

HAVING Clause

ADQL Reference:

Overview

Use the **HAVING** clause on the output produced by another aggregation. The **HAVING** clause operates on an expression to determine the intervals to be shown in the response. The metric specified in the expression must be numeric and the expression must return a boolean value for the **HAVING** clause to work.

Additionally, the **HAVING** clause must be referenced by an alias in the query as follows:

```
SELECT customerName, count(*) as Requests, avg(responseTime) as  
ResponseTime FROM transactions HAVING Requests > 10000
```

where `Requests` is the alias referenced in the query.

Syntax

```
SELECT selectItems  
FROM relation  
WHERE where=booleanExpression  
SINCE statement  
HAVING havingClause  
ORDER BY sortItems  
LIMIT limits  
;
```

where `havingClause` is a boolean expression. The **WHERE**, **ORDER BY**, and **LIMIT** clauses and **SINCE** `statement` are optional.

Examples

Usage of **HAVING** clause with different filters is given below:

Simple Comparison Filter

```
SELECT transactionName, avg(responseTime) as AVRT FROM type HAVING  
AVRT > 0
```

Simple Range Filter

```
SELECT transactionName, avg(responseTime) as AVRT FROM type HAVING  
AVRT BETWEEN [10, 90)
```

Simple List Filter

```
SELECT transactionName, avg(responseTime) as AVRT FROM type HAVING  
AVRT IN (20, 30, 40, 60)
```

Compound Filter

```
SELECT transactionName, avg(responseTime) as AVRT FROM type HAVING  
AVRT > 20 AND AVRT < 90  
SELECT transactionName, avg(responseTime) as AVRT FROM type HAVING  
AVRT < 20 OR AVRT > 90  
SELECT transactionName, avg(responseTime) as AVRT FROM type HAVING  
AVRT > 20 AND AVRT < 90 AND AVRT IN (70, 80)
```

Not Filter

```
SELECT transactionName, avg(responseTime) as AVRT FROM type HAVING
NOT AVRT > 50
```

Not Compound Filter

```
SELECT transactionName, avg(responseTime) as AVRT FROM type HAVING
NOT (AVRT > 20 AND AVRT < 90)
```

Multi-Metric Filter

```
SELECT transactionName, avg(responseTime) as AVRT, max(responseTime)
as MAXRT FROM type HAVING NOT (AVRT > 20 AND AVRT < 90) AND MAXRT >
130
```

Limitations

The **HAVING** clause is not supported for the following:

- Nested fields in Elasticsearch
- Multi-metric functions, such as `percentiles` and `stdev`
- Filtered metric functions (filter function)
- Invalid queries as given below

Unsupported Conditions/Filters	Queries
Like	SELECT transactionName, avg(responseTime) as AVRT FROM transactions HAVING AVRT LIKE '10'
Exists	SELECT transactionName, avg(responseTime) as AVRT FROM transactions HAVING AVRT IS NOT NULL
Not Aliased	SELECT avg(responseTime) as avtr FROM type HAVING avg(responseTime) > 10 SELECT txName, count(*), node, count(*) FROM transactions HAVING count(*) > 10
Nested	SELECT transactionId, AVG(segments.numCalls) as av_numCalls, MIN(segments.numCalls), MAX(segments.numCalls), SUM(segments.numCalls) FROM transactions HAVING av_numCalls > 2

Filter Function	SELECT appraisalrating, filter(min(salary), where salary > 60000) as val FROM transactions HAVING val > 60000
Multi-Valued Metric Functions	<p>Percentile :</p> <p>SELECT appraisalrating, percentile(salary, 94) as per_sal FROM transactions HAVING per_sal > 2 ORDER By appraisalrating</p> <p>Standard Deviation :</p> <p>SELECT appraisalrating, stdev(salary) as stdev_sal FROM transactions HAVING stdev_sal > 2 ORDER By appraisalrating</p>
Missing Parent Aggregation	SELECT avg(responseTime) as AVRT FROM transactions HAVING AVRT > 2