1. **List All the directors who have got an award in current year.**

**SQL Server:**

select concat(s.DirFirstName,s.DirLastName) as DirectorName from dbo.AwardHeader as ah inner join dbo.AwardCategory as ac

on ac.AwardHeaderID = ah.AwardHeaderID

inner join dbo.AwardDetails as ad

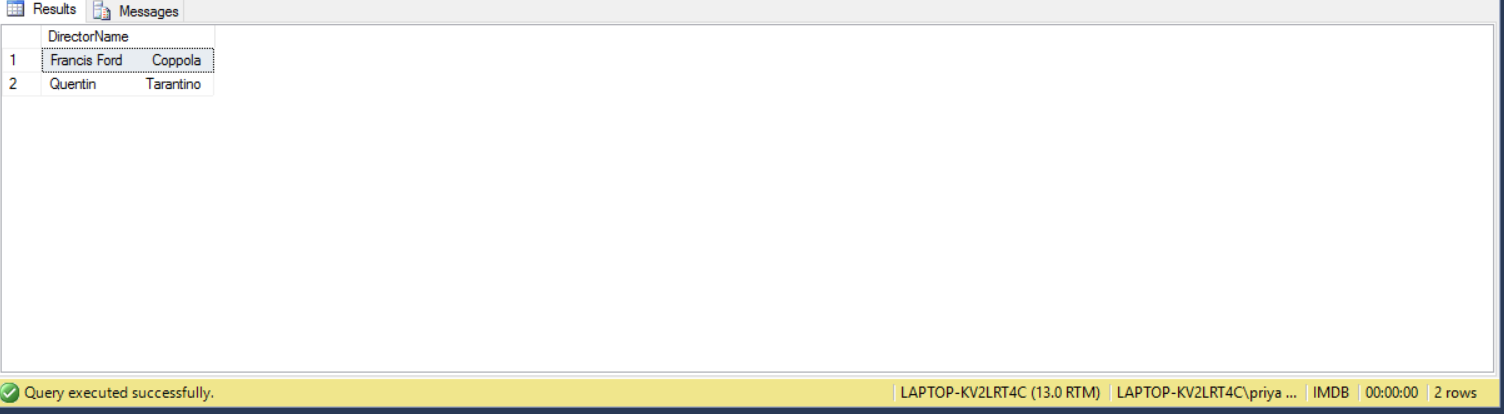
on ac.AwardCategoryID=ad.AwardCategoryID

inner join

dbo.Director as s on s.DirectorID=ad.DirectorID

where ah.Year = 2016

and ad.Status='Won';



**Oracle:**

select concat(rtrim(d.DirFirstName),concat(' ',ltrim(d.DirLastName))) as DirectorName

from AwardHeader ah

inner join AwardCategory ac on ah.AwardHeaderID = ac.AwardHeaderID

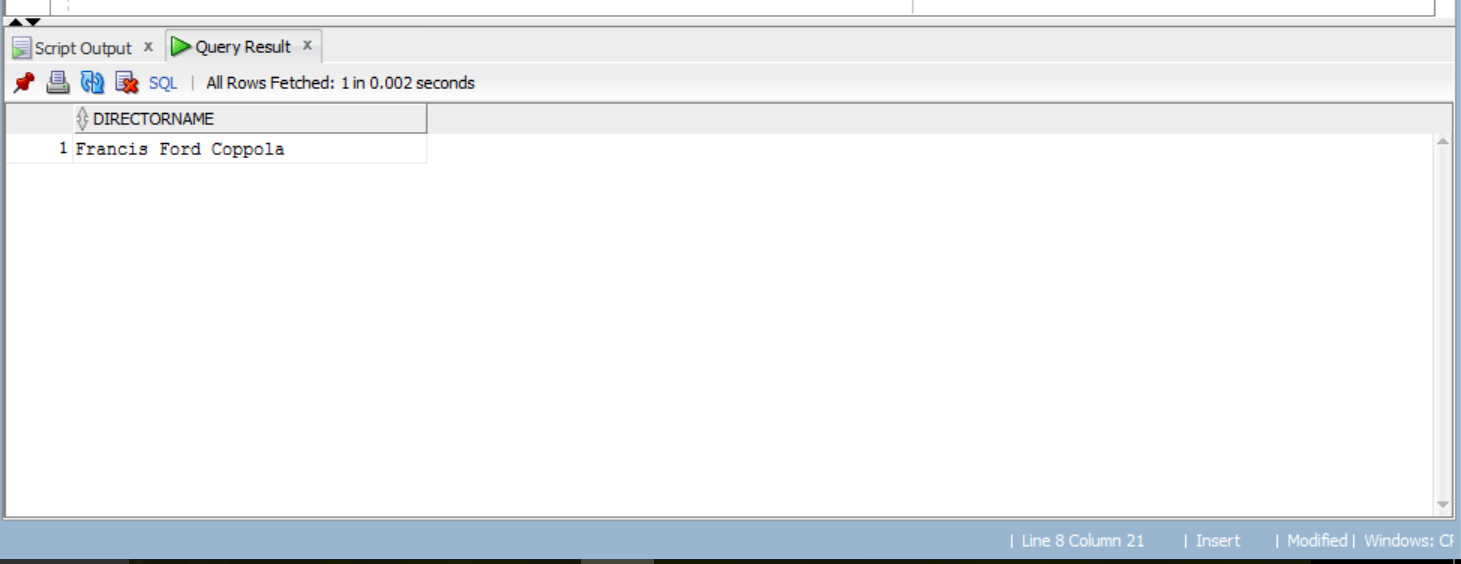
inner join AwardDetails ad on ac.AwardCategoryID = ad.AwardCategoryID

inner join Director d

on d.DirectorID = ad.DirectorID

where ah.Year = 2016

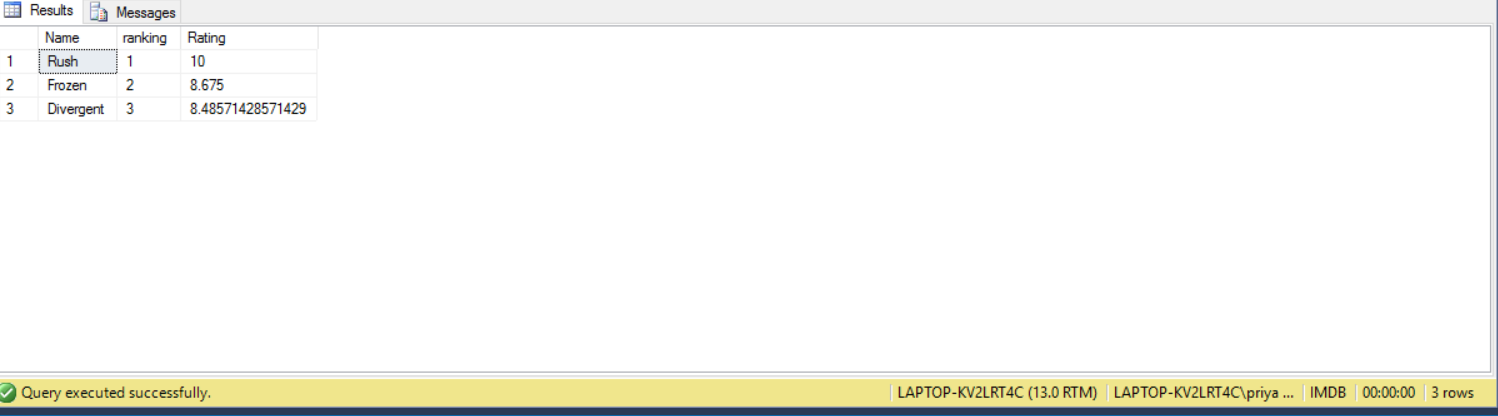
and ad.Status='Won';



1. **Find the top 3 movies based on rating using Rank function().**

**SQL SERVER:**

select top(3) Name, rank() over(order by rating desc) as ranking,Rating from dbo.MovieORTV;

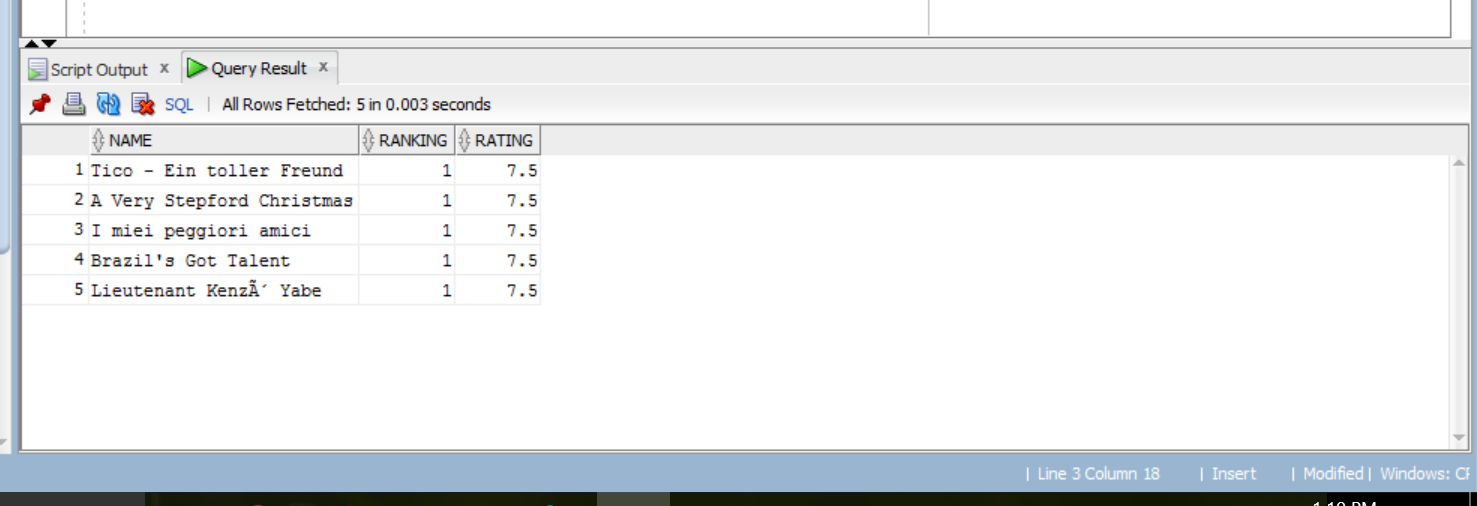


**Oracle:**

select Name, rank() over(order by rating desc) as ranking,Rating

from MovieORTV

where rownum <=5;



**3)Count the number of movies released every year using CTE.**

**SQL SERVER:**

with CTE\_Year(RealeaseYear,movieID)

as

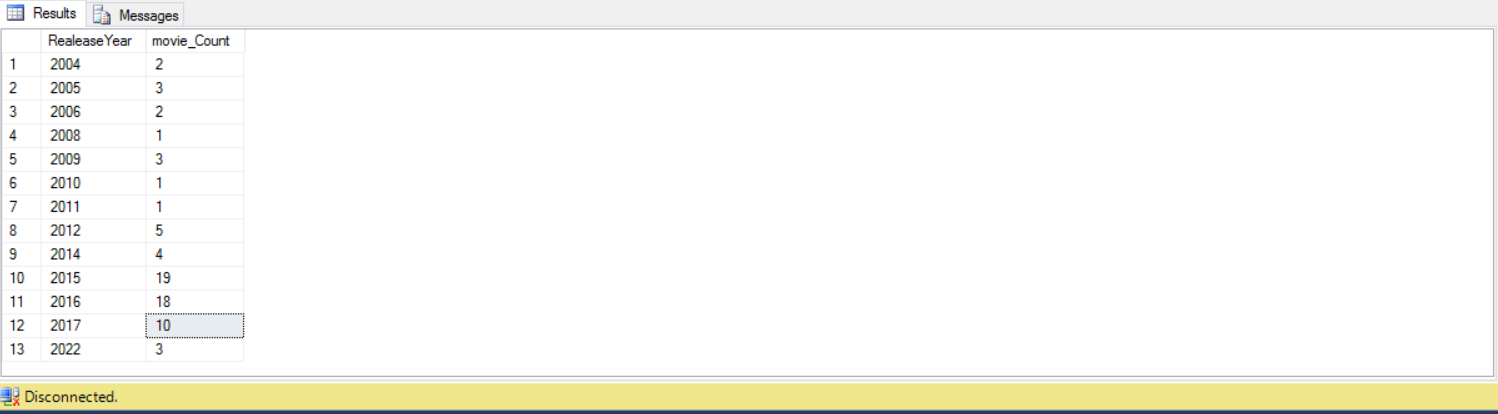
(

select YEAR(rd.Date),m.MovieORTVID from MovieORTV as m inner join ReleaseDate as rd on m.MovieORTVID=rd.MovieORTVID

)

select RealeaseYear,count( movieID) as movie\_Count from CTE\_Year

group by RealeaseYear;



**Another Method:**

select DATEPART(YEAR,rd.Date ) as year,count(m.MovieORTVID) as no\_of\_movies from MovieORTV as m inner join ReleaseDate as rd on m.MovieORTVID=rd.MovieORTVID

group by DATEPART(YEAR,rd.Date );

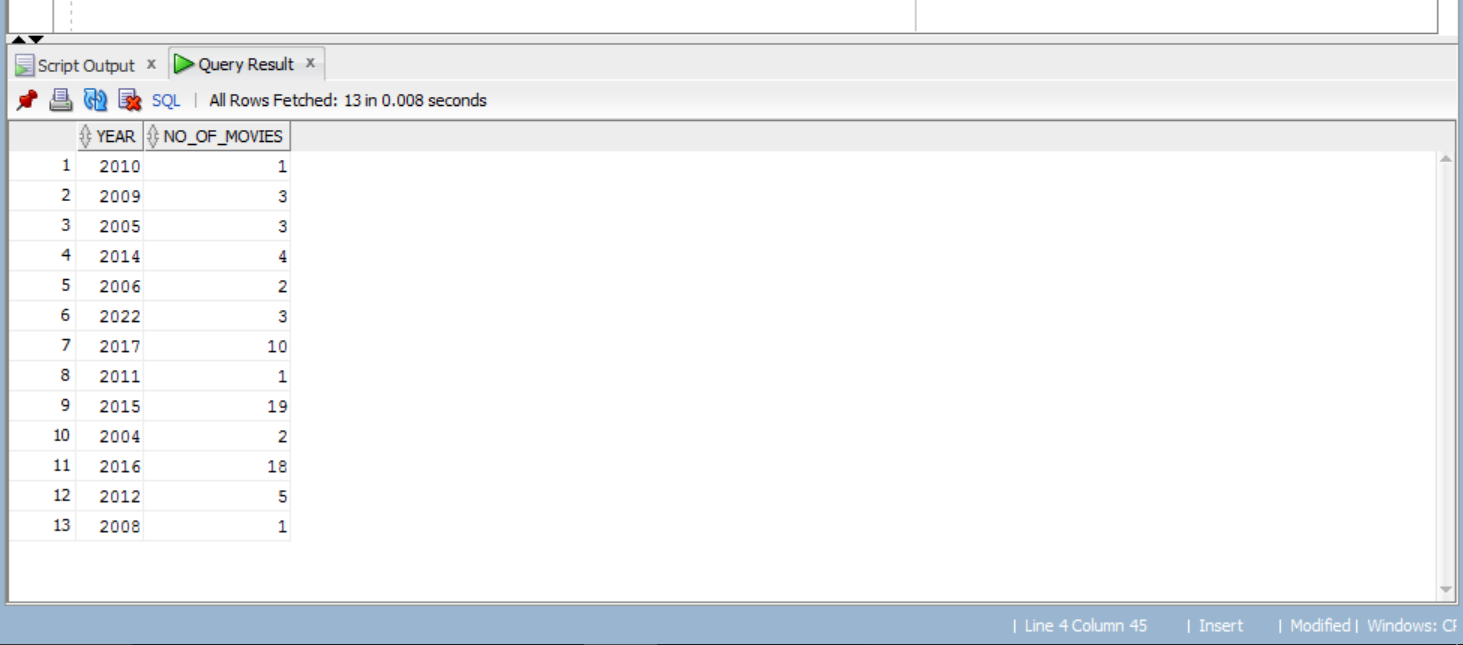
**ORACLE:**

select EXTRACT(YEAR from rd.releasedate ) as year,count(m.MovieORTVID) as no\_of\_movies

from MovieORTV m

inner join ReleaseDate rd on m.MovieORTVID = rd.MovieORTVID

group by EXTRACT(YEAR from rd.releasedate );



1. **Find out the details of Oscar winning movies Name and actor name who won it.**

**SQL SERVER:**

select 'OSCAR' as AwardName,mv.NAME,ac.AWARDCATEGORYNAME

,COALESCE(

(Select concat(STARFIRSTNAME,STARLASTNAME) from Star

where ad.STARID=star.starID),

(Select concat(WRITERFIRSTNAME,WRITERLASTNAME) from WRITER

where ad.WriterID=WRITER.WriterID),

(Select concat(DIRFIRSTNAME,DIRLASTNAME) from DIRECTOR

where ad.DIRECTORID=DIRECTOR.DIRECTORID),

(Select concat(PRODFIRSTNAME,PRODLASTNAME) from PRODUCER

where ad.PRODUCERID=PRODUCER.PRODUCERID))as WinnerName

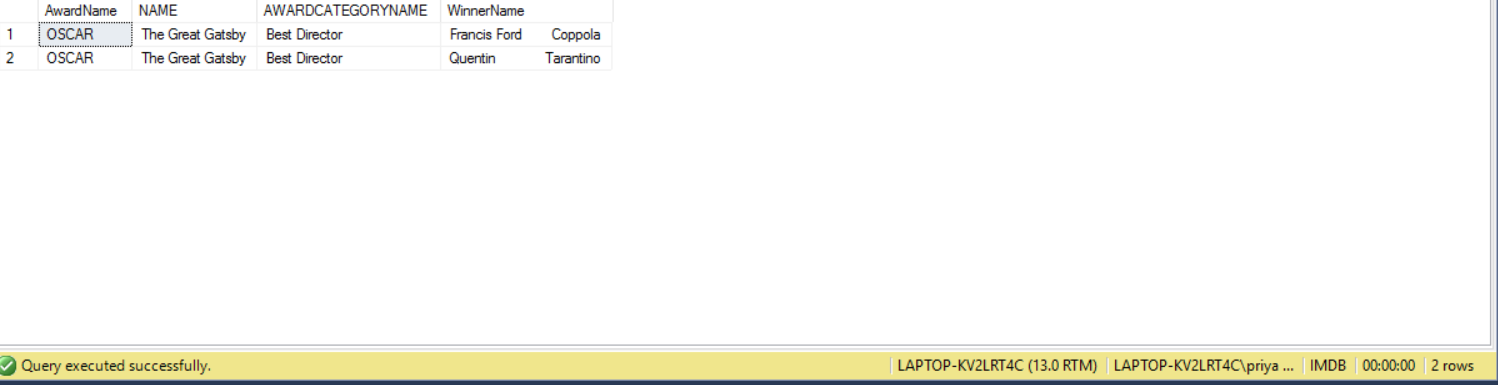
from AWARDHEADER ah

inner join MOVIEORTV mv on ah.MOVIEORTVID = mv.MOVIEORTVID

inner join AWARDCATEGORY ac on ac.AWARDHEADERID = ah.AWARDHEADERID

inner join AWARDDETAILS ad on ad.AWARDCATEGORYID = ac.AWARDCATEGORYID

where ah.AWARDNAME = 'Oscars';



**ORACLE:**

select 'OSCAR' as AwardName,ah.MOVIEORTVID,mv.NAME,ac.AWARDCATEGORYNAME

,COALESCE(

(Select concat(STARFIRSTNAME,STARLASTNAME) from Star

where ad.STARID=star.starID),

(Select concat(WRITERFIRSTNAME,WRITERLASTNAME) from WRITER

where ad.WriterID=WRITER.WriterID),

(Select concat(DIRFIRSTNAME,DIRLASTNAME) from DIRECTOR

where ad.DIRECTORID=DIRECTOR.DIRECTORID),

(Select concat(PRODFIRSTNAME,PRODLASTNAME) from PRODUCER

where ad.PRODUCERID=PRODUCER.PRODUCERID))as WinnerName

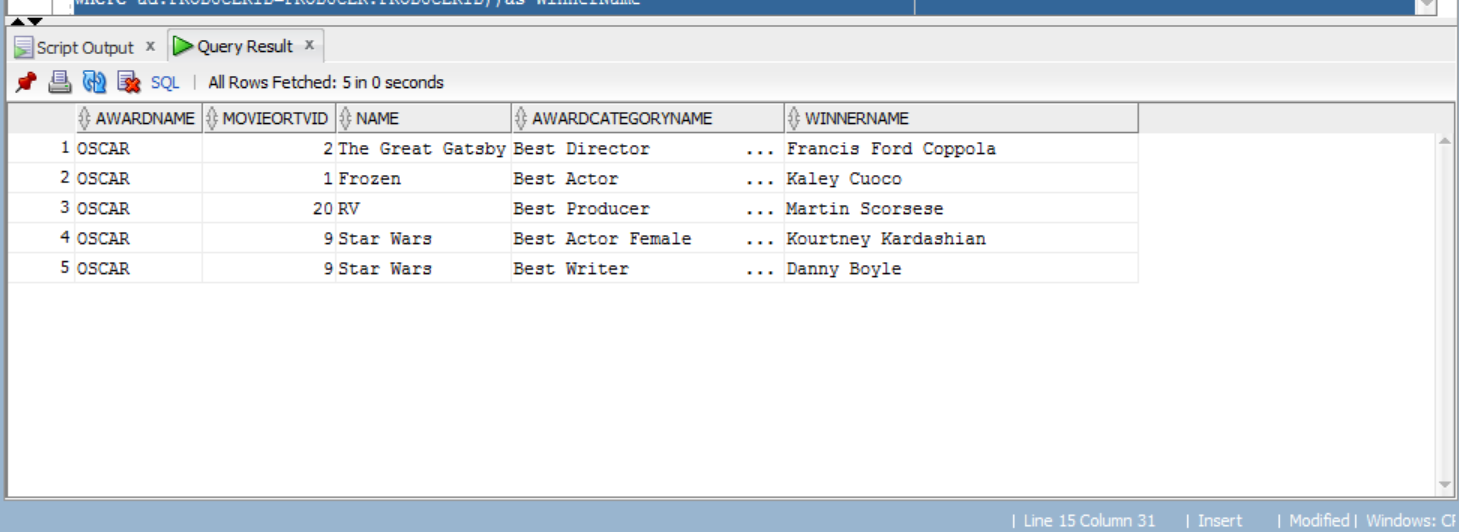
from AWARDHEADER ah

inner join MOVIEORTV mv on ah.MOVIEORTVID = mv.MOVIEORTVID

inner join AWARDCATEGORY ac on ac.AWARDHEADERID = ah.AWARDHEADERID

inner join AWARDDETAILS ad on ad.AWARDCATEGORYID = ac.AWARDCATEGORYID

where ah.AWARDNAME = 'Oscars';



1. Distribute the movies in groups of Excellent,good,bad,very bad based on Rating

SQL SERVER:

select count(MOVIEORTVID) as NoOfMovie,'Excellent' as MovieType from MOVIEORTV

where RATING >= 7.5

and Type = 'Movie'

union

select count(MOVIEORTVID) as NoOfMovie,'Good' as MovieType from MOVIEORTV

where RATING BETWEEN 5 and 7.4

and Type = 'Movie'

union

select count(MOVIEORTVID) as NoOfMovie,'Poor' as MovieType from MOVIEORTV

where RATING BETWEEN 3 and 5

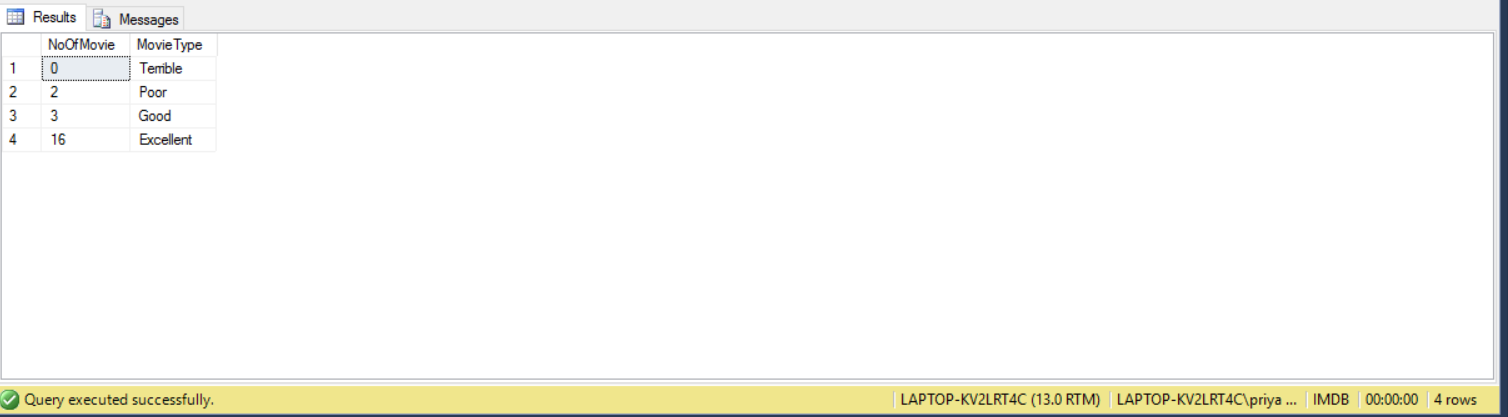
and Type = 'Movie'

union

select count(MOVIEORTVID) as NoOfMovie,'Terrible' as MovieType from MOVIEORTV

where RATING < 3

and Type = 'Movie';



**ORACLE:**

select count(MOVIEORTVID) as NoOfMovie,'Excellent' as MovieType from MOVIEORTVwhere RATING >= 7.5and Type = 'Movie'union

select count(MOVIEORTVID) as NoOfMovie,'Good' as MovieType from MOVIEORTV

where RATING BETWEEN 5 and 7.4

and Type = 'Movie'

union

select count(MOVIEORTVID) as NoOfMovie,'Poor' as MovieType from MOVIEORTV

where RATING BETWEEN 3 and 5

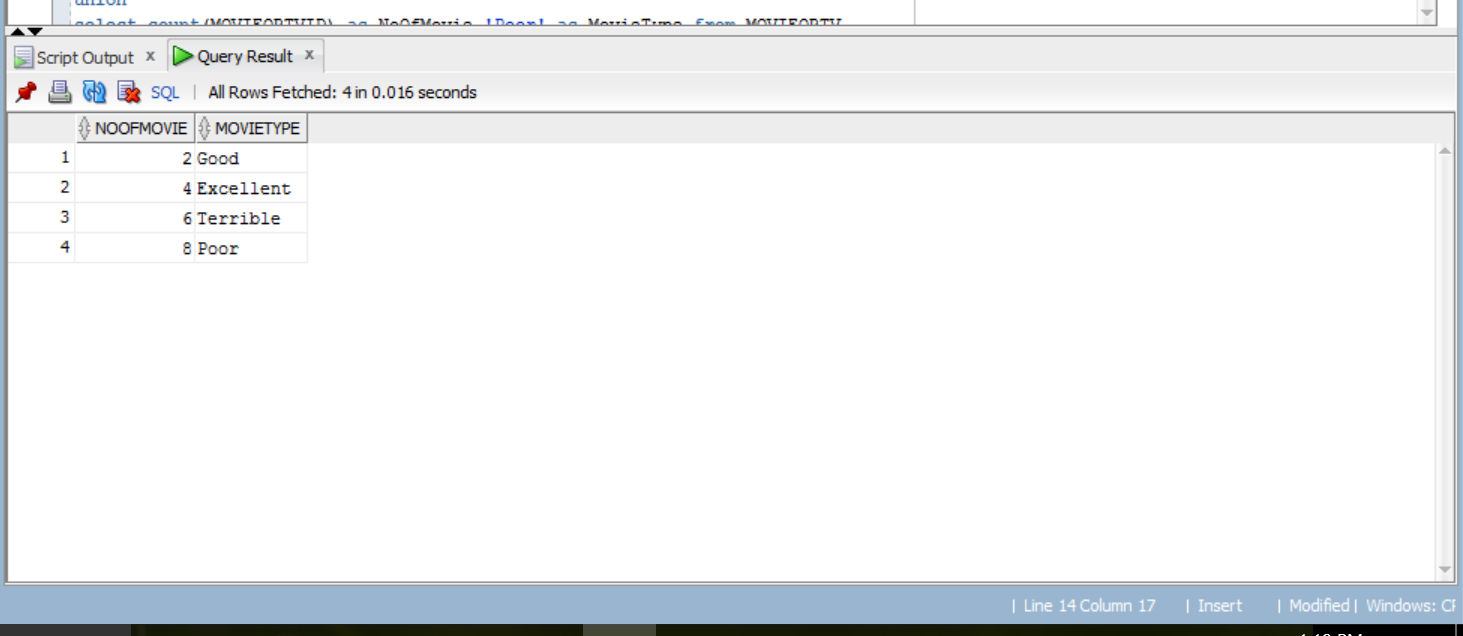
and Type = 'Movie'

union

select count(MOVIEORTVID) as NoOfMovie,'Terrible' as MovieType from MOVIEORTV

where RATING < 3

and Type = 'Movie';



1. **Count the movies according to Genre using Pivot.**

**SQL SERVER:**

Select Crime,Romance,Drama,Fantasy,Comedy,Family

from

(select g.GenreCategoryName,gm.MovieORTVID from dbo.Genre g

inner join dbo.GenreMovie gm on g.GenreID = gm.GenreID) as NoOFMovies

pivot (COUNT(MovieORTVID) For GenreCategoryName IN ( [Crime],[Romance],[Drama],[Fantasy],[Comedy],[Family])) as pvt;



**ORACLE:**

Select Crime,Romance,Drama,Fantasy,Comedy,Family

from

(select g.GenreCategoryName,gm.MovieORTVID from Genre g

inner join GenreMovie gm on g.GenreID = gm.GenreID)

PIVOT (COUNT(MovieORTVID) For GenreCategoryName IN ('Crime' as Crime ,'Romance' as Romance,'Drama' As Drama,'Fantasy' as Fantasy

,'Comedy' as Comedy,'Family' as Family));

1. **Award Details of the movies Nominated or Won.**

select movie.Name,head.AwardName from dbo.MovieORTV as movie

join dbo.AwardHeader as head

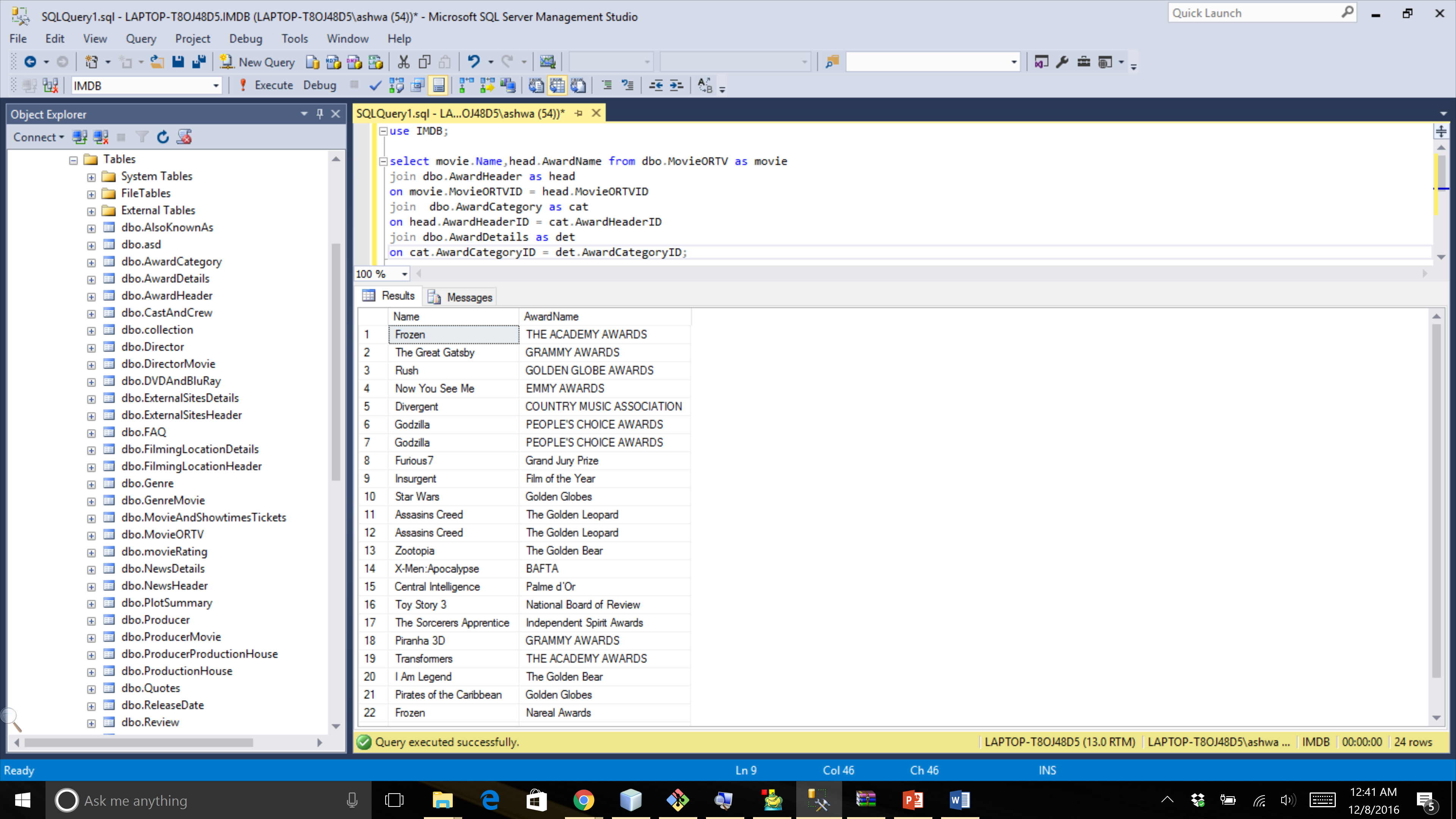
on movie.MovieORTVID = head.MovieORTVID

join dbo.AwardCategory as cat

on head.AwardHeaderID = cat.AwardHeaderID

join dbo.AwardDetails as det

on cat.AwardCategoryID = det.AwardCategoryID;



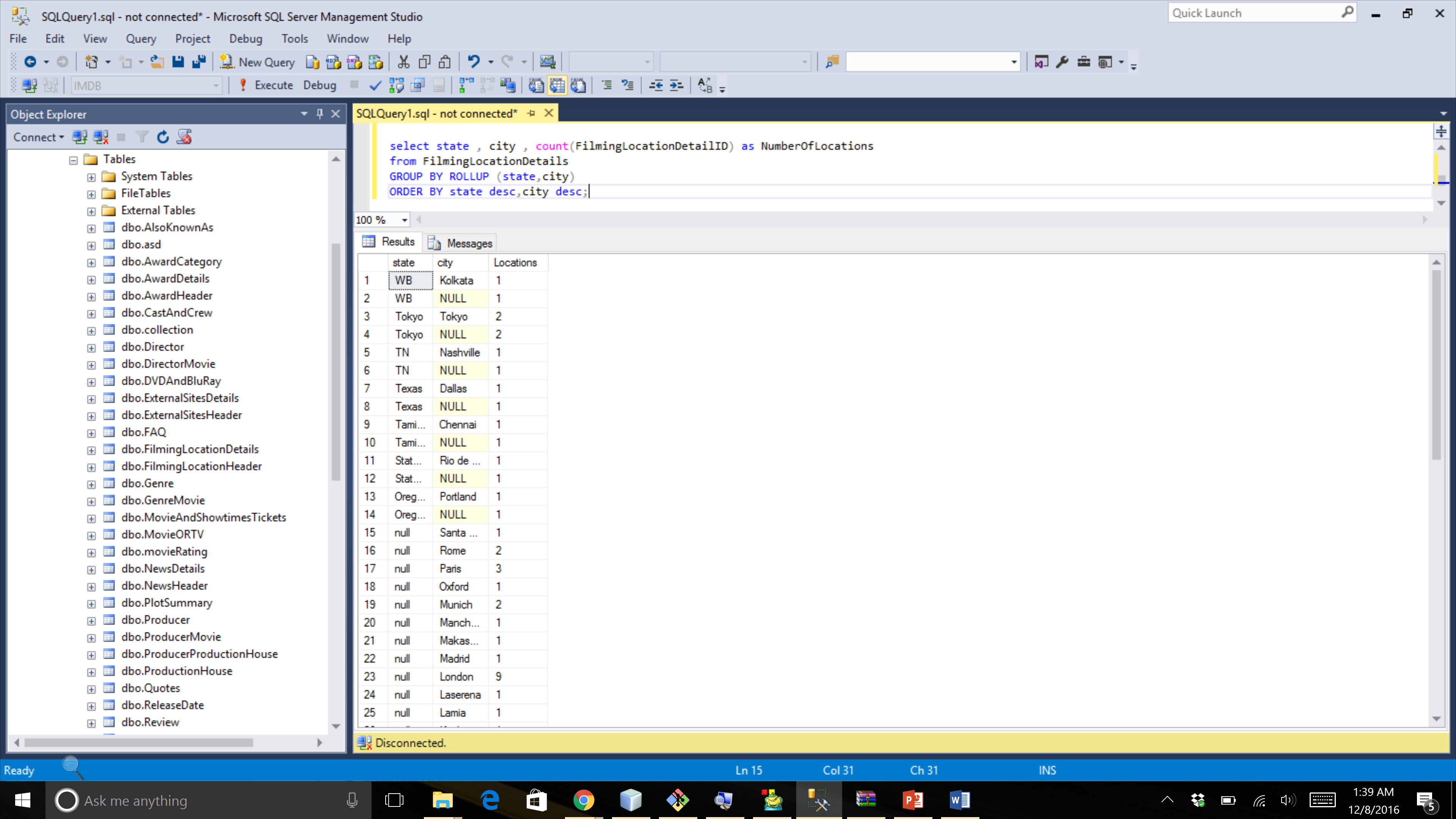
1. **How many films are shot at a particular location display it according to subcategories like state and city.**

select state , city , count(FilmingLocationDetailID) as NumberOfLocations

from FilmingLocationDetails

GROUP BY ROLLUP (state,city)

ORDER BY state desc,city desc;



**ORACLE:**

**select state , city , count(FilmingLocationDetailID) as NumberOfLocations**

**from FilmingLocationDetails**

**GROUP BY ROLLUP (state,city)**

**ORDER BY state desc,city desc;**

1. **List of directors who have directed English movies select top 10 rows only.**

select TOP 10 movie.name as MovieOrTv,dir.DirFirstName+' '+dir.DirLastName as FullName from dbo.MovieORTV as movie

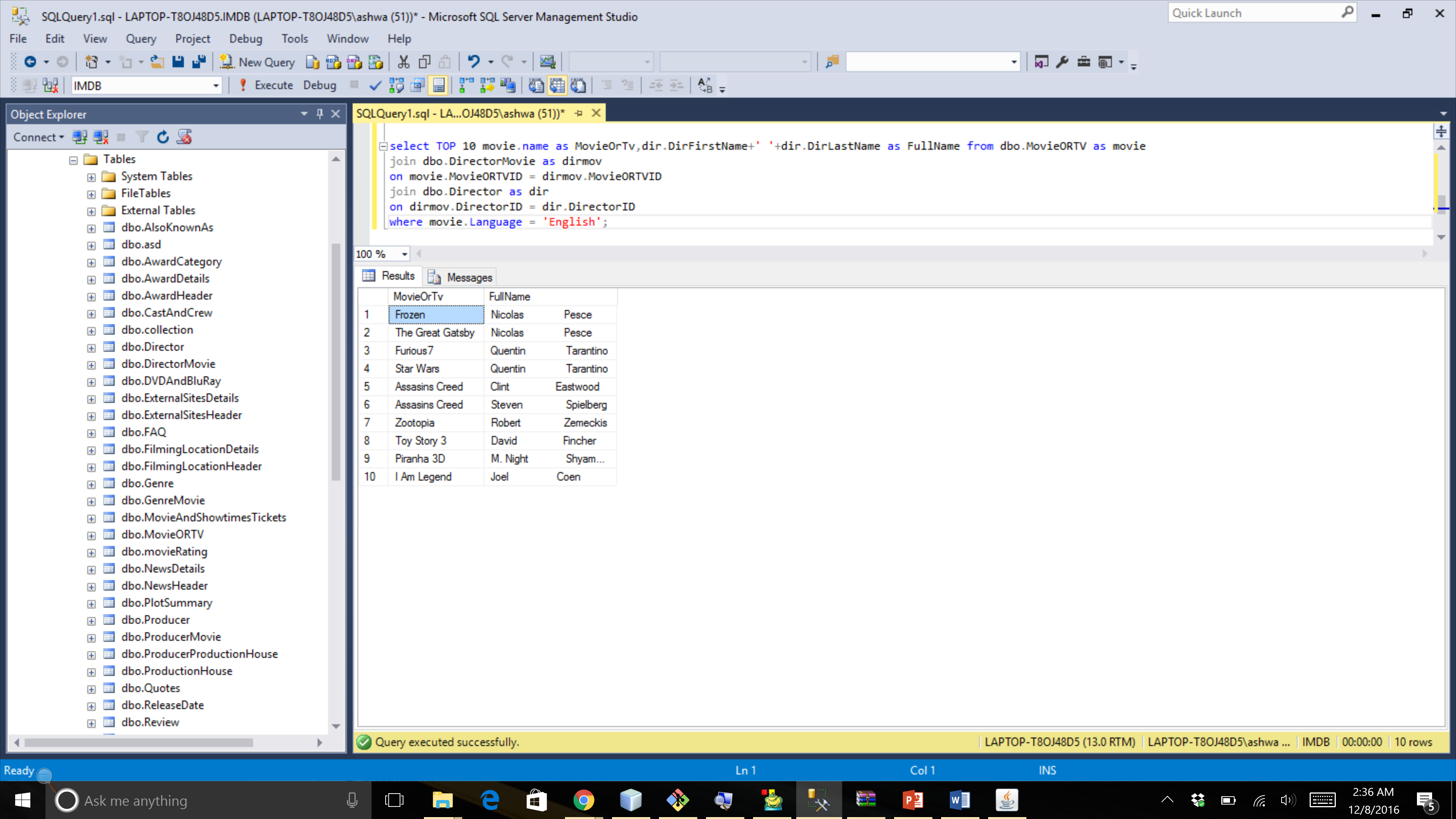
join dbo.DirectorMovie as dirmov

on movie.MovieORTVID = dirmov.MovieORTVID

join dbo.Director as dir

on dirmov.DirectorID = dir.DirectorID

where movie.Language = 'English';



**Procedure:**

1. **Procedure to backup the database:**

CREATE PROCEDURE sp\_full\_bkp

as

BEGIN

DECLARE @tab\_name char(50);

DECLARE @q varchar(1500);

DECLARE @finished1 INTEGER=0;

DECLARE cur4 CURSOR FOR SELECT TABLE\_NAME FROM INFORMATION\_SCHEMA.TABLES WHERE table\_schema='dbo' and table\_type='BASE TABLE';

DROP DATABASE IF EXISTS IMDB\_backup;

CREATE DATABASE IMDB\_backup;

open cur4;

block2: begin

FETCH cur4 into @tab\_name;

while(@@FETCH\_STATUS=0)

begin

--DECLARE CONTINUE HANDLER FOR NOT FOUND SET finished1 = 1;

-- IF finished1= 1 THEN

--LEAVE cur\_loop;

--END IF;

SET @q='DROP TABLE IF EXISTS IMDB\_backup.'+ @tab\_name;

EXECUTE (@q);

SET @q='Select \* into IMDB\_backup.dbo.' + @tab\_name +' from IMDB.dbo.'+ @tab\_name + ' WHERE 1=1';

EXECUTE (@q);

FETCH cur4 into @tab\_name;

END

end

close cur4;

END

1. **Procedure to find top 50 Actors:**

create procedure Top\_50\_Actors

as

begin

select TOP 50 count(\*) as AwardCount, concat(rtrim(s.StarFirstName),concat(' ',ltrim(s.StarLastName))) as StarName

from Star as s

inner join AwardDetails as ad

on s.StarID = ad.StarID

where ad.Status = 'Won' and s.Gender = 'M'

group by concat(rtrim(s.StarFirstName),concat(' ',ltrim(s.StarLastName)))

order by AwardCount desc, StarName

end;

1. **Procedure to find top 50 Actress:**

create procedure Top\_50\_Actors

as

begin

select TOP 50 count(\*) as AwardCount, concat(rtrim(s.StarFirstName),concat(' ',ltrim(s.StarLastName))) as StarName

from Star as s

inner join AwardDetails as ad

on s.StarID = ad.StarID

where ad.Status = 'Won' and s.Gender = 'F'

group by concat(rtrim(s.StarFirstName),concat(' ',ltrim(s.StarLastName)))

order by AwardCount desc, StarName

1. **Procedure to find top 50 Directors:**

create procedure Top\_50\_Directors

as

begin

select TOP 50 count(\*) as AwardCount, concat(rtrim(d.DirFirstName),concat(' ',ltrim(d.DirLastName))) as DirecorName

from Director as d

inner join AwardDetails as ad

on d.DirectorID = ad.DirectorID

where ad.Status = 'Won'

group by concat(rtrim(d.DirFirstName),concat(' ',ltrim(d.DirLastName)))

order by AwardCount desc, DirecorName

end;

**Triggers:**

1. **To update and calculate the rating for movie**

create trigger dbo.Rating

on dbo.movieRating

after insert,update

as

BEGIN

DECLARE @movieId as int;

declare @temp as float;

select @movieId=MovieORTVID from inserted;

select @temp=avg(rating) from dbo.movieRating where MovieORTVID=@movieId;

update dbo.MovieORTV set Rating = @temp where MovieORTVID = @movieId

END;

1. **To update gross collection of movie based on everyday collection.**

create trigger dbo.Gross\_Collection

on dbo.Collection

after insert,update

as

BEGIN

DECLARE @count as int;

declare @temp as float;

declare @amount as float;

declare @movieId as int;

select @movieId=MovieORTVID from inserted;

select @amount=sum(dayCollection) from dbo.collection as c

where c.MovieORTVID=@movieId;

update dbo.MovieORTV set GrossCollection = @amount where MovieORTVID = @movieId

END;

**VIEWS:**

1. **View for movie details.**

Create view MovieDetailsView as (select mv.MOVIEORTVID,mv.NAME,g.GENRECATEGORYNAME,mv.DESCRIPTION,mv.LANGUAGE,mv.RATING,concat(d.DIRFIRSTNAME,d.DIRLASTNAME) as Director,

concat(p.PRODFIRSTNAME,p.PRODLASTNAME) as Producer,concat(s.STARFIRSTNAME,s.STARLASTNAME) as Star,rd.Date as ReleaseDate,

t.THEATRENAME,t.ADDRESSLINE,t.CITY,t.STATE,t.COUNTRY,st.SHOWTIME

from MOVIEORTV mv

inner join DIRECTORMOVIE dm on mv.MOVIEORTVID = dm.MOVIEORTVID

inner join DIRECTOR d on dm.DIRECTORID = d.DIRECTORID

inner join PRODUCERMOVIE pm on mv.MOVIEORTVID = pm.MOVIEORTVID

inner join PRODUCER p on pm.PRODUCERID = p.PRODUCERID

inner join STARMOVIE sm on mv.MOVIEORTVID = sm.MOVIEORTVID

inner join STAR s on sm.STARID = s.STARID

inner join RELEASEDATE rd on mv.MOVIEORTVID = rd.MOVIEORTVID

inner join MovieAndShowtimesTickets mt on mv.MOVIEORTVID = mt.MOVIEANDTVID

inner join THEATRE t on mt.SHOWTIMESANDTICKETSID = t.SHOWTIMESANDTICKETSID

inner join SHOWTIMES st on t.SHOWTIMESANDTICKETSID = st.SHOWTIMESANDTICKETSID

inner join GENREMOVIE gm on mv.MOVIEORTVID = gm.MOVIEORTVID

inner join GENRE g on gm.GENREID = g.GENREID

where mv.TYPE = 'Movie');

1. **View for details of tvShow**

Create view TVShowView as (select mv.NAME,mv.MOVIEORTVID,tv.SHOWSEASONNUMBER,episode.AIRDATE,episode.EPISODNAME from MOVIEORTV mv

inner join TVSHOWSEASON tv on mv.MOVIEORTVID = tv.MOVIEORTVID

inner join TVSHOWSEASONEPISODE episode on tv.TVSHOWSEASONID = episode.TVSHOWSEASONID

where mv.TYPE = 'TV Series');

**Table Values Function**

1. **Count of movies coming Soon**

create function dbo.Comming\_Soon()

returns table

as

return

(select mt.Name,rd.Date from dbo.MovieORTV as mt inner join dbo.ReleaseDate as rd on mt.MovieORTVID = rd.MovieORTVID

where rd.Date>getdate()

);