

Prerequisites:

- Download, Install and Setup Hadoop, YARN and Hive (previous lectures)
- Download, Install and Setup Apache HiveServer2
- Start HDFS, YARN and HiveServer2
- Download, Install and Configure JDBC Rich-client:
 - e.g. DBeaver,
 - SquirrelSQL,
 - ...
- Execute all preparation and example tasks of previous HandsOn slides of last lecture



Exercise 2:

2.1 Create table imdb_actors_partitioned partitioned by column partition is alive:



Exercise 2:

2.2 Use static partitioning to create and fill partition 'alive'

```
hive > INSERT OVERWRITE TABLE imdb_actors_partitioned
partition(partition_is_alive='alive')

SELECT

a.nconst,
a.primary_name,
a.birth_year,
a.birth_year,
a.death_year,
a.primary_profession,
a.known_for_titles

FROM imdb_actors a WHERE a.death_year IS NULL
```

Exercise 2:

2.3 Use static partitioning to create and fill partition 'dead'

```
hive > INSERT OVERWRITE TABLE imdb_actors_partitioned
partition(partition_is_alive='dead')

SELECT

a.nconst,
a.primary_name,
a.birth_year,
a.birth_year,
a.death_year,
a.primary_profession,
a.primary_profession,
a.known_for_titles

FROM imdb_actors a WHERE a.death_year IS NOT NULL
```

Exercise 2:

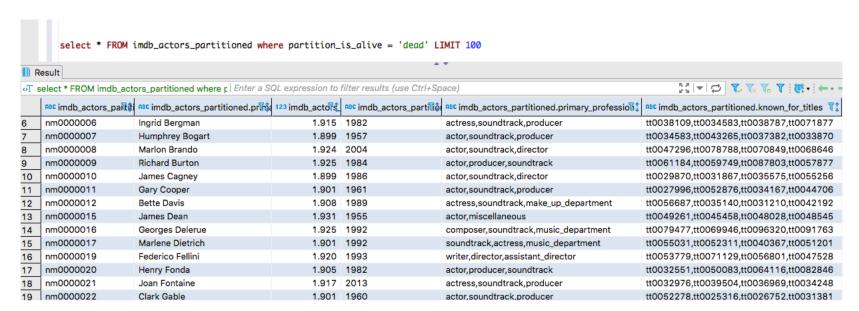
2.4 Check Results:

```
hadoop fs -ls /user/hadoop/imdb/actors partitioned
drwxr-xr-x - hadoop supergroup
                                         0 2018-10-09 20:48 /user/hadoop/imdb/actors partitioned/
partition is alive=alive
drwxr-xr-x - hadoop supergroup
                                         0 2018-10-09 20:54 /user/hadoop/imdb/actors partitioned/
partition is alive=dead
```



Exercise 2:

2.4 Check Results:





Exercise 3:

3.1 Create table imdb_movies_and_ratings_partitioned partitioned by column
partition year using fields of table imdb movies and imdb ratings:

Exercise 3:

3.2 Use dynamic partitioning to create and fill partition partition year:

```
hive >
            SET hive.exec.dynamic.partition.mode=nonstrict;
            INSERT OVERWRITE TABLE imdb movies and ratings partitioned partition (partition year)
            SELECT
                  m.tconst,
                  m.title type,
                  m.primary title,
                  m.original title,
                  m.is adult,
                  m.start year,
                  m.end year,
                  m.runtime minutes,
                  m.genres,
                  r.average rating,
                  r.num votes,
                  m.start year
            FROM imdb movies m JOIN imdb ratings r ON (m.tconst = r.tconst)
```

Exercise 3:

3.3 Check Results:

```
hadoop fs -ls /user/hadoop/imdb/movies and ratings partitioned
drwxr-xr-x
             - hadoop supergroup
                                          0 2018-10-09 21:14 /user/hadoop/imdb/movies and ratings
partitioned/partition year=2012
                                          0 2018-10-09 21:14 /user/hadoop/imdb/movies and ratings
drwxr-xr-x
             - hadoop supergroup
partitioned/partition year=2013
             - hadoop supergroup
                                          0 2018-10-09 21:14 /user/hadoop/imdb/movies and ratings
drwxr-xr-x
partitioned/partition year=2014
                                          0 2018-10-09 21:14 /user/hadoop/imdb/movies and ratings
drwxr-xr-x
             - hadoop supergroup
partitioned/partition year=2015
             - hadoop supergroup
                                          0 2018-10-09 21:14 /user/hadoop/imdb/movies and ratings
drwxr-xr-x
partitioned/partition year=2016
drwxr-xr-x
             - hadoop supergroup
                                          0 2018-10-09 21:14 /user/hadoop/imdb/movies and ratings
partitioned/partition year=2017
             - hadoop supergroup
drwxr-xr-x
                                          0 2018-10-09 21:14 /user/hadoop/imdb/movies and ratings
partitioned/partition year=2018
```



Exercise 3:

3.3 Check Results:

