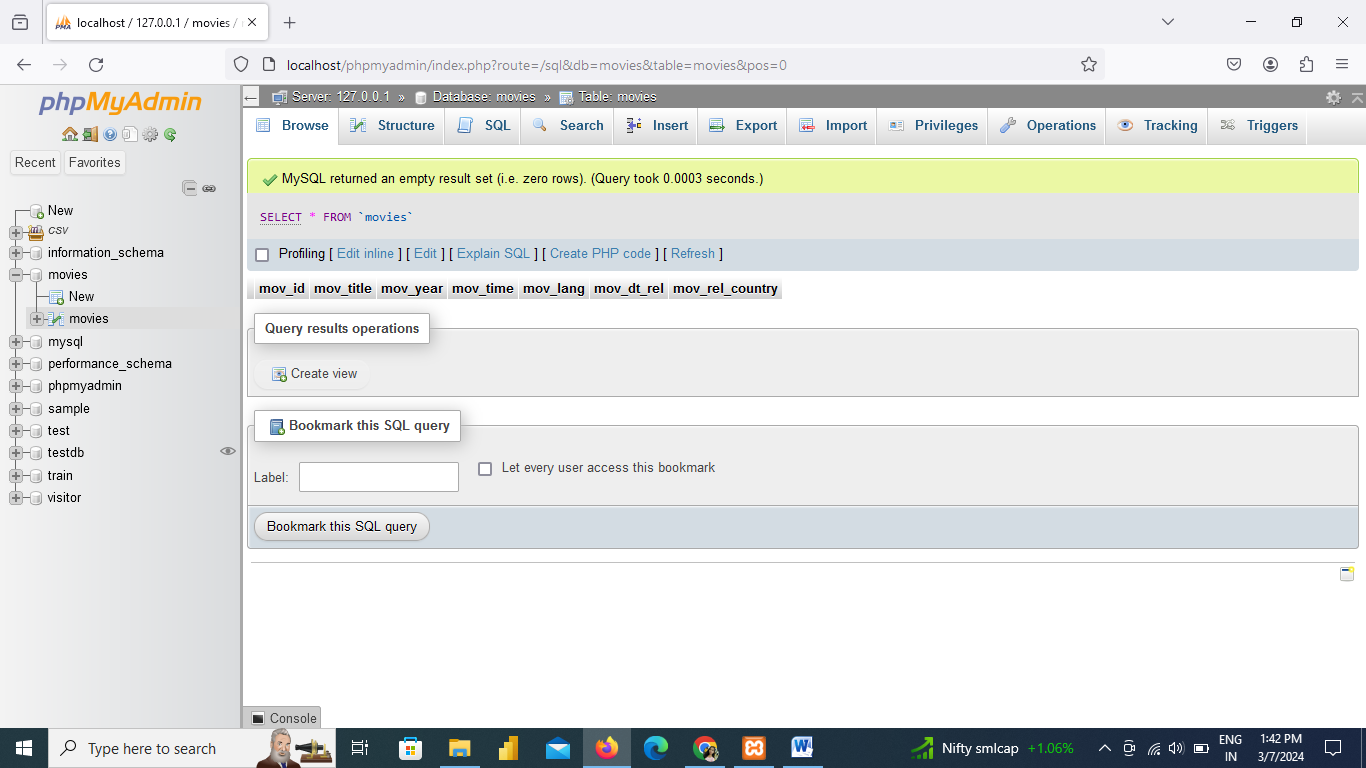
Query To Create Table using Sql Table in the movies Database.

Syntax:[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) Movies (mov\_id INTEGER(20), mov\_title VARCHAR(250), mov\_year INTEGER(20), mov\_time INTEGER(20), mov\_lang VARCHAR(250), mov\_dt\_rel DATE, mov\_rel\_country VARCHAR(100));

Output:



INSERT INTO movies (id, title, year, duration, language, release\_date, country)

VALUES

(901, 'Vertigo', 1958, 128, 'English', '1958-08-24', 'UK'),

(902, 'The Innocents', 1961, 100, 'English', '1962-02-19', 'SW'),

(903, 'Lawrence of Arabia', 1962, 216, 'English', '1962-12-11', 'UK'),

(904, 'The Deer Hunter', 1978, 183, 'English', '1979-03-08', 'UK'),

(905, 'Amadeus', 1984, 160, 'English', '1985-01-07', 'UK'),

(906, 'Blade Runner', 1982, 117, 'English', '1982-09-09', 'UK'),

(907, 'Eyes Wide Shut', 1999, 159, 'English', NULL, 'UK'),

(908, 'The Usual Suspects', 1995, 106, 'English', '1995-08-25', 'UK'),

(909, 'Chinatown', 1974, 130, 'English', '1974-08-09', 'UK'),

(910, 'Boogie Nights', 1997, 155, 'English', '1998-02-16', 'UK'),

(911, 'Annie Hall', 1977, 93, 'English', '1977-04-20', 'USA'),

(912, 'Princess Mononoke', 1997, 134, 'Japanese', '2001-10-19', 'UK'),

(913, 'The Shawshank Redemption', 1994, 142, 'English', '1995-02-17', 'UK'),

(914, 'American Beauty', 1999, 122, 'English', NULL, 'UK'),

(915, 'Titanic', 1997, 194, 'English', '1998-01-23', 'UK'),

(916, 'Good Will Hunting', 1997, 126, 'English', '1998-06-03', 'UK'),

(917, 'Deliverance', 1972, 109, 'English', '1982-10-05', 'UK'),

(918, 'Trainspotting', 1996, 94, 'English', '1996-02-23', 'UK'),

(919, 'The Prestige', 2006, 130, 'English', '2006-11-10', 'UK'),

(920, 'Donnie Darko', 2001, 113, 'English', NULL, 'UK'),

(921, 'Slumdog Millionaire', 2008, 120, 'English', '2009-01-09', 'UK'),

(922, 'Aliens', 1986, 137, 'English', '1986-08-29', 'UK'),

(923, 'Beyond the Sea', 2004, 118, 'English', '2004-11-26', 'UK'),

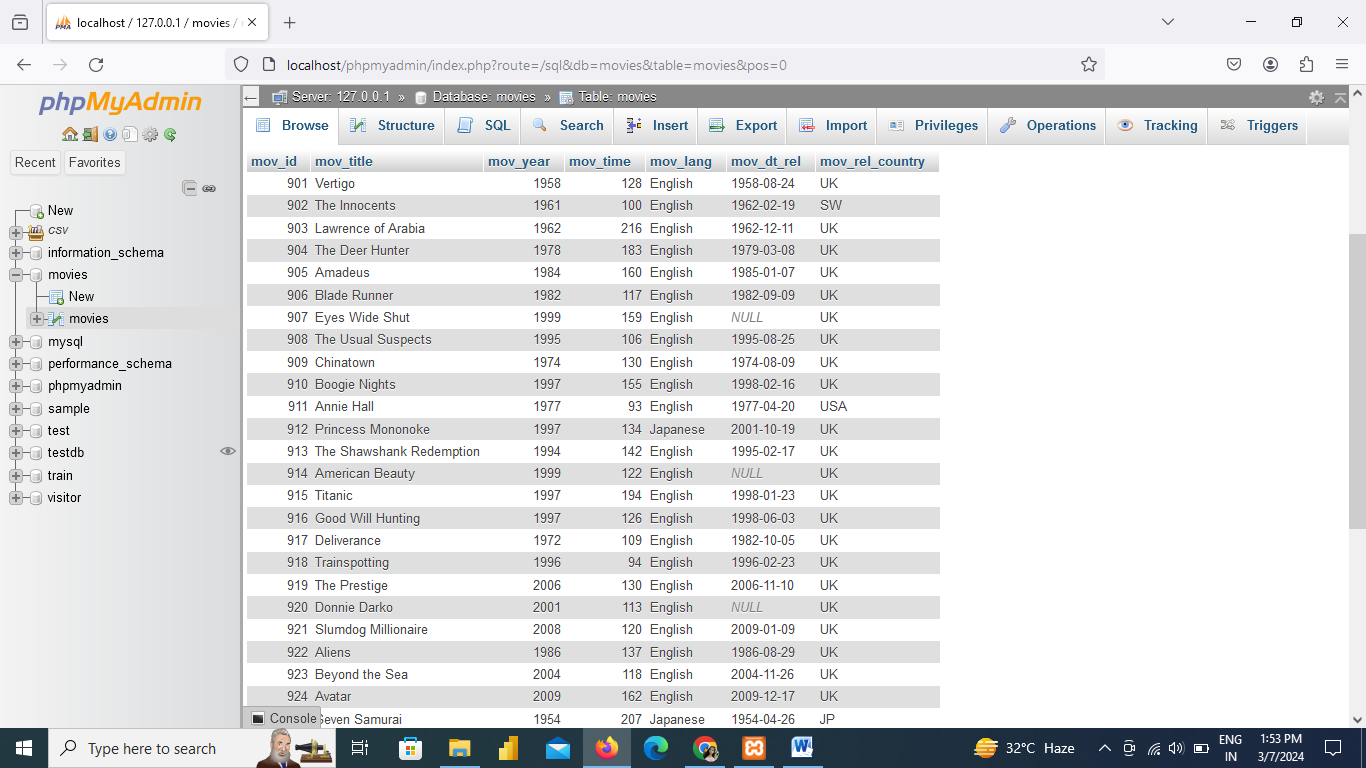
(924, 'Avatar', 2009, 162, 'English', '2009-12-17', 'UK'),

(926, 'Seven Samurai', 1954, 207, 'Japanese', '1954-04-26', 'JP'),

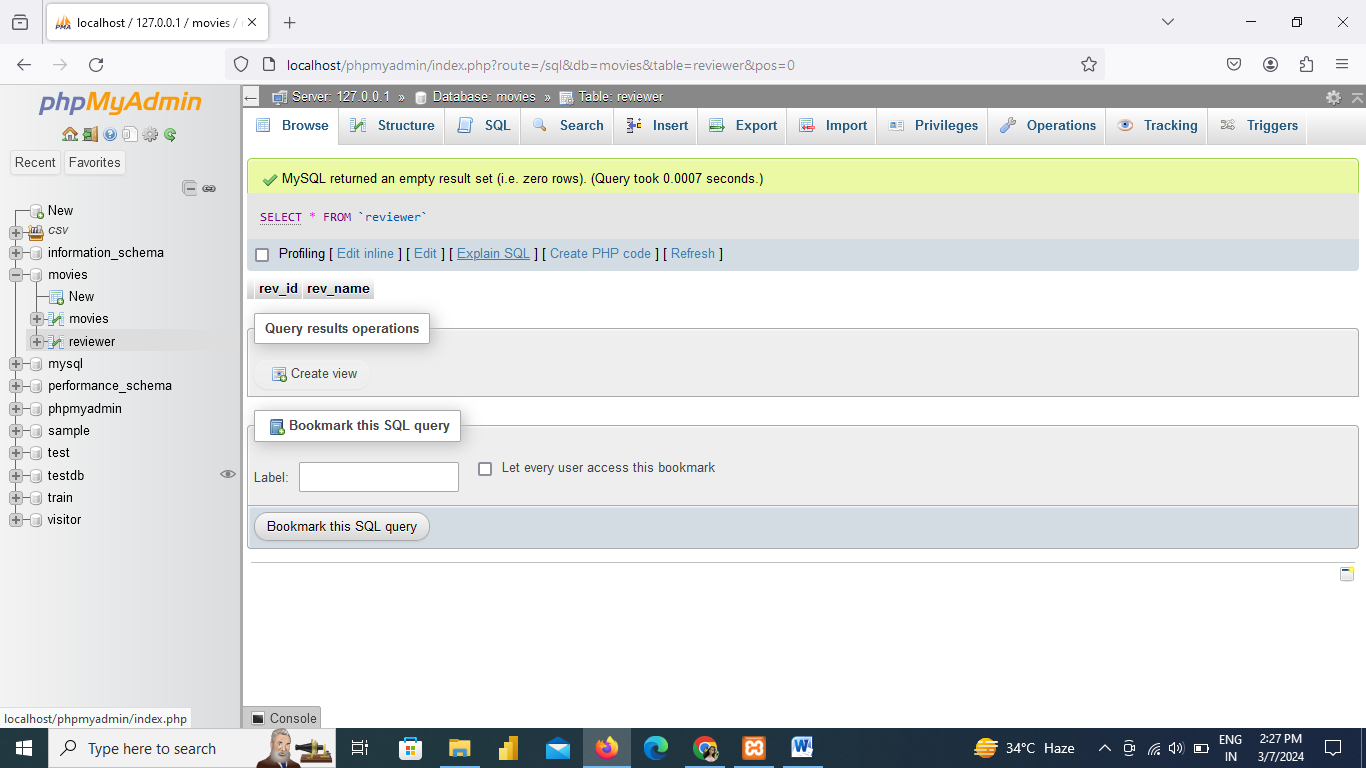
(927, 'Spirited Away', 2001, 125, 'Japanese', '2003-09-12', 'UK'),

(928, 'Back to the Future', 1985, 116, 'English', '1985-12-04', 'UK'),

(925, 'Braveheart', 1995, 178, 'English', '1995-09-08', 'UK');



CREATE TABLE reviewer (rev\_id INTEGER(20),rev\_name VARCHAR(255));



INSERT INTO reviewer (rev\_id, rev\_name)

VALUES

(9001, 'Righty Sock'),

(9002, 'Jack Malvern'),

(9003, 'Flagrant Baronessa'),

(9004, 'Alec Shaw'),

(9005, NULL), -- You need to decide how to handle a case where the name is not provided

(9006, 'Victor Woeltjen'),

(9007, 'Simon Wright'),

(9008, 'Neal Wruck'),

(9009, 'Paul Monks'),

(9010, 'Mike Salvati'),

(9011, NULL), -- Another case where the name is not provided

(9012, 'Wesley S. Walker'),

(9013, 'Sasha Goldshtein'),

(9014, 'Josh Cates'),

(9015, 'Krug Stillo'),

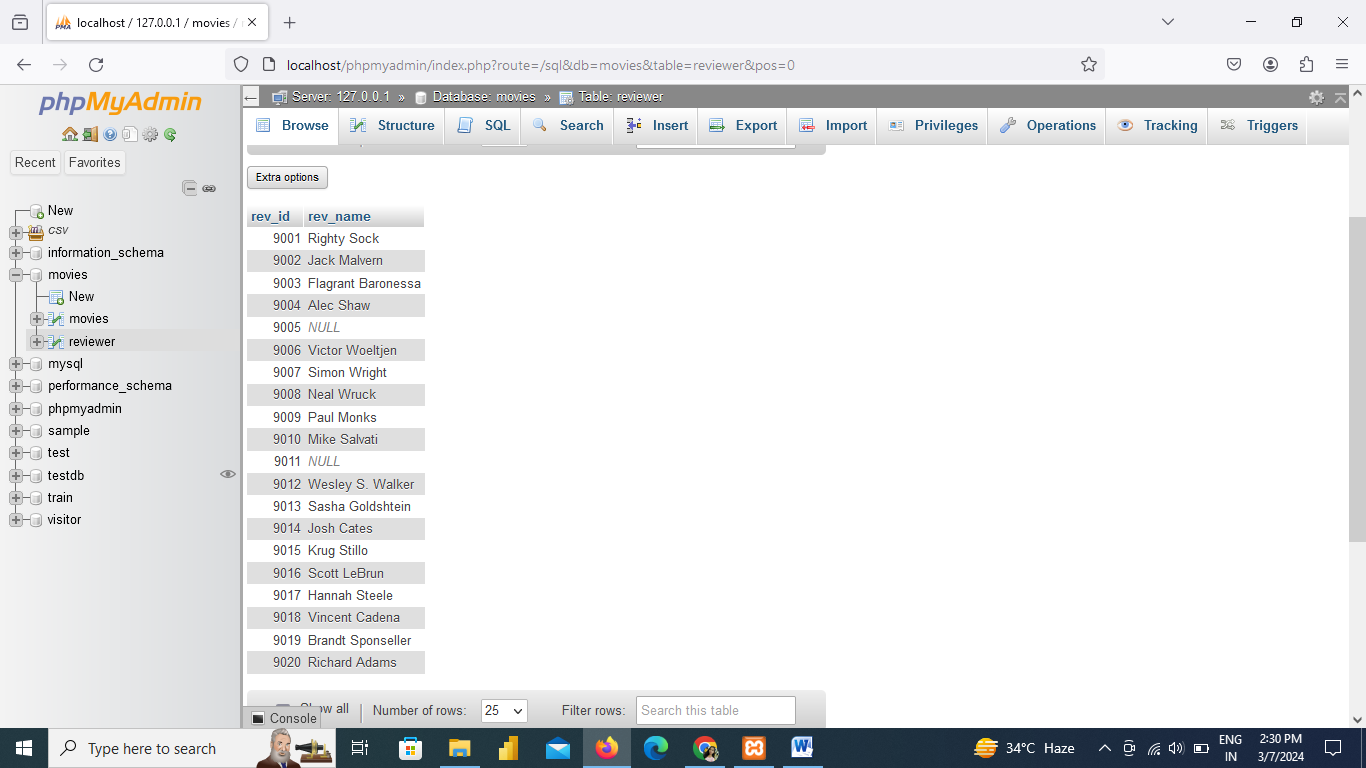
(9016, 'Scott LeBrun'),

(9017, 'Hannah Steele'),

(9018, 'Vincent Cadena'),

(9019, 'Brandt Sponseller'),

(9020, 'Richard Adams');

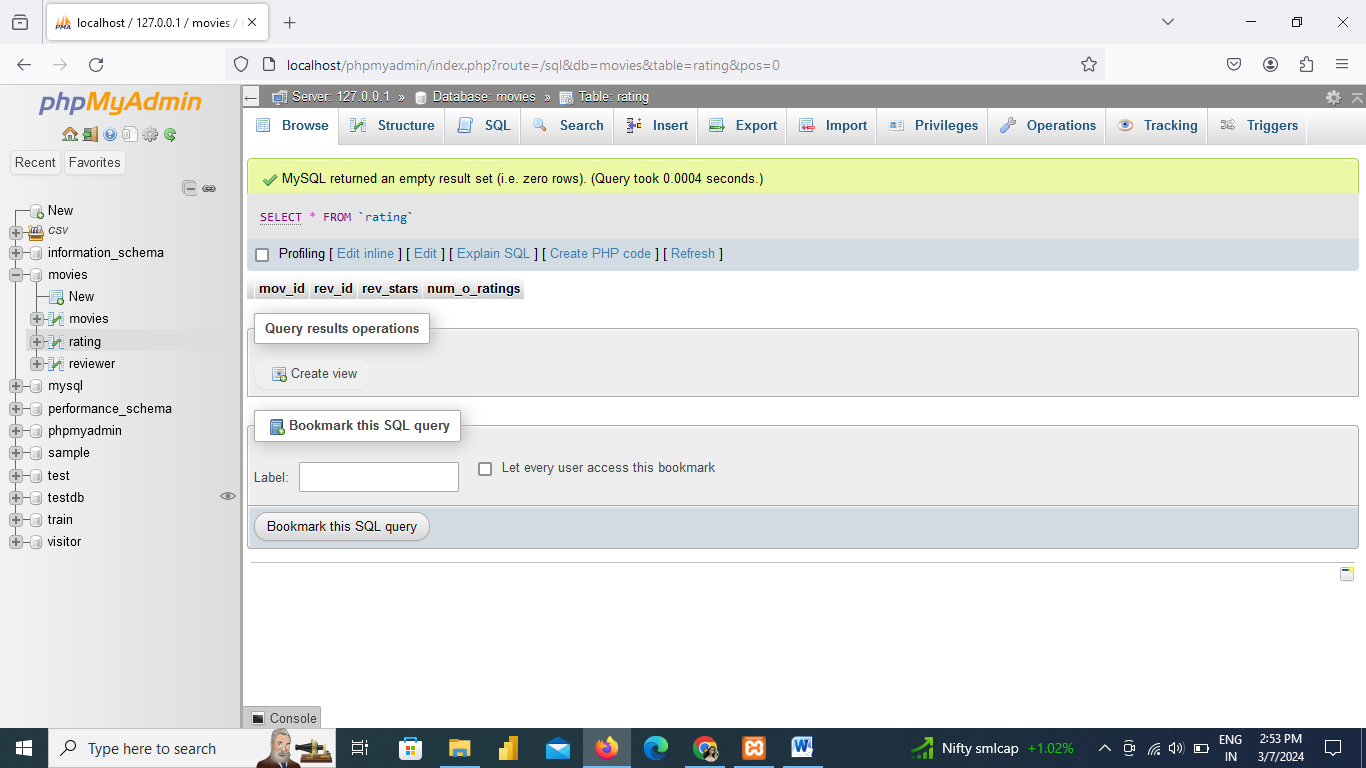


CREATE TABLE rating(mov\_id INTEGER(20),

rev\_id INTEGER(25),

rev\_stars VARCHAR(100),

num\_o\_ratings INTEGER(25));



INSERT INTO rating (mov\_id, rev\_id, rev\_stars, num\_o\_ratings)

VALUES

(901, 9001, 8.40, 263575),

(902, 9002, 7.90, 20207),

(903, 9003, 8.30, 202778),

(906, 9005, 8.20, 484746),

(924, 9006, 7.30, NULL),

(908, 9007, 8.60, 779489),

(909, 9008, NULL, 227235),

(910, 9009, 3.00, 195961),

(911, 9010, 8.10, 203875),

(912, 9011, 8.40, NULL),

(914, 9013, 7.00, 862618),

(915, 9001, 7.70, 830095),

(916, 9014, 4.00, 642132),

(925, 9015, 7.70, 81328),

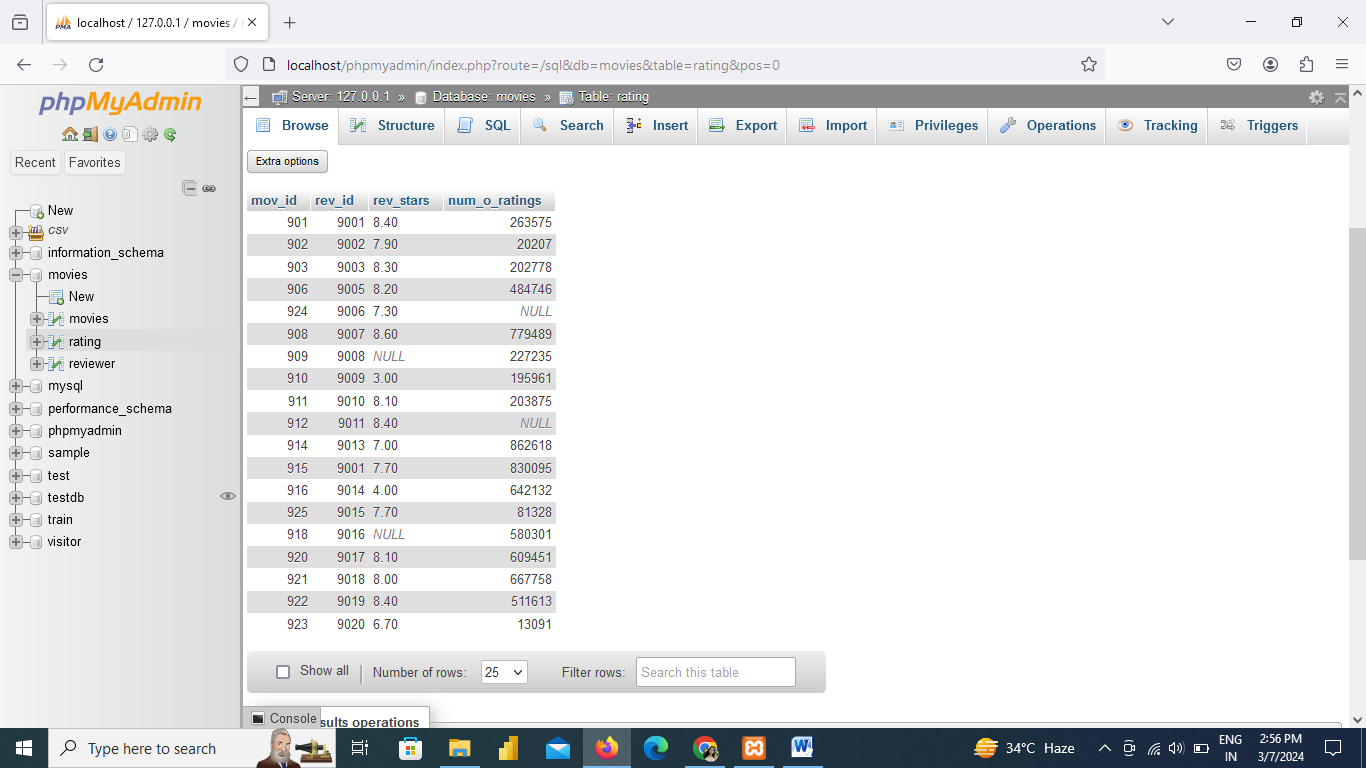
(918, 9016, NULL, 580301),

(920, 9017, 8.10, 609451),

(921, 9018, 8.00, 667758),

(922, 9019, 8.40, 511613),

(923, 9020, 6.70, 13091);

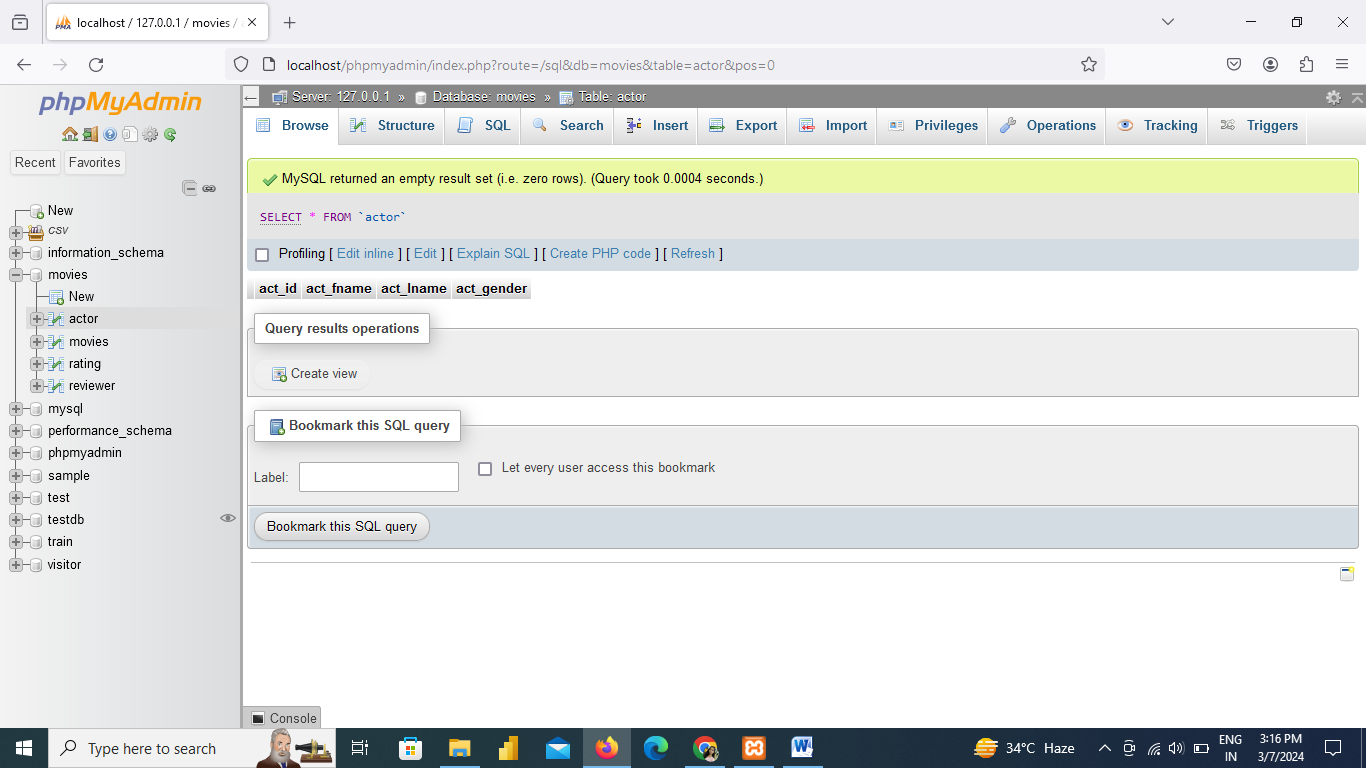


CREATE TABLE actor (act\_id INTEGER(20),

act\_fname VARCHAR(250),

act\_lname VARCHAR(250),

act\_gender VARCHAR(250));



INSERT INTO actor (act\_id, act\_fname, act\_lname, act\_gender)

VALUES

(101, 'James', 'Stewart', 'M'),

(102, 'Deborah', 'Kerr', 'F'),

(103, 'Peter', 'OToole', 'M'),

(104, 'Robert', 'De Niro', 'M'),

(105, 'F. Murray', 'Abraham', 'M'),

(106, 'Harrison', 'Ford', 'M'),

(107, 'Nicole', 'Kidman', 'F'),

(108, 'Stephen', 'Baldwin', 'M'),

(109, 'Jack', 'Nicholson', 'M'),

(110, 'Mark', 'Wahlberg', 'M'),

(111, 'Woody', 'Allen', 'M'),

(112, 'Claire', 'Danes', 'F'),

(113, 'Tim', 'Robbins', 'M'),

(114, 'Kevin', 'Spacey', 'M'),

(115, 'Kate', 'Winslet', 'F'),

(116, 'Robin', 'Williams', 'M'),

(117, 'Jon', 'Voight', 'M'),

(118, 'Ewan', 'McGregor', 'M'),

(119, 'Christian', 'Bale', 'M'),

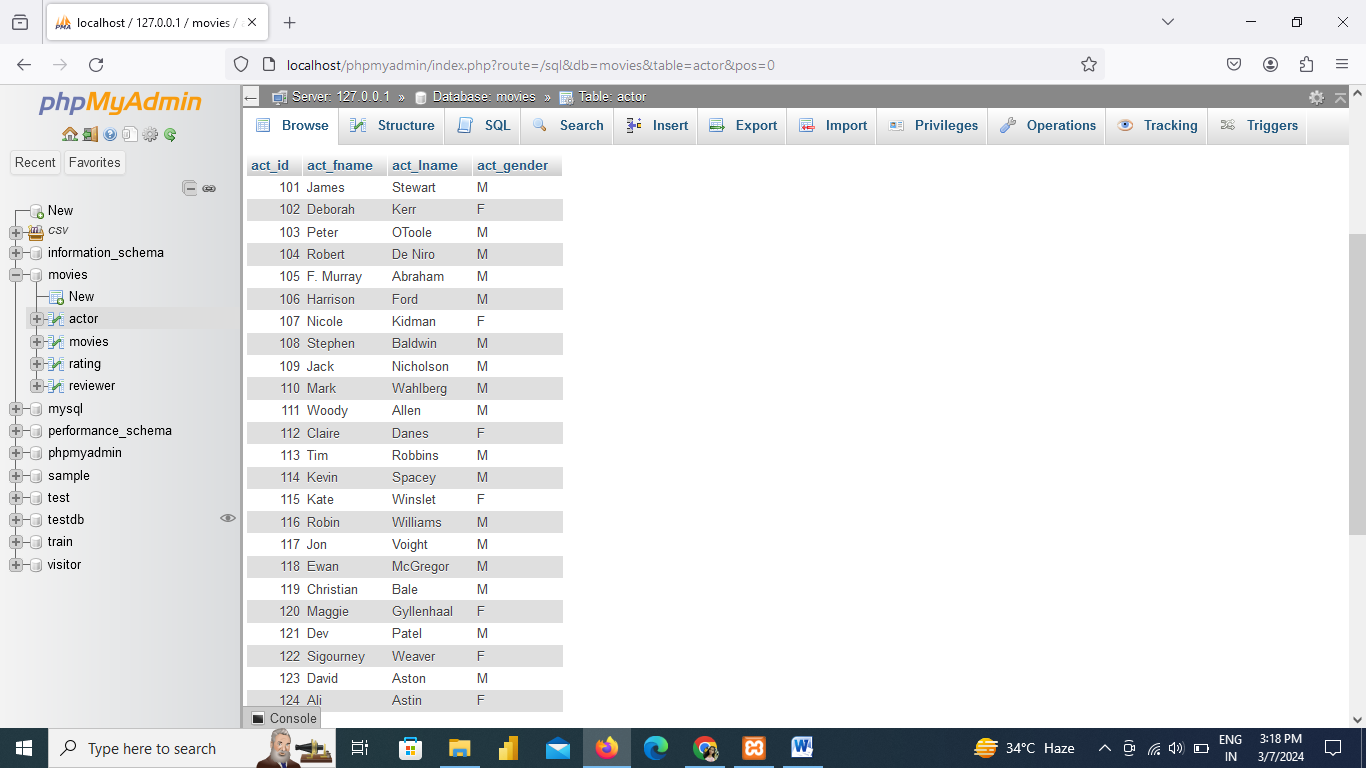
(120, 'Maggie', 'Gyllenhaal', 'F'),

(121, 'Dev', 'Patel', 'M'),

(122, 'Sigourney', 'Weaver', 'F'),

(123, 'David', 'Aston', 'M'),

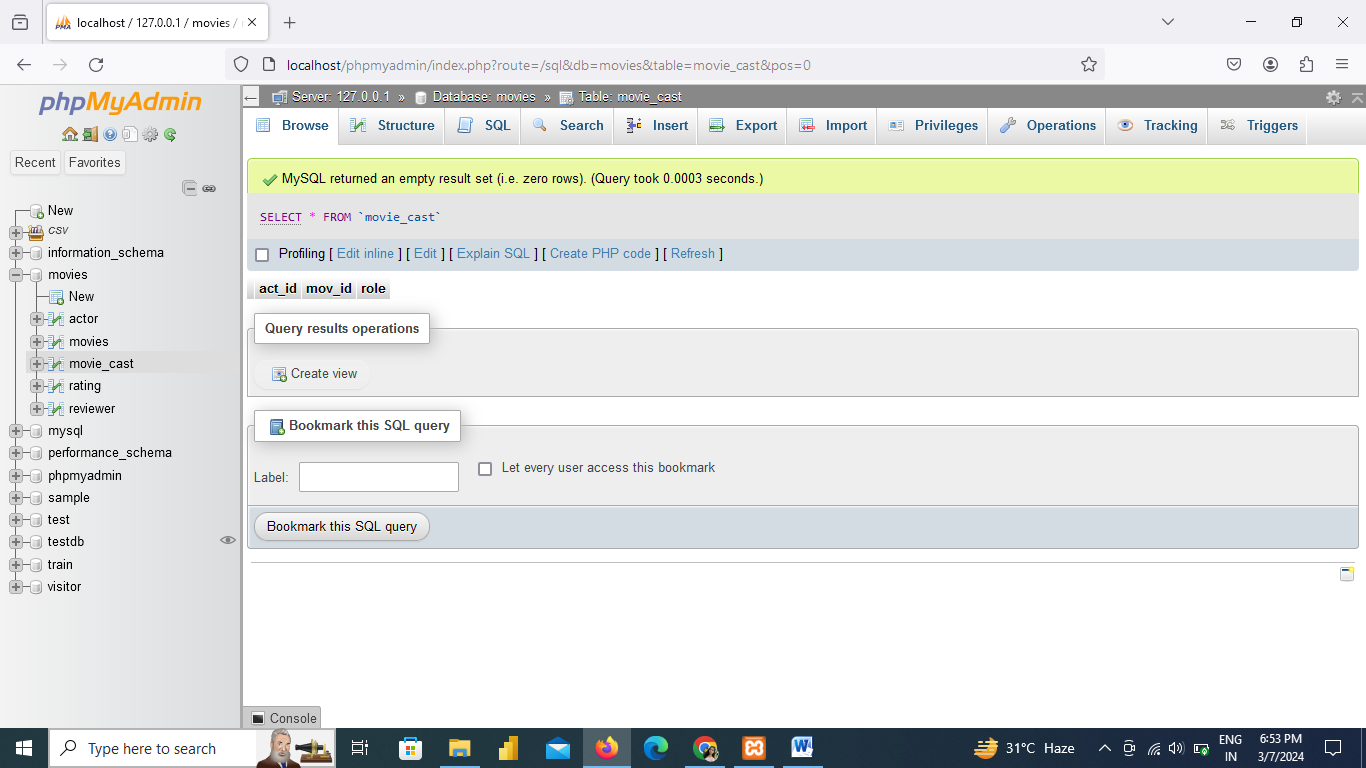
(124, 'Ali', 'Astin', 'F');



CREATE TABLE movie\_cast (act\_id INTEGER(50),

mov\_id INTEGER(50),

role VARCHAR(255) );



INSERT INTO movie\_cast (act\_id, mov\_id, role) VALUES

(101, 901, 'John Scottie Ferguson'),

(102, 902, 'Miss Giddens'),

(103, 903, 'T.E. Lawrence'),

(104, 904, 'Michael'),

(105, 905, 'Antonio Salieri'),

(106, 906, 'Rick Deckard'),

(107, 907, 'Alice Harford'),

(108, 908, 'McManus'),

(110, 910, 'Eddie Adams'),

(111, 911, 'Alvy Singer'),

(112, 912, 'San'),

(113, 913, 'Andy Dufresne'),

(114, 914, 'Lester Burnham'),

(115, 915, 'Rose DeWitt Bukater'),

(116, 916, 'Sean Maguire'),

(117, 917, 'Ed'),

(118, 918, 'Renton'),

(120, 920, 'Elizabeth Darko'),

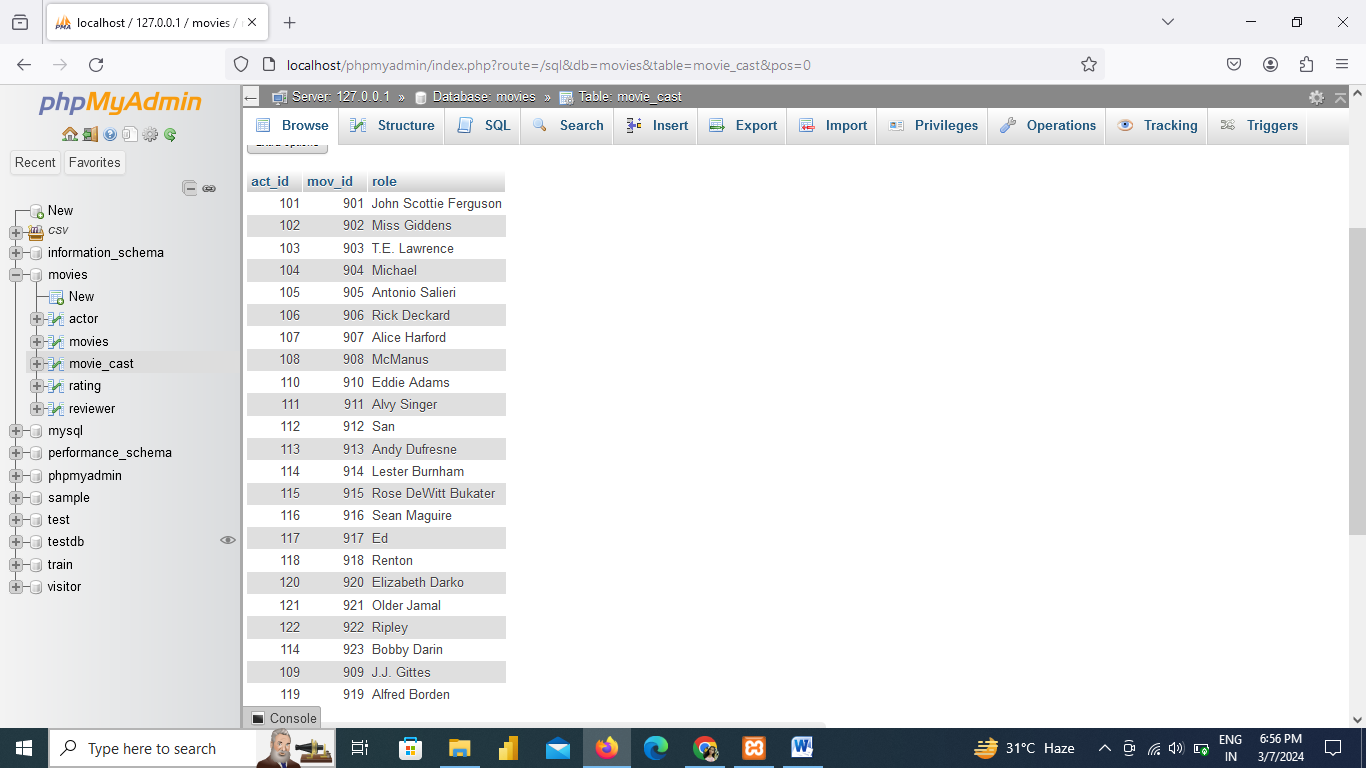
(121, 921, 'Older Jamal'),

(122, 922, 'Ripley'),

(114, 923, 'Bobby Darin'),

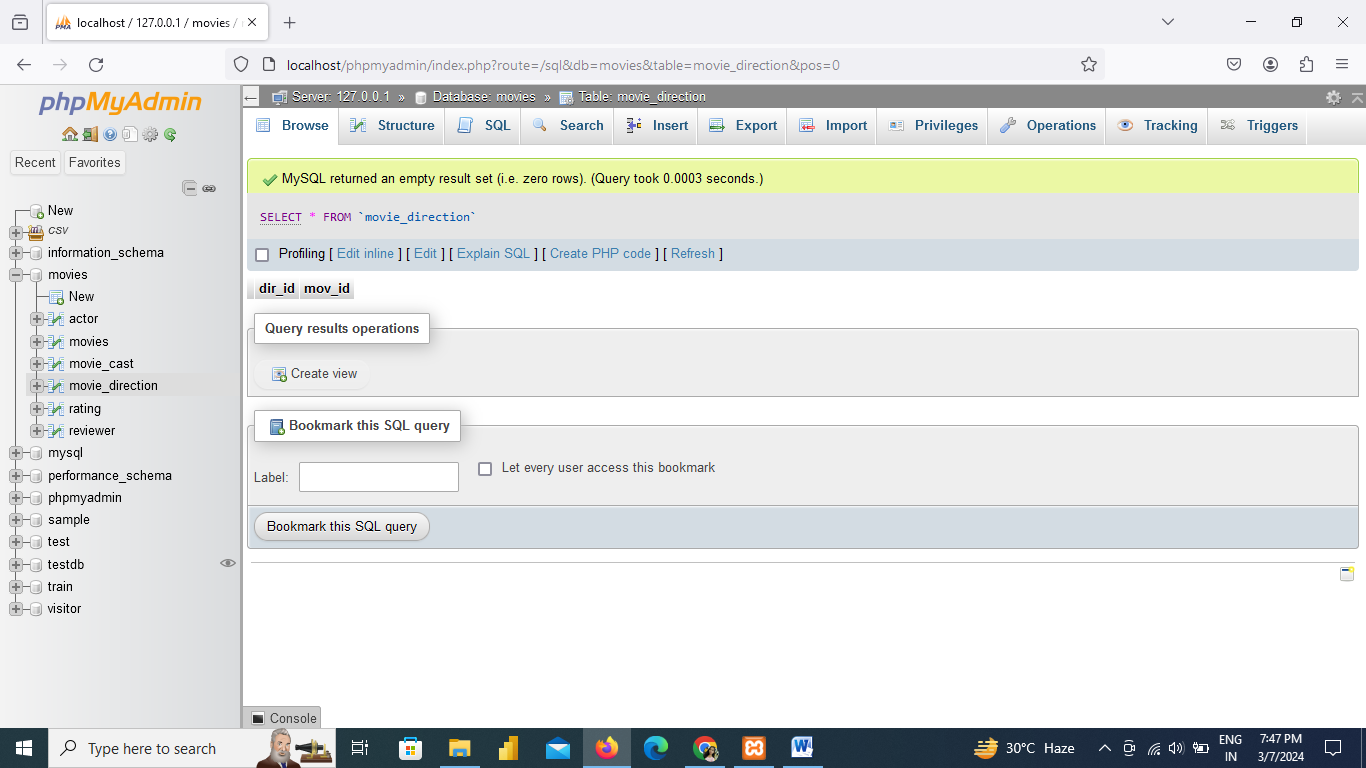
(109, 909, 'J.J. Gittes'),

(119, 919, 'Alfred Borden');



CREATE TABLE movie\_direction(dir\_id INTEGER (20),

mov\_id INTEGER (20));



INSERT INTO movie\_direction (dir\_id, mov\_id) VALUES

(201, 901),

(202, 902),

(203, 903),

(204, 904),

(205, 905),

(206, 906),

(207, 907),

(208, 908),

(209, 909),

(210, 910),

(211, 911),

(212, 912),

(213, 913),

(214, 914),

(215, 915),

(216, 916),

(217, 917),

(218, 918),

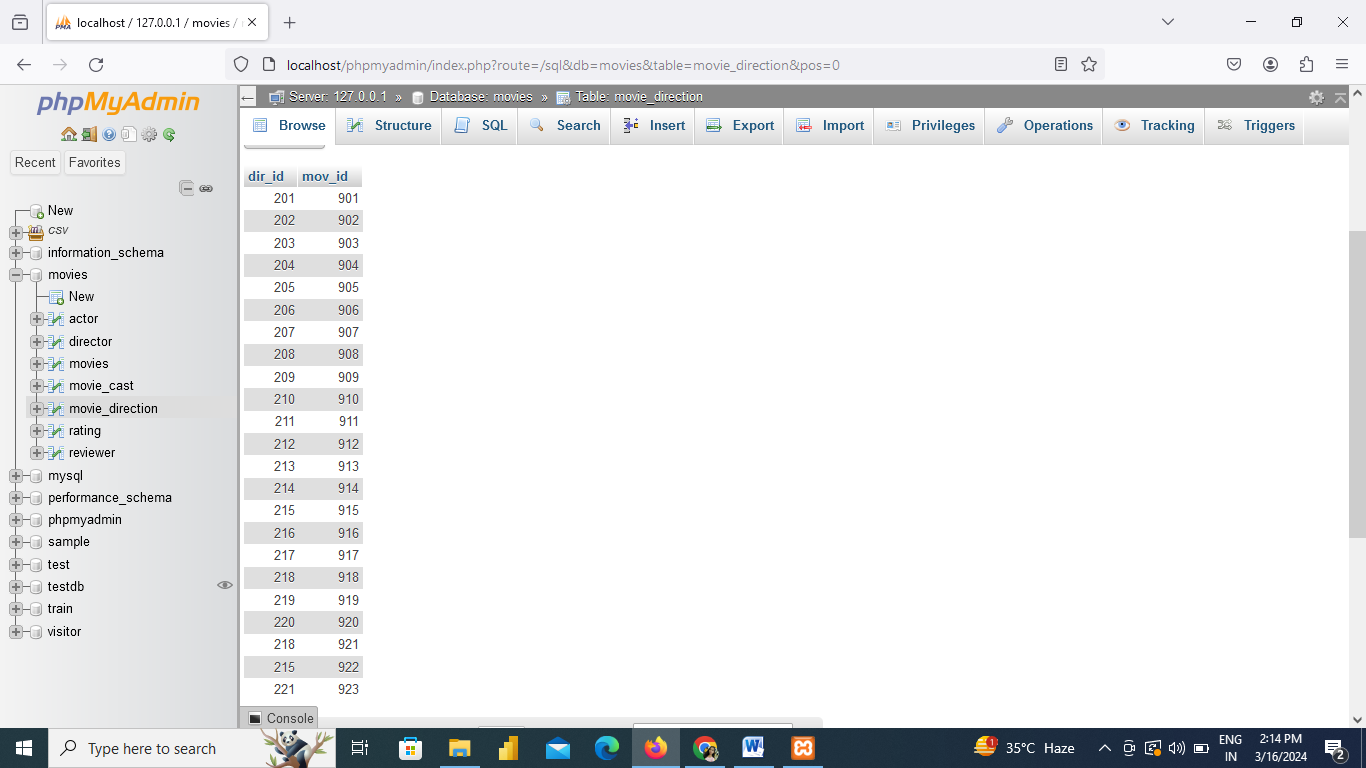
(219, 919),

(220, 920),

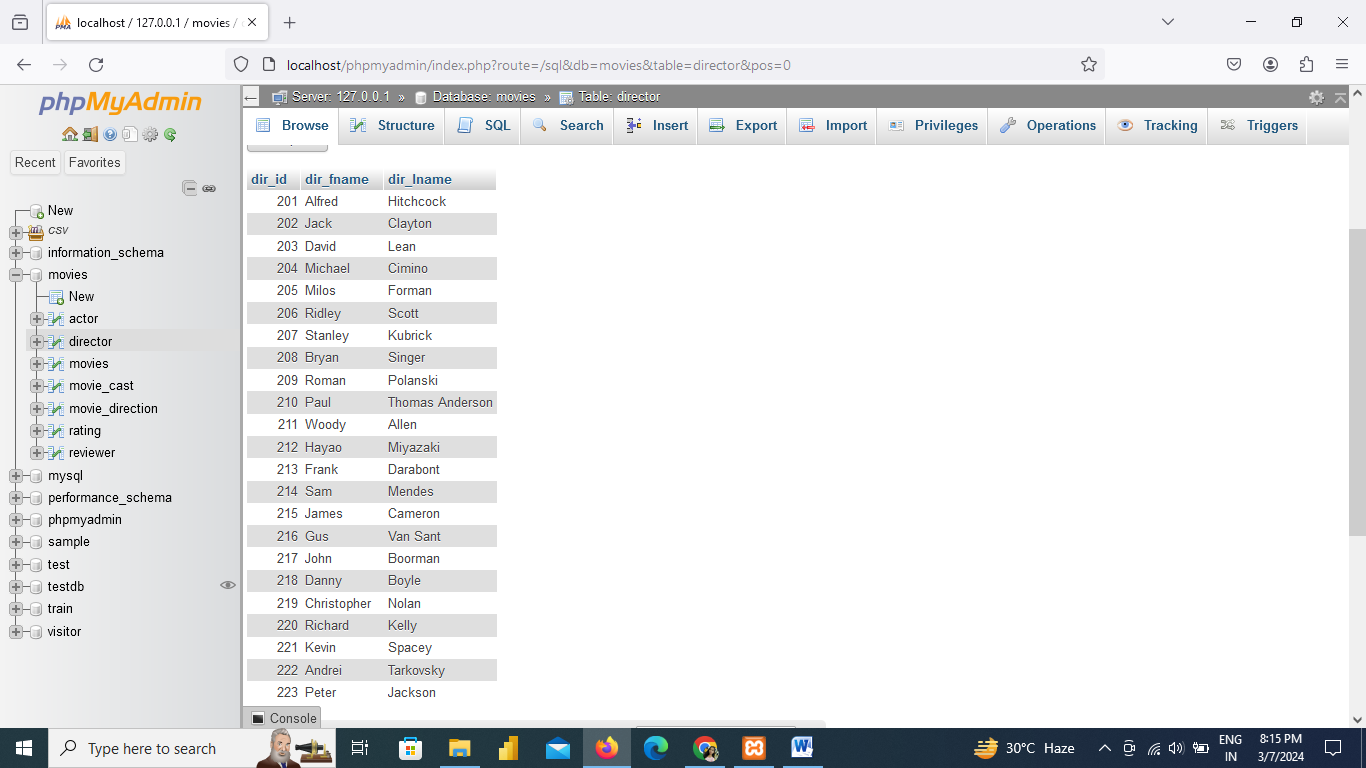
(218, 921),

(215, 922),

(221, 923);

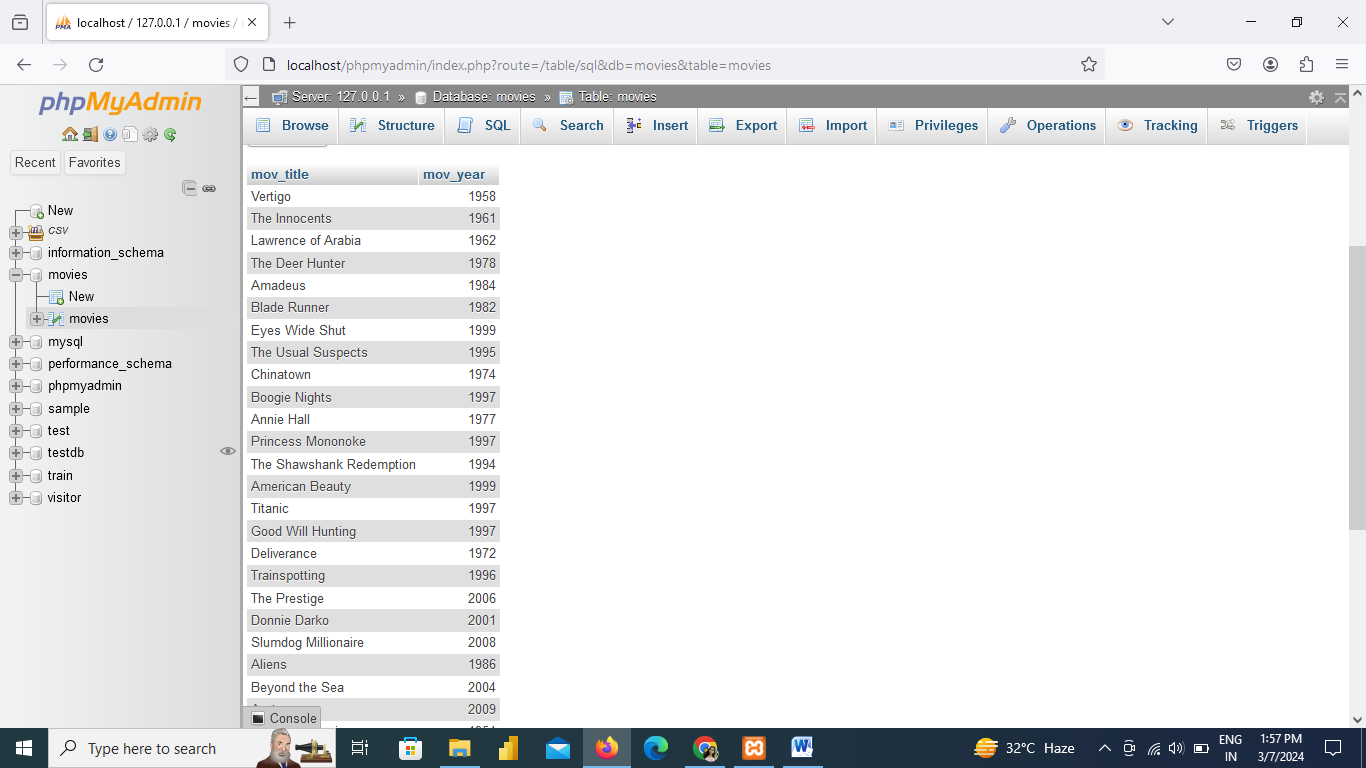


[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) director( dir\_id INTEGER(11), dir\_fname VARCHAR(255), dir\_lname VARCHAR(255));



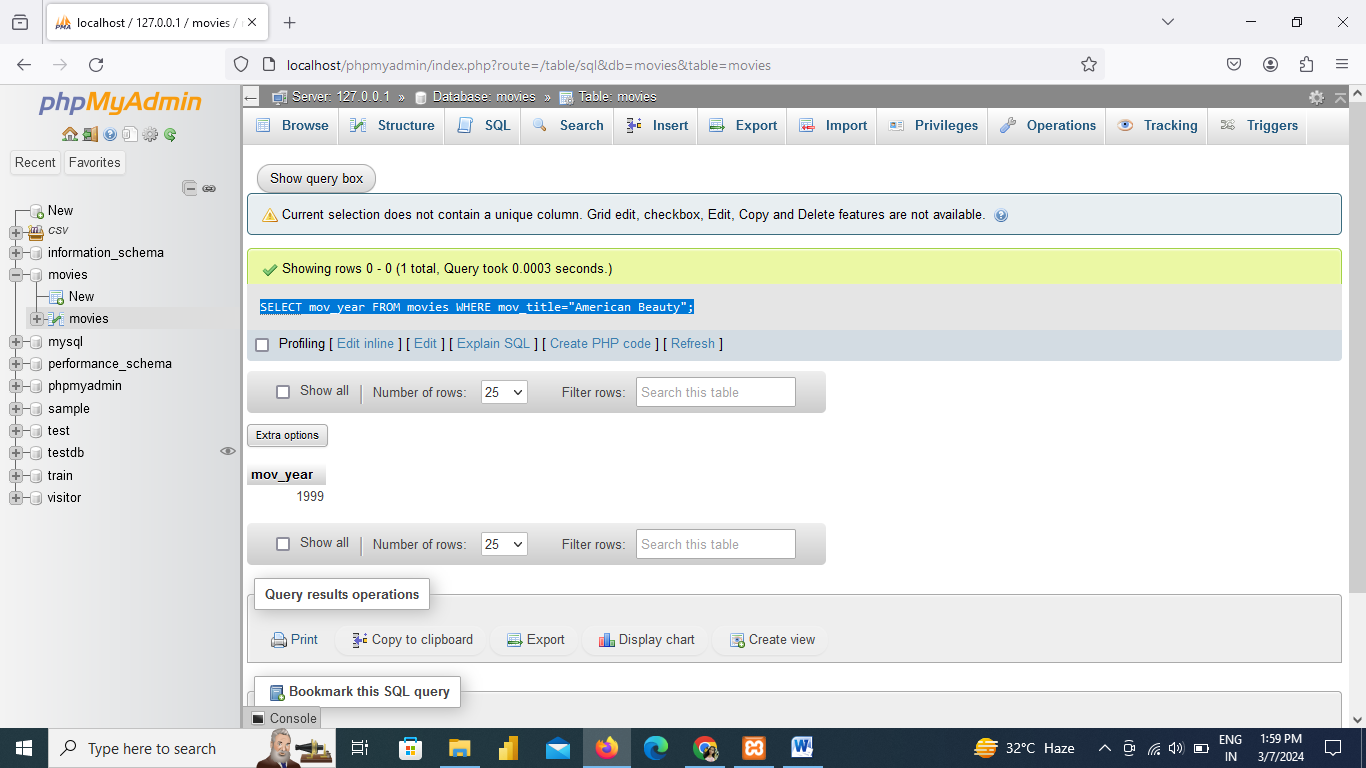
1. From the following table, write a SQL query to find the name and year of the movies. Return movie title, movie release year.

Syntax:[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html" \t "mysql_doc) mov\_title, mov\_year FROM movies;



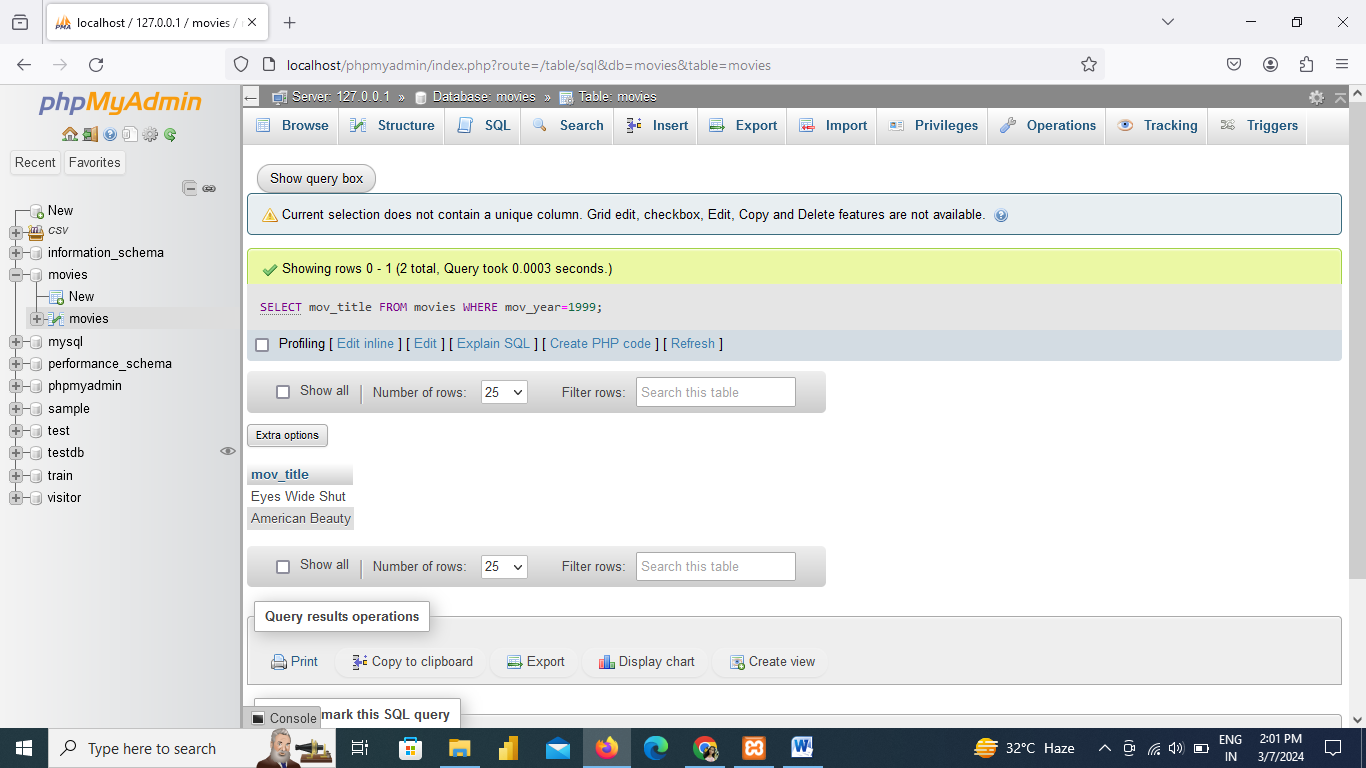
1. From the following table, write a SQL query to find when the movie 'American Beauty' released. Return movie release year.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_year FROM movies WHERE mov\_title="American Beauty";



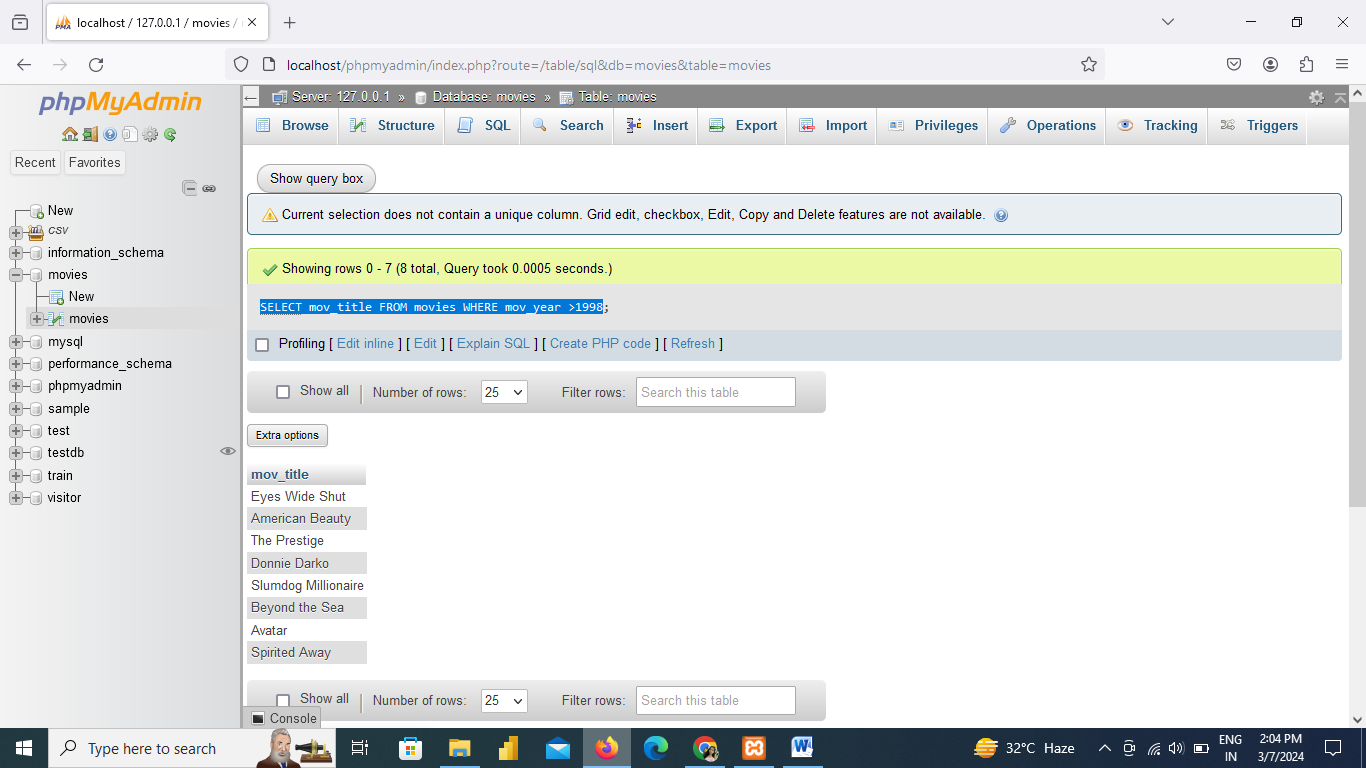
1. From the following table, write a SQL query to find the movie that was released in 1999. Return movie title.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_title FROM movies WHERE mov\_year=1999;



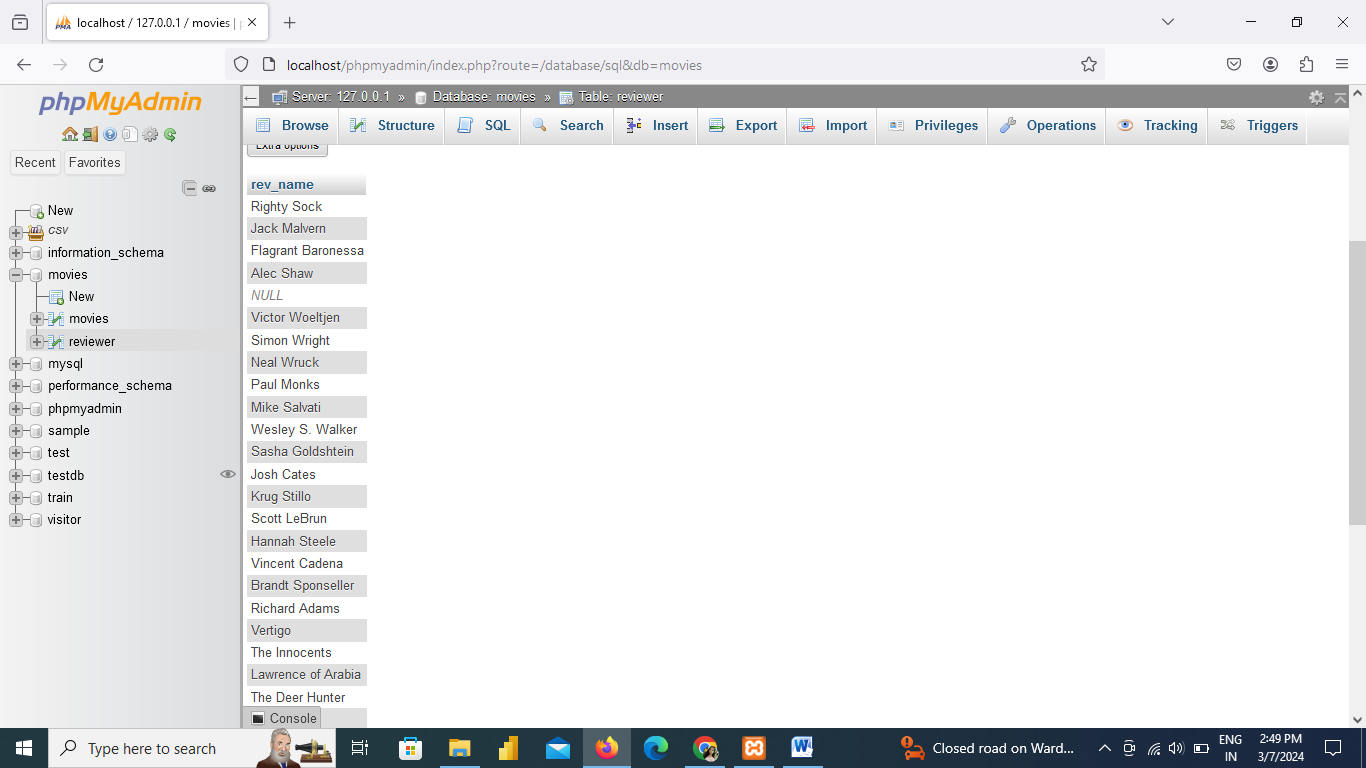
1. From the following table, write a SQL query to find those movies, which were released before 1998. Return movie title.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_title FROM movies WHERE mov\_year >1998



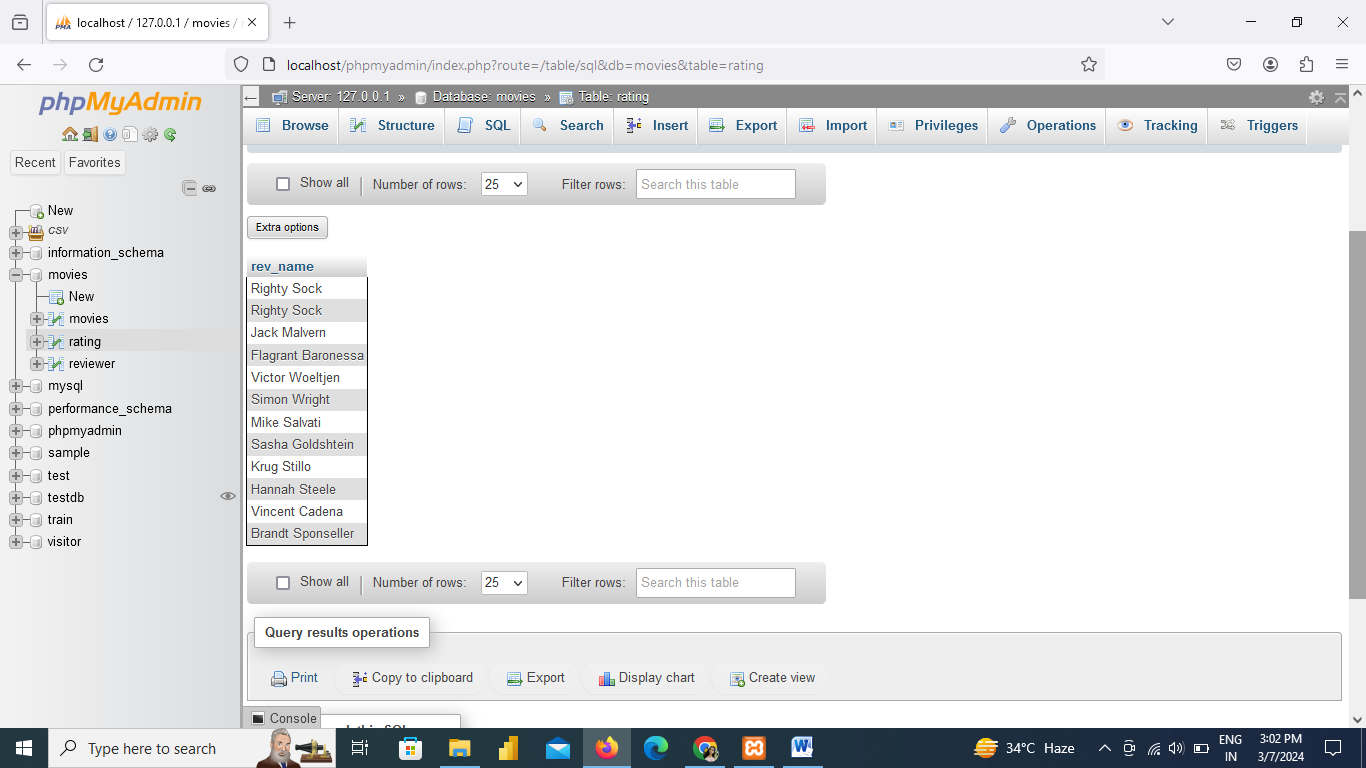
From the following tables, write a SQL query to find the name of all reviewers and movies together in a single list.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) r.rev\_name FROM reviewer r UNION ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) m.mov\_title FROM movies m);



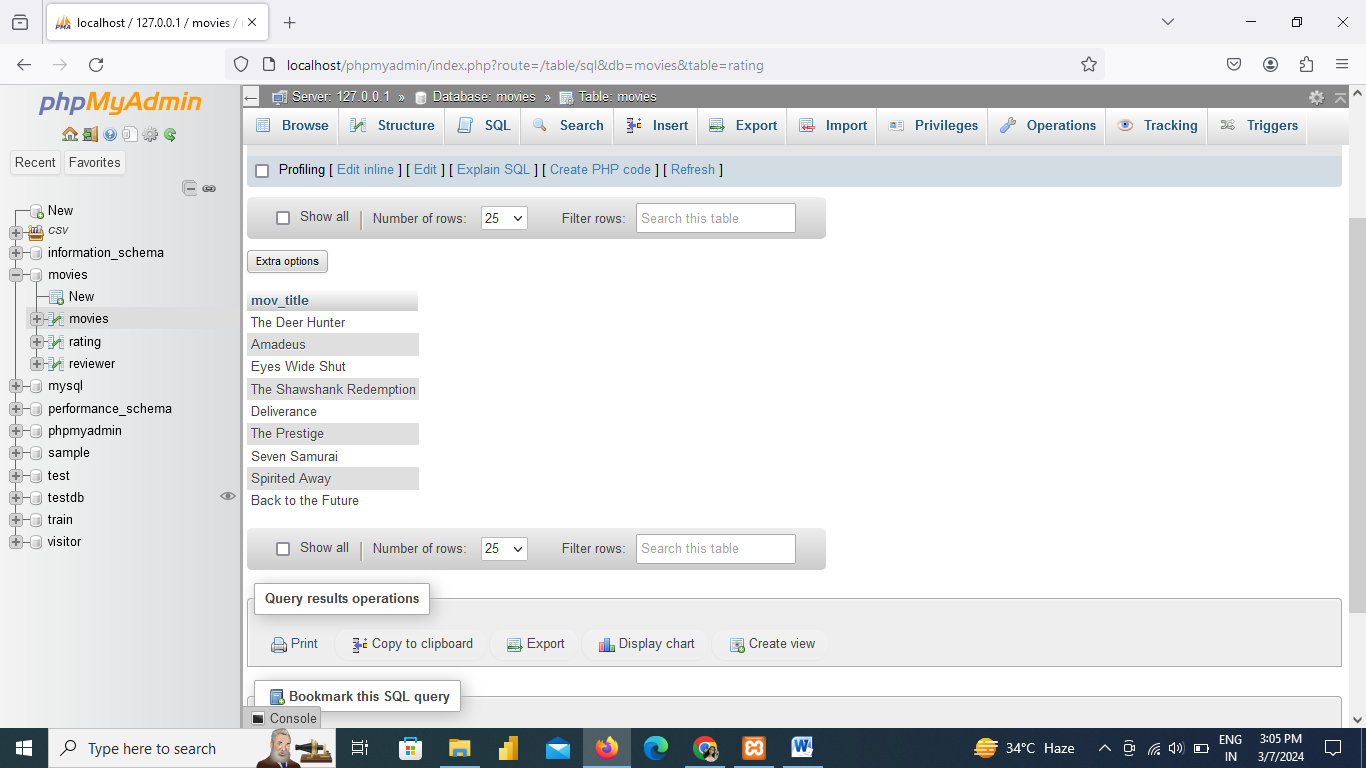
**6.** From the following table, write a SQL query to find all reviewers who have rated seven or more stars to their rating. Return reviewer name.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) rev\_name FROM reviewer,rating WHERE rev\_stars >=7 [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_and) reviewer.rev\_id= rating.rev\_id [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_and) reviewer.rev\_name [IS](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23operator_is) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL;



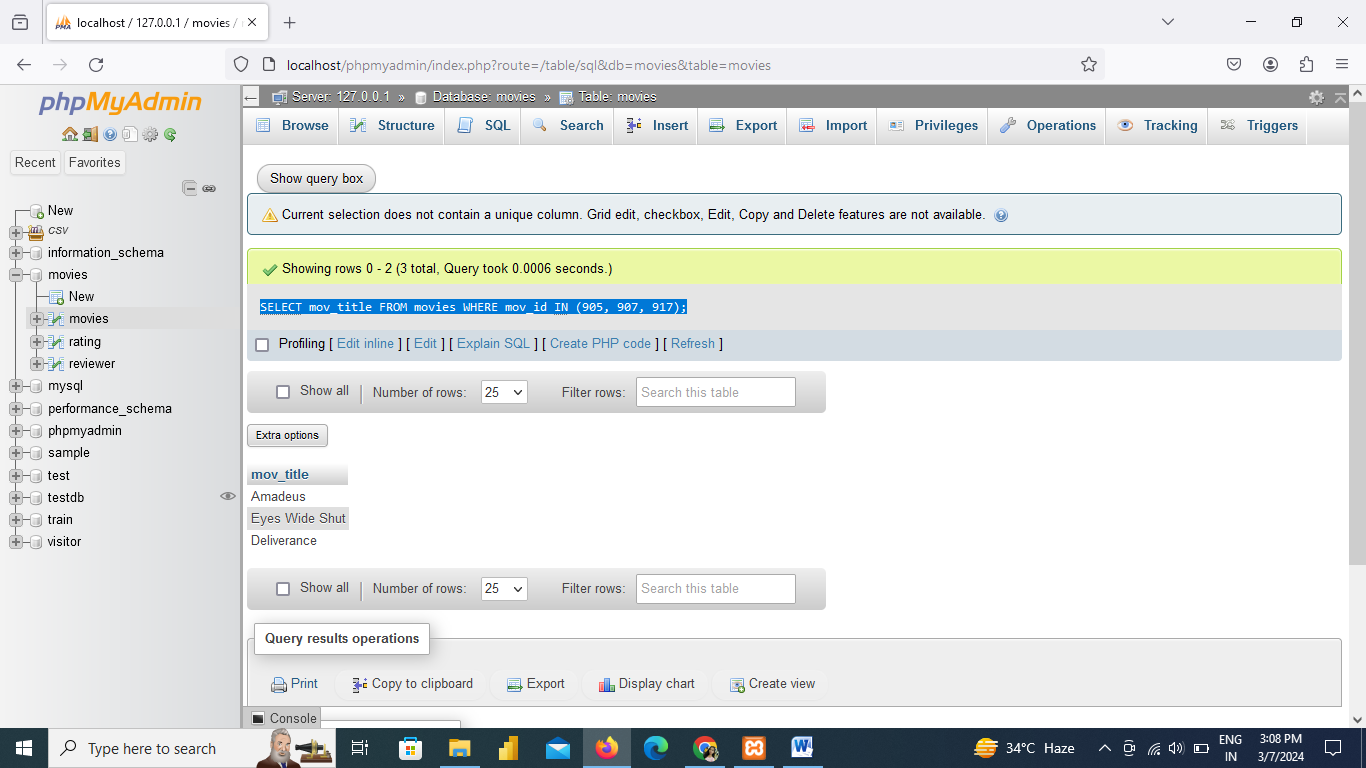
From the following tables, write a SQL query to find the movies without any rating. Return movie title.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_title FROM movies WHERE mov\_id [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_in) ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_id FROM rating);



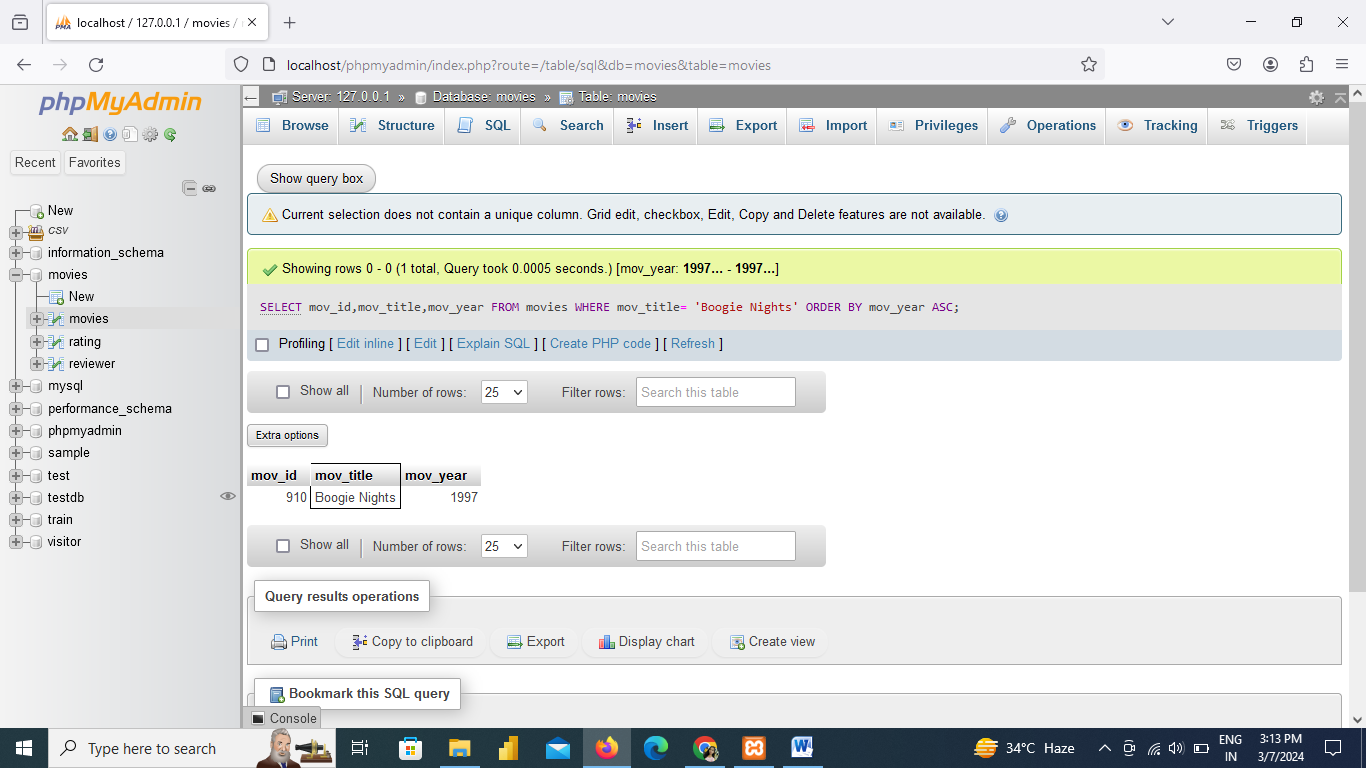
**8.** From the following table, write a SQL query to find the movies with ID 905 or 907 or 917. Return movie title.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_title FROM movies WHERE mov\_id [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_in) (905, 907, 917);



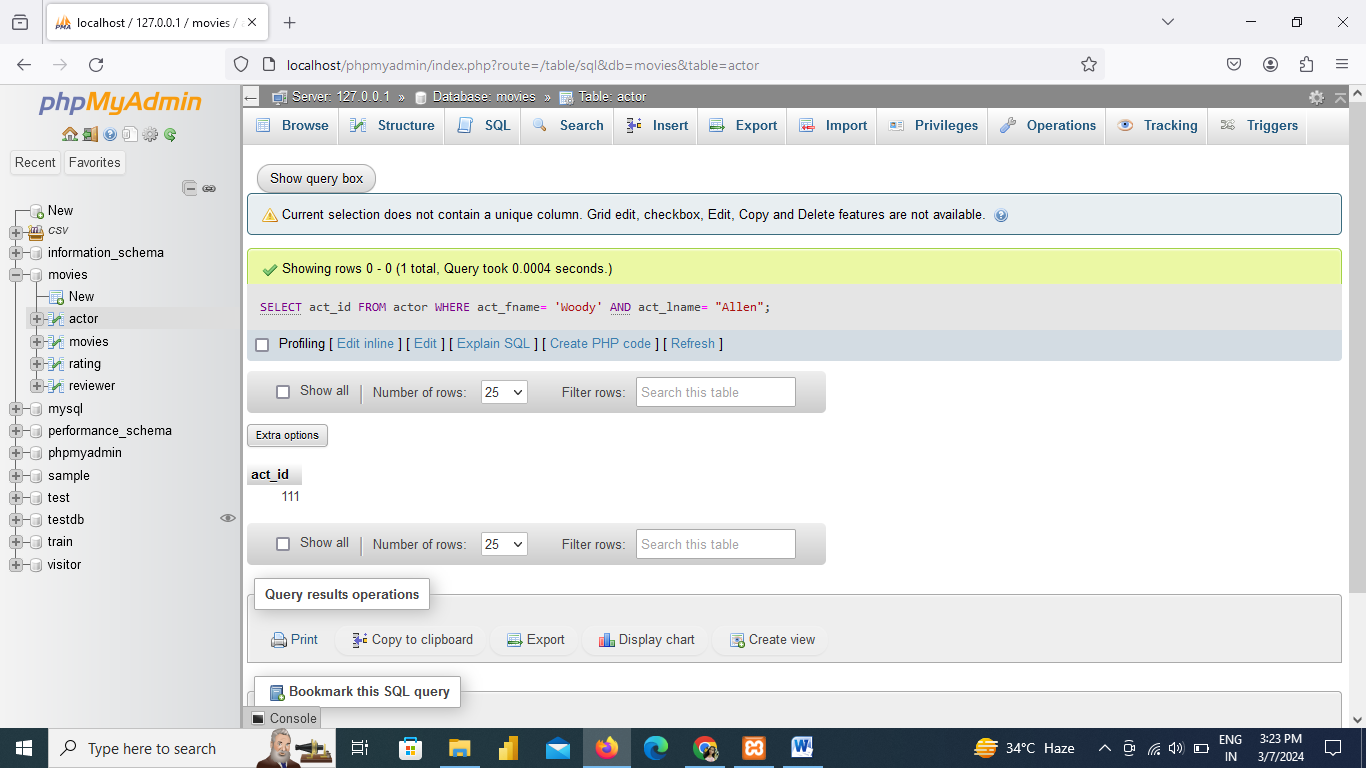
From the following table, write a SQL query to find the movie titles that contain the word 'Boogie Nights'. Sort the result-set in ascending order by movie year. Return movie ID, movie title and movie release year.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_id,mov\_title,mov\_year FROM movies WHERE mov\_title= 'Boogie Nights' ORDER BY mov\_year ASC;



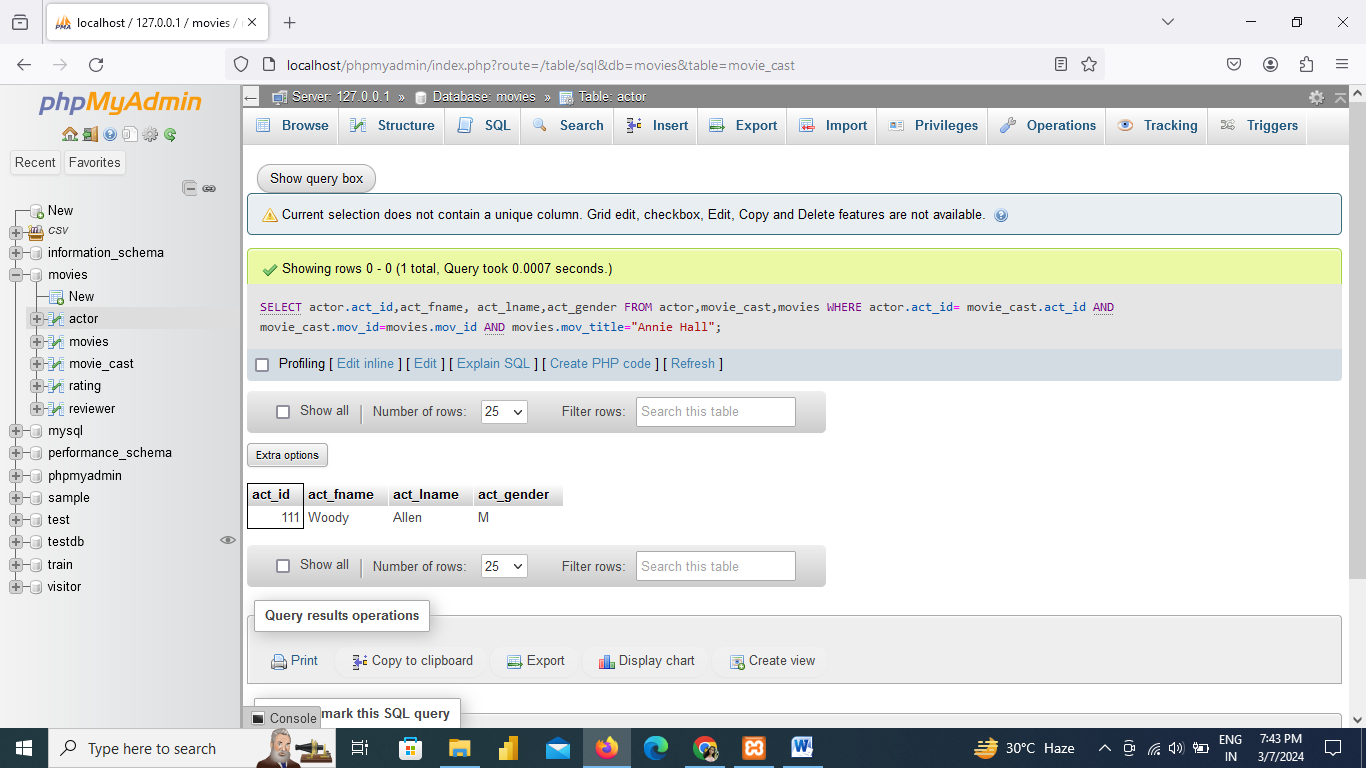
 From the following table, write a SQL query to find those actors with the first name 'Woody' and the last name 'Allen'. Return actor ID.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) act\_id FROM actor WHERE act\_fname= 'Woody' [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_and) act\_lname= "Allen";



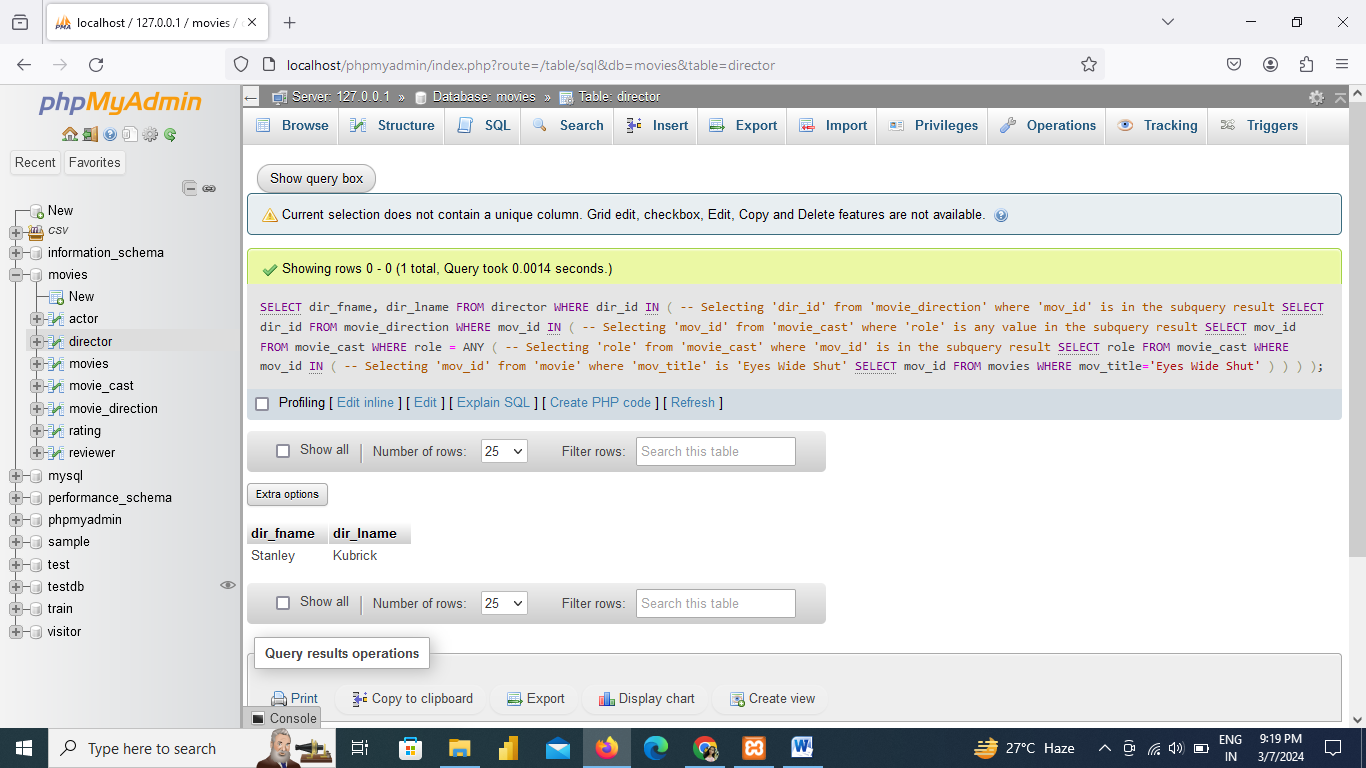
From the following table, write a SQL query to find the actors who played a role in the movie 'Annie Hall'. Return all the fields of actor table.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) actor.act\_id,act\_fname, act\_lname,act\_gender FROM actor,movie\_cast,movies WHERE actor.act\_id= movie\_cast.act\_id [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_and) movie\_cast.mov\_id=movies.mov\_id [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_and) movies.mov\_title="Annie Hall";

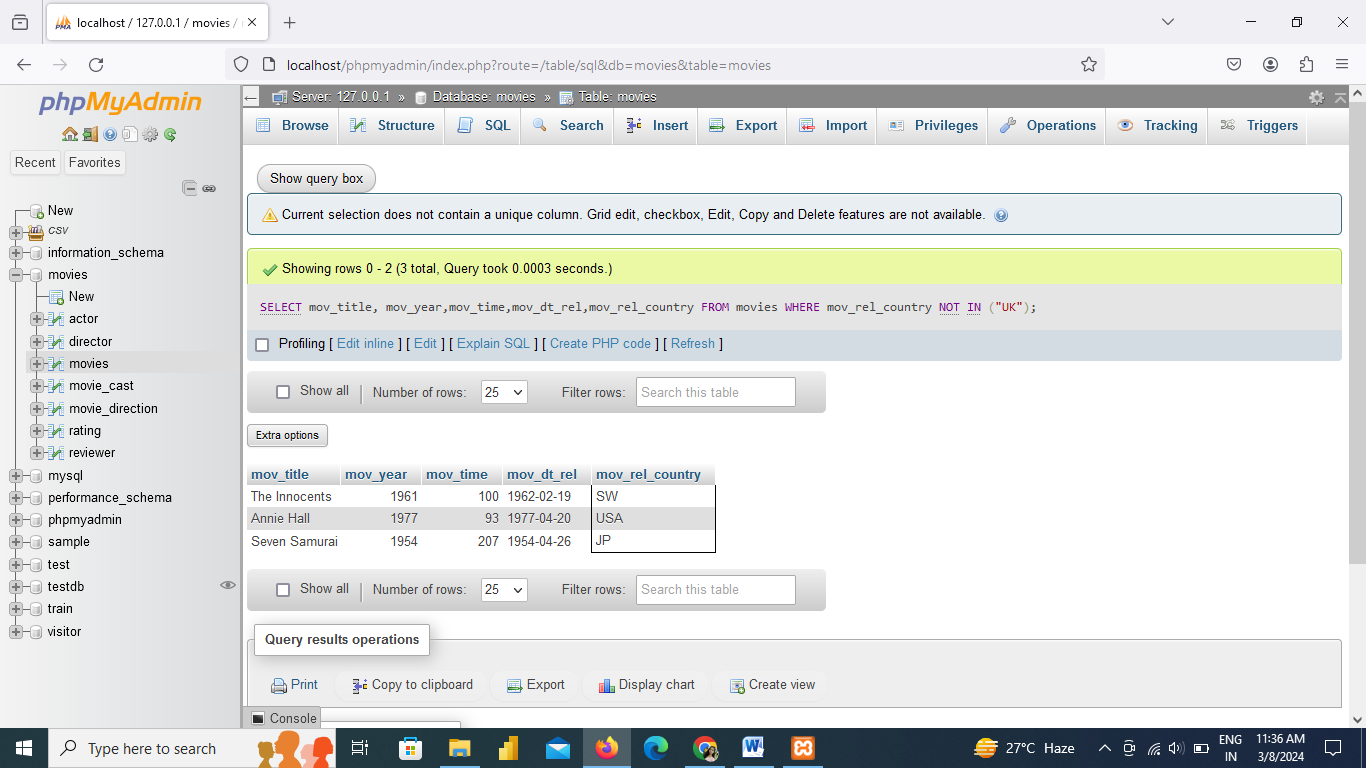


From the following tables, write a SQL query to find the director of a film that cast a role in 'Eyes Wide Shut'. Return director first name, last name.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) dir\_fname, dir\_lname FROM director WHERE dir\_id [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_in) ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) dir\_id FROM movie\_direction WHERE mov\_id [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_in) ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_id FROM movie\_cast WHERE role = ANY ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) role FROM movie\_cast WHERE mov\_id [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_in) ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_id FROM movies WHERE mov\_title='Eyes Wide Shut' ) ) ) );

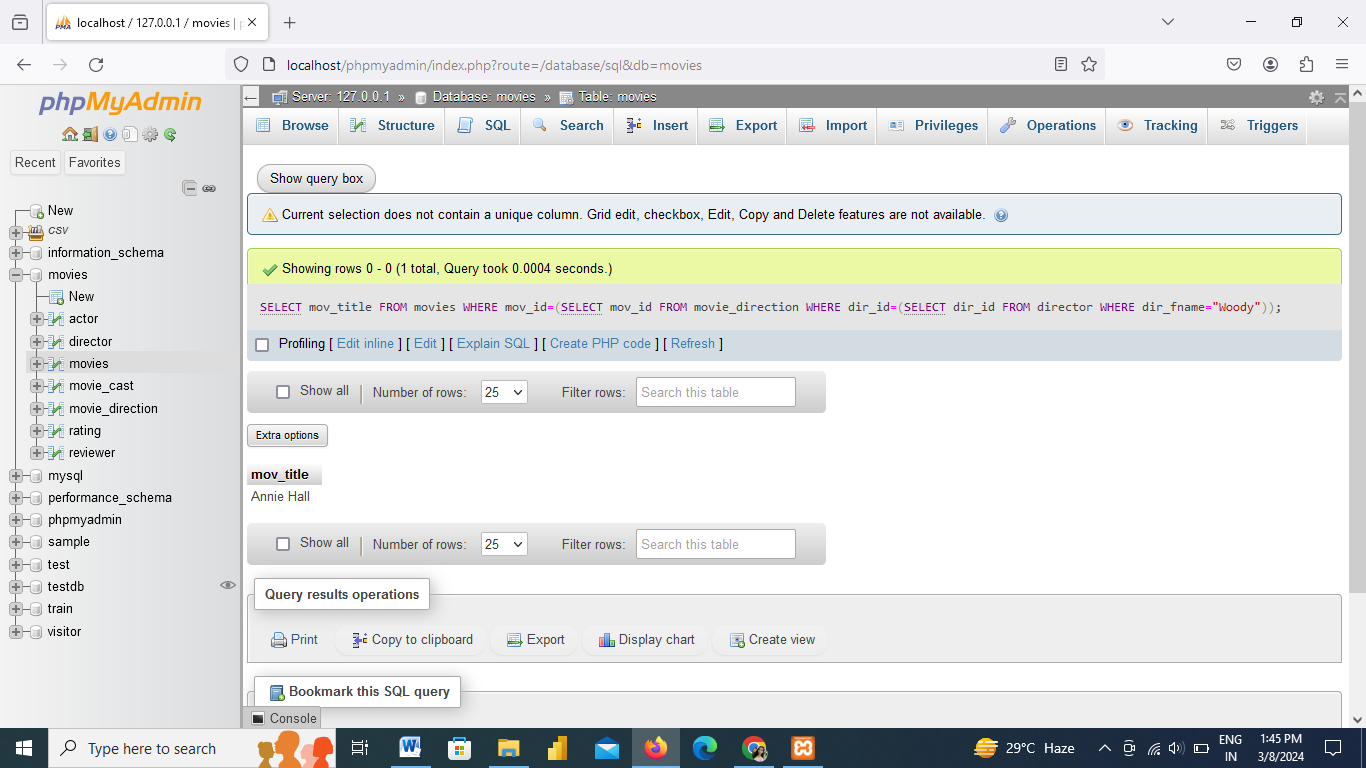


1. From the following table, write a SQL query to find those movies that have been released in countries other than the United Kingdom. Return movie title, movie year, movie time, and date of release, releasing country.
2. [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_title, mov\_year,mov\_time,mov\_dt\_rel,mov\_rel\_country FROM movies WHERE mov\_rel\_country [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_in) ("UK");



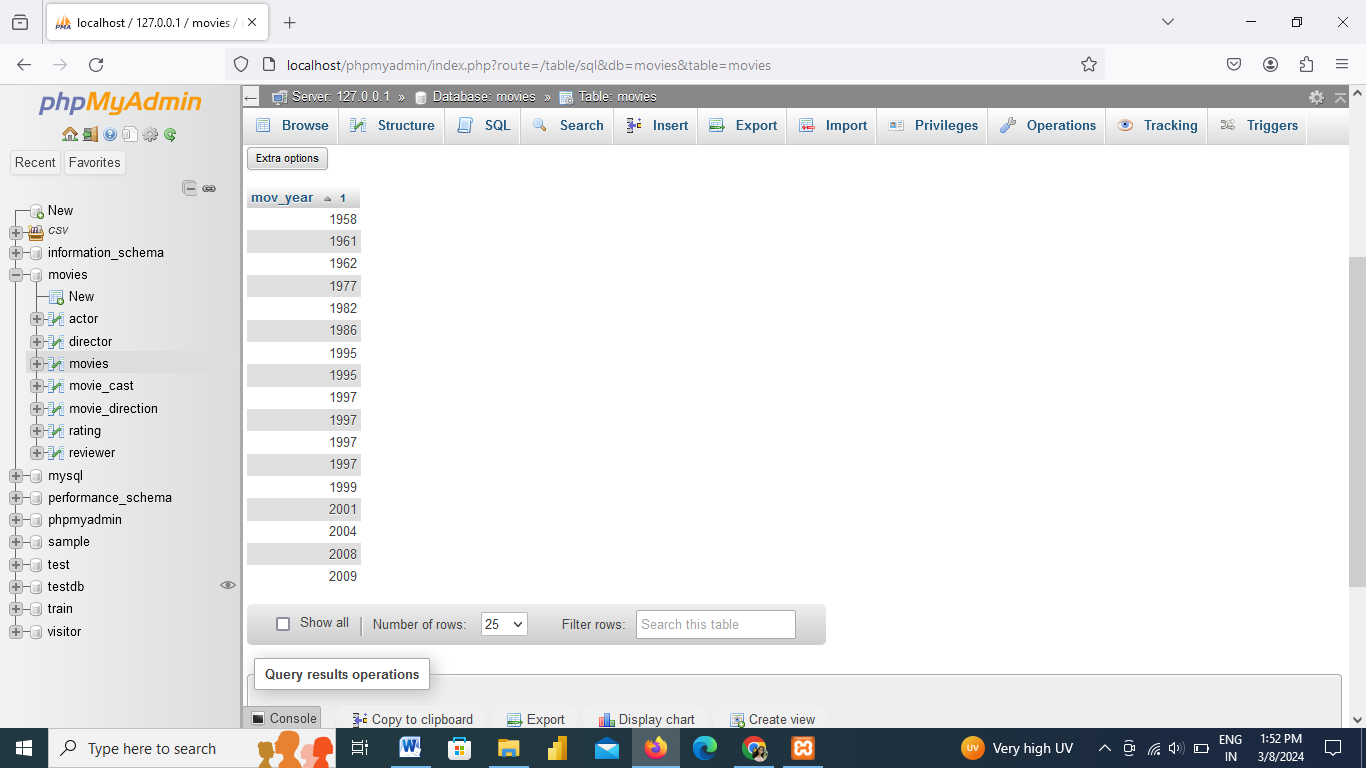
From the following tables, write a SQL query to find those movies directed by the director whose first name is Woddy and last name is Allen. Return movie title

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_title FROM movies WHERE mov\_id=([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html" \t "mysql_doc) mov\_id FROM movie\_direction WHERE dir\_id=([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) dir\_id FROM director WHERE dir\_fname="Woody"));



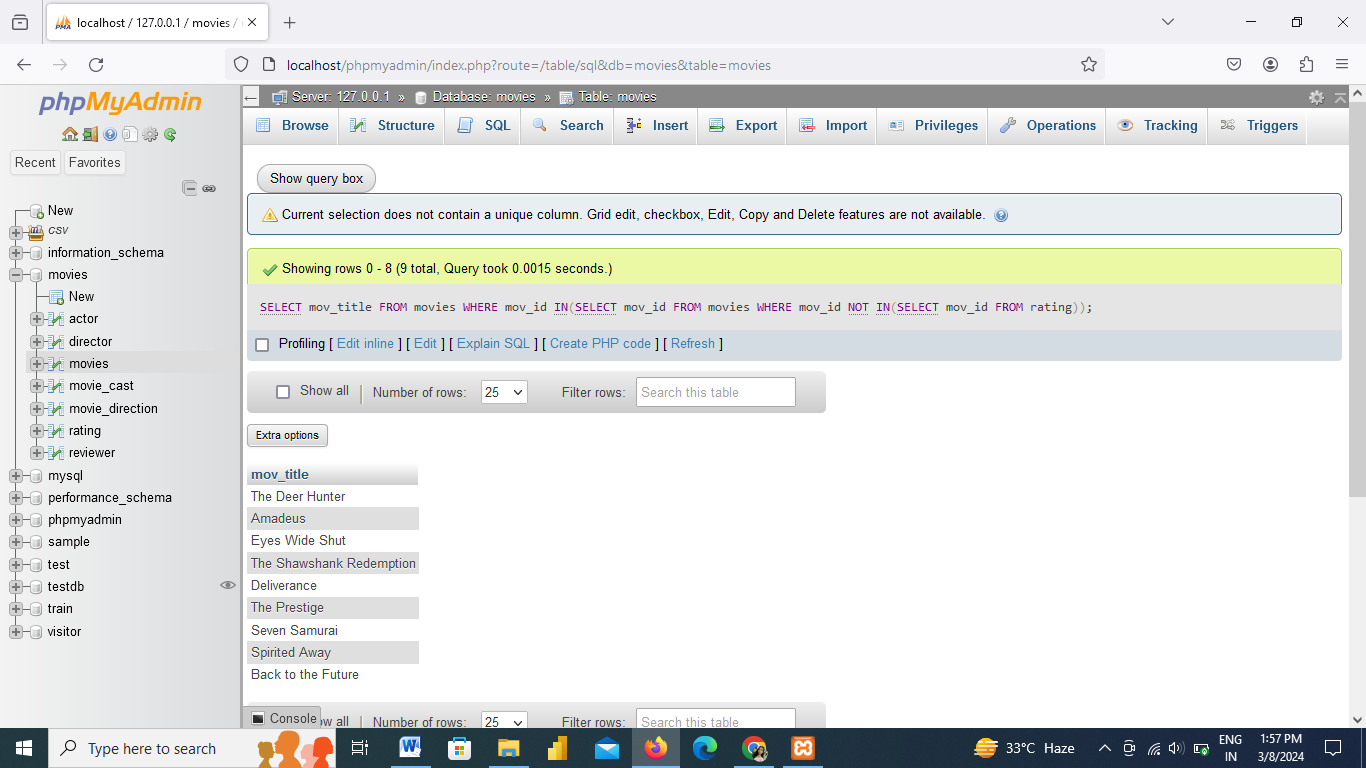
From the following tables, write a SQL query to determine those years in which there was at least one movie that received a rating of at least three stars. Sort the result-set in ascending order by movie year. Return movie year.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_year FROM movies WHERE mov\_id [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_in)([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_id FROM rating WHERE rev\_stars>=3) ORDER BY mov\_year;



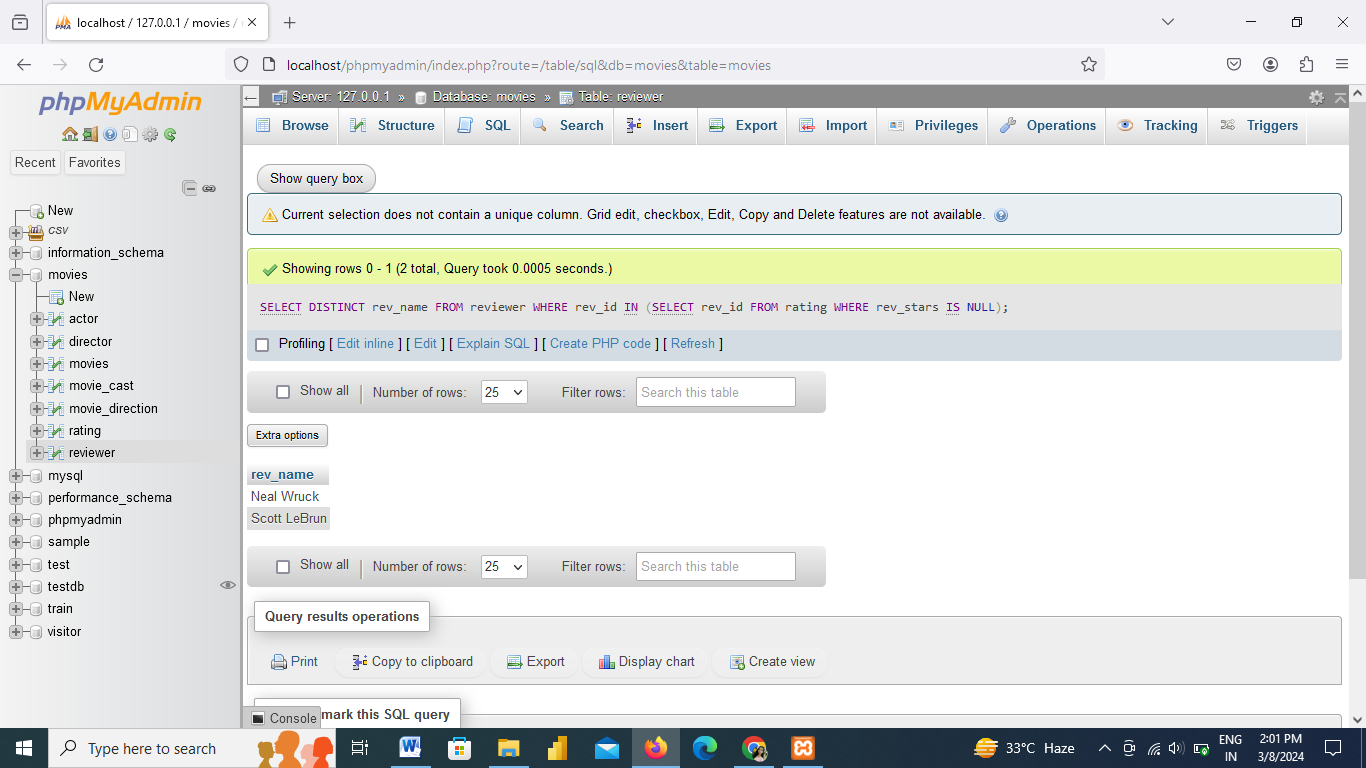
From the following table, write a SQL query to search for movies that do not have any ratings. Return movie title.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_title FROM movies WHERE mov\_id [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_in)([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_id FROM movies WHERE mov\_id [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_in)([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_id FROM rating));



From the following table, write a SQL query to find those reviewers who have not given a rating to certain films. Return reviewer name.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DISTINCT rev\_name FROM reviewer WHERE rev\_id [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_in) ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) rev\_id FROM rating WHERE rev\_stars [IS](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23operator_is) NULL);



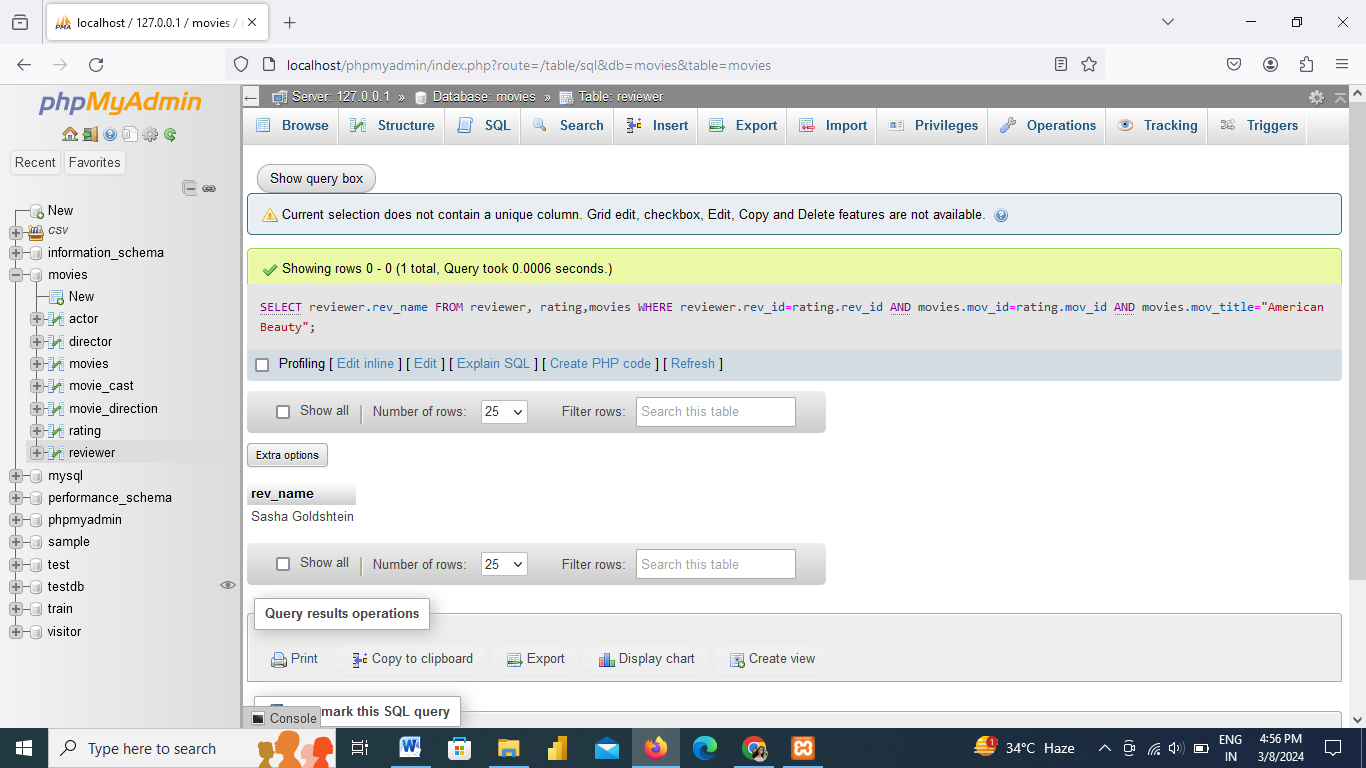
From the following tables, write a SQL query to find those movies, which have received highest number of stars. Group the result set on movie title and sorts the result-set in ascending order by movie title. Return movie title and maximum number of review stars.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_title,[MAX](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_max)(rev\_stars) FROM movies, rating WHERE movies.mov\_id=rating.mov\_id [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_and) rating.rev\_stars [IS](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23operator_is) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL GROUP BY mov\_title ORDER BY mov\_title;



From the following tables, write a SQL query to find all reviewers who rated the movie 'American Beauty'. Return reviewer name.

SELECT reviewer.rev\_name FROM reviewer, rating,movies WHERE reviewer.rev\_id=rating.rev\_id AND movies.mov\_id=rating.mov\_id AND movies.mov\_title="American Beauty";



From the following table, write a SQL query to find the movies that have not been reviewed by any reviewer body other than 'Paul Monks'. Return movie title.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) movies.mov\_title FROM movies WHERE movies.mov\_id [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_in) ( [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) mov\_id FROM rating WHERE rev\_id [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_in) ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) rev\_id FROM reviewer WHERE rev\_name="Paul Monks"));

