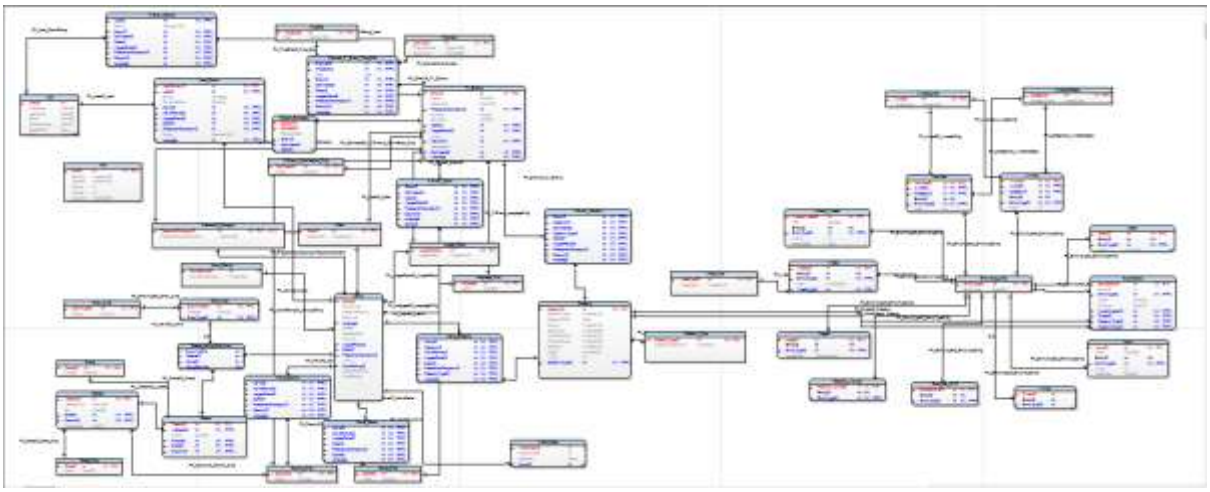
DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

Database Design and Data Management - Fall 2016

Final Project – Modelling IMDB database

ER model:



Microsoft SQL Server:

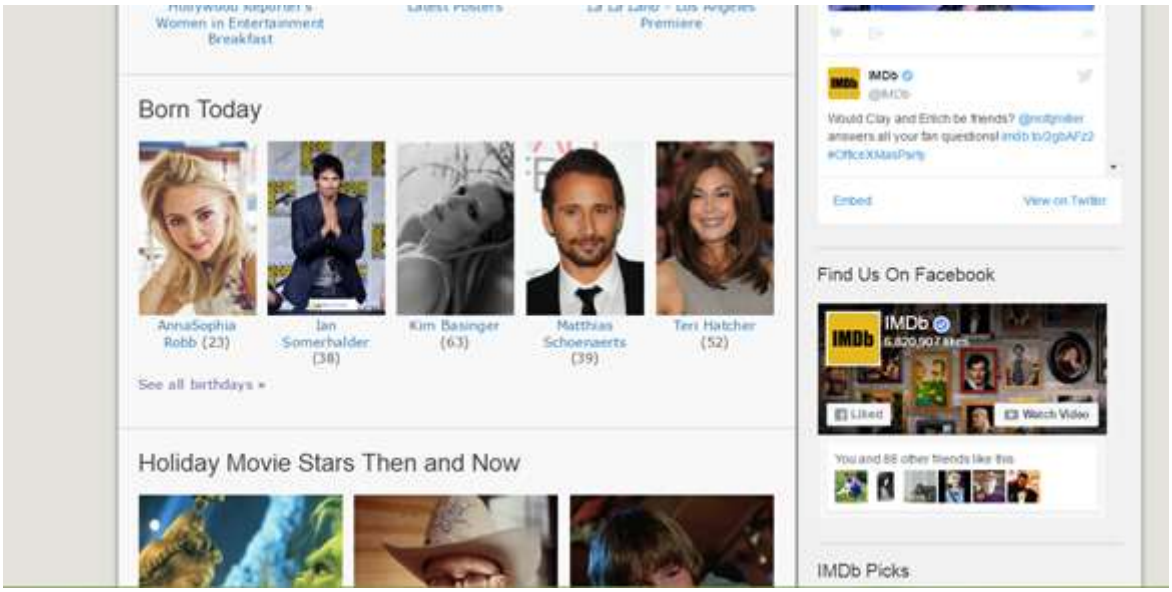
1) Celebrities born today

```
CREATE FUNCTION ufn_CurrentBirthdateAge()
Returns @CelebTable TABLE(
AGE INT,
CELEB_Name Varchar(40),
CELEB_Alternate_Name VARCHAR(40),
Personal_Quote VARCHAR(200)
)
AS
BEGIN


INSERT INTO @CelebTable
SELECT DATEDIFF(Year,CelebrityDOB,GETDATE())-

              CASE

              WHEN(MONTH(CelebrityDOB)>MONTH(GETDATE()) OR
              (MONTH(CelebrityDOB)>MONTH(GETDATE()) AND DAY(CelebrityDOB)>DAY(GETDATE())))
              THEN 1
              ELSE 0 END As AGE,CelebrityName,AlternateName,PersonalQuote from
Celebrity.Celebrity
where day(CelebrityDOB) = day(GETDATE()) AND MONTH(CelebrityDOB)  =MONTH(GETDATE())

RETURN
```

DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

```
END

SELECT * FROM ufn_CurrentBirthdateAge()
```

	AGE	CELEB_Name	CELEB_Alternate_Name	Personal_Quote
1	32	Leonardo Di Caprio	Lenny D	I think everyone wants to be recognized by th...
2	37	Quentin Tarantino	Q	As far as I'm concerned, digital projection is the...
3	24	Selena Gomez	Sel	It's really toxic what they do to the girls in the i...



2) Finding movies or tv shows common to multiple celebrities

```
CREATE PROC usp_FindMoviesForCelebrities @names varchar(100)
AS

	CREATE TABLE #temp (names varchar(100))
	DECLARE @len int
	SET @len = LEN(@names)
	DECLARE @charind int
	SELECT @charind = CHARINDEX(' ',@names)
	while(@charind>0)
	BEGIN
		INSERT INTO #temp SELECT substring(@names, 1,charindex(' ',@names));
		SELECT @len = len(@names)-charindex(' ',@names);
		SELECT  @names = SUBSTRING(@names, charindex(' ',@names)+1, @len);
		SELECT @charind = CHARINDEX(' ',@names)
	END
	INSERT INTO #temp values(@names)

	CREATE TABLE #result (Name varchar(100), Description varchar(200), Movie_Rating
varchar(10))
	CREATE TABLE #result1 (Name varchar(100), Description varchar(200), Movie_Rating
varchar(10))
```

DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

```
	DECLARE @name varchar(50)

	SELECT TOP 1 @name = names From #temp t

	INSERT INTO #result
	SELECT m.MovieName, m.StoryLine, mr.MovieRatingName FROM Movies.Movie m
	JOIN Celebrity.Movie_Celebrity mc ON mc.MovieID = m.MovieID
	JOIN Celebrity.Celebrity c ON c.CelebrityID = mc.CelebrityID
	JOIN Movies.Movie_Rating mr ON mr.MovieRatingID = m.MovieRatingID
	WHERE c.CelebrityName LIKE '%'+@name+'%'

	DELETE #temp WHERE #temp.names = @name

	WHILE(SELECT COUNT(*) FROM #temp t) >0
	BEGIN
		SELECT TOP 1 @name = names From #temp t

		DELETE FROM #result1

		INSERT INTO #result1
		SELECT m.MovieName, m.StoryLine, mr.MovieRatingName FROM Movies.Movie m
		JOIN Celebrity.Movie_Celebrity mc ON mc.MovieID = m.MovieID
		JOIN Celebrity.Celebrity c ON c.CelebrityID = mc.CelebrityID AND
c.CelebrityName LIKE '%'+@name+'%'
		JOIN Movies.Movie_Rating mr ON mr.MovieRatingID = m.MovieRatingID
		JOIN #result r ON r.Name = m.MovieName

		DELETE #temp WHERE #temp.names = @name

		DELETE FROM #result
		INSERT INTO #result SELECT * FROM #result1 r

	END

	SELECT * FROM #result r

	DROP table #temp

GO


EXEC dbo.usp_FindMoviesForCelebrities @names = 'Lee Leonardo'
```

	Name	Description	Movie_Rating
1	A Christmas Tale (2008)	The troubled Vuillard family is no stranger to illness,...	PG
2	Agents of S.H.I.E.L.D	The missions of the Strategic Homeland Interventio...	TV-PG
3	Allied(2016)	In 1942, an intelligence officer in North Africa enco...	R
4	Bad Santa 2 (2016)	Fueled by cheap whiskey, greed and hatred, Willie...	R
5	Bones	Forensic anthropologist, Dr. Temperance "Bones" ...	TV-14
6	Sherlock	A modern update finds the famous sleuth and his d...	TV-14
7	Supernatural	Two brothers follow their fathers footsteps as "hunt...	TV-14
8	Westworld	A series inspired by the 1973 film of the same title	TV-MA

3) List the Award Winners for a particular year and award

DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

```
CREATE PROC usp_GetAwardWinners @awardID int, @year int
AS
SELECT ac2.CategoryName, w.[Year],
CASE
       WHEN w.EntityTypeID = 1 THEN m.MovieName
       ELSE c.CelebrityName
END Winner
FROM Awards.Winners w
JOIN Awards.Awards_Cnfg ac ON ac.AwardID = w.AwardID
JOIN Awards.Award_Category ac2 ON ac2.CategoryID = w.CategoryID
LEFT JOIN Movies.Movie m ON m.MovieID = w.EntityID
LEFT JOIN Celebrity.Celebrity c ON c.CelebrityID =w.EntityID
WHERE w.[Year] = @year AND w.AwardID = @awardID
GO

EXEC dbo.usp_GetAwardWinners
       @awardID = 1,
       @year = 2011
GO
```

	CategoryName	Year	Winner
1	Best Supporting Actor	2011	Jason Statham
2	Best Supporting Actress	2011	Emma Watson

4) List the Award Nominees for a particular year and award

```
CREATE PROC usp_GetAwardNominees @awardID int, @year int
AS
SELECT ac2.CategoryName, w.[Year],
CASE
       WHEN w.EntityTypeID = 1 THEN m.MovieName
       ELSE c.CelebrityName
END Winner
FROM Awards.Nominee w
JOIN Awards.Awards_Cnfg ac ON ac.AwardID = w.AwardID
JOIN Awards.Award_Category ac2 ON ac2.CategoryID = w.CategoryID
LEFT JOIN Movies.Movie m ON m.MovieID = w.EntityID
LEFT JOIN Celebrity.Celebrity c ON c.CelebrityID =w.EntityID
WHERE w.[Year] = @year AND w.AwardID = @awardID
ORDER BY ac2.CategoryName
GO

EXEC dbo.usp_GetAwardNominees
       @awardID = 1,
       @year = 2011
```

DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

	CategoryName	Year	Winner
1	Best Supporting Actor	2011	Leonardo Di Caprio
2	Best Supporting Actor	2011	Matthew McConaughey
3	Best Supporting Actress	2011	Quentin Tarantino
4	Best Supporting Actress	2011	Jason Statham

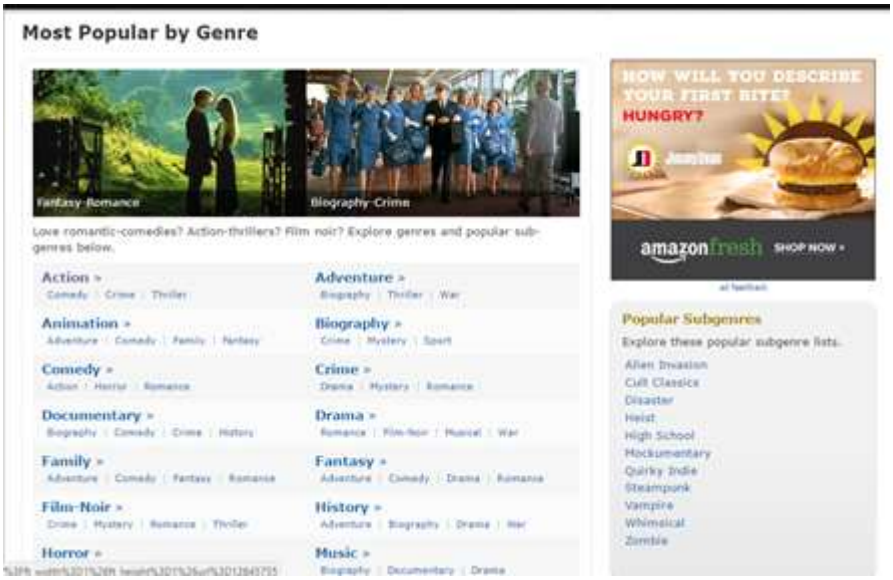
5) Most Popular movies for a given genre

```
IF EXISTS (SELECT * FROM sys.procedures p WHERE name = 'usp_PopularMovieGenre')

DROP PROCEDURE usp_PopularMovieGenre

GO

CREATE PROC usp_PopularMovieGenre @GenreName  varchar(20)
AS
 BEGIN

 SELECT m.MovieName,
        m.MetacriticReview,
        m.StoryLine,
        gc.Name GenreName
 FROM Movies.Movie m
 JOIN Details.Movie_Genre mg
 ON m.MovieID = mg.MovieID
 JOIN Movies.Genre_Cnfg gc
 ON mg.GenreID = gc.GenreID
 WHERE gc.Name = @GenreName
 ORDER BY m.Views DESC

END
GO
```

```
EXEC usp_PopularMovieGenre Drama
```

	MovieName	MetacriticReview	StoryLine	GenreName
1	No Country for Old Men (2007)	91	Violence and mayhem ensue after a hunter stumbl...	Drama
2	Love Actually (2003)	59	Follows the lives of eight very different couples in d...	Drama
3	Minority Report (2002)	80	In a future where a special police unit is able to arr...	Drama
4	Shivaay (2016)	65	Shivaay is a Himalayan mountaineer who is an inn...	Drama

DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan



6) List of TV shows aired on a particular date and channel

```
IF EXISTS (SELECT * FROM sys.objects WHERE name = 'usp_TVShowsOnTonight')

DROP PROCEDURE usp_TVShowsOnTonight

GO

CREATE PROC usp_TVShowsOnTonight @ChannelName varchar(20),@Date date
AS
 BEGIN

 SELECT ts.ShowID,
       ts.Name AS ShowName,
          c.ChannelName,
          ctsts.[Date],
          t.Slot
 FROM TVShows.TV_Shows ts
 JOIN TVShows.Channel_TV_Show_Time_Slot ctsts
 ON ts.ShowID = ctsts.ShowID
 JOIN TVShows.Channels c
 ON ctsts.ChannelID = c.ChannelID
 JOIN TVShows.Timeslot t
 ON ctsts.TimeSlotID = t.TimeSlotID
 WHERE c.ChannelName = @ChannelName AND ctsts.[Date] = @Date

END


EXEC usp_TVShowsOnTonight Fox,'2016-01-07'
```

	ShowID	ShowName	ChannelName	Date	Slot
1	34	House of Cards	Fox	2016-01-07	04.30

DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

7) List recently viewed entities

```
CREATE FUNCTION ufn_RecentlyViewed()
RETURNS @RecentlyViewed TABLE (Recently_Viewed varchar(50))
AS
BEGIN

INSERT INTO @RecentlyViewed
SELECT
CASE
       WHEN rv.EntityTypeID = 1 THEN m.MovieName
       WHEN rv.EntityTypeID = 2 THEN c.CelebrityName
       ELSE ts.Name
END RecentlyViewed
FROM Details.Recently_Viewed rv
LEFT JOIN Movies.Movie m ON rv.EntityID = m.MovieID
LEFT JOIN Celebrity.Celebrity c ON rv.EntityID = c.CelebrityID
LEFT JOIN TVShows.TV_Shows ts ON rv.EntityID = ts.ShowID

RETURN
END


SELECT * FROM dbo.ufn_RecentlyViewed() urv
```

	Recently_Viewed
1	Allied(2016)
2	Westworld
3	Leonardo Di Caprio
4	Top Gear
5	Steven Price
6	Andrew Loog
7	Agents of S.H.I.E.L.D
8	Friends
9	Rooney Mara
10	A Christmas Tale (2008)
11	Two and a Half Men

DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

8) List all the news items

```
CREATE FUNCTION ufn_News()
RETURNS @News TABLE (News varchar(50), Entity varchar(50))
AS
BEGIN

INSERT INTO @News
SELECT rv.Headlines,
CASE
       WHEN rv.EntityTypeID = 1 THEN m.MovieName
       WHEN rv.EntityTypeID = 2 THEN c.CelebrityName
       ELSE ts.Name
END Entity
FROM Details.News rv
LEFT JOIN Movies.Movie m ON rv.EntityID = m.MovieID
LEFT JOIN Celebrity.Celebrity c ON rv.EntityID = c.CelebrityID
LEFT JOIN TVShows.TV_Shows ts ON rv.EntityID = ts.ShowID

RETURN
END


SELECT * FROM dbo.ufn_News() un
```

	News	Entity
1	Max budget Movie	Allied(2016)
2	Min budget Movie	Almost Christmas(2016)
3	Max budget Movie	Arrival(2016)
4	Min budget Movie	Bad Santa 2 (2016)
5	Max budget Movie	Doctor Strange(2016)
6	Min budget Movie	Fantastic Beasts and Where to Find Them (2016)
7	Graphics enhanced Movie	La La Land (2016)
8	Sound enhanced Movie	Office Christmas Party (2016)
9	Graphics enhanced Movie	Friend Request (2016)
10	Sound enhanced Movie	Harry Benson: Shoot First (2016)
11	Daring Actor	Leonardo Di Caprio

9) List all poll questions and top voted answers

```
CREATE FUNCTION ufn_Poll()
RETURNS @Poll TABLE (Question varchar(100), Answer varchar(50))
AS
BEGIN

INSERT INTO @Poll
SELECT p.Question, p.OptionA Option FROM [User].Poll p WHERE p.CountA > p.CountB
UNION
SELECT p.Question, p.OptionB Option FROM [User].Poll p WHERE p.CountB > p.CountA

RETURN
END


SELECT * FROM dbo.ufn_Poll() up
```

DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

	Question	Answer
1	Which is the best action movie?	Rocky
2	Which is the best animation movie?	Frozen
3	Which is the best Christmas movie?	Ghosts of girlfriend
4	Which is the best comedy movie?	Copout
5	Which is the best cook?	Jimmy Fallon
6	Which is the best football movie?	Messi
7	Which is the best horror movie?	Dont Disturb
8	Which is the best host?	Jimmy Fallon
9	Which is the best movie?	12 years a slave
10	Which is the best movie?	God Father
11	Which is the best romantic movie?	The notebook

10) Updating Gross Revenue for all movies

```
CREATE PROCEDURE uspTotalMovieGross
AS
SELECT Movies.Week_Gross.MovieID, SUM(Collection) Gross INTO #temp FROM Movies.Week_Gross
GROUP BY Movies.Week_Gross.MovieID
DECLARE @ID int
DECLARE @gross money
WHILE(SELECT COUNT(*) FROM #temp t) >0
BEGIN
	SELECT TOP 1 @ID = MovieID From #temp t
	SELECT @gross = t.Gross FROM #temp t WHERE t.MovieID = @ID
	UPDATE Movies.Movie SET Movies.Movie.Gross_Revenue = @gross WHERE Movies.Movie.MovieID
= @ID
	DELETE #temp WHERE #temp.MovieID = @Id
END
DROP table #temp
GO

EXEC dbo.uspTotalMovieGross

SELECT * FROM Movie
```

11) Calculating IMDB Rating for all movies

```
CREATE PROC uspCalculateTMDBRating
AS
SELECT MovieID, AVG(Rating) Rating INTO #temp FROM [User].User_Review GROUP BY MovieID
DECLARE @ID int
DECLARE @Rating decimal(18,2)
WHILE(SELECT COUNT(*) FROM #temp t) > 0
BEGIN
	SELECT TOP 1 @ID = MovieID From #temp t
```

```
        SELECT @Rating = t.Rating FROM #temp t WHERE t.MovieID = @ID
        UPDATE Movies.Movie SET Movies.Movie.IMDBRating = @Rating WHERE MovieID = @ID
        DELETE #temp WHERE #temp.MovieID = @Id
END
DROP table #temp
GO

EXEC uspCalculateTMDBRating

SELECT * FROM Movie
```

12) Calculating IMDB rating for a movie

```
CREATE PROC uspCalculateTMDBRatingByMovie @MovieID int
AS
DECLARE @Rating decimal(18,2)
SELECT @Rating = AVG(Rating) FROM [User].User_Review WHERE MovieID = @MovieID GROUP BY
MovieID
UPDATE Movies.Movie SET Movies.Movie.IMDBRating = @Rating WHERE MovieID = @MovieID
GO

EXEC uspCalculateTMDBRatingByMovie 2

SELECT * FROM Movie
```

13) Creating DB backup

```
CREATE PROC usp_DBBackup
AS

DECLARE @name VARCHAR(50) -- database name
DECLARE @path VARCHAR(256) -- path for backup files
DECLARE @fileName VARCHAR(256) -- filename for backup
DECLARE @fileDate VARCHAR(20) -- used for file name


-- specify database backup directory
SET @path = 'C:\IMDB_Images\'


-- specify filename format
SELECT @fileDate = CONVERT(VARCHAR(20),GETDATE(),112)
```

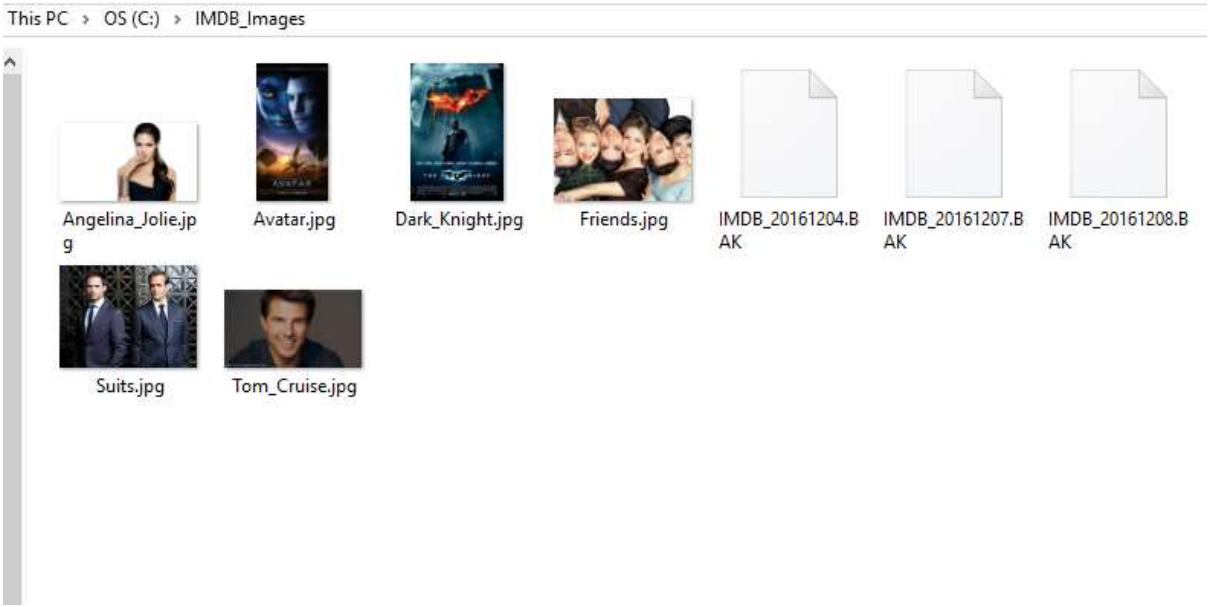
DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

```
DECLARE db_cursor CURSOR FOR
SELECT name
FROM master.dbo.sysdatabases
WHERE name = 'IMDB'


OPEN db_cursor
FETCH NEXT FROM db_cursor INTO @name


WHILE @@FETCH_STATUS = 0
BEGIN
      SET @fileName = @path + @name + '_' + @fileDate + '.BAK'
      BACKUP DATABASE @name TO DISK = @fileName


      FETCH NEXT FROM db_cursor INTO @name
END


CLOSE db_cursor
DEALLOCATE db_cursor


GO

EXEC usp_DBBackup
```

Messages

```
Processed 1208 pages for database 'IMDB', file 'IMDB' on file 2.
Processed 8 pages for database 'IMDB', file 'IMDB_log' on file 2.
BACKUP DATABASE successfully processed 1216 pages in 0.606 seconds (15.670 MB/sec).
```

DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

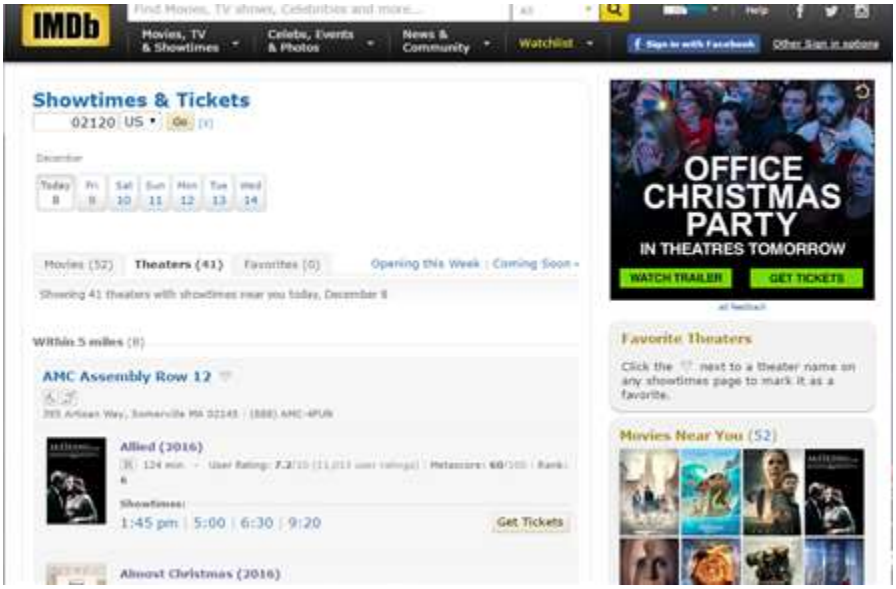


14) Finding theatres and movies based on a required location

```
CREATE PROC usp_Currently_Running_Location_Wise    @zipCode varchar(30),@countryName
varchar(30)
AS
BEGIN

select theatre.TheatreID,theatre.[Name] TheaterName,
RTRIM(fr.AddressLine)+','+RTRIM(fr.City)+','+RTRIM(fs.[Name])+','+RTRIM(fc.[Name])+','+RTRIM(
fr.PostalCode) AS TheaterAddress,
movie.MovieName,movie.RunTime,movie.MetacriticReview,mr.MovieRatingName,ts.Timeslot into
tempTable from Theatre.Theatre theatre
Join Theatre.Theatre_Movie_Showtime tms on  theatre.TheatreID=tms.TheatreID
Join Movies.Movie movie on movie.MovieID=tms.MovieID
Join Theatre.Showtimes ts on ts.ShowtimesID=tms.ShowtimesID
Join Movies.Movie_Rating mr on mr.MovieRatingID=movie.MovieRatingID
Join Features.[Address] fr on fr.AddressID=theatre.AddressID
Join Features.State_Cnfg fs on fs.StateID=fr.StateID
Join Features.Country_Cnfg fc on fc.CountryID=fr.CountryID where fr.PostalCode=@zipCode  AND
fc.Name=@countryName
order by theatre.TheatreID;


select distinct ST2.TheatreID,ST2.TheaterName,ST2.TheaterAddress
,ST2.MovieName,ST2.RunTime,ST2.MetacriticReview,ST2.MovieRatingName,
    substring(
        (
            Select ','+ST1.[Timeslot]
            From tempTable ST1
            Where ST1.MovieName = ST2.MovieName
            ORDER BY ST2.TheatreID,ST2.TheaterName,ST2.TheaterAddress
,ST2.MovieName,ST2.RunTime,ST2.MetacriticReview,ST2.MovieRatingName
            For XML PATH ('')
        ), 2, 100) [ShowTimes]

From tempTable ST2

drop table tempTable

END
```

DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

```
EXEC usp_Currently_Running_Location_Wise 'M4S2C6','Canada'
```



15) Finding theatres and movies based on user location

```
CREATE PROC  usp_Currently_Running_User_Location    @userID int
AS
BEGIN
Declare @userCity varchar(30);
select @userCity=City from [User].[User] where userid=@userID;

select theatre.TheatreID,theatre.[Name] TheaterName,
RTRIM(fr.AddressLine)+','+RTRIM(fr.City)+','+RTRIM(fs.[Name])+','+RTRIM(fc.[Name])+','+RTRIM(
fr.PostalCode) AS TheaterAddress,
movie.MovieName,movie.RunTime,movie.MetacriticReview,mr.MovieRatingName,ts.Timeslot into
tempTable from Theatre.Theatre theatre
Join Theatre.Theatre_Movie_Showtime tms on  theatre.TheatreID=tms.TheatreID
Join Movies.Movie movie on movie.MovieID=tms.MovieID
Join Theatre.Showtimes ts on ts.ShowtimesID=tms.ShowtimesID
Join Movies.Movie_Rating mr on mr.MovieRatingID=movie.MovieRatingID
Join Features.[Address] fr on fr.AddressID=theatre.AddressID
Join Features.State_Cnfg fs on fs.StateID=fr.StateID
Join Features.Country_Cnfg fc on fc.CountryID=fr.CountryID where fr.City=@userCity
order by theatre.TheatreID;


select distinct ST2.TheatreID,ST2.TheaterName,ST2.TheaterAddress
,ST2.MovieName,ST2.RunTime,ST2.MetacriticReview,ST2.MovieRatingName,
    substring(
        (
            Select ','+ST1.[Timeslot]
```

```
            From tempTable ST1
            Where ST1.MovieName = ST2.MovieName
            ORDER BY ST2.TheatreID,ST2.TheaterName,ST2.TheaterAddress
,ST2.MovieName,ST2.RunTime,ST2.MetacriticReview,ST2.MovieRatingName
            For XML PATH ('')
        ), 2, 100) [ShowTimes]

From tempTable ST2

drop table tempTable

END

EXEC usp_Currently_Running_User_Location 11
```

	TheatreID	TheaterName	TheaterAddress	MovieName	RunTime	MetacriticReview	MovieRatingName	ShowTimes
1	12	Cineplex Odeon Varsity & VIP Cinemas	55 Bloor Street West,Toronto,Ontario,Canada,M4W1A5	Almost Christmas(2016)	111	55	PG-13	5:10 pm ,12:10 pm
2	12	Cineplex Odeon Varsity & VIP Cinemas	55 Bloor Street West,Toronto,Ontario,Canada,M4W1A5	Office Christmas Party (2016)	105	0	NC-17	10:10 am ,7:10 pm
3	13	Famous Players Canada Square Cinemas	2190 Yonge Street,Toronto,Ontario,Canada,M4S2C6	Arrival(2016)	116	81	PG-13	7:10 pm
4	13	Famous Players Canada Square Cinemas	2190 Yonge Street,Toronto,Ontario,Canada,M4S2C6	Office Christmas Party (2016)	105	0	NC-17	10:10 am ,7:10 pm
5	20	Satyam Cinemas	55 Bloor Street West,Toronto,Ontario,Canada,M4W1A5	Almost Christmas(2016)	111	55	PG-13	5:10 pm ,12:10 pm
6	20	Satyam Cinemas	55 Bloor Street West,Toronto,Ontario,Canada,M4W1A5	Bad Santa 2 (2016)	92	38	R	7:10 pm ,4:10 pm
7	21	Regal Cinemas	2190 Yonge Street,Toronto,Ontario,Canada,M4S2C6	Bad Santa 2 (2016)	92	38	R	7:10 pm ,4:10 pm

16) Movies releasing this week

```
CREATE proc usp_Releasing_This_Week
AS
BEGIN


SELECT Movie.MovieID,MovieName,RunTime,GCnfg.[Name],StoryLine,ReleaseDate into tempTable
from Movies.Movie Movie
Join Details.Movie_Genre Genre ON Movie.MovieID=Genre.MovieID
Join Movies.Genre_Cnfg GCnfg on Genre.GenreID=GCnfg.GenreID  where ReleaseDate>GETDATE() AND
ReleaseDate<(DATEADD(DAY,7,GETDATE()));


select distinct ST2.MovieID,ST2.MovieName,ST2.Runtime ,ST2.Storyline,ST2.ReleaseDate,
    substring(
        (
            Select ','+RTRIM(ST1.[Name])
            From tempTable ST1
            Where ST1.MovieName = ST2.MovieName
            ORDER BY ST2.MovieID,ST1.MovieName,ST2.Runtime ,ST2.Storyline,ST2.ReleaseDate
            For XML PATH ('')
        ), 2, 100) [Genre]
              into tempTable2
From tempTable ST2



SELECT Distinct
Movie.MovieID,movie.MovieName,movie.RunTime,movie.StoryLine,tt2.Genre,CelebrityName into
tempTable3 from Movies.Movie Movie

JOIN  Celebrity.Movie_Celebrity mc ON Movie.MovieID=mc.MovieID
```

```
Join  Celebrity.Celebrity c on mc.CelebrityID=c.CelebrityID
Join tempTable2 tt2 on tt2.MovieName=Movie.MovieName  ;


Select distinct ST2.MovieID,ST2.MovieName,ST2.Runtime ,ST2.Storyline,ST2.Genre,
    substring(
        (
            Select ','+ST1.CelebrityName
            From tempTable3 ST1
            Where ST1.MovieName = ST2.MovieName
            ORDER BY ST2.MovieID,ST1.MovieName,ST2.Runtime ,ST2.Storyline,ST2.Genre
            For XML PATH ('')
        ), 2, 100) [Stars]
From tempTable3 ST2


 drop table temptable;
 drop table temptable2;

 drop table temptable3;


END



EXEC usp_Releasing_This_Week
```

	MovieID	MovieName	Runtime	Storyline	Genre	Stars
1	8	Office Christmas Party (2016)	105	When the CEO (Jennifer Aniston) tries to close h...	Sci-Fi,Romance	Hans Zimmer,Jason Statham,Jennifer Lawrence,Michael Caine,Rooney Mara,Simon Cowell

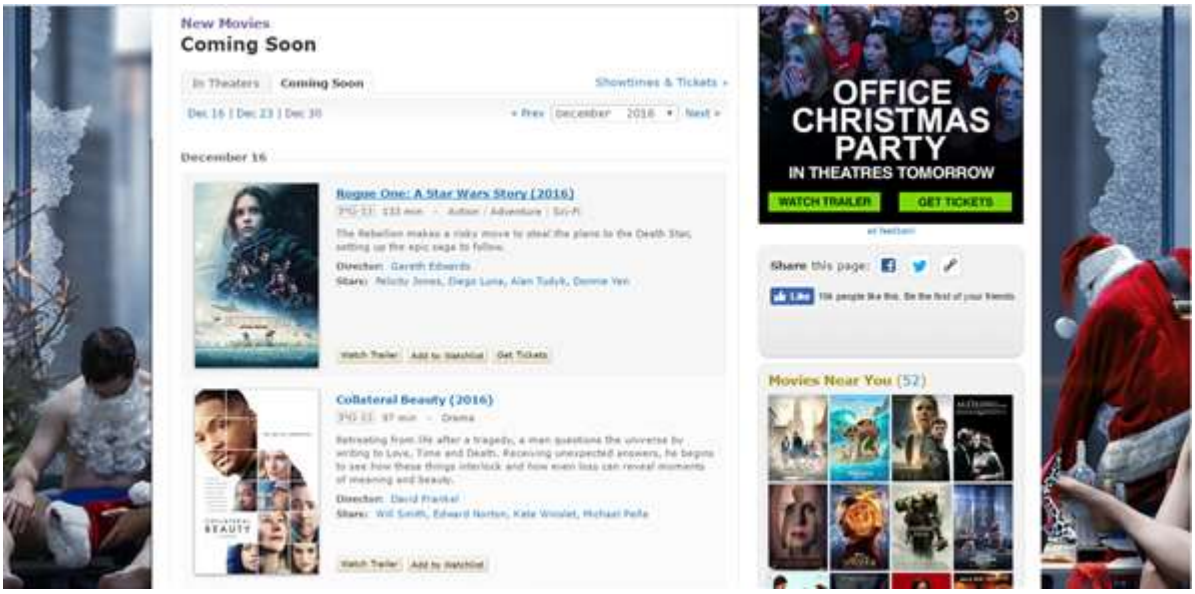
17) Movies releasing soon

DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

```
CREATE proc usp_Comming_Soon
AS
BEGIN


SELECT Movie.MovieID,MovieName,RunTime,GCnfg.[Name],StoryLine,ReleaseDate into tempTable
from Movies.Movie Movie
Join Details.Movie_Genre Genre ON Movie.MovieID=Genre.MovieID
Join Movies.Genre_Cnfg GCnfg on Genre.GenreID=GCnfg.GenreID  where
ReleaseDate>(DATEADD(DAY,7,GETDATE()));


select distinct ST2.MovieID,ST2.MovieName,ST2.Runtime ,ST2.Storyline,ST2.ReleaseDate,
    substring(
        (
            Select ','+RTRIM(ST1.[Name])
            From tempTable ST1
            Where ST1.MovieName = ST2.MovieName
            ORDER BY ST2.MovieID,ST1.MovieName,ST2.Runtime ,ST2.Storyline,ST2.ReleaseDate
            For XML PATH ('')
        ), 2, 100) [Genre]
              into tempTable2
From tempTable ST2



SELECT Distinct
Movie.MovieID,movie.MovieName,movie.RunTime,movie.StoryLine,tt2.Genre,CelebrityName into
tempTable3 from Movies.Movie Movie

JOIN  Celebrity.Movie_Celebrity mc ON Movie.MovieID=mc.MovieID
Join  Celebrity.Celebrity c on mc.CelebrityID=c.CelebrityID
Join tempTable2 tt2 on tt2.MovieName=Movie.MovieName  ;


Select distinct ST2.MovieID,ST2.MovieName,ST2.Runtime ,ST2.Storyline,ST2.Genre,
    substring(
        (
            Select ','+ST1.CelebrityName
            From tempTable3 ST1
            Where ST1.MovieName = ST2.MovieName
            ORDER BY ST2.MovieID,ST1.MovieName,ST2.Runtime ,ST2.Storyline,ST2.Genre
            For XML PATH ('')
        ), 2, 100) [Stars]
From tempTable3 ST2


 drop table temptable;
 drop table temptable2;

 drop table temptable3;


END



EXEC usp_Comming_Soon
```

DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

	MovieID	MovieName	Runtime	Storyline	Genre	Stars
1	7	La La Land (2016)	128	A jazz pianist falls for an aspiring actress in Los ...	Comedy	Amanda Bo,J K Simmons,Jason Statham,Jennifer Lawrence,Kate Winslet,Quentin Tarantino
2	13	Friend Request (2016)	92	When a college student unfriends a mysterious ...	Romance,Mystery	Joseph Leonardo Di Caprio,Martin Scorsese,Matthew McConaughey,Rachel McAdams,Scarlett Johansson,The



18) Movies in theatres

```
CREATE PROC usp_Currently_Running
AS
BEGIN

select theatre.TheatreID,theatre.[Name] TheaterName,
RTRIM(fr.AddressLine)+','+RTRIM(fr.City)+','+RTRIM(fs.[Name])+','+RTRIM(fc.[Name])+','+RTRIM(
fr.PostalCode) AS TheaterAddress,
movie.MovieName,movie.RunTime,movie.MetacriticReview,mr.MovieRatingName,ts.Timeslot into
tempTable from Theatre.Theatre theatre
Join Theatre.Theatre_Movie_Showtime tms on  theatre.TheatreID=tms.TheatreID
Join Movies.Movie movie on movie.MovieID=tms.MovieID
Join Theatre.Showtimes ts on ts.ShowtimesID=tms.ShowtimesID
Join Movies.Movie_Rating mr on mr.MovieRatingID=movie.MovieRatingID
Join Features.[Address] fr on fr.AddressID=theatre.AddressID
Join Features.State_Cnfg fs on fs.StateID=fr.StateID
Join Features.Country_Cnfg fc on fc.CountryID=fr.CountryID
order by theatre.TheatreID;


select distinct ST2.TheatreID,ST2.TheaterName,ST2.TheaterAddress
,ST2.MovieName,ST2.RunTime,ST2.MetacriticReview,ST2.MovieRatingName,
    substring(
        (
            Select ','+RTRIM(ST1.[Timeslot])
            From tempTable ST1
            Where ST1.MovieName = ST2.MovieName
            ORDER BY ST2.TheatreID,ST2.TheaterName,ST2.TheaterAddress
,ST2.MovieName,ST2.RunTime,ST2.MetacriticReview,ST2.MovieRatingName
            For XML PATH ('')
```

```
        ), 2, 100) [ShowTimes]

From tempTable ST2

drop table tempTable

END


EXEC usp_Currently_Running
```

	TheaterID	TheaterName	TheaterAddress	MovieName	RunTime	MetacriticReview	MovieRatingName	ShowTimes
1	1	Regal Westborough Stadium 12	231 Turnpike Road,Westborough,Massachusetts,Unit...	Allied(2016)	124	60	R	10:15 am,1:25 pm,4:10 pm,6:10 pr
2	2	Blackstone Valley 14: Cinema de Lux	70 Worcester/Providence Turnpike,Millbury,Massachu...	Allied(2016)	124	60	R	10:15 am,1:25 pm,4:10 pm,6:10 pr
3	2	Blackstone Valley 14: Cinema de Lux	70 Worcester/Providence Turnpike,Millbury,Massachu...	Almost Christmas(2016)	111	55	PG-13	10:15 am,4:10 pm,6:10 pm,8:10 pr
4	3	Showcase Cinemas Worcester North	135 Brooks Street,Worcester,Massachusetts,United St...	Arrival(2016)	116	81	PG-13	7:10 pm,7:10 pm,8:10 pm,8:10 pm
5	3	Showcase Cinemas Worcester North	135 Brooks Street,Worcester,Massachusetts,United St...	Me Before You (2016)	110	51	PG-13	8:10 pm
6	4	Cinemagic in Stubridge MA	Center at Hobbs Brook,Studridge,Massachusetts,Unit...	Allied(2016)	124	60	R	10:15 am,1:25 pm,4:10 pm,6:10 pr
7	4	Cinemagic in Stubridge MA	Center at Hobbs Brook,Studridge,Massachusetts,Unit...	Captain Fantastic (2016)	118	72	R	10:15 am
8	5	AMC Framingham 16 with Dine-in Theatres	22 Flute Pass,Framingham,Massachusetts,United Stat...	Allied(2016)	124	60	R	10:15 am,1:25 pm,4:10 pm,6:10 pr
9	6	Cinema World Lincoln Mall 16	622 George Washington Highway,Lincoln,Rhode Islan...	Fantastic Beasts and Where to Find Them (2016)	133	66	PG-13	4:10 pm,4:10 pm,4:10 pm,7:10 pm
10	8	Route One Cinema Pub	652 East Washington St.,North Attleboro,Massachuset...	Arrival(2016)	116	81	PG-13	7:10 pm,7:10 pm,8:10 pm,8:10 pm

19) Listing top 250 movies based on Ranking, IMDB rating, Release Date, Most User Reviews

```
CREATE Proc usp_Top_250_Movies(@value int)
AS
BEGIN

if @value=1
BEGIN
Select TOP 250
Dense_RANK() OVER (ORDER BY Cast(avg(Rating) AS int) desc ) AS Rank  ,
MovieName,Cast(avg(Rating) AS int) As IMDBRating
from Movies.Movie movie
Join [User].User_Review ur on  ur.MovieID=Movie.MovieID
group by MovieName
order by IMDBRating Desc;
END
```

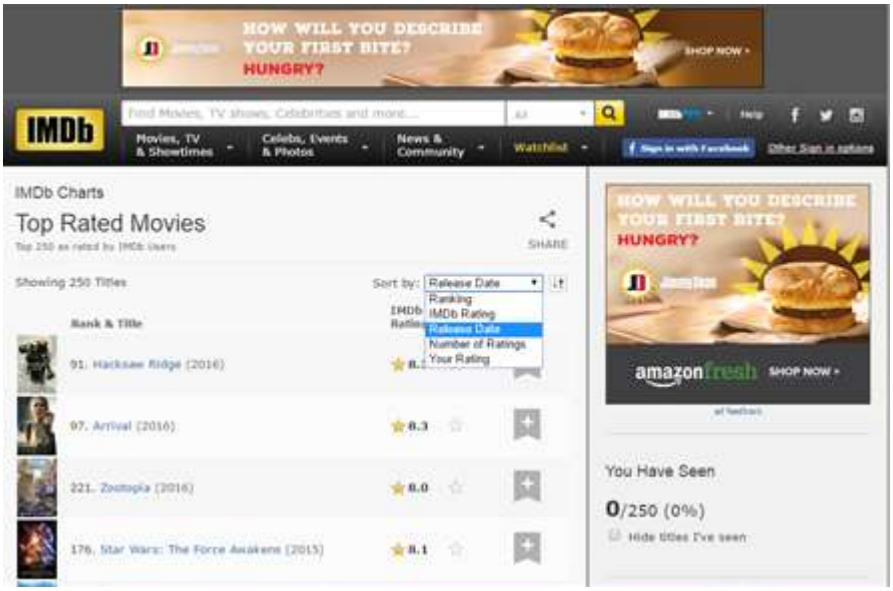
DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

```
if @value=2
BEGIN
Select TOP 250
Dense_RANK() OVER (ORDER BY Cast(avg(Rating) AS int) desc ) AS Rank  ,
MovieName,Cast(avg(Rating) AS int) As IMDBRating
from Movies.Movie movie
Join [User].User_Review ur on  ur.MovieID=Movie.MovieID
group by MovieName
order by IMDBRating Desc;
END

else if @value=3
BEGIN
Select TOP 250
Dense_RANK() OVER (ORDER BY Cast(avg(Rating) AS int)desc ) AS Rank  ,
MovieName,ReleaseDate,Cast(avg(Rating) AS int) As IMDBRating
from Movies.Movie movie
Join [User].User_Review ur on  ur.MovieID=Movie.MovieID
group by MovieName,ReleaseDate
order by ReleaseDate Desc;
END

else if @value=4
BEGIN
Select TOP 250
Dense_RANK() OVER (ORDER BY Cast(avg(Rating) AS int)desc ) AS Rank ,
 MovieName,Cast(avg(Rating) AS int) As IMDBRating,count(*) TotalUserReview from  Movies.Movie
movie
Join [User].User_Review  ur on movie.MovieID=ur.MovieID
Group by MovieName
order by TotalUserReview DESC;
END
END
```

```
EXEC usp_Top_250_Movies 2
```

	Rank	MovieName	IMDBRating
1	1	No Country for Old Men (2007)	9
2	2	Notre musique (2004)	8
3	2	Pearl Harbor (2001)	8
4	2	Rush Hour 2 (2001)	8
5	2	King Kong (2005)	8
6	2	Love Actually (2003)	8
7	2	Me Before You (2016)	8
8	2	Minority Report (2002)	8
9	2	The Fundamentals of Caring (2016)	8
10	2	The Notebook (2004)	8
11	2	Toy Story 3 (2010)	8

DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan



20) Top amazon videos by views

```
SELECT
CASE
       WHEN EntityTypeID = 1 THEN m.MovieName
       ELSE ts.Name
END Entity, URL, av.Views
FROM Details.Amazon_Videos av
LEFT JOIN Movies.Movie m ON av.EntityID = m.MovieID
LEFT JOIN TVShows.TV_Shows ts ON av.EntityID = ts.ShowID
ORDER BY av.Views DESC
```

	Entity	URL	Views
1	Big Bang Theory	http://www.imdb.com	87542
2	24	http://www.imdb.com	87542
3	Supergirl	http://www.imdb.com	87542
4	Casino Royale (2006)	http://www.imdb.com	84557
5	Supernatural	http://www.imdb.com	61455
6	Big Bang Theory	http://www.imdb.com	54512
7	A Christmas Tale (2008)	http://www.imdb.com	45478
8	Deadpool (2016)	http://www.imdb.com	45221
9	White Collar	http://www.imdb.com	25448
10	Avatar (2009)	http://www.imdb.com	24457
11	Sex and the City	http://www.imdb.com	21565

21) Top amazon videos by IMDB rating

```
SELECT MovieName, URL, av.Views
FROM Details.Amazon_Videos av
JOIN Movies.Movie m ON av.EntityID = m.MovieID
ORDER BY m.IMDBRating
```

DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

	MovieName	URL	Views	IMDBRating
1	Me Before You (2016)	http://www.imdb.com	87542	8.00
2	Me Before You (2016)	http://www.imdb.com	8542	8.00
3	Deadpool (2016)	http://www.imdb.com	12455	8.00
4	23 (1998)	http://www.imdb.com	9457	8.00
5	Deadpool (2016)	http://www.imdb.com	45221	8.00
6	Casino Royale (2006)	http://www.imdb.com	84557	8.00
7	Allied(2016)	http://www.imdb.com	12457	7.70
8	Allied(2016)	http://www.imdb.com	12457	7.70
9	Inglourious Basterds (2009)	http://www.imdb.com	1267	7.67
10	Avatar (2009)	http://www.imdb.com	24457	7.67
11	Harry Benson: Shoot First (2016)	http://www.imdb.com	6757	7.67

22) List all DVDs

```
SELECT
CASE
       WHEN EntityTypeID = 1 THEN m.MovieName
       ELSE ts.Name
END DVD
FROM Details.Blue_Ray_DVD brd
LEFT JOIN Movies.Movie m ON brd.EntityID = m.MovieID
LEFT JOIN TVShows.TV_Shows ts ON brd.EntityID = ts.ShowID
```

	DVD
1	Allied(2016)
2	Almost Christmas(2016)
3	Arrival(2016)
4	Bad Santa 2 (2016)
5	Doctor Strange(2016)
6	Fantastic Beasts and Where to Find Them (2016)
7	Friend Request (2016)
8	Harry Benson: Shoot First (2016)
9	Deadpool (2016)
10	Shivaay (2016)
11	The Fundamentals of Caring (2016)

23) List all videos

```
SELECT
CASE
       WHEN v.EntityTypeID = 1 THEN m.MovieName
       WHEN v.EntityTypeID = 2 THEN c.CelebrityName
       ELSE ts.Name
END Video, v.URL, vt.VideoName
FROM Details.Videos v
JOIN Details.Video_Type vt ON vt.VideoTypeID = v.VideoTypeID
LEFT JOIN Movies.Movie m ON v.EntityID = m.MovieID
LEFT JOIN TVShows.TV_Shows ts ON v.EntityID = ts.ShowID
LEFT JOIN Celebrity.Celebrity c ON  v.EntityID = c.CelebrityID
```

DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

	Video	URL	VideoType
1	Allied(2016)	http://www.imdb.com/	Bloopers
2	Allied(2016)	http://www.imdb.com/	Trailers
3	Knocked Up (2007)	http://www.imdb.com/	Bloopers
4	23 (1998)	http://www.imdb.com/	Bloopers
5	Harry Benson: Shoot First (2016)	http://www.imdb.com/	Bloopers
6	The Blacklist	http://www.imdb.com/	Bloopers
7	The Vampire Diaries	http://www.imdb.com/	Bloopers
8	Spartacus: War of the Damned	http://www.imdb.com/	Bloopers
9	Sex and the City	http://www.imdb.com/	Bloopers
10	Big Bang Theory	http://www.imdb.com/	Bloopers
11	Allied(2016)	http://www.imdb.com/	Video Song

24) Latest movie trailers

```
SELECT m.MovieName Video, v.URL
FROM Details.Videos v
JOIN Details.Video_Type vt ON vt.VideoTypeID = v.VideoTypeID
JOIN Movies.Movie m ON v.EntityID = m.MovieID
WHERE v.VideoTypeID = 12
ORDER BY m.ReleaseDate DESC
```

	Video	URL
1	Allied(2016)	http://www.imdb.com/
2	Captain Fantastic (2016)	http://www.imdb.com/
3	Sing Street (2016)	http://www.imdb.com/
4	Inception(2010)	http://www.imdb.com/

25) Details of all soundtracks

```
SELECT s.Name,
CASE
       WHEN  s.EntityTypeID = 1 THEN m.MovieName
       ELSE ts.Name
END Entity, s.Playtime, c.CelebrityName MusicDirector, c2.CelebrityName Singer
FROM Details.Soundtracks s
JOIN Celebrity.Celebrity c ON c.CelebrityID = s.MusicDirectorID
JOIN Celebrity.Celebrity c2 ON c2.CelebrityID = s.SingerID
LEFT JOIN Movies.Movie m ON s.EntityID = m.MovieID
LEFT JOIN TVShows.TV_Shows ts ON s.EntityID = ts.ShowID
```

	Name	Entity	Playtime	MusicDirector	Singer
1	impression	Allied(2016)	8	Adam Levine	A. R. Rahman
2	Never You Find Me	Almost Christmas(2016)	9	Adam Levine	Hans Zimmer
3	Exlibris	Arrival(2016)	10	Adam Levine	Steven Price
4	Kosta T Informality	Bad Santa 2 (2016)	5	Selena Gomez	Steven Price
5	Spaces of L	Doctor Strange(2016)	5	Selena Gomez	Hans Zimmer
6	Fresh for N	Fantastic Beasts and Where to Find Them (2016)	7	Selena Gomez	A. R. Rahman
7	Podington Bear	La La Land (2016)	7	The Weekend	A. R. Rahman
8	Thread Clouds	Office Christmas Party (2016)	8	The Weekend	Steven Price
9	Thread of fire	Friend Request (2016)	9	The Weekend	A. R. Rahman
10	Tonight I love u	Harry Benson: Shoot First (2016)	8	The Weekend	Hans Zimmer
11	Blank Space	Deadpool (2016)	6	The Weekend	Steven Price

26) View for all image details

```
CREATE VIEW View_ImageStorage
AS
	SELECT
	CASE
		WHEN i.EntityTypeID = 1 THEN m.MovieName
		WHEN i.EntityTypeID = 2 THEN c.CelebrityName
		ELSE ts.Name
	END Entity, i.ImageBlob
	FROM Details.Images i
	LEFT JOIN Movies.Movie m ON m.MovieID = i.EntityID
	LEFT JOIN Celebrity.Celebrity c ON c.CelebrityID = i.EntityID
	LEFT JOIN TVShows.TV_Shows ts ON ts.ShowID = i.EntityID
GO

SELECT * FROM dbo.View_ImageStorage vis
```

	Entity	ImageBlob
1	The Dark Knight (2008)	0xFFD8FFE000104A4649460001010000010001000FFDB004...
2	A. R. Rahman	0xFFD8FFE000104A4649460001010000010001000FFDB004...
3	Steven Price	0xFFD8FFE000104A4649460001010000010001000FFFE003...
4	Inception(2010)	0xFFD8FFE000104A464946000101010012C012C0000FFED24...
5	Top Gear	0xFFD8FFE000104A4649460001010100480048000FFDB004...
6	Friend Request (2016)	0xFFD8FFE000104A4649460001010000010001000FFDB004...
7	Adam Levine	0xFFD8FFE000104A4649460001010000010001000FFFE003...
8	Matthew McConaughey	0xFFD8FFE000104A4649460001010000010001000FFFE003...
9	Office Christmas Party (2016)	0xFFD8FFE000104A4649460001010100480048000FFDB004...
10	Westworld	0xFFD8FFE000104A464946000101010012C012C0000FFED29...

27) Details of TV shows

```
CREATE VIEW View_TVShows_Genre_ProductionCompany
AS
	SELECT ts.Name TV_Show, ts.Description, gc.Name Genre, lc.Name Language, ts.Runtime,
ts.SoundMix, pc.ProductionCompanyName
	FROM TVShows.TV_Shows ts
	JOIN Details.TVShow_Genre tg ON tg.ShowID = ts.ShowID
	JOIN Movies.Genre_Cnfg gc ON gc.GenreID = tg.GenreID
	JOIN Features.Production_Company pc ON pc.ProductionCompanyID = ts.ProductionCompanyID
	JOIN Details.Language_Cnfg lc ON lc.Language = ts.LanguageID
```

DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

```
GO

SELECT * FROM dbo.View_TVShows_Genre_ProductionCompany vtgpc
```

	TV_Show	Description	Genre	Language	Runtime	SoundMix	ProductionCompanyName
1	Westworld	A series inspired by the 1973 film of the same title	Horror	English	60	Dolby Digital	HBO
2	Westworld	A series inspired by the 1973 film of the same title	History	English	60	Dolby Digital	HBO
3	Big Bang Theory	A woman moves into an apartment across the hall fr...	Mystery	English	22	Stereo	HBO
4	Two and a Half Men	A hedonistic jingle writers free-wheeling life comes t...	Adventure	English	22	Dolby	Helen Control
5	Two and a Half Men	A hedonistic jingle writers free-wheeling life comes t...	Crime	English	22	Dolby	Helen Control
6	Two and a Half Men	A hedonistic jingle writers free-wheeling life comes t...	Mystery	English	22	Dolby	Helen Control
7	Supernatural	Two brothers follow their fathers footsteps as "hunt...	Action	English	44	Dolby Digital	LeeFilm
8	The Vampire Diaries	A teenage girl is torn between two vampire brothers.	Mystery	English	43	Dolby Digital	Legendary
9	Teen Wolf	A somewhat awkward teen is attacked by a werew...	Comedy	English	41	Dolby Digital	MGM Films
10	Teen Wolf	A somewhat awkward teen is attacked by a werew...	Mystery	English	41	Dolby Digital	MGM Films
11	White Collar	A white collar criminal agrees to help the FBI catch ...	Comedy	English	40	Dolby Digital	Star Movies

28) Trigger for calculating Total movie revenue based on weekly collections

```
CREATE TRIGGER trgAfterWeeklyGrossInsert ON Movies.Week_Gross
FOR INSERT
AS

        DECLARE @gross money
        DECLARE @ID int
        DECLARE @TotalGross money

        SELECT @ID = i.MovieID FROM INSERTED i
        SELECT @gross = i.Collection FROM INSERTED i

        SELECT @TotalGross = m.Gross_Revenue FROM Movies.Movie m WHERE m.MovieID = (SELECT
wg.MovieID FROM Movies.Week_Gross wg WHERE wg.WeekGrossID = @ID)

        UPDATE Movies.Movie SET Movies.Movie.Gross_Revenue = @TotalGross+@gross WHERE
Movies.Movie.MovieID = (SELECT wg.MovieID FROM Movies.Week_Gross wg WHERE wg.WeekGrossID =
@ID)

GO


INSERT INTO Movies.Week_Gross
(
    --WeekGrossID - this column value is auto-generated
    WeekNumber,
    Collection,
    MovieID
)
VALUES
(
    -- WeekGrossID - int
    5, -- WeekNumber - int
    20000, -- Collection - money
    1 -- MovieID - int
)
```

Before inserting data

	MovieID	MovieName	Gross_Revenue
1	1	Allied(2016)	260000.00

After inserting data

	MovieID	MovieName	Gross_Revenue
1	1	Allied(2016)	280000.00

29) Trigger to calculate IMDB on insert of user review

```
CREATE TRIGGER trgAfterUserReviewInsert ON [User].User_Review
FOR INSERT
AS

	DECLARE @ID int
	SELECT @ID = i.MovieID FROM INSERTED i
	EXEC uspCalculateTMDBRatingByMovie @ID

GO


INSERT INTO [User].User_Review
(
    --UserReviewID - this column value is auto-generated
    UserID,
    Review,
    ReviewedDate,
    MovieID,
    Rating
)
VALUES
(
    -- UserReviewID - int
    9, -- UserID - int
    'Worth the money spent', -- Review - char
    getdate(), -- ReviewedDate - datetime
    6, -- MovieID - int
    8.3 -- Rating - decimal
)
```

Before inserting data

DeivaKumaran Dhanasegaran
Aswathnarayan
Pranesh Ramakrishnan

	MovieID	MovieName	IMDBRating
1	6	Fantastic Beasts and Where to Find Them (2016)	6.00

After inserting data

	MovieID	MovieName	IMDBRating
1	6	Fantastic Beasts and Where to Find Them (2016)	7.15

30) INDEXES

```
CREATE INDEX idx_MovieName
ON Movies.Movie (MovieName);

CREATE INDEX idx_CelebrityName
ON Celebrity.Celebrity (CelebrityName);

CREATE INDEX idx_TVShowName
ON TVShows.TV_Shows (Name);

CREATE INDEX idx_NewsEntityID
ON Details.News (EntityID);

CREATE INDEX idx_RecentlyViewedEntityID
ON Details.Recently_Viewed (EntityID);

CREATE INDEX idx_TriviasEntityID
ON Details.Trivias (EntityID);

CREATE INDEX idx_SoundtracksEntityID
ON Details.Soundtracks (EntityID);

CREATE INDEX idx_VideosEntityID
ON Details.Videos (EntityID);

CREATE INDEX idx_ImagesEntityID
ON Details.Images (EntityID);

CREATE INDEX idx_AmazonVideosEntityID
ON Details.Amazon_Videos (EntityID);
```