**Movies- Database Assignment**

**1. Create a database “Movies”.**

**2. Create a “mov” schema under “Movies”.**

**3. TABLE 1: Create a table call “Movie\_Director “under mov schema with the following** **specifications:**

**a. Movie\_Director must have the following attribute.**

Column Name Data Type Description

Director\_ID Integer Director ID

Director\_First\_Name Varchar Director First Name

Director\_Last\_Name Varchar Director Last Name

Director\_Age\_in\_Years Integer Director Age in Year

Director\_Gender Varchar Date of Joining

CreatedOn Date Record Creation Date

**b. Add the following constraint.**

i. Director\_ID: Auto Increment, Primary Key, Not null, clustered index.

ii. CreatedOn: Not Null, Default as Server date.

**c. Insert the following records based on the following specifications.**

i. Director\_ID: Director ID must be starts from 100 and incremented by 10.

ii. CreatedOn: Created as default date as system date.

**d. After Insertion, Data looks like-**

Director\_ID Director\_First\_Name Director\_Last\_Name Director\_Age\_in\_Years Director\_Gender

100 Kevin Smith 52 Male

110 Miguel Arteta 41 Male

120 Mark Johnson 58 Male

130 Tom Vaughan 37 Male

140 Francis Lawrence 52 Male

150 Adrienne Shelly 40 Female

160 David Slade 53 Male

170 Mark Palansky 53 Male

180 Jeff Lowell 49 Male

190 Simon Curtis 37 Male

200 Marc Lawrence 95 Male

210 Anand Tucker 59 Male

220 Judd Apatow 55 Male

230 Cary Fukunaga 45 Male

240 Mark Helfrich 49 Male

250 Nanette Burstein 52 Female

260 James McAvoy 44 Male

270 Mark Waters 58 Male

280 Seth Gordon 46 Male

290 Alex Kendrick 52 Male

300 Kevin Lima 60 Male

310 Lasse Hallström 76 Male

320 Ewan McGregor 52 Male

330 Rajkumar Hirani 60 Male

340 Ashutosh Gowariker 59 Male

350 Karan Johar 50 Male

360 S.S Rajamouli 49 Male

370 Sukumar NULL 53 Male

380 Aditya Chopra 51 Male

390 Umesh Shukla 52 Male

**4. TABLE 2: Create a Movies table under mov schema with the following specifications-**

**a. Movies table must have the following attributes.**

Column Name Data Type Description

Movie\_ID Integer Director ID

Movie\_Name Varchar Movie Name

Movie\_Released\_Year Integer Movie Released Year

Movie\_Lead\_Studio Varchar Movie Released By

Movie\_Language Varchar Movie Language

Movie\_Category Varchar Movie Category

Movie\_Duration\_in\_Min Integer Movie Duration in minutes

Movie\_Worldwide\_Earning\_in\_$M

Decimal (15,2) Movie Worldwide earnings

Movie\_Type Varchar Movie Type

Director\_ID Integer Director ID

CreatedOn Date Record Creation Date

**a. Add the following constraint.**

i. Movie\_ID: Auto Increment, Primary Key, Not null, clustered index.

ii. CreatedOn: Not Null, Default as Server date.

**b. Insert the following records based on the following specifications.**

i. Movie\_ID: Movie ID must be starts from 1000 and incremented by 1.

ii. CreatedOn: Created as default date as system date.

iii. Director\_ID: Foreign key from Movie\_Director table.

iv. Movie\_Type: Movie Type as Hollywood and Bollywood.

c. After Insertion, Table looks like as shown below-

Movie\_ ID Movie\_ Name

Movie\_

Released\_

Year

Movie\_Lead\_

Studio

Movie\_

Language

Movie\_

Category

Movie\_

Duration\_

in\_Min

Movie\_

Worldwide

\_Earning\_

in\_$M

Movie\_

Type Director\_ID

1000

Zack and Miri

Make a Porno 2008

The Weinstein

Company English Romance 101 41.94 Hollywood 100

1001 Youth in Revolt 2010

The Weinstein

Company English Comedy 90 19.62 Hollywood 110

1002 When in Rome 2010 Disney English Comedy 91 43.04 Hollywood 120

1003

What Happens

in Vegas 2008 Fox English Comedy 99 219.37 Hollywood 130

1004

Water For

Elephants 2011 20th Century Fox English Drama 120 117.09 Hollywood 140

1005 Waitress 2007 Independent English Romance 108 22.18 Hollywood 150

1006 Twilight 2008 Summit English Romance 122 376.66 Hollywood 160

1007 Penelope 2008 Summit English Comedy 144 20.74 Hollywood 170

1008

Over Her Dead

Body 2008 New Line English Comedy 95 20.71 Hollywood 180

1009

My Week with

Marilyn 2011

The Weinstein

Company English Drama 99 8.26 Hollywood 190

1010 Music and Lyrics 2007 Warner Bros. English Romance 104 145.9 Hollywood 200

1011 Leap Year 2010 Universal English Comedy 100 32.59 Hollywood 210

1012 Knocked Up 2007 Universal English Comedy 129 219 Hollywood 220

1013 Jane Eyre 2011 Universal English Romance 120 30.15 Hollywood 230

1014

Good Luck

Chuck 2007 Lionsgate English Comedy 101 59.19 Hollywood 240

1015

Going the

Distance 2010 Warner Bros. English Comedy 103 42.05 Hollywood 250

1016

Gnomeo and

Juliet 2011 Disney English Animation 84 193.97 Hollywood 260

1017

Ghosts of

Girlfriends Past 2009 Warner Bros. English Comedy 100 102.22 Hollywood 270

1018

Four

Christmases 2008 Warner Bros. English Comedy 88 161.83 Hollywood 280

1019 Fireproof 2008 Independent English Drama 122 33.47 Hollywood 290

1020 Enchanted 2007 Disney English Comedy 107 340.49 Hollywood 300

1021 Dear John 2010 Sony English Drama 108 114.97 Hollywood 310

1022 Beginners 2011 Independent English Comedy 105 14.31 Hollywood 320

1023 3 Idiots 2009 Vinod Chopra Films Hindi Comedy 171 4000 Bollywood 330

1024 Lagaan 2001

Aamir Khan

Productions Hindi Romance 224 659 Bollywood 340

1025

My Name Is

Khan 2010 Dharma Productions Hindi Drama 165 48.77 Bollywood 350

1026 Baahubali 2015 Arka Media Works Telugu Thriller 159 6500 Bollywood 360

1027

Dilwale Dulhania

Le Jayenge 1995 Yash Chopra Hindi Romance 189 2000 Bollywood 380

1028 Oh My God 2012 Hindi Comedy 165 1200 Bollywood 390

1029 Pushpa 2021

Mythri Movie

Makers Telugu Drama 179 3730 Bollywood 370

**5. TABLE 3: Create a Movie\_Actor table under mov schema with the following specifications-**

**a. Movie\_Actor table must have the following attributes.**

Column Name Data Type Description

Actor\_ID Integer Actor ID

Actor\_First\_Name Varchar Actor First Name

Actor\_Last\_Name Varchar Actor Last Name

Actor\_Age\_in\_Years Integer Actor Age in Years

Actor\_Location Varchar Actor Location

Movie\_ID Integer Movie ID

CreatedOn Date Record Creation Date

**b. Add the following constraint.**

i. Actor\_ID: Auto Increment, Primary Key, Not null, clustered index.

ii. CreatedOn: Not Null, Default as Server date.

**c. Insert the following records based on the following specifications.**

i. Actor\_ID: Actor ID must be starts from 10 and incremented by 1.

ii. CreatedOn: Created as default date as system date.

iii. Movie\_ID: Foreign key from Movies table.

**d. After Insertion, Table looks like as shown below-**

Actor\_I

D

Actor\_First\_Nam

e Actor\_Last\_Name

Actor\_Age\_in\_Year

s Actor\_Location Movie\_ID

10 Seth Rogen 53 Los Angeles 1000

11 Michael Cera 49 New York 1001

12 Josh Duhamel 37 North Dakota 1002

13 Jason Sudeikis 60 Kansas 1003

14 Robert Pattinson 45 Los Angeles 1004

15 Nathan Fillion 55 Canada 1005

16 Robert Pattinson 45 Los Angeles 1006

17 James McAvoy 49 Scotland 1007

18 Paul Rudd 52 New York 1008

19 Kenneth Branagh 44 Northern Ireland 1009

20 Hugh Grant 58 London 1010

21 Matthew Goode 46 England 1011

22 Judd Apatow 58 Los Angeles 1012

23 Michael Fassbender 46 Germany 1013

24 Dane Cook 52 United States 1014

25 Jason Sudeikis 60 Kansas 1015

26 Kelly Asbury 76 United States 1016

27 Matthew McConaughey 52 Los Angeles 1017

28 Vince Vaughn 60 Minnesota 1018

29 Kirk Cameron 59 United States 1019

30 James Marsden 50 Columbia 1020

31 Channing Tatum 58 Alabama 1021

32 Mike Mills 37 New York 1022

33 Aamir Khan 52 India 1023

34 Aamir Khan 52 India 1024

35 Shah Rukh Khan 53 India 1025

36 Prabhas NULL 53 India 1026

37 Allu Arjun 49 India 1027

38 Shah Rukh Khan 53 India 1028

39 Akshay Kumar 50 India 1029

**6. Table 4: Create a Movie\_Rating table under mov schema with the following specifications-**

**a. Movie\_Rating table must have the following attributes.**

Column Name Data Type Description

Movie\_Rating\_ID Integer Movie Rating ID

Rating\_Audience\_Score Varchar Actor First Name

Rating\_Rotten\_Tomatoes Varchar Actor Last Name

Movie\_ID Integer Movie ID

CreatedOn Date Record Creation Date

**b. Add the following constraint.**

i. Movie\_Rating\_ID: Auto Increment, Primary Key, Not null, clustered index.

ii. CreatedOn: Not Null, Default as Server date.

**c. Insert the following records based on the following specifications.**

iv. Movie\_Rating\_ID: Actor ID must be starts from 1 and incremented by 1.

v. CreatedOn: Created as default date as system date.

vi. Movie\_ID: Foreign key from Movies table.

**d. After Insertion, Table looks like as shown below-**

Movie\_Rating\_ID Rating\_Audience\_Score Rating\_Rotten\_Tomatoes Movie\_ID

1 70 64 1000

2 52 68 1001

3 44 15 1002

4 72 28 1003

5 72 60 1004

6 67 89 1005

7 82 49 1006

8 74 52 1007

9 47 15 1008

10 84 83 1009

11 70 63 1010

12 49 21 1011

13 83 91 1012

14 77 85 1013

15 61 3 1014

16 56 53 1015

17 52 56 1016

18 47 27 1017

19 52 26 1018

20 51 40 1019

21 80 93 1020

22 66 29 1021

23 80 84 1022

24 95 100 1023

25 81 95 1024

26 79 83 1025

27 80 90 1026

28 76 82 1027

29 85 100 1028

30 81 74 1029

**7. Write the following Query based on the above datasets. (Fiza)**

a. List all the Movies information from the Movies table.

b. List all the Director information from the Director table.

c. List all the Actor information from the Actor table.

d. List all the Rating information from the Rating table.

e. List all the movie released in year “2010”.

f. List all the movie released by “The Weinstein Company” studio.

g. List all the movie released in “English”.

h. List all the movie whose name starts with ‘G’.

i. Display all the movie under “Comedy” category.

j. Display all the movie where movie type is “Hollywood”.

k. Display all the “Female” directors.

l. Display all the director whose Age is more than 45 years.

m. Display all the Actors from “Los Angeles”.

n. Display all the Actor whose Age is less than 50 years.

o. Display all the Actor whose name is starts from “J”.

p. Display all the Actor who is from “Los Angeles” or “New York”.

q. List Director\_FullName, Director\_Age\_in\_Years, Director\_Gender from Director

[Director\_FullName =Director\_First\_Name + " " + Director\_Last\_Name]

r. List Director\_FullName, Director\_Age\_in\_Years, Director\_Gender from Director whose Age is

less than 45 years. [Director\_FullName =Director\_First\_Name + " " + Director\_Last\_Name]

**8. Write the following Query based on the above datasets. (Anushiya)**

a. Display all the Movies and their Actors information based on the relationship.

b. Display the Movies name and their Ratings.

c. Display all the Movies, Actors, and Directors information based on the relationship.

d. Display all the Movies, Actors, Directors, and Movie Rating information based on the

relationship.

e. Display all the Movies, Actors, Directors, and Movie Rating information whose

Rating\_Audience\_Score is more than 80% based on the relationship.

f. Display all the Movies information whose Rating\_Rotten\_Tomatoes is more than 90%.

**9. Write the following Query based on the above datasets. (Fiza)**

a. Create new table MovieCopy and copy all records of Movie table.

b. Create a new table DirectorCopy and copy only the schema of director table.

c. Create new table ActorCopy and copy all records of Actor table.

d. Create new table RatingCopy and copy all records of Rating table.

e. Create new table RatingCopies and copy only the schema from Rating table.

**10. Write the following Query based on the above datasets. (Fiza)**

a. Delete all the record from RatingCopy table.

b. Delete all the movie from MovieCopy whose released in year “2010”.

c. Delete all the movie from MovieCopy where language is ‘Hindi.

d. Delete all the movie from MovieCopy where Rating\_Audience\_Score is less than 80%.

e. Delete all the movie from MovieCopy where Rating\_Rotten\_Tomatoes is less than 70%.

**11. Write the following Query based on the above datasets. (Anushiya)**

a. Update the Rating\_Audience\_Score by 85% for the movie, released by “The Weinstein

Company” studio.

b. Update the Rating\_Rotten\_Tomatoes by 75% for the movie, released in Year 2010.

c. Increase the Actor Age by 2 years whose Name is “Michael Cera”.

d. Increase the Director Age by 3 years who has directed the movie “Leap Year”.

e. Increase the Director and Actor Age by 1 years who has directed the movie “Leap Year”.

**12. Write the following Query based on the above datasets. (Anushiya)**

a. Create a view to display all the movie information’s.

b. Create a view to display all the movies and their rating information.

c. Create a view to display all the movies and their actor information.

d. Create a view to display all the movies, rating, actor along with director information.

e. Create a view to display all the information based on the result set returned by the query as

shown below-

List movie, Director\_FullName, Director\_Age\_in\_Years, Director\_Gender from Director

[Director\_FullName =Director\_First\_Name + " " + Director\_Last\_Name]

**13. Write the following Query based on the above datasets. (Nicole)**

a. Retrieve the list of all Databases.

b. Display the byte size of all tables in databases.

c. List of tables with number of records.

d. List of Primary Key and Foreign Key for Whole Database.

e. Get all Nullable columns of a table.

f. Get All table that do not have primary key.

g. Get All table that do not have identity column.

h. Get First Date of Current Month.

i. Get Last date of Current month.

j. Get first date of next month.

k. Get Last date of next month.

l. Get all the information from the tables.

m. Get all columns contain any constraints.

n. Get all tables that contain a view.

o. Get all columns of table that using in views.