Opendistro-sql-jdbc 支持操作

来源文档：<https://opendistro.github.io/for-elasticsearch-docs/docs/sql/operations/>

## Statements

| **Statement** | **Example** |
| --- | --- |
| Select | SELECT \* FROM my-index |
| Delete | DELETE FROM my-index WHERE \_id=1 |
| Where | SELECT \* FROM my-index WHERE ['field']='value' |
| Order by | SELECT \* FROM my-index ORDER BY \_id asc |
| Group by | SELECT \* FROM my-index GROUP BY range(age, 20,30,39) |
| Limit | SELECT \* FROM my-index LIMIT 50聽(default is 200) |
| Union | SELECT \* FROM my-index1 UNION SELECT \* FROM my-index2 |
| Minus | SELECT \* FROM my-index1 MINUS SELECT \* FROM my-index2 |

Like any complex query, large UNION and MINUS statements can strain or even crash your cluster.

## Conditions

| **Condition** | **Example** |
| --- | --- |
| Like | SELECT \* FROM my-index WHERE name LIKE 'j%' |
| And | SELECT \* FROM my-index WHERE name LIKE 'j%' AND age > 21 |
| Or | SELECT \* FROM my-index WHERE name LIKE 'j%' OR age > 21 |
| Count distinct | SELECT count(distinct age) FROM my-index |
| In | SELECT \* FROM my-index WHERE name IN ('alejandro', 'carolina') |
| Not | SELECT \* FROM my-index WHERE name NOT IN ('jane') |
| Between | SELECT \* FROM my-index WHERE age BETWEEN 20 AND 30 |
| Aliases | SELECT avg(age) AS Average\_Age FROM my-index |
| Date | SELECT \* FROM my-index WHERE birthday='1990-11-15' |
| Null | SELECT \* FROM my-index WHERE name IS NULL |

## Aggregations

| **Aggregation** | **Example** |
| --- | --- |
| avg() | SELECT avg(age) FROM my-index |
| count() | SELECT count(age) FROM my-index |
| max() | SELECT max(age) AS Highest\_Age FROM my-index |
| min() | SELECT min(age) AS Lowest\_Age FROM my-index |
| sum() | SELECT sum(age) AS Age\_Sum FROM my-index |

## Include and exclude fields

| **Pattern** | **Example** |
| --- | --- |
| include() | SELECT include('a\*'), exclude('age') FROM my-index |
| exclude() | SELECT exclude('\*name') FROM my-index |

## Functions

You must enable fielddata in the document mapping for most string functions to work properly.

| **Function** | **Example** |
| --- | --- |
| floor | SELECT floor(number) AS Rounded\_Down FROM my-index |
| trim | SELECT trim(name) FROM my-index |
| log | SELECT log(number) FROM my-index |
| log10 | SELECT log10(number) FROM my-index |
| substring | SELECT substring(name, 2,5) FROM my-index |
| round | SELECT round(number) FROM my-index |
| sqrt | SELECT sqrt(number) FROM my-index |
| concat\_ws | SELECT concat\_ws(' ', age, height) AS combined FROM my-index |
| / | SELECT number / 100 FROM my-index |
| % | SELECT number % 100 FROM my-index |
| date\_format | SELECT date\_format(date, 'Y') FROM my-index |

## Joins

See [Joins](https://opendistro.github.io/for-elasticsearch-docs/docs/sql/joins) for constraints and limitations.

| **Join** | **Example** |
| --- | --- |
| Inner join | SELECT p.firstname, p.lastname, p.gender, dogs.name FROM people p JOIN dogs d ON d.holdersName = p.firstname WHERE p.age > 12 AND d.age > 1 |
| Left outer join | SELECT p.firstname, p.lastname, p.gender, dogs.name FROM people p LEFT JOIN dogs d ON d.holdersName = p.firstname |
| Cross join | SELECT p.firstname, p.lastname, p.gender, dogs.name FROM people p CROSS JOIN dogs d |

## Show

Show commands, well, show you indices and mappings that match an index pattern. You can use \* or % for wildcards.

| **Show** | **Example** |
| --- | --- |
| Show tables like | SHOW TABLES LIKE logs-\* |