

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 143186  
movie 659107  
tvEpisode 7790928  
tvSeries 250260  
[...]  
  
Time taken: 32.908 seconds, Fetched: 11 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:

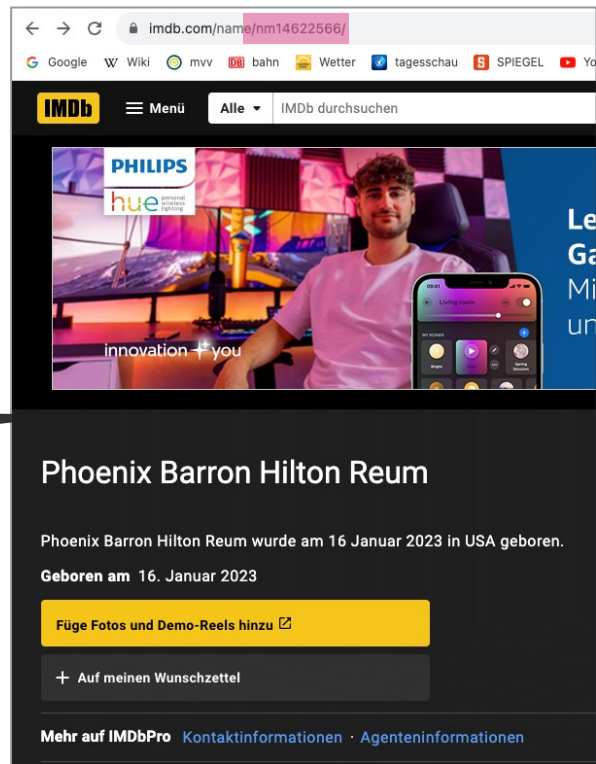
b) *Who is the youngest actor/writer/... within the dataset?*

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

And it's **Phoenix Barron Hilton Reum**, son of Paris Hilton.

```
nm14622566 Phoenix Barron Hilton Reum 2023 NULL tt21841986
nm15209345 Kipekee 2023 NULL tt0044298
nm15263526 Ntombi Hlatwayo 2023 NULL set_decorator tt29144943

Time taken: 65.166 seconds, Fetched: 5 row(s)
```



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 2473810
tt15097216 Jai Bhim 2021 8.8 210868
tt0816692 Interstellar 2014 8.7 1996152
tt9362722 Spider-Man: Across the Spider-Verse 2023 8.7 271837
tt10189514 Soorarai Pottru 2020 8.7 121040
tt10811166 The Kashmir Files 2022 8.6 571706
tt15398776 Oppenheimer 2023 8.6 453992
tt1853728 Django Unchained 2012 8.5 1638685
tt2582802 Whiplash 2014 8.5 935160
tt1675434 Intouchables 2011 8.5 900121
[...]
```

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;
```

59

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  
2019 7  
2009 6  
2014 6  
2001 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 1737999
tt0114814 The Usual Suspects 1995 8.5 1120738
tt0114709 Toy Story 1995 8.3 1039428
tt0113277 Heat 1995 8.3 694703
tt0112573 Braveheart 1995 8.3 1069937
tt0112641 Casino 1995 8.2 546487
tt0113247 La haine 1995 8.1 187871
tt0112471 Before Sunrise 1995 8.1 328387

[...]
```