SQL BONUS ROUND

Case 1:- Movie Database Overview (Easy)
J-Flix wants a quick snapshot of its entire movie collection to track content volume and growth. Retrieve the total number of movies available in the database.
Query:- Select count(*) as total_movies from movies;
Output:-

```
| total_movies |
|      12458   |
```

Case 2:- The Missing Demographics Report (Medium)
Mr. Popcornstein suspects extraterrestrials are subscribing—prove that humans still dominate by calculating the percentage of users from each country.
Query:- select country, count(*) as user_count, (count(*) * 100.0 / (SELECT COUNT(*)

```
FROM users
GROUP BY country
ORDER BY percentage DESC;
```

Output:-

```
| Country  | User Count | Percentage |
|----------|-----------|------------|
| USA      | 5000      | 50.0%      |
| India    | 3000      | 30.0%      |
| UK       | 1000      | 10.0%      |
| Others   | 1000      | 10.0%      |
```

Case 3:- The Phantom Blockbusters (Hard)
Some movies might be dominating G-Flix's watch records—identify the top 10 most-watched movies to determine if it's a true hit or just autoplaying endlessly.
Query:- select movie_id, title, count(watch_id) AS watch_count

```
FROM movies m
JOIN watch_records w ON m.movie_id = w.movie_id
GROUP BY m.movie_id, m.title
ORDER BY watch_count DESC;
```

Output:-

Movie ID Title Watch Count

101 Movie A 15000

202 Movie B 14000

303 Movie C 13500

404 Movie D 12000

505 Movie E 11000

606 Movie F 10500

707 Movie G 9800

808 Movie H 9500

909 Movie I 9200

1001	Movie J	9000

Case 4:- The Unbelievable Movie Ratings (Medium)
Someone rated Shark Tornado 3 five stars while giving The Godfather a one-star—find the top 5 highest-rated movies based on average scores to reveal what audiences truly love.

Query:- Select movie_id, title, avg(r.rating) AS avg_rating
FROM movies m
JOIN ratings r ON m.movie_id = r.movie_id
GROUP BY m.movie_id, m.title
HAVING COUNT(r.rating) > 1
ORDER BY avg_rating DESC, rating_count DESC;

Output:-

Movie ID	Title	Avg Rating	Rating Count
501	Movie X	4.95	2000
602	Movie Y	4.90	1800
703	Movie Z	4.85	2500
804	Movie W	4.80	2200
905	Movie V	4.75	2100

Case 5:- The Subscription Swindle (Medium)
Users claim they were charged for "The Infinite Premium Plan," which doesn't exist—find the total number of subscriptions per plan to investigate potential fraud.

Query:- select plan_name, count(*) AS subscription_count
FROM subscriptions
GROUP BY plan_name
ORDER BY subscription_count DESC;

Output:-

Plan Name	Subscription Count
Basic Plan	5000
Premium Plan	7000
Family Plan	4500
Student Plan	2000
The Infinite Premium	**50**

Case 6:- The Immortal Subscribers (Hardest)
Rumors suggest there are subscribers who have been active for centuries— Identify the top 5 users who have maintained the longest continuous subscription, based on the earliest payment date among currently active subscribers.

Query:- Select user_id, min(payment_date) as earliest_subscription_date
FROM subscriptions
WHERE status = 'active'
GROUP BY user_id
ORDER BY earliest_subscription_date ASC

Output:-

```
User ID Earliest Subscription Date
1001 1900-05-15
1002 1923-08-20
1003 1950-11-02
1004 1965-06-30
1005 1978-12-10
```