SQL skills: goods by storages

Data Output | Messages | Notifications

	good_id integer	article character	supplier_article character	good_name_en character	category_id integer	category_name character	point_id integer	point_name_en character	in_qnt real	out_qnt real	balance real
1	3	1058	12970	Ceylon OP Blend "Adam...	1	Red Tea	1	Tea House	3200	300	2900
2	2	1057	11113	Assam TGFOPI Keyhun...	1	Red Tea	1	Tea House	2000	50	1950
3	3	1058	12970	Ceylon OP Blend "Adam...	1	Red Tea	2	Akacia Storage	500	[null]	500
4	1	1051	10999	Darjeeling FTGFOP-I To...	1	Red Tea	1	Tea House	3000	100	2900

```sql
-- actual amount of goods on the storages according to supplies, sales and inventories
SELECT
    a.good_id,
    g.article,
    g.supplier_article,
    g.good_name_en,
    g.category_id,
    ct.category_name,
    a.point_id,
    p.point_name_en,
    a.in_qnt,
    a.out_qnt,
    a.in_qnt - coalesce(a.out_qnt, 0) AS balance

FROM (
    SELECT * FROM (
        -- goods without inventory
        WITH i AS (
            SELECT DISTINCT good_id FROM r_inventory),
        -- goods in
        g_in AS (
            SELECT
                good_id,
                s.point_id,
                SUM(fact_qnt) AS in_qnt
            FROM r_goods_supplies
                JOIN r_supplies s USING (supply_id)
            GROUP BY good_id, point_id),
        -- goods out
        g_out AS (
            SELECT
                good_id,
                o.point_id,
                SUM(good_qnt) AS out_qnt
            FROM r_goods_orders
                JOIN r_orders o USING (order_id)
            GROUP BY good_id, point_id)
        SELECT
```

```
        good_id,
        g_in.point_id,
        g_in.in_qnt AS in_qnt,
        g_out.out_qnt AS out_qnt
    FROM goods
        LEFT JOIN i USING (good_id)
        JOIN g_in USING(good_id)
        LEFT JOIN g_out USING(good_id, point_id)
    WHERE i.good_id IS null) AS no_i
UNION ALL
SELECT * FROM(
    WITH i AS (
    -- goods with inventory
        SELECT
            DISTINCT good_id,
            point_id,
            -- last inventory: amount of goods
            last_value(fact_qnt) OVER
                (PARTITION BY good_id, point_id
                ORDER BY created_at
                ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS in_qnt,
            -- last inventory: date
            MAX(created_at) OVER
                (PARTITION BY good_id, point_id
                ORDER BY created_at
                ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS in_date
        FROM r_inventory),
    -- goods in after inventory
    g_in AS (
        SELECT
            good_id,
            s.point_id,
            SUM(fact_qnt) AS in_qnt
        FROM r_goods_supplies
            JOIN r_supplies s USING (supply_id)
            JOIN i USING (good_id, point_id)
        WHERE s.created_at > i.in_date
        GROUP BY good_id, point_id),
    -- goods out after inventory
    g_out AS (
        SELECT
            good_id,
            o.point_id,
            SUM(good_qnt) AS out_qnt
        FROM r_goods_orders
            JOIN r_orders o USING (order_id)
            JOIN i USING (good_id, point_id)
        WHERE o.created_at > i.in_date
```

```
        GROUP BY good_id, point_id)
    SELECT
        good_id,
        point_id,
        i.in_qnt + coalesce(g_in.in_qnt, 0) AS in_qnt,
        g_out.out_qnt AS out_qnt
    FROM i
        LEFT JOIN g_in USING(good_id, point_id)
        LEFT JOIN g_out USING(good_id, point_id)) AS i) AS a
JOIN goods g USING(good_id)
JOIN categories ct USING(category_id)
JOIN points p USING(point_id)
```