Solution – Exercise II

HiveQL, Create and work with External Tables on IMDb Data

Solution

Prerequisites:

- Setup Google Cloud SDK
- Start VM instance
- Pull docker container `marcelmittelstaedt/hive_base:latest`
- Start docker container: `docker run -dit --name hive_base_container -p 8088:8088 -p 9870:9870 -p 9864:9864 marcelmittelstaedt/hive_base:latest`
- Get into docker container
- Start Hadoop and Hive Shell:
  - `start-all.sh`
  - `hive`

Solution

Exercise 1-4:

1. Download and unzip https://datasets.imdbws.com/name.basics.tsv.gz

```
wget https://datasets.imdbws.com/name.basics.tsv.gz
gunzip name.basics.tsv.gz
```

2. Create HDFS directory **/user/hadoop/imdb/name_basics/** for file name.basics.tsv

```
hadoop fs -mkdir /user/hadoop/imdb/name_basics
```

3. Put TSV file to HDFS:

```
hadoop fs -put name.basics.tsv /user/hadoop/imdb/name_basics/name.basics.tsv
```


Solution

Exercise 1-4:

4. Create Hive Table `name_basics`:

```
hive > CREATE EXTERNAL TABLE IF NOT EXISTS name_basics(
    nconst STRING,
    primary_name STRING,
    birth_year INT,
    death_year STRING,
    primary_profession STRING,
    known_for_titles STRING
    ) COMMENT 'IMDb Actors' ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' ST
ORED AS TEXTFILE LOCATION '/user/hadoop/imdb/name_basics'
TBLPROPERTIES ('skip.header.line.count'='1');
```

Solution

Exercise 5:

a) How many movies and how many TV series are within the IMDB dataset?

```
hive >  SELECT m.title_type, count(*)
        FROM title_basics m GROUP BY m.title_type;

tvMovie 129948
movie 568078
tvEpisode 5523814
tvSeries 201466
[…]

Time taken: 35.792 seconds, Fetched: 13 row(s)
```

b) Who is the youngest actor/writer/… within the dataset?

```
hive >  SELECT * FROM name_basics n
        WHERE n.birth_year = ( SELECT MAX(birth_year) FROM name_basics);
```

Solution

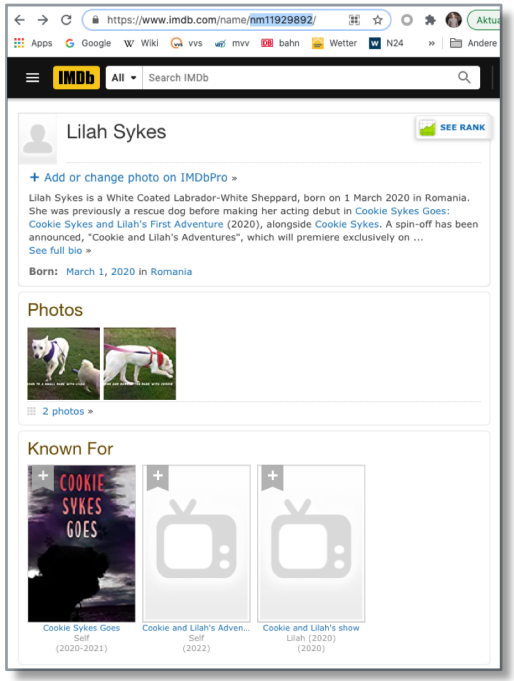
Exercise 5:

b) Who is the youngest actor/writer/… within the dataset?

```
hive >  SELECT * FROM name_basics n
        WHERE n.birth_year = ( SELECT MAX(birth_year) FROM name_basics);
```

```
nm11564929 Wilfred Johnson 2020 NULL NULL
nm11763191 Win Wilson 2020 NULL NULL
nm11929892 Lilah Sykes 2020 NULL tt11271396,tt13273876,tt13288898
nm11946672 Kyle Ivy 2020 NULL producer tt13326546
nm12036892 Rue Shumpert 2020 NULL NULL
nm12122609 Adam James Sanderson 2020 NULL actor tt12668798
nm12133841 Safire Samuels 2020 NULL tt1718437
nm12203950 Amairani Gómez 2020 NULL director,writer NULL
nm12222761 Buddy Danielson 2020 NULL tt5646172
nm12222762 Matteo Chigvintsev 2020 NULL tt5646172
nm12249587 Daisy Bloom 2020 NULL NULL
nm12266412 Isaiah Tota 2020 NULL NULL
Time taken: 70.015 seconds, Fetched: 12 row(s)
```

One of them is actually a dog:



Solution

Exercise 5:

c) Create a list (`m.tconst, m.original_title, m.start_year, r.average_rating, r.num_votes`) of movies which are:
- equal or newer than year 2010
- have an average rating equal or better than 8,1
- have been voted more than 100.000 times

```
hive >    SELECT m.tconst, m.original_title, m.start_year, r.average_rating, r.num_votes
          FROM title_basics m JOIN title_ratings r on (m.tconst = r.tconst)
          WHERE r.average_rating >= 8.1 and m.start_year >= 2010 and m.title_type = 'movie'
          and r.num_votes > 100000
          ORDER BY r.average_rating desc, r.num_votes DESC;
tt1375666 Inception 2010 8.8 2072273
tt5813916 Dag II 2016 8.8 105151
tt0816692 Interstellar 2014 8.6 1517274
tt6751668 Gisaengchung 2019 8.6 559422
tt1675434 Intouchables 2011 8.5 762821
tt2582802 Whiplash 2014 8.5 719944
tt1345836 The Dark Knight Rises 2012 8.4 1519356
tt1853728 Django Unchained 2012 8.4 1361412
tt7286456 Joker 2019 8.4 945473
[…]
```


Solution

Exercise 5:

d) How many movies are in list of c)?

```
hive >    SELECT count(*)
          FROM title_basics m JOIN title_ratings r on (m.tconst = r.tconst)
          WHERE r.average_rating >= 8.1 and m.start_year >= 2010 and m.title_type = 'movie'
          and r.num_votes > 100000;

45
```


Solution

Exercise 5:

e) We want to know which years have been great for cinema.
Create a list with one row per year and a related count of movies which:
- have an average rating better than 8
- have been voted more than 100.000 times
ordered descending by count of movies.

```
hive >  SELECT m.start_year, count(*)
        FROM title_basics m JOIN title_ratings r on (m.tconst = r.tconst)
        WHERE r.average_rating > 8 AND m.title_type = 'movie'
        AND r.num_votes > 100000
        GROUP BY m.start_year
        ORDER BY count(*) DESC;

1995 8
2016 6
2001 6
2000 6
2009 6
2004 6
[…]
```


Solution

Exercise 5:

So 1995 seems to be a really good year for cinema, 8 really good movies have been releases, but which are they?

```
hive > SELECT
            m.tconst, m.original_title, m.start_year, r.average_rating,
            r.num_votes
       FROM title_basics m JOIN title_ratings r ON (m.tconst = r.tconst)
       WHERE
            r.average_rating > 8 AND m.title_type = 'movie'
            AND r.num_votes > 100000 AND m.start_year = 1995
       ORDER BY r.average_rating DESC

tt0114369 Se7en 1995 8.6 1449007
tt0114814 The Usual Suspects 1995 8.5 993283
tt0114709 Toy Story 1995 8.3 889423
tt0112573 Braveheart 1995 8.3 960812
tt0113277 Heat 1995 8.2 578759
tt0112641 Casino 1995 8.2 467532
tt0113247 La haine 1995 8.1 150865
tt0112471 Before Sunrise 1995 8.1 273039

[…]
```