Database Systems
IMDb Homework

Instructor: Sharma Chakravarthy
Description of the IMDb Database and Questions

Made available on:	**11/21/2018**
Homework Due on:	**12/02/2018 (11:55 PM)**
Submit by:	**Blackboard (1 zipped folder containing a file with English questions, SQL queries, an answers obtained) https://elearn.uta.edu/**
Weight:	**5% of total**
Total Points:	**50**
Bonus Due on:	**12/02/2018 (11:55 PM) Provides a chance to makeup**
Submit by:	**Blackboard (1 zipped folder containing a file with English questions, SQL queries, an answers obtained) https://elearn.uta.edu/**
Weight:	**5% of total**

We have created a large database and populated it with Millions of rows of International Movies and TV episodes information. It is known as the IMDb database by the community (publicly available data set, but not as a relational DBMS) and used by researchers in databases and other fields. The details of the tables are given below. This has all movie and TV episode information from the beginning (1925) until 2018 for US and international movies and TV episodes. You can query this database for looking up certain information of interest, finding aggregate and statistical information that you are interested in, and OLAP analysis queries as well to the extent possible using SQL.

The IMDb database includes the following information: movie title, year produced, genres a movie belongs to, actors, writers, directors, runtime, adult or non-adult classification, reviews in terms of votes on the movie, average rating, region, language etc. Similarly for TV series.

The purpose of setting up this database and this homework is to provide you with an understanding of the differences between a toy DBMS and large real-world DBMS, in terms of the kinds of queries you can ask, the response time, and appreciate the technology behind a DBMS (query optimization, concurrency control, simple relational abstraction, easy-to-use, non-procedural query language etc.)

Please make sure you do not write queries that produces large amounts of output. You need to think in terms of aggregate queries so you can extract the sliver of information that you are interested in. Also, as many fields contain strings with some delimiter, you need to include the LIKE operator with % and _ for picking out the correct string of interest (can also use string matching). For example, genres can be matched using LIKE 'Comedy' or LIKE

‘Drama’. Note the first letter is capitalized. The other genres present are Horror, Short, Thriller, Sci-Fi, Music, Musical, to name a few. For years, use LIKE ‘200%’ to get values in the range 2000 to 2009. Similarly for others. Some populated field values have a \N as their value. So it is useful to have NOT LIKE ‘\N’ to exclude those.

I. The following tables are populated in the database:

Total Number of Tables: 9
Maximum Number of rows in a table: 27 million rows
Maximum Number of attributes in a table: 9; they are self-explanatory.

1. TITLE_BASICS table

```
SQL> describe TITLE_BASICS
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------

 TCONST                                   NOT NULL VARCHAR2(10)
 TITLETYPE                                NVARCHAR2(500)
 PRIMARYTITLE                             NVARCHAR2(950)
 ORIGINALTITLE                            NVARCHAR2(950)
 ISADULT                                  NUMBER(1)
 STARTYEAR                                NUMBER(4)
 ENDYEAR                                  NUMBER(4)
 RUNTIMEMINUTES                           NUMBER(10)
 GENRES                                   NVARCHAR2(350)

 SQL> select count(*) from TITLE_BASICS;

  COUNT(*)
----------
   Total number of row: 4809386 ( 4.8 million rows)
```
****

2. TITLE_CREW_WRITER table

```
SQL> describe TITLE_CREW_WRITER
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------

 TCONST                            NOT NULL     VARCHAR2(10)
 WRITERS                           NOT NULL     VARCHAR2(10)

 SQL> select count(*) from TITLE_CREW_WRITER;
```

```
  COUNT(*)
----------
  Total number of rows: 5297540 ( 5.2 million rows)
```

3. TITLE_CREW_DIR table

```
SQL> describe TITLE_CREW_DIR
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------

 TCONST                                   NOT NULL  VARCHAR2(10)
 DIRECTORS                                NOT NULL  VARCHAR2(10)

 SQL> select count(*) from TITLE_CREW_DIR;

  COUNT(*)
----------
  Total number of rows: 3408484 (3.4 million rows)
*************************************************************
```

4. TITLE_EPISODE table

```
SQL> describe TITLE_EPISODE
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------

 TCONST                                   NOT NULL  VARCHAR2(10)
 PARENTTCONST                             NOT NULL  VARCHAR2(10)
 SEASONNUMBER                                       NUMBER(9)
 EPISODENUMBER                                      NUMBER(9)

 SQL> select count(*) from TITLE_EPISODE;

  COUNT(*)
----------
   Total number of rows:3206322 (3.2 million rows)
**************************************************************
```

5. TITLE_PRINCIPALS table

```
SQL> describe TITLE_PRINCIPALS
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 TCONST                                    NOT NULL    VARCHAR2(10)
 ORDERING                                              NUMBER(4)
 NCONST                                    NOT NULL    VARCHAR2(10)
 CATEGORY                                              VARCHAR2(550)
 JOB                                                   VARCHAR2(500)
 CHARACTERS                                            NVARCHAR2(800)

 SQL> select count(*) from TITLE_PRINCIPALS;

  COUNT(*)
----------
 Total number of row: 27054380 ( 27 million rows)
*************************************************************
```

6. TITLE_RATINGS tables

```
SQL> describe TITLE_RATINGS
 Name                                      Null?       Type
 ----------------------------------------- -------- ----------------------------

 TCONST                                    NOT NULL    VARCHAR2(10)
 AVERAGERATING                             NOT NULL    NUMBER(5,2)
 NUMVOTES                                  NOT NULL    NUMBER(15)

 SQL> select count(*) from TITLE_RATINGS;

  COUNT(*)
----------
 Total number of rows:  805011 ( 0.8 million rows)
***************************************************************
```

7. TITLE_AKAS table

```
SQL> describe TITLE_AKAS
 Name                                      Null?            Type
 ----------------------------------------- -------- ----------------------------

 TITLEID                                   NOT NULL         VARCHAR2(10)
 ORDERING                                                   NUMBER(10)
 TITLE                                                      NVARCHAR2(950)
 REGION                                                     NVARCHAR2(550)
 LANGUAGE                                                   NVARCHAR2(550)
 TYPES                                                      NVARCHAR2(550)
 ATTRIBUTES                                                 NVARCHAR2(500)
 ISORIGINALTITLE                                            NUMBER(2)
 SQL> select count(*) from TITLE_AKAS;

  COUNT(*)
----------
   3563547 (3.5 million rows)
*************************************************************
```

8. NAME_TITLE_MAPPING table

```
SQL> describe NAME_TITLE_MAPPING
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------

 NCONST                                    NOT NULL VARCHAR2(10)
 TCONST                                    NOT NULL VARCHAR2(10)

SQL> select count(*) from NAME_TITLE_MAPPING;

  COUNT(*)
----------
  14144524 (14 million rows)
*************************************************************
```

9. NAME_BASICS table

```
SQL> describe NAME_BASICS
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------

 NCONST                                    NOT NULL VARCHAR2(10)
 PRIMARYNAME                               NOT NULL NVARCHAR2(950)
 BIRTHYEAR                                          NUMBER(4)
 DEATHYEAR                                          NUMBER(4)
 PRIMARYPROFESSION                                  VARCHAR2(900)

SQL> select count(*) from NAME_BASICS;

  COUNT(*)
----------
   8424762 (8.4 million rows)
```

II. In this homework, you will answer one of the two sets of English queries given below for you primary homework. You can use the other one for bonus part. Choice for each is yours. Answers to the queries have also been provided so you can see whether your queries are correct!

1. a) Retrieve by the years (for the period 2000 to 2009), the count of movies produced in a genre (choose one from Comedy, Drama, Horror, Sci-Fi) whose rating is greater than the average rating of movies in that genre for that year.
*Hint: This query may need the use of **with** clause that we did not cover in the course. It is very similar to the subqueries in the FROM clause.*

Start Year	Genres	Above_avg
2000	Comedy	134
2001	Comedy	136
2002	Comedy	123
2003	Comedy	153
2004	Comedy	176
2005	Comedy	220
2006	Comedy	224
2007	Comedy	207
2008	Comedy	257
2009	Comedy	267

b) For the above years, retrieve the total number of movies produced in each genre

Start Year	Genres	Movies_produced
2000	Comedy	325
2001	Comedy	312
2002	Comedy	314
2003	Comedy	398
2004	Comedy	443
2005	Comedy	539
2006	Comedy	567
2007	Comedy	562
2008	Comedy	689
2009	Comedy	751

2. a) Retrieve the average ratings of the movies for each year during 2010 and 2015 for the genres Comedy, Drama, Horror, and Sci-Fi. 6 output rows, one for each year.

```
start
year      YEARLY_AVG
---------- ----------
2010      6.28914355
2011      6.341491
2012      6.35382862
2013      6.34977264
2014      6.37963079
2015      6.38459016
```

b) Retrieve the average ratings of the movies by genre for each year during 2010 and 2019 for each genres, Comedy, Drama, Horror, and Sci-Fi. Should have 24 rows of output, for 6 years and 4 genres.

```
start
year          GENRES        YEARLY_AVG
----------    ----------    --------------
2010          Drama         6.35474326
2010          Horror        4.93798077
2010          Sci-Fi        5.27391304
2010          Thriller      5.6872

2011          Drama         6.40383653
2011          Horror        4.95151515
2011          Sci-Fi        5.17
2011          Thriller      5.78739496

2012          Drama         6.47727987
2012          Horror        4.98027211
2012          Sci-Fi        4.99583333
2012          Thriller      5.48928571

2013          Drama         6.46666667
2013          Horror        4.8939759
2013          Sci-Fi        5.46666667
2013          Thriller      5.61341463

2014          Drama         6.54582185
2014          Horror        4.94344828
2014          Sci-Fi        5.28
```

2014	Thriller	5.6625
2015	Drama	6.4867052
2015	Horror	4.79851301
2015	Sci-Fi	5.7137931
2015	Thriller	5.9154321

III. Grading Scheme

1.	Completely correct (query and output)	50 (25 each)
2.	Output not correct, but the query runs; partial points Depending upon how close the query is (subjective)	partial (not more than 15 per query)
3.	Query is totally incorrect or does not run (Provide an explanation)	0 to 5 points per query

We may ask you to self-grade this HW and send us your grade using the above rubric after you finish submit the HW on the bb. The final grade will be decided by us.