Group 3:

Tejesh Rathi

Palak Agrawal

Sonali Chaudhari

Abstract

All the queries with its resultset

IMDB

Internet Movie Database

-----------------------------------------------------------------------------------------

/\*Databases : SQL Server

Author : Group 3

Creation Date : 12-06-2016

Query Number : 1

Query : Fetch Actor and Actress for each Movie not 'IN-THEATRE', also print Director for them AND order them by MOVIE alphabetically\*/

-----------------------------------------------------------------------------------------

SELECT \* FROM (SELECT CFM1.MOVIENAME AS MOVIE,

CFM1.CELEBRITYNAME AS DIRECTOR,

CFM2.CELEBRITYNAME AS ACTORORACTRESS,

SR.Status

FROM (SELECT DISTINCT(CELEBRITYNAME), MOVIENAME, MOVIEID FROM MOVIES.v\_CASTFORMOVIE WHERE TYPEID = 4 ) CFM1

LEFT JOIN (SELECT DISTINCT(CELEBRITYNAME), MOVIENAME, MOVIEID FROM MOVIES.v\_CASTFORMOVIE WHERE TYPEID IN (1,2)) CFM2

ON CFM2.MOVIEID = CFM1.MOVIEID

LEFT JOIN MOVIES.Movie M

ON M.MovieID = CFM2.MOVIEID

LEFT JOIN MOVIES.MovieTVStatusRef SR

ON SR.StatusID = M.StatusID

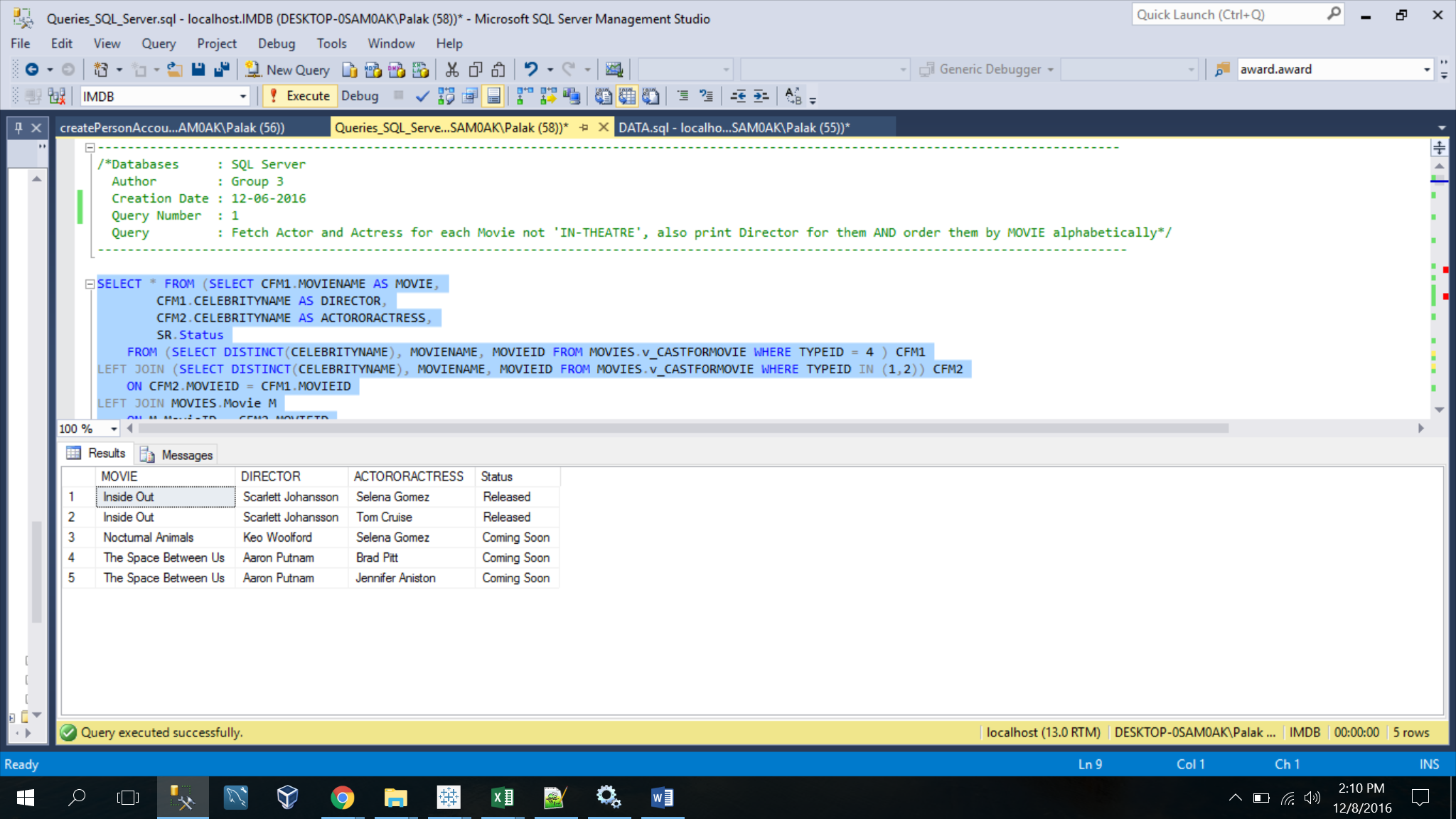
WHERE SR.Status NOT IN ('In-Theatre')

GROUP BY ROLLUP (CFM1.MOVIENAME,CFM1.CELEBRITYNAME,CFM2.CELEBRITYNAME, SR.Status)) CFM

WHERE CFM.ACTORORACTRESS IS NOT NULL

AND CFM.Status IS NOT NULL

ORDER BY CFM.MOVIE;



-----------------------------------------------------------------------------------------

/\*Databases : SQL Server

Author : Group 3

Creation Date : 12-06-2016

Query Number : 2

Query : Show the total collection till today for Movie bookings as per the Showtype for each Day and each Theatre\*/

-----------------------------------------------------------------------------------------

SELECT FORMAT(CAST(SHOWDATE AS DATE),'dd-MMM-yyyy') AS SHOWDATE,

THEATRENAME,

EARLYBIRDS,

BIRDSOFDAY,

BIRDSOFEVE,

BIRDSOFNIGHT,

LAZYBIRDS

FROM

(SELECT VS.SHOWNAME,

UB.ShowDate,

T.Name AS THEATRENAME,

UB.TicketCount

FROM MOVIES.v\_SHOWS VS

INNER JOIN MOVIES.MovieShowInTheatre MST

ON MST.SHOWTIMEID = VS.ShowTimeID

INNER JOIN MOVIES.UserBooking UB

ON UB.ShowTimeID = MST.ShowTimeID

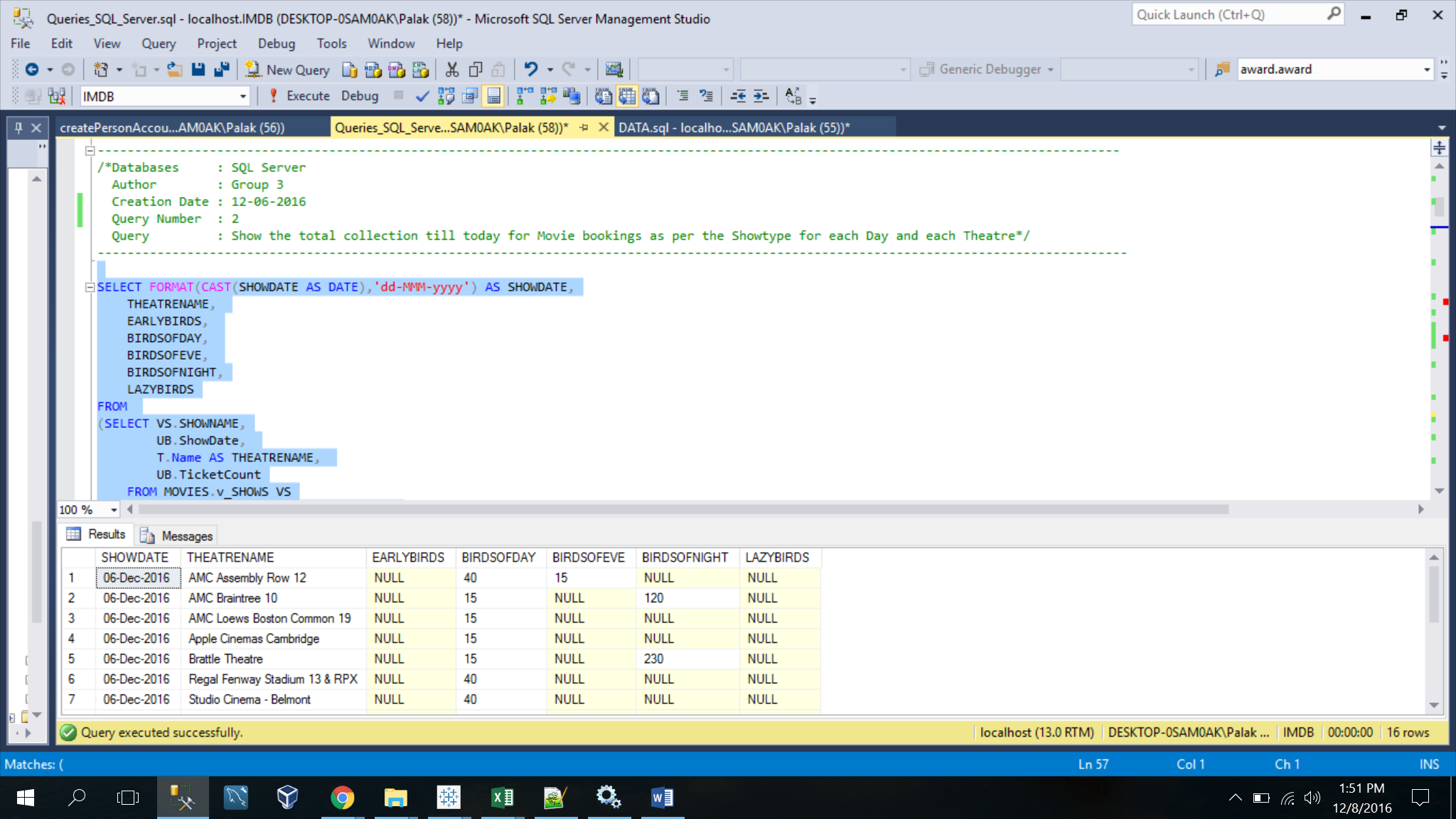
INNER JOIN MOVIES.Theatre T

ON T.TheatreID = MST.TheatreID

WHERE UB.ShowDate <= GETDATE() ) AS SHOWSCOLLECTION

PIVOT (SUM(TicketCount) FOR SHOWNAME IN([EARLYBIRDS],[BIRDSOFDAY],[BIRDSOFEVE],[BIRDSOFNIGHT],[LAZYBIRDS])

) AS pvt

ORDER BY SHOWDATE; 

-----------------------------------------------------------------------------------------

/\*Databases : SQL Server

Author : Group 3

Creation Date : 12-06-2016

Query Number : 3

Query : Retrieve highest rated movie by Genre and If any Genre has two movies with same rating order by movie name alphabetically\*/

-----------------------------------------------------------------------------------------

WITH HIGHESTRATEDGENRE\_CTE(GENRE,MOVIENAME,Rating)

AS

(

SELECT G.Genre,

M.Name AS MOVIENAME,

MAX(M.Rating)

FROM MOVIES.Movie M

LEFT JOIN MOVIES.MovieTVGenre MTG

ON MTG.MovieID = M.MovieID

LEFT JOIN MOVIES.Genre G

ON G.GenreID = MTG.GenreID

GROUP BY M.Name, Genre

)

SELECT HCTE.GENRE,

HCTE1.MOVIENAME,

HCTE.HIGHESTRATING

FROM (SELECT MAX(RATING) AS HIGHESTRATING, GENRE

FROM HIGHESTRATEDGENRE\_CTE

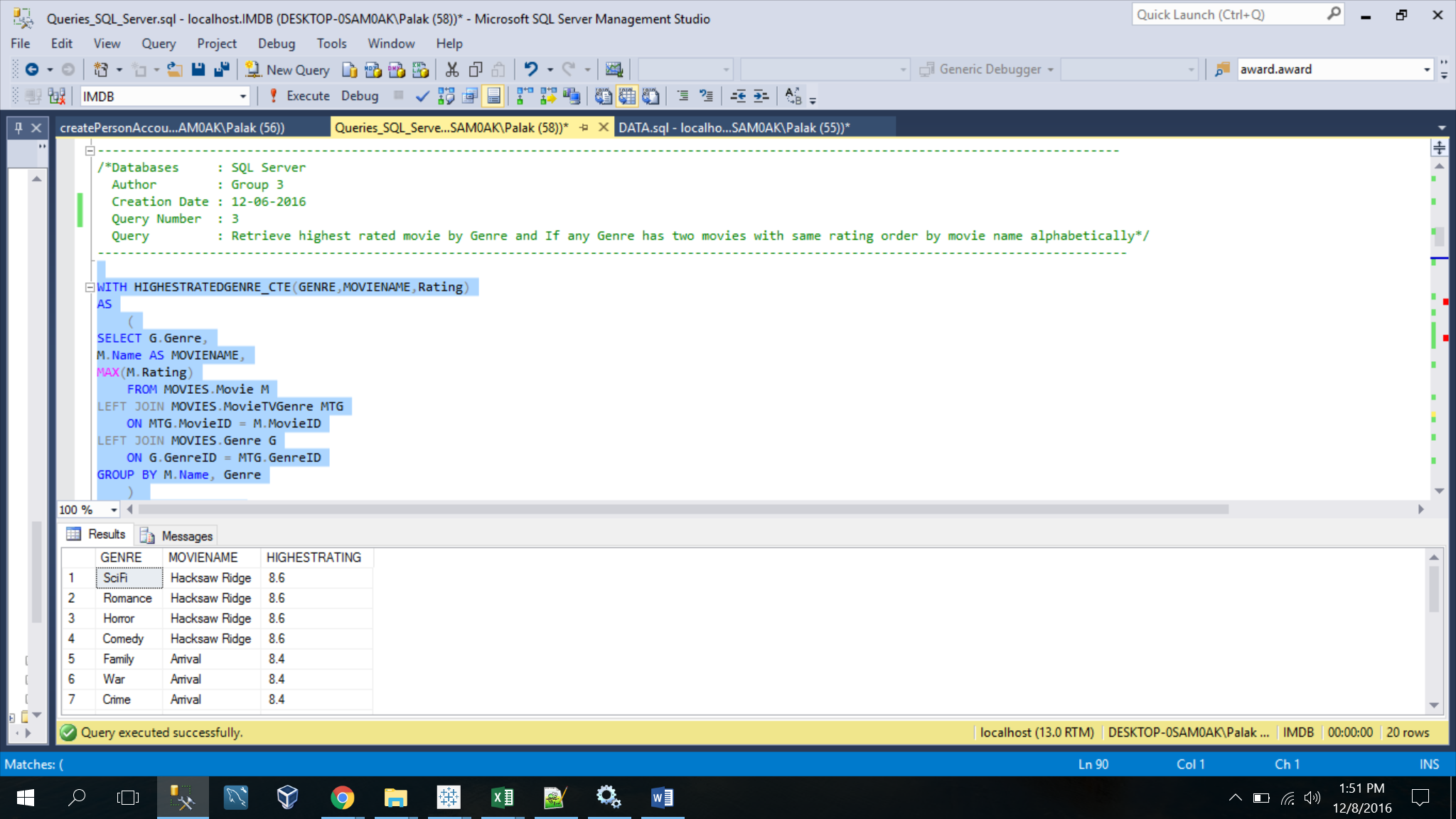
GROUP BY GENRE) HCTE

INNER JOIN (SELECT \* FROM HIGHESTRATEDGENRE\_CTE) HCTE1

ON HCTE.GENRE = HCTE1.GENRE

WHERE HCTE.HIGHESTRATING = HCTE1.RATING

AND HCTE.GENRE = HCTE1.GENRE

ORDER BY HCTE.HIGHESTRATING DESC,HCTE1.MOVIENAME; 

----------------------------------------------------------------------------------------

/\*Databases : SQL Server

Author : Group 3

Creation Date : 12-06-2016

Query Number : 4

Query : Retrieve Top 10 Movie which has collected hihest amount on BoxOffice and Print Number of weeks for which movie was running in descending order\*/

-----------------------------------------------------------------------------------------

SELECT TOP 10 C.MOVIENAME,

COUNT(DOC.COLLECTIONDAY) AS NumberOfWeeks,

C.TOTALBOXOFFICECOLLECTION AS TotalBoxOfficeCollectionInMi

FROM (SELECT DATEPART(DW, BOCD.CollectionDate) AS COLLECTIONDAY,

M.MovieID

FROM MOVIES.Movie M

LEFT JOIN MOVIES.BoxOfficeCollectionOfMovie BOCM

ON BOCM.MovieID = M.MovieID

LEFT JOIN MOVIES.BoxOfficeCollectionDate BOCD

ON BOCD.BoxOfficeCollectionID = BOCM.BoxOfficeCollectionID

GROUP BY BOCD.CollectionDate,M.MovieID) DOC

LEFT JOIN (SELECT M.MovieID,

M.Name AS MOVIENAME,

M.ReleaseDate,

DATEPART(DW, M.ReleaseDate) AS RELEASEDDAY,

SUM(BOCM.CollectionAmount) AS TOTALBOXOFFICECOLLECTION

FROM MOVIES.Movie M

LEFT JOIN MOVIES.BoxOfficeCollectionOfMovie BOCM

ON BOCM.MovieID = M.MovieID

LEFT JOIN MOVIES.BoxOfficeCollectionDate BOCD

ON BOCD.BoxOfficeCollectionID = BOCM.BoxOfficeCollectionID

GROUP BY M.MovieID,

M.Name,

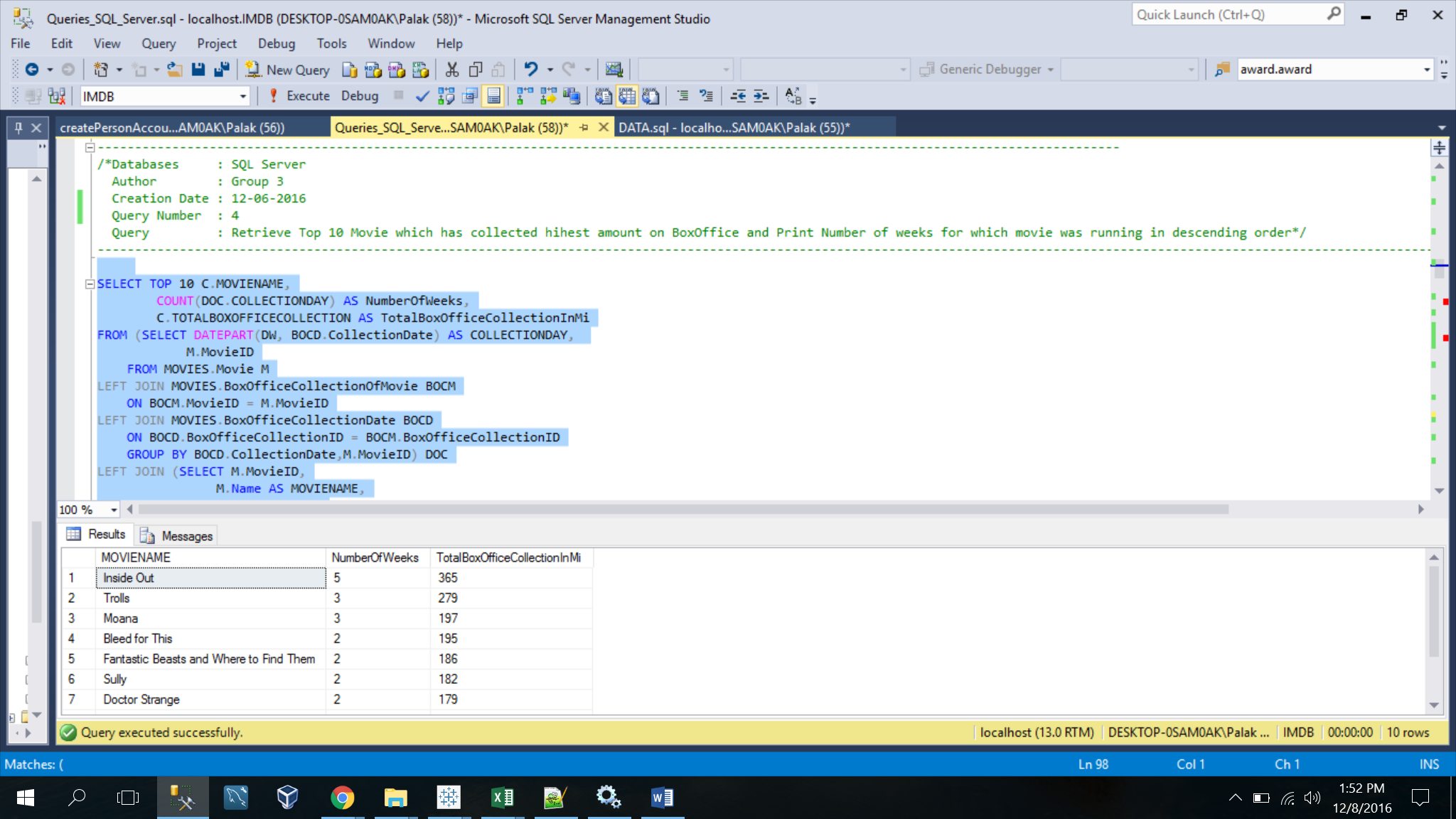
M.ReleaseDate) C

ON C.MovieID = DOC.MovieID

WHERE C.RELEASEDDAY = DOC.COLLECTIONDAY

GROUP BY C.TOTALBOXOFFICECOLLECTION,C.MOVIENAME

ORDER BY C.TOTALBOXOFFICECOLLECTION DESC, NUMBEROFWEEKS DESC;



-----------------------------------------------------------------------------------------

/\*Databases : SQL Server

Author : Group 3

Creation Date : 12-06-2016

Query Number : 5

Query : Highest voted poll option for every poll question\*/

-----------------------------------------------------------------------------------------

SELECT POP.PollQuestionID,

POP.Options,

POP.UserVotes,

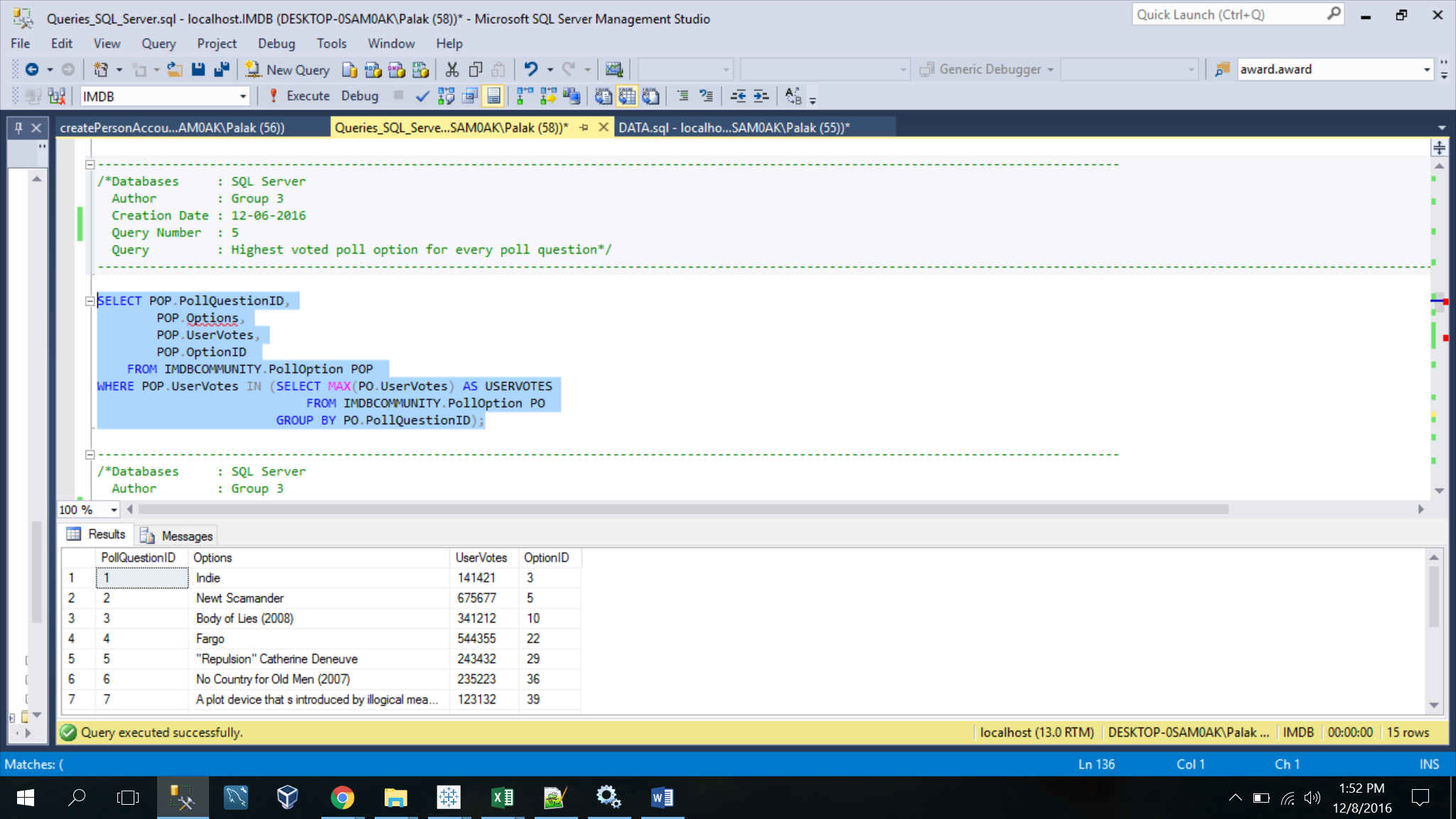
POP.OptionID

FROM IMDBCOMMUNITY.PollOption POP

WHERE POP.UserVotes IN (SELECT MAX(PO.UserVotes) AS USERVOTES

FROM IMDBCOMMUNITY.PollOption PO

GROUP BY PO.PollQuestionID);



-----------------------------------------------------------------------------------------

/\*Databases : SQL Server

Author : Group 3

Creation Date : 12-06-2016

Query Number : 6

Query : NUMBER OF REVIEWS FOR TOP 5 MOVIES BY RATING BY USERS AND CRITICS\*/

-----------------------------------------------------------------------------------------

SELECT M.MovieID,

M.Name AS MOVIE\_NAME,

COUNT(R.ReviewID) AS TOTAL\_REVIEWS

FROM MOVIES.Movie M

INNER JOIN REVIEW.Review R

ON R.MovieID = M.MovieID

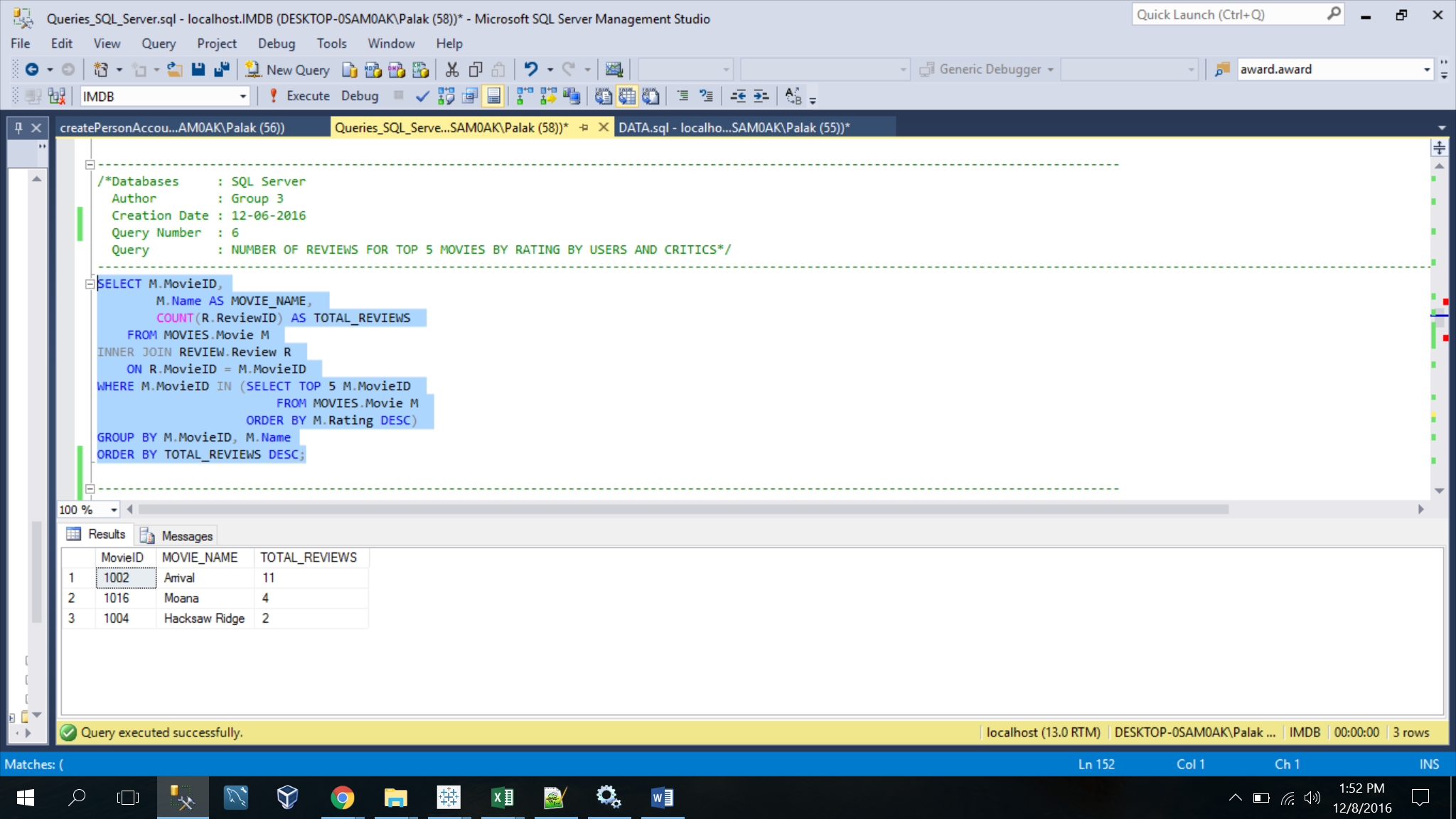
WHERE M.MovieID IN (SELECT TOP 5 M.MovieID

FROM MOVIES.Movie M

ORDER BY M.Rating DESC)

GROUP BY M.MovieID, M.Name

ORDER BY TOTAL\_REVIEWS DESC;



-----------------------------------------------------------------------------------------

/\*Databases : SQL Server

Author : Group 3

Creation Date : 12-06-2016

Query Number : 7

Query : Retrieve Budget, Box Office Collection of each Movie and decide whether it is in loss or profit\*/

-----------------------------------------------------------------------------------------

SELECT BOM.MovieID,

M.NAME, M.BUDGET/1000000 AS BUDGET,

SUM(BOM.CollectionAmount) AS TOTAL\_COLLECTION,

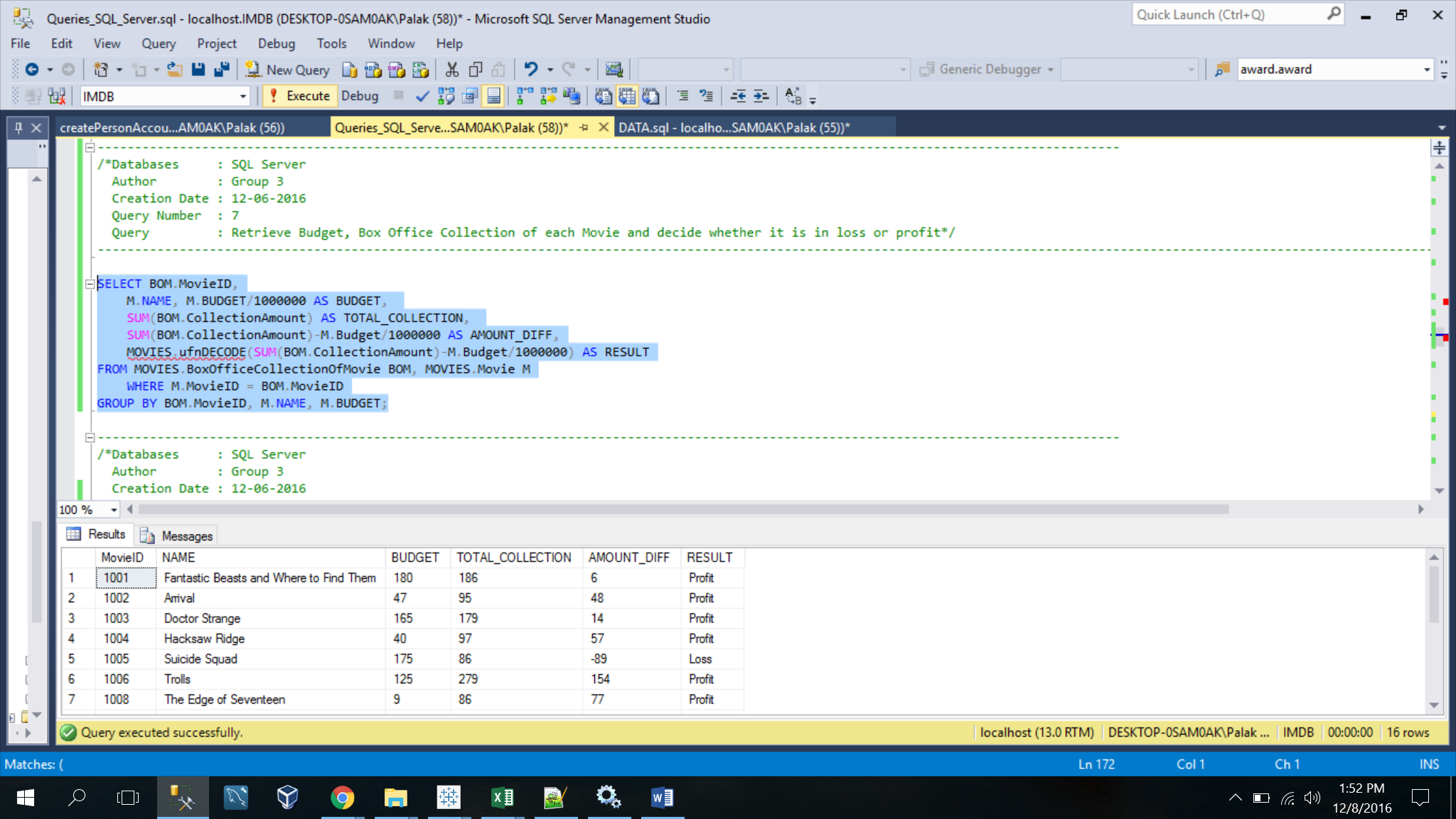
SUM(BOM.CollectionAmount)-M.Budget/1000000 AS AMOUNT\_DIFF,

MOVIES.ufnDECODE(SUM(BOM.CollectionAmount)-M.Budget/1000000) AS RESULT

FROM MOVIES.BoxOfficeCollectionOfMovie BOM, MOVIES.Movie M

WHERE M.MovieID = BOM.MovieID

GROUP BY BOM.MovieID, M.NAME, M.BUDGET;



-----------------------------------------------------------------------------------------

/\*Databases : SQL Server

Author : Group 3

Creation Date : 12-06-2016

Query Number : 8

Query : Retrieve number of seasons and episodes of TV Show\*/

-----------------------------------------------------------------------------------------

SELECT TT.Name,

TS.SeasonNumber,

COUNT(TE.EPISODENUMBER) TOTAL\_EPISODES

FROM TELEVISION.TVShows TT,

TELEVISION.TVSeasonShow TS,

TELEVISION.TVShowEpisode TE

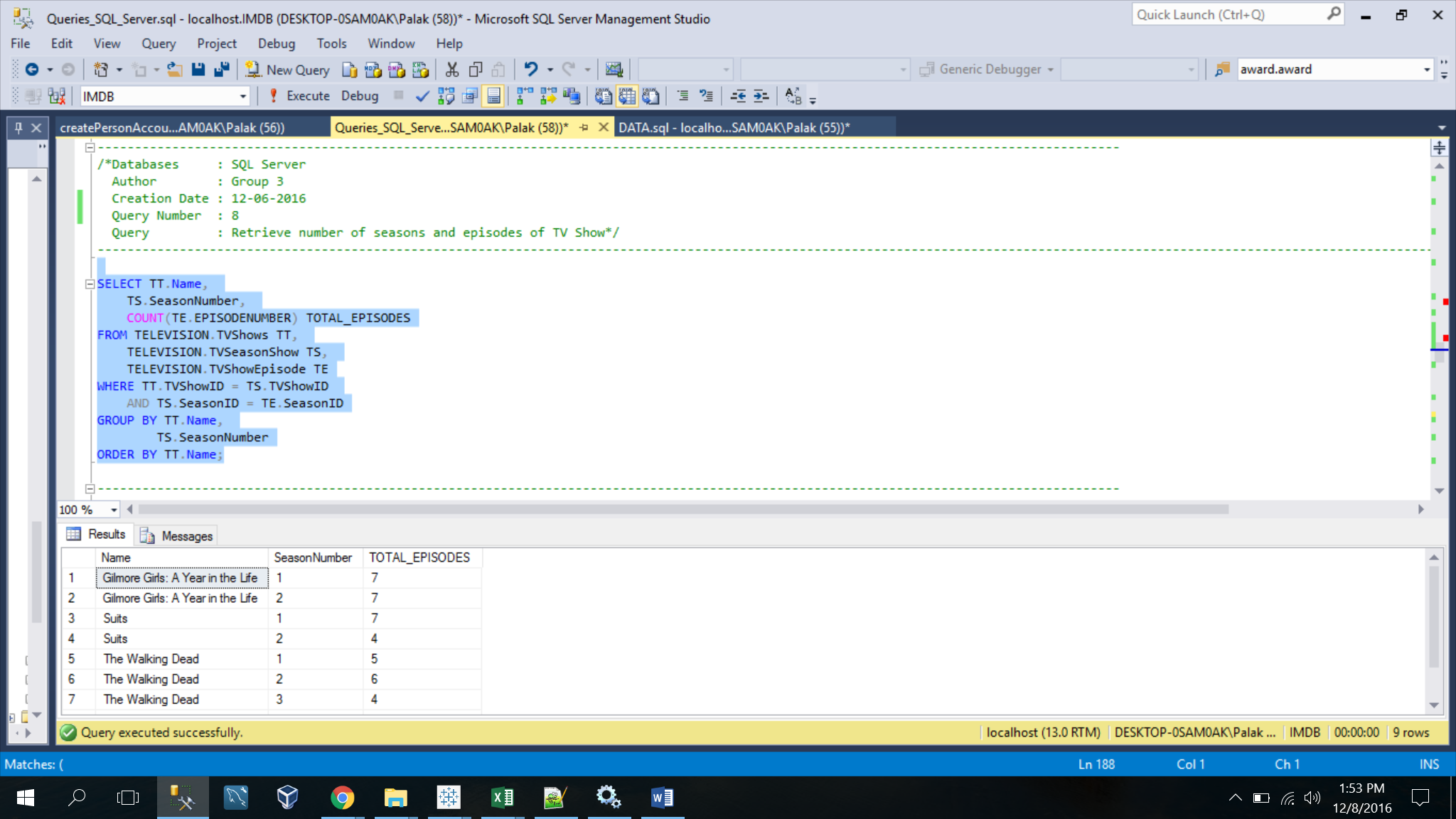
WHERE TT.TVShowID = TS.TVShowID

AND TS.SeasonID = TE.SeasonID

GROUP BY TT.Name,

TS.SeasonNumber

ORDER BY TT.Name;



-----------------------------------------------------------------------------------------

/\*Databases : SQL Server

Author : Group 3

Creation Date : 12-06-2016

Query Number : 9

Query : Retrieve the details of the TV Shows\*/

-----------------------------------------------------------------------------------------

SELECT SHOW.Name,

GENRE.Genre,

SEASON.SeasonNumber,

COUNT(EPISODE.EpisodeNumber) AS TOTAL\_EPISODES,

CHANNEL.Channel,

FORMAT(CAST(ST.ShowTime AS DATETIME), 'hh:mm tt') AS SHOWTIME

FROM TELEVISION.TVShows SHOW,

TELEVISION.TVSeasonShow SEASON,

TELEVISION.TVShowEpisode EPISODE,

TELEVISION.Channel CHANNEL,

TELEVISION.TVShowOnChannel SOC,

TELEVISION.TVShowTime ST,

MOVIES.MovieTVGenre MTG,

MOVIES.Genre GENRE

WHERE SHOW.TVShowID = SEASON.TVShowID

AND SOC.TVShowID = SHOW.TVShowID

AND SEASON.SeasonID = EPISODE.SeasonID

AND SOC.ShowTimeID = ST.ShowTimeID

AND CHANNEL.ChannelID = SOC.ChannelID

AND GENRE.GenreID = MTG.GenreID

AND SHOW.TVShowID = MTG.TVShowID

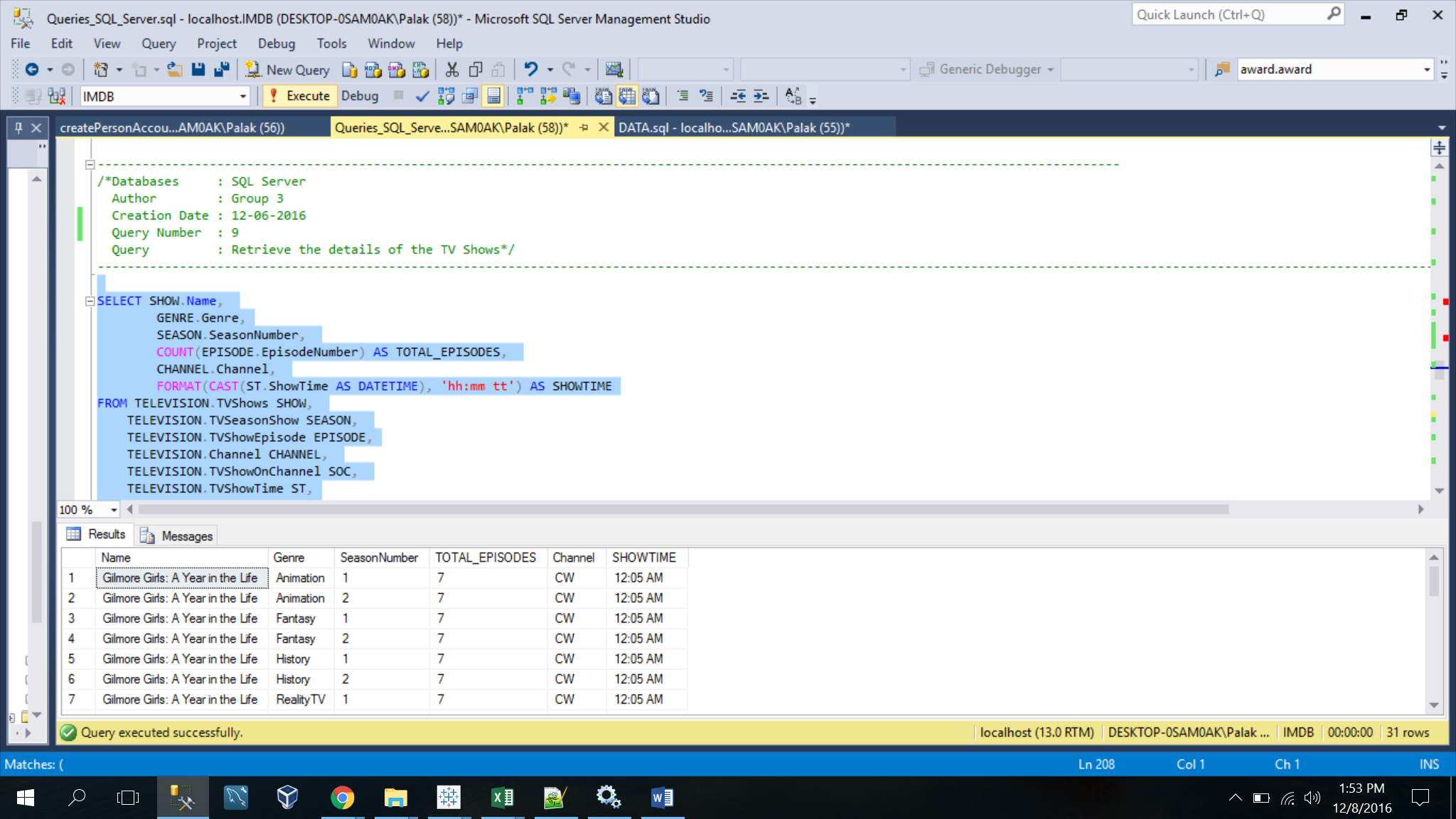
GROUP BY SHOW.Name,

GENRE.Genre,

SEASON.SeasonNumber,

CHANNEL.Channel,

ST.ShowTime;



-----------------------------------------------------------------------------------------

/\*Databases : SQL Server

Author : Group 3

Creation Date : 12-06-2016

Query Number : 10

Query : Retrieve the number of Users started particular Discussion in IMDB Community's Messageboard\*/

------------------------------------------------------------------------------------------

SELECT UD.UserID,

CONCAT(P.FirstName,' ',P.LastName) NAME,

MBC.CategoryName,

MBT.TopicName,

MB.DiscussionContent

FROM IMDBCOMMUNITY.MessageBoard MB,

IMDBCOMMUNITY.MessageBoardCategory MBC,

IMDBCOMMUNITY.MessageBoardTopic MBT,

REVIEW.USERDETAILS UD,

PERSON.Person P

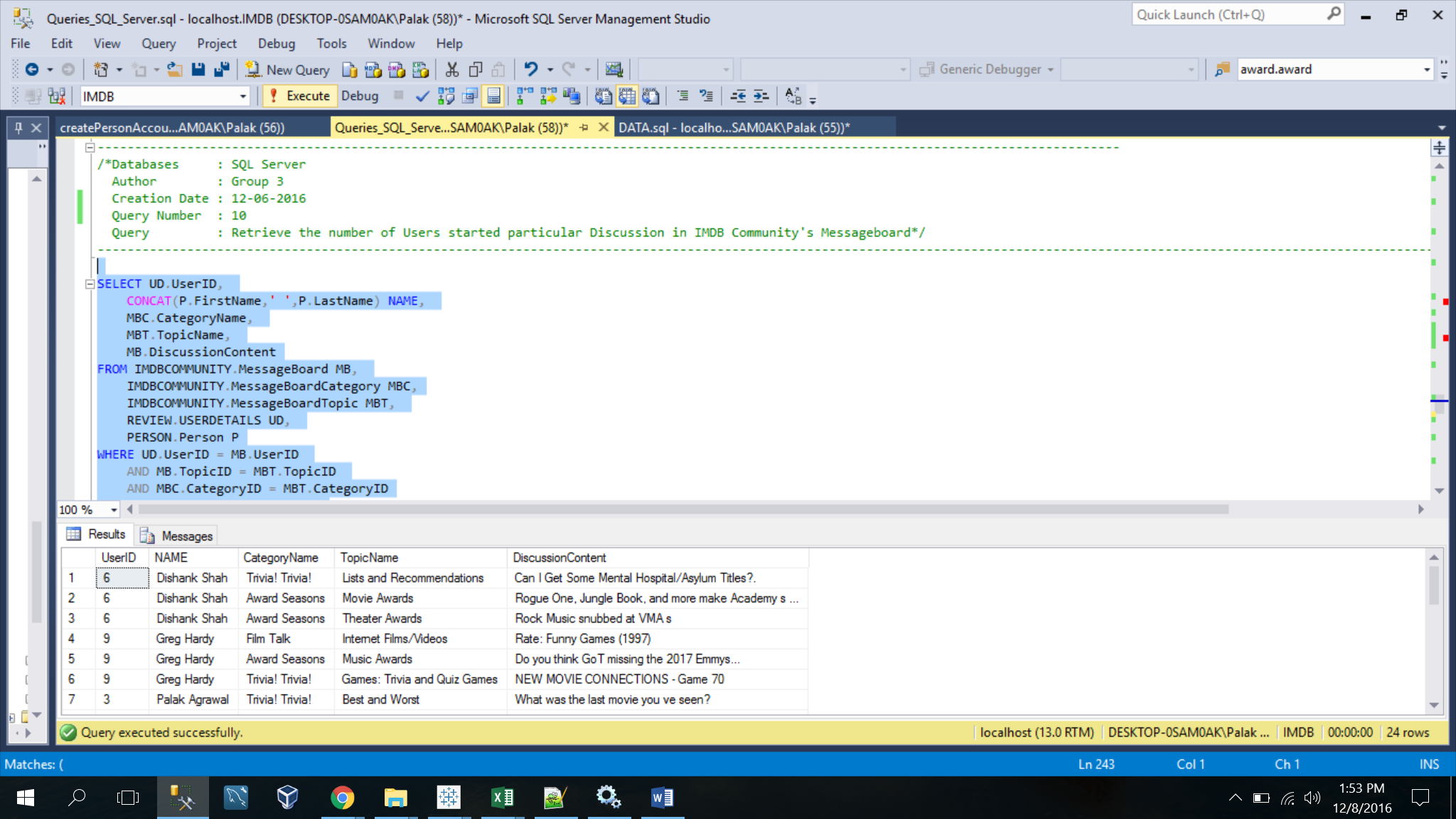
WHERE UD.UserID = MB.UserID

AND MB.TopicID = MBT.TopicID

AND MBC.CategoryID = MBT.CategoryID

AND P.PersonID = UD.UserID

ORDER BY NAME;



-----------------------------------------------------------------------------------------

/\*Databases : SQL Server

Author : Group 3

Creation Date : 12-06-2016

Query Number : 11

Query : Retrieve the Number of celebrities based on their type\*/

-----------------------------------------------------------------------------------------

SELECT \*

FROM (SELECT C.NAME,

TC.TYPE

FROM PERSON.CELEBRITY C,

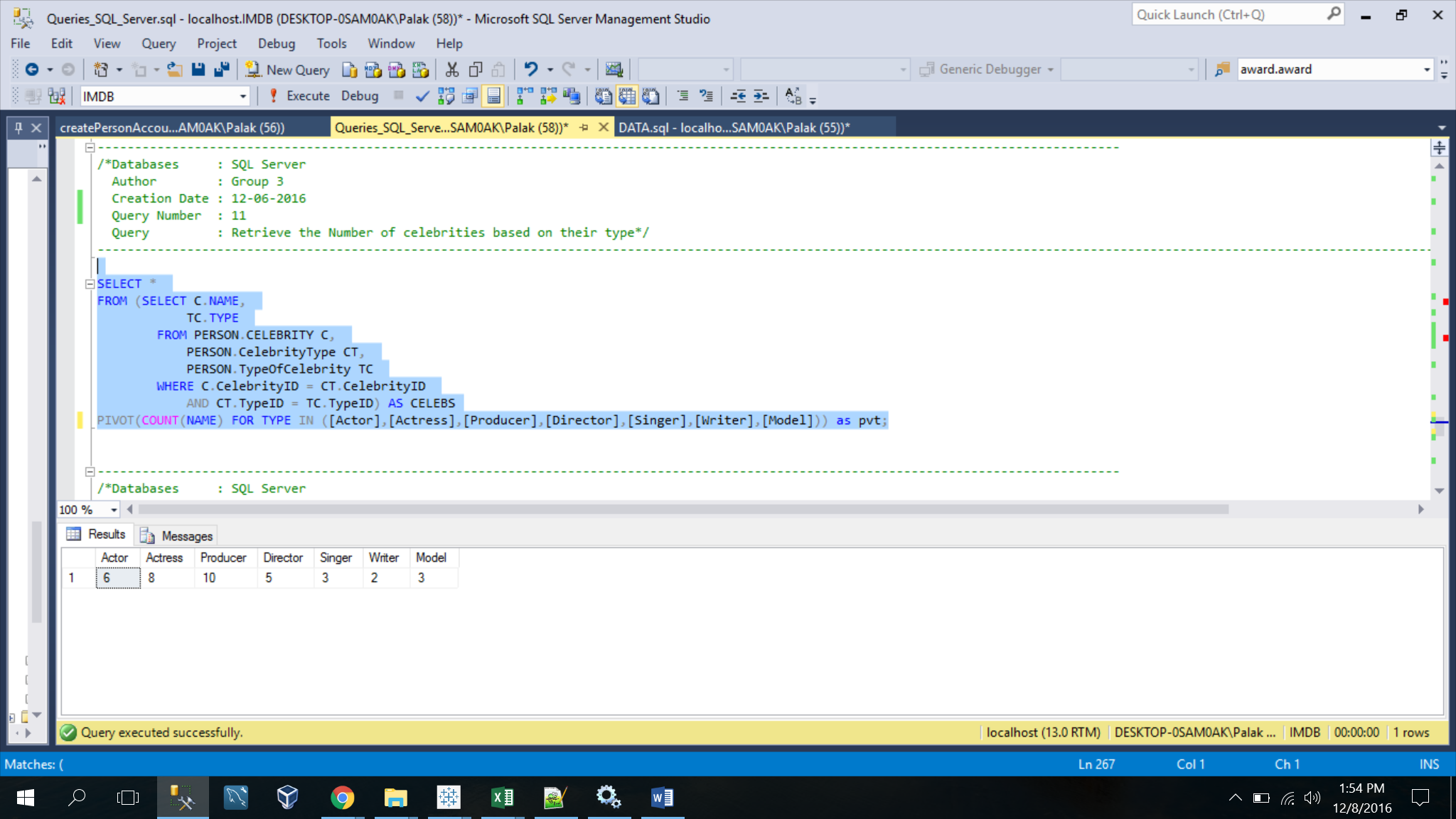
PERSON.CelebrityType CT,

PERSON.TypeOfCelebrity TC

WHERE C.CelebrityID = CT.CelebrityID

AND CT.TypeID = TC.TypeID) AS CELEBS

PIVOT(COUNT(NAME) FOR TYPE IN ([Actor],[Actress],[Producer],[Director],[Singer],[Writer],[Model])) as pvt;



-----------------------------------------------------------------------------------------

/\*Databases : SQL Server

Author : Group 3

Creation Date : 12-06-2016

Query Number : 12

Query : Retrieve the cheapest ticket for each movie in each theatre for different screens\*/

------------------------------------------------------------------------------------------

SELECT M.Name AS MOVIENAME,

T.Name AS THEATRENAME,

S.Name AS SCREEN,

MIN(MSIT.ShowPrice) AS CheapestTicket

FROM MOVIES.MovieShowInTheatre MSIT,

MOVIES.Movie M,

MOVIES.Theatre T,

MOVIES.Screen S,

MOVIES.ShowTime ST,

MOVIES.ShowDate SD

WHERE MSIT.MovieID = M.MovieID

AND T.TheatreID = MSIT.TheatreID

AND S.ScreenID = MSIT.ScreenID

AND ST.ShowTimeID = MSIT.ShowTimeID

AND SD.ShowDate = MSIT.ShowDate

GROUP BY M.Name,

T.NAME,

S.NAME

ORDER BY MOVIENAME DESC,

THEATRENAME DESC,

SCREEN DESC;

