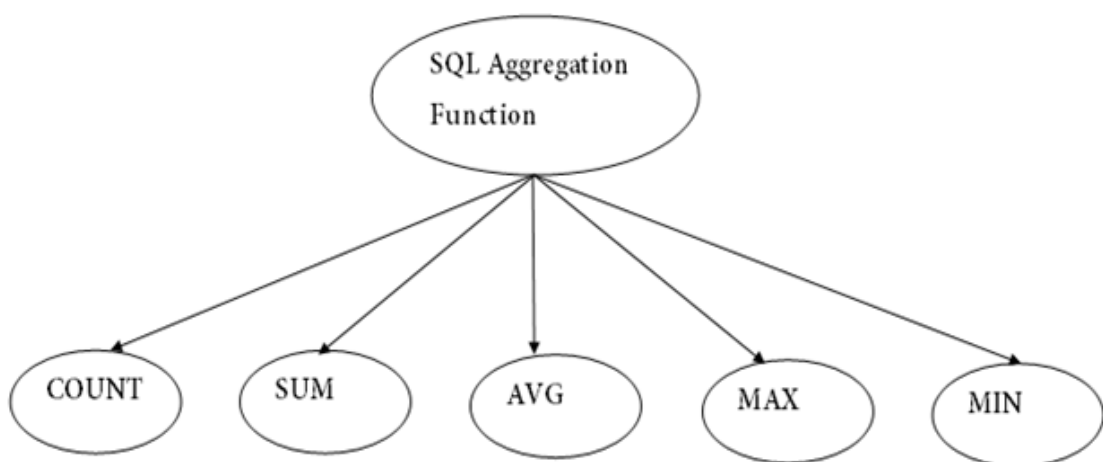


AGGREGATE OPERATORS

❖ DEFINATION:

- SQL aggregation function is used to perform the calculations on multiple rows of a single column of a table. It returns a single value.
- It is also used to summarize the data.
- Execute aggregation operations (\$avg,\$min,\$push,\$addToSet etc.)students encourage to execute several queries to demonstrate various aggregation operators)



❖ Syntax:

db.collection.aggregate(<AGGREGATE OPERATION>)

❖ TYPES:

Expression Type	Description	Syntax
Accumulators	Perform calculations on entire groups of documents	
* \$sum	Calculates the sum of all values in a numeric field within a group.	"\$fieldName": { \$sum: "\$fieldName" }
* \$avg	Calculates the average of all values in a numeric field within a group.	"\$fieldName": { \$avg: "\$fieldName" }
* \$min	Finds the minimum value in a field within a group.	"\$fieldName": { \$min: "\$fieldName" }
* \$max	Finds the maximum value in a field within a group.	"\$fieldName": { \$max: "\$fieldName" }
* \$push	Creates an array containing all unique or duplicate values from a field	"\$arrayName": { \$push: "\$fieldName" }
* \$addToSet	Creates an array containing only unique values from a field within a group.	"\$arrayName": { \$addToSet: "\$fieldName" }
* \$first	Returns the first value in a field within a group (or entire collection).	"\$fieldName": { \$first: "\$fieldName" }
* \$last	Returns the last value in a field within a group (or entire collection).	"\$fieldName": { \$last: "\$fieldName" }

1.Average GPA of all students:

```
Please enter a MongoDB connection string (Default: mongodb://locdb> db.stu.aggregate([
... {$group:{_id:null,averageGPA:{$avg:"$gpa"}}}
... ]);
```

Output:

```
[ { _id: null, averageGPA: 2.98556 } ]
```

Explanation:

- \$group: Groups all documents together.
- _id: null: Sets the group identifier to null (optional, as there's only one group in this case).
- averageGPA: Calculates the average value of the "gpa" field using the \$avg operator.

2.Minimum and Maximum age:

```
db> db.stu.aggregate([
...   {$group: {_id:null,minAge:{$min:"$age"},maxAge:{$max:"$age"}}}
... ]);
```

Output:

```
db> db.stu.aggregate([
...  {$group:{_id:null,minAge:{$min:"$age"},maxAge:{$max:"$age"}}
...  });
[ { _id: null, minAge: 18, maxAge: 25 } ]
```

Explanation:

- Similar to the previous example, it uses \$group to group all documents.
- minAge: Uses the \$min operator to find the minimum value in the "age" field.
- maxAge: Uses the \$max operator to find the maximum value in the "age" field.

3.To Get Average GPA For All Home Cities:

```
db> db.stu.aggregate([
... {$group: {_id: "$home_city", averageGPA: {$avg: "$gpa"}}}
... ]);
```

Output:

```
db> db.stu.aggregate([
... {$group: {_id: "$home_city", averageGPA: {$avg: "$gpa"}}}
... ]);
[
  { _id: null, averageGPA: 2.9784313725490197 },
  { _id: 'City 1', averageGPA: 3.003823529411765 },
  { _id: 'City 7', averageGPA: 2.847931034482759 },
  { _id: 'City 6', averageGPA: 2.8969444444444448 },
  { _id: 'City 3', averageGPA: 3.0100000000000002 },
  { _id: 'City 2', averageGPA: 3.01969696969697 },
  { _id: 'City 4', averageGPA: 2.8251851851851852 },
  { _id: 'City 8', averageGPA: 3.11741935483871 },
  { _id: 'City 10', averageGPA: 2.935227272727273 },
  { _id: 'City 9', averageGPA: 3.1174358974358976 },
  { _id: 'City 5', averageGPA: 3.0607499999999996 }
]
```

4.Pushing All Courses Into A Single Array:

```
db.students.aggregate([
  { $project: { _id: 0, allCourses: { $push: "$courses" } } }
]);
```

Explanation:

- \$project: Transforms the input documents.
- _id: 0: Excludes the _id field from the output documents.
- allCourses: Uses the \$push operator to create an array. It pushes all elements from the "courses" field of each student document into the allCourses array.

Output:

```
db> db.students.aggregate([
...   { $project: { _id: 0, allCourses: { $push: "$courses" } } }
... ]);
MongoServerError[Location31325]: Invalid $project :: caused by :: Unknown expression $push
db> |
```

This will return a list of documents, each with an allCourses array containing all unique courses offered (assuming courses might be duplicated across students).

This is because our Array is incorrect.