COMPLEX QUERIES

Presented below is a comprehensive suite of 22 advanced SQL queries designed to extract high-value business intelligence from an IMDb-style database. Crafted from the perspective of a lead data analyst, these queries move beyond simple data retrieval to address critical real-world business scenarios—ranging from financial ROI analysis and content strategy to user engagement tracking and talent metrics. By utilizing sophisticated SQL techniques such as Window Functions, Common Table Expressions (CTEs), subqueries, and complex multi-table joins, this collection demonstrates how raw data is transformed into actionable insights that drive strategic decision-making for a major media platform.

A) Financial & ROI Analysis

1) **The "Blockbuster" Efficiency Report (ROI by Genre)**
Business Goal: **Identify which genres provide the best Return on Investment (ROI) to guide future production budgets.**

```
SELECT
    g.Name AS Genre,
    COUNT(t.TitleID) AS Movie_Count,
    ROUND(AVG(bo.Budget), 2) AS Avg_Budget,
    ROUND(AVG(bo.Gross_Worldwide), 2) AS Avg_Global_Revenue,
    ROUND(((SUM(bo.Gross_Worldwide) - SUM(bo.Budget)) / NULLIF(SUM(bo.Budget), 0)) * 100, 2) AS ROI_Percentage
FROM Genre g
JOIN TitleGenre tg ON g.GenreID = tg.GenreID
JOIN Title t ON tg.TitleID = t.TitleID
JOIN BoxOffice bo ON t.TitleID = bo.TitleID
WHERE t.TitleType = 'Movie'
  AND bo.Budget > 0 -- Avoid division by zero
GROUP BY g.Name
HAVING COUNT(t.TitleID) > 2 -- Ignore niche genres with too little data
ORDER BY ROI_Percentage DESC;
```

Results Explain Describe Saved SQL History

GENRE	MOVIE_COUNT	AVG_BUDGET	AVG_GLOBAL_REVENUE	ROI_PERCENTAGE
Thriller	3	21133333.33	286276142	1254.62
Fantasy	3	57333333.33	526692114.67	818.65
Drama	11	33218181.82	263189281.09	692.3
Crime	8	42412500	303651176.13	615.95
Action	3	136000000	768304264.67	464.93
Sci-Fi	3	129333333.33	667361185.33	416

2) Top Grossing Production Companies (Weighted)

Business Goal: Find the most valuable production partners. Only consider companies with at least titles.

```
SELECT
    c.Name AS Company_Name,
    c.Company_Type,
    COUNT(tc.TitleID) AS Projects_Count,
    TO_CHAR(SUM(bo.Gross_Worldwide), '$999,999,999,999') AS Total_Revenue,
    RANK() OVER (ORDER BY SUM(bo.Gross_Worldwide) DESC) AS Revenue_Rank
FROM Company c
JOIN TitleCompany tc ON c.CompanyID = tc.CompanyID
JOIN BoxOffice bo ON tc.TitleID = bo.TitleID
GROUP BY c.Name, c.Company_Type
HAVING COUNT(tc.TitleID) >= 3
ORDER BY Revenue_Rank ASC;
```

Results Explain Describe Saved SQL History

COMPANY_NAME	COMPANY_TYPE	PROJECTS_COUNT	TOTAL_REVENUE	REVENUE_RANK
Warner Bros.	Production	42	$9,480,074,115	1

1 rows returned in 0.17 seconds Download

3. Revenue Per Minute (Pacing Efficiency)

Business Goal: **Which movies generated the most money per minute of runtime? (Often used for ad-revenue calculations/estimates).**

```
SELECT
    t.Primary_Title,
    t.Runtime_Minutes,
    bo.Gross_Worldwide,
    ROUND(bo.Gross_Worldwide / NULLIF(t.Runtime_Minutes, 0), 2) AS Revenue_Per_Minute
FROM Title t
JOIN BoxOffice bo ON t.TitleID = bo.TitleID
WHERE t.Runtime_Minutes > 60
ORDER BY Revenue_Per_Minute DESC
```

Results Explain Describe Saved SQL History

PRIMARY_TITLE	RUNTIME_MINUTES	GROSS_WORLDWIDE	REVENUE_PER_MINUTE
The Lion King	88	968511805	11005815.97
The Dark Knight	152	1004558444	6608937.13
Inception	148	836836967	5654303.83
WALL-E	98	521311860	5319508.78
The Lord of the Rings: The Fellowship of the Ring	178	897690072	5043202.65
Forrest Gump	142	678226133	4776240.37
Interstellar	169	701729206	4152243.82
Terminator 2: Judgment Day	137	520881154	3802052.22
The Matrix	136	463517383	3408216.05
Raiders of the Lost Ark	115	389925971	3390660.62

B. Talent & Casting Analytics

4. The "Director-Actor Power Couple" Analysis

Business Goal: **Identify Director/Actor pairs that collaborate frequently. Used for casting recommendations.**

```
SELECT
    Director.Name AS Director_Name,
    Actor.Name AS Actor_Name,
    COUNT(*) AS Collaboration_Count,
    LISTAGG(t.Primary_Title, ', ') WITHIN GROUP (ORDER BY t.Start_Year DESC) AS Movies
FROM Credit c1
JOIN Credit c2 ON c1.TitleID = c2.TitleID
JOIN Person Director ON c1.PersonID = Director.PersonID
JOIN Person Actor ON c2.PersonID = Actor.PersonID
JOIN Title t ON c1.TitleID = t.TitleID
WHERE c1.Credit_Role = 'Director'
  AND c2.Credit_Role = 'Actor'
GROUP BY Director.Name, Actor.Name
HAVING COUNT(*) >= 2
ORDER BY Collaboration_Count DESC;
```

Results Explain Describe Saved SQL History

DIRECTOR_NAME	ACTOR_NAME	COLLABORATION_COUNT	MOVIES
David Fincher	Brad Pitt	2	Fight Club, Seven

5. "One-Hit Wonder" vs. "Consistent Star" (Actor Volatility)

Business Goal: **Identify actors whose movies vary wildly in quality versus those who deliver consistent ratings**

```
SELECT
    p.Name,
    COUNT(c.TitleID) AS Movie_Count,
    ROUND(AVG(rs.Average_Rating), 2) AS Avg_Rating,
    ROUND(STDDEV(rs.Average_Rating), 2) AS Rating_Volatility_Score
FROM Person p
JOIN Credit c ON p.PersonID = c.PersonID
JOIN RatingSummary rs ON c.TitleID = rs.TitleID
WHERE c.Credit_Role = 'Actor'
GROUP BY p.Name
HAVING COUNT(c.TitleID) >= 3
ORDER BY Avg_Rating DESC, Rating_Volatility_Score ASC;
```

Results Explain Describe Saved SQL History

NAME	MOVIE_COUNT	AVG_RATING	RATING_VOLATILITY_SCORE
Morgan Freeman	3	8.33	.58

6. Most Versatile Talent (Genre Diversity)

Business Goal: **Which directors have successfully worked across the widest variety of genres?**

```
SELECT
    p.Name,
    COUNT(DISTINCT tg.GenreID) AS Unique_Genres,
    LISTAGG( g.Name, ', ') WITHIN GROUP (ORDER BY g.Name) AS Genre_List
FROM Person p
JOIN Credit c ON p.PersonID = c.PersonID
JOIN TitleGenre tg ON c.TitleID = tg.TitleID
JOIN Genre g ON tg.GenreID = g.GenreID
WHERE c.Credit_Role = 'Director'
GROUP BY p.Name
HAVING COUNT(DISTINCT tg.GenreID) >= 3
ORDER BY Unique_Genres DESC;
```

Results Explain Describe Saved SQL History

NAME	UNIQUE_GENRES	GENRE_LIST
Christopher Nolan	4	Action, Action, Adventure, Crime, Sci-Fi, Sci-Fi
David Fincher	3	Crime, Drama, Thriller

C. Content Strategy & Library Health

7. The "Cult Classic" Detector

Business Goal: **Identify movies with low box office performance (flops) but high user ratings and review engagement. These are prime candidates for licensing or remakes**

```
WITH Global_Stats AS (
    SELECT AVG(Gross_Worldwide) as Global_Avg_Gross FROM BoxOffice
)
SELECT
    t.Primary_Title,
    t.Start_Year,
    rs.Average_Rating,
    rs.Num_Votes,
    bo.Gross_Worldwide
FROM Title t
JOIN RatingSummary rs ON t.TitleID = rs.TitleID
JOIN BoxOffice bo ON t.TitleID = bo.TitleID
CROSS JOIN Global_Stats gs
WHERE rs.Average_Rating >= 8.0
  AND bo.Gross_Worldwide < (gs.Global_Avg_Gross * 0.5)
  AND t.Start_Year < EXTRACT(YEAR FROM SYSDATE) - 5;
```

Results Explain Describe Saved SQL History

PRIMARY_TITLE	START_YEAR	AVERAGE_RATING	NUM_VOTES	GROSS_WORLDWIDE
The Shawshank Redemption	1994	9	2	28817291
Fight Club	1999	8	1	101209702
Goodfellas	1990	8	1	47103304
City of God	2002	8	1	30641770
Psycho	1960	8	1	50000000
The Pianist	2002	8	1	120072577
Leon: The Professional	1994	8	1	45501238
Whiplash	2014	8	1	49396747
The Prestige	2006	8	1	109676311
Casablanca	1942	8	1	4000000
Rear Window	1954	8	1	37000000
Alien	1979	8	1	106285522
Apocalypse Now	1979	8	1	98000000
Memento	2000	8	1	40047236
The Lives of Others	2006	8	1	77356942
Oldboy	2003	8	1	15000000
Breaking Bad	2008	8	1	0

8. Regional Content Dominance

Business Goal: **Find the top-rated Genre for every Country**

```
WITH CountryGenreStats AS (
    SELECT
        cn.Name AS Country,
        g.Name AS Genre,
        COUNT(t.TitleID) AS Title_Count,
        RANK() OVER (PARTITION BY cn.Name ORDER BY COUNT(t.TitleID) DESC) as Rank_In_Countr
    FROM Country cn
    JOIN Title t ON cn.CountryID = t.CountryID
    JOIN TitleGenre tg ON t.TitleID = tg.TitleID
    JOIN Genre g ON tg.GenreID = g.GenreID
    GROUP BY cn.Name, g.Name
)
SELECT Country, Genre, Title_Count
FROM CountryGenreStats
WHERE Rank_In_Country = 1
ORDER BY Title_Count DESC;
```

Results Explain Describe Saved SQL History

COUNTRY	GENRE	TITLE_COUNT
United States	Drama	12
Italy	Comedy	1
Italy	War	1
Italy	Drama	1
Japan	Fantasy	1
New Zealand	Adventure	1
New Zealand	Fantasy	1
South Korea	Thriller	1
South Korea	Drama	1
Brazil	Crime	1
Japan	Animation	1

9. Streaming Platform War (Content Overlap)

Business Goal: **Which titles are available on the highest number of distinct streaming platforms? (Indicating non-exclusivity). // BUT ACCORDING TO OUR DATA HIGHEST NUMBER OF DISTINCT STREAMING PLATFORMS ARE 1 ONLY.**

```
SELECT
    t.Primary_Title,
    COUNT(DISTINCT sa.ProviderID) AS Platform_Count,
    LISTAGG(sp.Name, ' | ') WITHIN GROUP (ORDER BY sp.Name) AS Platforms
FROM StreamingAvailability sa
JOIN Title t ON sa.TitleID = t.TitleID
JOIN StreamingProvider sp ON sa.ProviderID = sp.ProviderID
WHERE sa.Deleted_At IS NULL
GROUP BY t.Primary_Title
HAVING COUNT(DISTINCT sa.ProviderID) >= 1
ORDER BY Platform_Count DESC;
```

Results Explain Describe Saved SQL History

PRIMARY_TITLE	PLATFORM_COUNT	PLATFORMS
Alien	1	Hulu
Whiplash	1	Disney+
Back to the Future	1	Netflix
Casablanca	1	Netflix
City of God	1	Netflix
Django Unchained	1	Amazon Prime Video
Fight Club	1	Netflix | Netflix
Forrest Gump	1	Amazon Prime Video
Gladiator	1	Amazon Prime Video
Goodfellas	1	Disney+
Inception	1	Netflix
Interstellar	1	Hulu | Hulu
Leon: The Professional	1	HBO Max
Life is Beautiful	1	Amazon Prime Video
Memento	1	HBO Max
Oldboy	1	HBO Max

D. User Engagement & Community

10. The "Super-User" Identification (Triple Threat)

Business Goal: **Identify users who are deeply engaged: they Rate, Review, AND create Public Lists.**

```
SELECT u.Username, u.JoinedDate
FROM User_def u
WHERE u.UserID IN (SELECT UserID FROM UserRating)
  AND u.UserID IN (SELECT UserID FROM Review)
  AND u.UserID IN (SELECT UserID FROM List WHERE Is_Public = 1);
```

Results Explain Describe Saved SQL History

USERNAME	JOINEDDATE
moviebuff01	01/01/2022
cinephile99	01/02/2022
critic_joe	01/03/2022
action_fan	01/05/2022
drama_queen	01/06/2022
scifi_nerd	01/07/2022
comedy_king	01/09/2022
docu_viewer	01/10/2022
user11	07/25/2024
user12	08/04/2024
user13	08/14/2024

11.Review Sentiment Gap (Controversial Movies)

Business Goal: **Find movies where the Critics (Review Text ratings) disagree with the Masses (One-click Star Ratings)**

// DUE TO OUR DATA RESTRICTIONS WE HAVE TO RESTRICT TO ONLY MINIMUM MASS 1

```
SELECT
    t.Primary_Title,
    rs.Average_Rating AS Mass_Rating,
    ROUND(AVG(rev.Rating), 1) AS Reviewer_Rating,
    ABS(rs.Average_Rating - AVG(rev.Rating)) AS Controversy_Gap
FROM Title t
JOIN RatingSummary rs ON t.TitleID = rs.TitleID
JOIN Review rev ON t.TitleID = rev.TitleID
GROUP BY t.Primary_Title, rs.Average_Rating
HAVING COUNT(rev.ReviewID) > 1 -- Minimum sample size
ORDER BY Controversy_Gap DESC;
```

Results Explain Describe Saved SQL History

PRIMARY_TITLE	MASS_RATING	REVIEWER_RATING	CONTROVERSY_GAP
Breaking Bad	8	9.5	1.5
Pulp Fiction	8.5	7	1.5
The Dark Knight	9.3	8.5	.8
The Lord of the Rings: The Fellowship of the Ring	8	7.5	.5
The Shawshank Redemption	9	8.5	.5
Interstellar	8	8.5	.5
Inception	9.5	9	.5
The Godfather	9	9	0

12. Top 3 Ranked Movies Per Decade

Business Goal: **Generate a "Hall of Fame" list by decade.**

HERE ALSO WE LIMIT NUM_VOTES TO 1

```
SELECT * FROM (
    SELECT
        FLOOR(Start_Year / 10) * 10 || 's' AS Decade,
        Primary_Title,
        Start_Year,
        Average_Rating,
        DENSE_RANK() OVER (PARTITION BY FLOOR(Start_Year / 10) * 10 ORDER BY Average_Rating DESC) as Rank
    FROM Title t
    JOIN RatingSummary rs ON t.TitleID = rs.TitleID
    WHERE t.TitleType = 'Movie'
      AND rs.Num_Votes > 1
)
WHERE Rank <= 3
ORDER BY Decade DESC, Rank ASC;
```

Results Explain Describe Saved SQL History

DECADE	PRIMARY_TITLE	START_YEAR	AVERAGE_RATING	RANK
2010s	Inception	2010	9.5	1
2010s	Parasite	2019	8.5	2
2010s	Interstellar	2014	8	3
2000s	The Dark Knight	2008	9.3	1
2000s	Spirited Away	2001	9	2
1990s	The Shawshank Redemption	1994	9	1
1990s	Pulp Fiction	1994	8.5	2
1970s	The Godfather	1972	9	1

E. Operational & Data Quality

13. "Orphaned" Credits Check

Business Goal: **Data Integrity Check. Find credits linked to a Person or Character that has been soft-deleted.**

// WE DO IT JUST FOR THE SAKE OF THE IMPORTANCE OF SOFT DELETENESS IN OUR DATABASE

```
SELECT
    c.CreditID,
    c.TitleID,
    p.Name AS Person_Name,
    ch.CharacterName
FROM Credit c
LEFT JOIN Person p ON c.PersonID = p.PersonID
LEFT JOIN Character ch ON c.CharacterID = ch.CharacterID
WHERE (p.Deleted_At IS NOT NULL)
   OR (ch.Deleted_At IS NOT NULL);
```

Results Explain Describe Saved SQL History

CREDITID	TITLEID	PERSON_NAME	CHARACTERNAME
12	4	John Travolta	Vincent Vega
31	10	Elijah Wood	Frodo Baggins
40	21	Christopher Lloyd	-

14. The "Oscar Bait" Calendar

Business Goal: **Analyze which release month produces the most award winners.**

```
SELECT
    TO_CHAR(t.Release_Date, 'Month') AS Release_Month,
    COUNT(n.NominationID) AS Wins_Count
FROM Nomination n
JOIN Title t ON n.NomineeTitleID = t.TitleID
WHERE n.Is_Winner = 1
  AND t.Release_Date IS NOT NULL
GROUP BY TO_CHAR(t.Release_Date, 'Month'), EXTRACT(MONTH FROM t.Release_Date)
ORDER BY EXTRACT(MONTH FROM t.Release_Date);
```

Results Explain Describe Saved SQL History

RELEASE_MONTH	WINS_COUNT
January	2
May	1

F. Complex Deep-Dives

15. The "Streaming Gem" Hunter

Business Goal: **Find movies available on 'Netflix' that have a high rating (>8) but low vote counts (<5000), indicating a hidden gem for recommendation algorithms.**

```
SELECT
    t.Primary_Title,
    g.Name AS Primary_Genre,
    rs.Average_Rating,
    rs.Num_Votes
FROM Title t
JOIN RatingSummary rs ON t.TitleID = rs.TitleID
JOIN TitleGenre tg ON t.TitleID = tg.TitleID
JOIN Genre g ON tg.GenreID = g.GenreID
WHERE t.TitleID IN (
    SELECT sa.TitleID
    FROM StreamingAvailability sa
    JOIN StreamingProvider sp ON sa.ProviderID = sp.ProviderID
    WHERE UPPER(sp.Name) LIKE '%NETFLIX%'
      AND SYSDATE BETWEEN sa.Start_Date AND NVL(sa.End_Date, SYSDATE)
)
```

Results Explain Describe Saved SQL History

PRIMARY_TITLE	PRIMARY_GENRE	AVERAGE_RATING	NUM_VOTES
Inception	Action	9.5	2
Inception	Sci-Fi	9.5	2
Parasite	Drama	8.5	2
Parasite	Thriller	8.5	2
Fight Club	Drama	8	1
City of God	Crime	8	1

16. Cast Size vs. Success

Business Goal: **Does having a larger cast correlation with higher Box Office?**

```
WITH CastCounts AS (
    SELECT TitleID, COUNT(PersonID) as Cast_Size
    FROM Credit
    WHERE Credit_Role = 'Actor'
    GROUP BY TitleID
)
SELECT
    CASE
        WHEN cc.Cast_Size < 5 THEN 'Small Ensemble'
        WHEN cc.Cast_Size BETWEEN 5 AND 15 THEN 'Medium Ensemble'
        ELSE 'Large Ensemble'
    END AS Cast_Category,
    ROUND(AVG(bo.Gross_Worldwide), 0) AS Avg_Revenue
FROM CastCounts cc
JOIN BoxOffice bo ON cc.TitleID = bo.TitleID
GROUP BY
    CASE
        WHEN cc.Cast_Size < 5 THEN 'Small Ensemble'
        WHEN cc.Cast_Size BETWEEN 5 AND 15 THEN 'Medium Ensemble'
        ELSE 'Large Ensemble'
    END
ORDER BY Avg_Revenue DESC;
```

Results Explain Describe Saved SQL History

CAST_CATEGORY	AVG_REVENUE
Small Ensemble	325839249

17. Keyword Trend Analysis (The "Time Travel" Trend)

Business Goal: **How has the popularity of specific keywords (like "Time Travel") evolved over decades?**

// QUERYING YOUR SPECIFIC KEYWORD POPULARITY

```
SELECT
    FLOOR(t.Start_Year / 10) * 10 AS Decade,
    COUNT(t.TitleID) AS Titles_With_Keyword
FROM Title t
JOIN TitleKeyword tk ON t.TitleID = tk.TitleID
JOIN Keyword k ON tk.KeywordID = k.KeywordID
WHERE UPPER(k.KeywordText) LIKE '%TIME TRAVEL%'
GROUP BY FLOOR(t.Start_Year / 10) * 10
ORDER BY Decade;
```

Results Explain Describe Saved SQL History

DECADE	TITLES_WITH_KEYWORD
1980	1

18. "Language Barrier" Report

Business Goal: **Movies that performed well domestically (US) but failed internationally (or vice versa).**

```
SELECT
    t.Primary_Title,
    bo.Gross_Domestic,
    bo.Gross_Worldwide - bo.Gross_Domestic AS Gross_International,
    ROUND(bo.Gross_Domestic / bo.Gross_Worldwide, 2) * 100 AS Domestic_Share_Pct
FROM BoxOffice bo
JOIN Title t ON bo.TitleID = t.TitleID
WHERE bo.Gross_Worldwide > 10000000
ORDER BY Domestic_Share_Pct DESC;
```

Results | Explain | Describe | Saved SQL | History

PRIMARY_TITLE	GROSS_DOMESTIC	GROSS_INTERNATIONAL	DOMESTIC_SHARE_PCT
Rear Window	36764313	235687	99
Goodfellas	46836394	266910	99
The Shawshank Redemption	28341469	475822	98
Apocalypse Now	83471511	14528489	85
Alien	80931801	25353721	76
Memento	25544867	14502369	64
Raiders of the Lost Ark	248159971	141766000	64
Psycho	32000000	18000000	64
The Godfather	134966411	111154563	55
Back to the Future	210609762	178190238	54

More than 10 rows available. Increase rows selector to view more rows.

19. Production Cycle Duration

Business Goal: **Calculate the time gap between a movie's release and the company's newest project (Operational cadence).** ***Modified for available data: Time between Start Year and Release Date (Delay Analysis).***

```
SELECT
    t.Primary_Title,
    t.Start_Year,
    t.Release_Date,
    (t.Release_Date - TO_DATE(t.Start_Year || '-01-01', 'YYYY-MM-DD')) AS Days_Delay
FROM Title t
WHERE t.Release_Date IS NOT NULL
  AND EXTRACT(YEAR FROM t.Release_Date) > t.Start_Year
ORDER BY Days_Delay DESC;
```

Results | Explain | Describe | Saved SQL | History

PRIMARY_TITLE	START_YEAR	RELEASE_DATE	DAYS_DELAY
The Prestige	2006	11/24/2008	1058
Generic Episode	2020	08/09/2022	951
Forrest Gump	1994	07/18/1996	929
Generic Episode	2020	06/21/2022	902
Gray Matter	2008	06/11/2010	892
The Vanishing of Will Byers	2020	05/27/2022	877
Lord Snow	2020	05/21/2022	871
Interstellar	2014	05/12/2016	862
Seven	1995	04/12/1997	832
Alien	1979	04/08/1981	828

More than 10 rows available. Increase rows selector to view more rows.

20. Top 10 Most Helpful Reviewers

Business Goal: **Identify community leaders whose reviews are marked as 'Helpful' most often.**

```
SELECT
    u.Username,
    COUNT(r.ReviewID) AS Reviews_Posted,
    SUM(r.Helpful_Count) AS Total_Helpful_Votes,
    ROUND(SUM(r.Helpful_Count) / COUNT(r.ReviewID), 1) AS Avg_Helpful_Per_Review
FROM User_def u
JOIN Review r ON u.UserID = r.UserID
GROUP BY u.Username
HAVING SUM(r.Helpful_Count) > 0
ORDER BY Total_Helpful_Votes DESC
```

Results | Explain | Describe | Saved SQL | History

USERNAME	REVIEWS_POSTED	TOTAL_HELPFUL_VOTES	AVG_HELPFUL_PER_REVIEW
action_fan	2	75	37.5
comedy_king	2	70	35
moviebuff01	2	65	32.5
scifi_nerd	2	63	31.5
cinephile99	2	58	29
user32	1	49	49
docu_viewer	2	45	22.5
horror_lover	2	45	22.5
user39	1	44	44
user12	1	43	43

21. Genre combinations (The "Action-Comedy" overlap)

Business Goal: **Matrix analysis. How often do two specific genres appear together?**

```
SELECT
    g1.Name AS Genre_1,
    g2.Name AS Genre_2,
    COUNT(*) AS Co_Occurrence
FROM TitleGenre tg1
JOIN TitleGenre tg2 ON tg1.TitleID = tg2.TitleID AND tg1.GenreID < tg2.GenreID
JOIN Genre g1 ON tg1.GenreID = g1.GenreID
JOIN Genre g2 ON tg2.GenreID = g2.GenreID
GROUP BY g1.Name, g2.Name
ORDER BY Co_Occurrence DESC
```

Results | Explain | Describe | Saved SQL | History

GENRE_1	GENRE_2	CO_OCCURRENCE
Crime	Drama	5
Drama	Fantasy	3
Drama	War	2
Action	Sci-Fi	2
Crime	Thriller	2
Adventure	Fantasy	2
Comedy	Drama	1
Drama	Romance	1
Fantasy	Horror	1
Action	Fantasy	1

22. "Empty" Titles Audit (Data Cleaning)

Business Goal: **Find Titles that have no Cast, no Crew, and no Genres. (Zombie records).**

```
SELECT t.TitleID, t.Primary_Title
FROM Title t
WHERE NOT EXISTS (SELECT 1 FROM Credit c WHERE c.TitleID = t.TitleID)
  AND NOT EXISTS (SELECT 1 FROM TitleGenre tg WHERE tg.TitleID = t.TitleID);
```

Results Explain Describe Saved SQL History

TITLEID	PRIMARY_TITLE
121	Generic Episode
136	Generic Episode
133	Generic Episode
146	Generic Episode
104	Cancer Man
102	Cats in the Bag
49	Rick and Morty
28	The Departed
107	The Kingsroad
128	Generic Episode

THE END