



# Solution – Exercise 03

HiveQL, HDFS/Hive Partitioning, HiveServer2



# Solution

## 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

# Solution

## Exercise 2:

2.1 Create table `imdb_actors_partitioned` partitioned by column `partition_is_alive`:

```
hive > CREATE EXTERNAL TABLE IF NOT EXISTS imdb_actors_partitioned(  
        nconst STRING,  
        primary_name STRING,  
        birth_year INT,  
        death_year STRING,  
        primary_profession STRING,  
        known_for_titles STRING  
    ) PARTITIONED BY (partition_is_alive STRING)  
    STORED AS ORCFILE LOCATION '/user/hadoop/imdb/actors_partitioned';
```

# Solution

## 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.death_year,
    a.primary_profession,
    a.known_for_titles
FROM imdb_actors a WHERE a.death_year IS NULL
```

# Solution

## 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.death_year,
    a.primary_profession,
    a.known_for_titles
FROM imdb_actors a WHERE a.death_year IS NOT NULL
```

# Solution

## 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
```

# Solution

## Exercise 2:

### 2.4 Check Results:

```
select * FROM imdb_actors_partitioned where partition_is_alive = 'dead' LIMIT 100
```

	ABC imdb_actors_partitioned	ABC imdb_actors_partitioned.primary_profession	123 imdb_actors	ABC imdb_actors_partitioned	ABC imdb_actors_partitioned.primary_profession	ABC imdb_actors_partitioned.known_for_titles
6	nm0000006	Ingrid Bergman	1.915	1982	actress,soundtrack,producer	tt0038109,tt0034583,tt0038787,tt0071877
7	nm0000007	Humphrey Bogart	1.899	1957	actor,soundtrack,producer	tt0034583,tt0043265,tt0037382,tt0033870
8	nm0000008	Marlon Brando	1.924	2004	actor,soundtrack,director	tt0047296,tt0078788,tt0070849,tt0068646
9	nm0000009	Richard Burton	1.925	1984	actor,producer,soundtrack	tt0061184,tt0059749,tt0087803,tt0057877
10	nm0000010	James Cagney	1.899	1986	actor,soundtrack,director	tt0029870,tt0031867,tt0035575,tt0055256
11	nm0000011	Gary Cooper	1.901	1961	actor,soundtrack,producer	tt0027996,tt0052876,tt0034167,tt0044706
12	nm0000012	Bette Davis	1.908	1989	actress,soundtrack,make_up_department	tt0056687,tt0035140,tt0031210,tt0042192
13	nm0000015	James Dean	1.931	1955	actor,miscellaneous	tt0049261,tt0045458,tt0048028,tt0048545
14	nm0000016	Georges Delerue	1.925	1992	composer,soundtrack,music_department	tt0079477,tt0069946,tt0096320,tt0091763
15	nm0000017	Marlene Dietrich	1.901	1992	soundtrack,actress,music_department	tt0055031,tt0052311,tt0040367,tt0051201
16	nm0000019	Federico Fellini	1.920	1993	writer,director,assistant_director	tt0053779,tt0071129,tt0056801,tt0047528
17	nm0000020	Henry Fonda	1.905	1982	actor,producer,soundtrack	tt0032551,tt0050083,tt0064116,tt0082846
18	nm0000021	Joan Fontaine	1.917	2013	actress,soundtrack,producer	tt0032976,tt0039504,tt0036969,tt0034248
19	nm0000022	Clark Gable	1.901	1960	actor,soundtrack,producer	tt0052278,tt0025316,tt0026752,tt0031381

# Solution

## 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`:

```
hive > CREATE TABLE IF NOT EXISTS imdb_movies_and_ratings_partitioned(  
        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,  
        average_rating DECIMAL(2,1),  
        num_votes BIGINT  
    ) PARTITIONED BY (partition_year int) STORED AS ORCFILE LOCATION '/user/hadoop/imdb/  
movies_and_ratings_partitioned';
```



# Solution

## 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)
```

# Solution

## 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  
drwxr-xr-x    - hadoop supergroup          0 2018-10-09 21:14 /user/hadoop/imdb/movies_and_ratings_  
partitioned/partition_year=2013  
drwxr-xr-x    - hadoop supergroup          0 2018-10-09 21:14 /user/hadoop/imdb/movies_and_ratings_  
partitioned/partition_year=2014  
drwxr-xr-x    - hadoop supergroup          0 2018-10-09 21:14 /user/hadoop/imdb/movies_and_ratings_  
partitioned/partition_year=2015  
drwxr-xr-x    - hadoop supergroup          0 2018-10-09 21:14 /user/hadoop/imdb/movies_and_ratings_  
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  
drwxr-xr-x    - hadoop supergroup          0 2018-10-09 21:14 /user/hadoop/imdb/movies_and_ratings_  
partitioned/partition_year=2018
```

# Solution

## Exercise 3:

### 3.3 Check Results:

```
select * FROM imdb_movies_and_ratings_partitioned mrp where mrp.partition_year = '2018' LIMIT 100
```

Result

select \* FROM imdb\_movies\_and\_ratings\_partitioned | Enter a SQL expression to filter results (use Ctrl+Space)

ABC mrp.tit	ABC mrp.primary_title	ABC mrp.original_title	123 mrp.is	123 mrp.star	ABC mrp	123 mrp	ABC mrp.genres	123 mrp.a	123 mrp.num	123 mrp.partition_year
movie	The Other Side of the Wind	The Other Side of the Wind	0	2.018	[NULL]	122	Comedy,Drama	7,5	174	2.018
movie	In Memoriam Alfons Vrancx	Le Saigneur est avec nous	0	2.018	[NULL]	90	Action,Adventure,Comedy	6,4	13	2.018
tvMovie	The Patchwork Girl of Oz	The Patchwork Girl of Oz	0	2.018	[NULL]	[NULL]	Adventure,Animation,Famil	5,1	14	2.018
movie	Nappily Ever After	Nappily Ever After	0	2.018	[NULL]	98	Comedy,Drama,Romance	6,5	2.354	2.018
movie	A Million Little Pieces	A Million Little Pieces	0	2.018	[NULL]	113	Drama	8,2	79	2.018
movie	Duel of Legends	Duel of Legends	0	2.018	[NULL]	[NULL]	Action,Drama,Sport	5,9	34	2.018
movie	Dukun	Dukun	0	2.018	[NULL]	108	Horror,Thriller	6,9	238	2.018
movie	Suspiria	Suspiria	0	2.018	[NULL]	152	Fantasy,Horror,Mystery	6,5	673	2.018
movie	Farming	Farming	0	2.018	[NULL]	107	Drama	7,5	40	2.018
movie	Death Wish	Death Wish	0	2.018	[NULL]	107	Action,Crime,Drama	6,4	37.212	2.018
movie	Beautiful Boy	Beautiful Boy	0	2.018	[NULL]	112	Biography,Drama	7,2	412	2.018
movie	Where Hands Touch	Where Hands Touch	0	2.018	[NULL]	122	Drama,Romance,War	5,1	101	2.018
movie	Den of Thieves	Den of Thieves	0	2.018	[NULL]	140	Action,Crime,Drama	7,0	52.408	2.018
movie	London Fields	London Fields	0	2.018	[NULL]	118	Crime,Mystery,Thriller	5,4	101	2.018
movie	The Guernsey Literary and Pot	The Guernsey Literary and Pota	0	2.018	[NULL]	124	Drama,History,Romance	7,4	12.586	2.018

