Part 2: Queries on the database (100 points)

1. Print names of the cast of the movie *"The Davinci Code"* in ascending alpha order.

Solution:

SELECT A.name AS Actor FROM actDirect A, movie M, roles R WHERE A.actDrID = R.actorID AND R.movieId = M.mid AND M.name = 'The Da Vinci Code' ORDER BY A.name ASC;

OutPut:

```
ACTOR
------------------------------------------------------------
Jessica Alba
Scarlett Johanson
Tom Hanks
```

2. Print all information (mid, title, year, num ratings, rating) for the movie(s) with the highest rating (include all that tie for first place). Order by ascending mid.

Solution:

SELECT M.mid, M.name AS TITLE, M.year, AVG(R.rating), COUNT(R.RATING) AS NUM_RATING FROM review R, movie M WHERE R.movieId = M.mid GROUP BY M.mid, M.name, M.year HAVING AVG(R.rating) = (SELECT MAX(AVG) FROM (SELECT R.movieId, AVG(R.rating) AS AVG FROM review R GROUP BY R.movieID)) ORDER BY M.mid ASC;

```
MID        TITLE   YEAR AVG(R.RATING) NUM_RATING
---------- ---------------------------------------
M11        Lucy     2015                      9.5   2

M3        My big fat greek wedding  2000       9.5   2
```

3. Print all information of the movies that have both a) the highest number of ratings; and b) the lowest average.

Solution:

```
SELECT M.mid, M.name, M.year, AVG(R.rating), COUNT(R.rating) FROM movie M, review R WHERE R.movieID = M.mid  GROUP BY M.mid, M.name, M.year HAVING COUNT(R.RATING) >= ALL(SELECT COUNT(R2.rating) FROM review R2 GROUP BY R2.movieId) INTERSECT
SELECT M.mid, M.name, M.year, AVG(R.rating), COUNT(R.rating) FROM review R, movie M WHERE R.movieId = M.mid GROUP BY M.mid, M.name, M.year HAVING AVG(R.rating) = (SELECT MIN(AVG) FROM (SELECT R.movieId, AVG(R.rating) AS AVG FROM review R GROUP BY R.movieID));
```

OutPut:

```
MID         NAME
------------- ----------------------------------------------------
-----
      YEAR AVG(R.RATING) COUNT(R.RATING)
---------- ------------- ---------------
M13         The God Father part II
      1975         4.5         2
```

4. A decade is any sequence of 10 consecutive years (e.g., 2000, 2001, ..., 2010 is a decade). Find the decade with the largest number of films (output only the first year of the decade).

Solution:

```
SELECT 10*FLOOR(M.year/10) AS MOVIE_DECADE, COUNT(10*FLOOR(M.year/10)) AS MOVIEDECADECOUNT FROM movie M GROUP BY (10*FLOOR(M.year/10)) HAVING COUNT(10*FLOOR(M.year/10)) >= ALL(SELECT COUNT(10*FLOOR(M1.year/10)) FROM movie M1 GROUP BY (10*FLOOR(M1.year/10)));
```

OutPut:

```
MOVIE_DECADE MOVIEDECADECOUNT
------------ ----------------
     2010         6
```

5. Find the film(s) with the largest cast. Return the movie title and the size of the cast. By "cast size" we mean the number of distinct actors that played in that movie: if an actor played multiple roles, or if the actor is simply listed more than once in CASTS, we still count her/him only once.

Solution:

SELECT * FROM (SELECT M.name AS Movie_Name, COUNT(DISTINCT R.actorId) AS Size_Of_Cast FROM roles R, movie M WHERE R.movieID = M.mid GROUP BY R.movieId, M.name ORDER BY COUNT(DISTINCT R.actorId) DESC) WHERE ROWNUM <= 1 ;

OutPut:

```
MOVIE_NAME                                SIZE_OF_CAST
----------------------------------------------------------
Lucy                                      4
```

6. Find actors that played five or more *distinct* roles in the same movie during the year 2010. Write a query that returns the actors' names, the movie name, and the number of distinct roles that they played in that movie (which will be ≥ 5).

Solution:

SELECT A.name AS ACTOR_NAME,M.name AS MOVIE_NAME, COUNT(DISTINCT R.role) AS NUM_DIST_ROLE FROM roles R , movie M, actDirect A WHERE R.movieId = M.mid AND M.year = 2010 AND R.actorId = A.actDrId GROUP BY A.name, M.name HAVING COUNT(DISTINCT R.role) >= 5;

OutPut:

```
ACTOR_NAME  MOVIE_NAME                                NUM_DIST_ROLE

Tom Hanks   The Polar Express                             6
```

7. Print the movie year, title and rating of the highest rated movie for each years in the period 2005-present, inclusive, in ascending year order. In case of a tie, print all, sorted in ascending alpha order on the title.

Solution:

CREATE VIEW V_AVGRATING AS SELECT M.year, M.name, AVG(R.rating) AS AVG FROM movie M, review R WHERE M.year >= 2005 AND M.mid = R.movieId GROUP BY M.year, M.name ORDER BY M.year;

CREATE VIEW MAX_RATING AS SELECT A.year, MAX(A.AVG) AS MAXRATING FROM V_AVGRATING A WHERE A.year >= 2005 GROUP BY A.year;

SELECT A.name, A.year, A.AVG FROM V_AVGRATING A, MAX_RATING M WHERE A.year = M.year AND A.AVG >= M.MAXRATING AND M.year >= 2005;

Solution:

```
NAME                                               YEAR
-------------------------------------------------------- ---------
-
       AVG----------
The Da Vinci Code                                  2005
     9

Angels and Daemons                                 2009
     8

The Island                                     2010
     7


NAME                                               YEAR
-------------------------------------------------------- ---------
-
       AVG
----------
The Polar Express                                  2010
     7

Her                                        2013
     8

Now You see me                                     2013
     8


NAME                                               YEAR
-------------------------------------------------------- ---------
-
       AVG
----------
Barely Lethal                                      2014
     8

Lucy                                       2015
       9.5
```

8. Find out who are the “no flop” actors: we will define a “no flop” actor as one who has played only in movies which have a rating greater than or equal to 8. Split this problem into the following steps.
Create a view called high ratings which contains the distinct names of all actors who have played in movies with a rating greater than or equal to 8. Similarly, create a view called low ratings which contains the distinct names of all actors who have played in movies with a rating less than 8.

Solution:

```
CREATE VIEW HIGH_RATINGS AS SELECT DISTINCT A.name FROM movie M, actDirect A, roles R1, review R WHERE M.mid = R.movieId AND M.mid = R1.movieId AND R1.actorId = A.actDrId GROUP BY A.name, M.mid HAVING AVG(R.rating) >= 8;

CREATE VIEW LOW_RATINGS AS SELECT DISTINCT A.name FROM movie M, actDirect A, roles R1, review R WHERE M.mid = R.movieId AND M.mid = R1.movieId AND R1.actorId = A.actDrId GROUP BY A.name, M.mid, A.actDrId HAVING AVG(R.rating) < 8;
```

Print a) the number of rows in the view high ratings and b) the number of rows in the view low ratings

Solution:

```
SELECT COUNT(*) AS COUNT_HIGH_RATING FROM HIGH_RATINGS;
```

OutPut:

```
COUNT_HIGH_RATING
-----------------
                9
```

```
SELECT COUNT(*) AS COUNT_LOW_RATING FROM LOW_RATINGS;
```

OutPut:

```
COUNT_LOW_RATING
----------------
               9
```

Use the above views to print the number of “no flop” actors in the database.

Solution:

SELECT COUNT(H.name) AS NO_FLOP_ACTOR FROM HIGH_RATINGS H WHERE H.name NOT IN (SELECT L.name FROM LOW_RATINGS L);

OutPut:

```
NO_FLOP_ACTOR
-------------
            4
```

Finally, use the above view to print the names of these “no flop” actors, along with the number M of movies they have played in, sorted by descending M and then by ascending name, and print only the top 10.

Solution:

SELECT * FROM (SELECT H.name NO_FLOP_ACTOR, COUNT(R.movieId) AS NUM_MOVIE_ACTED FROM HIGH_RATINGS H, roles R, actDirect A WHERE H.name = A.name AND A.actDrId = R.actorID AND H.name IN (SELECT * FROM (SELECT * FROM HIGH_RATINGS MINUS SELECT * FROM LOW_RATINGS)) GROUP BY H.name ORDER BY H.name ASC) WHERE ROWNUM <= 10;

OutPut:

```
NO_FLOP_ACTOR                                      NUM_MOVIE_ACTED
-------------------------------------------------- ---------
Alex Parish                                                1
Angelina Jolie                                             3
Jennifer Lawrence                                          1
Scarlett Johanson                                          3
```

9. Print the names of all actors who have starred in all movies in which *Al pacino* has starred in (it’s ok to report the name of *Al pacino* in the result; also, it is ok if these actors have starred in more movies than *Al pacino* has played in).

Solution:

SELECT A.name FROM actDirect A, roles R WHERE R.actorId = A.actDrId AND R.movieID IN (SELECT M1.mid FROM movie M1, roles R1, actDirect A1 WHERE M1.mid = R1.movieID and R1.actorId = A1.actDrId AND A1.name = 'Al Pacino') GROUP BY A.name;

OutPut:

```
NAME
------------------------------------------------------------
Scarlett Johanson
Julia Roberts
Al Pacino
Angelina Jolie
Brad Pitt
Morgan Freeman
```

10. Find out who is the actor with the highest “longevity.” Print the name of the actor/actress who has been playing in movies for the longest period of time (i.e. the time interval between their first movie and their last movie is the greatest.

Solution:

SELECT A.name, (MAX(M.year) - MIN(M.YEAR)) AS Longevity FROM roles R, movie M, actDirect A WHERE M.mid = R.movieId AND R.actorId = A.actDrId GROUP BY A.name HAVING (MAX(M.year) - MIN(M.YEAR)) >= ALL(SELECT (MAX(M1.year) - MIN(M1.YEAR)) FROM roles R1, movie M1 GROUP BY R1.actorId);

OutPut :

```
NAME                                    LONGEVITY
------------------------------------------------------------
Brad Pitt                                       50
Morgan Freeman                                  50
```