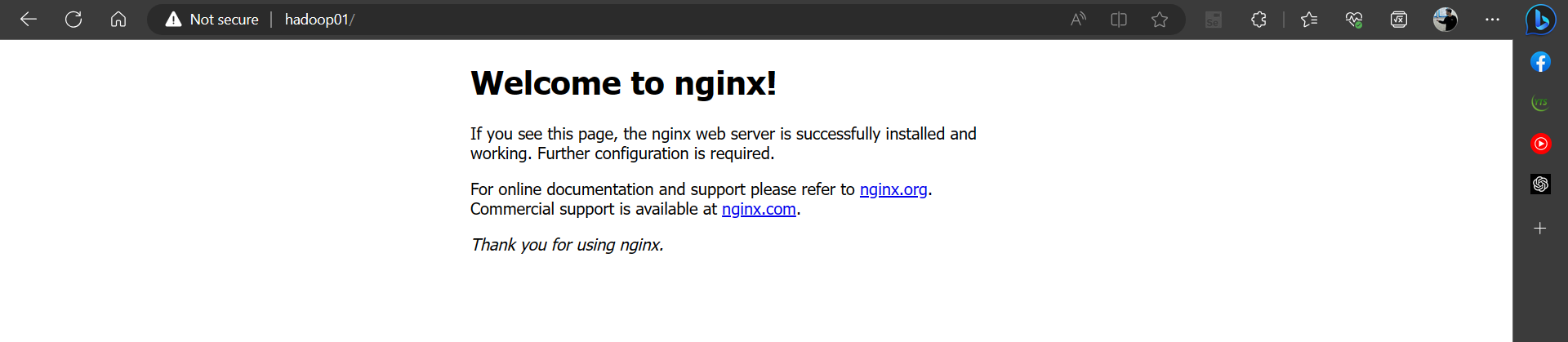
Then make & make install

start nginx : ./nginx

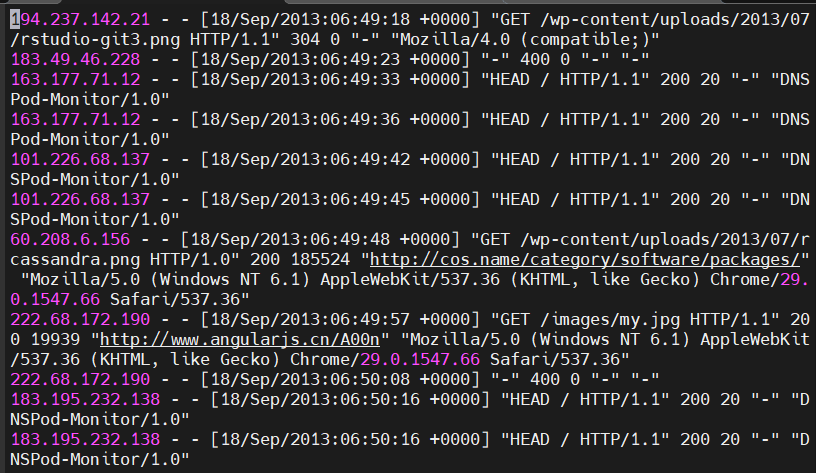


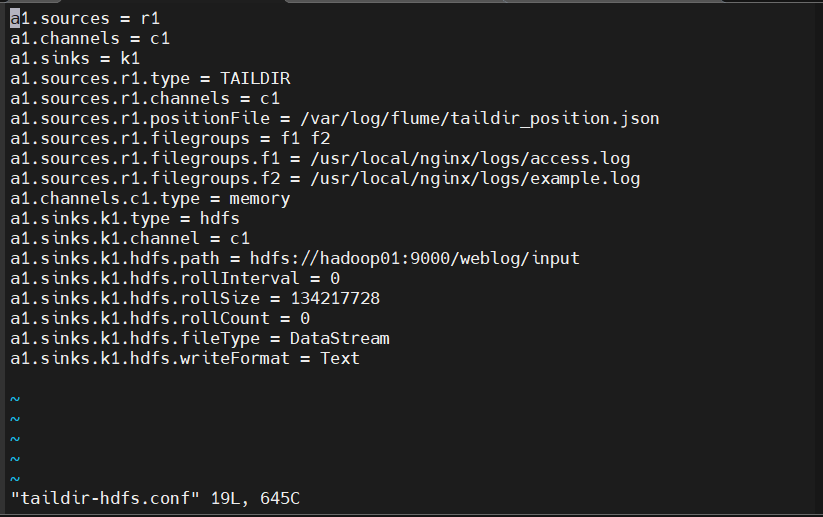
Check Nginx Server:

Then we can check the nginx is working or not. We need to visit: “hadoop01/”



View log file:





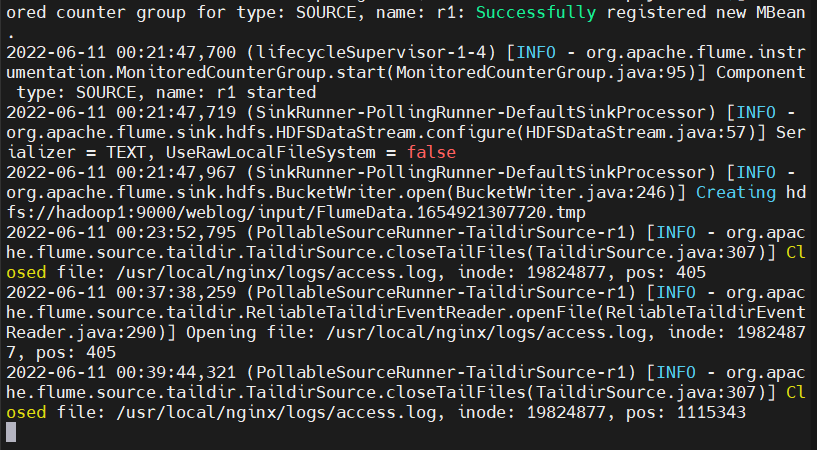
Upload the access.log file to simulate user’s visits to the website.

Upload access.log file to hadoop01

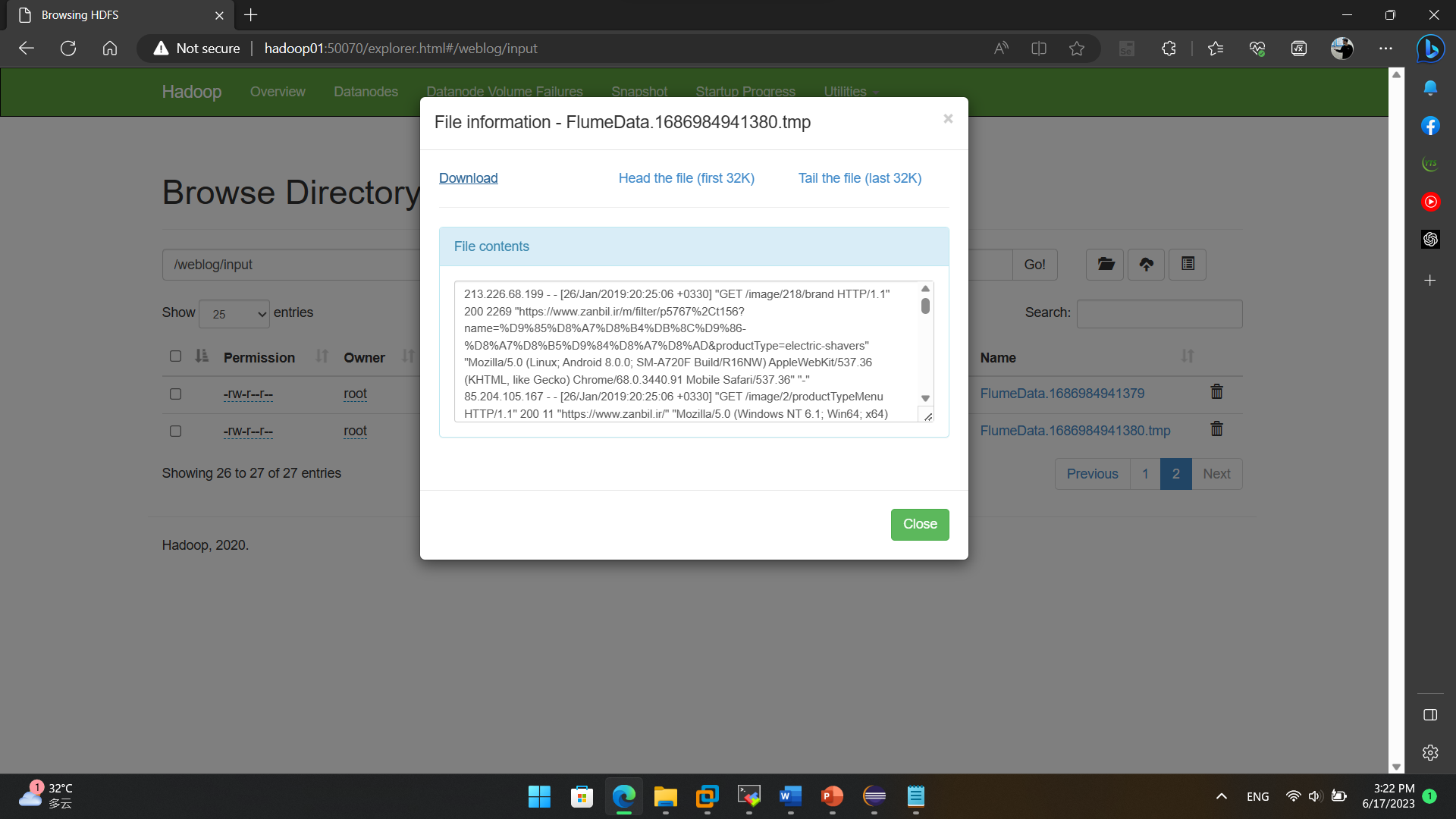


Simulate user’s visits

$ cat /export/data/weblog/access.log > /usr/local/nginx/logs/access.log



And check the flume log file to hadoop01



# Design and Development of Data Preprocessing Module

Data cleaning

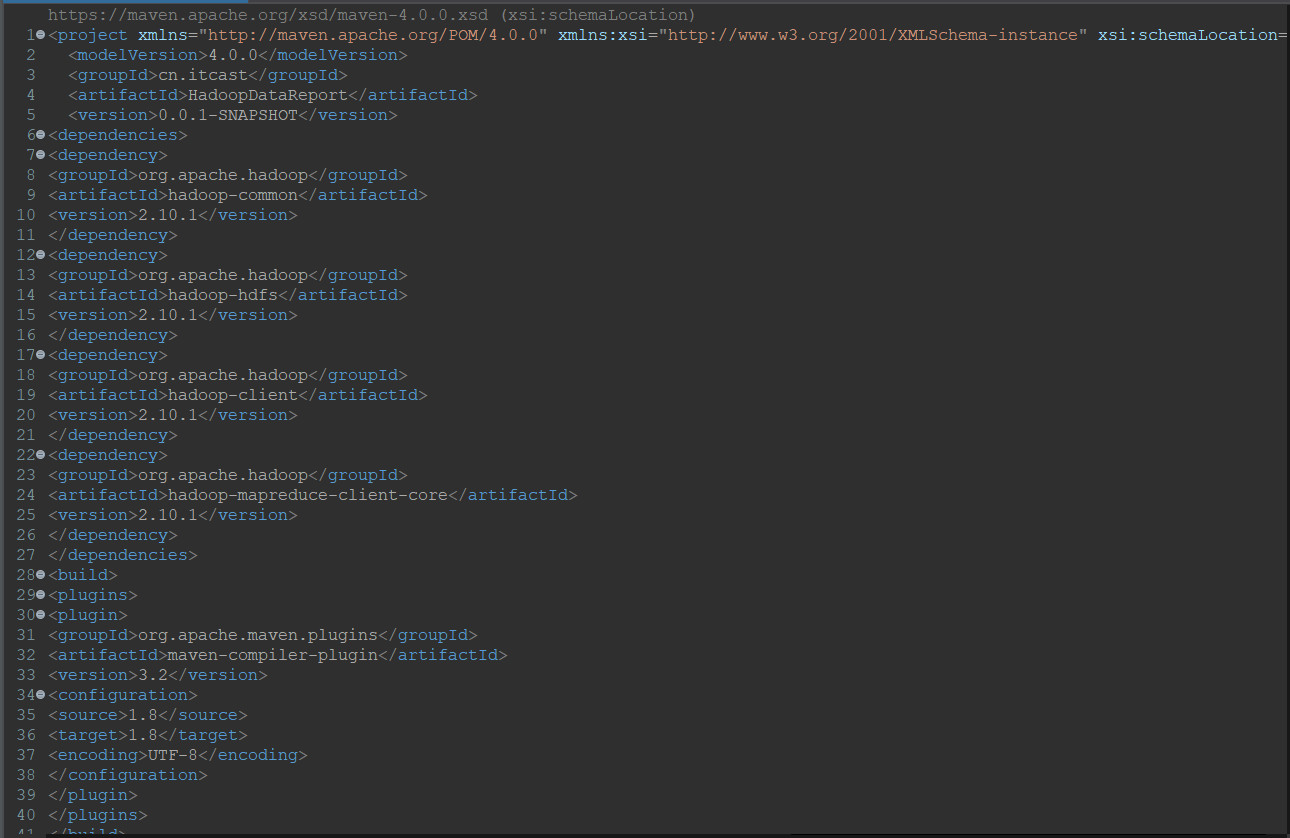
Data integration

Data transformation

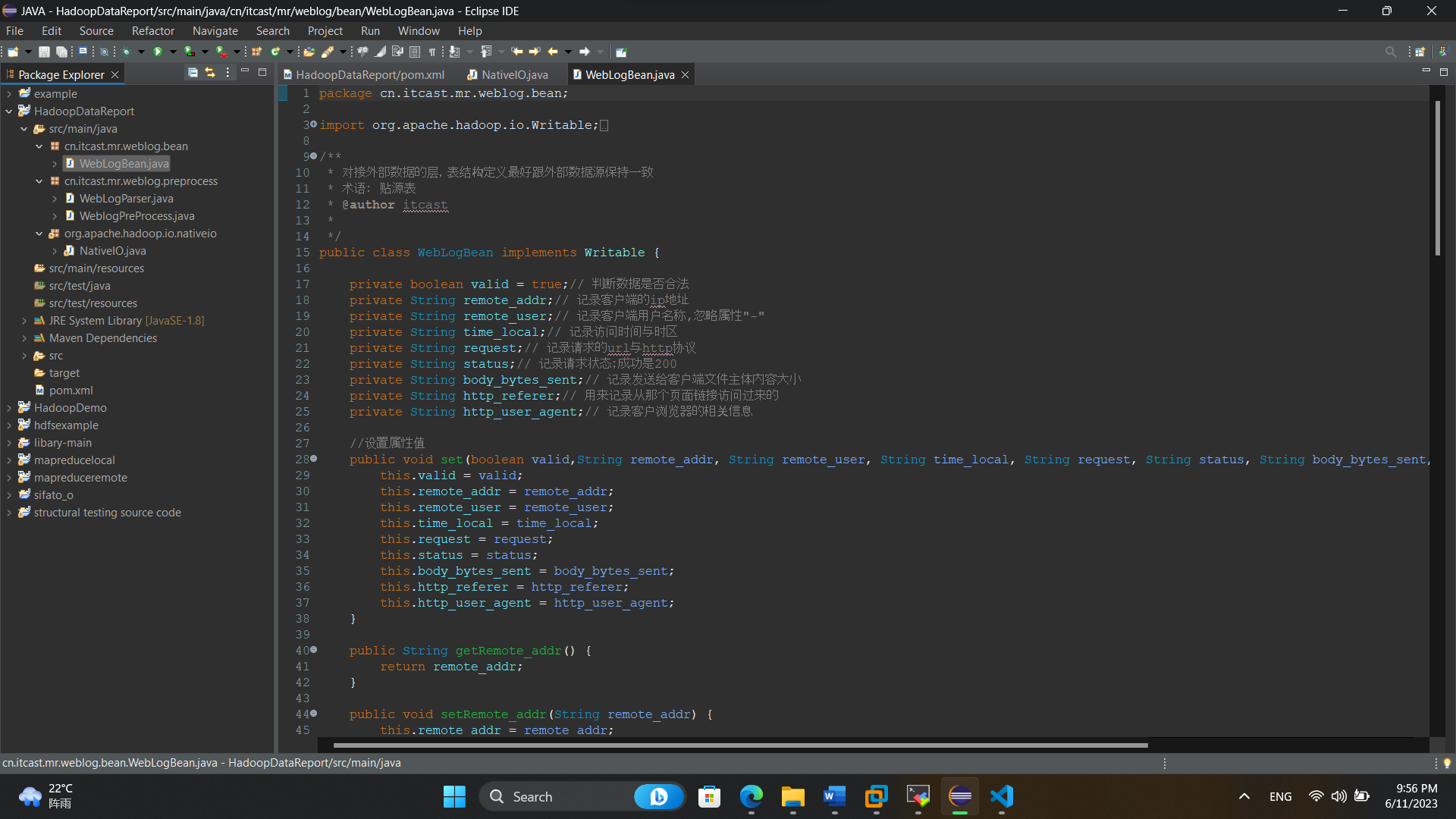
Data reduction

Data discretization

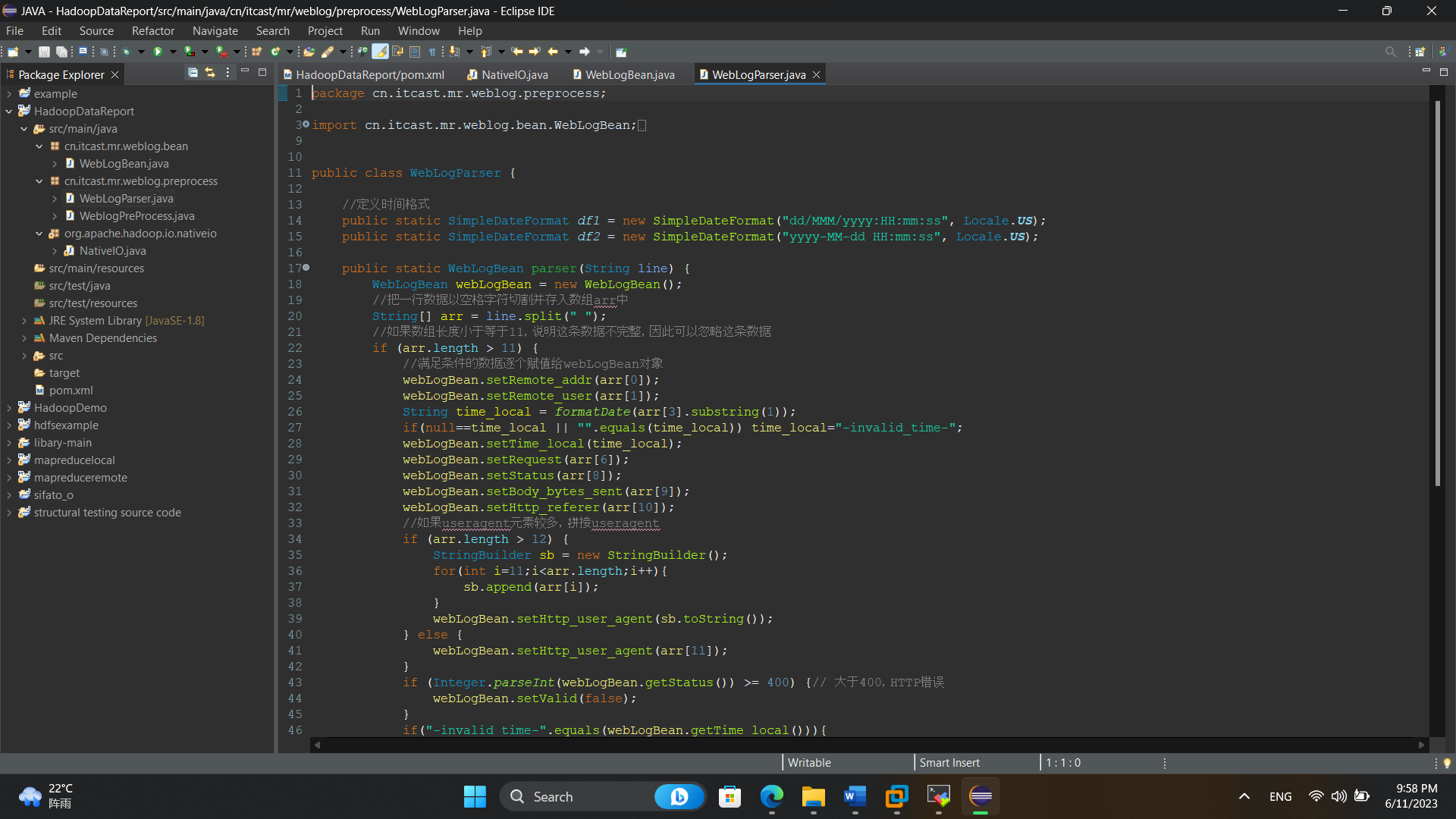
Create a Maven project add the dependencies in pom.xml



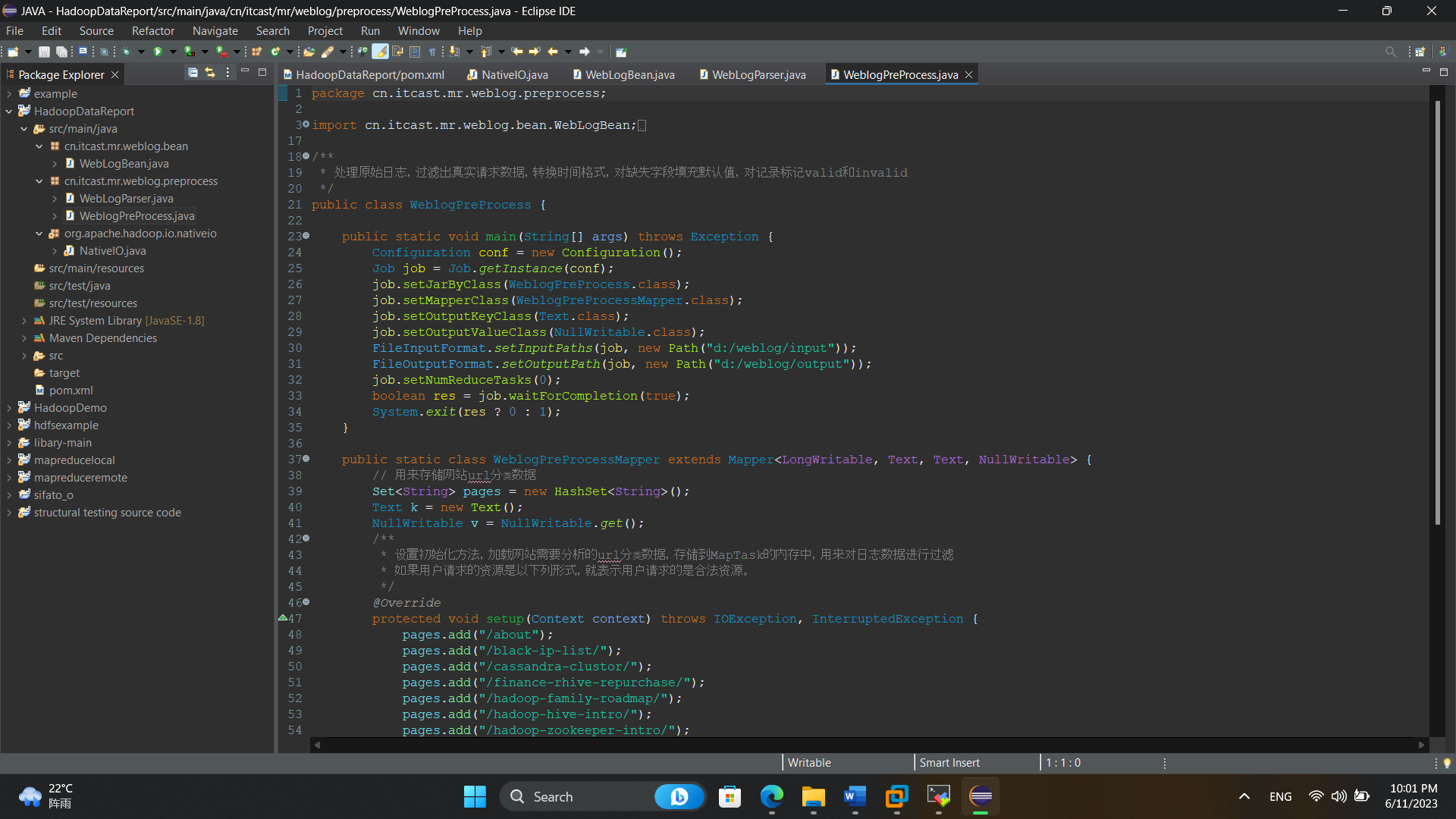
Create javaBean Object to represent log record

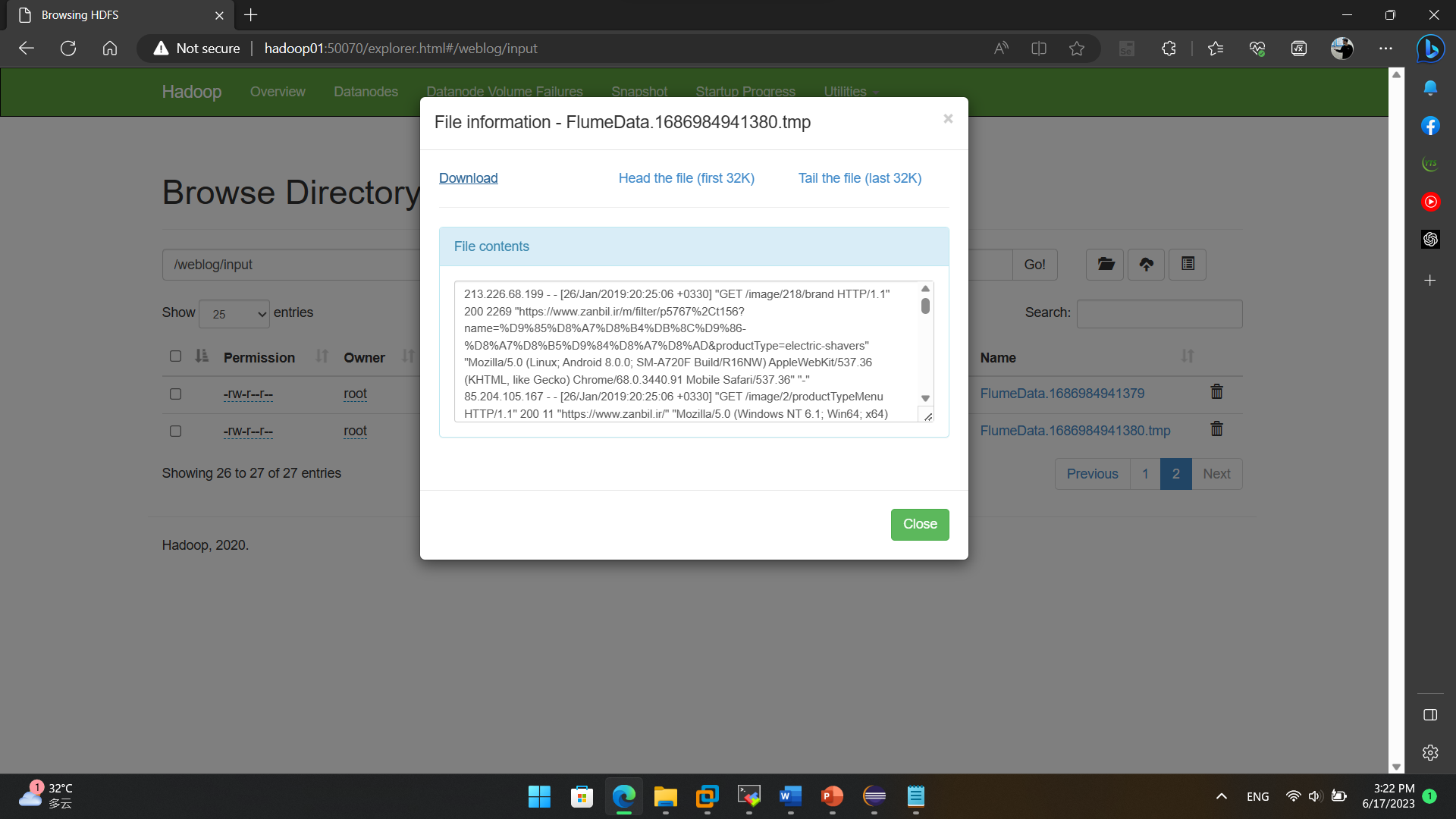


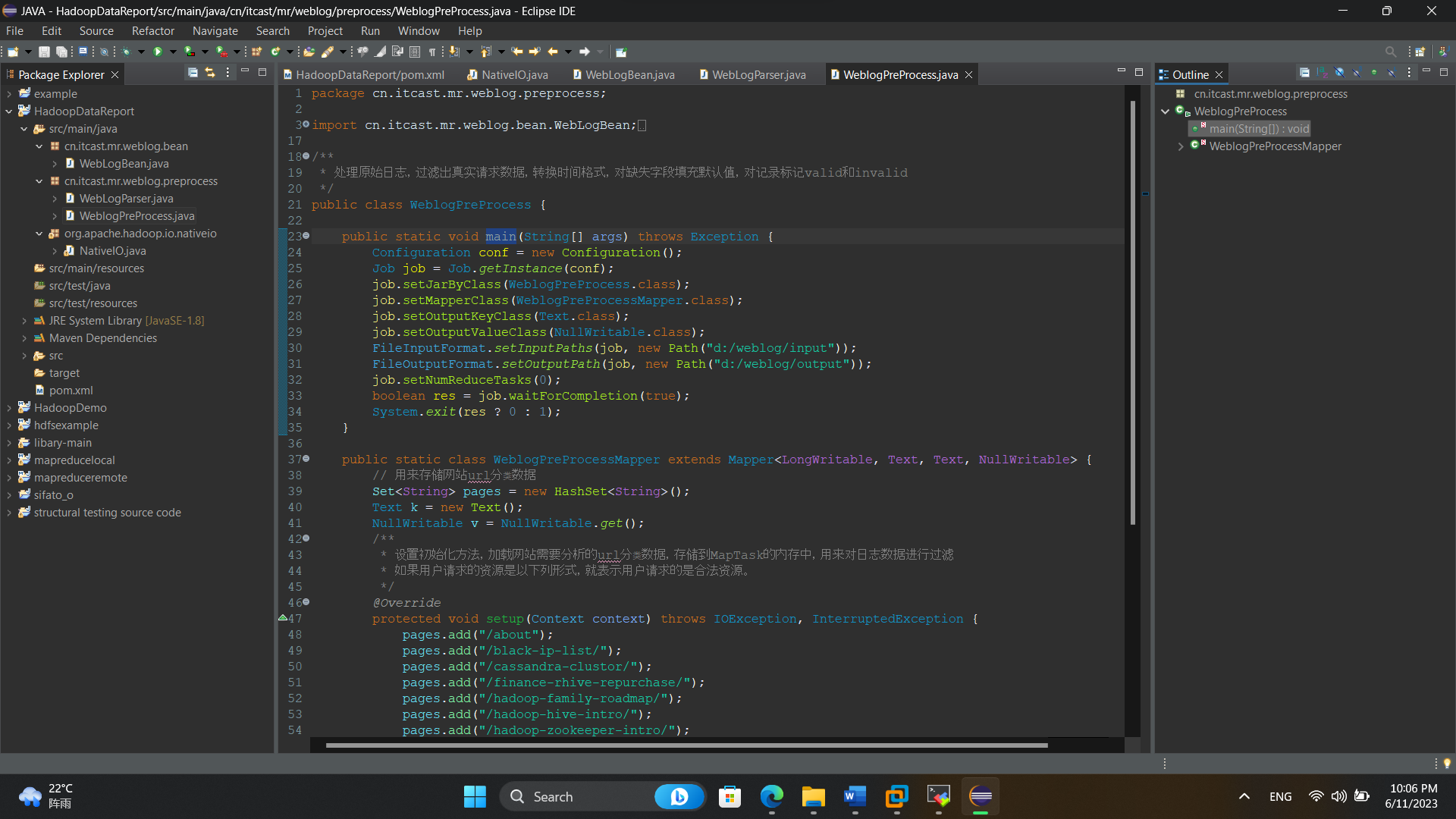
Write a MapReduce program for data pre-processing. It reads the original collected data, process it, and store it.



In weblogPreprocess.java we use the data which we upload on hadoop and download the process data



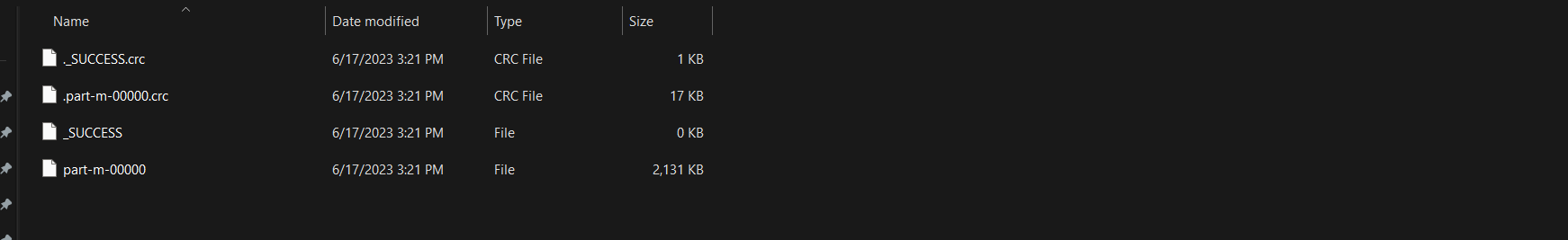




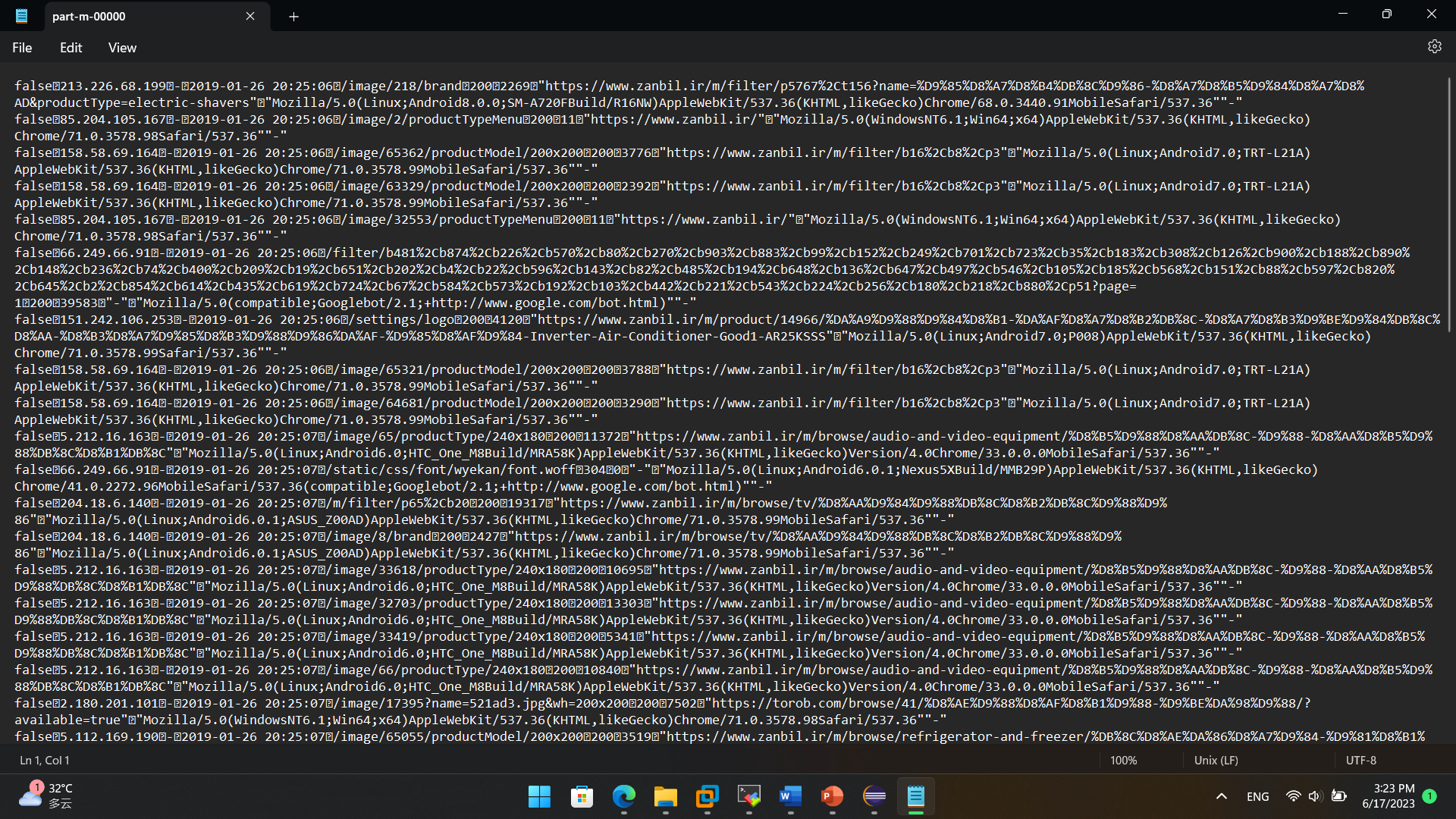
This downloaded data file we put it on :D/weblog/input/



After run the project in create a processed file on :D/weblog/output



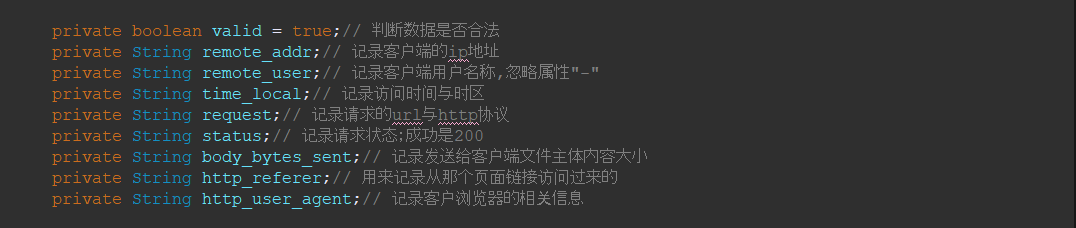
Now we get the data



# 4. Design and Development of Data Warehouse

Design ods\_weblog\_origin table to store original log data. The structure of ods\_weblog\_origin.

|  |  |
| --- | --- |
| **Column** | **Description** |
| **valid** | Whether it is valid or not |
| **remote\_addr** | Visiting IP |
| **remote\_user** | Visitor User Information |
| **time\_local** | Request Time |
| **requst** | Request URL |
| **status** | Response Code |
| **body\_byte\_sent** | Response Byte Size |
| **http\_referer** | Source URL |
| **http\_user\_agent** | Visitor Browser Information |



Design dw\_weblog\_detail table to store the detailed log data

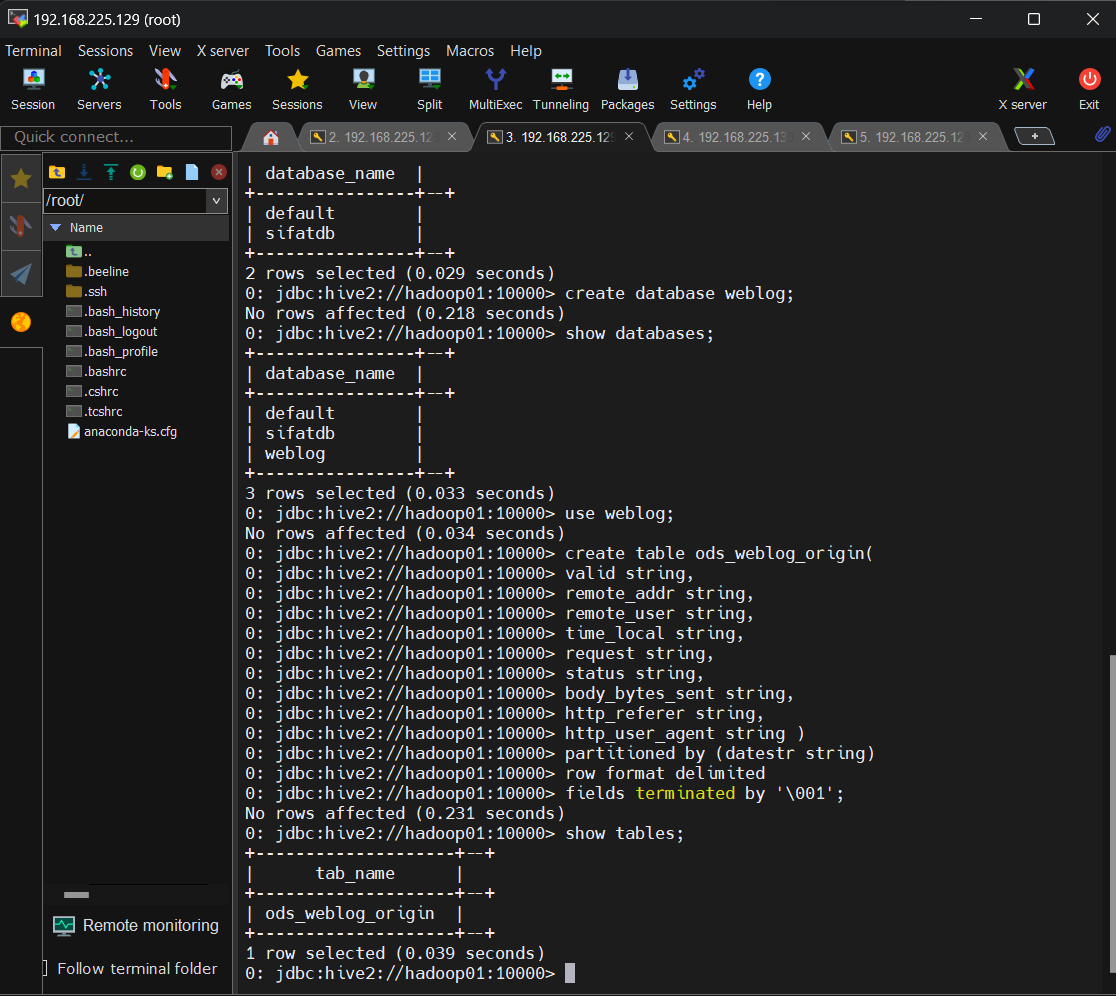
|  |  |  |  |
| --- | --- | --- | --- |
| Column | Description | Column | Description |
| **valid** | Whether it is valid or not | request | Request URL |
| **remote\_addr** | Visiting IP | status | Response Code |
| **remote\_user** | Visitor User Information | body\_byte\_sent | Response Byte Size |
| **time\_local** | Request Time | http\_referer | Source URL |
| **daystr** | Request Day | ref\_host | Source host |
| **timestr** | Request Time | ref\_path | Source Path |
| **month** | Request Month | ref\_query | Source Query Keys |
| **day** | Request Day | ref\_query\_id | Source Query Values |
| **IP\_address** | Requested IP | http\_user\_agent | Visitor Browser Information |

Design tables to store data analysis information. t\_avgpv\_num table :

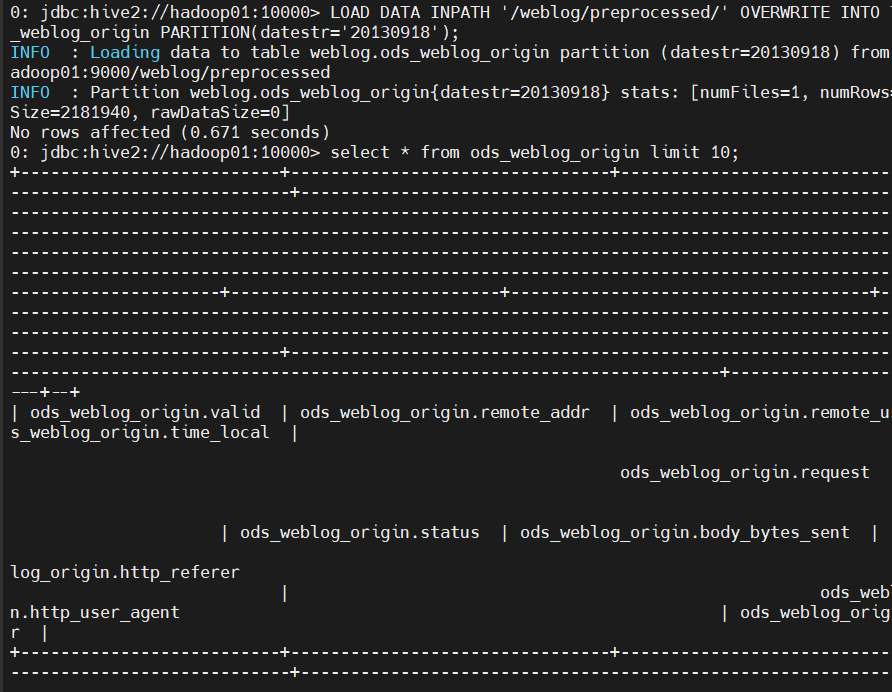
|  |  |
| --- | --- |
| Column | Description |
| **IPStr** | IP address |
| **Visited\_IP\_add** | Most visited IP address |

Now create Database:

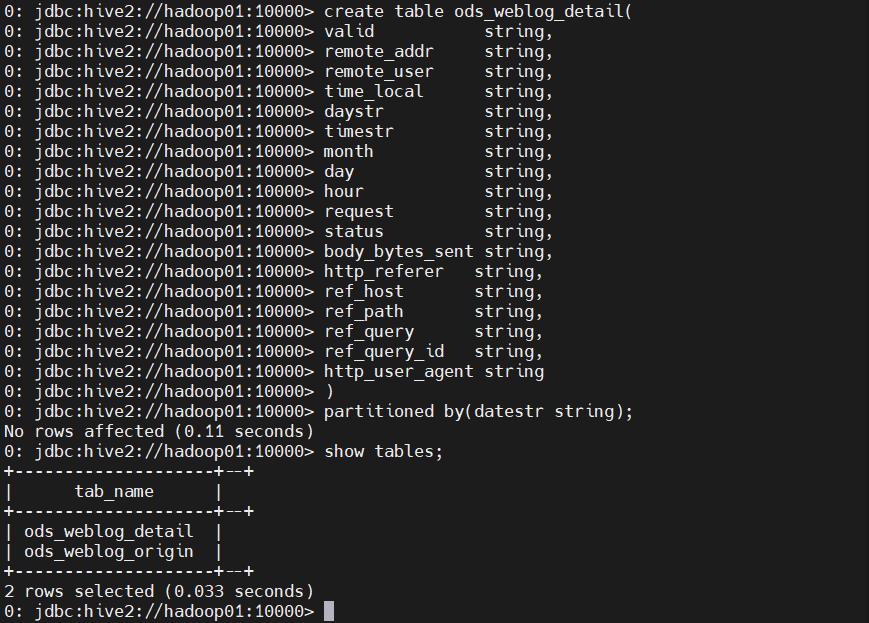
**Create** ods\_weblog\_origin **Table**



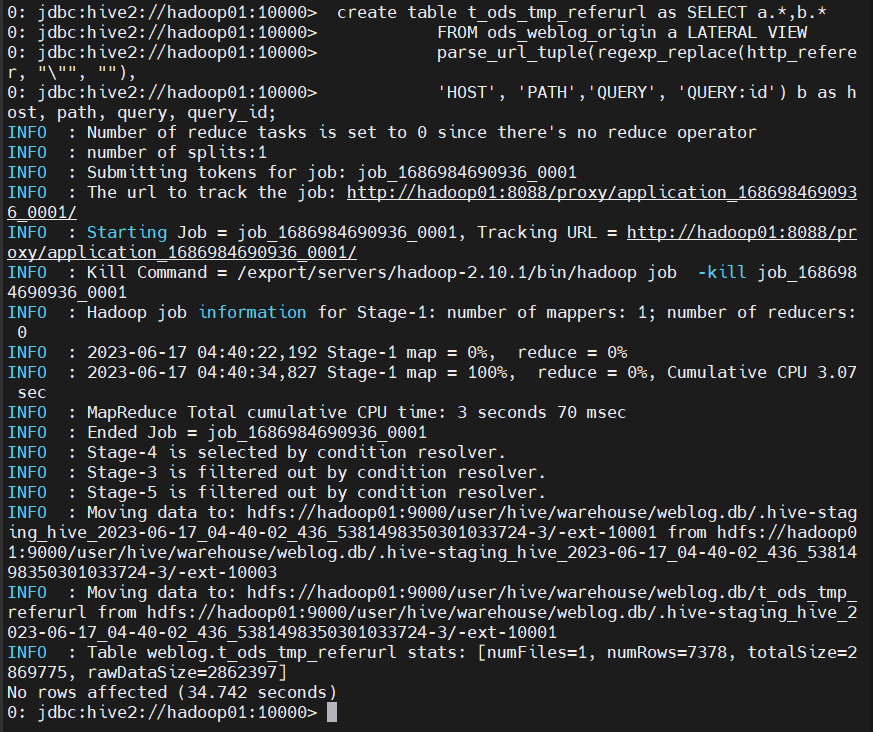
**Load data to hive:**

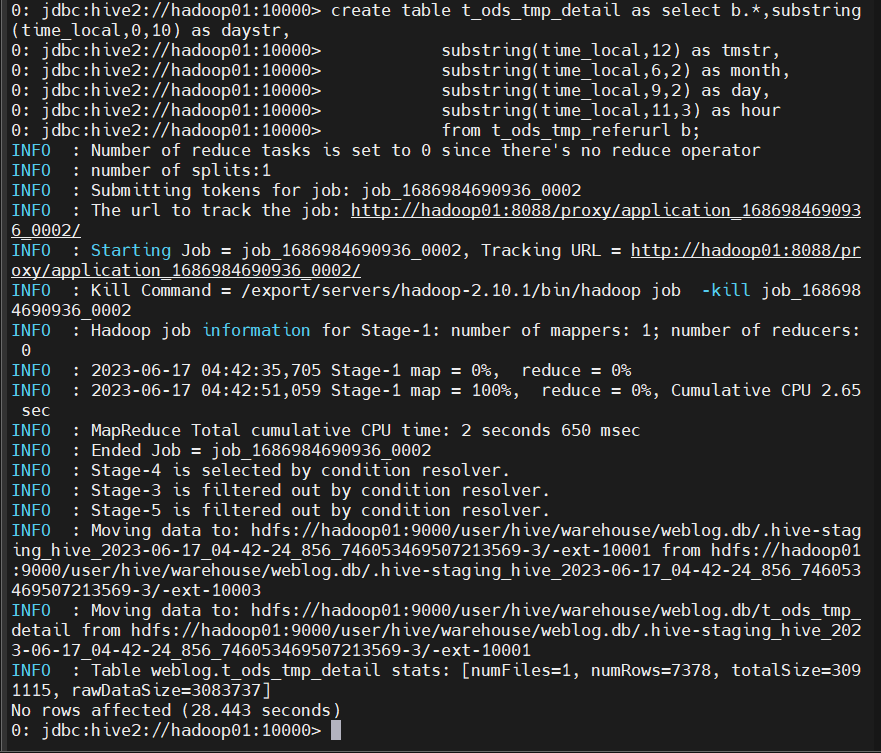
****

**Create ods\_weblog\_detail table:**



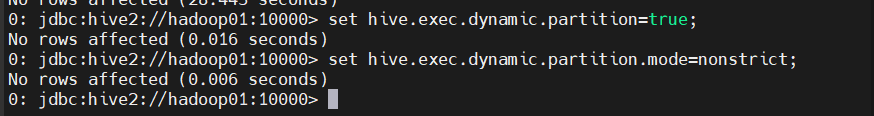
**Create temporary tables to process http\_refere and time\_local**



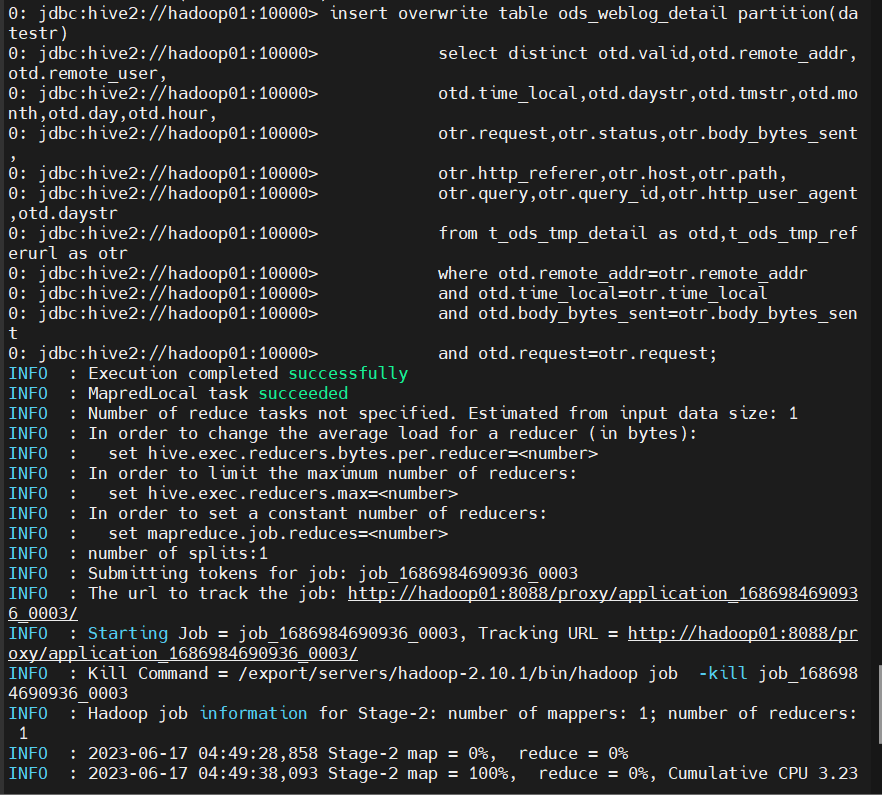


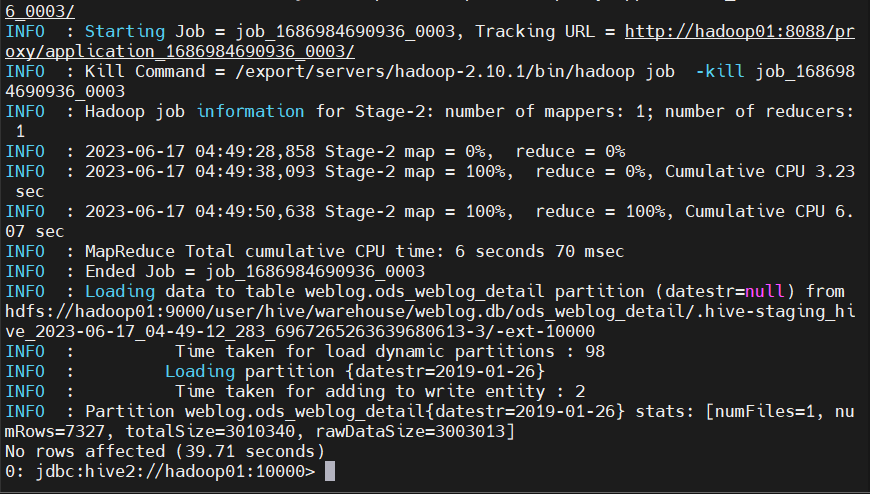
**set dw\_weblog\_detail table**

Hive cannot be dynamically partitioned by default. So, we need to modify the parameters of the default dynamic partition before loading data



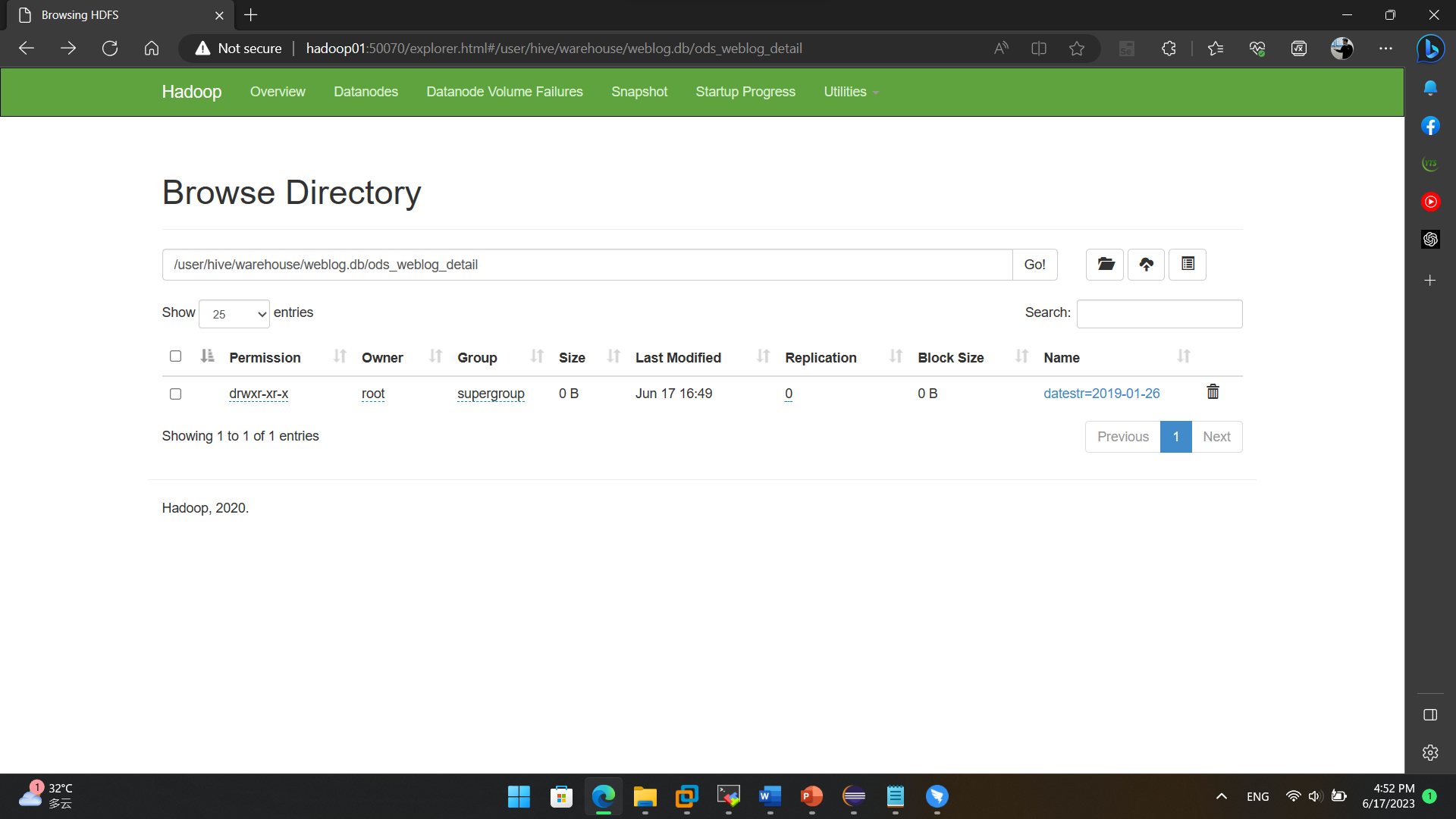
Load data to ods\_weblog\_detail table:





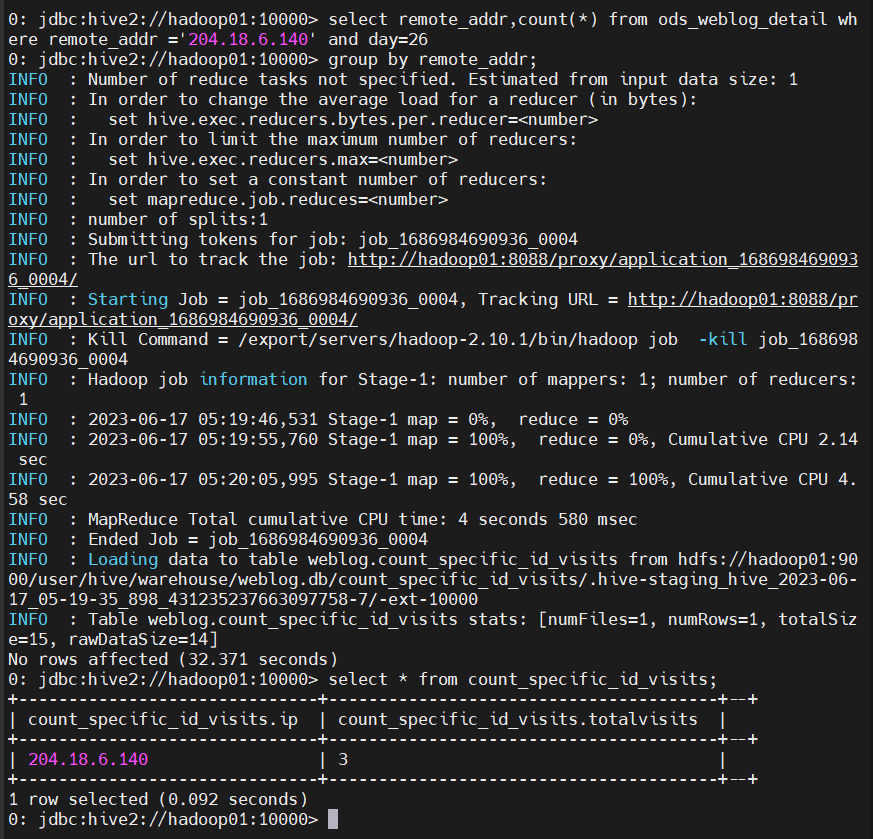
Here we show the 1 partition data group by dataStr

Check HDFS web the table data:

****

# Design and Development of Data Analysis Module

Here Statistic value for Specific Ip address:

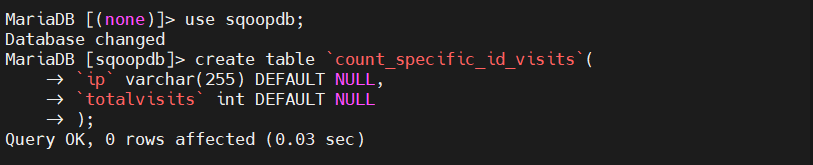


# Design and Development of Data Export Module

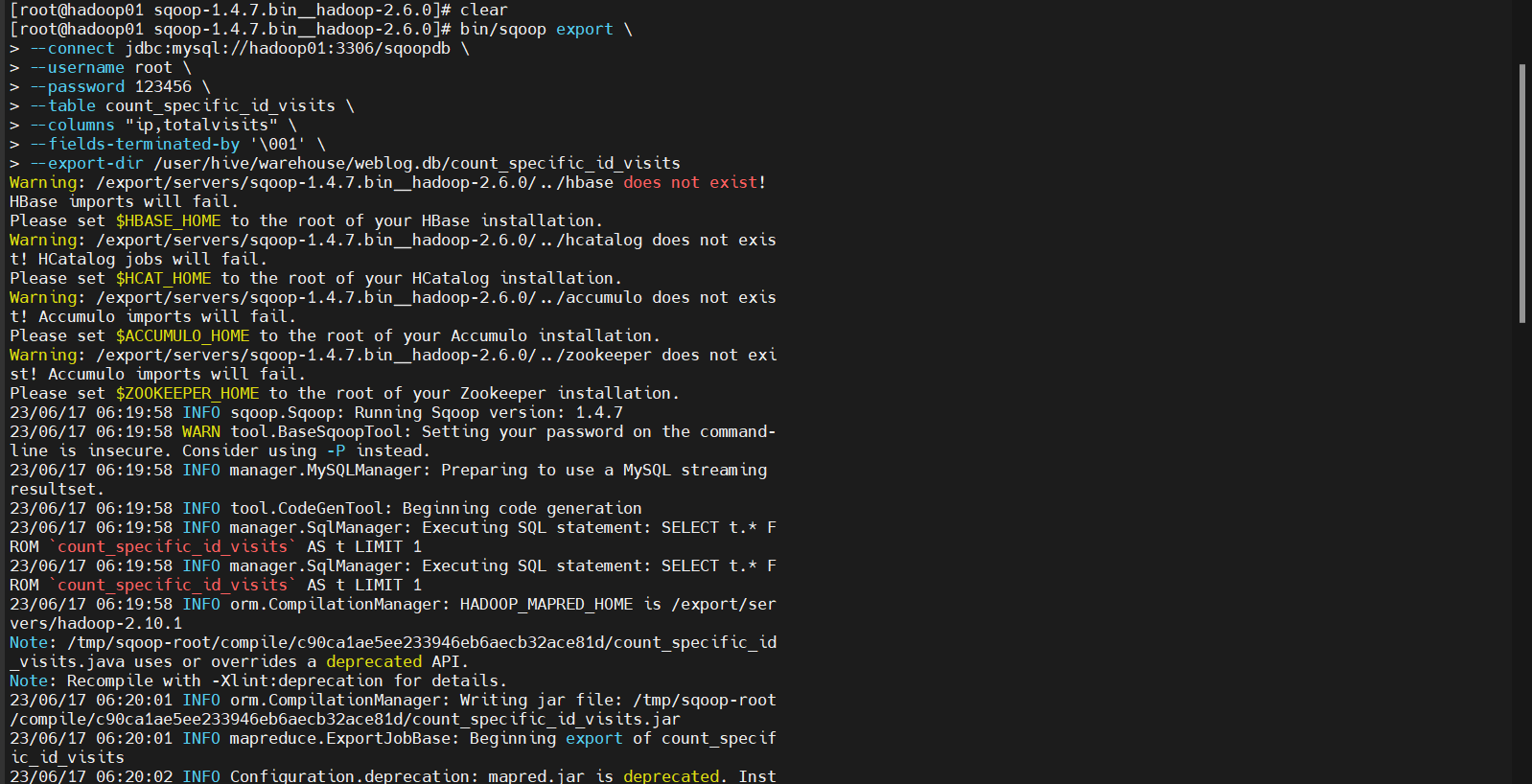
Start MySQL Service, and Connect

$ mysql –uroot –p

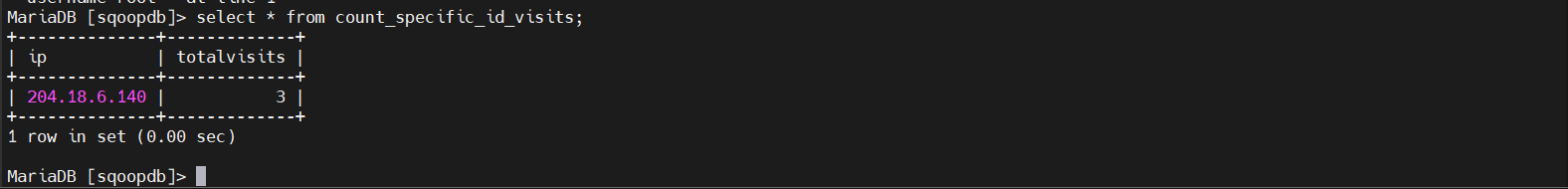
Create a table count\_specific\_id\_visits

Create a table count\_specific\_id\_visits

Export data using sqoop command



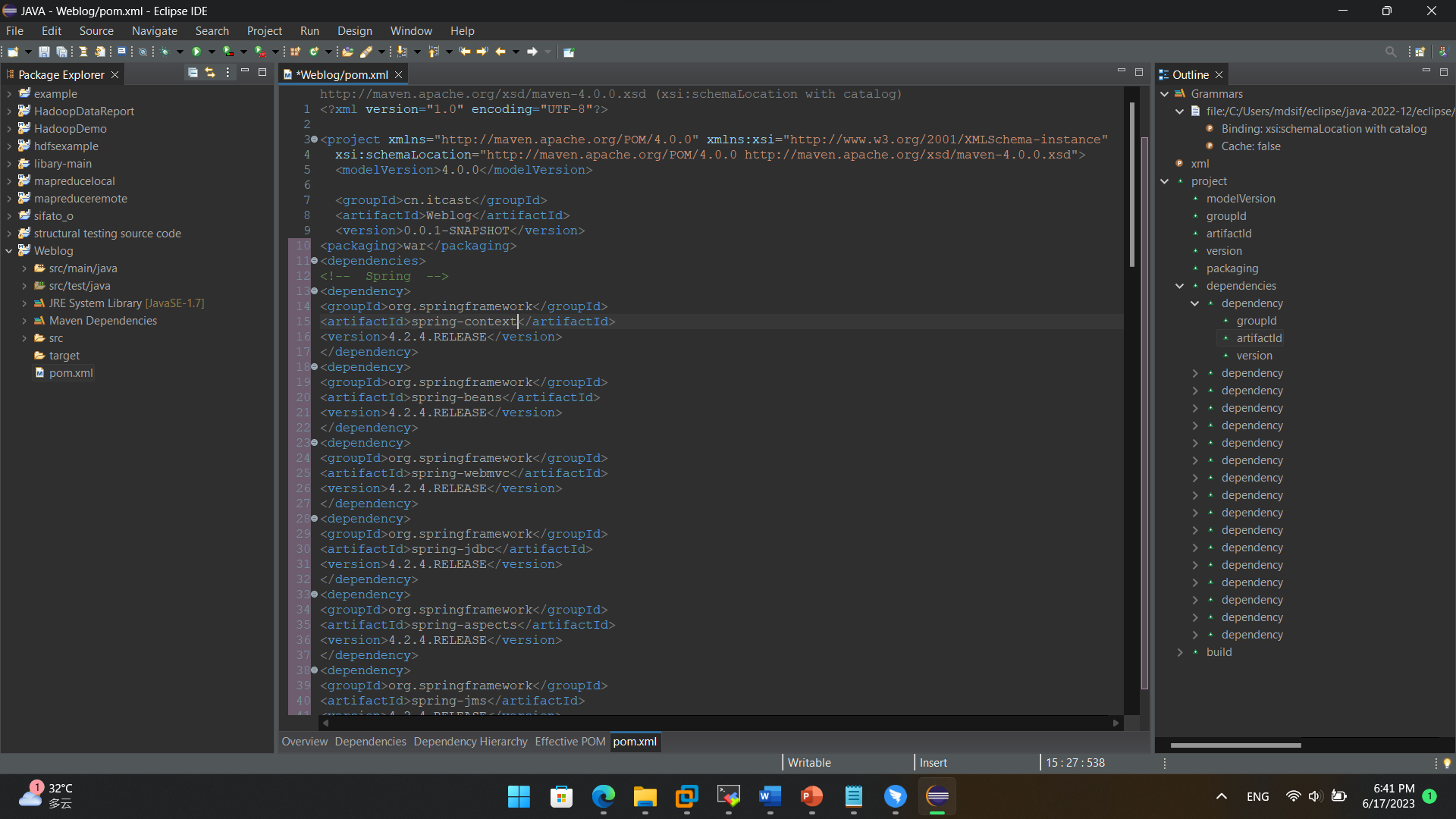
select \* from count\_specific\_id\_visits;



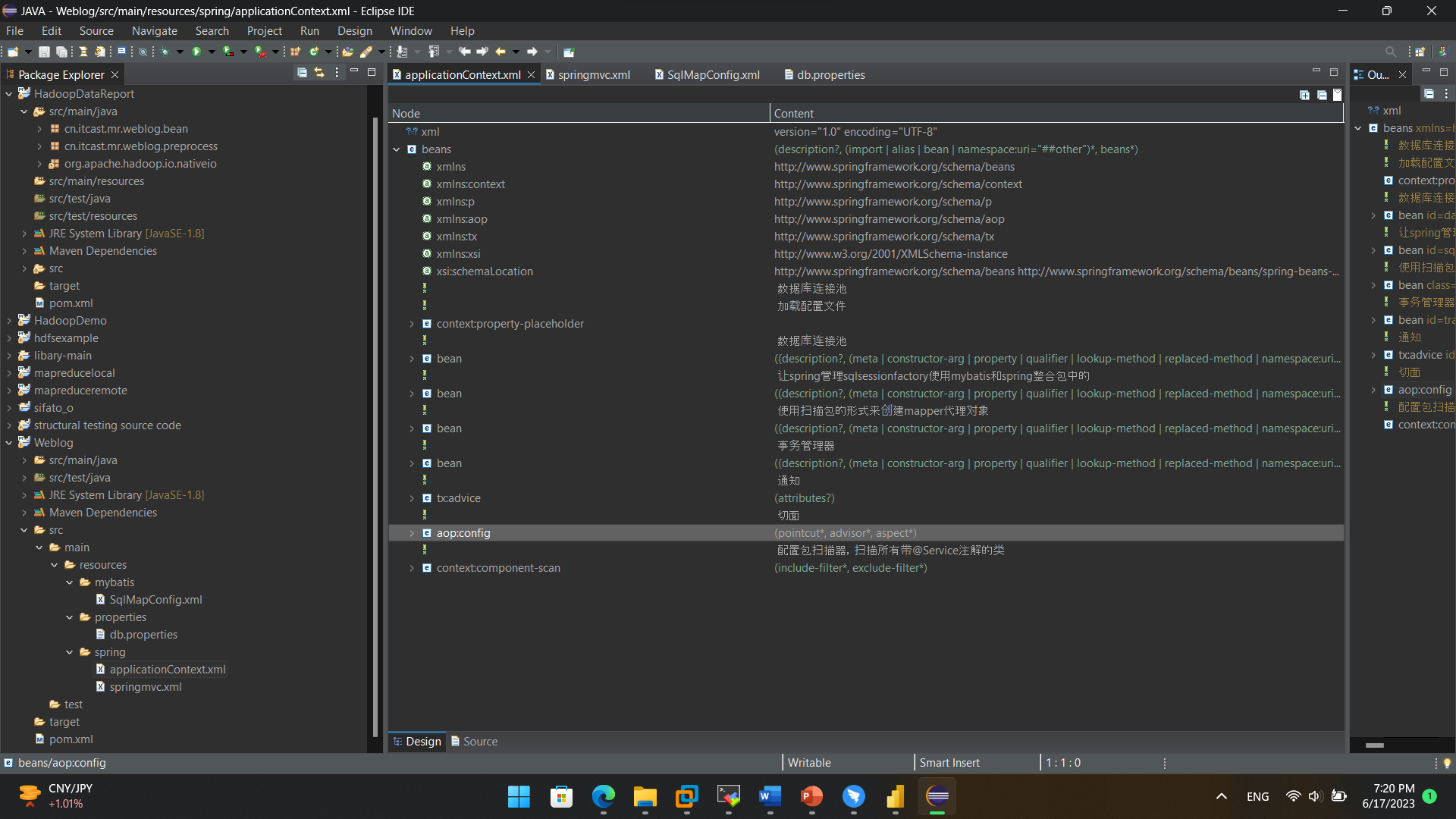
We get the data for visited Ip address.

# Design and Development of Data Visualization Module

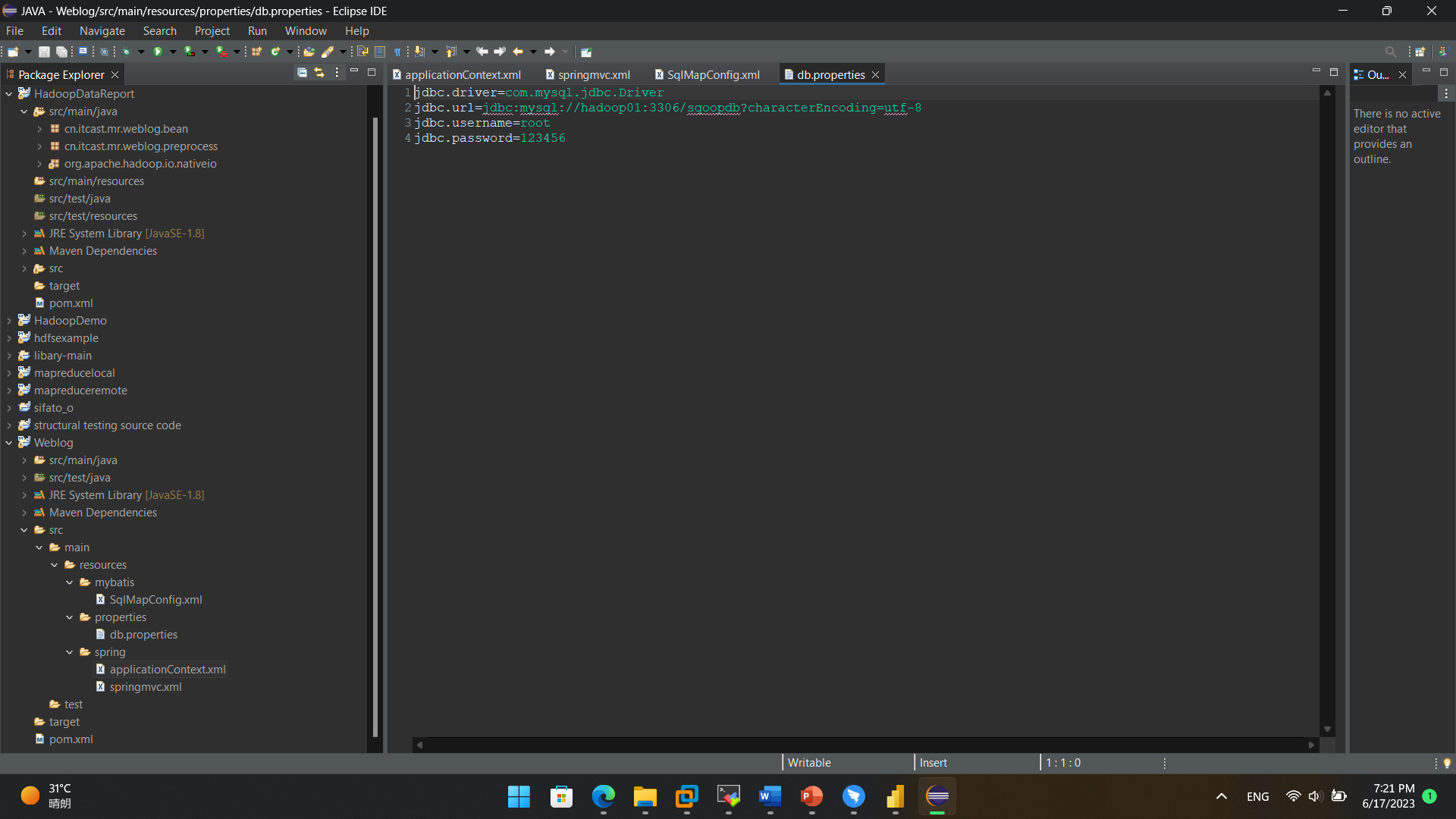
Create Maven project, add dependences in pom.xml



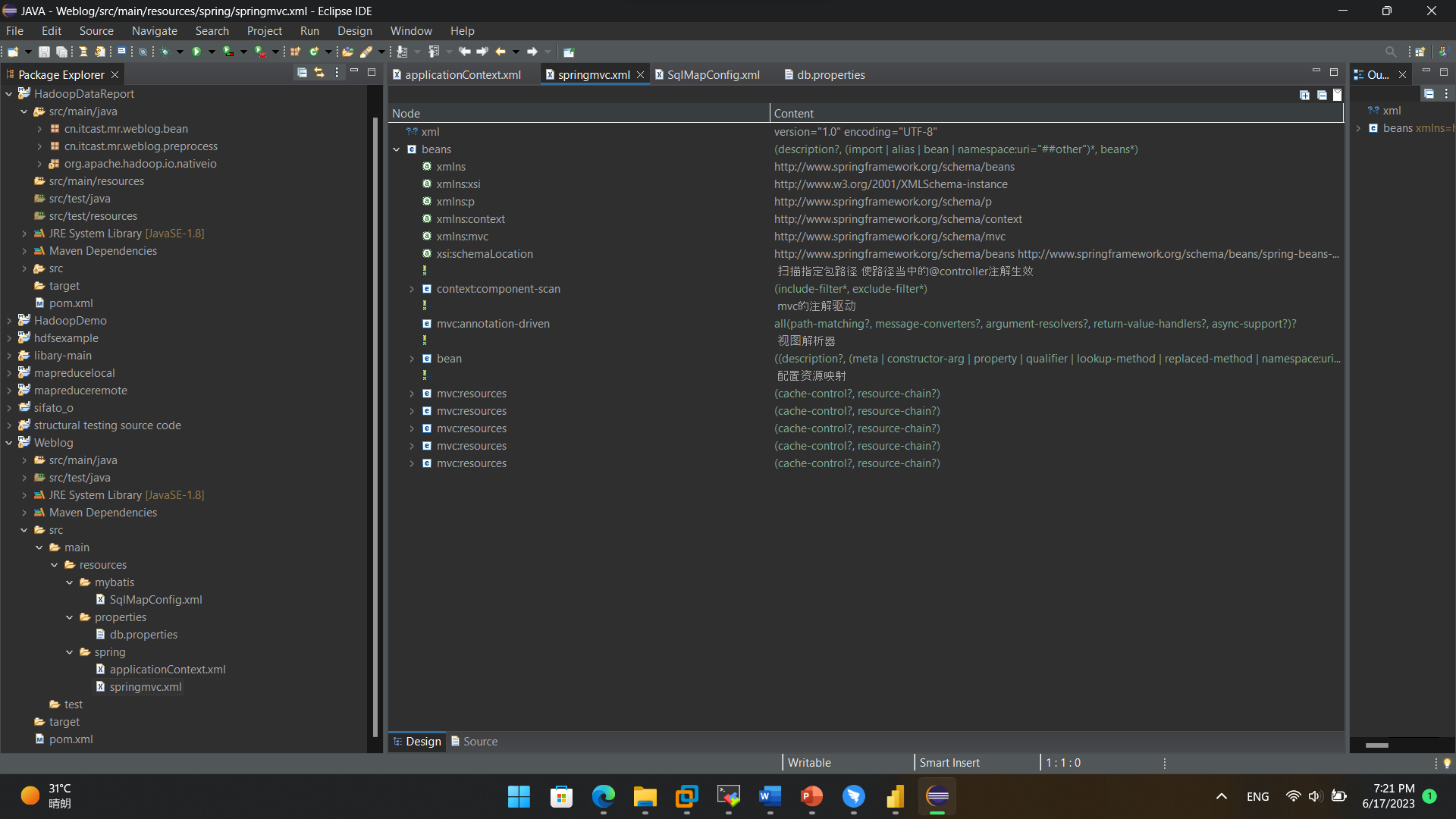
Write configuration file applicationContext.xml needed in SSM framework



Write configuration file db.properties for database setting. Here I use database name sqoopdb and my hadoop1



Write configure files springmvc.xml,web.xml and SqlMapConfig.xml for SpringMVC



# System Characteristics and Conclusion

1. Right click the project, select Run As -> Maven build, and Input "tomcat7:run" in Goal text input
2. Visit <http://localhost:8080/index.html> to check the web page of weblog data analysis system.

