01. (“Movie Popularity & Trends”) with the **DVD Rental database** is perfect for building a full student examination project. Below I’ve designed a **complete set of 30 questions**, ranging from **basic SQL queries** to **advanced analytics and optimization tasks**

🎬 Project 2: Movie Popularity & Trends

Database: DVD Rental (film, rental, inventory, payment, customer, store, etc.)
Objective: Analyze film popularity, rental frequency, and revenue trends using SQL.

🟢 Section A: Basic Queries (1–10)

1. List all films with their titles and release years.
2. Find the total number of films in the database.
3. Show the names of all customers who rented at least one film.
4. Retrieve the titles of films in the “Action” category.
5. Count how many films belong to each category.
6. Display the top 5 longest films by length.
7. Find the number of rentals made in January 2006.
8. Show all films that have rental rates greater than $3.
9. List all distinct cities where customers live.
10. Find the staff members who processed rentals.

🟡 Section B: Intermediate Queries (11–20)

11. Find the top 10 films by number of rentals.
12. Show the revenue generated by each film (sum of payments).
13. List customers who rented more than 20 films.
14. Find the average rental duration for each category.
15. Display films that were rented but never returned late.
16. Identify the most popular film category by rental count.
17. Show the monthly rental count trend for 2005–2006.
18. Find customers who rented films from more than 5 categories.
19. List films that were rented by customers from Canada.
20. Show the top 5 customers by total payment amount.

🔴 Section C: Advanced Queries (21–30)

21. Use a **window function** to rank films by rental frequency.
22. Create a **CTE** to calculate monthly revenue and display the top 3 months.
23. Find films that generated revenue above the overall average.
24. Write a query to calculate **customer lifetime value** (total payments per customer).
25. Identify films that were rented in consecutive months.
26. Create a **materialized view** showing film popularity by category.
27. Write a **stored procedure** to calculate revenue for a given film ID.
28. Build a **trigger** that logs whenever a rental is returned late.
29. Partition rentals by year and calculate yearly revenue.
30. Optimize a query using **indexes** to speed up film search by title.

📊 Deliverables for Students

- **SQL Scripts**: Queries for all 30 questions.
- **Outputs**: Screenshots or result sets.
- **Documentation**: Short explanation of logic used.
- **Bonus Task**: Visualize monthly rental trends in a BI tool (Power BI, Tableau, or PostgreSQL charts).

✨ This structure ensures students progress from **basic SELECT statements** to **advanced SQL concepts** like **CTEs, window functions, triggers, and materialized views**. It's exam-ready and covers the full spectrum of PostgreSQL skills.