# **OTT Streaming Analytics Dashboard** using SQL

## Project Description:

This project leverages SQL-based data analysis to derive actionable insights from streaming platform user data. It focuses on understanding user engagement, content popularity, device and platform usage, subscription behavior, and revenue trends. By querying watch history, user demographics, subscription records, and content metadata, the project provides a comprehensive overview of user activity patterns, content performance across genres and devices, retention metrics, and monetization breakdowns.

The insights obtained enable content curation, personalized recommendations, targeted marketing, user segmentation, retention strategy formulation, and performance monitoring—all essential for optimizing the performance and growth of an OTT streaming service.

**Key aspects covered include:**

* User viewing patterns by day, device, and platform
* Content popularity segmented by genre and age group
* User retention and churn analysis
* Revenue and subscription plan distribution
* Cross-device and multi-platform user engagement
* Peak usage times and geographic distribution

**Phase 1: Data Cleaning & Preparation (Messy Data Focus)**

1. **Fix Genre Inconsistencies**
   * Standardize genre naming (e.g., "sci fi", "SCI-FI", "Sci-Fi" → "Sci-Fi").
2. **Detect and Handle Duplicates**
   * Find duplicate records in users, content, and watch\_history.
3. **Identify Null or Invalid Watch Durations**
   * Find rows with null or 0 watch\_duration. How do we handle them?
4. **Detect Users with Overlapping Subscription Periods**
   * Any users with multiple active plans?
5. **Standardize Plan and Payment Formats**
   * Clean inconsistent plan types and payment modes.
6. **Missing Country or Age Group Entries**
   * Find and handle missing demographics in users table.

**Phase 2: Exploratory Data Analysis (EDA)**

1. **Top Genres Watched Overall and Per Platform**
   * Count of views grouped by genre and platform.

SELECT

u.platform,

c.genre,

COUNT(\*) AS total\_views

FROM

viewing\_history vh

JOIN

users u ON vh.user\_id = u.user\_id

JOIN

content c ON vh.content\_id = c.content\_id

GROUP BY

u.platform, c.genre

ORDER BY

u.platform, total\_views DESC;

**Insight:**  
We will discover which genres are most popular globally and how this varies across different platforms (e.g., Netflix may have more thrillers, Prime more comedies).

**Value:**

* Content acquisition: Invest in genres that are strong on each platform.
* Targeted promotion: Recommend genre-specific content to users on each platform.

1. **Most Used Device Types**
   * Which device is most commonly used to stream content?

SELECT

device\_type,

COUNT(\*) AS usage\_count

FROM

viewing\_history

GROUP BY

device\_type

ORDER BY

usage\_count DESC;

**Insight:**  
See which devices (e.g., mobile, smart TV, web) users prefer for streaming.

**Value:**

UI/UX investment: Optimize the app experiences on the most popular devices.

Marketing: Target device-specific ads or notifications.

1. **Average Watch Duration per Platform**
   * Breakdown of engagement across Netflix, Prime, etc.

SELECT

c.platform,

AVG(vh.duration\_watched) AS avg\_watch\_duration

FROM

viewing\_history vh

JOIN

content c ON vh.content\_id = c.content\_id

GROUP BY

c.platform

ORDER BY

avg\_watch\_duration DESC;

**Insight:**  
Identify differences in average engagement (watch duration) across platforms.

**Value:**

* Retention: Platforms with lower avg. watch durations may require better content or UI improvements.
* Partnerships: Negotiate advertising or content deals favoring high-engagement platforms.

1. **Monthly Active Users (MAU) Trend**
   * Count distinct users watching content each month.

SELECT

FORMAT(watch\_date, 'yyyy-MM') AS year\_month,

COUNT(DISTINCT user\_id) AS monthly\_active\_users

FROM

viewing\_history

GROUP BY

FORMAT(watch\_date, 'yyyy-MM')

ORDER BY

year\_month;

**Insight:**  
Track how your audience size changes over time — growth, stagnation, or decline.

**Value:**

* Business health: Steady MAU growth signals strong user engagement.
* Detect churn: Sudden drops indicate issues needing quick attention.

1. **Top 10 Most Watched Titles**
   * Based on total watch time.

SELECT TOP 10

c.title,

SUM(vh.duration\_watched) AS total\_watch\_time

FROM

viewing\_history vh

JOIN

content c ON vh.content\_id = c.content\_id

GROUP BY

c.title

ORDER BY

total\_watch\_time DESC;

**Insight:**  
Surface the most engaging titles based on total watch time, not just view counts.

**Value:**

* Recommendation algorithms: Promote trending or highly watched titles.
* Content investment: Produce/buy similar titles or renew popular series.

1. **User Retention Analysis**

**Are users coming back after signing up? And how long do they stay active?**

* + % of users active 30, 60, and 90 days after signup.

WITH activity\_flags AS (

SELECT

u.user\_id,

MAX(CASE WHEN DATEDIFF(DAY, u.signup\_date, vh.watch\_date) BETWEEN 1 AND 30 THEN 1 ELSE 0 END) AS active\_30,

MAX(CASE WHEN DATEDIFF(DAY, u.signup\_date, vh.watch\_date) BETWEEN 1 AND 60 THEN 1 ELSE 0 END) AS active\_60,

MAX(CASE WHEN DATEDIFF(DAY, u.signup\_date, vh.watch\_date) BETWEEN 1 AND 90 THEN 1 ELSE 0 END) AS active\_90

FROM

users u

LEFT JOIN

viewing\_history vh ON u.user\_id = vh.user\_id

GROUP BY

u.user\_id

),

retention\_summary AS (

SELECT

COUNT(\*) AS total\_users,

SUM(active\_30) AS retained\_30,

SUM(active\_60) AS retained\_60,

SUM(active\_90) AS retained\_90

FROM

activity\_flags

)

SELECT

total\_users,

retained\_30,

CAST(retained\_30 \* 100.0 / total\_users AS DECIMAL(5,2)) AS pct\_retained\_30,

retained\_60,

CAST(retained\_60 \* 100.0 / total\_users AS DECIMAL(5,2)) AS pct\_retained\_60,

retained\_90,

CAST(retained\_90 \* 100.0 / total\_users AS DECIMAL(5,2)) AS pct\_retained\_90

FROM

retention\_summary;

**Insight:**  
See what % of new users remain active 30, 60, and 90 days post-signup, giving you a clear measure of churn and stickiness.

**Value:**

* Lifecycle marketing: Launch retention campaigns targeting at-risk users.
* Feature prioritization: Investigate what retained users do differently.
* Goal setting: Benchmark your retention rates against industry standards.

1. **Binge Watching Pattern**
   * Users watching more than 3 episodes on the same day.

SELECT

vh.user\_id,

vh.watch\_date,

COUNT(\*) AS episodes\_watched

FROM

viewing\_history vh

JOIN

content c ON vh.content\_id = c.content\_id

WHERE

c.type = 'Episode' -- filter only episodic content (optional)

GROUP BY

vh.user\_id, vh.watch\_date

HAVING

COUNT(\*) > 3

ORDER BY

vh.watch\_date, vh.user\_id;

**Insight:**  
Identify which users are binge watchers (watching >3 episodes in a day), highlighting super-engaged users or viral content.

**Value:**

* Content dropping strategy: Consider releasing entire seasons at once if binge behavior is high.
* Loyalty programs: Reward binge-watchers to promote loyalty or encourage sharing.

1. **Genre Popularity by Age Group**
   * Preferred content types by age segments.

select

c.genre,

u.age\_group,

count(distinct wh.user\_id) as Total\_genre

from ott\_users u

join ott\_watch\_history wh

on u.user\_id = wh.user\_id

join ott\_content c

on wh.content\_id = c.content\_id

group by c.genre, u.age\_group

ORDER BY u.age\_group, Total\_genre DESC

**Insight:**  
Understand which age segments prefer which genres. You’ll learn, for example, that younger users might prefer action or animation, while older segments watch more drama or documentaries.

**Value:**

* Personalized recommendations: Tailor suggestions based on user age group.
* Marketing: Target age-specific campaigns or content promotions.
* Content production/acquisition: Invest in age-appropriate content.

**Phase 3: Analytical Insights (CTEs, Window Functions, Joins)**

1. **Rank Content by Watch Time Within Each Genre**
   * Use RANK() or DENSE\_RANK() over genre.

WITH content\_watch\_time AS (

SELECT

c.genre,

c.title,

SUM(vh.duration\_watched) AS total\_watch\_time

FROM

viewing\_history vh

JOIN

content c ON vh.content\_id = c.content\_id

GROUP BY

c.genre, c.title

),

ranked\_content AS (

SELECT

genre,

title,

total\_watch\_time,

RANK() OVER (PARTITION BY genre ORDER BY total\_watch\_time DESC) AS genre\_rank

FROM

content\_watch\_time

)

SELECT

genre,

title,

total\_watch\_time,

genre\_rank

FROM

ranked\_content

ORDER BY

genre, genre\_rank;

**Insight:**  
You will know which content items are most-watched **within each genre**, ranked by total watch time.

**Business Value:**

* **Content curation:** Highlight top-performing content in each genre to inform recommendations, homepage banners, and marketing.
* **Licensing & investments:** Guide future acquisitions or production investments toward genres/content that consistently rank high.

1. **User Quartiles Based on Total Watch Duration**
   * Use NTILE(4) to segment user engagement.

WITH user\_watch\_time AS (

SELECT

user\_id,

SUM(duration\_watched) AS total\_watch\_time

FROM

viewing\_history

GROUP BY

user\_id

),

quartiled\_users AS (

SELECT

user\_id,

total\_watch\_time,

NTILE(4) OVER (ORDER BY total\_watch\_time) AS engagement\_quartile

FROM

user\_watch\_time

)

SELECT

engagement\_quartile,

COUNT(\*) AS user\_count,

MIN(total\_watch\_time) AS min\_watch\_time,

MAX(total\_watch\_time) AS max\_watch\_time,

AVG(total\_watch\_time) AS avg\_watch\_time

FROM

quartiled\_users

GROUP BY

engagement\_quartile

ORDER BY

engagement\_quartile;

**Insight:**  
Users are grouped into four quartiles reflecting their total watch duration (from least to most engaged).

**Business Value:**

* **Segmentation:** Tailor campaigns, offers, or nudges to low-engagement users (1st quartile) and reward top customers (4th quartile).
* **Churn prediction:** Target the lowest quartile for re-engagement before they churn.
* **Personalization:** Prioritize features for high-quartile (engaged) users vs. onboarding for low-quartile users.

1. **Top Users by Platform (Window Function)**
   * ROW\_NUMBER() to get top watchers per platform.

WITH user\_platform\_watch\_time AS (

SELECT

c.platform,

vh.user\_id,

SUM(vh.duration\_watched) AS total\_watch\_time

FROM

viewing\_history vh

JOIN

content c ON vh.content\_id = c.content\_id

GROUP BY

c.platform, vh.user\_id

),

ranked\_users AS (

SELECT

platform,

user\_id,

total\_watch\_time,

ROW\_NUMBER() OVER (PARTITION BY platform ORDER BY total\_watch\_time DESC) AS rank\_per\_platform

FROM

user\_platform\_watch\_time

)

SELECT

platform,

user\_id,

total\_watch\_time

FROM

ranked\_users

WHERE

rank\_per\_platform = 1

ORDER BY

platform;

**Insight:**  
Identifies the single most engaged (highest watch time) user per platform.

**Business Value:**

* **Loyalty & advocacy:** Identify super-users suitable for ambassador or testimonial programs.
* **Platform comparison:** Understand which platforms generate your most valuable users for resource and UX prioritization.
* **User outreach:** Potential to collect feedback from top users on platform experience.

1. **Year-on-Year Growth in Subscribers and Revenue**
   * Window functions + LAG() to calculate growth rates.

WITH yearly\_metrics AS (

SELECT

YEAR(subscription\_date) AS year,

COUNT(DISTINCT user\_id) AS total\_subscribers,

SUM(amount) AS total\_revenue

FROM

subscriptions

GROUP BY

YEAR(subscription\_date)

),

yearly\_growth AS (

SELECT

year,

total\_subscribers,

total\_revenue,

LAG(total\_subscribers) OVER (ORDER BY year) AS prev\_year\_subscribers,

LAG(total\_revenue) OVER (ORDER BY year) AS prev\_year\_revenue

FROM

yearly\_metrics

)

SELECT

year,

total\_subscribers,

total\_revenue,

prev\_year\_subscribers,

(CAST(total\_subscribers - prev\_year\_subscribers AS FLOAT) / NULLIF(prev\_year\_subscribers, 0)) \* 100) AS subs\_growth\_pct,

prev\_year\_revenue,

(CAST(total\_revenue - prev\_year\_revenue AS FLOAT) / NULLIF(prev\_year\_revenue, 0)) \* 100 hgh AS revenue\_growth\_pct

FROM

yearly\_growth

ORDER BY

year;

**Insight:**  
Gives subscriber and revenue growth rates for each year, comparing current year values to those of the previous year.

**Business Value:**

* **Strategic planning:** Monitor business health and scalability.
* **Goal setting:** Set realistic targets based on historical growth.
* **Trend analysis:** Early detection of slowdowns or surges that may require intervention.

1. **Content Saturation**
   * How many unique users have watched a specific content?

SELECT

c.content\_id,

c.title,

COUNT(DISTINCT vh.user\_id) AS unique\_user\_count

FROM

viewing\_history vh

JOIN

content c ON vh.content\_id = c.content\_id

GROUP BY

c.content\_id, c.title

ORDER BY

unique\_user\_count DESC;

**Insight:**  
Shows how many unique users have watched each content item—measuring content reach.

**Business Value:**

* **Content lifecycle:** Identify evergreen titles vs. niche or underperforming content.
* **Promotional targeting:** Push content with low saturation to wider audiences, or re-market high-performing content.
* **Acquisition decisions:** Helps decide on renewals, removals, or similar content investments.

1. **\*\*\* Subscription Plan Distribution Over Time**
   * Monthly count of active users per plan type.

WITH months AS (

SELECT DATEFROMPARTS(y, m, 1) AS month\_start

FROM (

SELECT TOP 60

YEAR(DATEADD(MONTH, -ROW\_NUMBER() OVER (ORDER BY (SELECT NULL)), GETDATE())) AS y,

MONTH(DATEADD(MONTH, -ROW\_NUMBER() OVER (ORDER BY (SELECT NULL)), GETDATE())) AS m

FROM master.dbo.spt\_values

) AS t

),

active\_users AS (

SELECT

m.month\_start,

s.plan\_type,

s.user\_id

FROM

months m

JOIN

subscriptions s

ON m.month\_start BETWEEN

DATEFROMPARTS(YEAR(s.subscription\_date), MONTH(s.subscription\_date), 1)

AND

ISNULL(DATEFROMPARTS(YEAR(s.subscription\_end\_date), MONTH(s.subscription\_end\_date), 1), GETDATE())

)

SELECT

FORMAT(month\_start, 'yyyy-MM') AS month,

plan\_type,

COUNT(DISTINCT user\_id) AS active\_users

FROM

active\_users

GROUP BY

FORMAT(month\_start, 'yyyy-MM'), plan\_type

ORDER BY

month;

**Insight:**  
Tracks how many users are actively subscribed to each plan type on a monthly basis.

**Business Value:**

* **Pricing strategy:** See which plans gain or lose traction; adjust pricing, features, or promotions accordingly.
* **Forecasting:** Plan infrastructure and support according to active user volume per plan.
* **Churn analysis:** Identify if changes in plan benefits or pricing lead to subscriber shifts.

1. **Identify Inactive Users (No Activity in Last 90 Days)**
   * Use MAX(watch\_date) + DATEDIFF.

SELECT

u.user\_id,

u.name,

MAX(vh.watch\_date) AS last\_watch\_date,

DATEDIFF(DAY, MAX(vh.watch\_date), GETDATE()) AS days\_since\_last\_watch

FROM

users u

LEFT JOIN

viewing\_history vh ON u.user\_id = vh.user\_id

GROUP BY

u.user\_id, u.name

HAVING

MAX(vh.watch\_date) IS NULL

OR DATEDIFF(DAY, MAX(vh.watch\_date), GETDATE()) > 90

ORDER BY

days\_since\_last\_watch DESC;

**Insight:**  
Flags users who haven’t watched anything for at least 90 days.

**Business Value:**

* **Retention campaigns:** Target inactive users with win-back offers, surveys, or reminders.
* **Churn detection:** Predict and address churn drivers.
* **User lifecycle:** Inform customer journey mapping and improve onboarding or engagement flows.

1. **% of Users Watching Across Multiple Devices**
   * How many users watched on >1 device type?

WITH device\_counts AS (

SELECT

user\_id,

COUNT(DISTINCT device\_type) AS distinct\_device\_count

FROM

viewing\_history

GROUP BY

user\_id

),

multi\_device\_users AS (

SELECT

COUNT(\*) AS users\_with\_multiple\_devices

FROM

device\_counts

WHERE

distinct\_device\_count > 1

),

total\_users AS (

SELECT COUNT(DISTINCT user\_id) AS total\_unique\_users

FROM viewing\_history

)

SELECT

m.users\_with\_multiple\_devices,

t.total\_unique\_users,

CAST(m.users\_with\_multiple\_devices \* 100.0 / t.total\_unique\_users AS DECIMAL(5,2)) AS percentage\_multi\_device\_users

FROM

multi\_device\_users m, total\_users t;

**Insight:**  
Calculates the percentage of users who watch on more than one device type.

**Business Value:**

* **Cross-device strategy:** Indicates cross-device engagement and which users are highly engaged.
* **Product development:** Prioritize seamless cross-device experience.
* **Personalization:** Offer device-specific recommendations or encourage single-device users to try other devices.

## Additional Questions Focused on Percentages

1. **% of Users Who Watched Content on Weekends vs. Weekdays**
   * Use DATEPART(weekday, watch\_date).

;WITH view\_type AS (

SELECT

user\_id,

watch\_date,

CASE

WHEN DATEPART(WEEKDAY, watch\_date) IN (1,7) THEN 'Weekend'

ELSE 'Weekday'

END AS day\_type

FROM ott\_watch\_history

)

, user\_daytype AS (

SELECT DISTINCT user\_id, day\_type

FROM view\_type

)

SELECT

day\_type,

COUNT(DISTINCT user\_id) AS user\_count,

COUNT(DISTINCT user\_id) \* 100.0 /

(SELECT COUNT(DISTINCT user\_id) FROM ott\_watch\_history) AS percentage\_users

FROM user\_daytype

GROUP BY day\_type;

**Insight:**  
You learn what proportion of your users prefer streaming on weekends versus weekdays.

**Business Value:**

* **Content scheduling:** Launch new episodes, promotions, or live events timed for high weekend viewership.
* **Marketing:** Adjust campaigns to target peak usage days, improving engagement.
* **User behavior:** Identify potential differences in casual (weekend) vs. habitual (weekday) watchers.

1. **% of Users Who Watched from Multiple Platforms**
   * Based on content across Netflix, Prime, etc.

WITH user\_platforms AS (

SELECT

user\_id,

COUNT(DISTINCT platform) AS platforms\_watched

FROM ott\_watch\_history

GROUP BY user\_id

)

, user\_classification AS (

SELECT

user\_id,

CASE

WHEN platforms\_watched > 1 THEN 'Multi-Platform'

ELSE 'Single-Platform'

END AS platform\_type

FROM user\_platforms

)

SELECT

platform\_type,

COUNT(\*) AS user\_count,

ROUND(

COUNT(\*) \* 100.0 /

(SELECT COUNT(\*) FROM user\_classification),

2

) AS percentage\_users

FROM user\_classification

GROUP BY platform\_type;

**Insight:**  
Shows how many users consume content across more than one platform (e.g., Netflix, Prime), indicating cross-platform engagement.

**Business Value:**

* **Cross-platform strategy:** Foster multi-platform usage by promoting app installs or account linking.
* **Personalization & retention:** Recognize super-engaged users for tailored offers.
* **Product development:** Prioritize seamless cross-platform features and syncing.

1. **% of Users Who Renewed Their Subscription**
   * Compare those with overlapping or new start\_date after end\_date.

;WITH subscription\_history AS (

SELECT

user\_id,

start\_date,

end\_date,

LEAD(start\_date) OVER (PARTITION BY user\_id ORDER BY start\_date) AS next\_start\_date

FROM ott\_subscriptions

)

**Insight:**  
Measures user loyalty through subscription renewals by comparing successive subscription periods.

**Business Value:**

* **Retention focus:** Identify churn risk and evaluate success of retention strategies.
* **Pricing & features:** Adjust plans or perks to encourage renewals.
* **Customer lifecycle:** Inform timing and content of engagement campaigns.

1. **% of Content That Is Movies vs. TV Shows**
   * Use type column in content.

**Insight:**  
Breakdown of your content library by type, exposing content format mix.

**Business Value:**

* **Content acquisition:** Guide balance of future investments to meet platform mix goals.
* **User preferences:** Align content creation/promotion with user preferences for movies vs. episodic TV.
* **Catalog management:** Ensure strategic variety that appeals to different viewer segments.

1. **% of Revenue From Premium Plans**
   * Breakdown revenue by plan\_type.

;WITH plan\_revenue AS (

SELECT

plan\_type,

SUM(amount\_paid) AS revenue

FROM ott\_subscriptions

GROUP BY plan\_type

),

total\_revenue AS (

SELECT SUM(revenue) AS total FROM plan\_revenue

)

SELECT

pr.plan\_type,

pr.revenue,

ROUND((pr.revenue \* 100.0) / tr.total, 2) AS revenue\_percentage

FROM plan\_revenue pr

JOIN total\_revenue tr ON 1 = 1;

**Insight:**  
Shows the contribution of premium plans to total revenue, indicating their financial importance.

**Business Value:**

* **Revenue optimization:** Justify marketing spend and new feature development on premium tiers.
* **Pricing strategy:** Evaluate if premium pricing aligns with revenue goals.
* **Upselling:** Identify potential growth areas in premium subscriptions.

1. **% of Users Who Watched at Least 1 Hour Daily on Average**

Aggregate watch time, divide by number of days active.

WITH user\_activity AS (

SELECT

user\_id,

SUM(watch\_duration) AS total\_watch\_minutes,

COUNT(DISTINCT CAST(watch\_date AS DATE)) AS active\_days

FROM ott\_watch\_history

GROUP BY user\_id

)

, qualified\_users AS (

SELECT

user\_id,

total\_watch\_minutes,

active\_days,

total\_watch\_minutes \* 1.0 / NULLIF(active\_days, 0) AS avg\_watch\_per\_day

FROM user\_activity

)

SELECT

ROUND(

COUNT(CASE WHEN avg\_watch\_per\_day >= 60 THEN 1 END) \* 100.0 /

COUNT(\*), 2

) AS percentage\_users\_1hr\_or\_more

FROM qualified\_users;

**Insight:**  
The share of users with high engagement, watching 60+ minutes daily on average.

**Business Value:**

* **Content engagement:** Validate content attractiveness and binge potential.
* **User segmentation:** Differentiate casual vs. heavy users for targeted campaigns.
* **Monetization:** Focus advertising or subscription upsells on highly engaged users.

1. **% Change in Average Watch Duration MoM**
   * Use LAG() or ROUND((current - prev)/prev\*100, 2).

WITH monthly\_avg AS (

SELECT

FORMAT (watch\_date, 'yyyy-MM') AS month,

AVG(watch\_duration) AS avg\_watch\_duration

FROM ott\_watch\_history

GROUP BY FORMAT (watch\_date, 'yyyy-MM')

)

, lagged\_avg AS (

SELECT

month,

avg\_watch\_duration,

LAG(avg\_watch\_duration) OVER (ORDER BY month) AS prev\_month\_avg

FROM monthly\_avg

)

SELECT

month,

avg\_watch\_duration,

prev\_month\_avg,

ROUND(

(avg\_watch\_duration - prev\_month\_avg) \* 100.0 / NULLIF(prev\_month\_avg, 0),

2

) AS pct\_change

FROM lagged\_avg;

**Insight:**  
Tracks trends in average engagement over time, showing growth or decline.

**Business Value:**

* **Trend spotting:** Detect success/failure of content releases or UI changes.
* **Forecasting:** Inform capacity planning, content pipeline, and marketing.
* **Problem detection:** Quickly identify slipping engagement and act.

1. **% of Users by Country**
   * Distribution of total users by region.

SELECT

country,

COUNT(\*) AS total\_users,

ROUND(

COUNT(\*) \* 100.0 / (SELECT COUNT(\*) FROM ott\_users),

2

) AS percentage\_of\_users

FROM ott\_users

GROUP BY country

ORDER BY percentage\_of\_users DESC;

**Insight:**  
User base geographic distribution revealing where your users are concentrated.

**Business Value:**

* **Localization:** Tailor content, language, pricing, and promotions by region.
* **Expansion:** Identify markets for further growth or partnerships.
* **Compliance:** Prepare for geo-specific regulations or taxation.

1. **% of Titles Watched at Least Once**
   * Compare distinct content\_ids watched vs. total available.

SELECT COUNT(DISTINCT content\_id) AS total\_titles

FROM ott\_content;

SELECT COUNT(DISTINCT content\_id) AS watched\_titles

FROM ott\_watch\_history;

SELECT

COUNT(DISTINCT w.content\_id) \* 100.0 / COUNT(DISTINCT c.content\_id) AS pct\_titles\_watched

FROM ott\_content c

LEFT JOIN ott\_watch\_history w

ON c.content\_id = w.content\_id;

**Insight:**  
Percent of your content catalog that has been streamed at least once by any user.

**Business Value:**

* **Content utilization:** Identify potentially under-watched content for marketing or removal.
* **Catalog health:** Understand user discoverability and content diversity effectiveness.
* **Content strategy:** Optimize acquisition/production to maximize watched content.

1. **% of Users Active During Peak Hours (7PM–10PM)**
   * Filter by DATEPART(hour, watch\_date).

Count Distinct Users Active During Peak Hours

WITH peak\_users AS (

SELECT DISTINCT user\_id

FROM ott\_watch\_history

WHERE DATEPART(HOUR, watch\_date) BETWEEN 19 AND 22

)

, total\_users AS (

SELECT COUNT(DISTINCT user\_id) AS total FROM ott\_watch\_history

)

-- Final Percentage

SELECT

ROUND(

(SELECT COUNT(\*) FROM peak\_users) \* 100.0 / total,

2

) AS pct\_peak\_hour\_users

FROM total\_users;

**Insight:**  
Proportion of users who watch during prime time windows.

**Business Value:**

* **Peak load management:** Optimize infrastructure and content delivery during high-demand periods.
* **Programming:** Schedule premieres and key content during peak engagement times.
* **Advertising:** Target ads or promotions for maximum visibility and impact.