**IMDB dataset**

1. **Step: Convert the TSV files to CVS file in jupyter (It is needed for import the data later to MYSQL)**

**Code:**

import pandas as pd

tsv\_file='name.basics.tsv'

csv\_table=pd.read\_table(tsv\_file,sep='\t')

csv\_table.to\_csv('name.basics.csv',index=False)

tsv\_file='title.akas.tsv'

csv\_table=pd.read\_table(tsv\_file,sep='\t')

csv\_table.to\_csv('title.akas.csv',index=False)

tsv\_file='title.basics.tsv'

csv\_table=pd.read\_table(tsv\_file,sep='\t')

csv\_table.to\_csv('title.basics.csv',index=False)

tsv\_file='title.crew.tsv'

csv\_table=pd.read\_table(tsv\_file,sep='\t')

csv\_table.to\_csv('title.crew.csv',index=False)

tsv\_file='title.episode.tsv'

csv\_table=pd.read\_table(tsv\_file,sep='\t')

csv\_table.to\_csv('title.episode.csv',index=False)

tsv\_file='title.principals.tsv'

csv\_table=pd.read\_table(tsv\_file,sep='\t')

csv\_table.to\_csv('title.principals.csv',index=False)

tsv\_file='title.ratings.tsv'

csv\_table=pd.read\_table(tsv\_file,sep='\t')

csv\_table.to\_csv('title.ratings.csv',index=False)

1. **Step: Create the database for the data in MYSQL**

**Code:**

DROP DATABASE IF EXISTS imdb;

CREATE DATABASE imdb;

USE imdb;

DROP TABLE IF EXISTS name\_basics;

CREATE TABLE name\_basics

(

nconst varchar(255) primary key not null,

primaryName varchar(255),

birthYear int,

deathYear int,

primaryProfession varchar(255),

knownForTitles varchar(255)

);

USE imdb;

DROP TABLE IF EXISTS title\_akas;

CREATE TABLE title\_akas

(

titleId varchar(255),

ordering varchar(255),

title varchar(1000),

region varchar(255),

lang varchar(255),

typ varchar(255),

attributes varchar(255),

isOriginalTitle int

);

USE imdb;

DROP TABLE IF EXISTS title\_basics;

CREATE TABLE title\_basics

(

tconst varchar(255) primary key not null,

titleType varchar(255),

primaryTitle varchar(1000),

originalTitle varchar(1000),

isAdult int,

startYear int,

endYear int,

runtimeMinutes varchar(255),

genres varchar(255)

);

USE imdb;

DROP TABLE IF EXISTS title\_crew;

CREATE TABLE title\_crew

(

tconst varchar(255) primary key not null,

directors varchar(7000),

writers varchar(8000)

);

USE imdb;

DROP TABLE IF EXISTS title\_episode;

CREATE TABLE title\_episode

(

tconst varchar(255) primary key not null,

parentTconst varchar(255),

seasonNumber int,

episodeNumber int

);

USE imdb;

DROP TABLE IF EXISTS title\_principals;

CREATE TABLE title\_principals

(

tconst varchar(255),

ordering int,

nconst varchar(255),

category varchar(255),

job varchar(255),

characters varchar(255)

);

USE imdb;

DROP TABLE IF EXISTS title\_ratings;

CREATE TABLE title\_ratings

(

tconst varchar(255) primary key not null,

averageRating float,

numVotes int

);

1. **Step: Upload the data.**

**Code:**

SET global local\_infile = 'ON';

LOAD DATA INFILE 'C:/Projects/imdb\_dataset/TSV\_CSV/name.basics.csv' INTO TABLE name\_basics

CHARACTER SET utf8

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\r\n'

IGNORE 1 LINES;

SET global local\_infile = 'ON';

LOAD DATA INFILE 'C:/Projects/imdb\_dataset/TSV\_CSV/title.akas.csv' INTO TABLE title\_akas

CHARACTER SET utf8

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\r\n'

IGNORE 1 LINES;

SET global local\_infile = 'ON';

LOAD DATA INFILE 'C:/Projects/imdb\_dataset/TSV\_CSV/title.basics.csv' INTO TABLE title\_basics

CHARACTER SET utf8

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\r\n'

IGNORE 1 LINES;

SET global local\_infile = 'ON';

LOAD DATA INFILE 'C:/Projects/imdb\_dataset/TSV\_CSV/title.crew.csv' INTO TABLE title\_crew

CHARACTER SET utf8

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\r\n'

IGNORE 1 LINES;

SET global local\_infile = 'ON';

LOAD DATA INFILE 'C:/Projects/imdb\_dataset/TSV\_CSV/title.episode.csv' INTO TABLE title\_episode

CHARACTER SET utf8

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\r\n'

IGNORE 1 LINES;

SET global local\_infile = 'ON';

LOAD DATA INFILE 'C:/Projects/imdb\_dataset/TSV\_CSV/title.principals.csv' INTO TABLE title\_principals

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\r\n'

IGNORE 1 LINES;

SET global local\_infile = 'ON';

LOAD DATA INFILE 'C:/Projects/imdb\_dataset/TSV\_CSV/title.ratings.csv' INTO TABLE title\_ratings

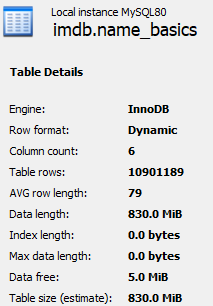
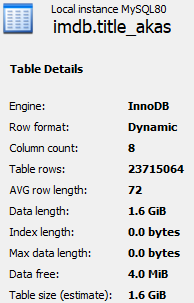
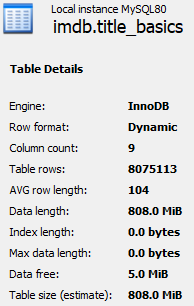
FIELDS TERMINATED BY ','

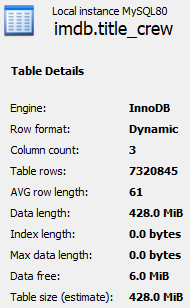
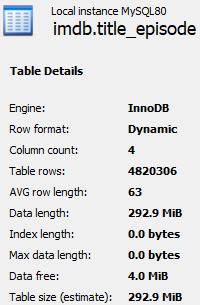
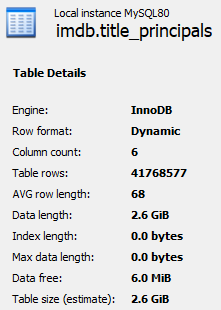
ENCLOSED BY '"'

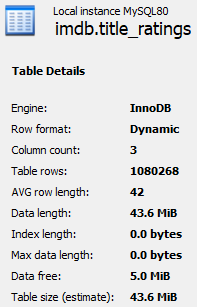
LINES TERMINATED BY '\r\n'

IGNORE 1 LINES;

**Shape of the tables:**



**Who directed an arbitrary movie? For instance: The Dark Knight.**

**Code:**

use imdb;

drop procedure if exists director\_searcher;

Delimiter $$

create procedure director\_searcher(in movie\_name varchar(255), out director varchar(255))

begin

SELECT

nb.primaryName

into director FROM

name\_basics AS nb

JOIN

title\_crew AS tc ON tc.directors = nb.nconst

JOIN

title\_basics AS tb ON tb.tconst = tc.tconst

WHERE

tb.primaryTitle = movie\_name and tb.titleType = 'movie';

end$$

Delimiter ;

set @director = '0';

call imdb.director\_searcher('The Dark Knight', @director);

select @director;



**What was the top 10 movie based on the most votes and ratings?**

**Code:**

create index idx\_titleid on title\_akas(titleID);

create index idx\_tconst on title\_ratings(tconst);

create index idx\_tisOriginalTitle on title\_akas(isOriginalTitle);

SELECT

ta.title, tr.numVotes, tr.averageRating

FROM

title\_akas AS ta

JOIN

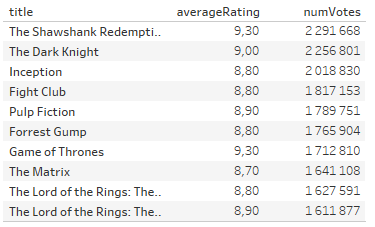
title\_ratings AS tr ON tr.tconst = ta.titleId

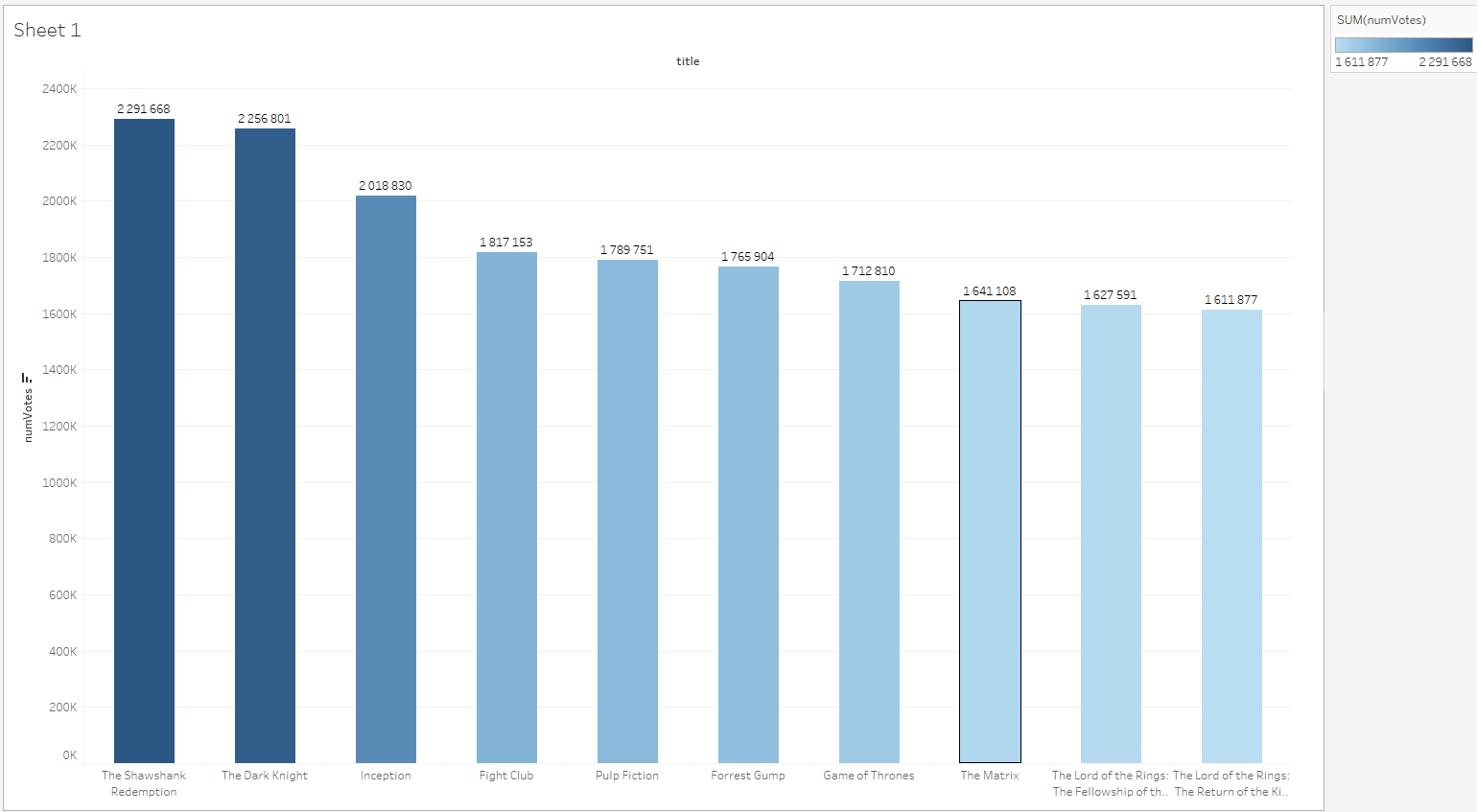
WHERE

ta.isOriginalTitle = '1'

ORDER BY tr.numVotes DESC , tr.averageRating DESC

LIMIT 10;





**What was the ratio between the actor and actress, who were born between 1950 and 1999?**

**Code:**

CREATE INDEX idx\_primaryProfession on name\_basics(primaryProfession);

CREATE INDEX idx\_birthYear on name\_basics(birthYear);

SELECT

primaryName,

birthYear,

CASE

WHEN

(primaryProfession LIKE '%actor%'

AND birthYear BETWEEN '1950-01-01' AND '1999-12-31')

THEN

'actor'

WHEN

(primaryProfession LIKE '%actress%'

AND birthYear BETWEEN '1950-01-01' AND '1999-12-31')

THEN

'actress'

ELSE 'other'

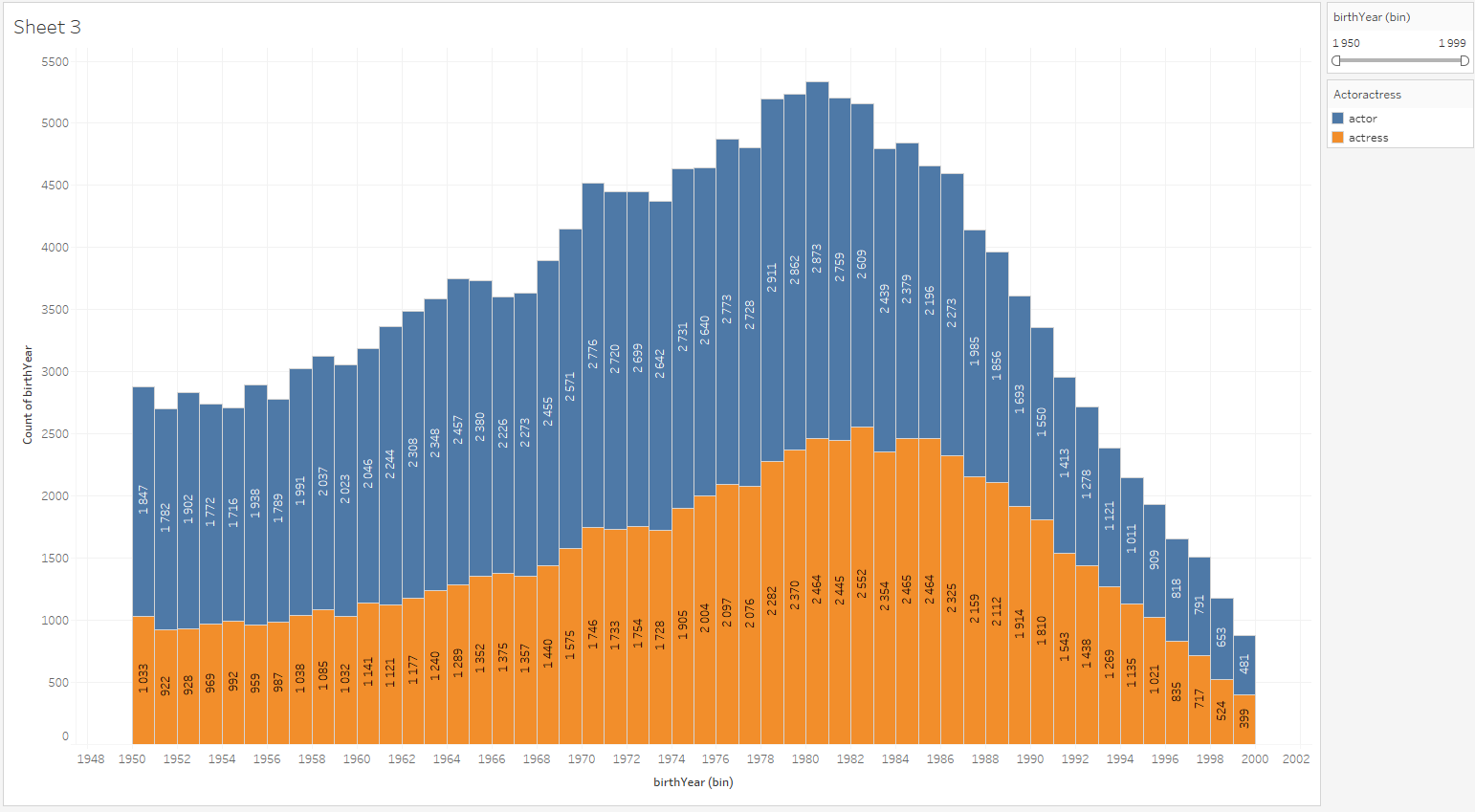
END AS Actoractress

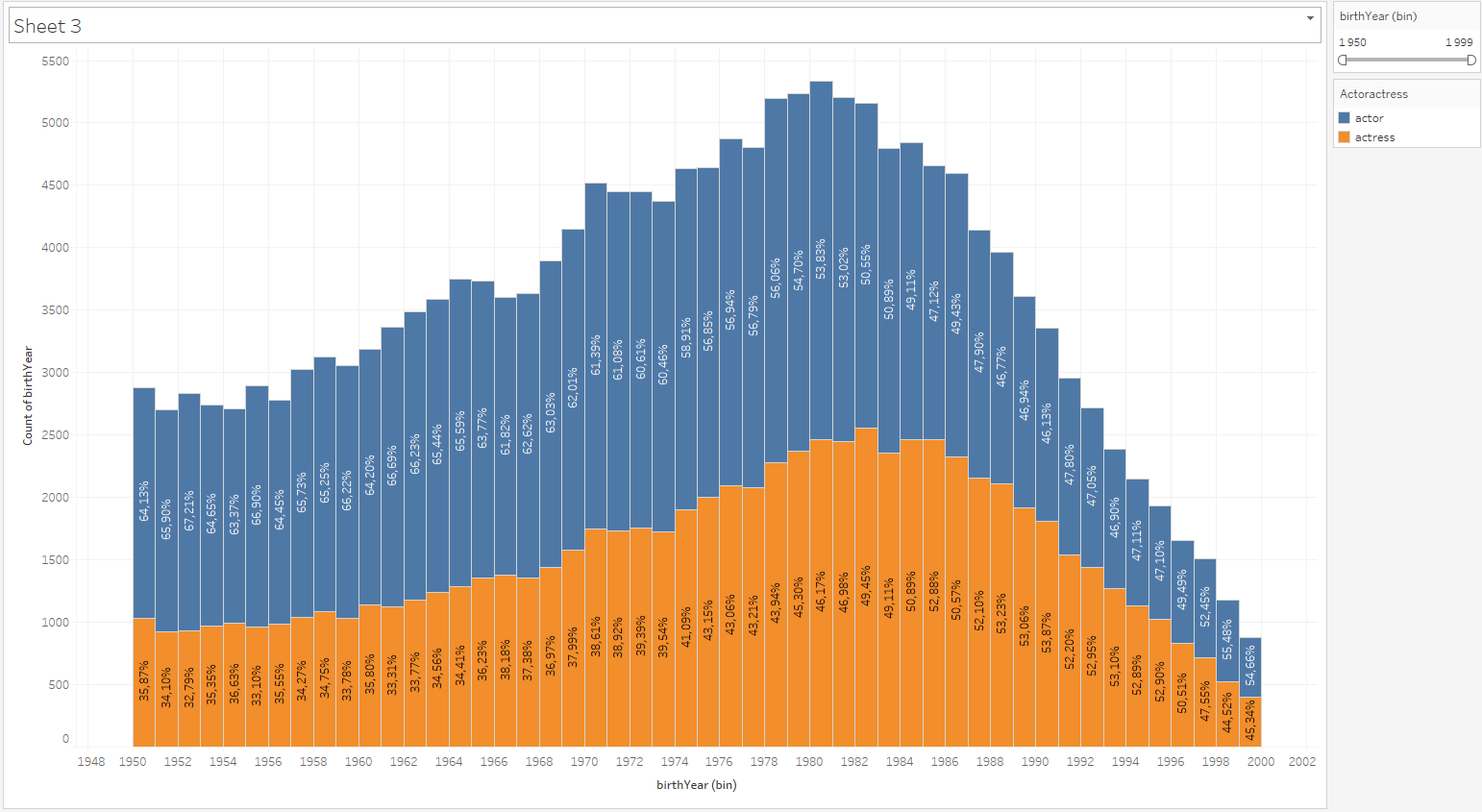
FROM

name\_basics

WHERE

birthYear BETWEEN '1950-01-01' AND '1999-12-31';





**How many movies were appeared after 2000? Where the duritaion of movie was longer than 30 mins. and it was not adult movie!**

**Code:**

CREATE INDEX idx\_isAdult on title\_basics(isAdult);

CREATE INDEX idx\_runtimeMinutes on title\_basics(runtimeMinutes);

CREATE INDEX idx\_startYear on title\_basics(startYear);

CREATE INDEX idx\_titleType on title\_basics(titleType);

SELECT

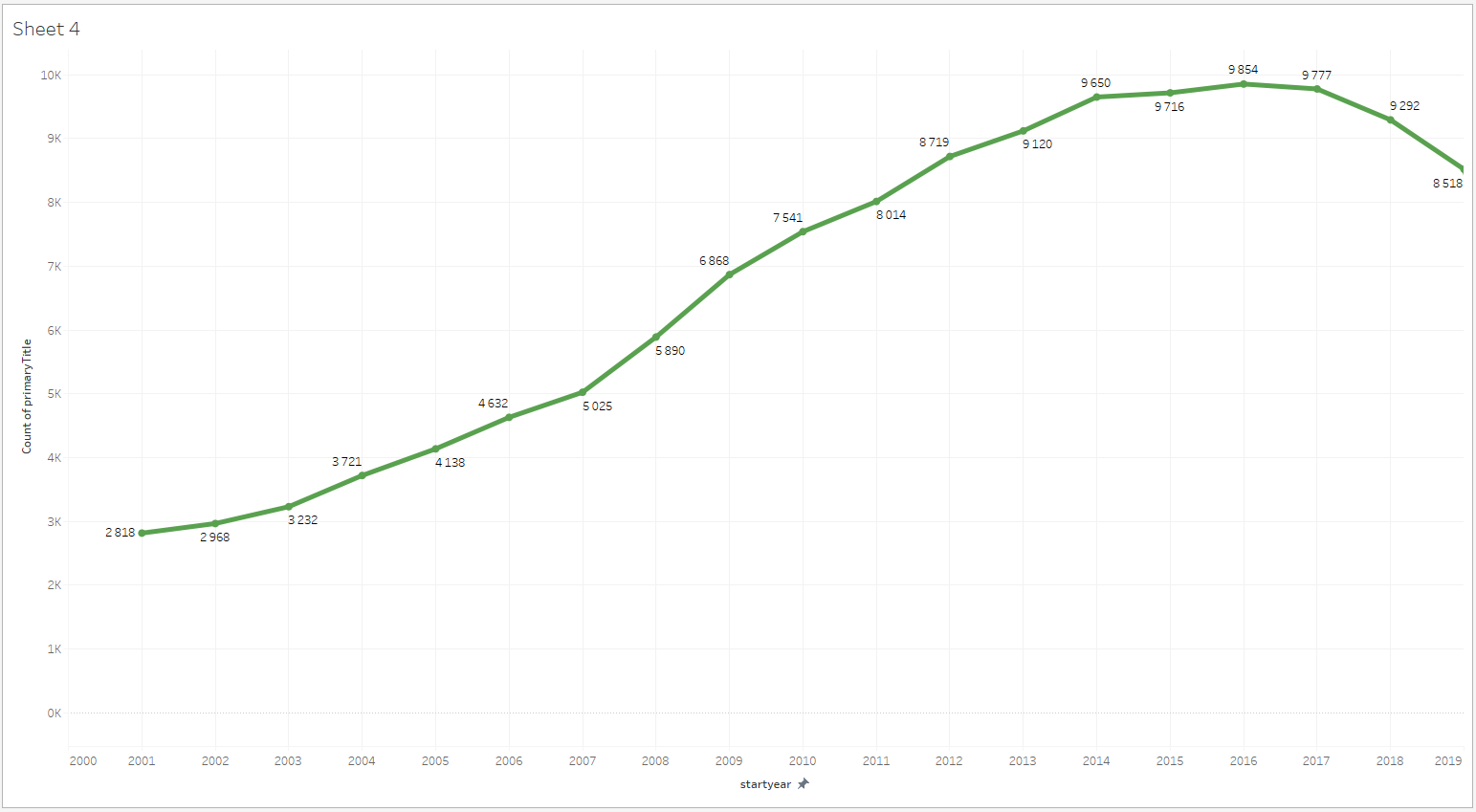
primaryTitle, startyear

FROM

title\_basics

WHERE

isAdult = '0' AND runtimeMinutes > '30' AND startYear > '2000-01-01' AND titleType = 'movie';

****

**Who were the top 10 actor or actress, who acted the most in films?**

**Code:**

create index idx\_nnconst on name\_basics(nconst);

create index idx\_tnconst on title\_principals(nconst);

create index idx\_ttconst on title\_principals(tconst);

create index idx\_ptconst on title\_basics(tconst);

SELECT

nb.primaryName, COUNT(nb.primaryName) AS number\_of\_movies

FROM

name\_basics AS nb

JOIN

title\_principals AS tp ON tp.nconst = nb.nconst

JOIN

title\_basics AS tb ON tb.tconst = tp.tconst

WHERE

(tp.category = 'actor'

OR tp.category = 'actress')

AND tb.titleType = 'movie'

GROUP BY nb.primaryName

ORDER BY COUNT(nb.primaryName) DESC

limit 10;

