

```
-- Q1) ORDER TOTAL HESABI (order_id, user_id, order_date, order_total)
```

```
WITH order_totals AS (
    SELECT
        oi.order_id,
        SUM(oi.quantity * oi.unit_price) AS order_total
    FROM order_items oi
    GROUP BY oi.order_id
)
```

```
SELECT
    o.order_id,
    o.user_id,
    o.order_date,
    o.channel,
    ot.order_total
FROM orders o
JOIN order_totals ot ON ot.order_id = o.order_id
ORDER BY o.order_date, o.order_id;
```

```
-- Q2) Kampanya indirimi uygulanmış net_total (order_total + net_total)
```

```
WITH order_totals AS (
    SELECT
        oi.order_id,
        SUM(oi.quantity * oi.unit_price) AS order_total
    FROM order_items oi
    GROUP BY oi.order_id
)
```

```
SELECT
    o.order_id,
    o.user_id,
    o.order_date,
    o.channel,
    o.campaign_id,
    ot.order_total,
    ROUND (
        CASE
            WHEN o.campaign_id IS NULL THEN ot.order_total
            ELSE ot.order_total * (1 - c.discount_rate)
        END
    ,2 ) AS net_total
```

```
FROM orders o
JOIN order_totals ot ON ot.order_id = o.order_id
LEFT JOIN campaigns c ON c.campaign_id = o.campaign_id
ORDER BY o.order_date, o.order_id;
```

```
-- Q3) Kullanıcı bazlı total_orders, total_spend, AOV
```

```

WITH order_totals AS (
    SELECT
        oi.order_id,
        SUM(oi.quantity * oi.unit_price) AS order_total
    FROM order_items oi
    GROUP BY oi.order_id
),
order_net AS (
    SELECT
        o.order_id,
        o.user_id,
        ROUND(
            CASE
                WHEN o.campaign_id IS NULL THEN ot.order_total
                ELSE ot.order_total * (1 - c.discount_rate)
            END
        ,2) AS net_total
    FROM orders o
    JOIN order_totals ot ON ot.order_id = o.order_id
    LEFT JOIN campaigns c ON c.campaign_id = o.campaign_id
)
SELECT
    u.user_id,
    u.full_name,
    COUNT(onet.order_id) AS total_orders,
    ROUND(COALESCE(SUM(onet.net_total), 0), 2) AS total_spend,
    ROUND(
        CASE WHEN COUNT (onet.order_id) = 0 THEN 0
        ELSE SUM (onet.net_total) / COUNT(onet.order_id)
        END
    , 2) AS avg_order_value
FROM users u
LEFT JOIN order_net onet ON onet.user_id = u.user_id
GROUP BY u.user_id, u.full_name
ORDER BY total_spend DESC, u.user_id;

```

-- Q4) En çok harcayan top 3 kullanıcı (eşitlik varsa hepsi)

```

WITH order_totals AS (
    SELECT
        oi.order_id,
        SUM(oi.quantity * oi.unit_price) AS order_total
    FROM order_items oi
    GROUP BY oi.order_id
),
order_net AS (
    SELECT
        o.user_id,
        ROUND(
            CASE
                WHEN o.campaign_id IS NULL THEN ot.order_total

```

```
        ELSE ot.order_total * (1 - c.discount_rate)
    END
    , 2) AS net_total
FROM orders o
JOIN order_totals ot ON ot.order_id = o.order_id
LEFT JOIN campaigns c ON c.campaign_id = o.campaign_id
),
user_spend AS (
    SELECT
        u.user_id,
        u.full_name,
        ROUND(COALESCE(SUM(onet.net_total) , 0)
            , 2) AS total_spend
    FROM users u
    LEFT JOIN order_net onet ON onet.user_id = u.user_id
    GROUP BY u.user_id, u.full_name
),
ranked AS (
    SELECT
        *,
        DENSE_RANK() OVER(ORDER BY total_spend DESC) AS rnk
    FROM user_spend
)
SELECT
    user_id, full_name, total_spend
FROM ranked
WHERE rnk <= 3
ORDER BY total_spend DESC, user_id;

-- Q5) İlk siparişten sonraki 14 günnde 2.sipariş yapanlar first_order_date,
second_order_date

WITH users_order AS (
    SELECT
        o.user_id,
        o.order_id,
        o.order_date,
        ROW_NUMBER () OVER (PARTITION BY o.user_id ORDER BY o.order_date,
o.order_id) AS rn
    FROM orders o
),
first_second AS (
    SELECT
        u1.user_id,
        MIN(CASE WHEN u1.rn = 1 THEN u1.order_date END) AS first_order_date,
        MIN(CASE WHEN u1.rn = 2 THEN u1.order_date END) AS second_order_date
    FROM users_order u1
    GROUP BY u1.user_id
)
SELECT
    fs.user_id,
```

```
u.full_name,
fs.first_order_date,
fs.second_order_date
FROM first_second fs
JOIN users u ON u.user_id = fs.user_id
WHERE fs.second_order_date IS NOT NULL
    AND fs.second_order_date <= fs.first_order_date + INTERVAL '14 Days'
ORDER BY fs.user_id;

-- Q6) Siparişler arası gün farkı (prev_order_date, days_since_prev)

WITH x AS (
    SELECT
        o.user_id,
        u.full_name,
        o.order_id,
        o.order_date,
        LAG(o.order_date) OVER (PARTITION BY o.user_id ORDER BY o.order_date,
o.order_id) AS prev_order_date
    FROM orders o
    JOIN users u ON u.user_id = o.user_id
)
SELECT
    user_id,
    full_name,
    order_id,
    order_date,
    prev_order_date,
    (order_date - prev_order_date) AS days_since_prev
FROM x
WHERE prev_order_date IS NOT NULL
ORDER BY user_id, order_date, order_id;

-- Q7) Günlük net gelir + 3 günlük hareketli ortalama

WITH order_totals AS (
    SELECT
        oi.order_id,
        SUM(oi.quantity * oi.unit_price) AS order_total
    FROM order_items oi
    GROUP BY oi.order_id
),
order_net AS (
    SELECT
        o.order_date,
        ROUND(
            CASE
                WHEN o.campaign_id IS NULL THEN ot.order_total
                ELSE ot.order_total * (1-c.discount_rate)
            END
            , 2 ) AS net_total
    FROM orders o
)
```

```
    JOIN order_totals ot ON ot.order_id = o.order_id
    LEFT JOIN campaigns c ON c.campaign_id = o.campaign_id
),
daily AS (
    SELECT
        order_date,
        ROUND(SUM(net_total), 2) AS daily_net_revenue
    FROM order_net
    GROUP BY order_date
)
SELECT
    order_date,
    daily_net_revenue,
    ROUND(
        AVG(daily_net_revenue) OVER (ORDER BY order_date ROWS BETWEEN 2 PRECEDING
        AND CURRENT ROW)
        , 2) AS revenue_3d_ma
FROM daily
ORDER BY order_date;

-- Q8) Kanal bazlı gelir ve toplam içindeki payı (%)

WITH order_totals AS (
    SELECT
        oi.order_id,
        SUM (oi.quantity * oi.unit_price) AS order_total
    FROM order_items oi
    GROUP BY oi.order_id
),
order_net AS (
    SELECT
        o.channel,
        ROUND(
            CASE
                WHEN o.campaign_id IS NULL THEN ot.order_total
                ELSE ot.order_total * (1 - c.discount_rate)
            END
            , 2 ) AS net_total
    FROM orders o
    JOIN order_totals ot ON ot.order_id = o.order_id
    LEFT JOIN campaigns c ON c.campaign_id = o.campaign_id
),
channel_sum AS (
    SELECT
        channel,
        ROUND(SUM(net_total), 2) AS channel_net_revenue
    FROM order_net
    GROUP BY channel
)
```

```
SELECT
    channel,
    channel_net_revenue,
    SUM(channel_net_revenue) OVER() AS total_net_revenue,
    ROUND(100.0 * channel_net_revenue) / NULLIF(SUM(channel_net_revenue) OVER (), 2) AS channel_share_pct

FROM channel_sum
ORDER BY channel_net_revenue DESC, channel;

-- Q9) Kategori bazlı en çok satan ürün

WITH product_qty AS (
    SELECT
        p.category,
        p.product_id,
        p.product_name,
        SUM(oi.quantity) AS total_qty
    FROM order_items oi
    JOIN products p ON p.product_id = oi.product_id
    GROUP BY p.category, p.product_id, p.product_name
),

ranked AS (
    SELECT
        *,
        ROW_NUMBER() OVER (PARTITION BY category ORDER BY total_qty DESC, product_id) AS rn
    FROM product_qty
)

SELECT
    category,
    product_id,
    product_name,
    total_qty
FROM ranked
WHERE rn = 1
ORDER BY category;

-- Q10) Churn Label (EN SON SİPARİŞ 21'DEN ESKİ)

WITH last_order AS (
    SELECT
        u.full_name,
        u.user_id,
        MAX(o.order_date) AS last_order_date
    FROM users u
    LEFT JOIN orders o ON o.user_id = u.user_id
    GROUP BY u.user_id, u.full_name
)
```

```
SELECT
    full_name,
    user_id,
    last_order_date,
    CASE
        WHEN last_order_date IS NULL THEN 1
        WHEN last_order_date < ('2026-01-16'::date - INTERVAL '21 Days') THEN 1
        ELSE 0
    END AS is_churned_21d
FROM last_order
ORDER BY user_id;
```