SQL Queries

**1 🔎 Case 01: 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.**

SELECT COUNT(\*) AS total\_movies

FROM Movies;

**Output: 5876**

**2 🌍 Case 02: 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.**

SELECT country,

ROUND(COUNT(\*) \* 100.0 / (SELECT COUNT(\*) FROM Users), 2) AS percentage

FROM Users

GROUP BY country

ORDER BY percentage DESC;

**Output:**

| **country** | **percentage** |
| --- | --- |
| **India** | **19.87** |
| **United States** | **16.87** |
| **United Kingdom** | **13.3** |
| **Australia** | **9.97** |
| **Japan** | **8.83** |
| **Canada** | **7.87** |
| **South Korea** | **5.6** |
| **France** | **3.4** |
| **Germany** | **3.2** |
| **Singapore** | **3.13** |
| **Brazil** | **2.23** |
| **Netherlands** | **2.2** |
| **Mexico** | **1.5** |
| **Italy** | **1.43** |
| **Spain** | **0.6** |

**3 🎥 Case 03: 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.**

SELECT m.title,

COUNT(w.watch\_id) AS watch\_count

FROM watch\_history w

JOIN movies m ON w.movie\_id = m.movie\_id

GROUP BY m.title

ORDER BY COUNT(w.watch\_id) DESC

LIMIT 10;

**Ouput:**

| **title** | **watch\_count** |
| --- | --- |
| **Star Wars: The Rise Of Skywalker** | **1213** |
| **Takers** | **1130** |
| **They Live** | **1024** |
| **Little Nemo: Adventures in Slumberland** | **1012** |
| **Green Room** | **1006** |
| **The House That Jack Built** | **997** |
| **Space Jam: A New Legacy** | **940** |
| **Rescue Dawn** | **939** |
| **Doctor Sleep** | **911** |
| **Splice** | **906** |

**4 ⭐ Case 04: 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.**

SELECT m.title,

AVG(r.rating) AS average\_rating

FROM ratings r

JOIN movies m ON r.movie\_id = m.movie\_id

GROUP BY m.title

ORDER BY AVG(r.rating) DESC

LIMIT 5;

**Output:**

| **title** | **average\_rating** |
| --- | --- |
| **20,000 Leagues Under the Sea** | **5** |
| **5 Centimeters per Second** | **5** |
| **A Christmas Carol** | **5** |
| **A Monster Calls** | **5** |
| **A Whisker Away** | **5** |

**5 💰 Case 05: 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.**

SELECT plan\_type,

COUNT(\*) AS total\_subscriptions

FROM Subscriptions

GROUP BY plan\_type

ORDER BY total\_subscriptions DESC;

**Output:**

| **plan\_type** | **total\_subscriptions** |
| --- | --- |
| **Standard** | **3097** |
| **Premium** | **2320** |
| **Student Plan** | **1785** |

6 🏆 Case 06: 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.

SELECT u.name,

MIN(s.payment\_date) AS earliest\_payment\_date

FROM Subscriptions s

JOIN Users u ON s.user\_id = u.user\_id

WHERE s.churn\_status = 'active'

GROUP BY u.name

ORDER BY MIN(s.payment\_date) ASC

LIMIT 5;

**Output:**

| **name** | **earliest\_payment\_date** |
| --- | --- |
| **Andrea Gomez** | **2024-06-01** |
| **Billy Cooper** | **2024-06-01** |
| **Blake Alexander** | **2024-06-01** |
| **Cathy Martinez** | **2024-06-01** |
| **Christina Wallace** | **2024-06-01** |