

1. Calculating total number of customers

```
SELECT COUNT(*) AS total_customers FROM customer_accounts;
```

2. Segmenting active and churned customers into 2 groups

```
SELECT is_active, COUNT(*) AS customer_count  
FROM subscriptions  
GROUP BY is_active;
```

3. Segmenting different product subscriptions based on their MRR contribution

```
SELECT plan, AVG(mrr) AS avg_mrr, SUM(mrr) AS total_mrr, COUNT(*) AS customer_count  
FROM subscriptions  
GROUP BY plan  
ORDER BY total_mrr DESC;
```

4. Analyzing product usage patterns of active and churned customers

```
SELECT s.is_active, AVG(u.logins) AS avg_logins, AVG(u.tickets_raised) AS avg_tickets,  
AVG(u.feature_usage_score) AS avg_feature_usage  
FROM usage_metrics u  
JOIN subscriptions s ON u.customer_id = s.customer_id  
GROUP BY s.is_active;
```

5. Segmenting customers based on their status and subscription duration

```
SELECT customer_id,  
CASE WHEN is_active = TRUE THEN DATEDIFF('day', start_date, CURRENT_DATE())  
ELSE DATEDIFF('day', start_date, end_date)  
END AS subscription_duration,  
  
-- Adding these columns for better context  
CASE WHEN is_active = TRUE THEN 'Active'  
ELSE 'Churned'  
END AS customer_status,
```

```

-- Converting days to months and years for better readability
CASE WHEN is_active = TRUE THEN FLOOR(DATEDIFF('day', start_date, CURRENT_DATE()) / 365)
|| ' years, ' || FLOOR((DATEDIFF('day', start_date, CURRENT_DATE()) % 365) / 30) || ' months'
ELSE FLOOR(DATEDIFF('day', start_date, end_date) / 365) || ' years, ' || FLOOR((DATEDIFF('day',
start_date, end_date) % 365) / 30) || ' months'
END AS subscription_duration_in_years,

start_date,
CASE WHEN is_active = TRUE THEN 'Current'
ELSE TO_CHAR(end_date, 'YYYY-MM-DD')
END AS end_date

FROM subscriptions
ORDER BY subscription_duration DESC;

```

6. Segregating customers based on their usage metrics

```

SELECT customer_id, AVG(logins) AS avg_monthly_logins, AVG(tickets_raised) AS
avg_tickets_raised, AVG(feature_usage_score) AS avg_feature_usage
FROM usage_metrics
GROUP BY customer_id;

```

7. Labeling the customers based on status

```

SELECT customer_id,
CASE WHEN is_active = FALSE THEN 1 ELSE 0 END AS churn_label
FROM subscriptions;

```

8. Adding data features through customer duration info to calculate metrics such as active time periods, retention rate, etc.

```

WITH duration_metrics AS (SELECT
CASE WHEN is_active = TRUE THEN DATEDIFF('day', start_date, CURRENT_DATE())
ELSE DATEDIFF('day', start_date, end_date)
END AS subscription_duration,
is_active

```

```
FROM subscriptions  
)
```

```
SELECT  
'Overall' AS metric, COUNT(*) AS total_customers,  
ROUND(AVG(subscription_duration)/30, 1) AS avg_duration_months,  
ROUND(AVG(CASE WHEN is_active = TRUE THEN subscription_duration END)/30, 1) AS  
avg_duration_active_months,  
ROUND(AVG(CASE WHEN is_active = FALSE THEN subscription_duration END)/30, 1) AS  
avg_duration_churned_months,  
ROUND(SUM(CASE WHEN is_active = TRUE THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 1) || '%' AS  
retention_rate  
FROM duration_metrics
```

```
UNION ALL
```

```
SELECT s.plan AS metric,  
COUNT(*) AS total_customers,  
ROUND(AVG(  
CASE  
WHEN s.is_active = TRUE  
THEN DATEDIFF('day', s.start_date, CURRENT_DATE())  
ELSE  
DATEDIFF('day', s.start_date, s.end_date)  
END)/30, 1) AS avg_duration_months,
```

```
ROUND(AVG(  
CASE  
WHEN s.is_active = TRUE  
THEN DATEDIFF('day', s.start_date, CURRENT_DATE())  
END)/30, 1) AS avg_duration_active_months,
```

```
ROUND(AVG(  
CASE  
WHEN s.is_active = FALSE  
THEN DATEDIFF('day', s.start_date, s.end_date)  
END)/30, 1) AS avg_duration_churned_months,  
ROUND(SUM(CASE WHEN s.is_active = TRUE THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 1) || '%'  
AS retention_rate  
FROM subscriptions s  
GROUP BY s.plan  
ORDER BY  
CASE WHEN metric = 'Overall' THEN 0 ELSE 1 END,  
total_customers DESC;
```

9. Calculating metrics from depth of product usage by customers

```
WITH usage_aggregates AS (  
  SELECT customer_id,  
    AVG(logins) AS avg_monthly_logins,  
    AVG(tickets_raised) AS avg_tickets_raised,  
    AVG(feature_usage_score) AS avg_feature_usage  
  FROM usage_metrics  
  GROUP BY customer_id),  
  
  subscription_features AS (SELECT customer_id, plan, mrr,  
    CASE  
      WHEN is_active = TRUE  
      THEN DATEDIFF('day', start_date, CURRENT_DATE())  
      ELSE  
        DATEDIFF('day', start_date, end_date)  
      END AS subscription_duration,  
  
    CASE  
      WHEN is_active = TRUE  
      THEN 'Active'  
      ELSE 'Churned'  
      END AS customer_status,  
  
    CASE  
      WHEN is_active = FALSE THEN 1 ELSE 0 END AS churn_label  
    FROM subscriptions)  
  
  SELECT ca.customer_id, ca.segment, ca.industry, ca.country, sf.plan, sf.mrr,  
    sf.subscription_duration, sf.customer_status, sf.churn_label, ua.avg_monthly_logins,  
    ua.avg_tickets_raised, ua.avg_feature_usage  
  FROM customer_accounts ca  
  LEFT JOIN subscription_features sf ON ca.customer_id = sf.customer_id  
  LEFT JOIN usage_aggregates ua ON ca.customer_id = ua.customer_id;
```