Project Documentation: Hadoop + Hive Pipeline for MIMIC-III Data

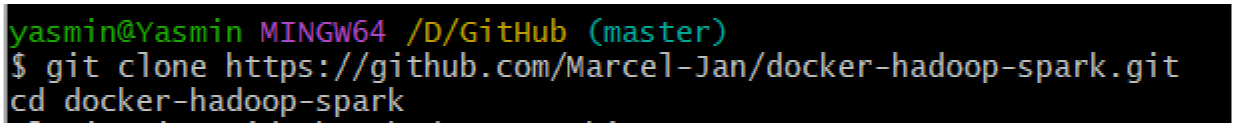
Project Overview

This project demonstrates how I set up a simple Hadoop and Hive big data pipeline using Docker. The goal is to store and query MIMIC-III clinical data (in Parquet format) using Hive on top of HDFS.

Step-by-Step Implementation

1. Clone the Docker Hadoop-Spark Project

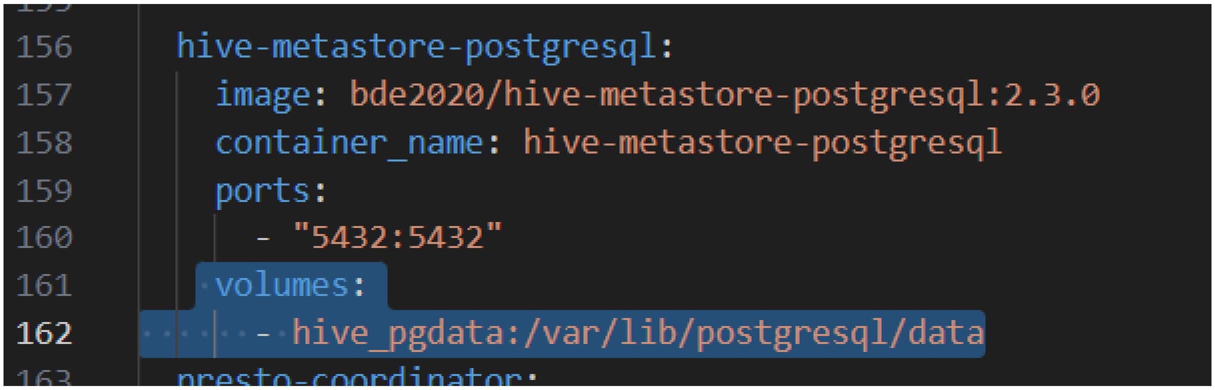
I used an existing GitHub repository to quickly spin up a Hadoop + Hive environment:



```
yasmin@Yasmin MINGW64 /D/GitHub (master)
$ git clone https://github.com/Marcel-Jan/docker-hadoop-spark.git
cd docker-hadoop-spark
```

2. Add Persistent Storage for Hive Metastore

To make sure Hive external tables don't get lost when restarting the containers, I modified the docker-compose.yml file to persist the PostgreSQL database used by Hive Metastore:

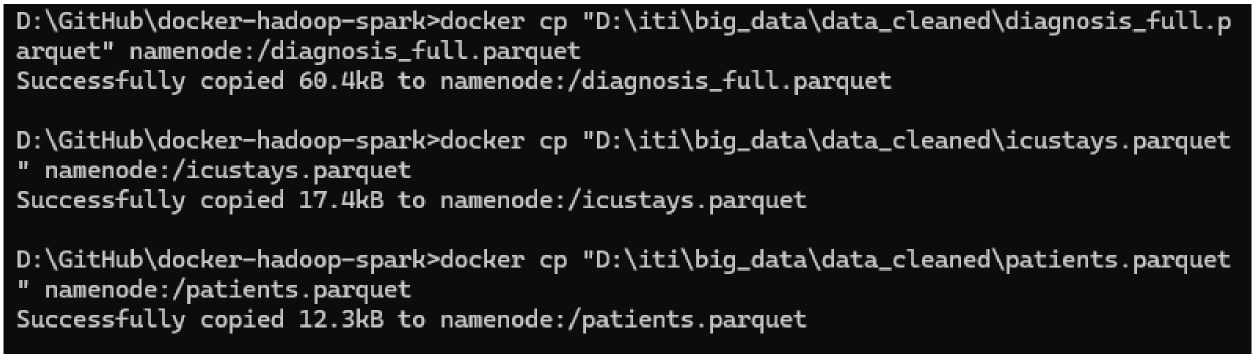


```
hive-metastore-postgresql:
  image: bde2020/hive-metastore-postgresql:2.3.0
  container_name: hive-metastore-postgresql
  ports:
    - "5432:5432"
  volumes:
    - hive_pgdata:/var/lib/postgresql/data
presto-coordinator:
```

---

3. Copy Data Files into Namenode

I copied the cleaned .parquet files from my local machine into the namenode container using docker cp:

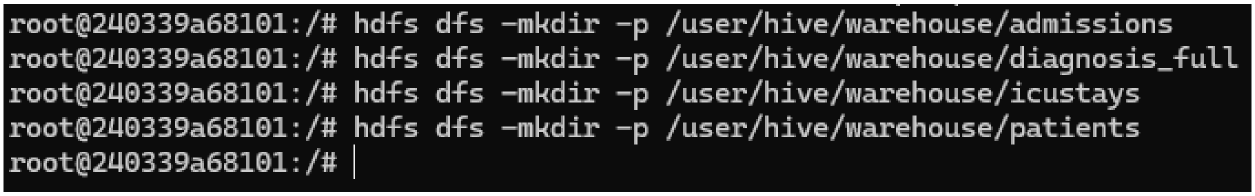


```
D:\GitHub\docker-hadoop-spark>docker cp "D:\iti\big_data\data_cleaned\diagnosis_full.p
arquet" namenode:/diagnosis_full.parquet
Successfully copied 60.4kB to namenode:/diagnosis_full.parquet

D:\GitHub\docker-hadoop-spark>docker cp "D:\iti\big_data\data_cleaned\icustays.parquet
" namenode:/icustays.parquet
Successfully copied 17.4kB to namenode:/icustays.parquet

D:\GitHub\docker-hadoop-spark>docker cp "D:\iti\big_data\data_cleaned\patients.parquet
" namenode:/patients.parquet
Successfully copied 12.3kB to namenode:/patients.parquet
```

(I did this for the diagnosis_full.parquet file as well)

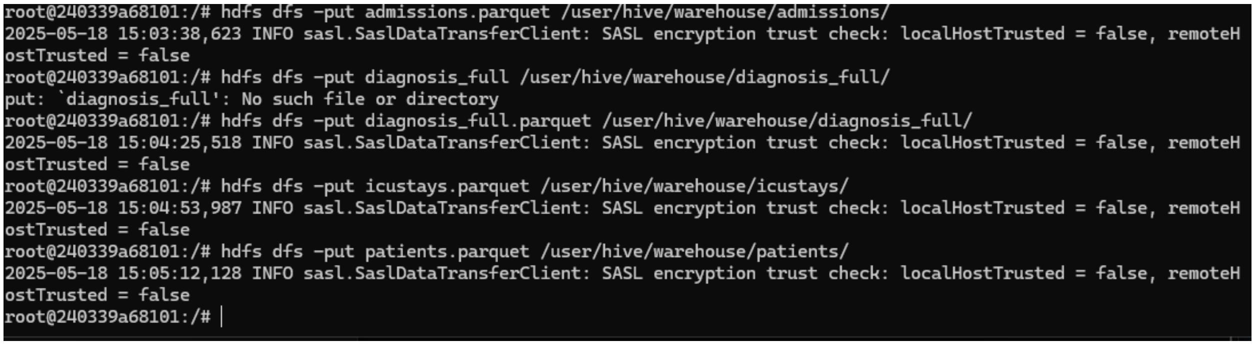
4. Upload Data to HDFS

First I created a directory for every file:



```
root@240339a68101:/# hdfs dfs -mkdir -p /user/hive/warehouse/admissions
root@240339a68101:/# hdfs dfs -mkdir -p /user/hive/warehouse/diagnosis_full
root@240339a68101:/# hdfs dfs -mkdir -p /user/hive/warehouse/icustays
root@240339a68101:/# hdfs dfs -mkdir -p /user/hive/warehouse/patients
root@240339a68101:/# |
```

Then I uploaded the files:



```
root@240339a68101:/# hdfs dfs -put admissions.parquet /user/hive/warehouse/admissions/
2025-05-18 15:03:38,623 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteH
ostTrusted = false
root@240339a68101:/# hdfs dfs -put diagnosis_full /user/hive/warehouse/diagnosis_full/
put: `diagnosis_full': No such file or directory
root@240339a68101:/# hdfs dfs -put diagnosis_full.parquet /user/hive/warehouse/diagnosis_full/
2025-05-18 15:04:25,518 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteH
ostTrusted = false
root@240339a68101:/# hdfs dfs -put icustays.parquet /user/hive/warehouse/icustays/
2025-05-18 15:04:53,987 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteH
ostTrusted = false
root@240339a68101:/# hdfs dfs -put patients.parquet /user/hive/warehouse/patients/
2025-05-18 15:05:12,128 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteH
ostTrusted = false
root@240339a68101:/# |
```

---

5. Create External Tables in Hive

I went inside hive-server container with the following code: “docker exec -it hive-server bash”, then I got connected to hive by just typing “hive”.

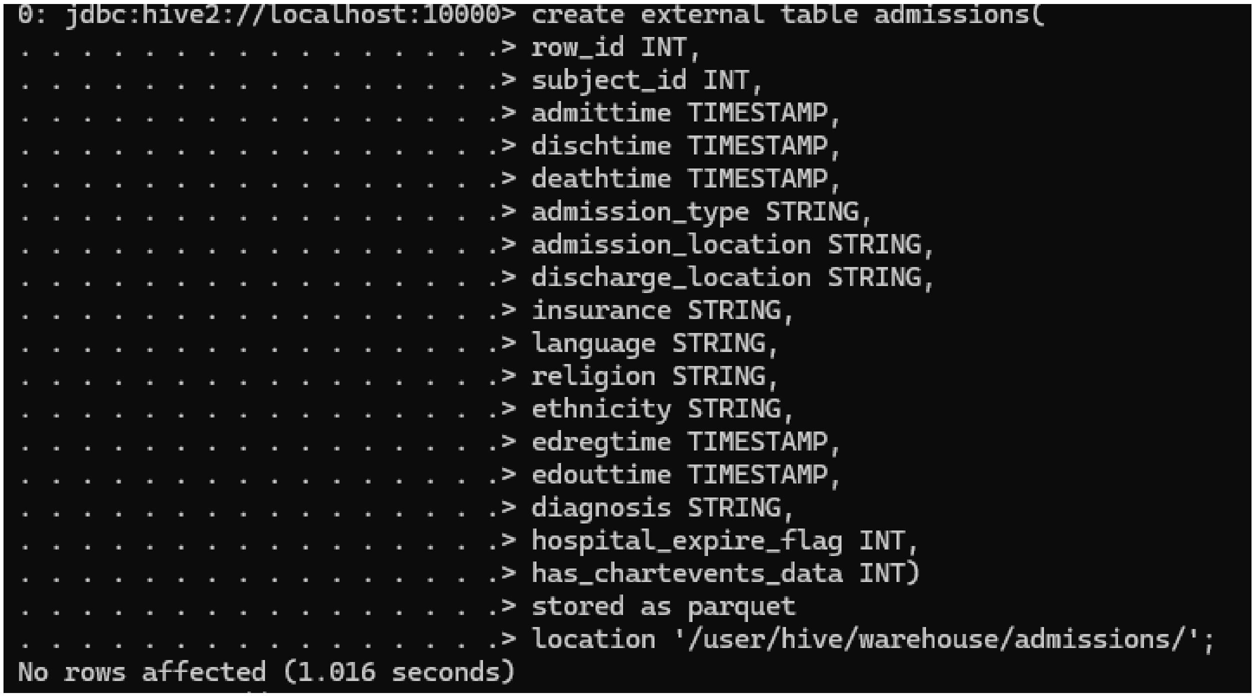
I created a database named mimic using the following code:

Create database mimic;

Use mimic;

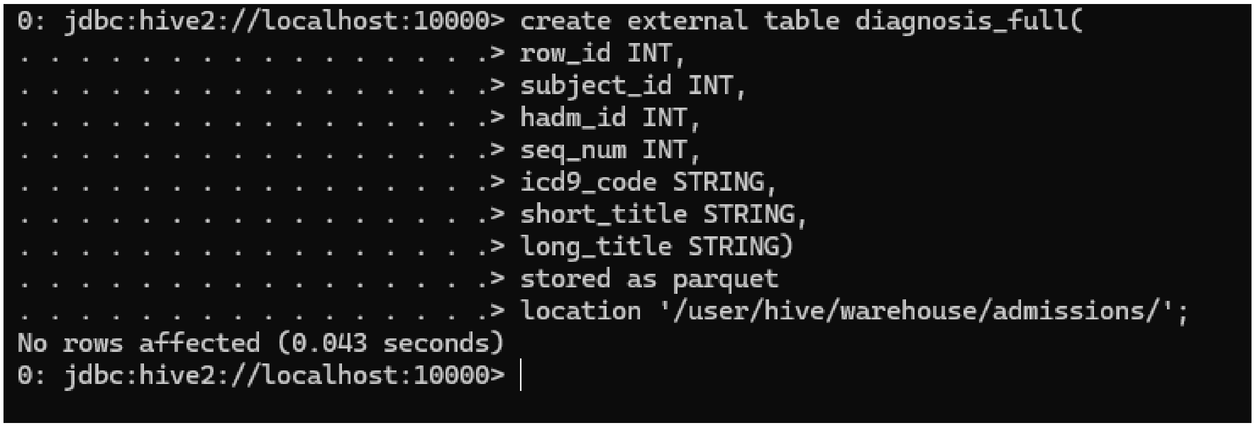
Now we can starting creating our tables.

Admissions Table:



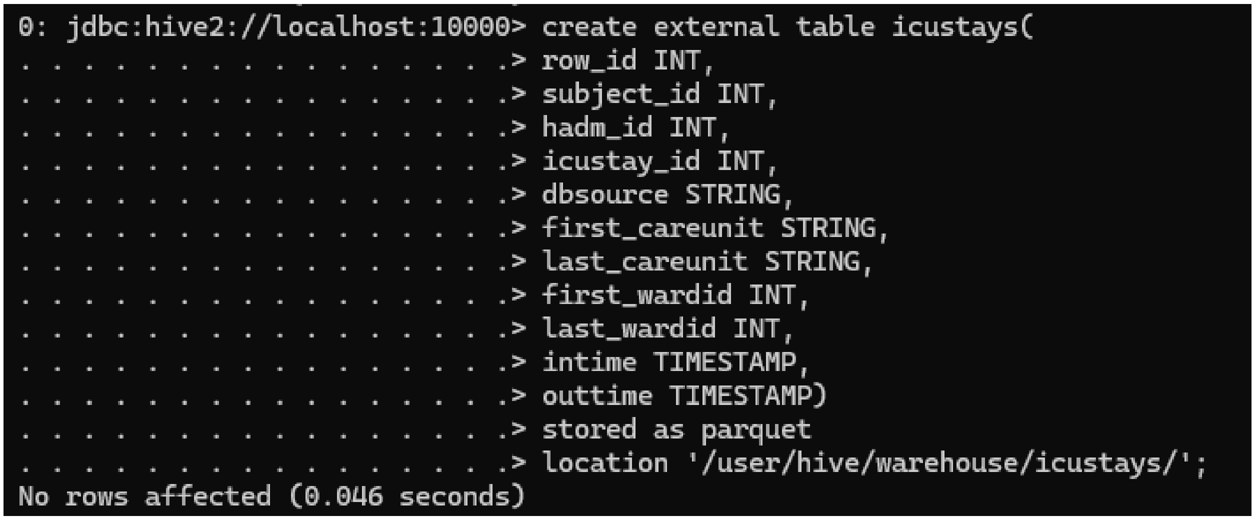
```
0: jdbc:hive2://localhost:10000> create external table admissions(
. . . . . . . . . . . . . . . .> row_id INT,
. . . . . . . . . . . . . . . .> subject_id INT,
. . . . . . . . . . . . . . . .> admittime TIMESTAMP,
. . . . . . . . . . . . . . . .> dischtime TIMESTAMP,
. . . . . . . . . . . . . . . .> deathtime TIMESTAMP,
. . . . . . . . . . . . . . . .> admission_type STRING,
. . . . . . . . . . . . . . . .> admission_location STRING,
. . . . . . . . . . . . . . . .> discharge_location STRING,
. . . . . . . . . . . . . . . .> insurance STRING,
. . . . . . . . . . . . . . . .> language STRING,
. . . . . . . . . . . . . . . .> religion STRING,
. . . . . . . . . . . . . . . .> ethnicity STRING,
. . . . . . . . . . . . . . . .> edregtime TIMESTAMP,
. . . . . . . . . . . . . . . .> edouttime TIMESTAMP,
. . . . . . . . . . . . . . . .> diagnosis STRING,
. . . . . . . . . . . . . . . .> hospital_expire_flag INT,
. . . . . . . . . . . . . . . .> has_chartevents_data INT)
. . . . . . . . . . . . . . . .> stored as parquet
. . . . . . . . . . . . . . . .> location '/user/hive/warehouse/admissions/';
No rows affected (1.016 seconds)
```

Diagnosis_full Table:



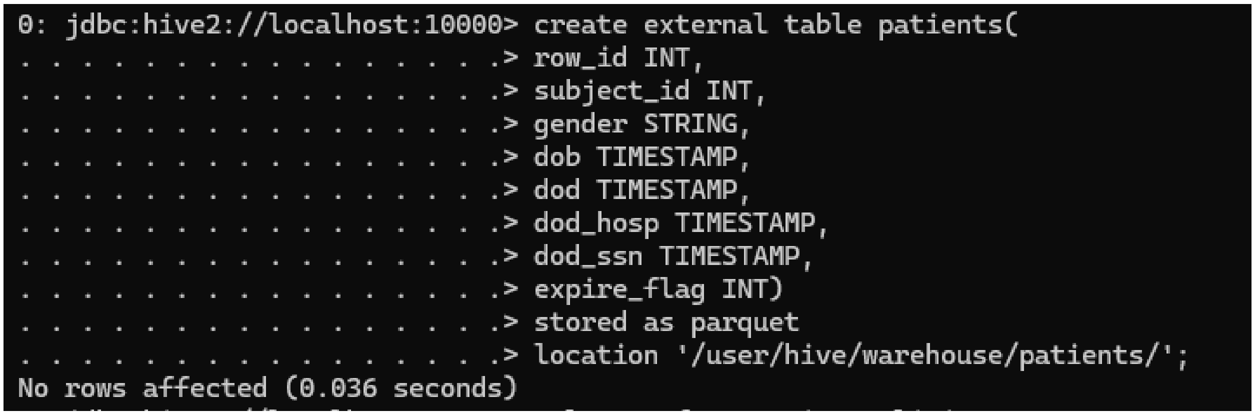
```
0: jdbc:hive2://localhost:10000> create external table diagnosis_full(
. . . . . . . . . . . . . . . .> row_id INT,
. . . . . . . . . . . . . . . .> subject_id INT,
. . . . . . . . . . . . . . . .> hadm_id INT,
. . . . . . . . . . . . . . . .> seq_num INT,
. . . . . . . . . . . . . . . .> icd9_code STRING,
. . . . . . . . . . . . . . . .> short_title STRING,
. . . . . . . . . . . . . . . .> long_title STRING)
. . . . . . . . . . . . . . . .> stored as parquet
. . . . . . . . . . . . . . . .> location '/user/hive/warehouse/admissions/';
No rows affected (0.043 seconds)
0: jdbc:hive2://localhost:10000>
```

icustays Table:



```
0: jdbc:hive2://localhost:10000> create external table icustays(
. . . . . . . . . . . . . . . . . . . .> row_id INT,
. . . . . . . . . . . . . . . . . . . .> subject_id INT,
. . . . . . . . . . . . . . . . . . . .> hadm_id INT,
. . . . . . . . . . . . . . . . . . . .> icustay_id INT,
. . . . . . . . . . . . . . . . . . . .> dbsource STRING,
. . . . . . . . . . . . . . . . . . . .> first_careunit STRING,
. . . . . . . . . . . . . . . . . . . .> last_careunit STRING,
. . . . . . . . . . . . . . . . . . . .> first_wardid INT,
. . . . . . . . . . . . . . . . . . . .> last_wardid INT,
. . . . . . . . . . . . . . . . . . . .> intime TIMESTAMP,
. . . . . . . . . . . . . . . . . . . .> outtime TIMESTAMP)
. . . . . . . . . . . . . . . . . . . .> stored as parquet
. . . . . . . . . . . . . . . . . . . .> location '/user/hive/warehouse/icustays/';
No rows affected (0.046 seconds)
```

Patients Table:



```
0: jdbc:hive2://localhost:10000> create external table patients(
. . . . . . . . . . . . . . . . . . . .> row_id INT,
. . . . . . . . . . . . . . . . . . . .> subject_id INT,
. . . . . . . . . . . . . . . . . . . .> gender STRING,
. . . . . . . . . . . . . . . . . . . .> dob TIMESTAMP,
. . . . . . . . . . . . . . . . . . . .> dod TIMESTAMP,
. . . . . . . . . . . . . . . . . . . .> dod_hosp TIMESTAMP,
. . . . . . . . . . . . . . . . . . . .> dod_ssn TIMESTAMP,
. . . . . . . . . . . . . . . . . . . .> expire_flag INT)
. . . . . . . . . . . . . . . . . . . .> stored as parquet
. . . . . . . . . . . . . . . . . . . .> location '/user/hive/warehouse/patients/';
No rows affected (0.036 seconds)
```

Now that I have my tables ready, I can start querying.

---

6. Querying

1st Question: Average Length per Diagnosis:

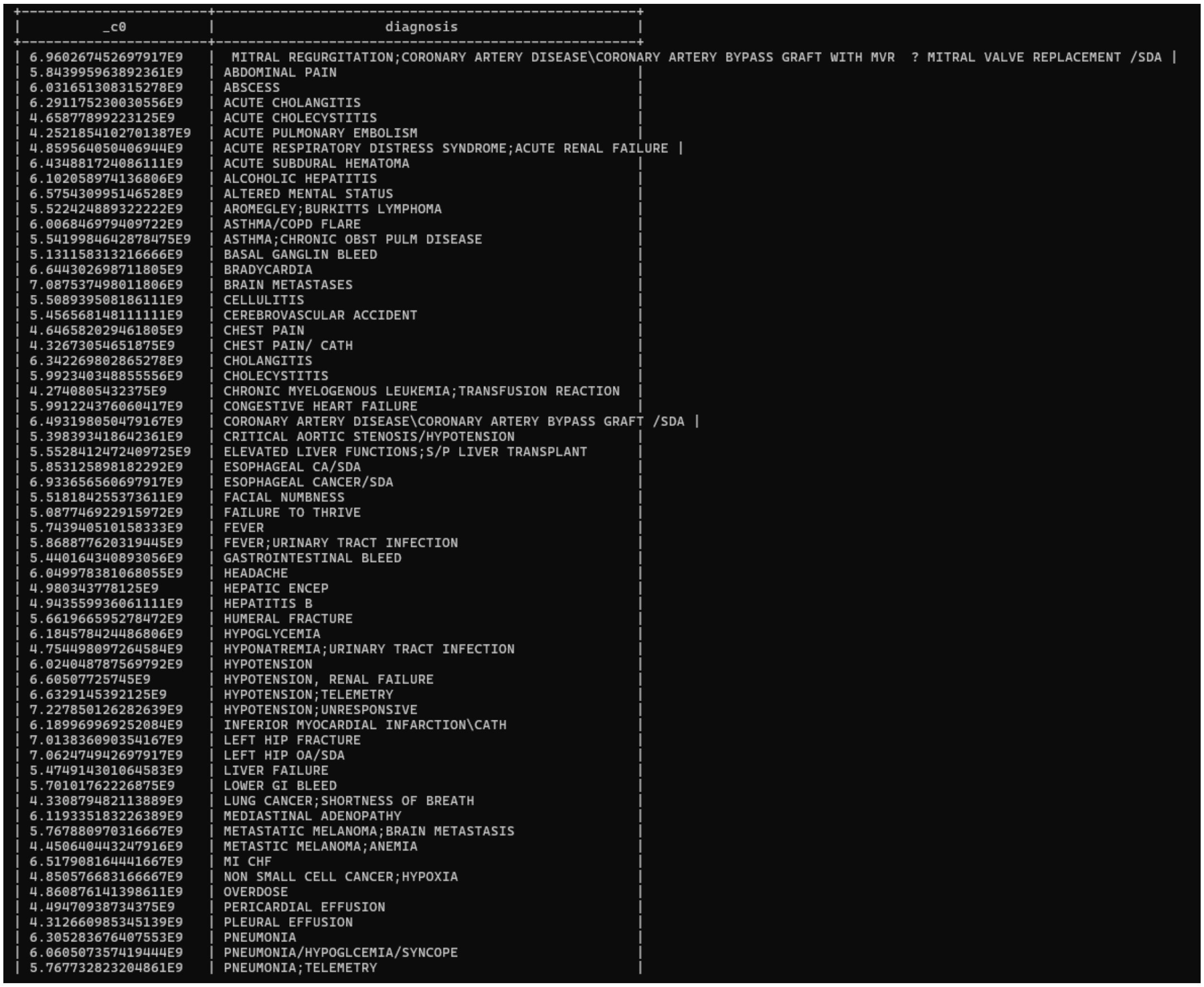
I ran the following query:

```
SELECT
  AVG(UNIX_TIMESTAMP(dischtime) - UNIX_TIMESTAMP(admittime)) / 86400 AS avg_los,
  diagnosis
```

FROM admissions

GROUP BY diagnosis;

Output:



```
+-----------------------+-----------------------------------------------------+
|          _c0          |                      diagnosis                      |
+-----------------------+-----------------------------------------------------+
| 6.960267452697917E9   |  MITRAL REGURGITATION;CORONARY ARTERY DISEASE\CORONARY ARTERY BYPASS GRAFT WITH MVR  ? MITRAL VALVE REPLACEMENT /SDA |
| 5.843995963892361E9   | ABDOMINAL PAIN                                      |
| 6.031651308315278E9   | ABSCESS                                             |
| 6.291175230030556E9   | ACUTE CHOLANGITIS                                   |
| 4.65877899223125E9    | ACUTE CHOLECYSTITIS                                 |
| 4.2521854102701387E9  | ACUTE PULMONARY EMBOLISM                            |
| 4.859564050406944E9   | ACUTE RESPIRATORY DISTRESS SYNDROME;ACUTE RENAL FAILURE |
| 6.434881724086111E9   | ACUTE SUBDURAL HEMATOMA                             |
| 6.102058974136806E9   | ALCOHOLIC HEPATITIS                                 |
| 6.575430995146528E9   | ALTERED MENTAL STATUS                               |
| 5.522424889322222E9   | AROMEGLEY;BURKITTS LYMPHOMA                         |
| 6.006846979409722E9   | ASTHMA/COPD FLARE                                   |
| 5.5419984642878475E9  | ASTHMA;CHRONIC OBST PULM DISEASE                    |
| 5.131158313216666E9   | BASAL GANGLIN BLEED                                 |
| 6.644302698711805E9   | BRADYCARDIA                                         |
| 7.087537498011806E9   | BRAIN METASTASES                                    |
| 5.508939508186111E9   | CELLULITIS                                          |
| 5.456568148111111E9   | CEREBROVASCULAR ACCIDENT                            |
| 4.646582029461805E9   | CHEST PAIN                                          |
| 4.32673054651875E9    | CHEST PAIN/ CATH                                    |
| 6.342269802865278E9   | CHOLANGITIS                                         |
| 5.992340348855556E9   | CHOLECYSTITIS                                       |
| 4.2740805432375E9     | CHRONIC MYELOGENOUS LEUKEMIA;TRANSFUSION REACTION   |
| 5.991224376060417E9   | CONGESTIVE HEART FAILURE                            |
| 6.493198050479167E9   | CORONARY ARTERY DISEASE\CORONARY ARTERY BYPASS GRAFT /SDA |
| 5.398393418642361E9   | CRITICAL AORTIC STENOSIS/HYPOTENSION                |
| 5.5528412472409725E9  | ELEVATED LIVER FUNCTIONS;S/P LIVER TRANSPLANT       |
| 5.853125898182292E9   | ESOPHAGEAL CA/SDA                                   |
| 6.933656560697917E9   | ESOPHAGEAL CANCER/SDA                               |
| 5.518184255373611E9   | FACIAL NUMBNESS                                     |
| 5.087746922915972E9   | FAILURE TO THRIVE                                   |
| 5.743940510158333E9   | FEVER                                               |
| 5.868877620319445E9   | FEVER;URINARY TRACT INFECTION                       |
| 5.440164340893056E9   | GASTROINTESTINAL BLEED                              |
| 6.049978381068055E9   | HEADACHE                                            |
| 4.980343778125E9      | HEPATIC ENCEP                                       |
| 4.943559936061111E9   | HEPATITIS B                                         |
| 5.661966595278472E9   | HUMERAL FRACTURE                                    |
| 6.184578424486806E9   | HYPOGLYCEMIA                                        |
| 4.754498097264584E9   | HYPONATREMIA;URINARY TRACT INFECTION                |
| 6.024048787569792E9   | HYPOTENSION                                         |
| 6.60507725745E9       | HYPOTENSION, RENAL FAILURE                          |
| 6.6329145392125E9     | HYPOTENSION;TELEMETRY                               |
| 7.227850126282639E9   | HYPOTENSION;UNRESPONSIVE                            |
| 6.189969969252084E9   | INFERIOR MYOCARDIAL INFARCTION\CATH                 |
| 7.013836090354167E9   | LEFT HIP FRACTURE                                   |
| 7.062474942697917E9   | LEFT HIP OA/SDA                                     |
| 5.474914301064583E9   | LIVER FAILURE                                       |
| 5.70101762226875E9    | LOWER GI BLEED                                      |
| 4.330879482113889E9   | LUNG CANCER;SHORTNESS OF BREATH                     |
| 6.119335183226389E9   | MEDIASTINAL ADENOPATHY                              |
| 5.767880970316667E9   | METASTATIC MELANOMA;BRAIN METASTASIS                |
| 4.450640443247916E9   | METASTIC MELANOMA;ANEMIA                            |
| 6.517908164441667E9   | MI CHF                                              |
| 4.850576683166667E9   | NON SMALL CELL CANCER;HYPOXIA                       |
| 4.860876141398611E9   | OVERDOSE                                            |
| 4.49470938734375E9    | PERICARDIAL EFFUSION                                |
| 4.312660985345139E9   | PLEURAL EFFUSION                                    |
| 6.305283676407553E9   | PNEUMONIA                                           |
| 6.060507357419444E9   | PNEUMONIA/HYPOGLCEMIA/SYNCOPE                       |
| 5.767732823204861E9   | PNEUMONIA;TELEMETRY                                 |
```

Question 2: Distribution of ICU readmissions:

Query:

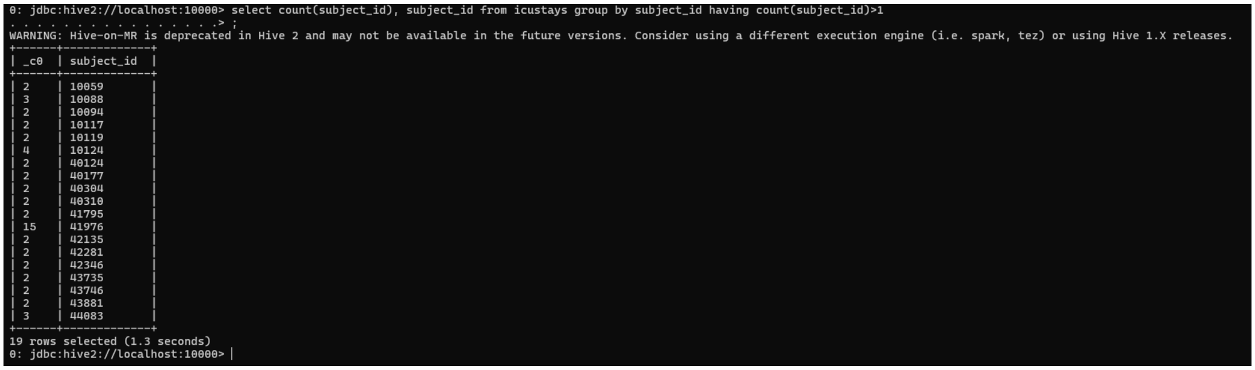
SELECT COUNT(subject_id), subject_id

FROM icustays

GROUP BY subject_id

HAVING COUNT(subject_id) > 1;

Output:



```
0: jdbc:hive2://localhost:10000> select count(subject_id), subject_id from icustays group by subject_id having count(subject_id)>1
. . . . . . . . . . . . . . . . .> ;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
+------+-------------+
| _c0  | subject_id  |
+------+-------------+
| 2    | 10059       |
| 3    | 10088       |
| 2    | 10094       |
| 2    | 10117       |
| 2    | 10119       |
| 4    | 10124       |
| 2    | 40124       |
| 2    | 40177       |
| 2    | 40304       |
| 2    | 40310       |
| 2    | 41795       |
| 15   | 41976       |
| 2    | 42135       |
| 2    | 42281       |
| 2    | 42346       |
| 2    | 43735       |
| 2    | 43746       |
| 2    | 43881       |
| 3    | 44083       |
+------+-------------+
19 rows selected (1.3 seconds)
0: jdbc:hive2://localhost:10000> |
```

3rd Question: Mortality rates by demographic groups:

Query:

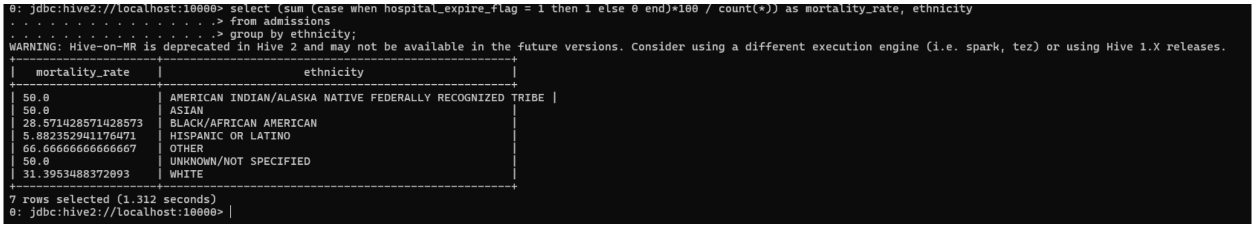
SELECT

(SUM(CASE WHEN hospital_expire_flag = 1 THEN 1 ELSE 0 END) * 100 / COUNT(*)) AS mortality_rate,

ethnicity

FROM admissions

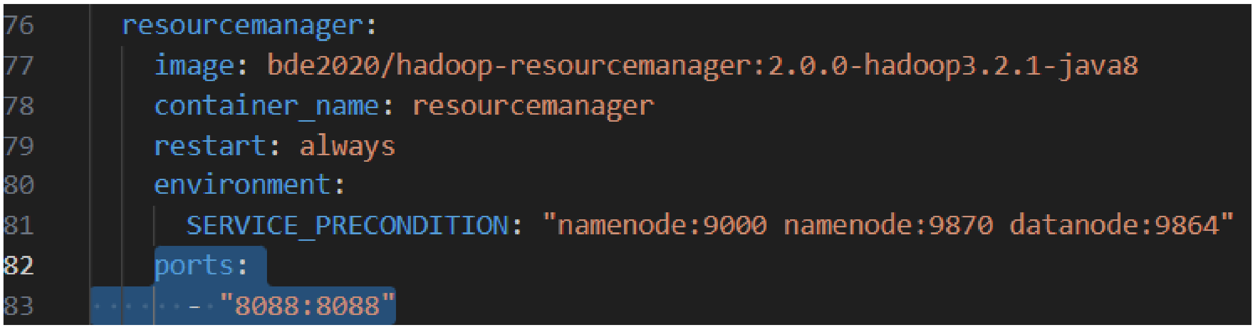
GROUP BY ethnicity;



```
0: jdbc:hive2://localhost:10000> select (sum (case when hospital_expire_flag = 1 then 1 else 0 end)*100 / count(*)) as mortality_rate, ethnicity
. . . . . . . . . . . . . . . . .> from admissions
. . . . . . . . . . . . . . . . .> group by ethnicity;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
+---------------------+----------------------------------------------------+
|   mortality_rate    |                     ethnicity                      |
+---------------------+----------------------------------------------------+
| 50.0                | AMERICAN INDIAN/ALASKA NATIVE FEDERALLY RECOGNIZED TRIBE |
| 50.0                | ASIAN                                              |
| 28.571428571428573  | BLACK/AFRICAN AMERICAN                             |
| 5.882352941176471   | HISPANIC OR LATINO                                 |
| 66.66666666666667   | OTHER                                              |
| 50.0                | UNKNOWN/NOT SPECIFIED                              |
| 31.3953488372093    | WHITE                                              |
+---------------------+----------------------------------------------------+
7 rows selected (1.312 seconds)
0: jdbc:hive2://localhost:10000> |
```

YARN Web UI Setup

To access the YARN ResourceManager on the web, I edited the docker-compose.yml file and added the ports part:

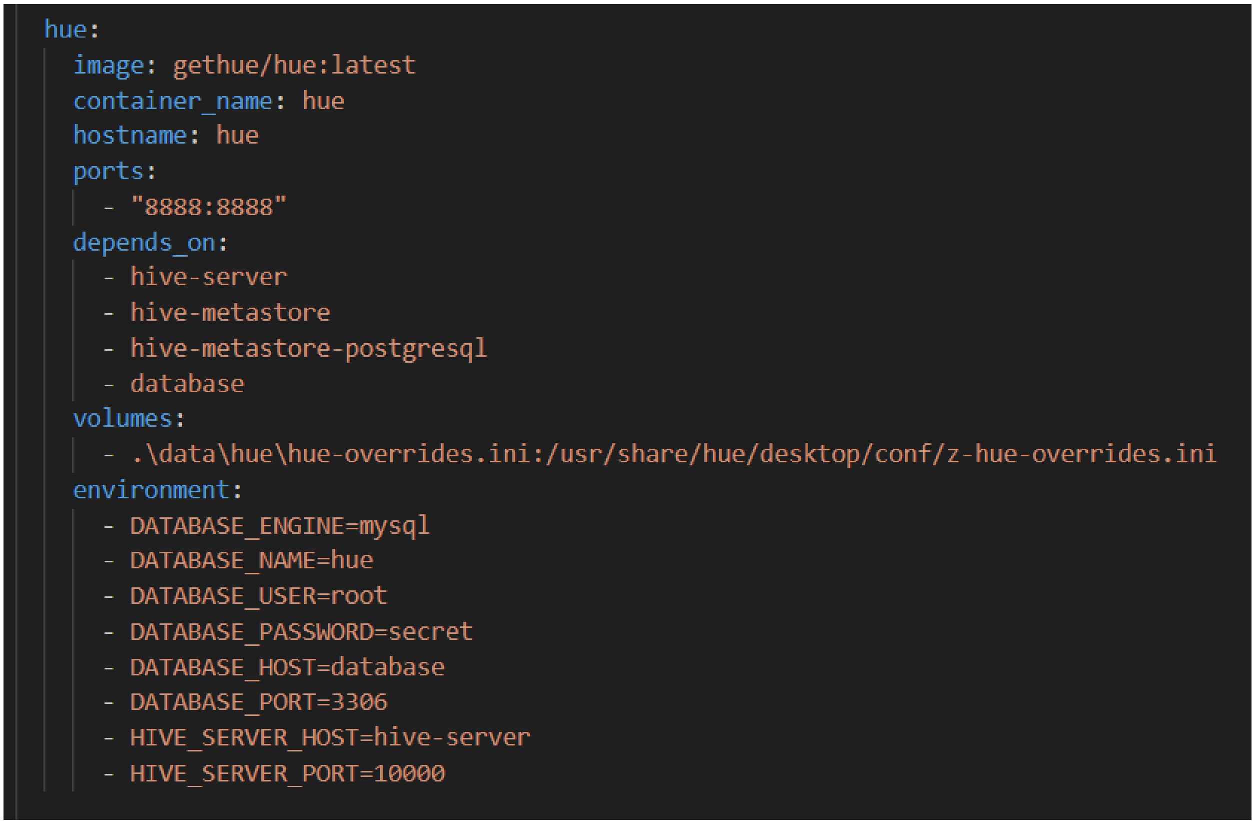


```yaml
  resourcemanager:
    image: bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8
    container_name: resourcemanager
    restart: always
    environment:
      SERVICE_PRECONDITION: "namenode:9000 namenode:9870 datanode:9864"
    ports:
      - "8088:8088"
```

I also checked the yarn-site.xml config inside the resourcemanager container to make sure it's binding to 0.0.0.0:8088.

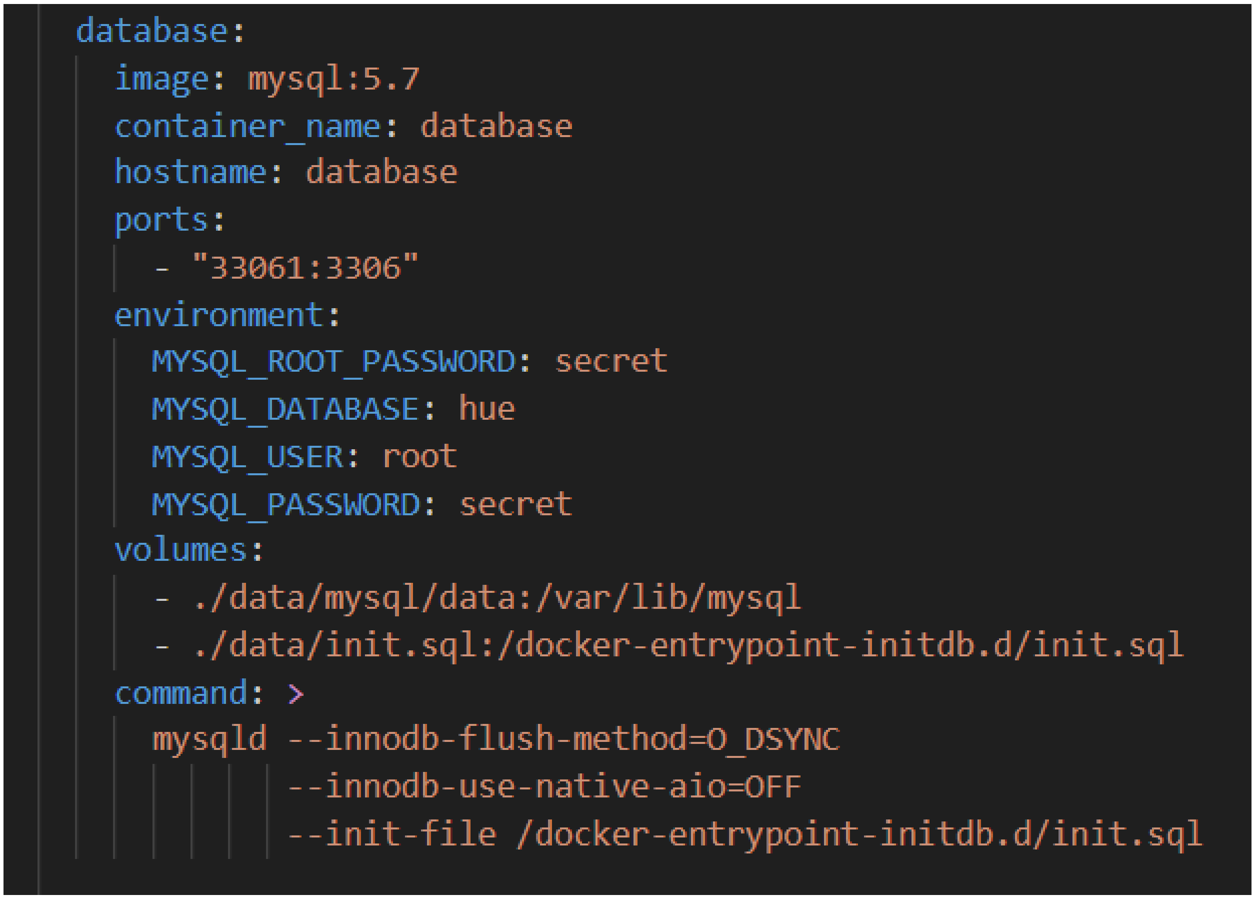
Hue Web UI Setup for Hive

To install Hue and connect it with Hive via Web UI, I added the following to my docker-compose.yml:



```yaml
  hue:
    image: gethue/hue:latest
    container_name: hue
    hostname: hue
    ports:
      - "8888:8888"
    depends_on:
      - hive-server
      - hive-metastore
      - hive-metastore-postgresql
      - database
    volumes:
      - .\data\hue\hue-overrides.ini:/usr/share/hue/desktop/conf/z-hue-overrides.ini
    environment:
      - DATABASE_ENGINE=mysql
      - DATABASE_NAME=hue
      - DATABASE_USER=root
      - DATABASE_PASSWORD=secret
      - DATABASE_HOST=database
      - DATABASE_PORT=3306
      - HIVE_SERVER_HOST=hive-server
      - HIVE_SERVER_PORT=10000
```

And for the database:



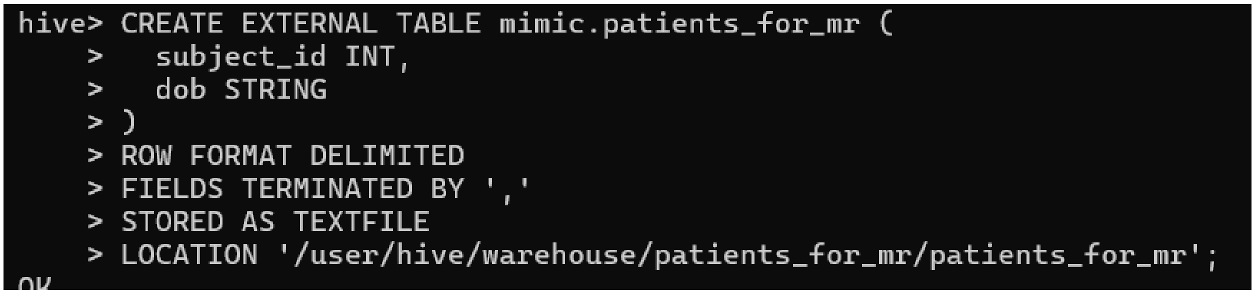
```
database:
  image: mysql:5.7
  container_name: database
  hostname: database
  ports:
    - "33061:3306"
  environment:
    MYSQL_ROOT_PASSWORD: secret
    MYSQL_DATABASE: hue
    MYSQL_USER: root
    MYSQL_PASSWORD: secret
  volumes:
    - ./data/mysql/data:/var/lib/mysql
    - ./data/init.sql:/docker-entrypoint-initdb.d/init.sql
  command: >
    mysqld --innodb-flush-method=O_DSYNC
           --innodb-use-native-aio=OFF
           --init-file /docker-entrypoint-initdb.d/init.sql
```

Currently, I can see the tables in the Hue Web UI, but queries result in a "database is locked" error. Likely because Hue is not properly connected to the MySQL backend.

MapReduce Job: Calculate Average Age

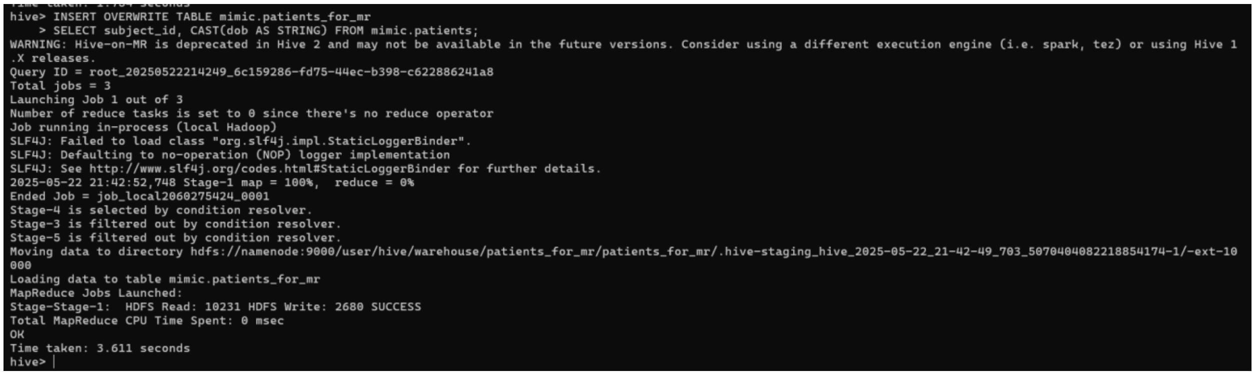
1. **Create a Table for MapReduce**

I created a table formatted in a way that map reduce understands



```
hive> CREATE EXTERNAL TABLE mimic.patients_for_mr (
    >   subject_id INT,
    >   dob STRING
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > STORED AS TEXTFILE
    > LOCATION '/user/hive/warehouse/patients_for_mr/patients_for_mr';
OK
```

2. Insert the Data

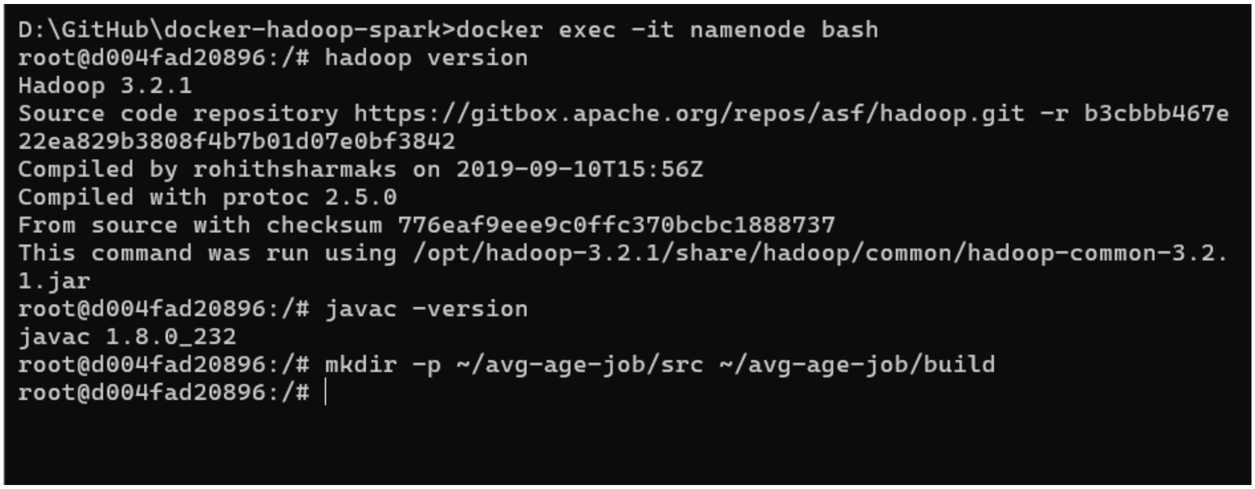


```
hive> INSERT OVERWRITE TABLE mimic.patients_for_mr
    > SELECT subject_id, CAST(dob AS STRING) FROM mimic.patients;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1
.X releases.
Query ID = root_20250522214249_6c159286-fd75-44ec-b398-c622886241a8
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Job running in-process (local Hadoop)
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
2025-05-22 21:42:52,748 Stage-1 map = 100%,  reduce = 0%
Ended Job = job_local2060275424_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://namenode:9000/user/hive/warehouse/patients_for_mr/patients_for_mr/.hive-staging_hive_2025-05-22_21-42-49_703_5070404082218854174-1/-ext-10
000
Loading data to table mimic.patients_for_mr
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 10231 HDFS Write: 2680 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 3.611 seconds
hive> 
```

2. Java Setup for MapReduce

I made sure first that java is available in the namenode container then I created 2 directories; 1 for the mapper and the other for the reducer.

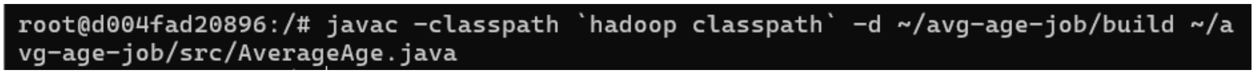
Inside the namenode:



```
D:\GitHub\docker-hadoop-spark>docker exec -it namenode bash
root@d004fad20896:/# hadoop version
Hadoop 3.2.1
Source code repository https://gitbox.apache.org/repos/asf/hadoop.git -r b3cbbb467e
22ea829b3808f4b7b01d07e0bf3842
Compiled by rohithsharmaks on 2019-09-10T15:56Z
Compiled with protoc 2.5.0
From source with checksum 776eaf9eee9c0ffc370bcbc1888737
This command was run using /opt/hadoop-3.2.1/share/hadoop/common/hadoop-common-3.2.
1.jar
root@d004fad20896:/# javac -version
javac 1.8.0_232
root@d004fad20896:/# mkdir -p ~/avg-age-job/src ~/avg-age-job/build
root@d004fad20896:/# 
```

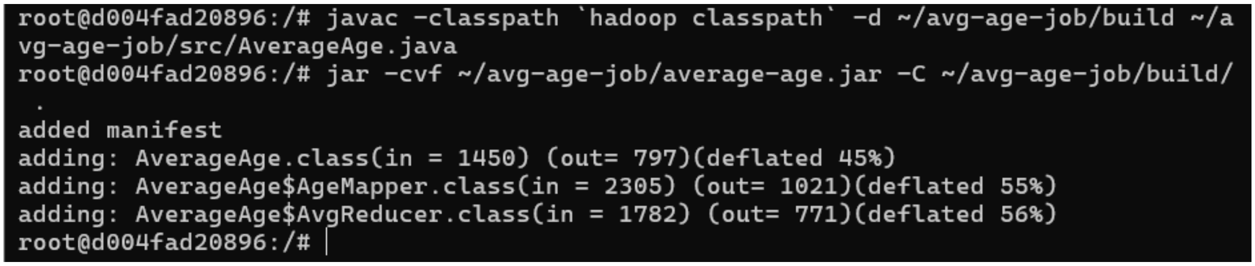
4. Compile and Package the Job

After putting the code into the files I created:



```
root@d004fad20896:/# javac -classpath `hadoop classpath` -d ~/avg-age-job/build ~/a
vg-age-job/src/AverageAge.java
```

5. Run the Job



```
root@d004fad20896:/# javac -classpath `hadoop classpath` -d ~/avg-age-job/build ~/a
vg-age-job/src/AverageAge.java
root@d004fad20896:/# jar -cvf ~/avg-age-job/average-age.jar -C ~/avg-age-job/build/
  .
added manifest
adding: AverageAge.class(in = 1450) (out= 797)(deflated 45%)
adding: AverageAge$AgeMapper.class(in = 2305) (out= 1021)(deflated 55%)
adding: AverageAge$AvgReducer.class(in = 1782) (out= 771)(deflated 56%)
root@d004fad20896:/# 
```

Summary

I now have a working Dockerized Hadoop + Hive environment:

- **Data** stored in HDFS (as Parquet)
- **Hive** external tables mapped to that data
- **Queries** run directly in Hive
- **Hue** as a GUI (with issues to fix)
- **YARN** accessible via browser
- **MapReduce job** successfully compiled and executed

Tools Used

- Docker
- Hadoop (HDFS)
- Hive + Hive Metastore
- PostgreSQL / MySQL
- Parquet files
- Hue Web UI
- Java (MapReduce)

Credits

Based on: https://github.com/Marcel-Jan/docker-hadoop-spark

Problems Faced

1. **Parquet-Hive Compatibility Issue**
   When I first tried to load the Parquet files into Hive, I encountered an error related to data types — Hive didn't support the timestamp format in the Parquet files generated by Pandas.

I initially tried to cast everything to string and later convert it to TIMESTAMP, but this didn't work properly. Hive kept showing me that the field was not compatible with timestamp. I wanted to keep datetime columns in proper format, so I fixed the problem by modifying my Parquet export code in Python:

I used the pyarrow engine and added:

```
use_deprecated_int96_timestamps=True
```

This allowed Hive to recognize the timestamp columns correctly.
