```sql
WITH params AS (
    SELECT
        DATE '2026-01-19' AS as_of_date
),

-- Q1) Aktif müşteriler son 30 gün usage + plan
q1 AS (
    SELECT
        c.full_name,
        p.plan_name,
        COALESCE(SUM(u.quantity), 0) AS api_calls_30d
    FROM subscriptions s
    JOIN customers c ON c.customer_id = s.customer_id
    JOIN plans p ON p.plan_id = s.plan_id
    LEFT JOIN usage_events u
        ON u.customer_id = c.customer_id
        AND u.event_Date >= (SELECT as_of_date - INTERVAL '30 days' FROM params)
    WHERE s.status= 'active'
    GROUP BY c.full_name, p.plan_name
),


--Q2) Plan bazında aktif müşteri + ortalama usage
q2 AS (
    SELECT
        p.plan_name,
        COUNT(DISTINCT s.customer_id) AS active_customers,
        ROUND(AVG(u.quantity), 2) AS avg_api_usage
    FROM subscriptions s
    JOIN plans p ON p.plan_id = s.plan_id
    LEFT JOIN usage_events u ON u.customer_id = s.customer_id
    WHERE s.status = 'active'
    GROUP BY p.plan_name
),

-- Q3) High churn risk (>0.7 ama aktif)
q3 AS (
    SELECT
        c.full_name,
        mp.predicted_probability
    FROM model_predictions mp
    JOIN customers c ON c.customer_id = mp.customer_id
    JOIN subscriptions s ON s.customer_id = c.customer_id
    WHERE mp.predicted_probability > 0.7
        AND s.status = 'active'
),

--Q4) Confusion matrix labels

q4 AS (
    SELECT
        mp.customer_id,
```

```
        mp.predicted_probability,
        ao.churned,
        CASE
            WHEN mp.predicted_probability > 0.5 AND ao.churned THEN 'TP'
            WHEN mp.predicted_probability > 0.5 AND NOT ao.churned THEN 'FP'
            WHEN mp.predicted_probability <= 0.5 AND ao.churned THEN 'FN'
            ELSE 'TN'
        END AS confusion_label
    FROM model_predictions mp
    JOIN actual_outcomes ao ON ao.customer_id = mp.customer_id
),

--Q5) Calibration - model version bazlı

q5 AS (
    SELECT
        model_version,
        ROUND(AVG(predicted_probability), 3) AS avg_pred_prob,
        ROUND(AVG(CASE WHEN ao.churned THEN 1 ELSE 0 END),3) AS actual_churn_rate
    FROM model_predictions mp
    JOIN actual_outcomes ao ON ao.customer_id = mp.customer_id
    GROUP BY model_version
),

-- Q6) Usage düşüşü (lag)

q6 AS (
    SELECT
        customer_id,
        event_date,
        quantity,
        quantity - LAG(quantity) OVER(PARTITION BY customer_id ORDER BY
event_date) AS usage_change
    FROM usage_events
),

--Q7) Churn öncesi 30 gün ortalama usage
q7 AS (
    SELECT
        ao.customer_id,
        ROUND(AVG(u.quantity), 2) AS avg_usage_before_churn
    FROM actual_outcomes ao
    LEFT JOIN usage_events u ON u.customer_id = ao.customer_id
    AND u.event_date >= ao.outcome_date - INTERVAL '30 days'
    WHERE ao.churned = true
    GROUP BY ao.customer_id
),

--Q8)Prediction -> outcome süresi
q8 AS (
    SELECT
        mp.customer_id,
        ao.outcome_date - mp.prediction_date AS days_between
    FROM model_predictions mp
```

```
    JOIN actual_outcomes ao ON ao.customer_id = mp.customer_id
    ORDER BY days_between
),

--Q9) En güncel tahmin
q9 AS (
    SELECT *
    FROM (
        SELECT
            mp.*,
            ROW_NUMBER() OVER(PARTITION BY customer_id ORDER BY prediction_date
DESC) AS rn
        FROM model_predictions mp
    ) t
    WHERE rn = 1
),

--Q10 Risk Segmentation
q10 AS (
    SELECT
        c.full_name,
        mp.predicted_probability,
        COALESCE(SUM(u.quantity), 0) AS api_calls_30d,
        CASE
            WHEN mp.predicted_probability > 0.7 AND COALESCE(SUM(u.quantity),0) =
0 THEN 'High Risk'
            WHEN mp.predicted_probability > 0.7 AND COALESCE(SUM(u.quantity), 0) >
0 THEN 'Medium Risk'
            ELSE 'Low Risk'
        END AS risk_segment
    FROM customers c
    JOIN model_predictions mp ON mp.customer_id = c.customer_id
    LEFT JOIN usage_events u ON u.customer_id = c.customer_id
        AND u.event_Date >= (SELECT as_of_date - INTERVAL '30 days' FROM params)
    GROUP BY c.full_name, mp.predicted_probability
)
SELECT * FROM q10;
```