kokchun giang

common table expression (CTE) in sql to modularize and reuse code for better readability

purpose of using **CTEs** in your queries

enhance readability

reuse intermediate results

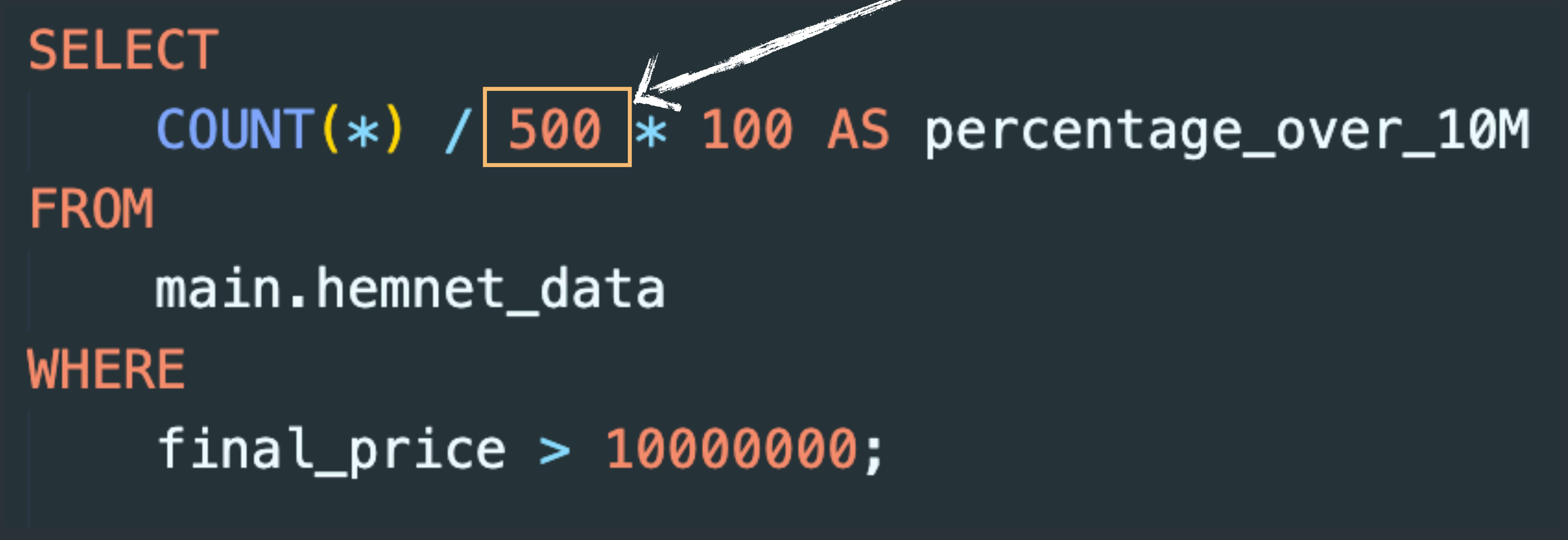
easier debugging and query development

separate complex logic into smaller steps

we start with an example without **CTE**

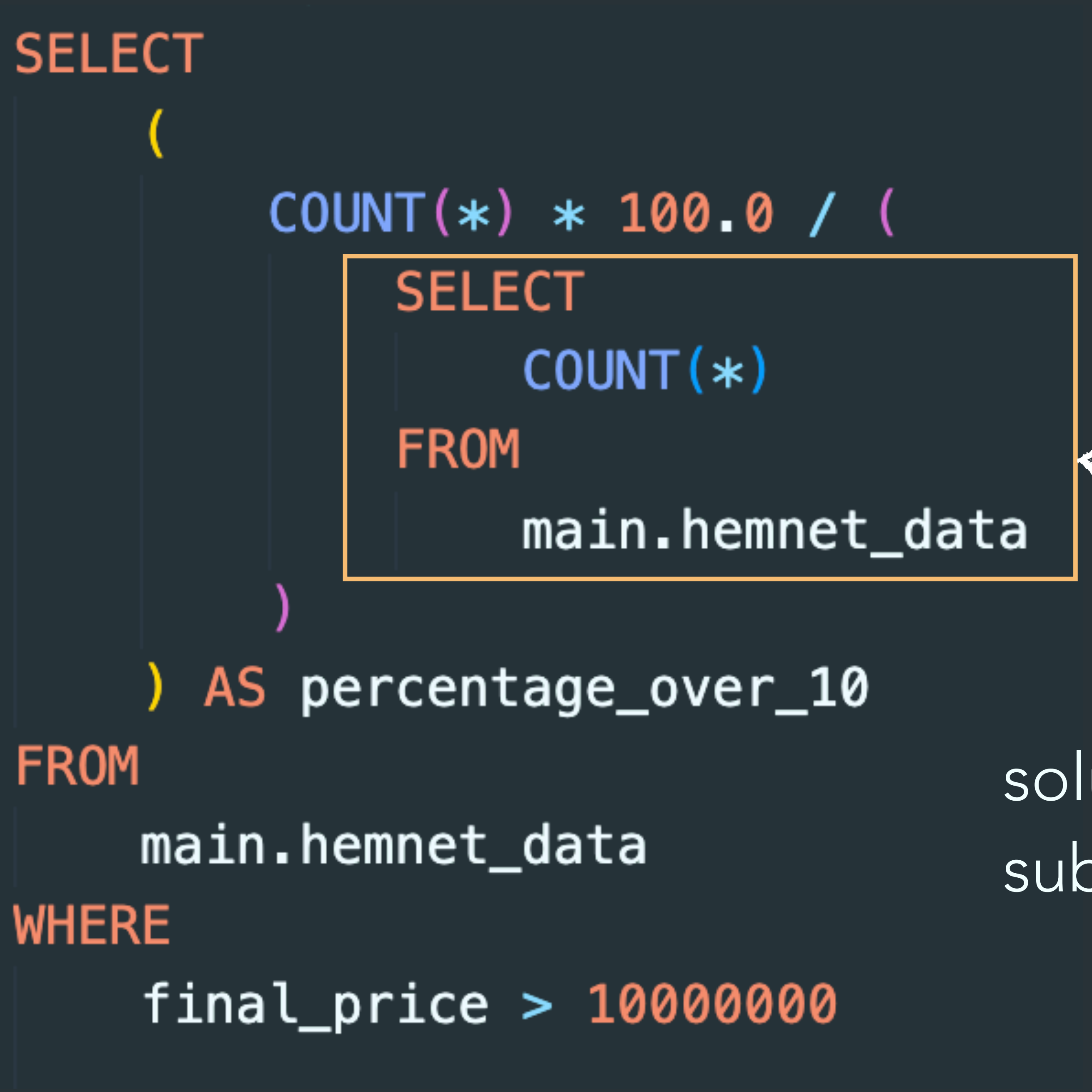
question: calculate percentage of homes that cost more than 10 million

hardcoded solution



```
SELECT
    COUNT(*) / 500 * 100 AS percentage_over_10M
FROM
    main.hemnet_data
WHERE
    final_price > 10000000;
```

hardcoding loses flexibility

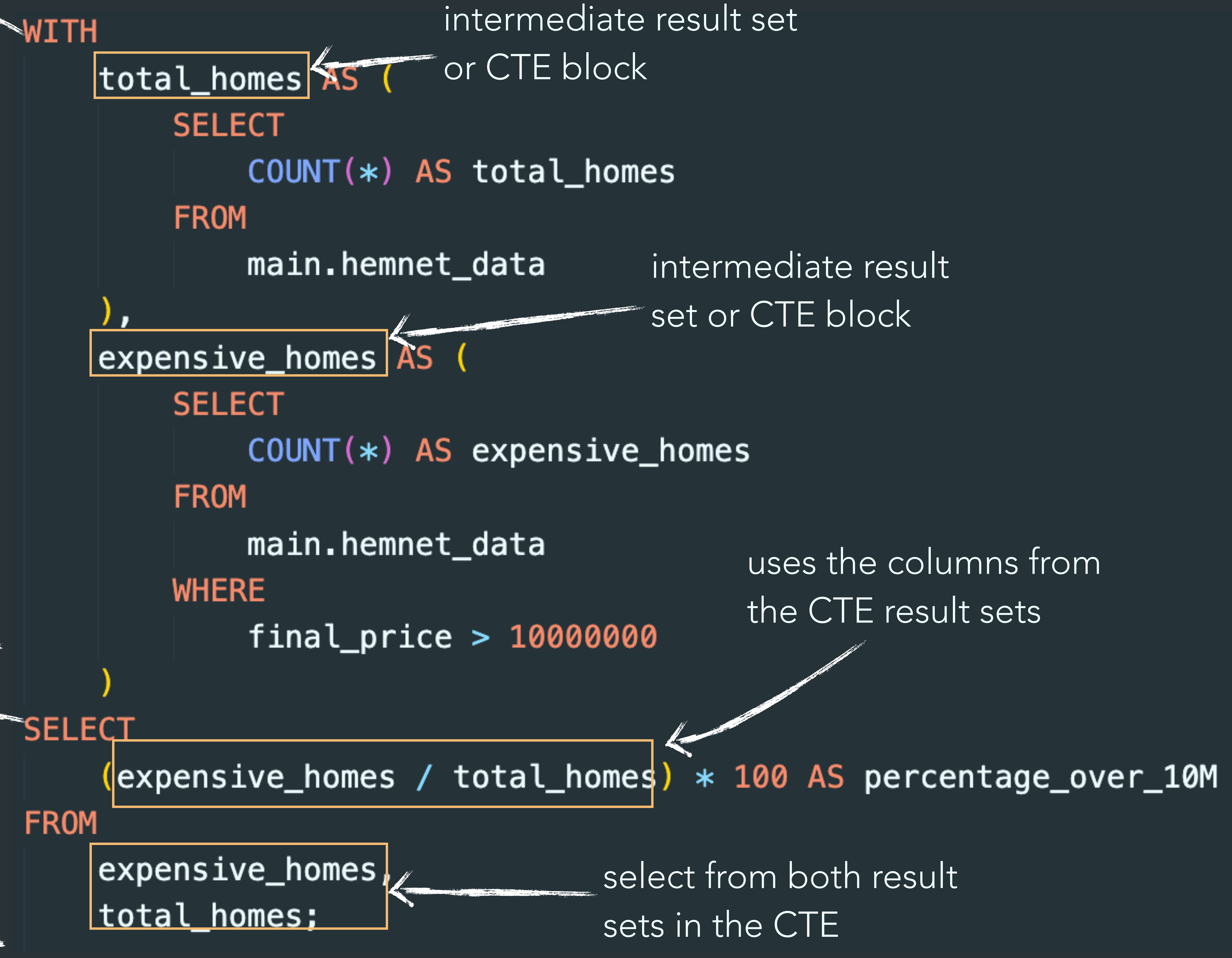


```
SELECT
    (
        COUNT(*) * 100.0 / (
            SELECT
                COUNT(*)
            FROM
                main.hemnet_data
        )
    ) AS percentage_over_10
FROM
    main.hemnet_data
WHERE
    final_price > 10000000
```

solution using subquery

subquery makes code harder to read

now lets solve this using **CTE**



```
WITH
    total_homes AS (
        SELECT
            COUNT(*) AS total_homes
        FROM
            main.hemnet_data
    ),
    expensive_homes AS (
        SELECT
            COUNT(*) AS expensive_homes
        FROM
            main.hemnet_data
        WHERE
            final_price > 10000000
    )
SELECT
    (expensive_homes / total_homes) * 100 AS percentage_over_10M
FROM
    expensive_homes,
    total_homes;
```

CTE part

main query

intermediate result set or CTE block

intermediate result set or CTE block

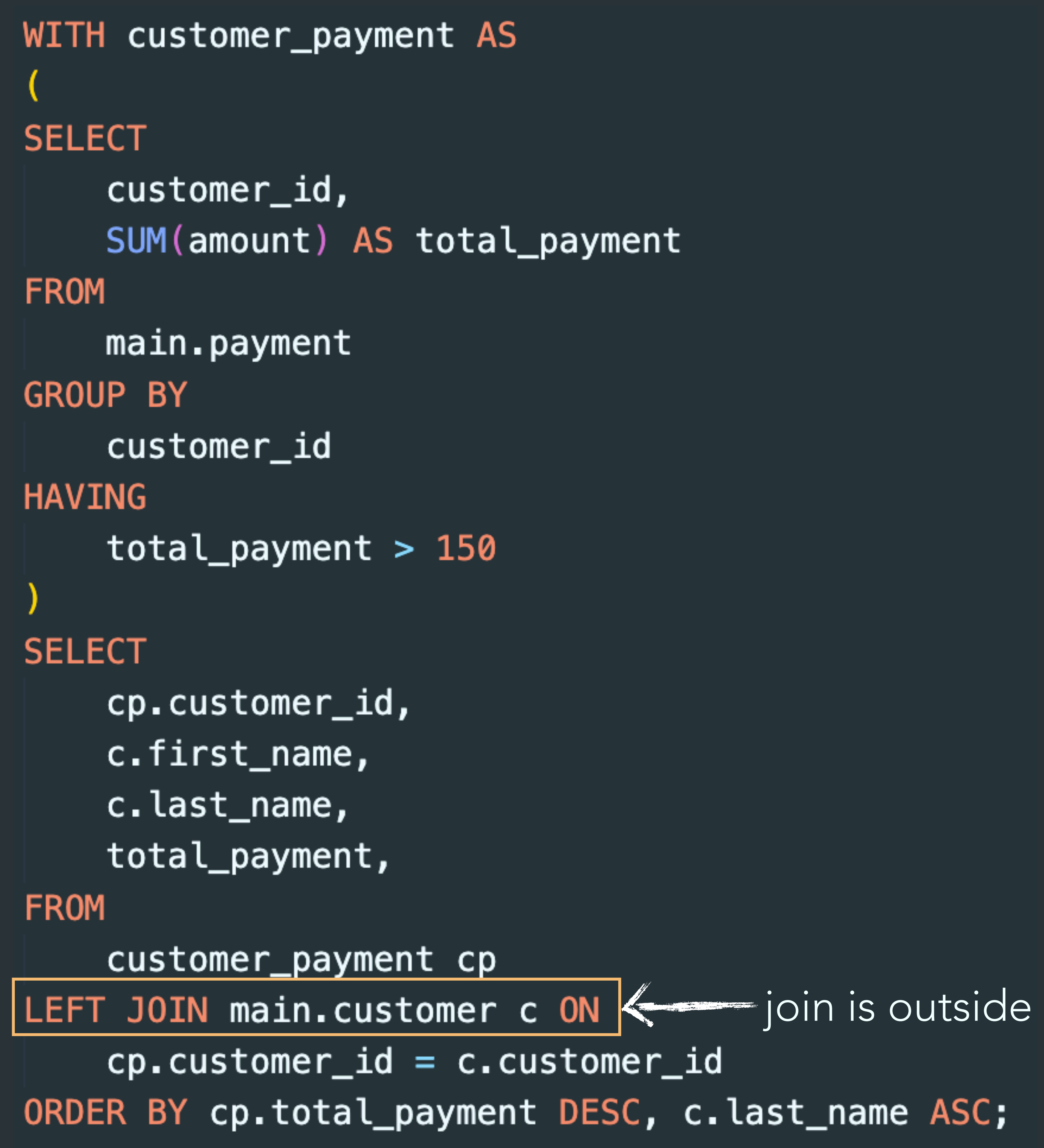
uses the columns from the CTE result sets

select from both result sets in the CTE

separates the logic so that we can reuse intermediate result sets, similarly to variables

easier to debug and build complex queries

joining tables and CTEs



```
WITH customer_payment AS
(
SELECT
    customer_id,
    SUM(amount) AS total_payment
FROM
    main.payment
GROUP BY
    customer_id
HAVING
    total_payment > 150
)
SELECT
    cp.customer_id,
    c.first_name,
    c.last_name,
    total_payment,
FROM
    customer_payment cp
LEFT JOIN main.customer c ON
    cp.customer_id = c.customer_id
ORDER BY cp.total_payment DESC, c.last_name ASC;
```

join is outside of CTE

joins can be done inside of CTEs and outside of CTEs

which to pick depends on which that gives best readability and reusability for your case

general guideline is to keep each CTE block modular and have a single responsibility