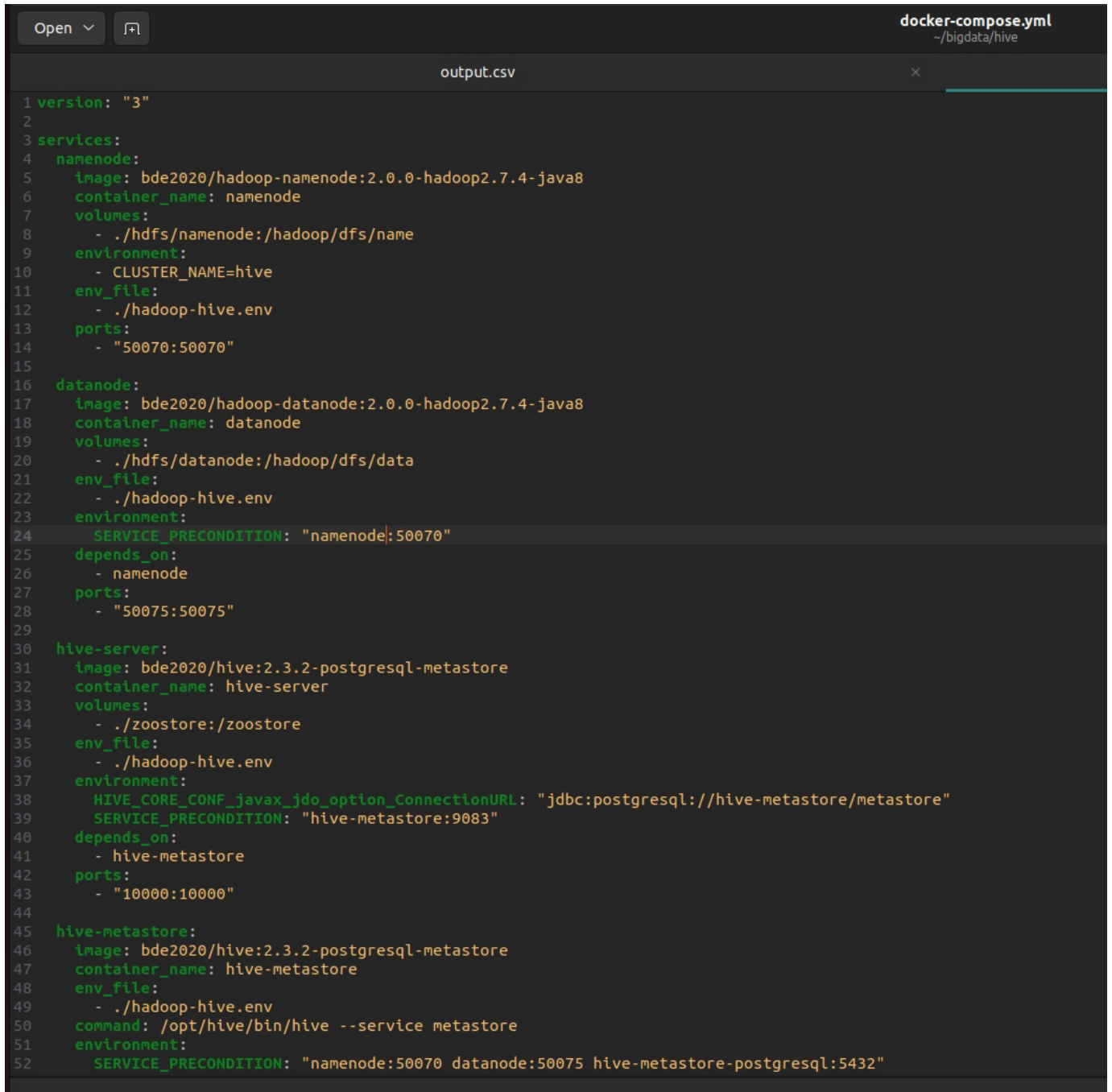
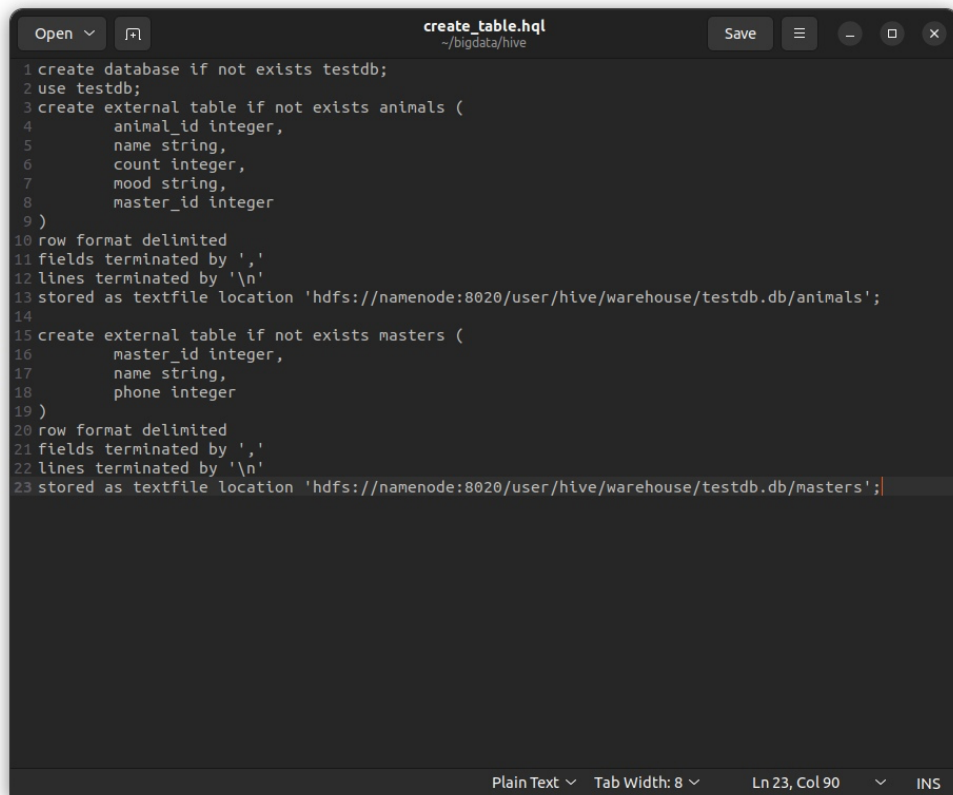


Отчет по лабораторной работе № 2



```
1 version: "3"
2
3 services:
4   namenode:
5     image: bde2020/hadoop-namenode:2.0.0-hadoop2.7.4-java8
6     container_name: namenode
7     volumes:
8       - ./hdfs/namenode:/hadoop/dfs/name
9     environment:
10      - CLUSTER_NAME=hive
11     env_file:
12      - ./hadoop-hive.env
13     ports:
14      - "50070:50070"
15
16   datanode:
17     image: bde2020/hadoop-datanode:2.0.0-hadoop2.7.4-java8
18     container_name: datanode
19     volumes:
20       - ./hdfs/datanode:/hadoop/dfs/data
21     env_file:
22       - ./hadoop-hive.env
23     environment:
24       SERVICE_PRECONDITION: "namenode:50070"
25     depends_on:
26       - namenode
27     ports:
28       - "50075:50075"
29
30   hive-server:
31     image: bde2020/hive:2.3.2-postgresql-metastore
32     container_name: hive-server
33     volumes:
34       - ./zoostore:/zoostore
35     env_file:
36       - ./hadoop-hive.env
37     environment:
38       HIVE_CORE_CONF_javax_jdo_option_ConnectionURL: "jdbc:postgresql://hive-metastore/metastore"
39       SERVICE_PRECONDITION: "hive-metastore:9083"
40     depends_on:
41       - hive-metastore
42     ports:
43       - "10000:10000"
44
45   hive-metastore:
46     image: bde2020/hive:2.3.2-postgresql-metastore
47     container_name: hive-metastore
48     env_file:
49       - ./hadoop-hive.env
50     command: /opt/hive/bin/hive --service metastore
51     environment:
52       SERVICE_PRECONDITION: "namenode:50070 datanode:50075 hive-metastore-postgresql:5432"
```

Рис. 1 – конфигурация файла docker-compose.yml

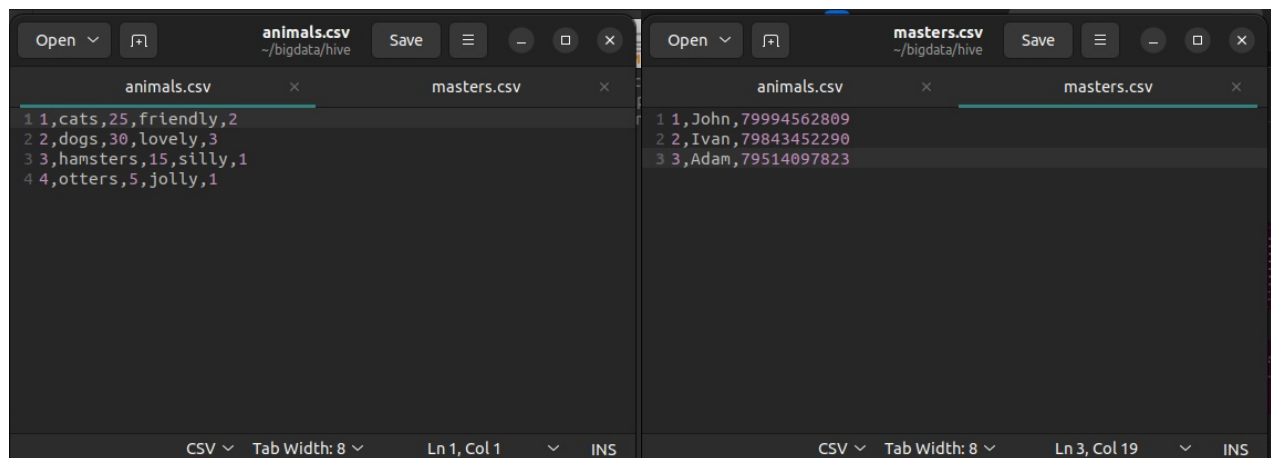


The screenshot shows a code editor window titled "create_table.hql" with the following HQL script:

```
1 create database if not exists testdb;
2 use testdb;
3 create external table if not exists animals (
4     animal_id integer,
5     name string,
6     count integer,
7     mood string,
8     master_id integer
9 )
10 row format delimited
11 fields terminated by ','
12 lines terminated by '\n'
13 stored as textfile location 'hdfs://namenode:8020/user/hive/warehouse/testdb.db/animals';
14
15 create external table if not exists masters (
16     master_id integer,
17     name string,
18     phone integer
19 )
20 row format delimited
21 fields terminated by ','
22 lines terminated by '\n'
23 stored as textfile location 'hdfs://namenode:8020/user/hive/warehouse/testdb.db/masters';
```

The status bar at the bottom indicates "Plain Text", "Tab Width: 8", "Ln 23, Col 90", and "INS".

Рис. 2 – скрипт hql для создания базы данных и hive



The image shows two side-by-side screenshots of CSV files. The left screenshot shows the "animals.csv" file with the following content:

```
1 1,cats,25,friendly,2
2 2,dogs,30,lovely,3
3 3,hamsters,15,silly,1
4 4,otters,5,jolly,1
```

The right screenshot shows the "masters.csv" file with the following content:

```
1 1,John,79994562809
2 2,Ivan,79843452290
3 3,Adam,79514097823
```

Both screenshots show the status bar at the bottom indicating "CSV", "Tab Width: 8", and the current line and column numbers.

Рис. 3, 4 – содержание файлов animals.csv и masters.csv

```

lratwarrior@lratwarrior-0P156:~/bigdata/hive$ ls
animals.csv create_table.hql docker-compose.yml hadoop-hive.env hdfs masters.csv metastore-postgresql zoostore
lratwarrior@lratwarrior-0P156:~/bigdata/hive$ sudo docker ps -a
[sudo] password for lratwarrior:
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAME
5             bde2020/hive-metastore-postgresql:2.3.0 "/docker-entrypoint..." 41 minutes ago Up 41 minutes 5432/tcp                           hive
metastore-postgresql
6b6517e9b2bd   bde2020/hadoop-datanode:2.0.0-hadoop2.7.4-java8 "/entrypoint.sh /run..." 41 minutes ago Up 41 minutes (healthy) 0.0.0.0:50075->50075/tcp, :::50075->50075/tcp data
node
6b464f28d1cd   bde2020/hive:2.3.2-postgresql-metastore "entrypoint.sh /bin/..." 41 minutes ago Up 41 minutes 0.0.0.0:10000->10000/tcp, :::10000->10000/tcp, 10002/tcp hive
hive
65fb15c3ad74   bde2020/hive:2.3.2-postgresql-metastore "entrypoint.sh /opt/..." 41 minutes ago Up 41 minutes 10000/tcp, 0.0.0.0:9083->9083/tcp, :::9083->9083/tcp, 10002/tcp hive
hive
59ca84fae401   bde2020/hadoop-namenode:2.0.0-hadoop2.7.4-java8 "/entrypoint.sh /run..." 41 minutes ago Up 41 minutes (healthy) 0.0.0.0:50070->50070/tcp, :::50070->50070/tcp name
node
60b9284fac42   hello-world                  "/hello"                 17 hours ago   Exited (0) 17 hours ago
wiza
rdly_bose

lratwarrior@lratwarrior-0P156:~/bigdata/hive$ sudo docker cp create_table.hql hive-server
must specify at least one container source
lratwarrior@lratwarrior-0P156:~/bigdata/hive$ sudo docker cp create_table.hql hive-server:/tmp
Successfully copied 2.56kB to hive-server:/tmp
lratwarrior@lratwarrior-0P156:~/bigdata/hive$ sudo docker exec -it hive-server /bin/bash

```

```

root@6b464f28d1cd:/opt# cd ..
root@6b464f28d1cd:/# cd tmp
root@6b464f28d1cd:/tmp# hive -f create_table.hql
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in file:/opt/hive/conf/hive-log4j2.properties Async: true
OK
Time taken: 1.098 seconds
OK
Time taken: 0.015 seconds
OK
Time taken: 0.246 seconds
OK
Time taken: 0.054 seconds
root@6b464f28d1cd:/tmp#

```

Рис. 5, 6 – копирование скрипта hql для создания базы данных в hive с хоста в docker контейнер и создание этой базы данных

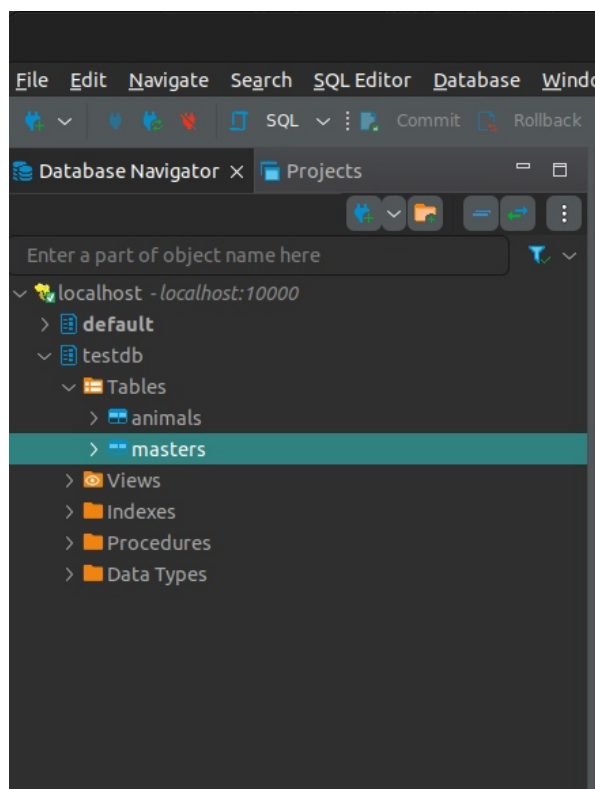


Рис. 7 – созданная база данных с двумя таблицами в DBeaver

```

iratewarrior@iratewarrior-DP156:~/bigdata/hive$ sudo docker cp animals.csv datanode:/tmp
Preparing to copy...
Copying to container - 2.048kB
Successfully copied 2.048kB to datanode:/tmp
iratewarrior@iratewarrior-DP156:~/bigdata/hive$ sudo docker cp masters.csv datanode:/tmp
Preparing to copy...
Copying to container - 2.048kB
Successfully copied 2.048kB to datanode:/tmp
iratewarrior@iratewarrior-DP156:~/bigdata/hive$ sudo docker exec -it datanode /bin/bash
root@060517e9b20d:/# cd /tmp/
root@060517e9b20d:/tmp# ls
Jetty_localhost_38487_datanode_...6vsinf animals.csv hserverdata_root masters.csv
root@060517e9b20d:/tmp# hdfs dfs -copyFromLocal animals.csv /user/hive/warehouse/testdb.db/animals
root@060517e9b20d:/tmp# hdfs dfs -copyFromLocal masters.csv /user/hive/warehouse/testdb.db/masters
root@060517e9b20d:/tmp#

```

Рис. 8 – копирование csv файлов на datanode

animal id	abc name	123 count	abc mood	123 master id
1	cats	25	friendly	2
2	dogs	30	lovely	3
3	hamsters	15	silly	1
4	otters	5	jolly	1

Рис. 9 – таблицы в базе данных заполнены данными из csv файлов

```

hive
iratewarrior@iratewarrior-DP156:~/bigdata/hive$ sudo docker exec -it hive-server /bin/bash
root@06b464f28d1cd:/opt# hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in file:/opt/hive/conf/hive-log4j2.properties Async: true
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.
hive> use testdb
> ;
OK
Time taken: 0.672 seconds

```

```

hive> select a.*, m.* from animals a as inner join masters as m on a.master_id = m.master_id;
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_20230330133839_a2dcfad0-8c67-44b8-82e2-1f222bbf119b
Total jobs = 1
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Execution log at: /tmp/root/root_20230330133839_a2dcfad0-8c67-44b8-82e2-1f222bbf119b.log
2023-03-30 13:38:46 Starting to launch local task to process map join; maximum memory = 477626368
2023-03-30 13:38:46 Dump the side-table for tag: 1 with group count: 3 into file: file:/tmp/root/ac3236f3-c1f9-46db-ad09-b527d95ede6e/hive_2023-03-30_13-38-39_952_1308499187162719818-1/-local-10004/HashTable-Stage-3/MapJoin-mapfile01...hashtable
2023-03-30 13:38:46 Uploaded 1 File to: file:/tmp/root/ac3236f3-c1f9-46db-ad09-b527d95ede6e/hive_2023-03-30_13-38-39_952_1308499187162719818-1/-local-10004/HashTable-Stage-3/MapJoin-mapfile01...hashtable (332 bytes)
2023-03-30 13:38:46 End of local task; Time Taken: 0.855 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Job running in-process (local Hadoop)
2023-03-30 13:38:49,327 Stage-3 map = 100%, reduce = 0%
Ended Job = job_local1023704723_0001
MapReduce Jobs Launched:
Stage-Stage-3: HDFS Read: 81 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 nsec
OK
1 cats 25 friendly 2 2 Ivan NULL
2 dogs 30 lovely 3 3 Adam NULL
3 hamsters 15 silly 1 1 John NULL
4 otters 5 jolly 1 1 John NULL
Time taken: 9.386 seconds, Fetched: 4 row(s)
hive>

```

Рис. 10, 11 – объединение двух таблиц внутри hive-server

```

insert into newtable
select a.master_id, a.animal_id, a.name, a.count, a.mood, m.phone, m.name
from animals as a inner join masters as m on a.master_id = m.master_id;

```

Рис. 12 – создание третьей объединяющей две предыдущие таблицы в DBeaver

	master id	animal id	name	count	mood	phone	name master
1	2	[NULL]	25	[NULL]	1	[NULL]	[NULL]
2	3	[NULL]	30	[NULL]	2	[NULL]	[NULL]
3	1	[NULL]	15	[NULL]	3	[NULL]	[NULL]
4	1	[NULL]	5	[NULL]	4	[NULL]	[NULL]
5	2	1	cats	25	friendly	[NULL]	Ivan
6	3	2	dogs	30	lovely	[NULL]	Adam
7	1	3	hamsters	15	silly	[NULL]	John
8	1	4	otters	5	jolly	[NULL]	John
9	2	1	cats	25	friendly	[NULL]	Ivan
10	3	2	dogs	30	lovely	[NULL]	Adam
11	1	3	hamsters	15	silly	[NULL]	John
12	1	4	otters	5	jolly	[NULL]	John

Рис. 13 – таблицы объединены

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help
/home/iratewarrior/Downloads/imcsv.py

import csv
from pyhive import hive

docker_host = 'localhost'
hive_port = 10000

# Подключение к Hive
conn = hive.Connection(host=docker_host, port=hive_port)

# Запрос к таблице Hive
database_name = 'testdb'
table_name = 'newtable'
query = f'SELECT * FROM {database_name}.{table_name}'

# Выполнение запроса и получение данных
with conn.cursor() as cursor:
    cursor.execute(query)
    data = cursor.fetchall()
    column_names = [desc[0] for desc in cursor.description]

# Сохранение данных в CSV-файл
output_csv = 'output.csv'

with open(output_csv, 'w', newline='') as csvfile:
    csv_writer = csv.writer(csvfile)
    csv_writer.writerow(column_names)
    csv_writer.writerows(data)

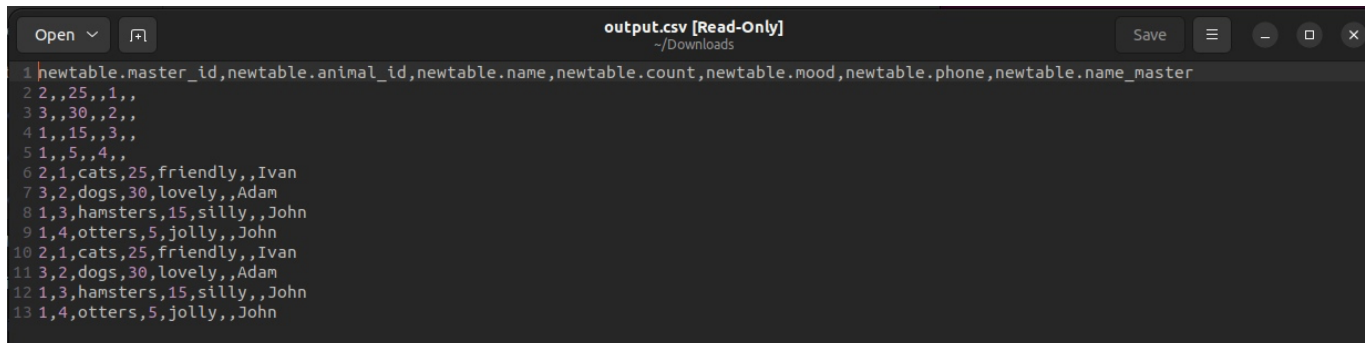
print(f"Данные из таблицы {database_name}.{table_name} успешно сохранены в файл {output_csv}.")

```

Рис. 14 – python скрипт для извлечения данных из newtable в csv файл (используются библиотеки csv и pyhive)


```
iratewarrior@iratewarrior-DP156:~/Downloads$ sudo python3 imcsv.py
Данные из таблицы testdb.newtable успешно сохранены в файл output.csv.
iratewarrior@iratewarrior-DP156:~/Downloads$
```

Рис. 15 – выполнение скрипта успешно



output.csv [Read-Only]
~/Downloads

newtable.master_id	newtable.animal_id	newtable.name	newtable.count	newtable.mood	newtable.phone	newtable.name_master
2	25	1				
3	30	2				
4	15	3				
5	5	4				
6	1	cats	25	friendly		Ivan
7	2	dogs	30	lovely		Adam
8	3	hamsters	15	silly		John
9	4	otters	5	jolly		John
10	1	cats	25	friendly		Ivan
11	2	dogs	30	lovely		Adam
12	3	hamsters	15	silly		John
13	4	otters	5	jolly		John

Рис. 16 – данные из таблицы перенесены в output.csv