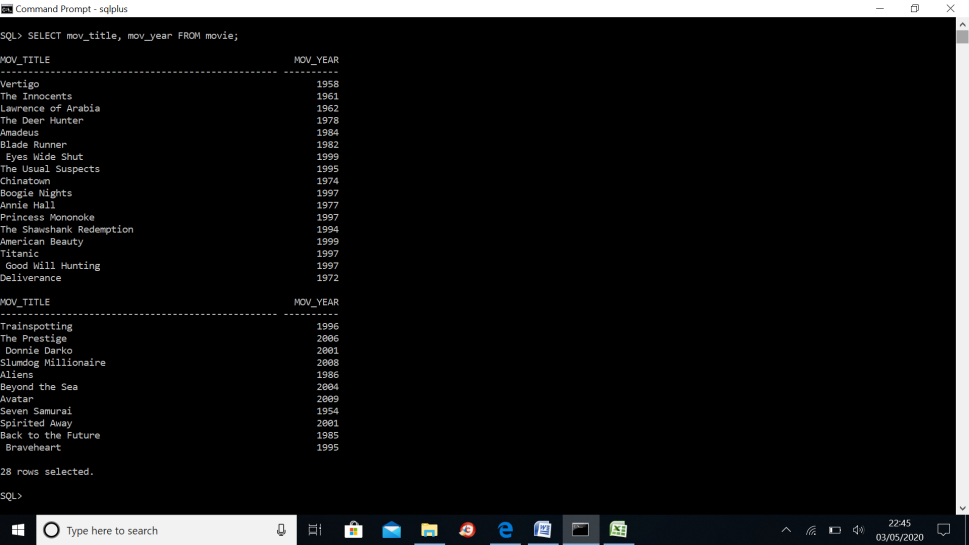
# SQL Exercises on Movie Database

## Using below ERD diagram and Sample Data.

## 

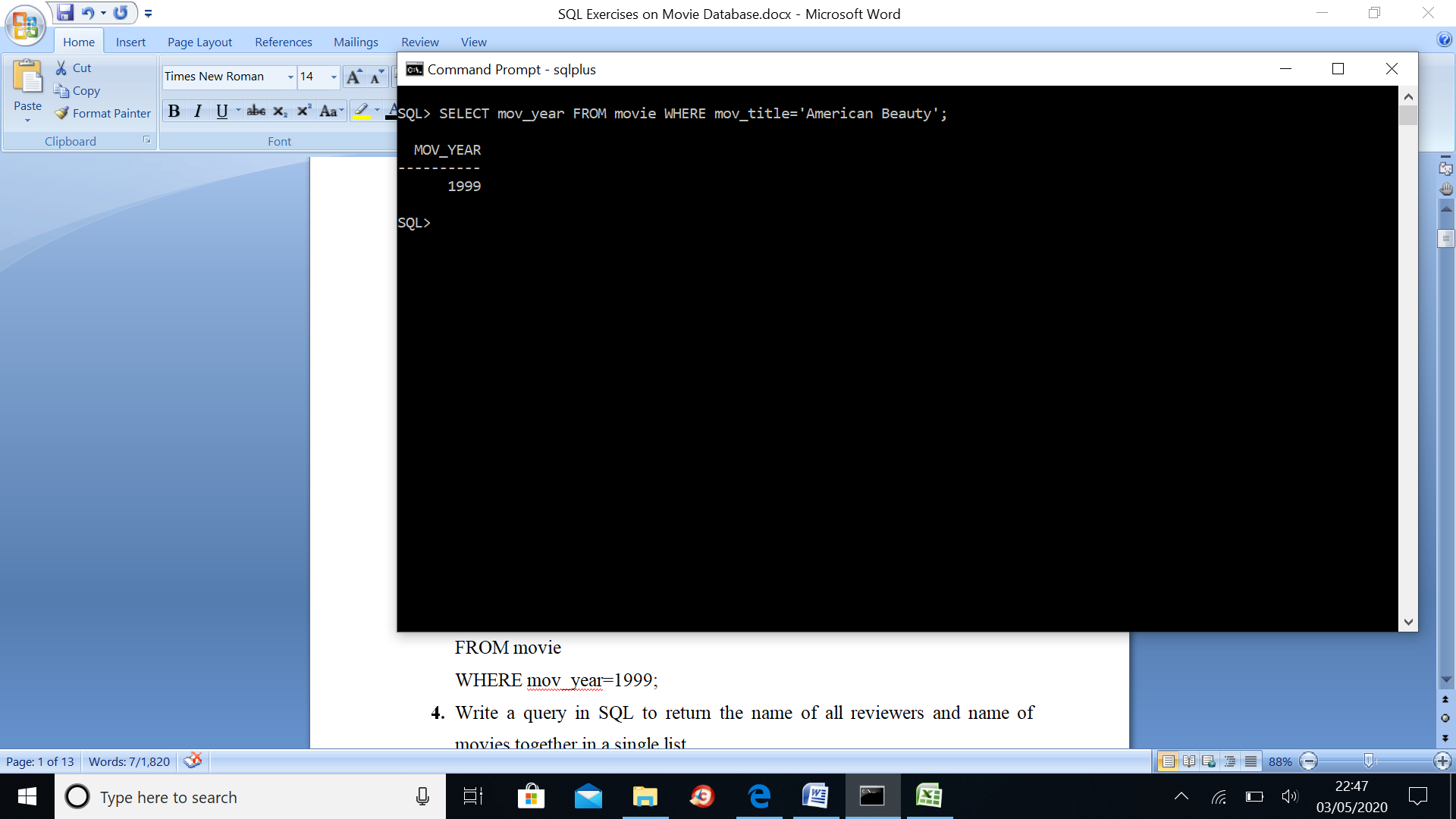
1. Write a query in SQL to find the name and year of the movies.#

SELECT mov\_title, mov\_year FROM movie;



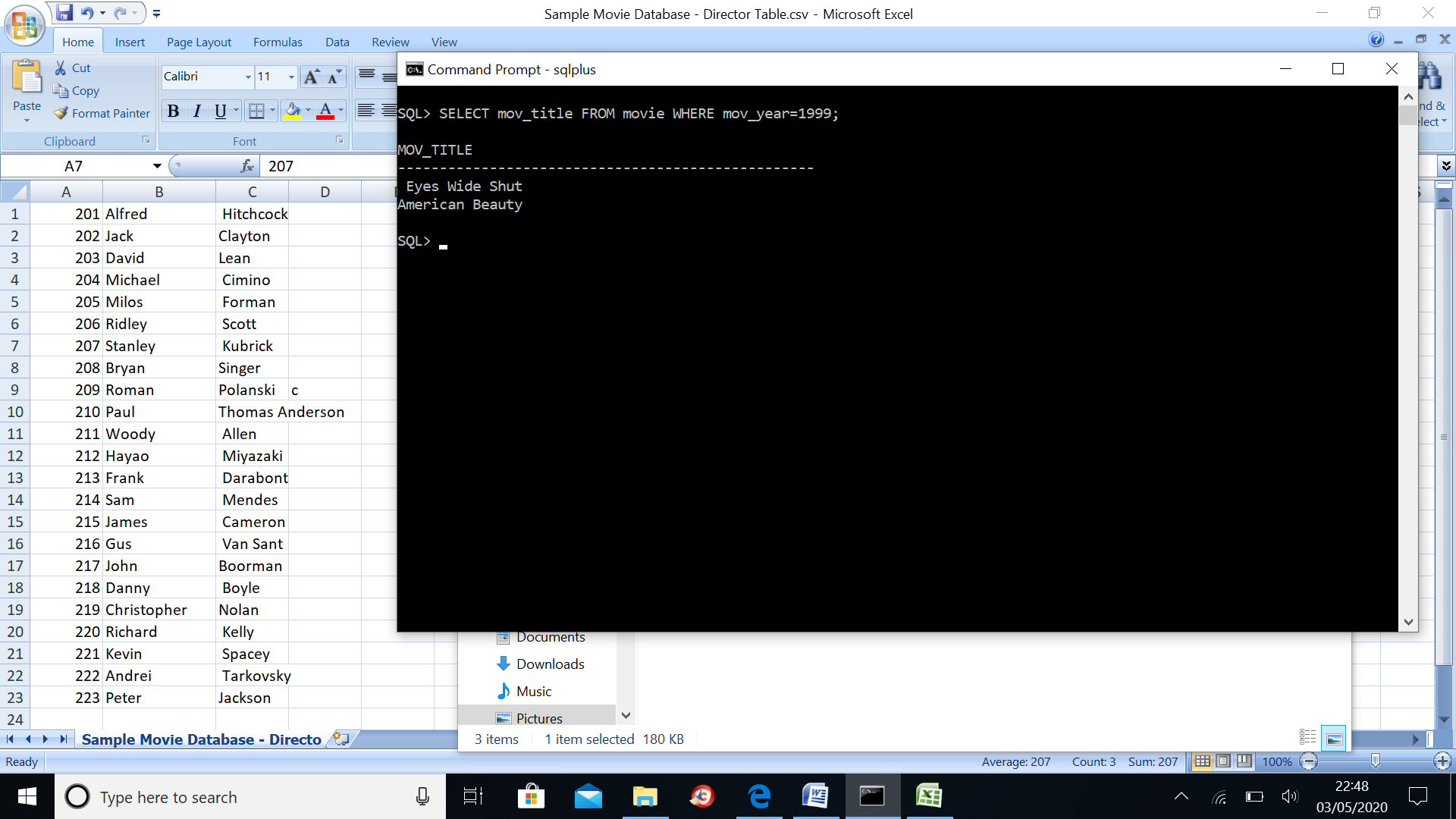
1. Write a query in SQL to find the year when the movie American Beauty released.

SELECT mov\_year FROM movie WHERE mov\_title='American Beauty';



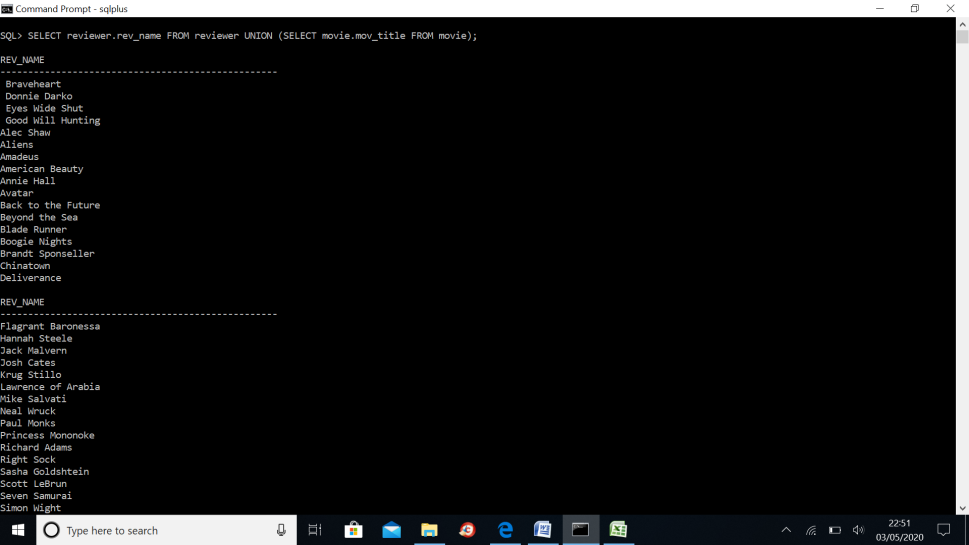
1. Write a query in SQL to find the movie which was released in the year 1999.

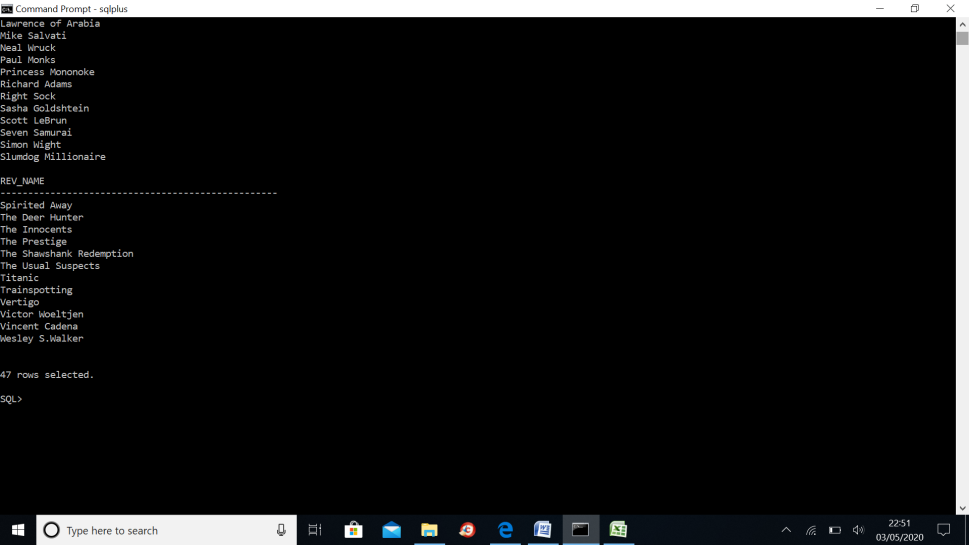
SELECT mov\_title FROM movie WHERE mov\_year=1999;



1. Write a query in SQL to return the name of all reviewers and name of movies together in a single list.

SELECT reviewer.rev\_name FROM reviewer UNION (SELECT movie.mov\_title FROM movie);



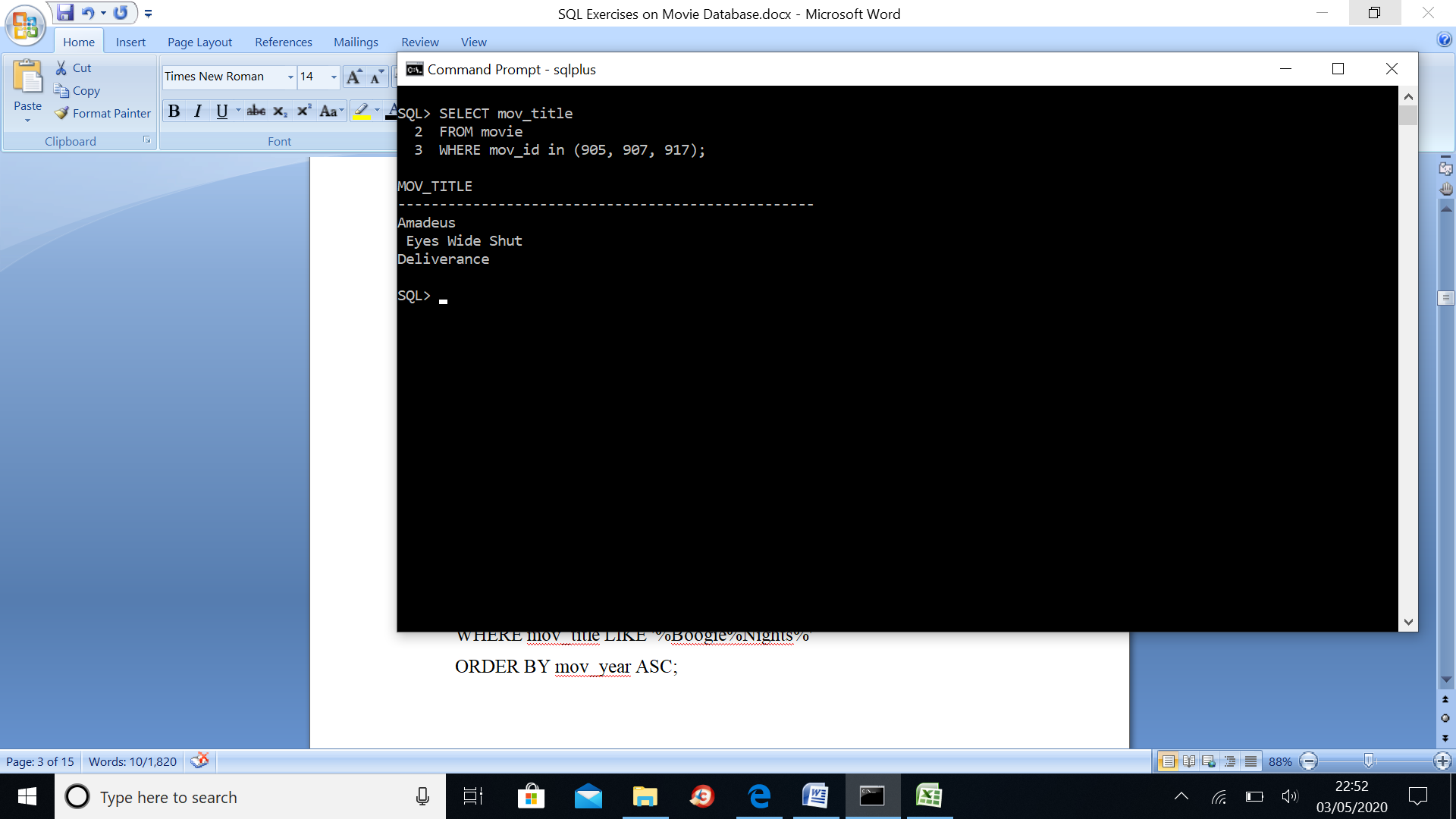


1. Write a query in SQL to find the titles of the movies with ID 905, 907, 917.

SELECT mov\_title

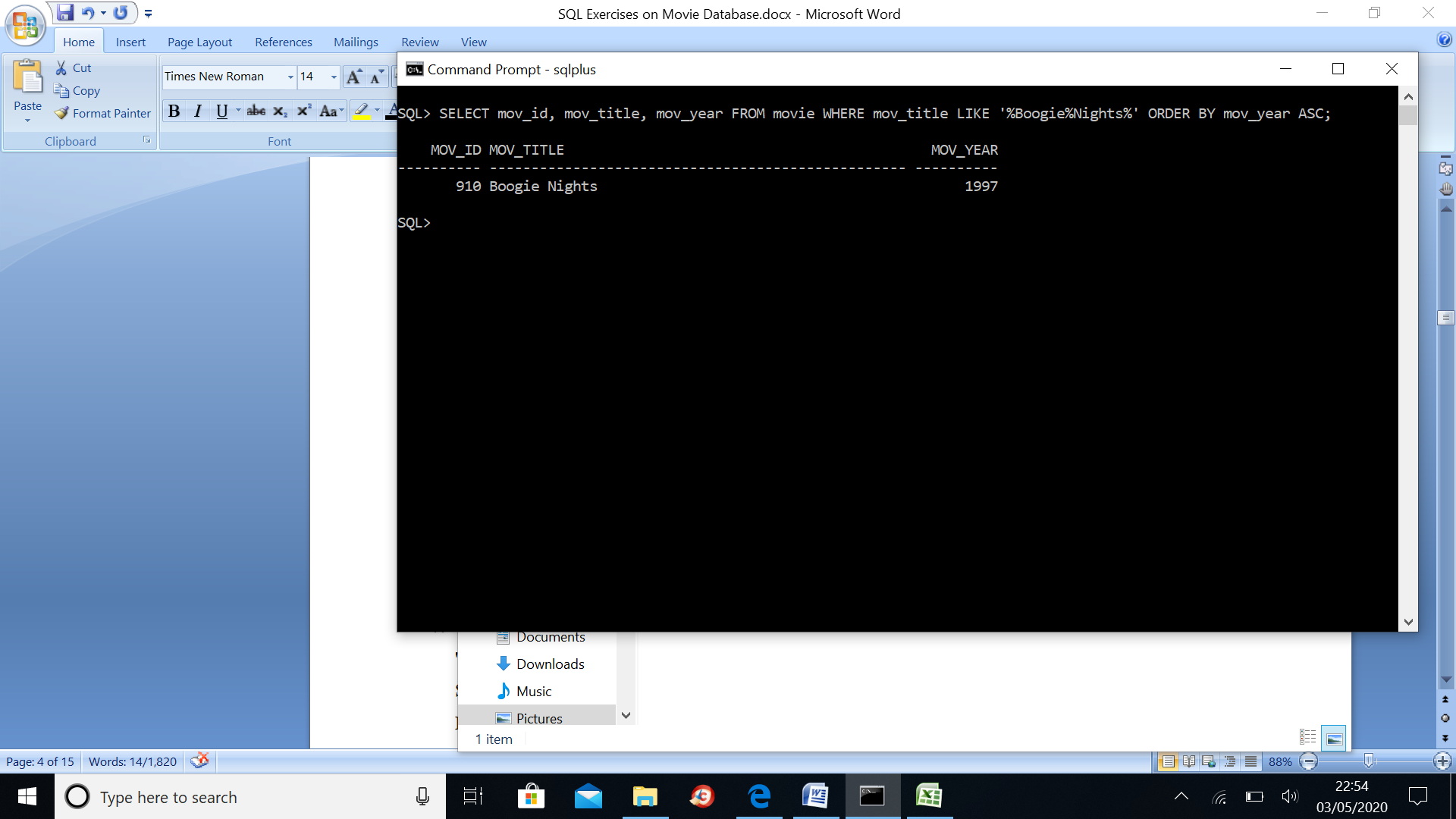
FROM movie

WHERE mov\_id in (905, 907, 917);



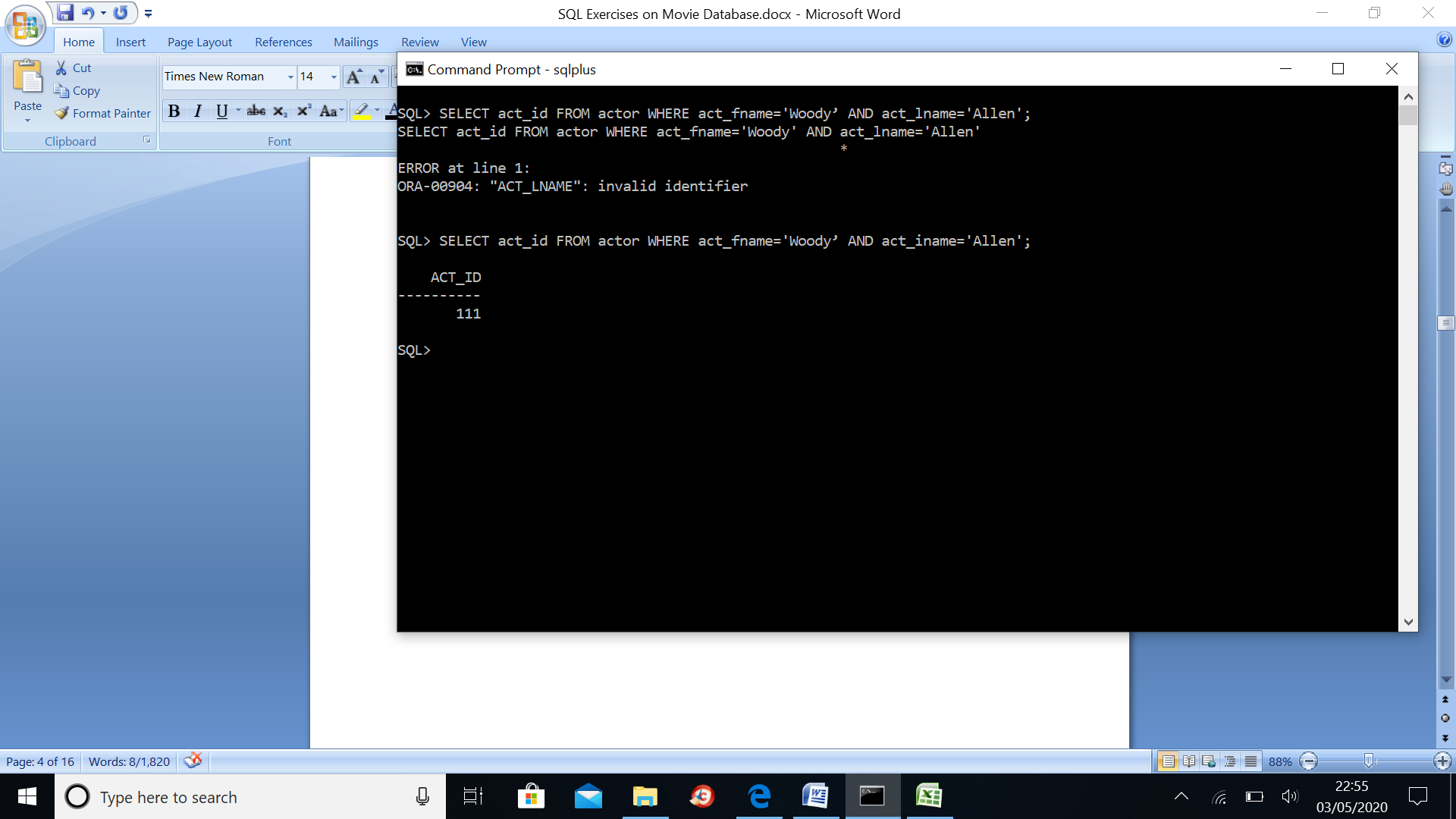
1. Write a query in SQL to find the list of all those movies with year which include the words Boogie Nights.

SELECT mov\_id, mov\_title, mov\_year FROM movie WHERE mov\_title LIKE '%Boogie%Nights%' ORDER BY mov\_year ASC;



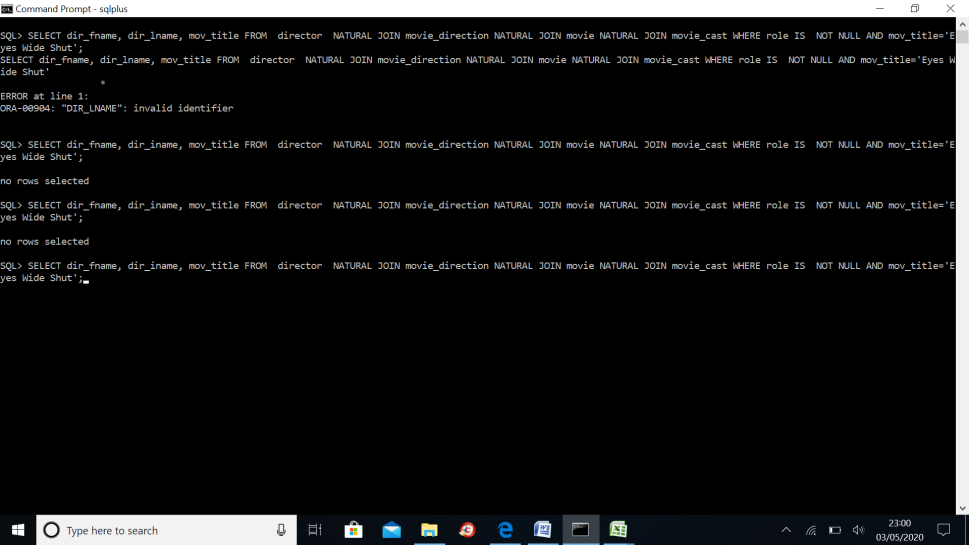
1. Write a query in SQL to find the ID number for the actor whose first name is 'Woody' and the last name is 'Allen'.

SELECT act\_id FROM actor WHERE act\_fname='Woody’ AND act\_lname='Allen';



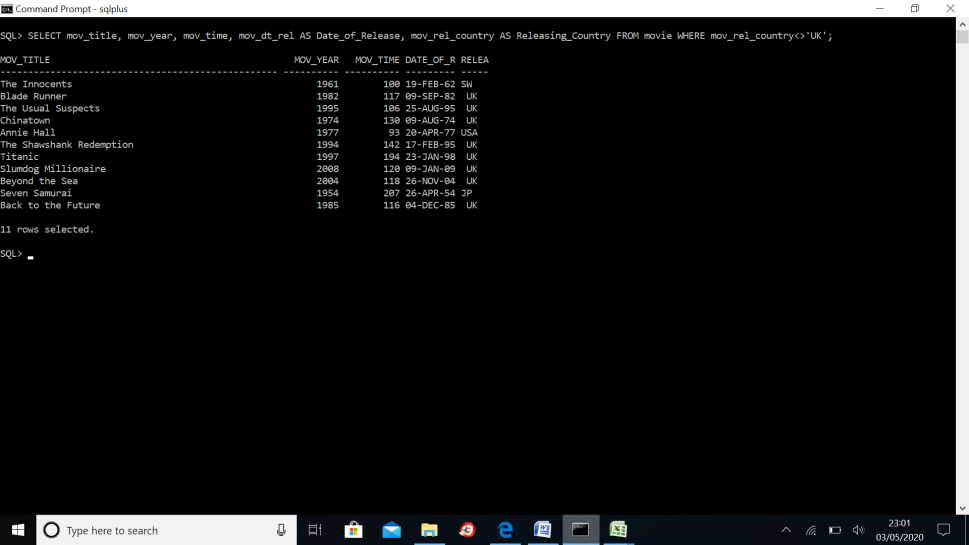
1. Write a query in SQL to find the name of the director (first and last names) who directed a movie that casted a role for 'Eyes Wide Shut'. (using subquery).

SELECT dir\_fname, dir\_lname, mov\_title FROM director NATURAL JOIN movie\_direction NATURAL JOIN movie NATURAL JOIN movie\_cast WHERE role IS NOT NULL AND mov\_title='Eyes Wide Shut';



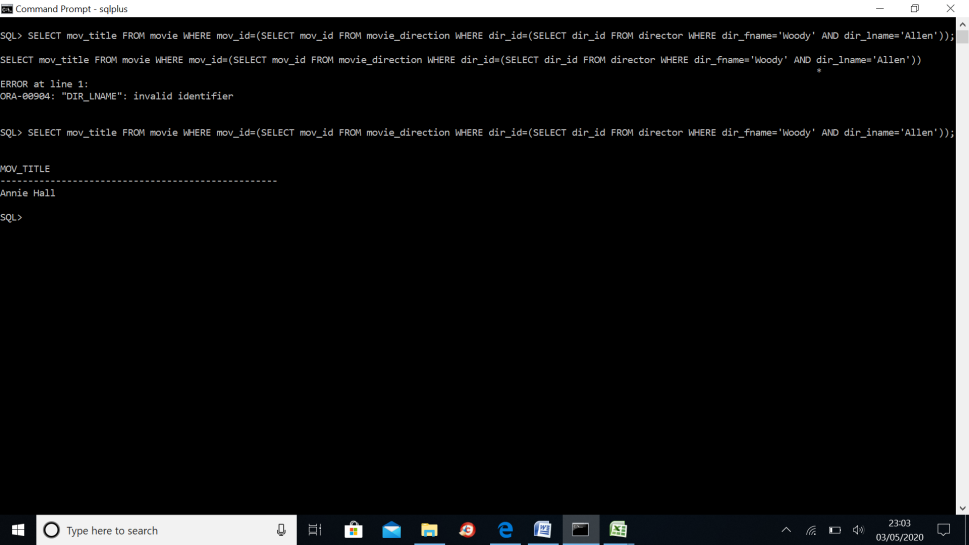
1. Write a query in SQL to list all the movies which released in the country other than UK.

SELECT mov\_title, mov\_year, mov\_time, mov\_dt\_rel AS Date\_of\_Release, mov\_rel\_country AS Releasing\_Country FROM movie WHERE mov\_rel\_country<>'UK';



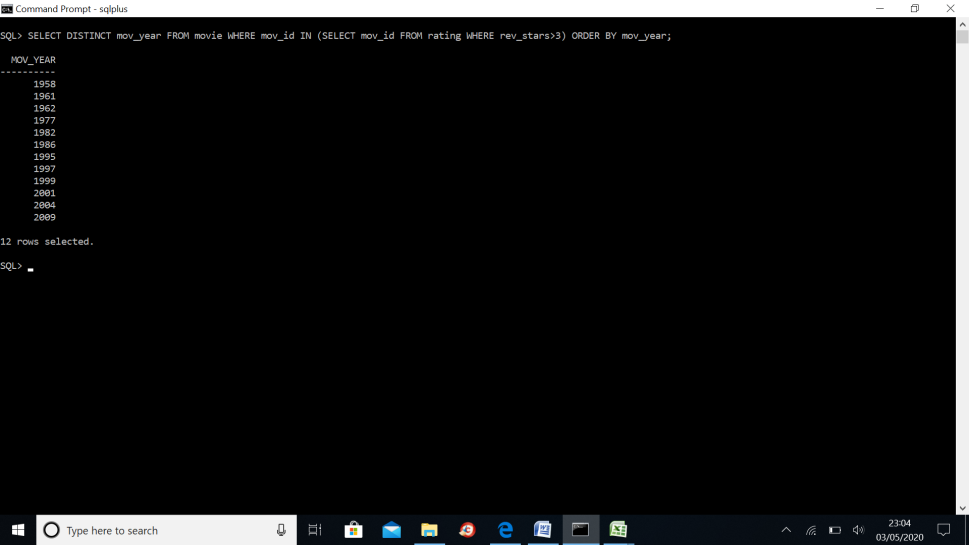
1. Write a query in SQL to find the titles of all movies directed by the director whose first and last name are Woddy Allen.

SELECT mov\_title FROM movie WHERE mov\_id=(SELECT mov\_id FROM movie\_direction WHERE dir\_id=(SELECT dir\_id FROM director WHERE dir\_fname='Woody' AND dir\_lname='Allen'));



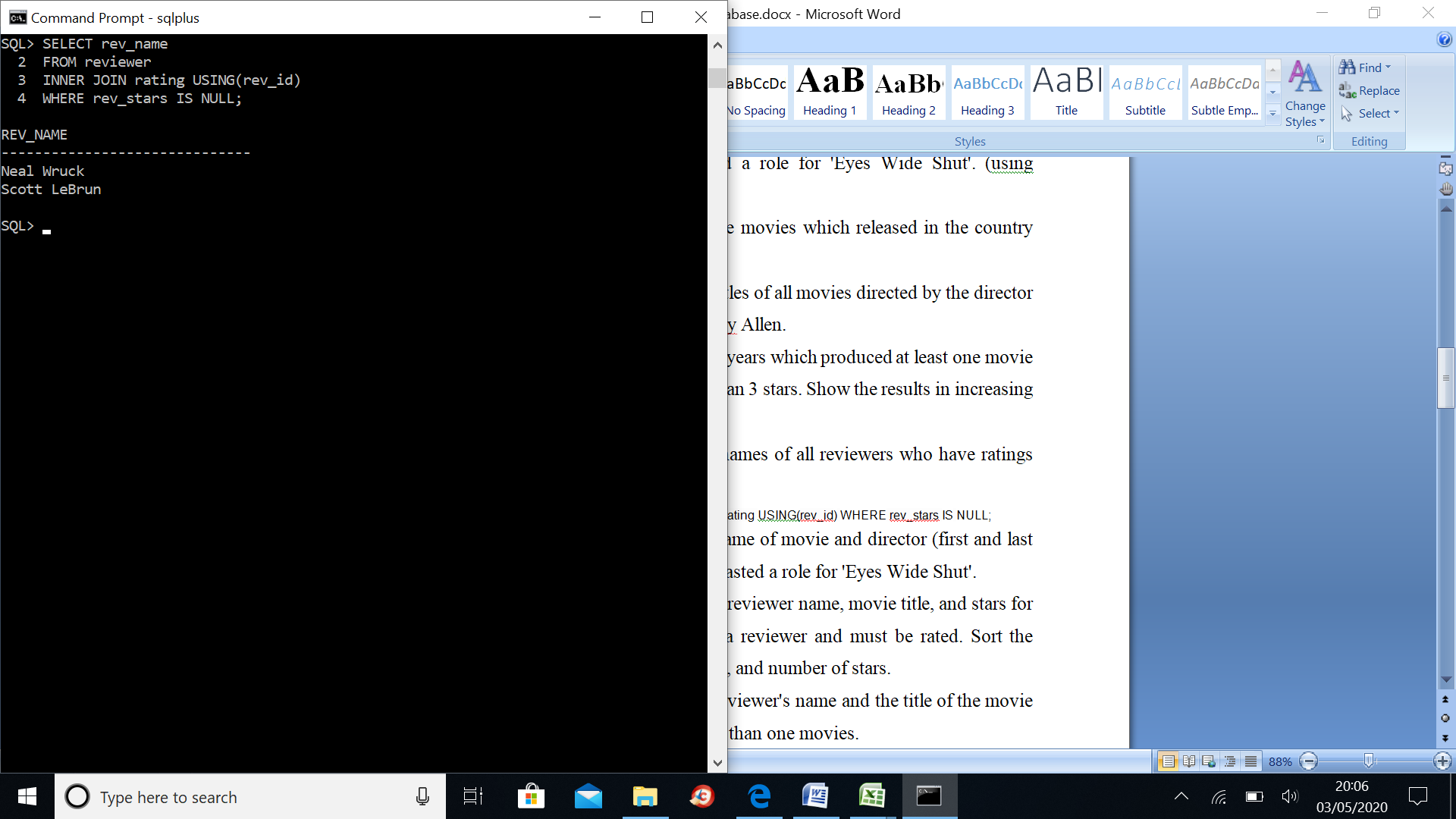
1. Write a query in SQL to find all the years which produced at least one movie and that received a rating of more than 3 stars. Show the results in increasing order.

SELECT DISTINCT mov\_year FROM movie WHERE mov\_id IN (SELECT mov\_id FROM rating WHERE rev\_stars>3) ORDER BY mov\_year;



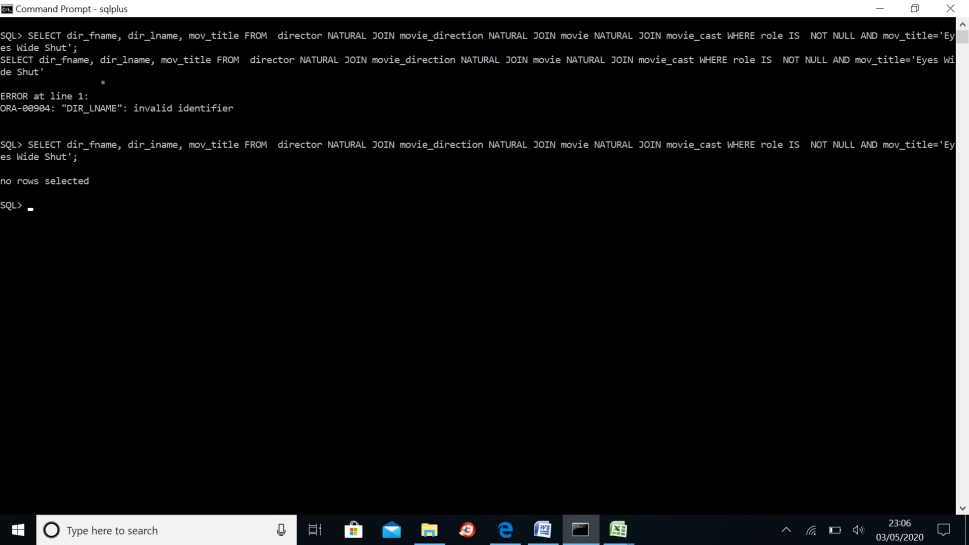
1. Write a query in SQL to find the names of all reviewers who have ratings with a NULL value.

SELECT rev\_name FROM reviewer INNER JOIN rating USING(rev\_id) WHERE rev\_stars IS NULL;



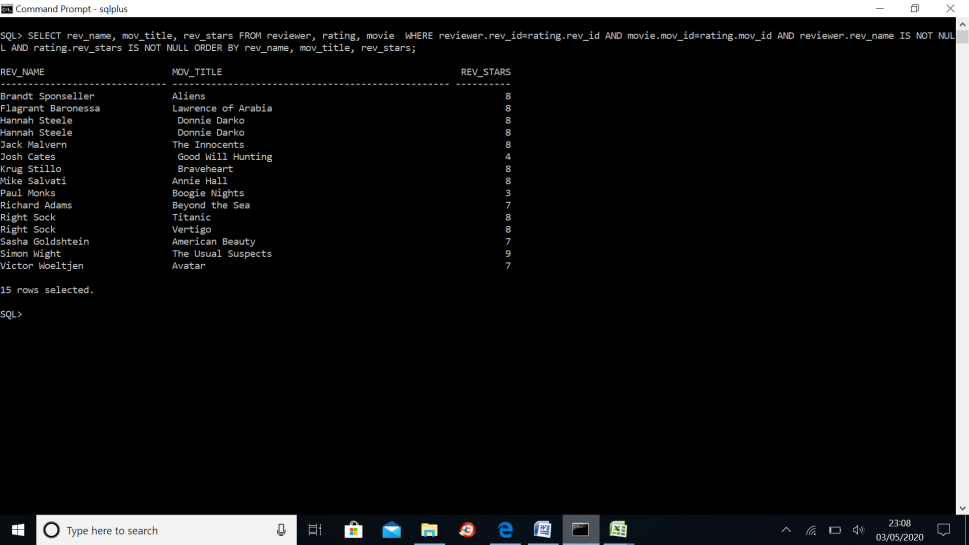
1. Write a query in SQL to find the name of movie and director (first and last names) who directed a movie that casted a role for 'Eyes Wide Shut'.

SELECT dir\_fname, dir\_lname, mov\_title FROM director NATURAL JOIN movie\_direction NATURAL JOIN movie NATURAL JOIN movie\_cast WHERE role IS NOT NULL AND mov\_title='Eyes Wide Shut';



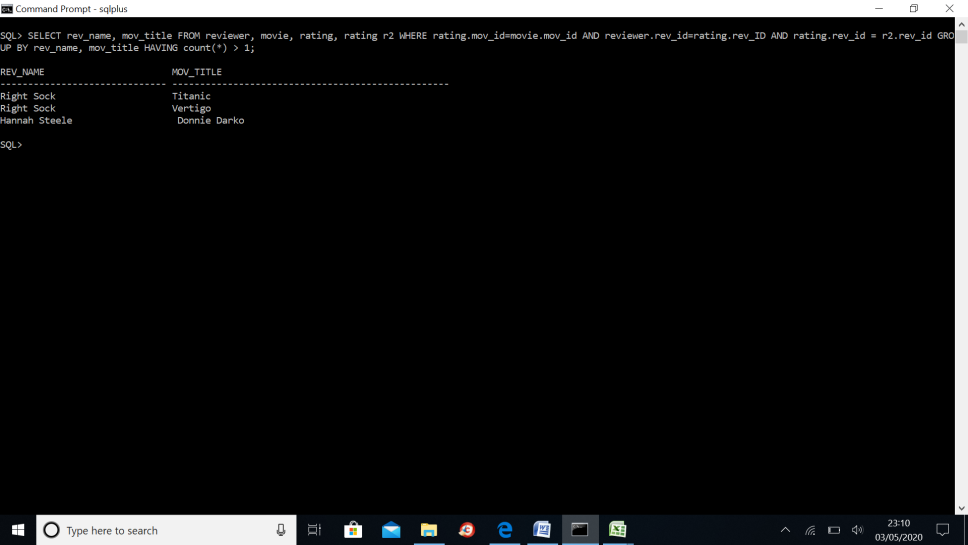
1. Write a query in SQL to return the reviewer name, movie title, and stars for those movies which reviewed by a reviewer and must be rated. Sort the result by reviewer name, movie title, and number of stars.

SELECT rev\_name, mov\_title, rev\_stars FROM reviewer, rating, movie WHERE reviewer.rev\_id=rating.rev\_id AND movie.mov\_id=rating.mov\_id AND reviewer.rev\_name IS NOT NULL AND rating.rev\_stars IS NOT NULL ORDER BY rev\_name, mov\_title, rev\_stars;



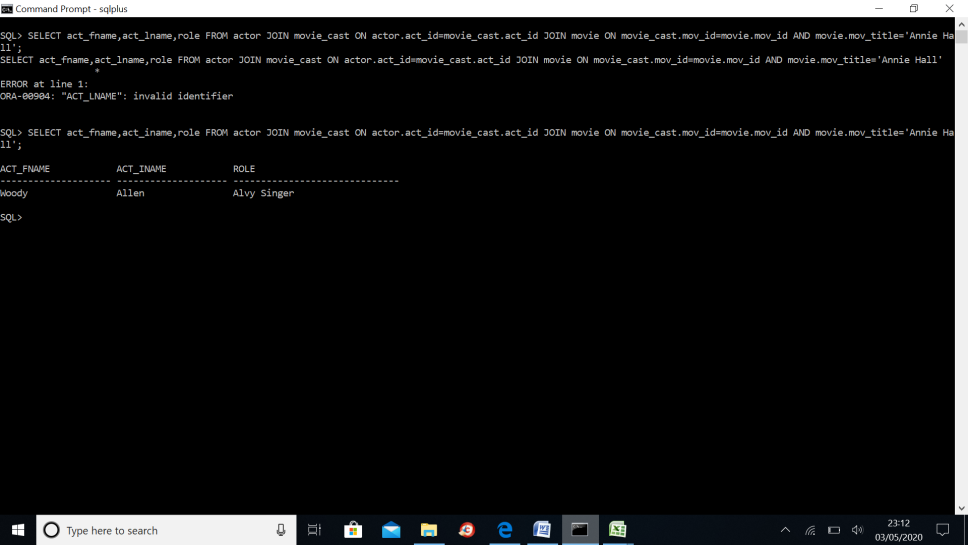
1. Write a query in SQL to find the reviewer's name and the title of the movie for those reviewers who rated more than one movies.

SELECT rev\_name, mov\_title FROM reviewer, movie, rating, rating r2 WHERE rating.mov\_id=movie.mov\_id AND reviewer.rev\_id=rating.rev\_ID AND rating.rev\_id = r2.rev\_id GROUP BY rev\_name, mov\_title HAVING count(\*) > 1;

~~~~

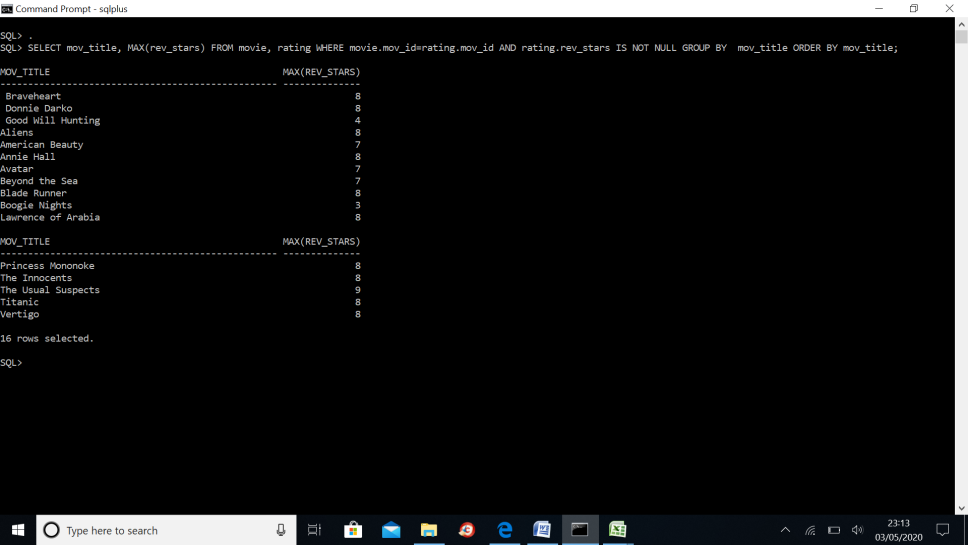
1. Write a query in SQL to list the first and last names of all the actors who were cast in the movie 'Annie Hall', and the roles they played in that production.

SELECT act\_fname,act\_lname,role FROM actor JOIN movie\_cast ON actor.act\_id=movie\_cast.act\_id JOIN movie ON movie\_cast.mov\_id=movie.mov\_id AND movie.mov\_title='Annie Hall';



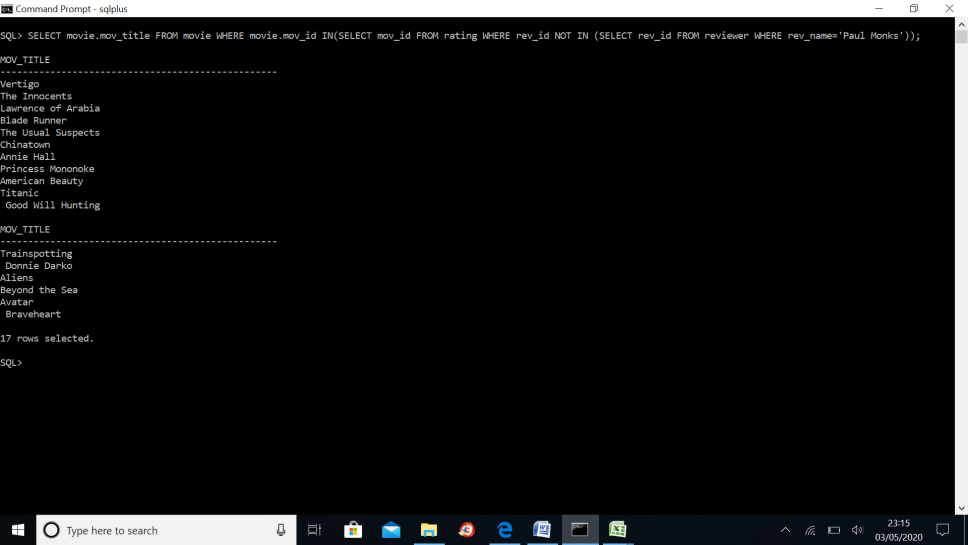
1. Write a query in SQL to find the movie title, and the highest number of stars that movie received and arranged the result according to the group of a movie and the movie title appear alphabetically in ascending order.

SELECT mov\_title, MAX(rev\_stars) FROM movie, rating WHERE movie.mov\_id=rating.mov\_id AND rating.rev\_stars IS NOT NULL GROUP BY mov\_title ORDER BY mov\_title;



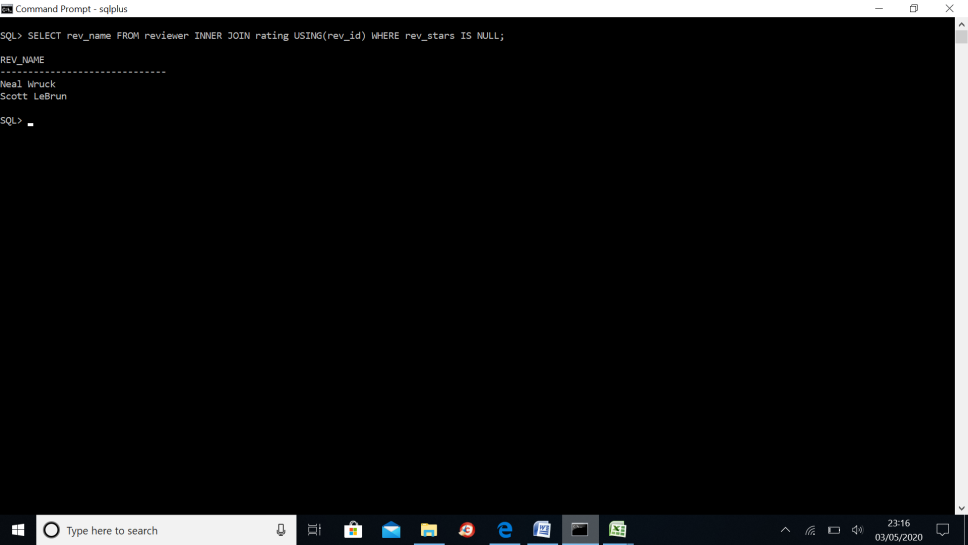
1. Write a query in SQL to find the titles of all movies which have been reviewed by anybody except by Paul Monks.

SELECT movie.mov\_title FROM movie WHERE movie.mov\_id IN(SELECT mov\_id FROM rating WHERE rev\_id NOT IN (SELECT rev\_id FROM reviewer WHERE rev\_name='Paul Monks'));



1. Write a query in SQL to find the name of all reviewers who have rated their ratings with a NULL value.

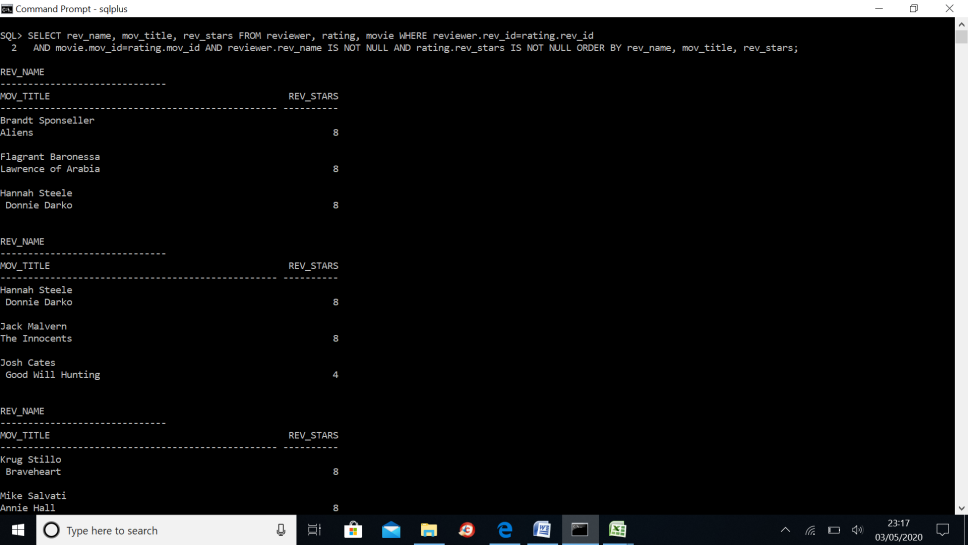
SELECT rev\_name FROM reviewer INNER JOIN rating USING(rev\_id) WHERE rev\_stars IS NULL;

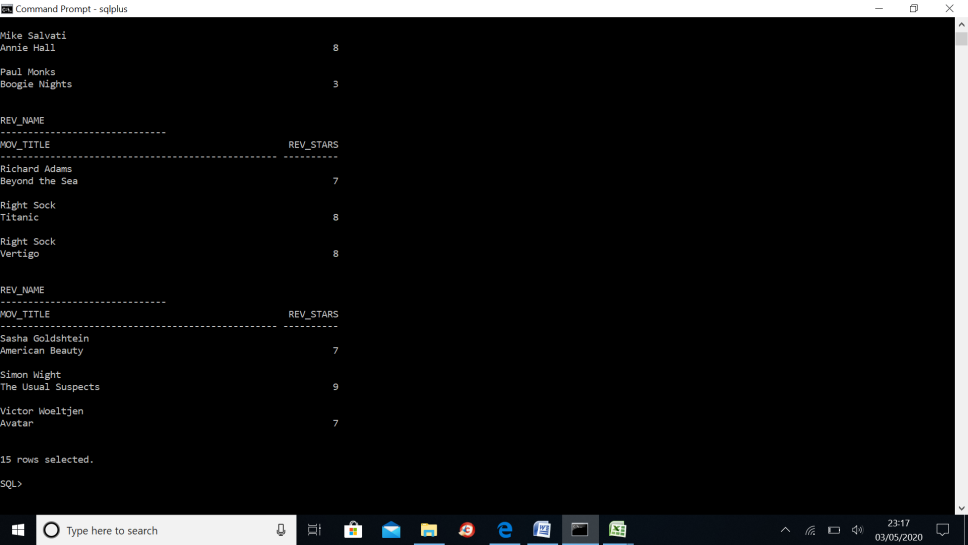


1. Write a query in SQL to return the reviewer name, movie title, and number of stars for those movies which rating is the lowest one.

SELECT rev\_name, mov\_title, rev\_stars FROM reviewer, rating, movie WHERE reviewer.rev\_id=rating.rev\_id

AND movie.mov\_id=rating.mov\_id AND reviewer.rev\_name IS NOT NULL AND rating.rev\_stars IS NOT NULL ORDER BY rev\_name, mov\_title, rev\_stars;

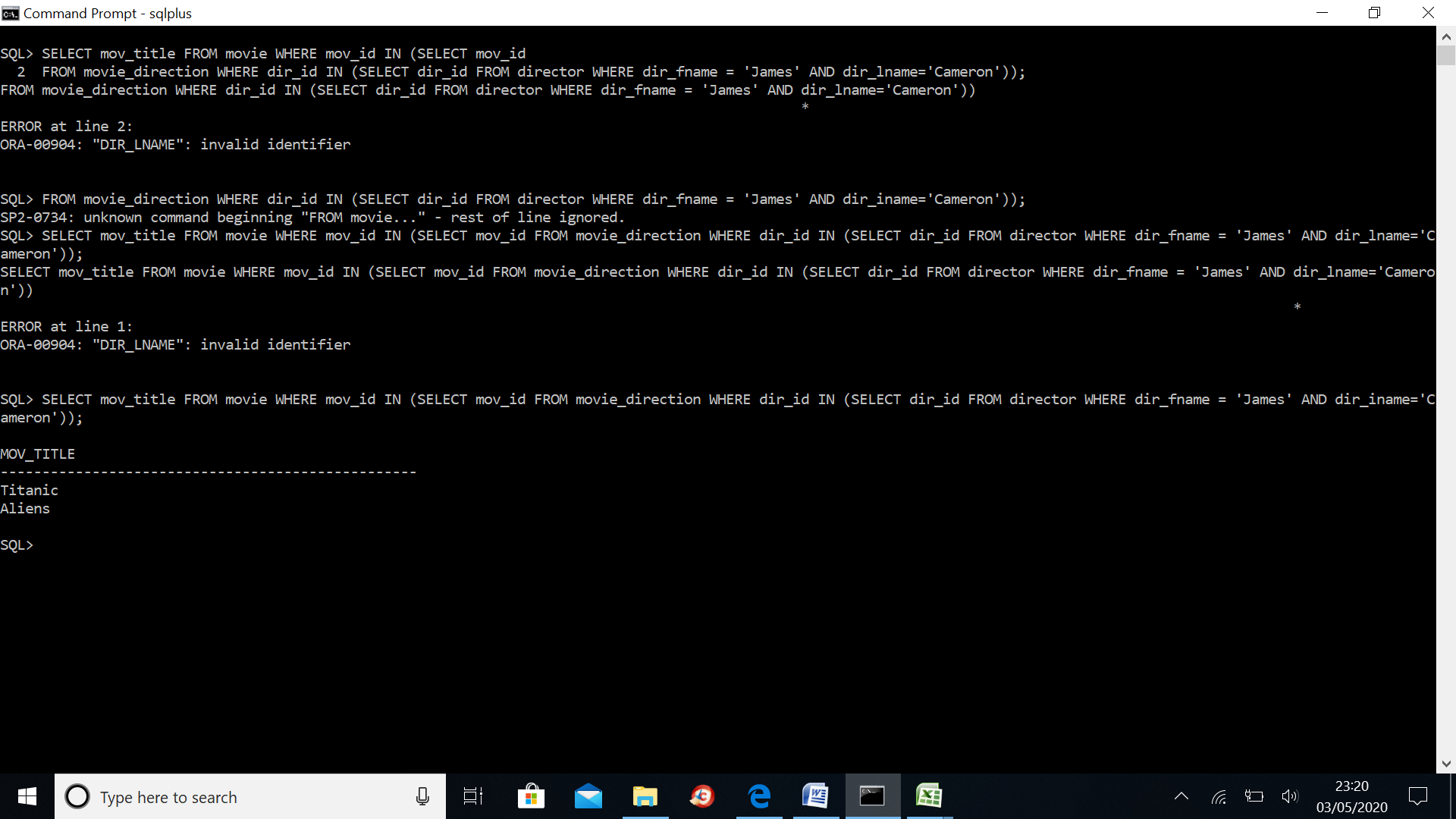




1. Write a query in SQL to find the titles of all movies directed by James Cameron.

SELECT mov\_title FROM movie WHERE mov\_id IN (SELECT mov\_id

FROM movie\_direction WHERE dir\_id IN (SELECT dir\_id FROM director WHERE dir\_fname = 'James' AND dir\_lname='Cameron'));

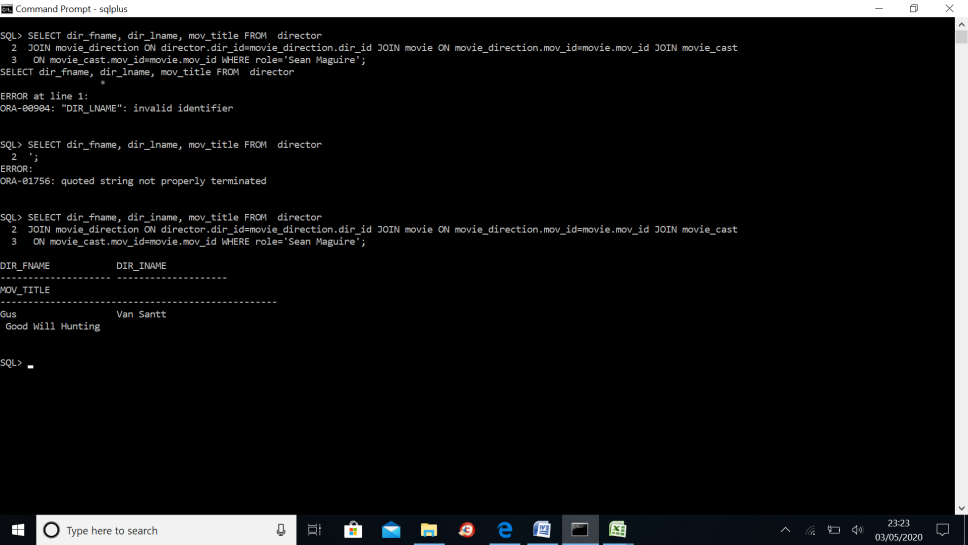


1. Write a query in SQL to find the name of movie and director (first and last names) who directed a movie that casted a role as Sean Maguire.

SELECT dir\_fname, dir\_lname, mov\_title FROM director

JOIN movie\_direction ON director.dir\_id=movie\_direction.dir\_id JOIN movie ON movie\_direction.mov\_id=movie.mov\_id JOIN movie\_cast

ON movie\_cast.mov\_id=movie.mov\_id WHERE role='Sean Maguire';



1. Write a query in SQL to list all the actors who have not acted in any movie between 1990 and 2000.

SELECT act\_fname, act\_lname, mov\_title, mov\_year

FROM actor

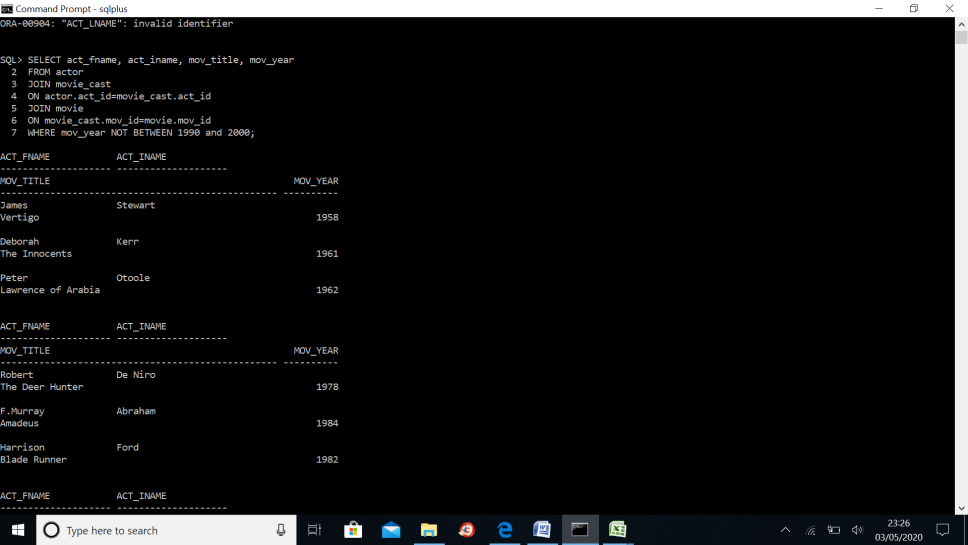
JOIN movie\_cast

ON actor.act\_id=movie\_cast.act\_id

JOIN movie

ON movie\_cast.mov\_id=movie.mov\_id

WHERE mov\_year NOT BETWEEN 1990 and 2000;





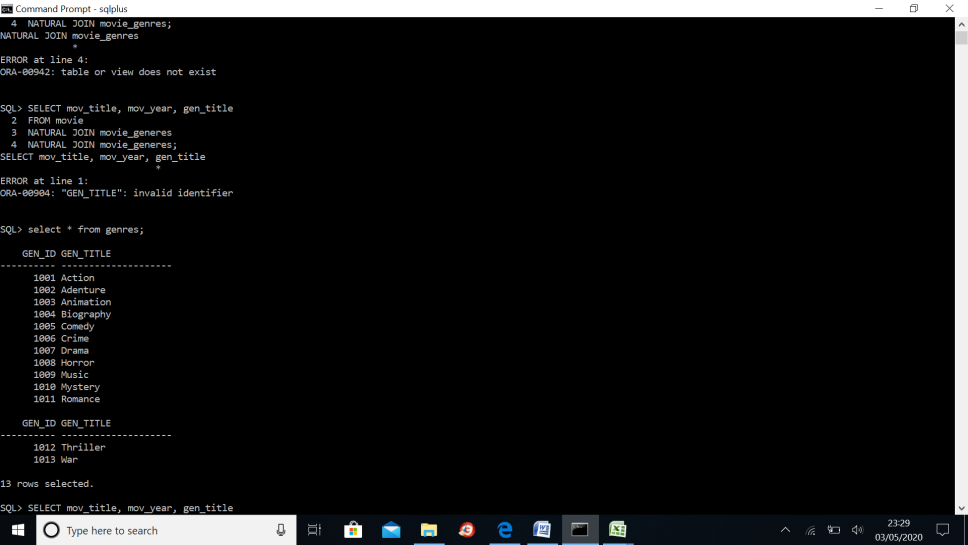
1. Write a query in SQL to list all the movies with year and genres.

SELECT mov\_title, mov\_year, gen\_title

FROM movie

NATURAL JOIN movie\_genres

NATURAL JOIN genres;



1. Write a query in SQL to list all the movies with year, genres, and name of the director.

SELECT mov\_title, mov\_year, gen\_title, dir\_fname, dir\_iname

FROM movie

NATURAL JOIN movie\_generes

NATURAL JOIN genres

NATURAL JOIN movie\_direction

NATURAL JOIN director;

1. Write a query in SQL to list all the movies with title, year, date of release, movie duration, and first and last name of the director which released before 24 august1989, and sort the result set according to release date from highest date to lowest.

SELECT movie.mov\_title, mov\_year, mov\_dt\_rel,

mov\_time,dir\_fname, dir\_iname

FROM movie

JOIN movie\_direction

ON movie.mov\_id = movie\_direction.mov\_id

JOIN director

ON movie\_direction.dir\_id=director.dir\_id

WHERE mov\_dt\_rel <'24/08/1989'

ORDER BY mov\_dt\_rel desc;

1. Write a query in SQL to compute a report which contain the genres of those movies with their average time and number of movies for each genres.

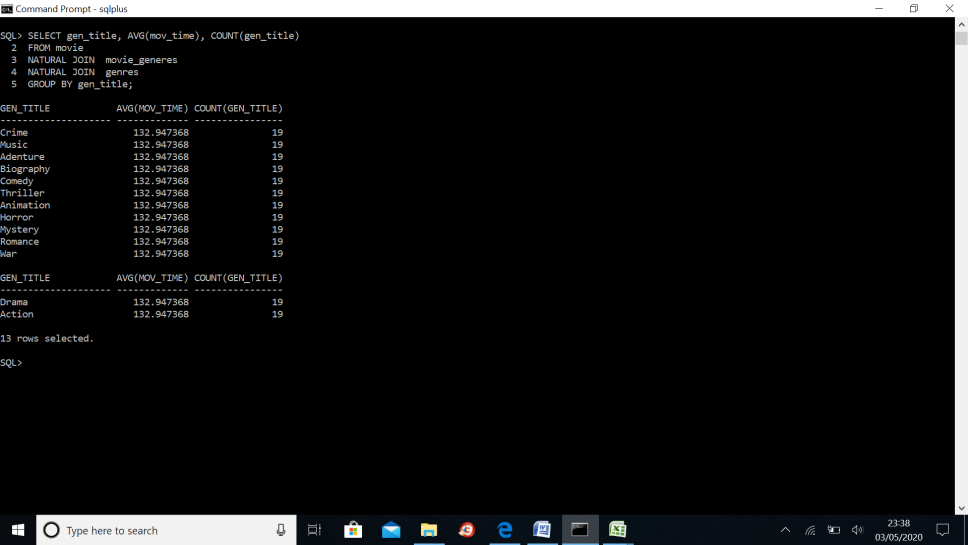
SELECT gen\_title, AVG(mov\_time), COUNT(gen\_title)

FROM movie

NATURAL JOIN movie\_generes

NATURAL JOIN genres

GROUP BY gen\_title;



1. Write a query in SQL to find all the years which produced a movie that received a rating of 3 or 4, and sort the result in increasing order.

SELECT DISTINCT mov\_year

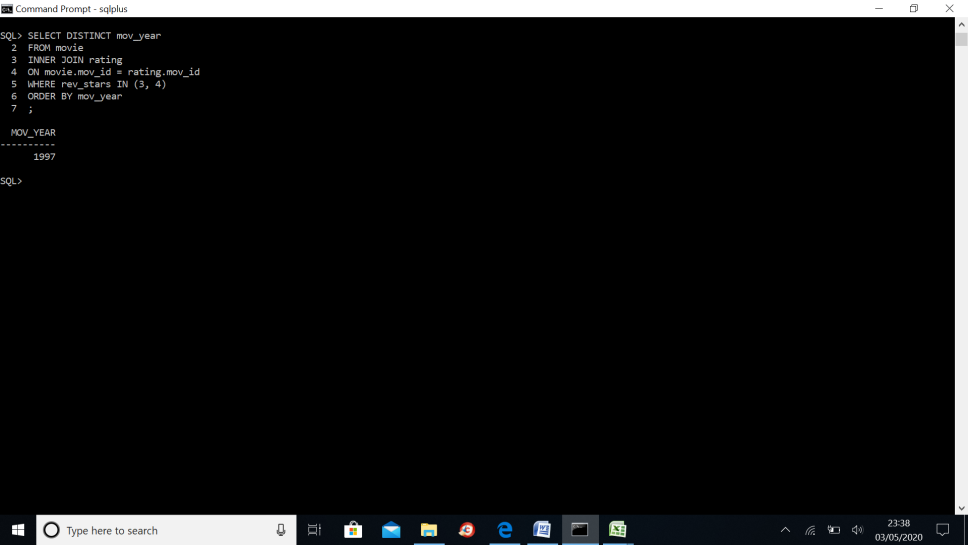
FROM movie

INNER JOIN rating

ON movie.mov\_id = rating.mov\_id

WHERE rev\_stars IN (3, 4)

ORDER BY mov\_year;



1. Write a query in SQL to return the reviewer name, movie title, and stars in an order that reviewer name will come first, then by movie title, and lastly by number of stars.

SELECT rev\_name, mov\_title, rev\_stars

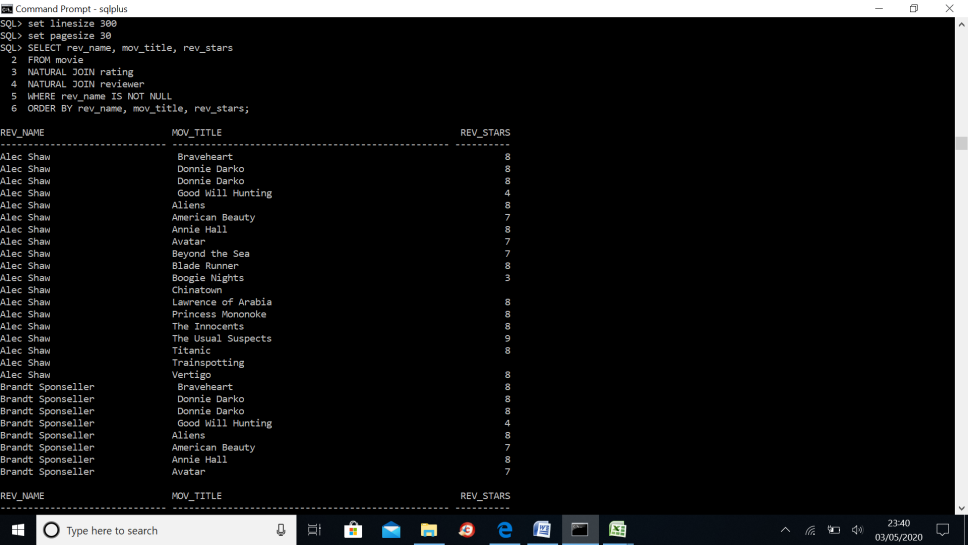
FROM movie

NATURAL JOIN rating

NATURAL JOIN reviewer

WHERE rev\_name IS NOT NULL

ORDER BY rev\_name, mov\_title, rev\_stars;



1. Write a query in SQL to find movie title and number of stars for each movie that has at least one rating and find the highest number of stars that movie received and sort the result by movie title.

SELECT mov\_title, MAX(rev\_stars)

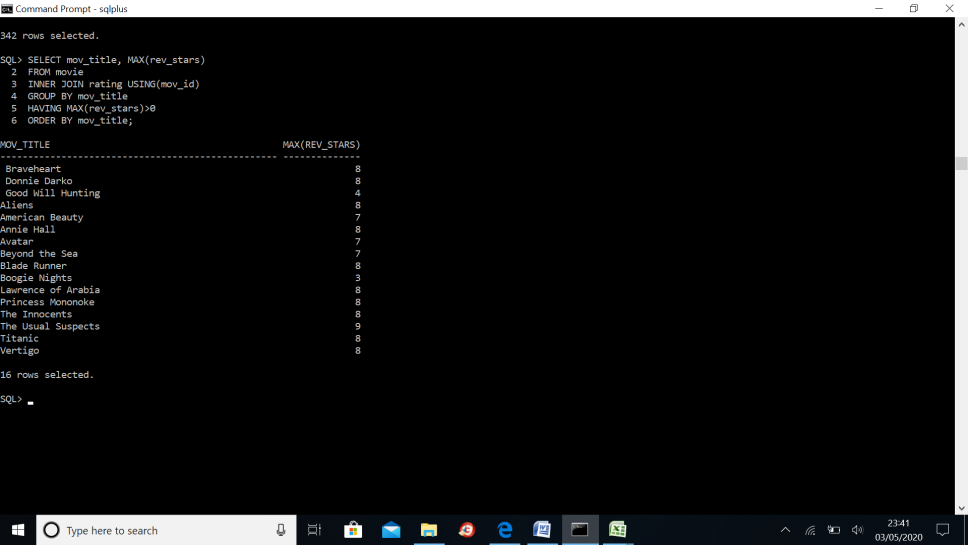
FROM movie

INNER JOIN rating USING(mov\_id)

GROUP BY mov\_title

HAVING MAX(rev\_stars)>0

ORDER BY mov\_title;



1. Write a query in SQL to find the movie title, actor first and last name, and the role for those movies where one or more actors acted in two or more movies.

SELECT mov\_title, act\_fname, act\_iname, role

FROM movie

JOIN movie\_cast

ON movie\_cast.mov\_id=movie.mov\_id

JOIN actor

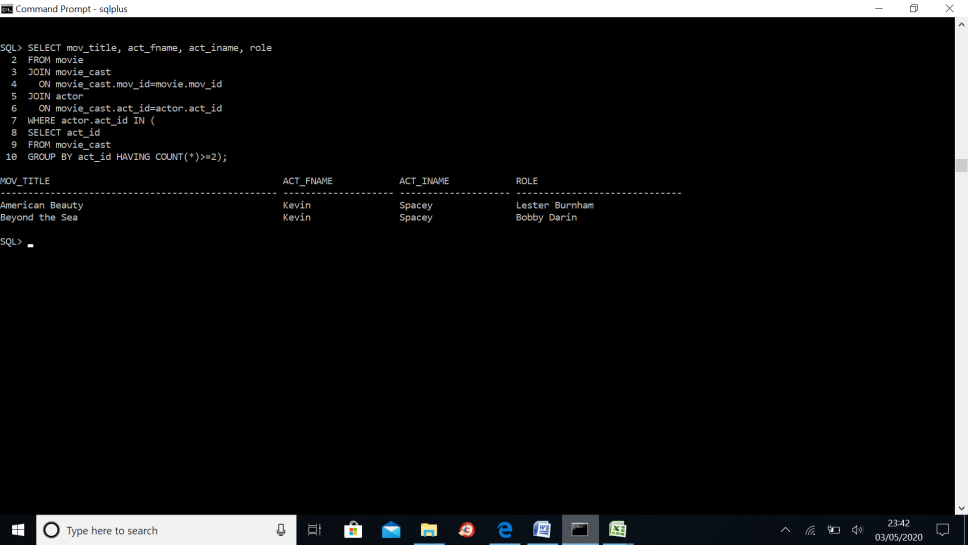
ON movie\_cast.act\_id=actor.act\_id

WHERE actor.act\_id IN (

SELECT act\_id

FROM movie\_cast

GROUP BY act\_id HAVING COUNT(\*)>=2);



1. Write a query in SQL to find the first and last name of a director and the movie he or she directed, and the actress appeared which first name was Claire and last name was Danes along with her role in that movie.

SELECT dir\_fname, dir\_iname, mov\_title, act\_fname, act\_iname, role

FROM actor

JOIN movie\_cast

ON actor.act\_id=movie\_cast.act\_id

JOIN movie\_direction

ON movie\_cast.mov\_id=movie\_direction.mov\_id

JOIN director

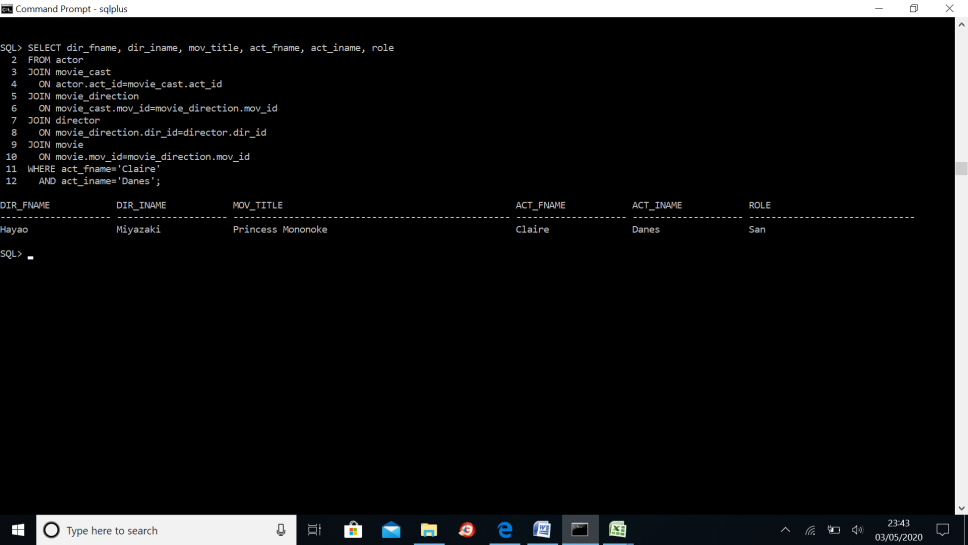
ON movie\_direction.dir\_id=director.dir\_id

JOIN movie

ON movie.mov\_id=movie\_direction.mov\_id

WHERE act\_fname='Claire'

AND act\_iname='Danes';



1. Write a query in SQL to find the first and last name of an actor with their role in the movie which was also directed by themselves.

SELECT act\_fname, act\_iname, mov\_title, role

FROM actor

JOIN movie\_cast

ON actor.act\_id=movie\_cast.act\_id

JOIN movie\_direction

ON movie\_cast.mov\_id=movie\_direction.mov\_id

JOIN director

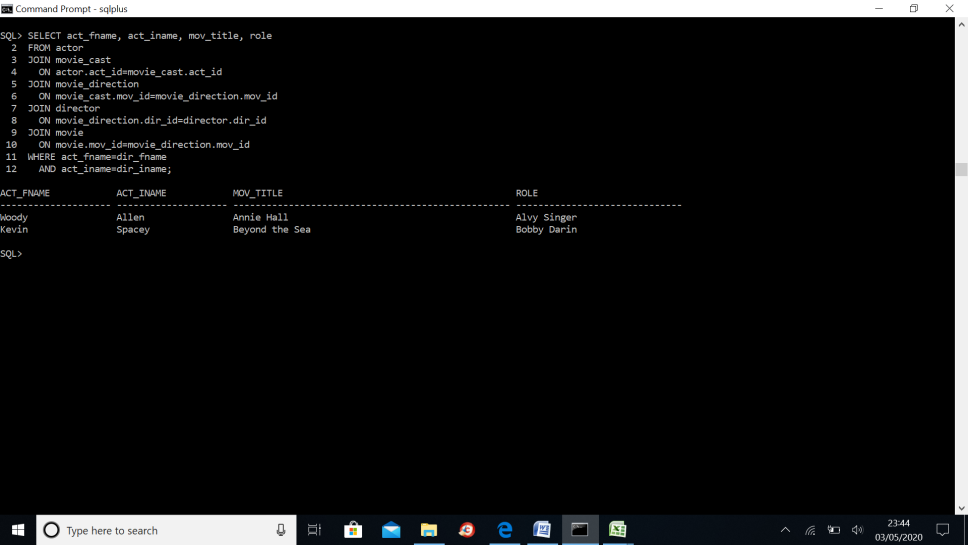
ON movie\_direction.dir\_id=director.dir\_id

JOIN movie

ON movie.mov\_id=movie\_direction.mov\_id

WHERE act\_fname=dir\_fname

AND act\_iname=dir\_iname;



1. Write a query in SQL to find the cast list for the movie Chinatown.

SELECT a.act\_fname, a.act\_iname

FROM

movie\_cast c

JOIN actor a ON

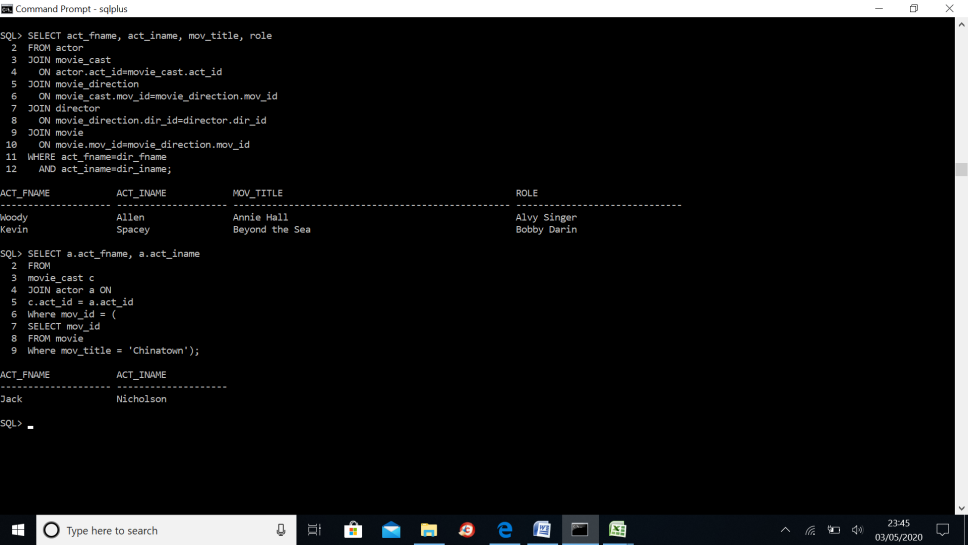
c.act\_id = a.act\_id

Where mov\_id = (

SELECT mov\_id

FROM movie

Where mov\_title = 'Chinatown');



1. Write a query in SQL to find the movie in which the actor appeared whose first and last name are 'Harrison' and 'Ford'.

SELECT m.mov\_title

FROM movie m

JOIN movie\_cast c

ON m.mov\_id = c.mov\_id

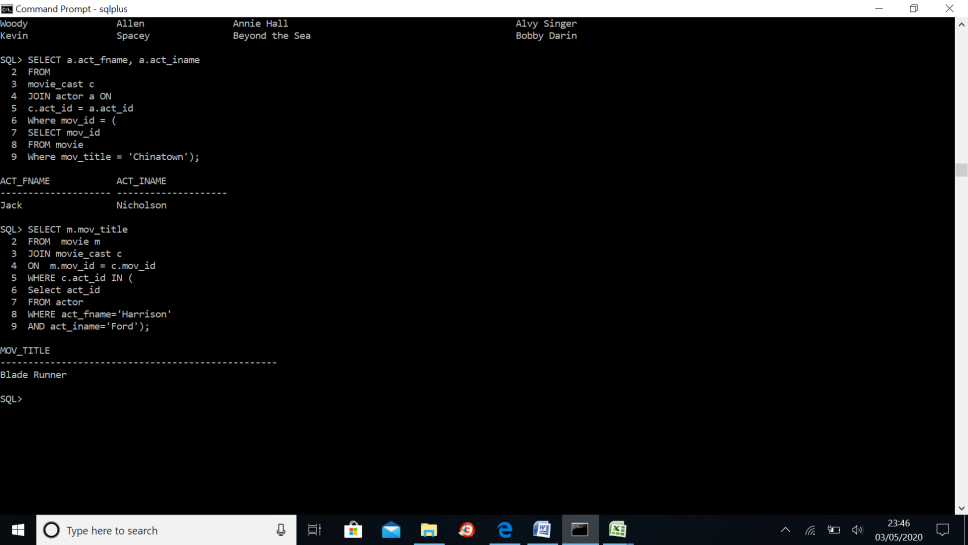
WHERE c.act\_id IN (

Select act\_id

FROM actor

WHERE act\_fname='Harrison'

AND act\_iname='Ford');



1. Write a query in SQL to generate a report which shows the year when most of the Mystery movies produces, and number of movies and their average rating.

SELECT mov\_year,gen\_title,count(gen\_title), avg(rev\_stars)

FROM movie

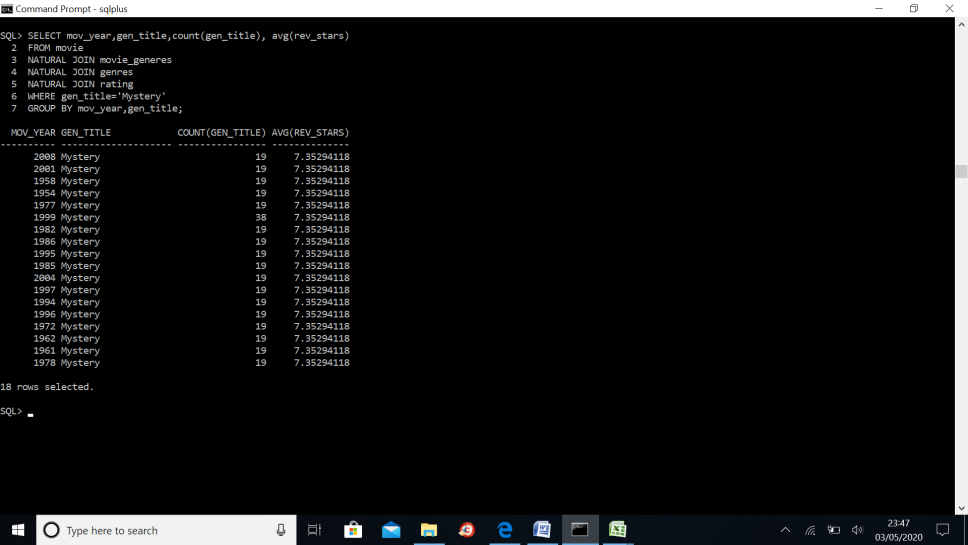
NATURAL JOIN movie\_generes

NATURAL JOIN genres

NATURAL JOIN rating

WHERE gen\_title='Mystery'

GROUP BY mov\_year,gen\_title;



1. Write a query in SQL to generate a report which contain the columns movie title, name of the female actor, year of the movie, role, movie genres, the director, date of release, and rating of that movie.

SELECT mov\_title, act\_fname, act\_iname,

mov\_year, role, gen\_title, dir\_fname, dir\_iname,

mov\_dt\_rel, rev\_stars

FROM movie

NATURAL JOIN movie\_cast

NATURAL JOIN actor

NATURAL JOIN movie\_generes

NATURAL JOIN genres

NATURAL JOIN movie\_direction

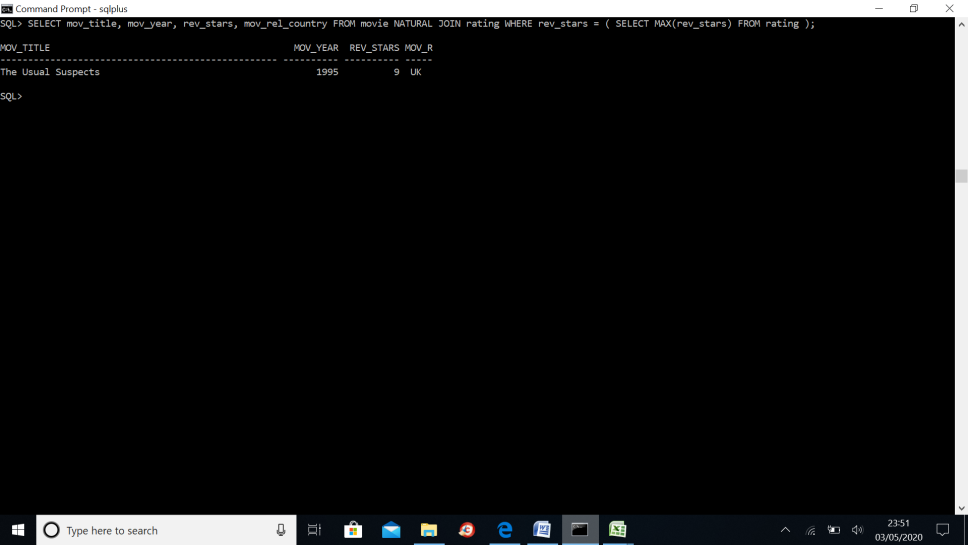
NATURAL JOIN director

NATURAL JOIN rating

WHERE act\_gender='F';

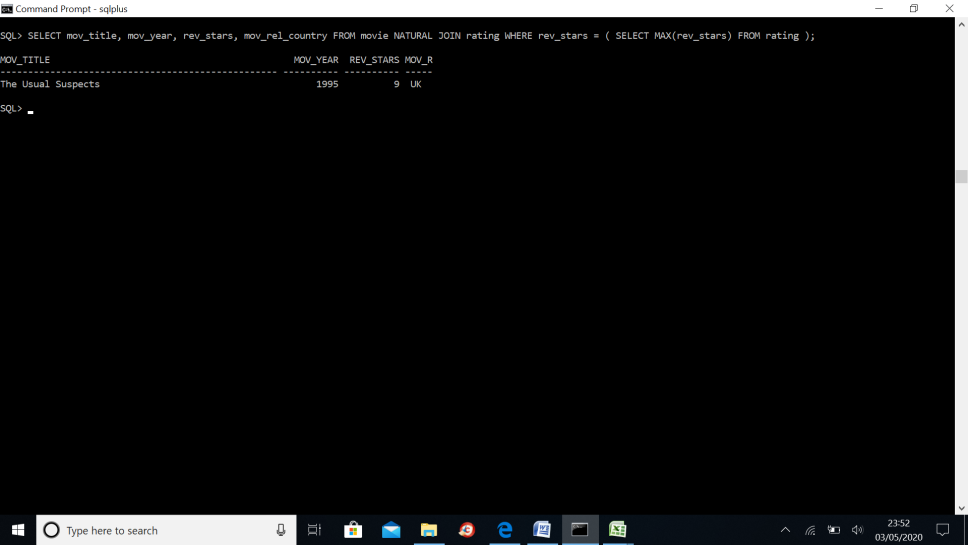
1. Write a query in SQL to find the highest-rated Mystery movie, and report the title, year, and rating.

SELECT mov\_title, mov\_year, rev\_stars, mov\_rel\_country FROM movie NATURAL JOIN rating WHERE rev\_stars = ( SELECT MAX(rev\_stars) FROM rating );



1. Write a query in SQL to find the highest-rated movie, and report its title, year, rating, and releasing country.

SELECT mov\_title, mov\_year, rev\_stars, mov\_rel\_country FROM movie NATURAL JOIN rating WHERE rev\_stars = ( SELECT MAX(rev\_stars) FROM rating );



1. Write a query in SQL to find the cast list for the movie The Shawshank Redemption.



