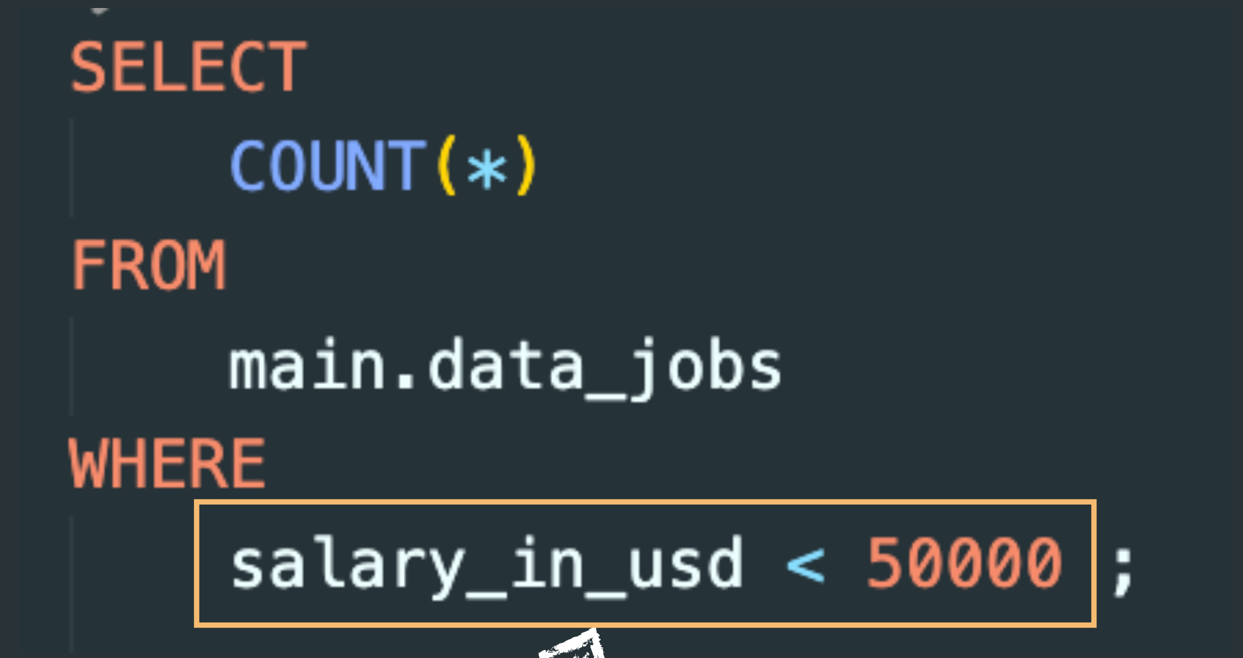
kokchun giang

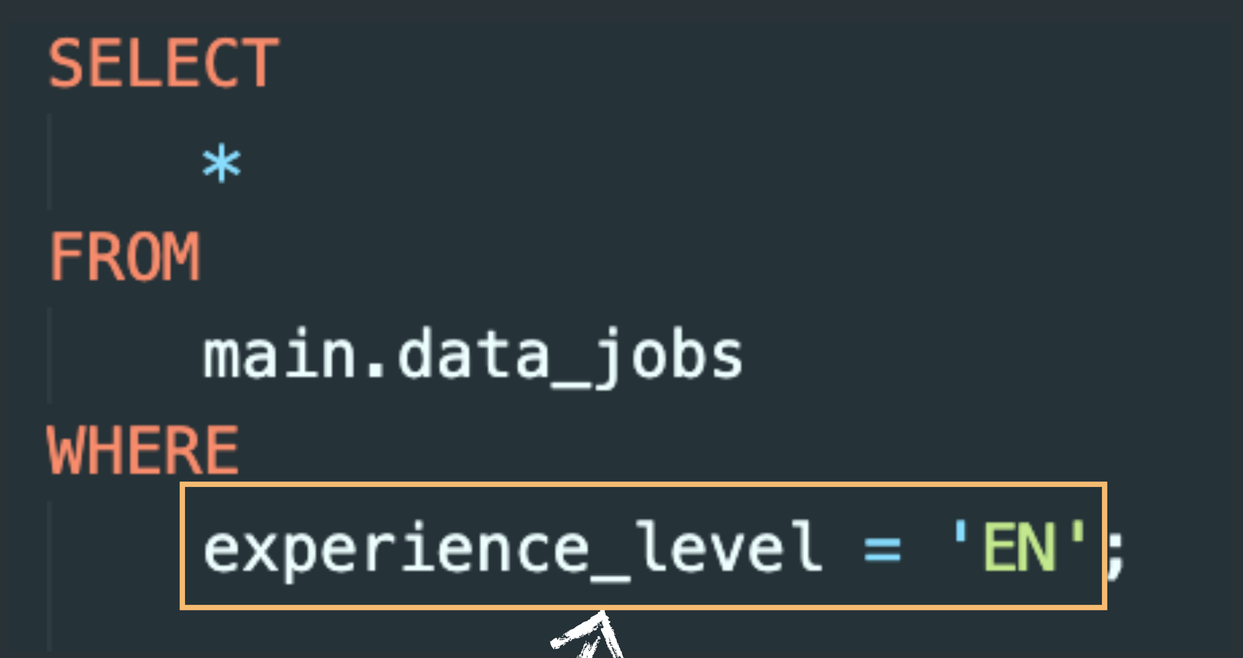
# **filtering** data to get more meaningful results

# using **WHERE clause** to filter rows based on conditions



```
SELECT
    COUNT(*)
FROM
    main.data_jobs
WHERE
    salary_in_usd < 50000 ;
```

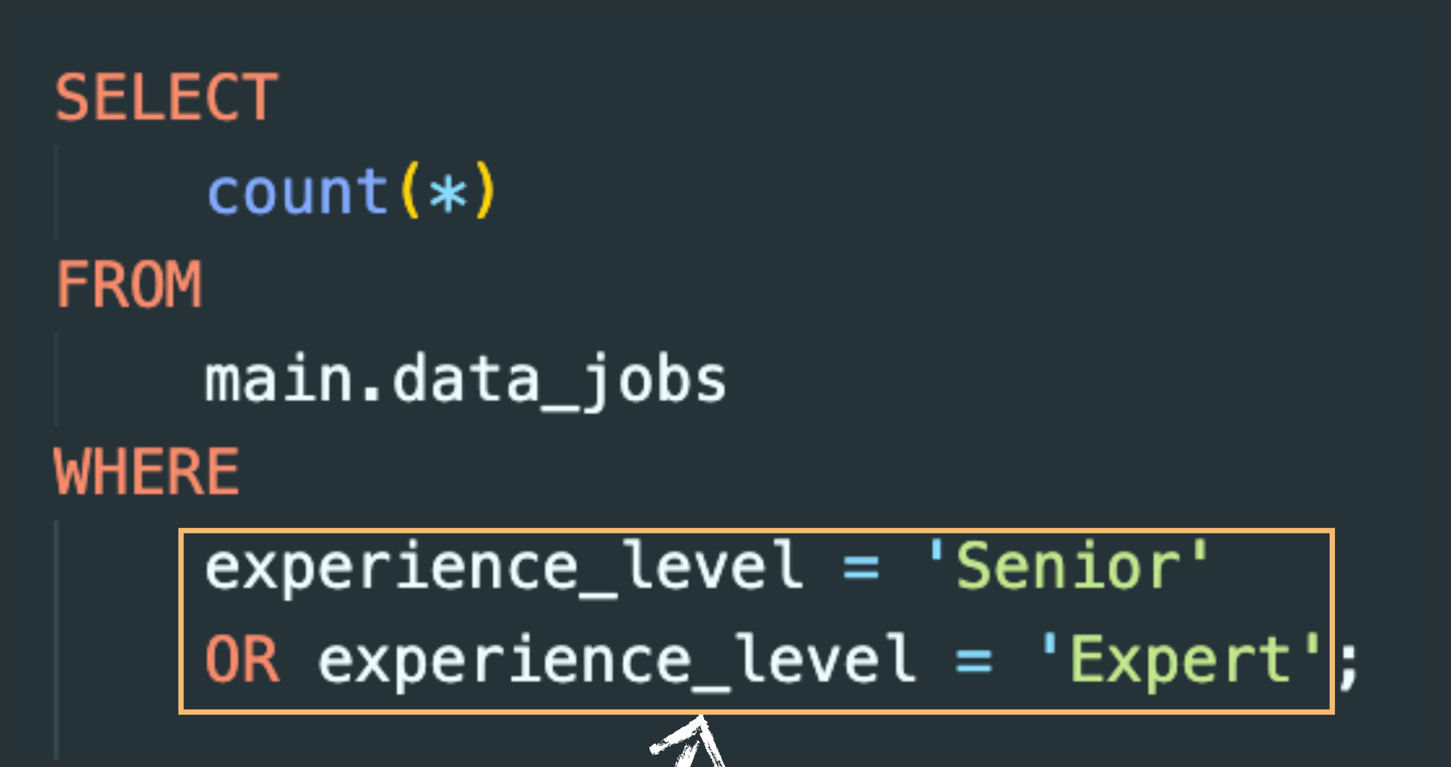
comparison condition
with an upper bound



```
SELECT
    *
FROM
    main.data_jobs
WHERE
    experience_level = 'EN';
```

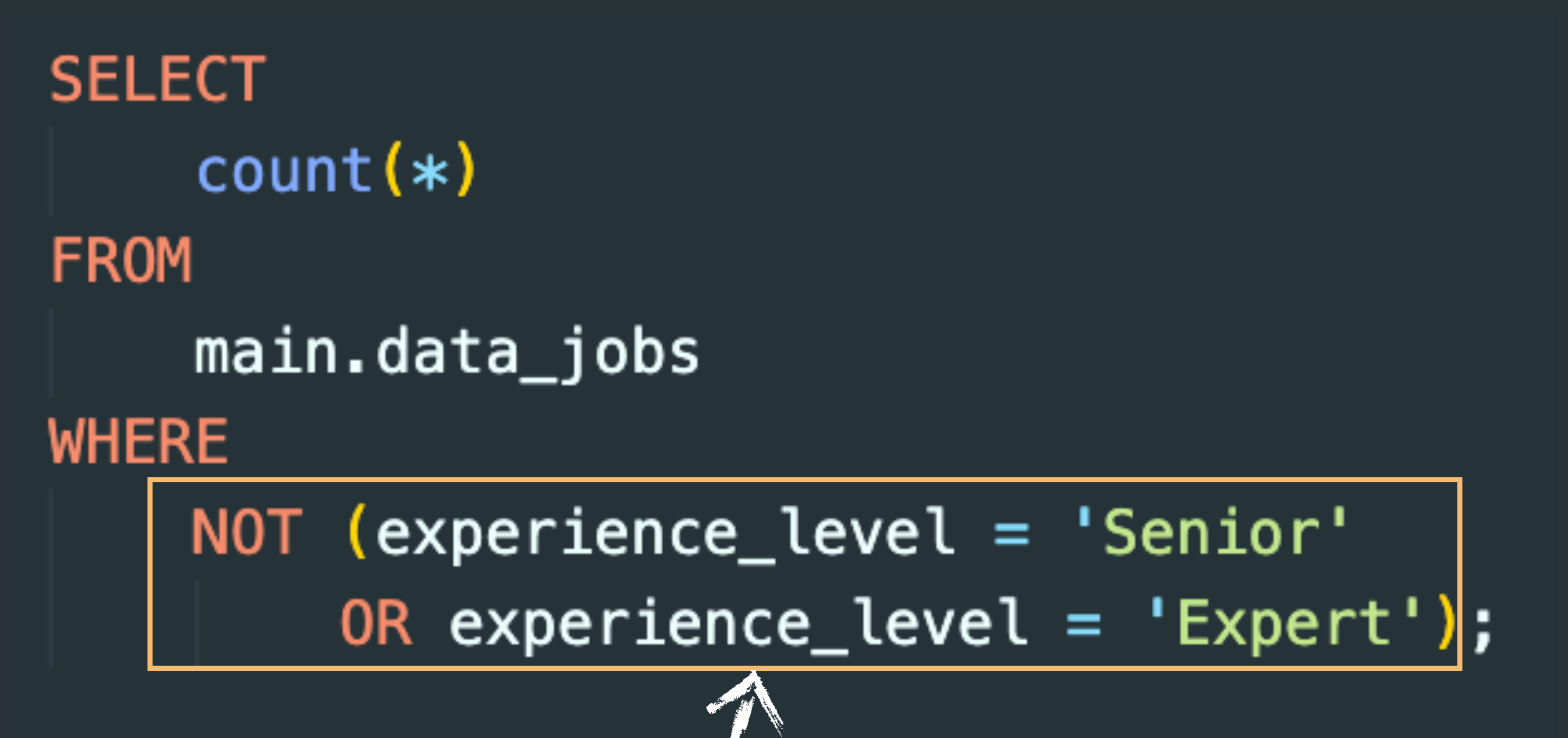
equality
condition

# logical operators to combine multiple conditions allowing for complex filtering criteria



```sql
SELECT
    count(*)
FROM
    main.data_jobs
WHERE
    experience_level = 'Senior'
    OR experience_level = 'Expert';
```

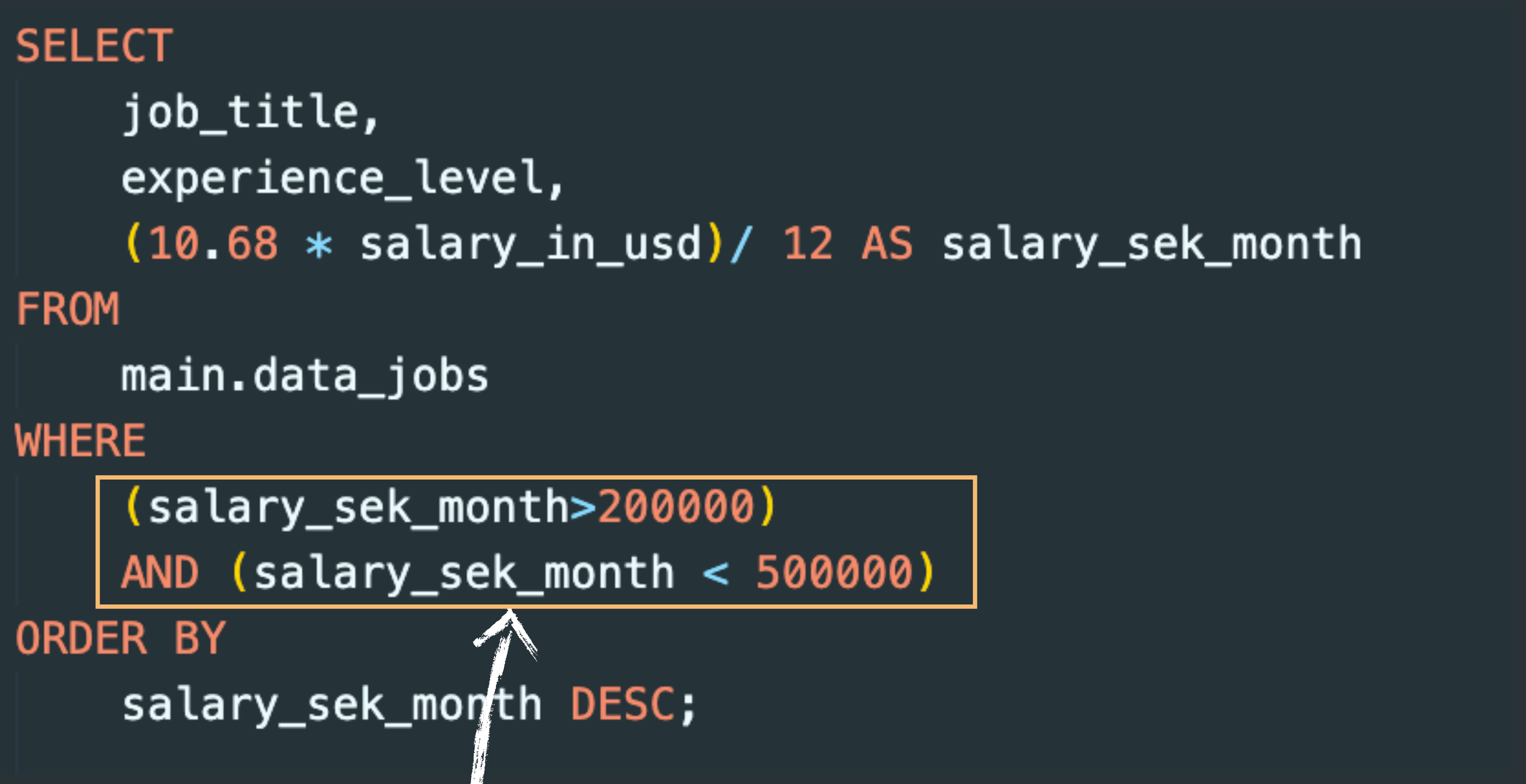
logical operator OR means experience_level can be either 'Senior' or 'Expert'



```sql
SELECT
    count(*)
FROM
    main.data_jobs
WHERE
    NOT (experience_level = 'Senior'
        OR experience_level = 'Expert');
```

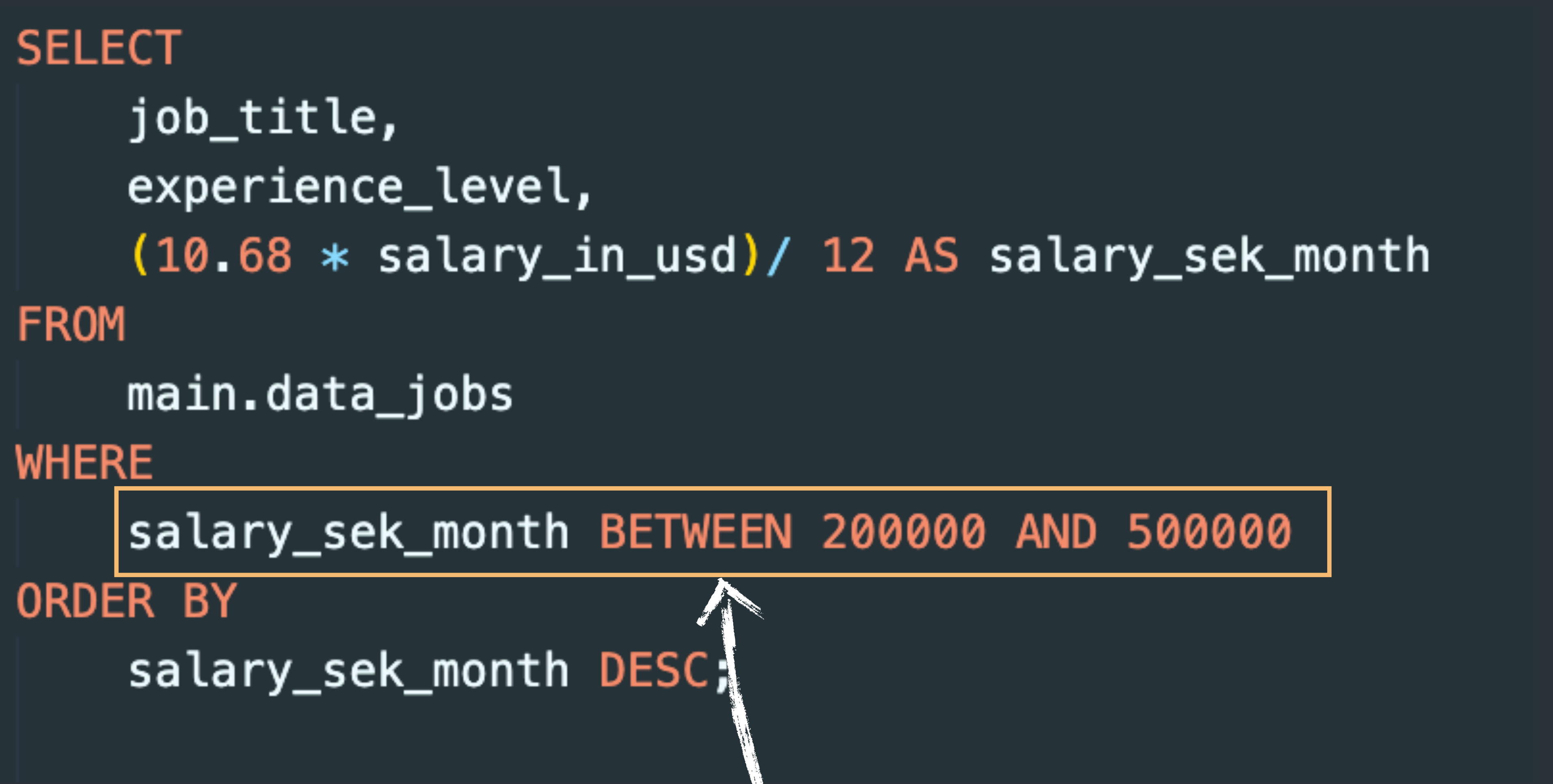
logical NOT operator negates what is inside the parenthesis

# retrieving data in a specific range with **range filtering**



```sql
SELECT
    job_title,
    experience_level,
    (10.68 * salary_in_usd)/ 12 AS salary_sek_month
FROM
    main.data_jobs
WHERE
    (salary_sek_month>200000)
    AND (salary_sek_month < 500000)
ORDER BY
    salary_sek_month DESC;
```

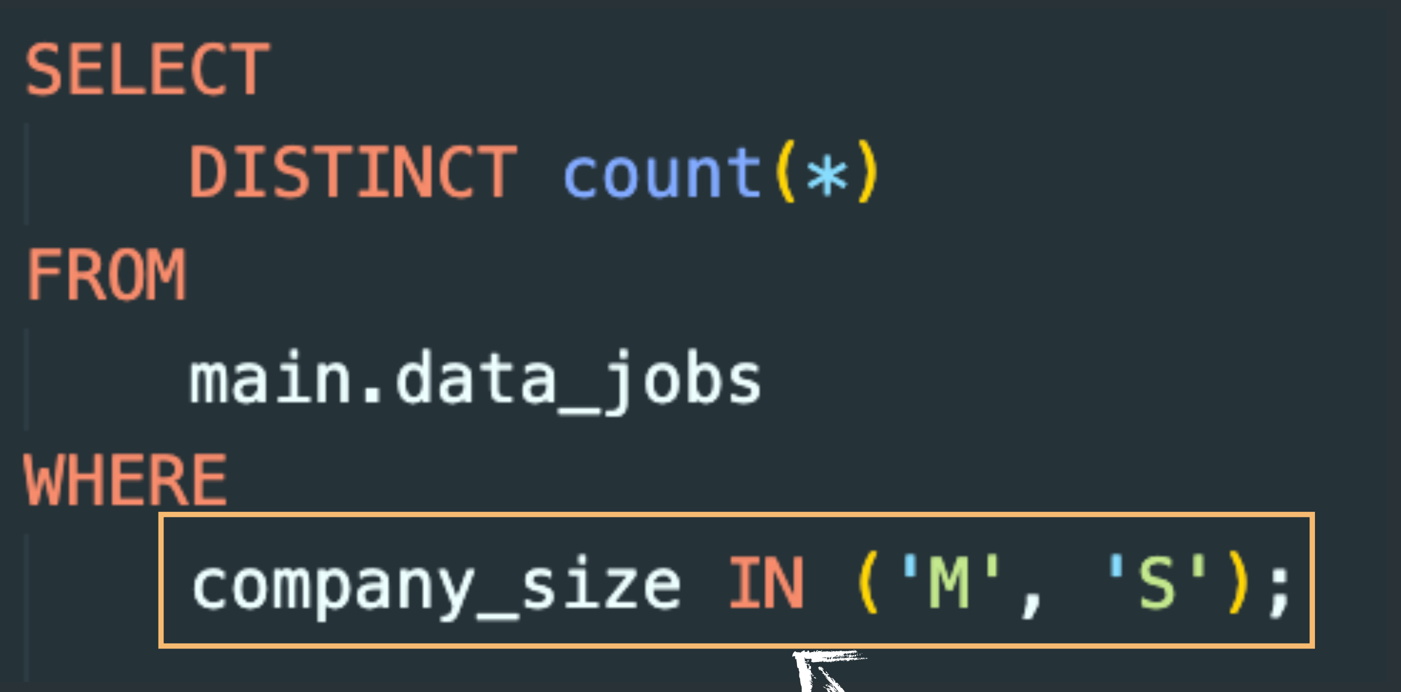
range filtering using a combination of comparison and logical operators



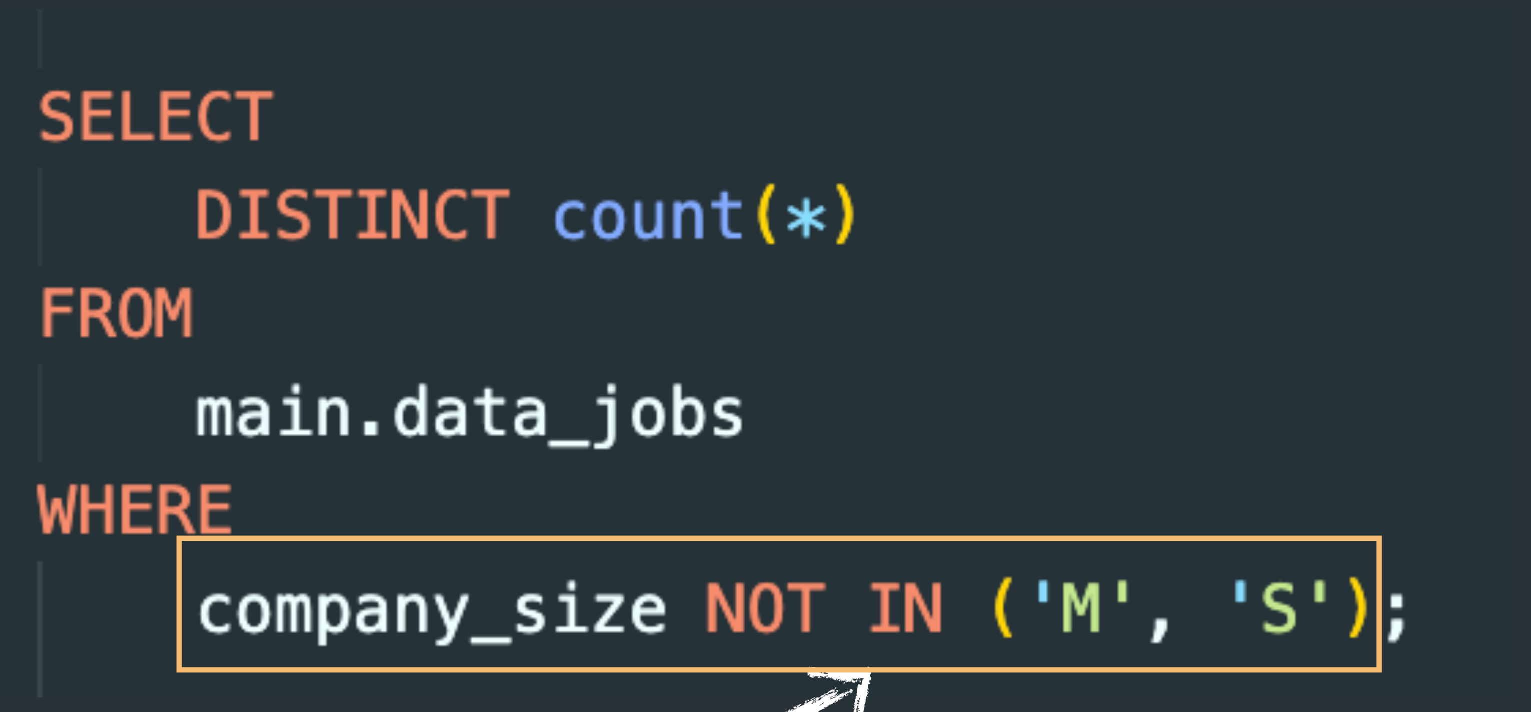
```sql
SELECT
    job_title,
    experience_level,
    (10.68 * salary_in_usd)/ 12 AS salary_sek_month
FROM
    main.data_jobs
WHERE
    salary_sek_month BETWEEN 200000 AND 500000
ORDER BY
    salary_sek_month DESC;
```

range filtering using BETWEEN … AND operator

# filtering from a list with **list filtering**



```
SELECT
    DISTINCT count(*)
FROM
    main.data_jobs
WHERE
    company_size IN ('M', 'S');
```

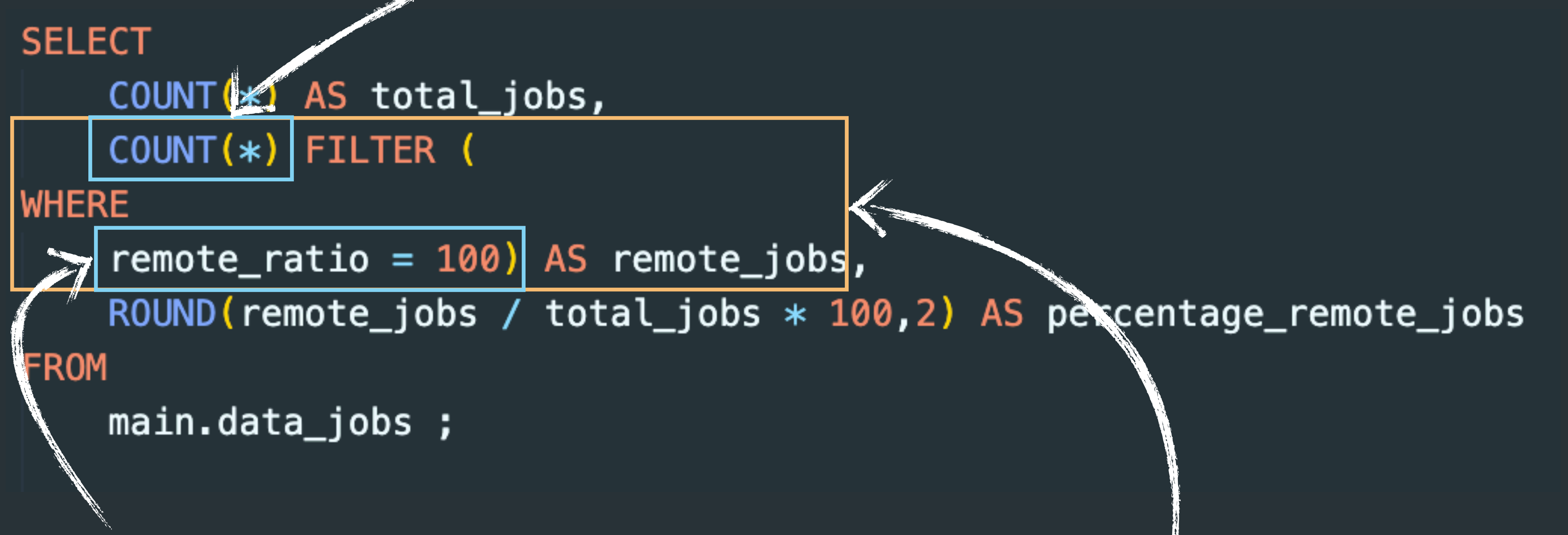


```
SELECT
    DISTINCT count(*)
FROM
    main.data_jobs
WHERE
    company_size NOT IN ('M', 'S');
```

list filtering using IN operator

# the **FILTER clause** to calculate aggregates based on a criteria

aggregate function



```
SELECT
    COUNT(*) AS total_jobs,
    COUNT(*) FILTER (
WHERE
    remote_ratio = 100) AS remote_jobs,
    ROUND(remote_jobs / total_jobs * 100,2) AS percentage_remote_jobs
FROM
    main.data_jobs ;
```

condition

FILTER clause used with aggregate functions to apply conditions to specific aggregate calculations