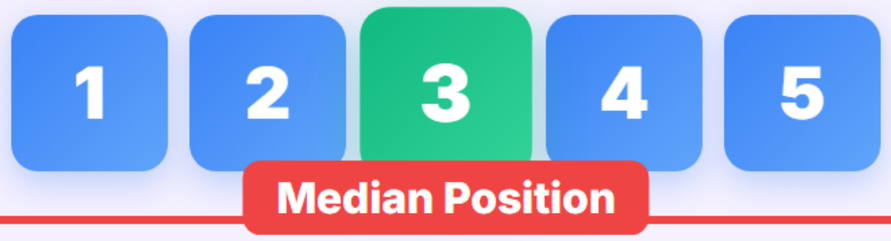
SQL Median Calculation

Calculate the median value from a list of numbers

Common Interview Question

Definition: The median is the **middle value** in a sorted list of numbers. For datasets with an **odd number** of values, it's the exact middle. For datasets with an **even number** of values, it's the **average of the two middle values**.

📊 **Visual Example:**



- 5 values → Odd count → Median is the 3rd value
- [1, 2, 3, 4, 5] → Median = 3

Key Challenge:

SQL doesn't have a built-in MEDIAN() function in all database systems

SQL Median Calculation

1 **Sort the values**

[1, 2, 3, 4, 5]

2 **Count the number of values**

n = 5

3 **Determine if n is odd or even**

5 is odd

4 **Calculate median position**

For odd n, median position = (n + 1) / 2 = (5 + 1) / 2 = 3

5 **Find the value at position 3**

The 3rd value is 3

💡 **Key Insight**

For an even number of values (e.g., [1, 2, 3, 4]), the median would be the average of the two middle values:

(2 + 3) / 2 = 2.5

SQL Median Calculation

🍎 Method 1: PERCENTILE_CONT() - SQL Server

SQL Server 2012+, PostgreSQL 9.4+, Oracle

```sql
SELECT DISTINCT
    PERCENTILE_CONT(0.5)
        WITHIN GROUP (ORDER BY value)
        OVER () AS median
FROM numbers
  -- 0.5 represents the 50th percentile (median)
-- Automatically handles both odd and even counts
```

⚡ Advantages

- **Simple & Readable:** One function call
- **Automatic Handling:** Works for both odd and even counts
- **Efficient:** Optimized by database engine
- **Standard:** Available in modern SQL databases

🍎 Method 2: Using ROW_NUMBER() + COUNT

Works in almost all databases

```sql
WITH ordered AS (
    SELECT
        value,
        ROW_NUMBER() OVER (ORDER BY value) AS rn,
        COUNT(*) OVER () AS cnt
    FROM numbers
)
SELECT AVG(value) AS median
FROM ordered
WHERE rn BETWEEN (cnt + 1) / 2 AND (cnt + 2) / 2;
-- (cnt + 1)/2 and (cnt + 2)/2 correctly select the median rows for both odd and even counts using integer division.
```

🎯 HOW IT WORKS

Step-by-Step:

1. Assign row numbers to sorted values
2. Get total count of rows
3. Calculate median position range:
   - For 5 rows: BETWEEN (5+1)/2 AND (5+2)/2 = BETWEEN 3 AND 3.5 = row 3
   - For 4 rows: BETWEEN (4+1)/2 AND (4+2)/2 = BETWEEN 2.5 AND 3 = rows 2 & 3
4. Average the selected rows

🍎 Method 3: Using ROW_NUMBER() - Easy but less optimized

Universal SQL Solution

```sql
WITH ranked_data AS (
  SELECT
    value_column,
    ROW_NUMBER() OVER (ORDER BY value_column) AS row_asc,
    ROW_NUMBER() OVER (ORDER BY value_column DESC) AS
row_desc
  FROM table_name
)
SELECT AVG(value_column) AS median
FROM ranked_data
WHERE row_asc = row_desc OR row_asc + 1 = row_desc;
-- This works because for median, ascending and descending
row numbers meet in the middle
```

🎯 WHY THIS WORKS

The ascending/descending row number trick:

For dataset [1, 2, 3, 4, 5]:

- Ascending rows: 1→1, 2→2, 3→3, 4→4, 5→5
- Descending rows: 1→5, 2→4, 3→3, 4→2, 5→1

Median rows are where: row_asc = row_desc (row 3)

For [1, 2, 3, 4]:

- Where row_asc = row_desc OR row_asc + 1 = row_desc
- This selects rows 2 and 3 for averaging