

CASSANDRA AGGREGATE FUNCTIONS

Creating table

```
cqlsh> CREATE KEYSPACE test1 with replication =  
... {'class' : 'SimpleStrategy', 'replication_factor' : 1} ;  
cqlsh> USE test1  
... ;  
cqlsh:test1> CREATE TABLE Emp_record  
... (  
...   E_id int PRIMARY KEY,  
...   E_score int,  
...   E_name text,  
...   E_city text  
... );  
cqlsh:test1>  
cqlsh:test1>  
cqlsh:test1> INSERT INTO Emp_record(E_id, E_score, E_name, E_city)  
...   values (101, 85, 'ashish', 'Noida');  
cqlsh:test1> INSERT INTO Emp_record(E_id, E_score, E_name, E_city)  
...   values (102, 90, 'ankur', 'meerut');  
cqlsh:test1> INSERT INTO Emp_record(E_id, E_score, E_name, E_city)  
...   values (103, 99, 'shivang', 'gurugram');  
cqlsh:test1> INSERT INTO Emp_record(E_id, E_score, E_name, E_city)  
...   values (104, 85, 'abi', 'meerut');  
cqlsh:test1> INSERT INTO Emp_record(E_id, E_score, E_city)  
...   values (105, 95, 'mumbai');  
cqlsh:test1> Select *  
... from Emp_record;  
  
e_id | e_city | e_name | e_score  
-----  
105 | mumbai | null | 95  
104 | meerut | abi | 85  
102 | meerut | ankur | 90  
101 | Noida | ashish | 85  
103 | gurugram | shivang | 99  
  
(5 rows)
```

Count

```
cqlsh:test1> SELECT COUNT(*)
... FROM Emp_record;

count
-----
      5

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:test1> SELECT COUNT(1)
... FROM Emp_record;

count
-----
      5

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:test1>
```

MIN MAX

```
cqlsh:test1> SELECT MIN(E_score)
... FROM Emp_record;

system.min(e_score)
-----
85

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:test1>
cqlsh:test1> SELECT MAX(E_score)
... FROM Emp_record;

system.max(e_score)
-----
99

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:test1>
```

SUM

```
cqlsh:test1> SELECT SUM(E_score)
... FROM Emp_record
... ;

system.sum(e_score)
-----
454

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:test1>
```

AVERAGE

```
cqlsh:test1>
cqlsh:test1> SELECT AVG(E_score)
... FROM Emp_record;

system.avg(e_score)
-----
90

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:test1>
```

CODES

CREATE KEYSPACE test1 with replication =

```
{'class' : 'SimpleStrategy', 'replication_factor' : 1};
```

USE test1;

CREATE TABLE Emp_record

(

E_id int PRIMARY KEY,

E_score int,

E_name text,

E_city text

);

INSERT INTO Emp_record(E_id, E_score, E_name, E_city)

```
values (101, 85, 'ashish', 'Noida');
```

```
INSERT INTO Emp_record(E_id, E_score, E_name, E_city)
```

```
values (102, 90, 'ankur', 'meerut');
```

```
INSERT INTO Emp_record(E_id, E_score, E_name, E_city)
```

```
values (103, 99, 'shivang', 'gurugram');
```

```
INSERT INTO Emp_record(E_id, E_score, E_name, E_city)
```

```
values (104, 85, 'abi', 'meerut');
```

```
INSERT INTO Emp_record(E_id, E_score, E_city)
```

```
values (105, 95, 'mumbai');
```

```
Select *
```

```
from Emp_record;
```

```
SELECT COUNT(*)
```

```
FROM Emp_record;
```

Alternatively, To get the same result we can use COUNT(1).

```
SELECT COUNT(1)
```

```
FROM Emp_record;
```

```
SELECT MIN(E_score)
```

```
FROM Emp_record;
```

```
SELECT MAX(E_score)
```

```
FROM Emp_record;
```

```
SELECT SUM(E_score)
```

```
FROM Emp_record;
```

```
SELECT AVG(E_score)
```

```
FROM Emp_record;
```