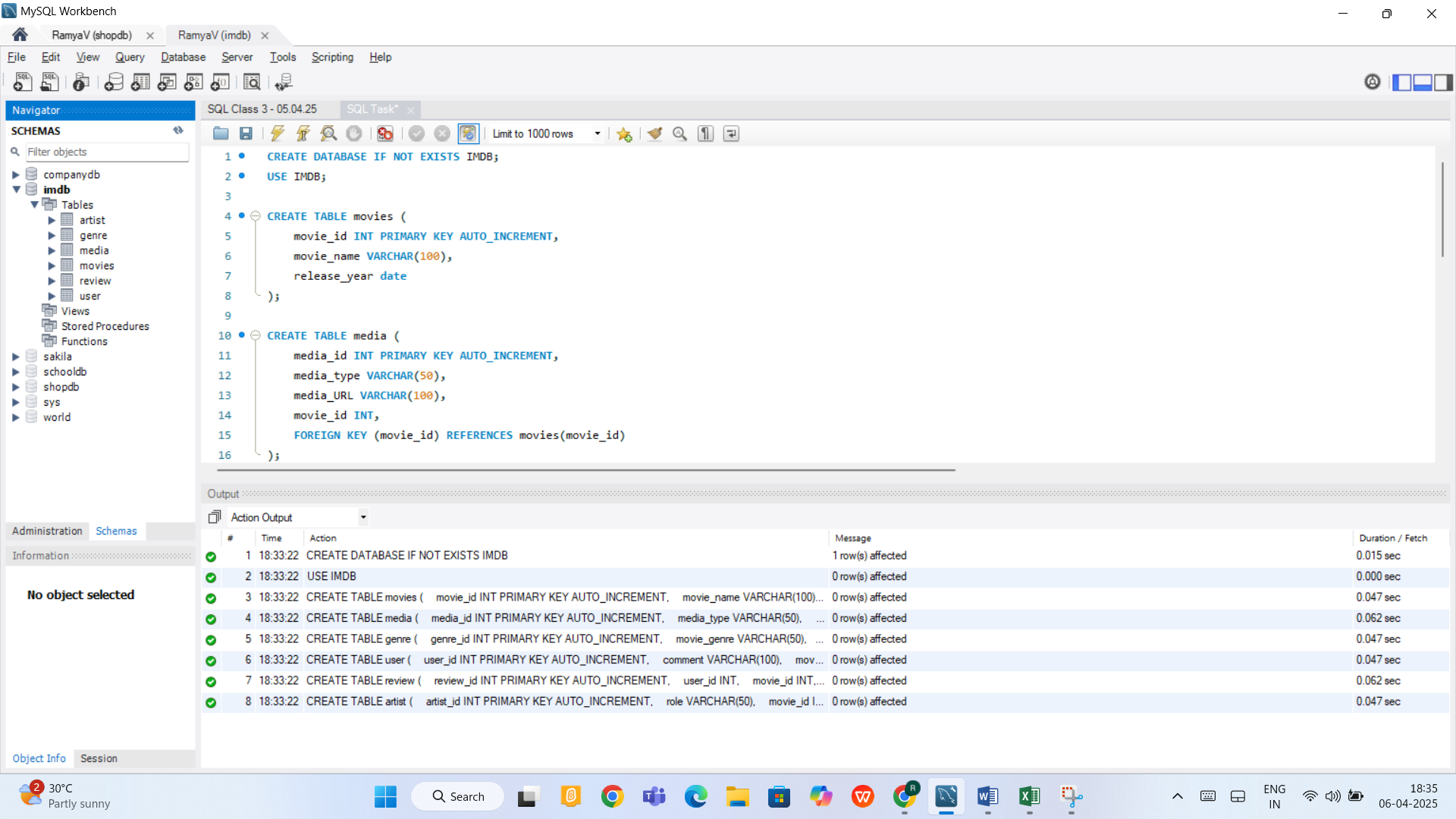
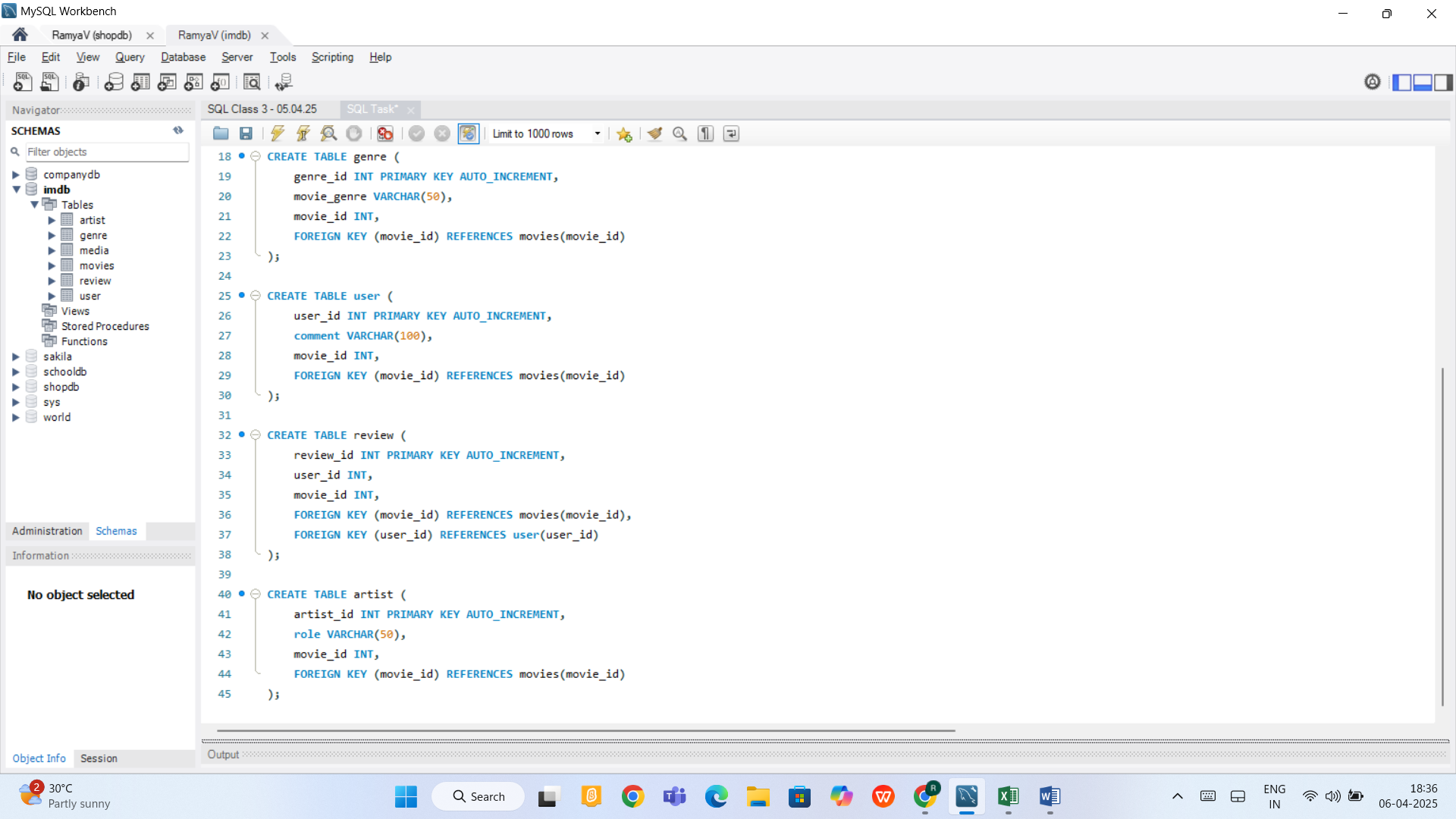
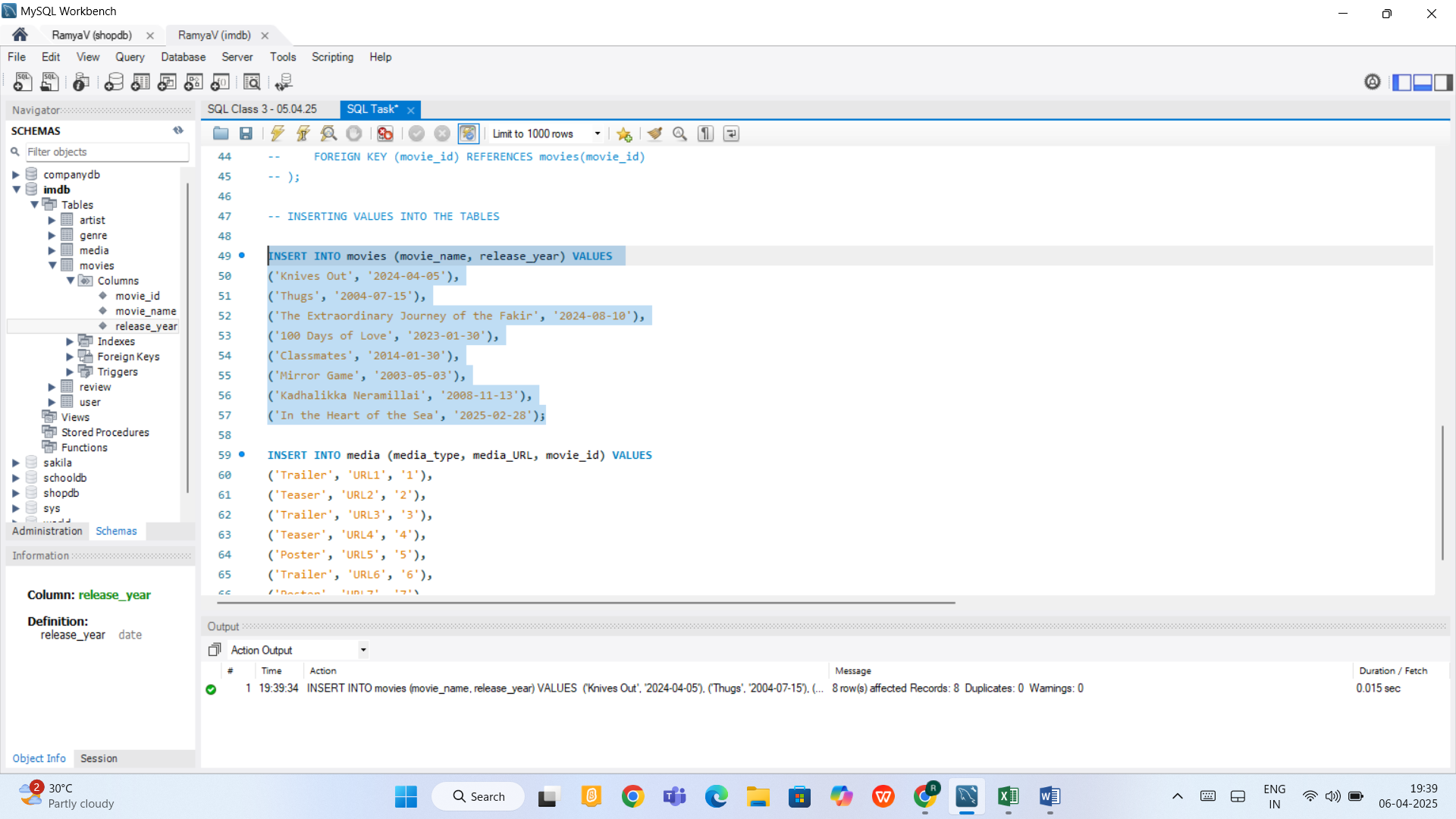
**TASK 7 – SQL TASK**

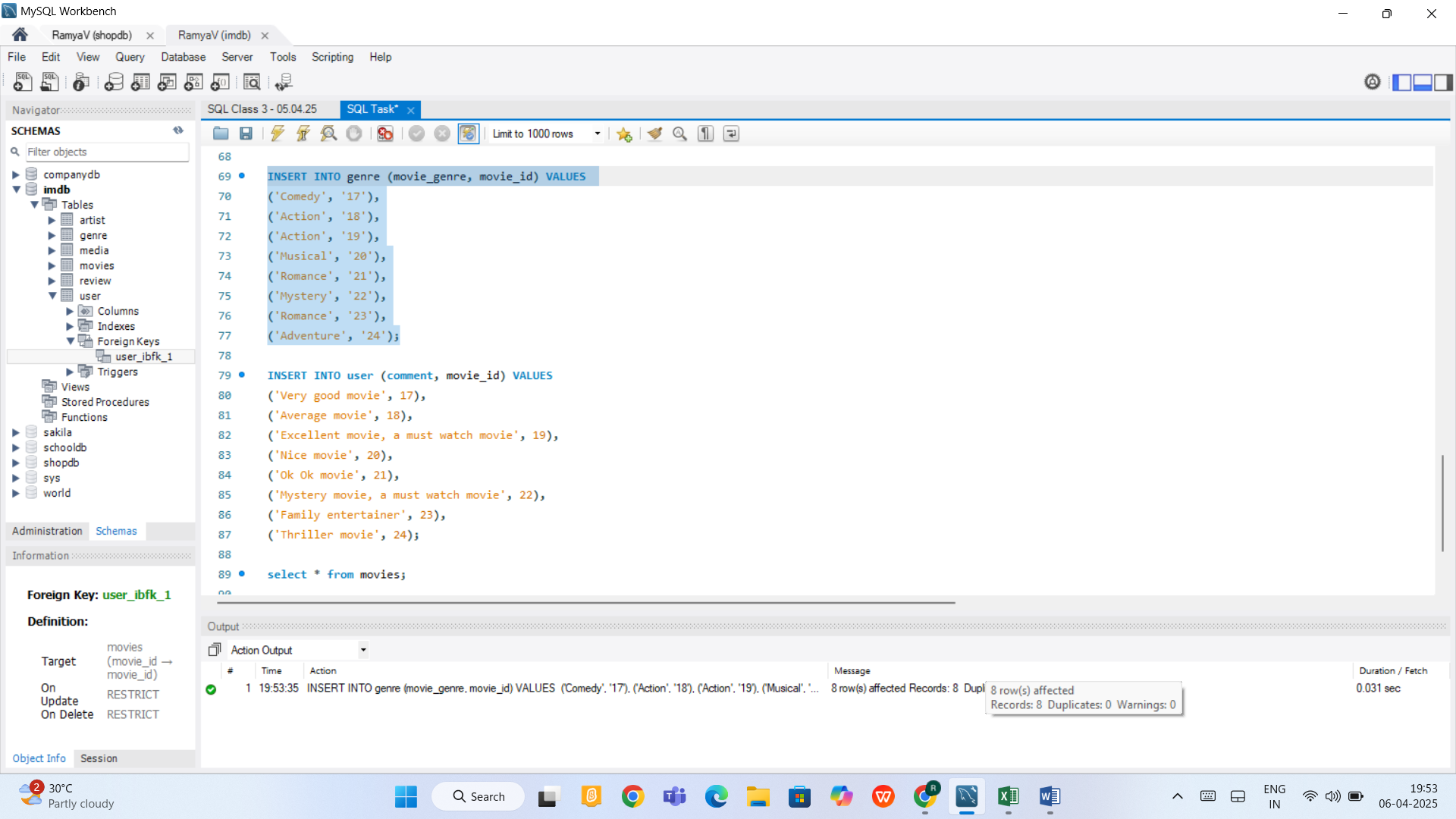
**CREATING DATABASE AND TABLES**





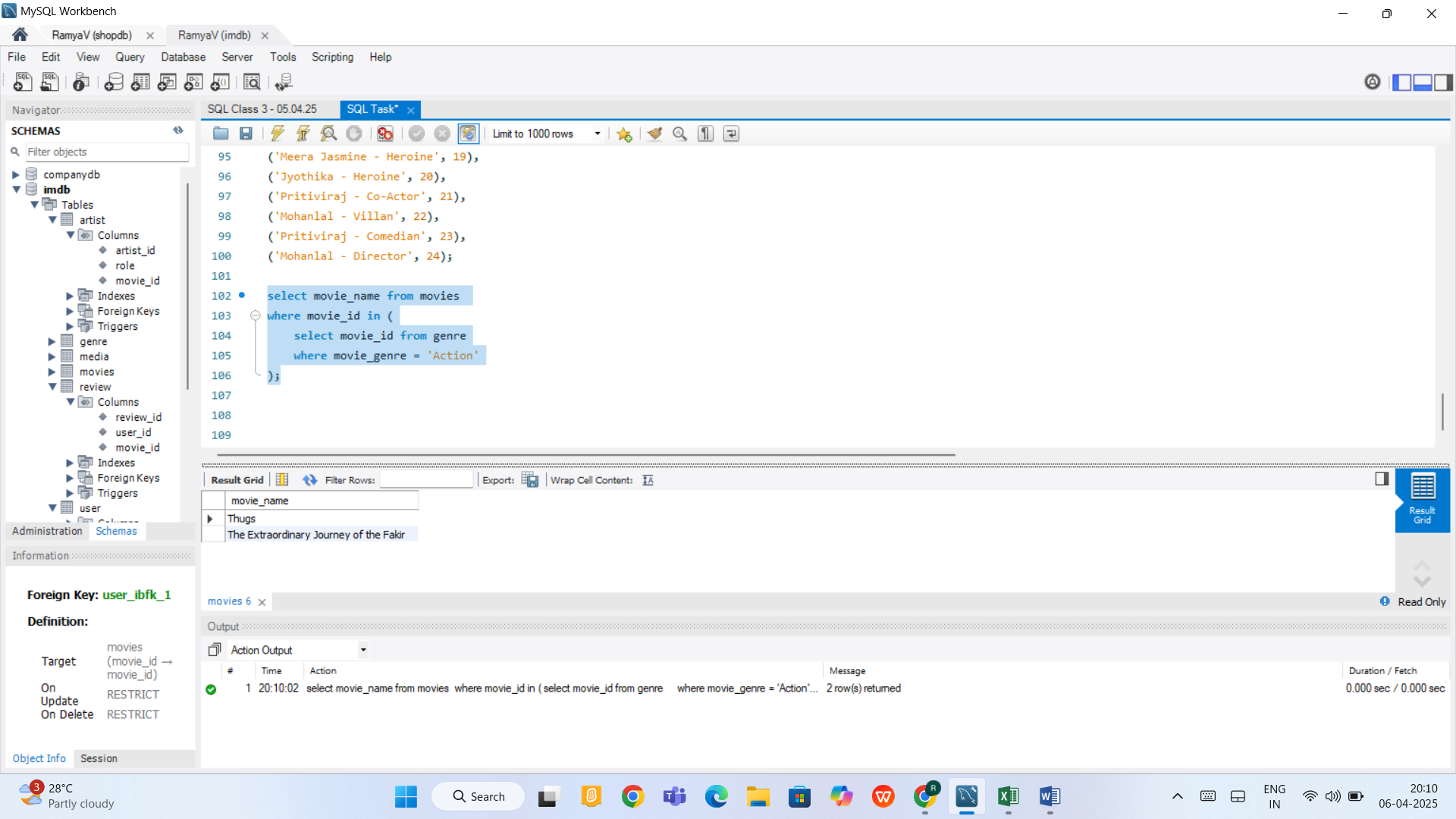
**INSERTING DATA INTO THE TABLES**





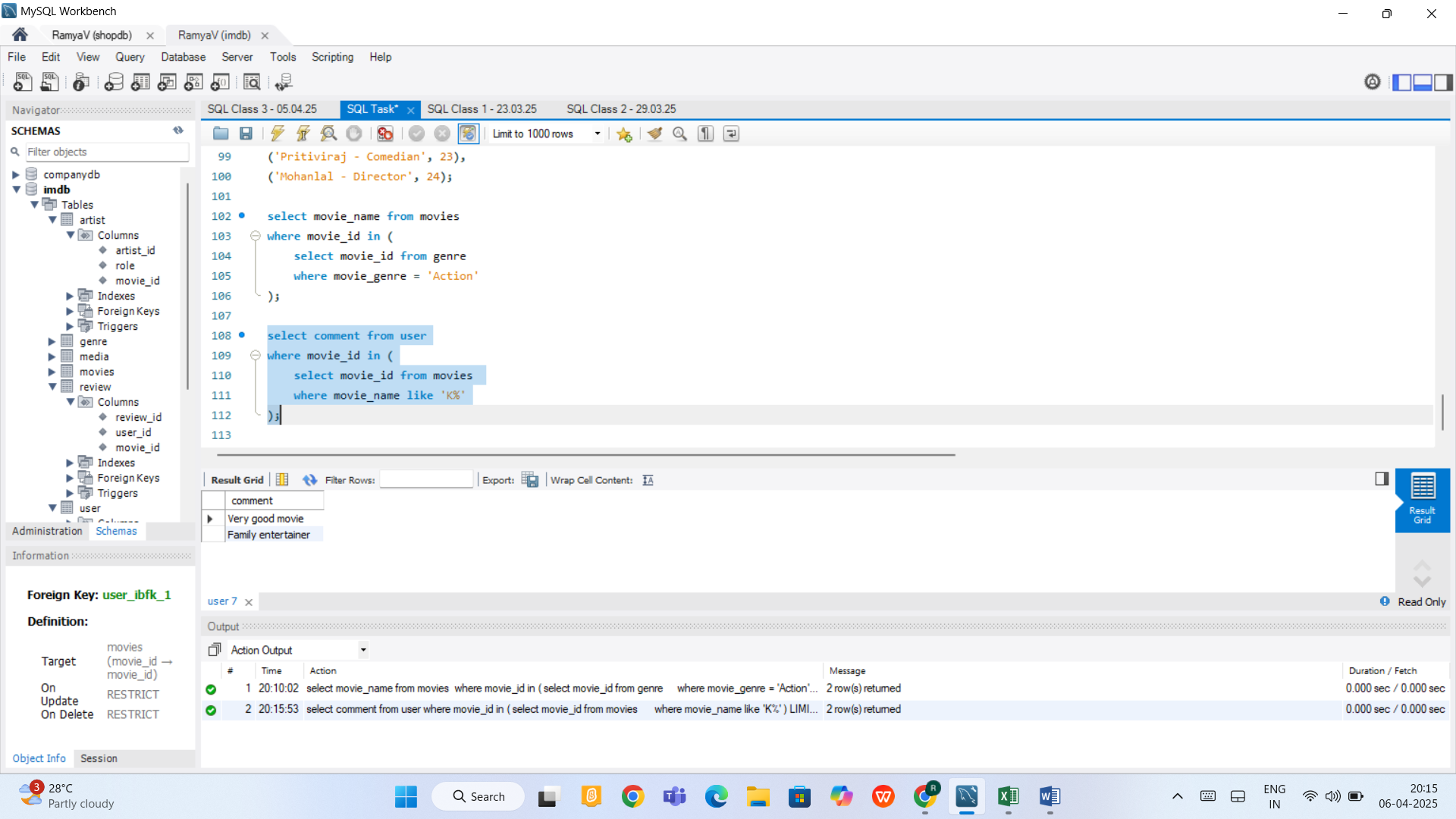
Executing query to fetch movie name from genre table based on movie genre as action

**Using Where condition, Sub query, Foreign key concept**

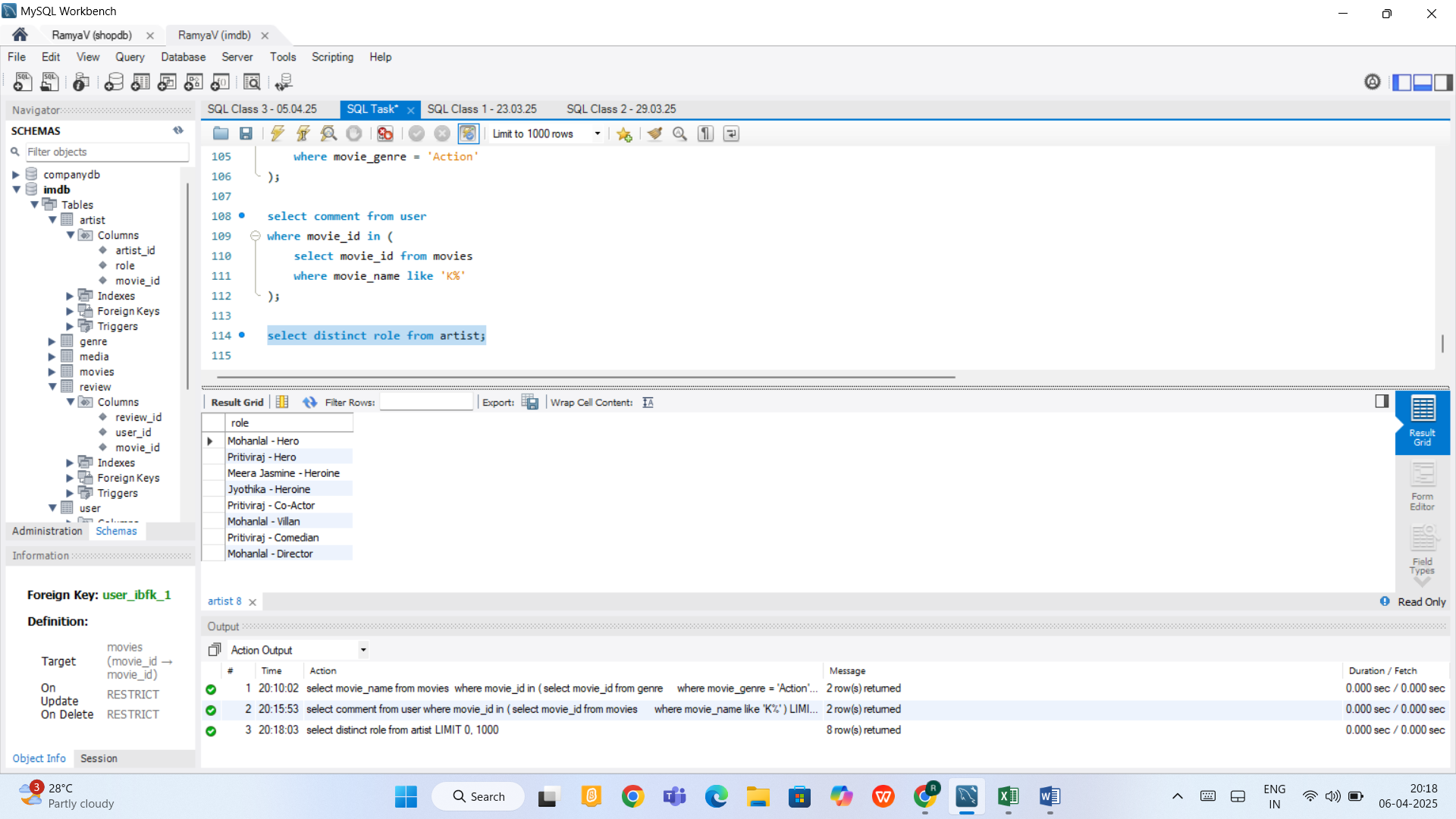


Executing query to fetch comment from user table for movies which start with K

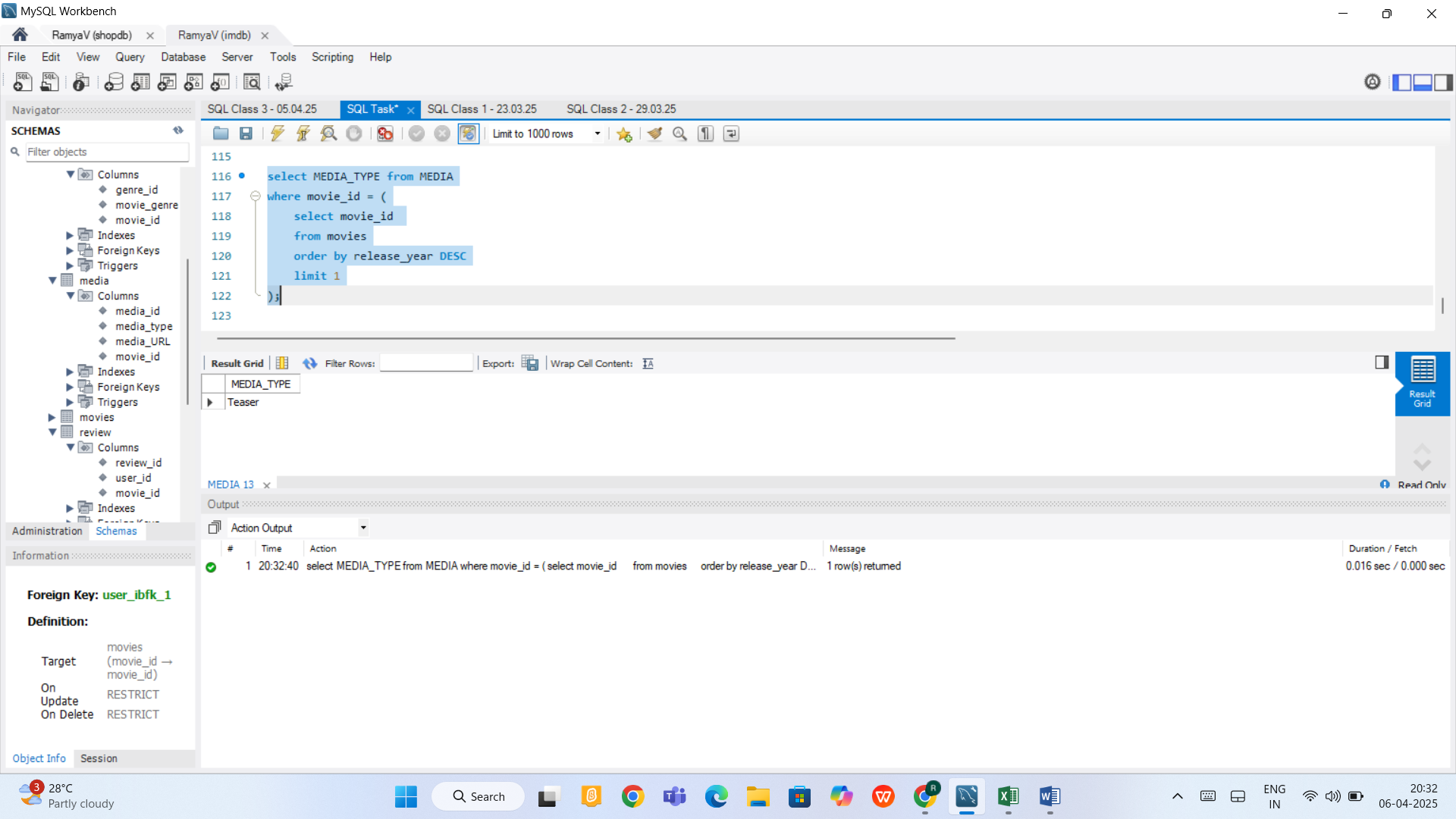
**Using Where condition, Sub query, Foreign key concept, Like**



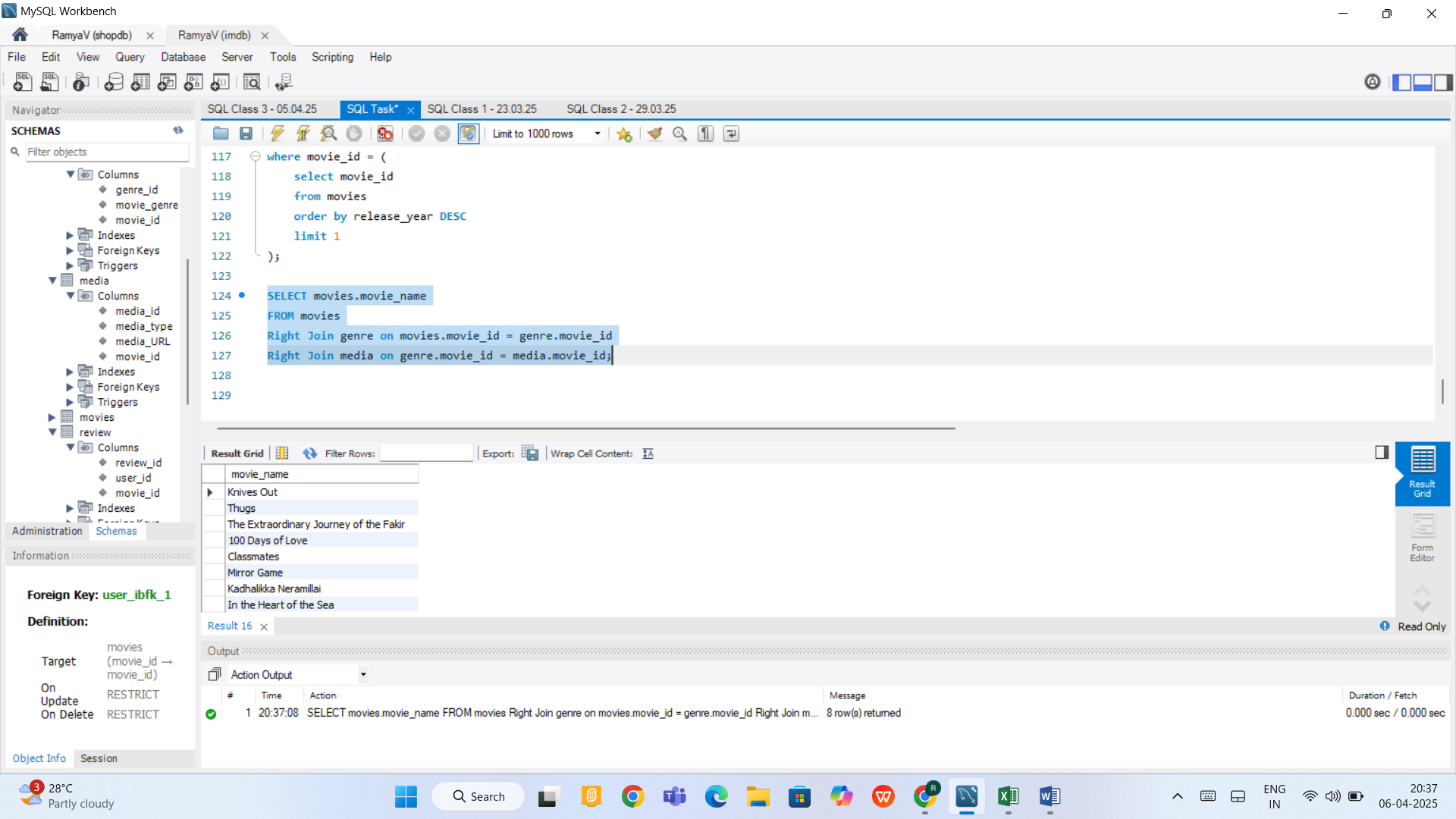
Selecting distinct values from artist table



Extracting data using order by DESC and using limit keyword



Using Join to fetch data



-- MySQL Queries performed in Database copied here

-- Creating database and tables in MySQL

CREATE DATABASE IF NOT EXISTS IMDB;

USE IMDB;

CREATE TABLE movies (

movie\_id INT PRIMARY KEY AUTO\_INCREMENT,

movie\_name VARCHAR(100),

release\_year date

);

CREATE TABLE media (

media\_id INT PRIMARY KEY AUTO\_INCREMENT,

media\_type VARCHAR(50),

media\_URL VARCHAR(100),

movie\_id INT,

FOREIGN KEY (movie\_id) REFERENCES movies(movie\_id)

);

CREATE TABLE genre (

genre\_id INT PRIMARY KEY AUTO\_INCREMENT,

movie\_genre VARCHAR(50),

movie\_id INT,

FOREIGN KEY (movie\_id) REFERENCES movies(movie\_id)

);

CREATE TABLE user (

user\_id INT PRIMARY KEY AUTO\_INCREMENT,

comment VARCHAR(100),

movie\_id INT,

FOREIGN KEY (movie\_id) REFERENCES movies(movie\_id)

);

CREATE TABLE review (

review\_id INT PRIMARY KEY AUTO\_INCREMENT,

user\_id INT,

movie\_id INT,

FOREIGN KEY (movie\_id) REFERENCES movies(movie\_id),

FOREIGN KEY (user\_id) REFERENCES user(user\_id)

);

CREATE TABLE artist (

artist\_id INT PRIMARY KEY AUTO\_INCREMENT,

role VARCHAR(50),

movie\_id INT,

FOREIGN KEY (movie\_id) REFERENCES movies(movie\_id)

);

-- INSERTING VALUES INTO THE TABLES

INSERT INTO movies (movie\_name, release\_year) VALUES

('Knives Out', '2024-04-05'),

('Thugs', '2004-07-15'),

('The Extraordinary Journey of the Fakir', '2024-08-10'),

('100 Days of Love', '2023-01-30'),

('Classmates', '2014-01-30'),

('Mirror Game', '2003-05-03'),

('Kadhalikka Neramillai', '2008-11-13'),

('In the Heart of the Sea', '2025-02-28');

INSERT INTO media (media\_type, media\_URL, movie\_id) VALUES

('Trailer', 'URL1', '17'),

('Teaser', 'URL2', '18'),

('Trailer', 'URL3', '19'),

('Teaser', 'URL4', '20'),

('Poster', 'URL5', '21'),

('Trailer', 'URL6', '22'),

('Poster', 'URL7', '23'),

('Teaser', 'URL8', '24');

INSERT INTO genre (movie\_genre, movie\_id) VALUES

('Comedy', '17'),

('Action', '18'),

('Action', '19'),

('Musical', '20'),

('Romance', '21'),

('Mystery', '22'),

('Romance', '23'),

('Adventure', '24');

INSERT INTO user (comment, movie\_id) VALUES

('Very good movie', 17),

('Average movie', 18),

('Excellent movie, a must watch movie', 19),

('Nice movie', 20),

('Ok Ok movie', 21),

('Mystery movie, a must watch movie', 22),

('Family entertainer', 23),

('Thriller movie', 24);

INSERT INTO review (user\_id, movie\_id) VALUES

(17, 17), (18,18),(19,19),(20,20),(21,21),(22,22),(23,23),(24,24);

INSERT INTO artist (role, movie\_id) VALUES

('Mohanlal - Hero', 17),

('Pritiviraj - Hero', 18),

('Meera Jasmine - Heroine', 19),

('Jyothika - Heroine', 20),

('Pritiviraj - Co-Actor', 21),

('Mohanlal - Villan', 22),

('Pritiviraj - Comedian', 23),

('Mohanlal - Director', 24);

-- Fetching values from MySQL using SQL Queries

select movie\_name from movies

where movie\_id in (

select movie\_id from genre

where movie\_genre = 'Action'

);

select comment from user

where movie\_id in (

select movie\_id from movies

where movie\_name like 'K%'

);

select distinct role from artist;

select MEDIA\_TYPE from MEDIA

where movie\_id = (

select movie\_id

from movies

order by release\_year DESC

limit 1

);

SELECT movies.movie\_name

FROM movies

Right Join genre on movies.movie\_id = genre.movie\_id

Right Join media on genre.movie\_id = media.movie\_id;