1. Total Users & Growth Trends

What is the total number of users for LioCinema and Jotstar, and how do they compare in terms of growth trends (January–November 2024)?

```sql
SELECT
    'Jotstar' AS platform, COUNT(DISTINCT user_id) AS total_users
FROM jotstar_db.subscribers
UNION ALL
SELECT
    'LioCinema' AS platform, COUNT(DISTINCT user_id) AS total_users
FROM liocinema_db.subscribers;
```

platform	total_users
Jotstar	44620
LioCinema	183446

comparison in terms of growth trends (January–November 2024)?

```sql
SELECT
    platform,
    COUNT(DISTINCT CASE WHEN MONTH(subscription_date) = 1 THEN user_id END) AS Jan,
    COUNT(DISTINCT CASE WHEN MONTH(subscription_date) = 2 THEN user_id END) AS Feb,
    COUNT(DISTINCT CASE WHEN MONTH(subscription_date) = 3 THEN user_id END) AS Mar,
    COUNT(DISTINCT CASE WHEN MONTH(subscription_date) = 4 THEN user_id END) AS Apr,
    COUNT(DISTINCT CASE WHEN MONTH(subscription_date) = 5 THEN user_id END) AS May,
    COUNT(DISTINCT CASE WHEN MONTH(subscription_date) = 6 THEN user_id END) AS Jun,
    COUNT(DISTINCT CASE WHEN MONTH(subscription_date) = 7 THEN user_id END) AS Jul,
    COUNT(DISTINCT CASE WHEN MONTH(subscription_date) = 8 THEN user_id END) AS Aug,
    COUNT(DISTINCT CASE WHEN MONTH(subscription_date) = 9 THEN user_id END) AS Sep,
    COUNT(DISTINCT CASE WHEN MONTH(subscription_date) = 10 THEN user_id END) AS Oct,
    COUNT(DISTINCT CASE WHEN MONTH(subscription_date) = 11 THEN user_id END) AS Nov
FROM (
    SELECT 'Jotstar' AS platform, user_id, subscription_date
    FROM jotstar_db.subscribers
    WHERE YEAR(subscription_date) = 2024
    UNION ALL
    SELECT 'LioCinema' AS platform, user_id, subscription_date
    FROM liocinema_db.subscribers
    WHERE YEAR(subscription_date) = 2024
) AS combined_users
WHERE MONTH(subscription_date) BETWEEN 1 AND 11
GROUP BY platform;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

platform	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov
Jotstar	3934	3939	3954	3984	3998	4020	4067	4103	4163	4196	4262
LioCinema	6758	7404	8397	9759	11977	13768	16161	19247	23873	29105	36997

2. Content Library Comparison

What is the total number of contents available on LioCinema vs. Jotstar? How do they differ in terms of language and content type?

Total Number of Contents on Each Platform

```
SELECT
    'Jotstar' AS platform, COUNT(*) AS total_contents
FROM jotstar_db.contents
UNION ALL
SELECT
    'LioCinema' AS platform, COUNT(*) AS total_contents
FROM liocinema_db.contents;
```

Result Grid | Filter Rows:

platform	total_contents
Jotstar	2360
LioCinema	1250

Content Distribution by Language

```
SELECT
    platform,
    language,
    COUNT(*) AS content_count
FROM (
    SELECT 'Jotstar' AS platform, language FROM jotstar_db.contents
    UNION ALL
    SELECT 'LioCinema' AS platform, language FROM liocinema_db.contents
) AS combined_contents
GROUP BY platform, language
ORDER BY platform, content_count DESC;
```

Result Grid | Filter Rows:

platform	language	content_count
Jotstar	English	800
Jotstar	Hindi	637
Jotstar	Tamil	251
Jotstar	Telugu	244
Jotstar	Kannada	121
Jotstar	Malayalam	118
Jotstar	Marathi	74
Jotstar	Bengali	60
Jotstar	Gujarati	28
Jotstar	Punjabi	27
LioCinema	Hindi	424
LioCinema	Telugu	242
LioCinema	Tamil	221
LioCinema	Malayalam	121
LioCinema	Kannada	118
LioCinema	Marathi	68
LioCinema	English	56

Content Distribution by Content Type

```
SELECT
    platform,
    content_type,
    COUNT(*) AS content_count
FROM (
    SELECT 'Jotstar' AS platform, content_type FROM jotstar_db.contents
    UNION ALL
    SELECT 'LioCinema' AS platform, content_type FROM liocinema_db.contents
) AS combined_contents
GROUP BY platform, content_type
ORDER BY platform, content_count DESC;
```

Result Grid | Filter Rows:

platform	content_type	content_count
Jotstar	Movie	1180
Jotstar	Series	826
Jotstar	Sports	354
LioCinema	Movie	900
LioCinema	Series	300
LioCinema	Sports	50

3. User Demographics

What is the distribution of users by age group, city tier, and subscription plan for each platform?

Distribution of Users by Age Group

```sql
SELECT
    platform,
    age_group,
    COUNT(*) AS user_count
FROM (
    SELECT 'Jotstar' AS platform, age_group FROM jotstar_db.subscribers
    UNION ALL
    SELECT 'LioCinema' AS platform, age_group FROM liocinema_db.subscribers
) AS combined_users
GROUP BY platform, age_group
ORDER BY platform, user_count DESC;
```

Result Grid | Filter Rows:

platform	age_group	user_count
Jotstar	25-34	20069
Jotstar	35-44	11274
Jotstar	18-24	7676
Jotstar	45+	5601
LioCinema	18-24	79813
LioCinema	25-34	52027
LioCinema	35-44	32560
LioCinema	45+	19046

Distribution of Users by City Tier

```sql
SELECT
    platform,
    city_tier,
    COUNT(*) AS user_count
FROM (
    SELECT 'Jotstar' AS platform, city_tier FROM jotstar_db.subscribers
    UNION ALL
    SELECT 'LioCinema' AS platform, city_tier FROM liocinema_db.subscribers
) AS combined_users
GROUP BY platform, city_tier
ORDER BY platform, user_count DESC;
```

Result Grid | Filter Rows:

platform	city_tier	user_count
Jotstar	Tier 1	25451
Jotstar	Tier 2	13424
Jotstar	Tier 3	5745
LioCinema	Tier 3	78587
LioCinema	Tier 2	63848
LioCinema	Tier 1	41011

Distribution of Users by Subscription Plan

```sql
SELECT
    platform,
    subscription_plan,
    COUNT(*) AS user_count
FROM (
    SELECT 'Jotstar' AS platform, subscription_plan FROM jotstar_db.subscribers
    UNION ALL
    SELECT 'LioCinema' AS platform, subscription_plan FROM liocinema_db.subscribers
) AS combined_users
GROUP BY platform, subscription_plan
ORDER BY platform, user_count DESC;
```

Result Grid | Filter Rows:

platform	subscription_plan	user_count
Jotstar	VIP	19157
Jotstar	Premium	13367
Jotstar	Free	12096
LioCinema	Free	104992
LioCinema	Basic	53362
LioCinema	Premium	25092

4. Active vs. Inactive Users

What percentage of LioCinema and Jotstar users are active vs. inactive? How do these rates vary by age group and subscription plan?

```sql
SELECT
    platform,
    ROUND(SUM(CASE
            WHEN last_active_date IS NOT NULL
            AND last_active_date >= subscription_date
            THEN 1 ELSE 0
        END) * 100.0 / COUNT(*), 2) AS active_percentage,
    ROUND(SUM(CASE
            WHEN last_active_date IS NULL
            THEN 1 ELSE 0
        END) * 100.0 / COUNT(*), 2) AS inactive_percentage
FROM (
    SELECT 'Jotstar' AS platform, user_id, subscription_date, last_active_date FROM jotstar_db.subscribers
    UNION ALL
    SELECT 'LioCinema' AS platform, user_id, subscription_date, last_active_date FROM liocinema_db.subscribers
) AS combined_users
GROUP BY platform;
```

Result Grid | Filter Rows: | Export:

platform	active_percentage	inactive_percentage
Jotstar	14.91	85.09
LioCinema	44.87	55.13

Active vs. Inactive Users by Age Group

```
SELECT
    platform,
    age_group,
    ROUND(SUM(CASE
            WHEN last_active_date IS NOT NULL
            AND last_active_date >= subscription_date
            THEN 1 ELSE 0
        END) * 100.0 / COUNT(*), 2) AS active_percentage,
    ROUND(SUM(CASE
            WHEN last_active_date IS NULL
            THEN 1 ELSE 0
        END) * 100.0 / COUNT(*), 2) AS inactive_percentage
FROM (
    SELECT 'Jotstar' AS platform, user_id, age_group, subscription_date, last_active_date FROM jotstar_db.subscribers
    UNION ALL
    SELECT 'LioCinema' AS platform, user_id, age_group, subscription_date, last_active_date FROM liocinema_db.subscrib
) AS combined_users
GROUP BY platform, age_group
ORDER BY platform, active_percentage DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cel

platform	age_group	active_percentage	inactive_percentage
Jotstar	18-24	15.59	84.41
Jotstar	25-34	15.57	84.43
Jotstar	35-44	14.07	85.93
Jotstar	45+	13.28	86.72
LioCinema	18-24	49.08	50.92
LioCinema	25-34	42.55	57.45
LioCinema	35-44	41.63	58.37
LioCinema	45+	39.06	60.94

Active vs. Inactive Users by Subscription Plan

```
SELECT
    platform, subscription_plan,
ROUND(SUM(CASE
            WHEN last_active_date IS NOT NULL
            AND last_active_date >= subscription_date
            THEN 1 ELSE 0
        END) * 100.0 / COUNT(*), 2) AS active_percentage,
    ROUND(SUM(CASE
            WHEN last_active_date IS NULL
            THEN 1 ELSE 0
        END) * 100.0 / COUNT(*), 2) AS inactive_percentage
FROM (
    SELECT 'Jotstar' AS platform, user_id, subscription_plan, subscription_date, last_active_date FROM jotstar_db.subs
    UNION ALL
    SELECT 'LioCinema' AS platform, user_id, subscription_plan, subscription_date, last_active_date FROM liocinema_db.
) AS combined_users
GROUP BY platform, subscription_plan
ORDER BY platform, active_percentage DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Conten

platform	subscription_plan	active_percentage	inactive_percentage
Jotstar	Free	25.74	74.26
Jotstar	VIP	13.77	86.23
Jotstar	Premium	6.74	93.26
LioCinema	Free	56.95	43.05
LioCinema	Basic	33.79	66.21
LioCinema	Premium	17.87	82.13

5. Watch Time Analysis

What is the average watch time for LioCinema vs. Jotstar during the analysis period? How do these compare by city tier and device type?

Average Watch Time for LioCinema vs. Jotstar

```
SELECT
    platform,
      ROUND(COALESCE(AVG(total_watch_time_mins), 0), 2) AS avg_watch_time_mins
FROM (
    SELECT 'Jotstar' AS platform, user_id, total_watch_time_mins FROM jotstar_db.content_consumption
    UNION ALL
    SELECT 'LioCinema' AS platform, user_id, total_watch_time_mins FROM liocinema_db.content_consumption
) AS combined_watch_time
GROUP BY platform;
```

Result Grid | Filter Rows:

platform	avg_watch_time_mins
Jotstar	7034.51
LioCinema	1536.83

Average Watch Time by City Tier

```
SELECT
    platform,
    city_tier,
  ROUND(COALESCE(AVG(total_watch_time_mins), 0), 2) AS avg_watch_time_mins
FROM (
    SELECT 'Jotstar' AS platform, s.city_tier, c.total_watch_time_mins
    FROM jotstar_db.content_consumption c
    JOIN jotstar_db.subscribers s ON c.user_id = s.user_id
    UNION ALL
    SELECT 'LioCinema' AS platform, s.city_tier, c.total_watch_time_mins
    FROM liocinema_db.content_consumption c
    JOIN liocinema_db.subscribers s ON c.user_id = s.user_id
) AS combined_watch_time
GROUP BY platform, city_tier
ORDER BY platform, avg_watch_time_mins DESC;
```

Result Grid | Filter Rows:

platform	city_tier	avg_watch_time_mins
Jotstar	Tier 1	7883.65
Jotstar	Tier 2	6300.34
Jotstar	Tier 3	4988.20
LioCinema	Tier 1	2106.14
LioCinema	Tier 2	1624.78
LioCinema	Tier 3	1093.48

Average Watch Time by Device Type

```
SELECT
    platform,
    device_type,
    ROUND(COALESCE(AVG(total_watch_time_mins)/60, 0), 2) AS avg_watch_time_hrs
FROM (
    SELECT 'Jotstar' AS platform, device_type, total_watch_time_mins FROM jotstar_db.content_consumption
    UNION ALL
    SELECT 'LioCinema' AS platform, device_type, total_watch_time_mins FROM liocinema_db.content_consumption
) AS combined_watch_time
GROUP BY platform, device_type
ORDER BY platform, avg_watch_time_hrs DESC;
```

Result Grid | Filter Rows:

platform	device_type	avg_watch_time_hrs
Jotstar	Mobile	176.06
Jotstar	TV	94.71
Jotstar	Laptop	80.95
LioCinema	Mobile	46.05
LioCinema	TV	12.65
LioCinema	Laptop	8.25

6. Inactivity Correlation

How do inactivity patterns correlate with total watch time or average watch time? Are less engaged users more likely to become inactive?

```sql
WITH user_activity AS (
    -- Combine data from both platforms
    SELECT
        'Jotstar' AS platform,
        s.user_id,
        s.last_active_date,
        c.total_watch_time_mins,
        DATEDIFF(CURDATE(), s.last_active_date) AS days_since_active
    FROM jotstar_db.subscribers s
    LEFT JOIN jotstar_db.content_consumption c ON s.user_id = c.user_id

    UNION ALL

    SELECT
        'LioCinema' AS platform,
        s.user_id,
        s.last_active_date,
        c.total_watch_time_mins,
        DATEDIFF(CURDATE(), s.last_active_date) AS days_since_active
    FROM liocinema_db.subscribers s
    LEFT JOIN liocinema_db.content_consumption c ON s.user_id = c.user_id
)

-- Classify users as Active or Inactive and analyze watch time patterns
SELECT
    platform,
    CASE
        WHEN days_since_active > 90 THEN 'Inactive'
        ELSE 'Active'
    END AS user_status,
    COUNT(user_id) AS user_count,
    ROUND(AVG(COALESCE(total_watch_time_mins, 0) / 60), 2) AS avg_watch_time_hrs,
    ROUND(SUM(COALESCE(total_watch_time_mins, 0) / (60*24)), 2) AS total_watch_time_days
FROM user_activity
GROUP BY platform, user_status
ORDER BY platform, user_status;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

platform	user_status	user_count	avg_watch_time_hrs	total_watch_time_days
Jotstar	Active	118218	128.29	631936.30
Jotstar	Inactive	15642	33.73	21980.34
LioCinema	Active	324160	29.88	403520.31
LioCinema	Inactive	106592	12.65	56197.80

From result it is obvious that less engaged users more likely to become inactive.

7. Downgrade Trends

How do downgrade trends differ between LioCinema and Jotstar? Are downgrades more prevalent on one platform compared to the other?

```sql
WITH downgrade_analysis AS (
    SELECT
        'Jotstar' AS platform,
        user_id,
        subscription_plan AS previous_plan,
        new_subscription_plan AS current_plan,
        CASE
            WHEN new_subscription_plan = 'VIP' and subscription_plan = 'FREE'  THEN 'Upgraded'
             WHEN new_subscription_plan = 'PREMIUM' and subscription_plan = 'FREE'  THEN 'Upgraded'
              WHEN new_subscription_plan = 'PREMIUM' and subscription_plan = 'VIP'  THEN 'Upgraded'
               WHEN new_subscription_plan = 'FREE' and subscription_plan = 'VIP'  THEN 'Downgraded'
             WHEN new_subscription_plan = 'FREE' and subscription_plan = 'PREMIUM'  THEN 'Downgraded'
              WHEN new_subscription_plan = 'VIP' and subscription_plan = 'PREMIUM'  THEN 'Downgraded'
            ELSE 'No Change'
        END AS change_status
    FROM jotstar_db.subscribers
    WHERE new_subscription_plan IS NOT NULL

    UNION ALL

    SELECT
        'LioCinema' AS platform,
        user_id,
        subscription_plan AS previous_plan,
        new_subscription_plan AS current_plan,
        CASE
             WHEN new_subscription_plan = 'Basic' and subscription_plan = 'FREE'  THEN 'Upgraded'
             WHEN new_subscription_plan = 'PREMIUM' and subscription_plan = 'Basic'  THEN 'Upgraded'
              WHEN new_subscription_plan = 'Premium' and subscription_plan = 'Free'  THEN 'Upgraded'
               WHEN new_subscription_plan = 'FREE' and subscription_plan = 'Basic'  THEN 'Downgraded'
             WHEN new_subscription_plan = 'FREE' and subscription_plan = 'PREMIUM'  THEN 'Downgraded'
              WHEN new_subscription_plan = 'Basic' and subscription_plan = 'Premium'  THEN 'Downgraded'
            ELSE 'No Change'
        END AS change_status
    FROM liocinema_db.subscribers
    WHERE new_subscription_plan IS NOT NULL
)

-- Aggregate results by platform
SELECT
    platform,
    change_status,
    COUNT(user_id) AS user_count,
    ROUND(100.0 * COUNT(user_id) / SUM(COUNT(user_id)) OVER (PARTITION BY platform), 2) AS percentage
FROM downgrade_analysis
GROUP BY platform, change_status
ORDER BY platform, percentage DESC;
```

Result Grid | Filter Rows: | Export:

platform	change_status	user_count	percentage
Jotstar	Upgraded	4348	61.33
Jotstar	Downgraded	2742	38.67
LioCinema	Downgraded	20859	83.39
LioCinema	Upgraded	4155	16.61

LioCinema has more downgrade percentage than Jotstar.

8. Upgrade Patterns

What are the most common upgrade transitions (e.g., Free to Basic, Free to VIP, Free to Premium) for LioCinema and Jotstar? How do these differ across platforms?

```sql
WITH upgrade_analysis AS (
    SELECT
        'Jotstar' AS platform,
        user_id,
        subscription_plan AS previous_plan,
        new_subscription_plan AS current_plan,
        CASE
            WHEN new_subscription_plan = 'VIP' and subscription_plan = 'FREE'  THEN 'Upgraded'
             WHEN new_subscription_plan = 'PREMIUM' and subscription_plan = 'FREE'  THEN 'Upgraded'
              WHEN new_subscription_plan = 'PREMIUM' and subscription_plan = 'VIP'  THEN 'Upgraded'
               WHEN new_subscription_plan = 'FREE' and subscription_plan = 'VIP'  THEN 'Downgraded'
             WHEN new_subscription_plan = 'FREE' and subscription_plan = 'PREMIUM'  THEN 'Downgraded'
              WHEN new_subscription_plan = 'VIP' and subscription_plan = 'PREMIUM'  THEN 'Downgraded'
            ELSE 'No Change'
        END AS change_status
    FROM jotstar_db.subscribers
    WHERE new_subscription_plan IS NOT NULL

    UNION ALL

    SELECT
        'LioCinema' AS platform,
        user_id,
        subscription_plan AS previous_plan,
        new_subscription_plan AS current_plan,
        CASE
            WHEN new_subscription_plan = 'Basic' and subscription_plan = 'FREE'  THEN 'Upgraded'
             WHEN new_subscription_plan = 'PREMIUM' and subscription_plan = 'Basic'  THEN 'Upgraded'
              WHEN new_subscription_plan = 'Premium' and subscription_plan = 'Free'  THEN 'Upgraded'
               WHEN new_subscription_plan = 'FREE' and subscription_plan = 'Basic'  THEN 'Downgraded'
             WHEN new_subscription_plan = 'FREE' and subscription_plan = 'PREMIUM'  THEN 'Downgraded'
              WHEN new_subscription_plan = 'Basic' and subscription_plan = 'Premium'  THEN 'Downgraded'
            ELSE 'No Change'
        END AS change_status
    FROM liocinema_db.subscribers
    WHERE new_subscription_plan IS NOT NULL
)

-- Count upgrade transitions for each platform
SELECT
    platform,
    previous_plan,
    current_plan,
    COUNT(user_id) AS transition_count,
    ROUND(100.0 * COUNT(user_id) / SUM(COUNT(user_id)) OVER (PARTITION BY platform), 2) AS percentage

FROM upgrade_analysis
WHERE change_status = 'Upgraded'
GROUP BY platform, previous_plan, current_plan
ORDER BY platform, transition_count DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

platform	previous_plan	current_plan	transition_count	percentage
Jotstar	VIP	Premium	2821	64.88
Jotstar	Free	VIP	844	19.41
Jotstar	Free	Premium	683	15.71
LioCinema	Free	Basic	2078	50.01
LioCinema	Basic	Premium	1362	32.78
LioCinema	Free	Premium	715	17.21

9. Paid Users Distribution

How does the paid user percentage (e.g., Basic, Premium for LioCinema; VIP, Premium for Jotstar) vary across different platforms? Analyse the proportion of premium users in Tier 1, Tier 2, and Tier 3 cities and identify any notable trends or differences.

```sql
WITH user_payment_status AS (
    SELECT
        'Jotstar' AS platform,
        user_id,
        city_tier,
        subscription_plan,
        CASE
            WHEN subscription_plan IN ('VIP', 'Premium') THEN 'Paid'
            ELSE 'Free'
        END AS user_type
    FROM jotstar_db.subscribers

    UNION ALL

    SELECT
        'LioCinema' AS platform,
        user_id,
        city_tier,
        subscription_plan,
        CASE
            WHEN subscription_plan IN ('Basic', 'Premium') THEN 'Paid'
            ELSE 'Free'
        END AS user_type
```

```
    FROM liocinema_db.subscribers
)

SELECT
    platform,
    user_type,
    COUNT(user_id) AS user_count,
    ROUND(100.0 * COUNT(user_id) / SUM(COUNT(user_id)) OVER (PARTITION BY platform), 2) AS percentage
FROM user_payment_status
GROUP BY platform, user_type
ORDER BY platform, percentage DESC;
```

Result Grid | Filter Rows: | Export:

platform	user_type	user_count	percentage
Jotstar	Paid	32524	72.89
Jotstar	Free	12096	27.11
LioCinema	Free	104992	57.23
LioCinema	Paid	78454	42.77

Premium User Distribution by City Tier

```
WITH premium_users AS (
    SELECT
        'Jotstar' AS platform,
        city_tier,
        user_id
    FROM jotstar_db.subscribers
    WHERE subscription_plan = 'Premium'

    UNION ALL

    SELECT
        'LioCinema' AS platform,
        city_tier,
        user_id
    FROM liocinema_db.subscribers
    WHERE subscription_plan = 'Premium'
)

SELECT
    platform,
    city_tier,
    COUNT(user_id) AS premium_user_count,
    ROUND(100.0 * COUNT(user_id) / SUM(COUNT(user_id)) OVER (PARTITION BY platform), 2) AS percentage
FROM premium_users
GROUP BY platform, city_tier
ORDER BY platform, percentage DESC;
```

Result Grid | Filter Rows: | Export: | Wra

platform	city_tier	premium_user_count	percentage
Jotstar	Tier 1	10178	76.14
Jotstar	Tier 2	2566	19.20
Jotstar	Tier 3	623	4.66
LioCinema	Tier 1	10306	41.07
LioCinema	Tier 2	9090	36.23
LioCinema	Tier 3	5696	22.70

Tier1 city has more premium percentage in both Jotstar and LioCinema.

10. Revenue Analysis ● Assume the following monthly subscription prices, calculate the total revenue generated by both platforms (LioCinema and Jotstar) for the analysis period (January to November 2024).

Platform	Plan	Price
LioCinema	Basic	₹ 69
	Premium	₹ 129
Jotstar	VIP	₹ 159
	Premium	₹ 359

The calculation should consider:

❖ Subscribers count under each plan.

❖ Active duration of subscribers on their respective plans.

❖ Upgrades and downgrades during the period, ensuring revenue reflects the time spent under each plan.

```
WITH active_subscription_duration AS (
    SELECT
        'Jotstar' AS platform,
        user_id,
        subscription_plan AS initial_plan,
        new_subscription_plan AS changed_plan,
        subscription_date,
        plan_change_date,
        TIMESTAMPDIFF(MONTH, subscription_date,
            COALESCE(plan_change_date, '2024-11-30')) AS initial_active_months,
        TIMESTAMPDIFF(MONTH, plan_change_date, '2024-11-30') AS changed_active_months
    FROM jotstar_db.subscribers
    WHERE subscription_date BETWEEN '2024-01-01' AND '2024-11-30'

    UNION ALL
```

```sql
    SELECT
        'LioCinema' AS platform,
        user_id,
        subscription_plan AS initial_plan,
        new_subscription_plan AS changed_plan,
        subscription_date,
        plan_change_date,
        TIMESTAMPDIFF(MONTH, subscription_date,
            COALESCE(plan_change_date, '2024-11-30')) AS initial_active_months,
        TIMESTAMPDIFF(MONTH, plan_change_date, '2024-11-30') AS changed_active_months
    FROM liocinema_db.subscribers
    WHERE subscription_date BETWEEN '2024-01-01' AND '2024-11-30'
)

SELECT
    a.platform,
    a.initial_plan AS old_subscription_plan,
    a.changed_plan AS new_subscription_plan,
    COUNT(DISTINCT a.user_id) AS total_users,
    SUM(
        CASE
            WHEN a.plan_change_date IS NULL
                THEN a.initial_active_months * IFNULL(p1.price, 0) -- Revenue for users who didn't change plans
            ELSE
                (a.initial_active_months * IFNULL(p1.price, 0) + a.changed_active_months * IFNULL(p2.price, 0))
        END
    ) AS total_revenue
FROM active_subscription_duration a
LEFT JOIN (
    SELECT 'LioCinema' AS platform, 'Basic' AS subscription_plan, 69 AS price
    UNION ALL
    SELECT 'LioCinema', 'Premium', 129
    UNION ALL
    SELECT 'Jotstar', 'VIP', 159
    UNION ALL
    SELECT 'Jotstar', 'Premium', 359
) p1
ON a.platform = p1.platform AND a.initial_plan = p1.subscription_plan
LEFT JOIN (
    SELECT 'LioCinema' AS platform, 'Basic' AS subscription_plan, 69 AS price
    UNION ALL
    SELECT 'LioCinema', 'Premium', 129
    UNION ALL
    SELECT 'Jotstar', 'VIP', 159
    UNION ALL
    SELECT 'Jotstar', 'Premium', 359
) p2
```

```
ON a.platform = p2.platform AND a.changed_plan = p2.subscription_plan
GROUP BY a.platform, a.initial_plan, a.changed_plan
ORDER BY a.platform, total_revenue DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

platform	old_subscription_plan	new_subscription_plan	total_users	total_revenue
Jotstar	Premium	NULL	12774	22100040
Jotstar	VIP	NULL	14187	10261701
Jotstar	VIP	Premium	2821	4958227
Jotstar	VIP	Free	2149	1284879
Jotstar	Free	Premium	683	934836
Jotstar	Premium	VIP	368	660414
Jotstar	Free	VIP	844	532491
Jotstar	Premium	Free	225	324536
Jotstar	Free	NULL	10569	0
LioCinema	Basic	NULL	41691	8837934
LioCinema	Premium	NULL	14542	4673154
LioCinema	Premium	Free	7439	1665777
LioCinema	Premium	Basic	3111	1265901
LioCinema	Basic	Free	10309	1201635
LioCinema	Basic	Premium	1362	582771
LioCinema	Free	Basic	2078	232668
LioCinema	Free	Premium	715	160089
LioCinema	Free	NULL	102199	0

Insights

1. **Jotstar generates higher revenue from Premium plans**, while **LioCinema has more Basic plan subscribers**.
2. **LioCinema's pricing model focuses on affordability**, attracting more users but generating lower revenue per user.
3. **Jotstar's Premium users contribute the most revenue**, showing strong demand for high-end content.