

Exercises Preparation

Install and Setup Hive



www.marcel-mittelstaedt.com

Create VM Instance

1. Delete previously created VM instance:

```
gcloud compute instances delete big-data
```

2. Create new instance:

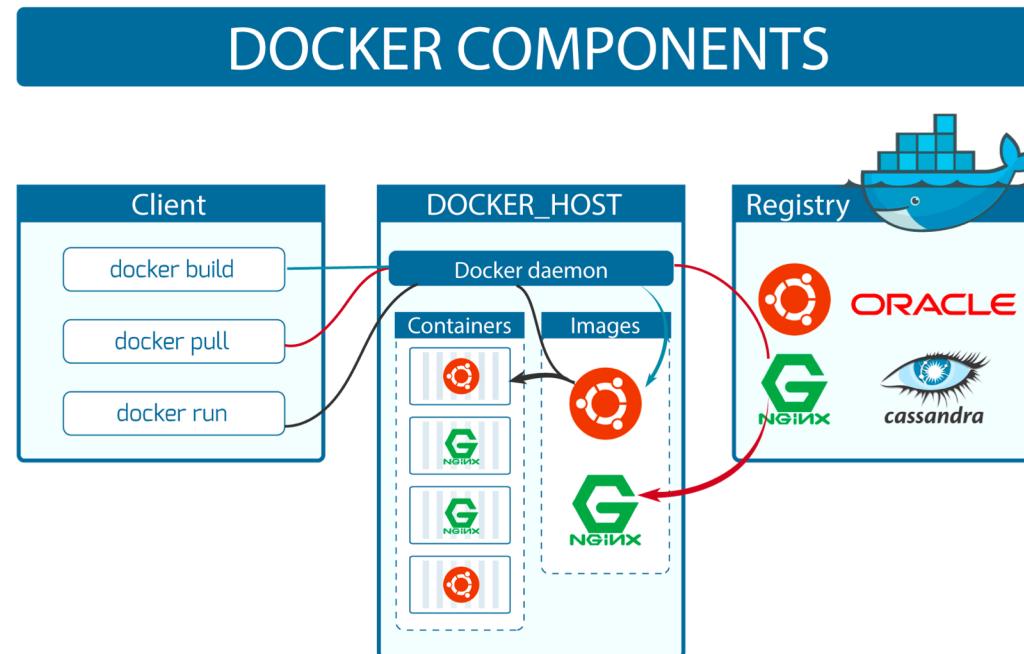
```
gcloud compute --project=[your-project-id] instances create big-data \
    --zone=europe-west3-c \
    --machine-type=n1-standard-4 \
    --subnet=default --network-tier=PREMIUM \
    --maintenance-policy=MIGRATE \
    --image=ubuntu-1804-bionic-v20190918 \
    --image-project=ubuntu-os-cloud \
    --boot-disk-size=30GB \
    --boot-disk-type=pd-standard \
    --boot-disk-device-name=big-data
```

```
ssh hans.wurst@XXX.XXX.XXX.XXX
```



Docker

To **speed things up** and not waste time on installation and configuration of Hive and other tools, we will make use of docker container I've already prepared.



Docker Images/Dockerfiles

A screenshot of a GitHub repository page for 'marcelmittelstaedt / BigData'. The repository has 1 star and 0 forks. The code tab is selected. A file named 'README.md' is shown with a commit message from 'marcelmittelstaedt' updating it two days ago. Below the file list, there is a section titled 'Docker Images' with a description and a link to the Docker Hub URL: <https://hub.docker.com/u/marcelmittelstaedt>. The 'Images:' section lists several Docker images:

- **Hadoop Base Image** Hadoop 3.1.2 Base Image (Ubuntu 18.04)
- **Hadoop and Hive Base Image** Hadoop 3.1.2 and Hive 3.1.2 Base Image (Ubuntu 18.04)
- **Hadoop, Hive and HiveServer2 Base Image** Hadoop 3.1.2, Hive 3.1.2 and HiveServer2 Base Image (Ubuntu 18.04)
- **Spark Base Image** Spark 2.3.4 on Hadoop 3.1.2 as well as Hive 3.1.2 and HiveServer2 Base Image (Ubuntu 18.04)
- **Airflow Base Image** Airflow 1.10.5 with PostgreSQL 10.10 as Metadata Store Base Image (Ubuntu 18.04)

<https://github.com/marcelmittelstaedt/BigData/tree/master/docker>

A screenshot of the Docker Hub interface showing a list of private repositories for user 'marcelmittelstaedt'. There is one repository named 'airflow'. The table includes columns for Repository, Description, and Last Modified.

REPOSITORY	DESCRIPTION	LAST MODIFIED
marcelmittelstaedt / airflow	Airflow 1.10.5 Image using PostgreSQL 10.10 for Metadata (Ubuntu...)	2 days ago
marcelmittelstaedt / spark_base	Hadoop 3.1.2 and Spark 2.3.4 Base Image (Ubuntu 18.04)	5 days ago
marcelmittelstaedt / hiveserver_base	Hive 3.1.2, Hadoop 3.1.2 and HiveServer2 Base Image (Ubuntu 18....)	5 days ago
marcelmittelstaedt / hive_base	Hive 3.1.2 and Hadoop 3.1.2 Base Image (Ubuntu 18.04)	6 days ago
marcelmittelstaedt / hadoop_base	Hadoop 3.1.2 Base Image (Ubuntu 18.04)	6 days ago
marcelmittelstaedt / ubuntu_18_04_base	Ubuntu 18.04 Base Image created from scratch using debootstrap.	8 days ago

<https://hub.docker.com/u/marcelmittelstaedt>



Setup Docker Container

3. Install and setup docker

```
sudo apt-get update  
sudo apt-get install docker.io  
sudo usermod -aG docker $USER  
# exit and login again
```

4. Pull Hadoop with Hive Image

```
docker pull marcelmittelstaedt/hive_base:latest
```

5. Start Container from pulled image:

```
docker run -dit --name hive_base_container -p 8088:8088 -p 9870:9870 -p 9864:  
9864 marcelmittelstaedt/hive_base:latest
```



Setup Docker Container

6. Show Running Container:

```
docker ps -a
```

```
marcel.mittelstaedt@big-data:~$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
c821a0e1bdcf      marcelmittelstaedt/hive_base:latest   "/startup.sh"   6 minutes ago    Up 6 minutes
marcel.mittelstaedt@big-data:~$
```

7. Show Logs of container (wait till finished):

```
docker logs hive_base_container
```

```
[...]
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [c821a0e1bdcf]
Stopping nodemanagers
Stopping resourcemanager
Container Startup finished.
```



Setup Docker Container

8. Get a shell inside the container:

```
hans.wurst@big-data:~$ docker exec -it hive_base_container bash  
root@c821a0e1bdcf:/#
```

9. Switch to hadoop user:

```
root@c821a0e1bdcf:/# sudo su hadoop  
hadoop@c821a0e1bdcf:/$ cd  
hadoop@c821a0e1bdcf:~$
```

10. Start DFS and YARN:

```
start-all.sh
```



Install and Setup Hive

11. Test if Hive Installation and Configuration is successful. Start Hive:

```
hive
```



```
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/home/hadoop/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/  
impl/StaticLoggerBinder.class]  
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7  
.25.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]  
Hive Session ID = c120d0b1-9025-43db-96e4-48ccfb875f1a  
  
Logging initialized using configuration in jar:file:/home/hadoop/hive/lib/hive-common-3.1.0.jar  
!/hive-log4j2.properties Async: true  
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider us  
ing a different execution engine (i.e. spark, tez) or using Hive 1.X releases.  
Hive Session ID = fdd6f06a-d4e4-48e6-8971-997e8a0a8e2c  
hive>
```



Install and Setup Hive

12. Execute First SQL Query:

```
hive> show databases;  
OK  
default  
Time taken: 0.083 seconds, Fetched: 1 row(s)  
hive>
```



Hive: Create and Work with External Tables

Using public dataset of IMDb.com



www.marcel-mittelstaedt.com

Get IMDb Data And Move It To HDFS

1. Get **IMDb Data** (<https://www.imdb.com/interfaces/>):

```
wget https://datasets.imdbws.com/title.basics.tsv.gz
wget https://datasets.imdbws.com/title.ratings.tsv.gz
```

2. Uncompress IMDb Data:

```
gunzip title.basics.tsv.gz
gunzip title.ratings.tsv.gz
```

3. Create HDFS Directories for IMDb Data:

```
hadoop fs -mkdir /user/hadoop/imdb
hadoop fs -mkdir /user/hadoop/imdb/title_basics
hadoop fs -mkdir /user/hadoop/imdb/title_ratings
```



Create External Tables In Hive

4. Transfer IMDb data files to HDFS:

```
hadoop fs -put title.basics.tsv /user/hadoop/imdb/title_basics/title.basics.tsv  
hadoop fs -put title.ratings.tsv /user/hadoop/imdb/title_ratings/title.ratings.tsv
```

5. Create External Table **title_ratings** (file *title.ratings.tsv*) in Hive:

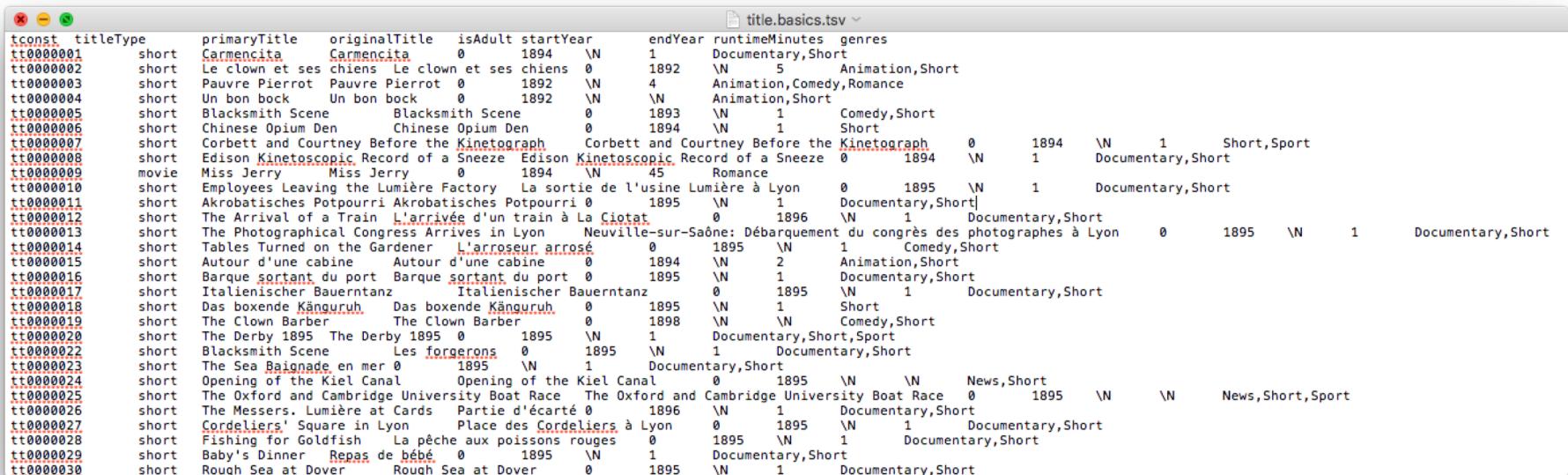
```
hive > CREATE EXTERNAL TABLE IF NOT EXISTS title_ratings(  
      tconst STRING,  
      average_rating DECIMAL(2,1),  
      num_votes BIGINT  
) COMMENT 'IMDb Ratings'  
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS  
TEXTFILE LOCATION '/user/hadoop/imdb/title_ratings'  
TBLPROPERTIES ('skip.header.line.count'=1');
```

tcn	averageRating	numVotes
tt0000001	5.8	1416
tt0000002	6.4	167
tt0000003	6.6	1013
tt0000004	6.4	100
tt0000005	6.2	1712
tt0000006	5.6	87
tt0000007	5.5	571
tt0000008	5.6	1520
tt0000009	5.6	68
tt0000010	6.9	5075
tt0000011	5.4	208
tt0000012	7.4	8479
tt0000013	5.7	1297
tt0000014	7.2	3683
tt0000015	6.2	642



Create External Tables In Hive

6. Create External Table `title_basics` for file `title.basics.tsv` in Hive:



tconst	titleType	primaryTitle	originalTitle	isAdult	startYear	endYear	runtimeMinutes	genres
tt0000001	short	Carmencita	Carmencita	0	1894	\N	1	Documentary,Short
tt0000002	short	Le clown et ses chiens	Le clown et ses chiens	0	1892	\N	5	Animation,Short
tt0000003	short	Pauvre Pierrot	Pauvre Pierrot	0	1892	\N	4	Animation,Comedy,Romance
tt0000004	short	Un bon bock	Un bon bock	0	1892	\N	1	Animation,Short
tt0000005	short	Blacksmith Scene	Blacksmith Scene	0	1893	\N	1	Comedy,Short
tt0000006	short	Chinese Opium Den	Chinese Opium Den	0	1894	\N	1	Short
tt0000007	short	Corbett and Courtney Before the Kinetograph	Corbett and Courtney Before the Kinetograph	0	1894	\N	1	Short,Sport
tt0000008	short	Edison Kinetoscopic Record of a Sneeze	Edison Kinetoscopic Record of a Sneeze	0	1894	\N	1	Documentary,Short
tt0000009	movie	Miss Jerry	Miss Jerry	0	1894	\N	45	Romance
tt0000010	short	Employees Leaving the Lumière Factory	La sortie de l'usine Lumière à Lyon	0	1895	\N	1	Documentary,Short
tt0000011	short	Akrobatisches Potpourri	Akrobatisches Potpourri	0	1895	\N	1	Documentary,Short
tt0000012	short	The Arrival of a Train	L'arrivée d'un train à La Ciotat	0	1896	\N	1	Documentary,Short
tt0000013	short	The Photographical Congress Arrives in Lyon	Neuville-sur-Saône: Débarquement du congrès des photographes à Lyon	0	1895	\N	1	Documentary,Short
tt0000014	short	Tables Turned on the Gardener	L'arroseur arrosé	0	1895	\N	1	Comedy,Short
tt0000015	short	Autour d'une cabine	Autour d'une cabine	0	1894	\N	2	Animation,Short
tt0000016	short	Barque sortant du port	Barque sortant du port	0	1895	\N	1	Documentary,Short
tt0000017	short	Italienischer Bauerntanz	Italienischer Bauerntanz	0	1895	\N	1	Documentary,Short
tt0000018	short	Das boxende Känguru	Das boxende Känguru	0	1895	\N	1	Short
tt0000019	short	The Clown Barber	The Clown Barber	0	1898	\N	1	Comedy,Short
tt0000020	short	The Derby 1895	The Derby 1895	0	1895	\N	1	Documentary,Short,Sport
tt0000022	short	Blacksmith Scene	Les forgerons	0	1895	\N	1	Documentary,Short
tt0000023	short	The Sea Baignade en mer	\N	0	1895	\N	1	Documentary,Short
tt0000024	short	Opening of the Kiel Canal	Opening of the Kiel Canal	0	1895	\N	\N	News,Short
tt0000025	short	The Oxford and Cambridge University Boat Race	The Oxford and Cambridge University Boat Race	0	1895	\N	\N	News,Short,Sport
tt0000026	short	The Messers. Lumière at Cards	Partie d'écarté	0	1896	\N	1	Documentary,Short
tt0000027	short	Cordeliers' Square in Lyon	Place des Cordeliers à Lyon	0	1895	\N	1	Documentary,Short
tt0000028	short	Fishing for Goldfish	La pêche aux poissons rouges	0	1895	\N	1	Documentary,Short
tt0000029	short	Baby's Dinner	Repas de bébé	0	1895	\N	1	Documentary,Short
tt0000030	short	Rough Sea at Dover	Rough Sea at Dover	0	1895	\N	1	Documentary,Short



Create External Tables In Hive

6. Create External Table **`title_basics`** for file *title.basics.tsv* in Hive:

```
hive > CREATE EXTERNAL TABLE IF NOT EXISTS title_basics (
    tconst STRING,
    title_type STRING,
    primary_title STRING,
    original_title STRING,
    is_adult DECIMAL(1,0),
    start_year DECIMAL(4,0),
    end_year STRING,
    runtime_minutes INT,
    genres STRING
) COMMENT 'IMDb Movies' ROW FORMAT DELIMITED FIELDS TERMINATED BY
'\t' STORED AS TEXTFILE LOCATION '/user/hadoop/imdb/title_basics'
TBLPROPERTIES ('skip.header.line.count'='1');
```



Create External Tables In Hive

7. Query Table **title_basics** in Hive using SQL (HiveQL):

```
hive> select * from title_basics limit 3;
OK
tt0000001 short Carmencita Carmencita 0 1894 NULL 1 Documentary,Short
tt0000002 short Le clown et ses chiens Le clown et ses chiens 0 1892 NULL 5 Animation,Short
tt0000003 short Pauvre Pierrot Pauvre Pierrot 0 1892 NULL 4 Animation,Comedy,Romance
Time taken: 0.139 seconds, Fetched: 3 row(s)
hive>
```

8. Query Table **title_ratings** in Hive using SQL (HiveQL):

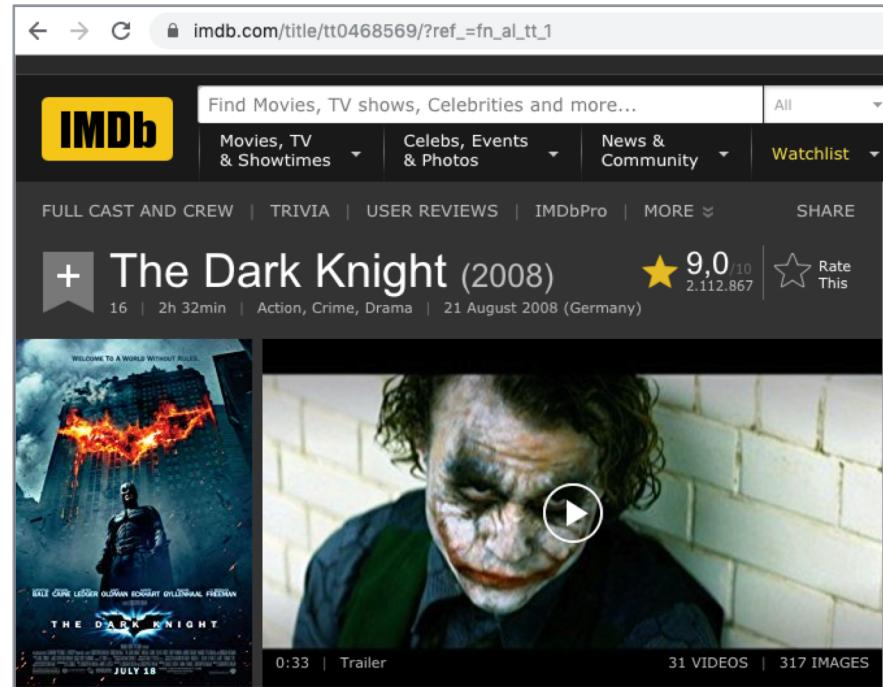
```
hive> select * from title_ratings limit 3;
OK
tt0000001 5.6 1540
tt0000002 6.1 186
tt0000003 6.5 1199
Time taken: 0.119 seconds, Fetched: 3 row(s)
hive>
```



Create External Tables In Hive

9. Run a complex query which starts a MapReduce Job on Yarn, e.g. get Rating of movie „The Dark Knight“:

```
SELECT
  *
FROM
  title_basics b
  JOIN title_ratings r ON (b.tconst=r.tconst)
WHERE
  original_title = 'The Dark Knight'
  AND title_type='movie';
```



Create External Tables In Hive

9. Execute Query

```
hive> SELECT * FROM title_basics b JOIN title_ratings r ON (b.tconst=r.tconst) WHERE original_title =  
'The Dark Knight' and title_type='movie';  
[...]  
Starting Job = job_1570963963548_0001, Tracking URL = http://c821a0e1bdcf:8088/proxy/application_1570963963548_0001/  
Kill Command = /home/hadoop/hadoop/bin/mapred job -kill job_1570963963548_0001  
Hadoop job information for Stage-1: number of mappers: 3; number of reducers: 3  
2019-10-13 11:38:22,730 Stage-1 map = 0%, reduce = 0%  
2019-10-13 11:38:34,307 Stage-1 map = 67%, reduce = 0%, Cumulative CPU 18.13 sec  
2019-10-13 11:38:45,726 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 28.54 sec  
2019-10-13 11:38:48,897 Stage-1 map = 100%, reduce = 33%, Cumulative CPU 36.24 sec  
2019-10-13 11:38:53,112 Stage-1 map = 100%, reduce = 67%, Cumulative CPU 43.51 sec  
2019-10-13 11:38:56,229 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 49.27 sec  
MapReduce Total cumulative CPU time: 49 seconds 270 msec  
Ended Job = job_1570963963548_0001  
MapReduce Jobs Launched:  
Stage-Stage-3: Map: 2 Cumulative CPU: 34.83 sec HDFS Read: 529291906 HDFS Write: 289 SUCCESS  
Total MapReduce CPU Time Spent: 34 seconds 830 msec  
OK  
tt0468569 movie The Dark Knight The Dark Knight 0 2008 NULL 152 Action,Crime,Drama tt0468569 9.0 2111245  
Time taken: 53.094 seconds, Fetched: 1 row(s)  
hive>
```



Create External Tables In Hive

9. Take a look at YARN (<http://XXX.XXX.XXX.XXX:8088/cluster/>):

All Applications

Logged in as: dr.who

Cluster Metrics											
Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Vcores Used	Vcores Total	Vcores Reserved	
1	0	1	0	3	8 GB	8 GB	0 B	3	3	0	

Cluster Nodes Metrics					
Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
1	0	0	0	0	0

Scheduler Metrics																			
Scheduler Type	Scheduling Resource Type			Minimum Allocation			Maximum Allocation			Maximum Cluster Application Priority									
Capacity Scheduler	[memory-mb (unit=Mi), vcores]			<memory:1024, vcores:1>			<memory:8192, vcores:4>			0									
Show 20 entries	Search:																		
ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU Vcores	Allocated Memory MB	Reserved CPU Vcores	Reserved Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
application_1570963963548_0001	hadoop	SELECT * FROM title_bas...title_type='movie' (Stage-1)	MAPREDUCE	default	0	Sun Oct 13 13:38:13 +0200 2019	N/A	RUNNING	UNDEFINED	3	3	8192	0	0	100.0	100.0		ApplicationMaster	0

Showing 1 to 1 of 1 entries

First Previous 1 Next Last



Exercises II

Hive: Create and Work with External Tables



HDFS and HiveQL Exercises - IMDB

1. Execute Tasks of previous HandsOn Slides
2. Download <https://datasets.imdbws.com/name.basics.tsv.gz>
3. Create HDFS Directory `/user/hadoop/imdb/name_basics/` for file name.basics.tsv
4. Create External Hive Table `name_basics` for name.basics.tsv
5. Use HiveQL to answer following questions:
 - a) How many **movies** and how many **TV series** are within the IMDB dataset?
 - b) Who is the **youngest** actor/writer/... within the dataset?
 - c) Create a list (`tconst, original_title, start_year, average_rating, 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
 - d) How many movies are in list of c)?



HDFS and HiveQL Exercises - IMDB

5. Use HiveQL to answer following questions:

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.