

以下是所有 MongoDB 聚合框架中的常用 ``$`` 操作符，已按照分類進行整理，包括比較、邏輯、數學、字符串、數組、日期、條件、數據類型等操作符。這份清單應涵蓋了 MongoDB 中常用的所有聚合操作符。

操作符	說明	語法和示例
<code>`\$eq`</code>	等於	<code>`{ field: { \$eq: value } }`</code> 例如： <code>`{ age: { \$eq: 30 } }`</code>
<code>`\$ne`</code>	不等於	<code>`{ field: { \$ne: value } }`</code> 例如： <code>`{ status: { \$ne: "A" } }`</code>
<code>`\$gt`</code>	大於	<code>`{ field: { \$gt: value } }`</code> 例如： <code>`{ price: { \$gt: 100 } }`</code>
<code>`\$gte`</code>	大於或等於	<code>`{ field: { \$gte: value } }`</code> 例如： <code>`{ age: { \$gte: 18 } }`</code>
<code>`\$lt`</code>	小於	<code>`{ field: { \$lt: value } }`</code> 例如： <code>`{ price: { \$lt: 50 } }`</code>
<code>`\$lte`</code>	小於或等於	<code>`{ field: { \$lte: value } }`</code> 例如： <code>`{ stock: { \$lte: 20 } }`</code>
<code>`\$in`</code>	在指定數組中	<code>`{ field: { \$in: [value1, value2, ...] } }`</code> 例如： <code>`{ category: { \$in: ["Electronics"] } }`</code>
<code>`\$nin`</code>	不在指定數組中	<code>`{ field: { \$nin: [value1, value2, ...] } }`</code> 例如： <code>`{ status: { \$nin: ["B", "C"] } }`</code>

2. 邏輯操作符 (Logical Operators)

操作符	說明	語法和示例
<code>`\$and`</code>	邏輯與，所有條件必須為真	<code>`{ \$and: [{ condition1 }, { condition2 }, ...] }`</code> 例如： <code>`{ \$and: [{ price: { \$gt: 100 } }, { stock: { \$lte: 50 } }] }`</code>
<code>`\$or`</code>	邏輯或，至少有一個條件為真	<code>`{ \$or: [{ condition1 }, { condition2 }, ...] }`</code> 例如： <code>`{ \$or: [{ status: "A" }, { quantity: { \$lt: 20 } }] }`</code>
<code>`\$not`</code>	邏輯非，反轉指定條件的布爾值	<code>`{ field: { \$not: { condition } } }`</code> 例如： <code>`{ price: { \$not: { \$gt: 100 } } }`</code>
<code>`\$nor`</code>	邏輯非或，所有條件必須為假	<code>`{ \$nor: [{ condition1 }, { condition2 }, ...] }`</code> 例如： <code>`{ \$nor: [{ status: "A" }, { qty: { \$lt: 20 } }] }`</code>

3. 數學操作符 (Arithmetic Operators)

操作符	說明	語法和示例
<code>`\$add`</code>	將兩個或更多數字相加	<code>`{ \$add: [<expression1>, <expression2>, ...] }`</code> 例如： <code>`{ total: { \$add: ["\$price", "\$tax"] } }`</code>
<code>`\$subtract`</code>	從第一個數字減去第二個數字	<code>`{ \$subtract: [<expression1>, <expression2>] }`</code> 例如： <code>`{ difference: { \$subtract: ["\$price", "\$discount"] } }`</code>

操作符	說明	語法和示例
<code>`\$multiply`</code>	將兩個或更多數字相乘	<code>`{ \$multiply: [<expression1>, <expression2>, ...] }`</code> 例如： <code>`{ total: { \$multiply: ["\$price", "\$quantity"] } }`</code>
<code>`\$divide`</code>	將第一個數字除以第二個數字	<code>`{ \$divide: [<expression1>, <expression2>] }`</code> 例如： <code>`{ ratio: { \$divide: ["\$total", "\$count"] } }`</code>
<code>`\$mod`</code>	返回第一個數字除以第二個數字的餘數	<code>`{ \$mod: [<expression1>, <expression2>] }`</code> 例如： <code>`{ remainder: { \$mod: ["\$total", 100] } }`</code>

4. 字符串操作符 (String Operators)

操作符	說明	語法和示例
<code>`\$concat`</code>	將兩個或多個字符串拼接在一起	<code>`{ \$concat: [<expression1>, <expression2>, ...] }`</code> 例如： <code>`{ fullName: { \$concat: ["\$first_name", " ", "\$last_name"] } }`</code>
<code>`\$substr`</code>	返回指定位置的子字符串	<code>`{ \$substr: [<string>, <start>, <length>] }`</code> 例如： <code>`{ shortName: { \$substr: ["\$name", 0, 3] } }`</code>
<code>`\$toLower`</code>	將字符串轉換為小寫	<code>`{ \$toLower: <string> }`</code> 例如： <code>`{ lowerCaseName: { \$toLower: "\$name" } }`</code>
<code>`\$toUpper`</code>	將字符串轉換為大寫	<code>`{ \$toUpper: <string> }`</code> 例如： <code>`{ upperCaseName: { \$toUpper: "\$name" } }`</code>
<code>`\$regexMatch`</code>	使用正則表達式匹配字符串	<code>`{ \$regexMatch: { input: "\$name", regex: "^J" } }`</code>

5. 數組操作符 (Array Operators)

操作符	說明	語法和示例
<code>`\$size`</code>	返回數組的大小 (長度)	<code>`{ \$size: <array> }`</code> 例如： <code>`{ itemCount: { \$size: "\$items" } }`</code>
<code>`\$slice`</code>	返回數組的子集	<code>`{ \$slice: [<array>, <position>, <n>] }`</code> 例如： <code>`{ topItems: { \$slice: ["\$items", 0, 3] } }`</code>
<code>`\$arrayElemAt`</code>	返回數組中指定索引位置的元素	<code>`{ \$arrayElemAt: [<array>, <idx>] }`</code> 例如： <code>`{ firstItem: { \$arrayElemAt: ["\$items", 0] } }`</code>
<code>`\$in`</code>	檢查值是否存在於數組中	<code>`{ <field>: { \$in: [<value1>, <value2>, ...] } }`</code> 例如： <code>`{ status: { \$in: ["A", "B"] } }`</code>

6. 日期操作符 (Date Operators)

操作符	說明	語法和示例
<code>`\$dateToString`</code>	將日期轉換為字符串	<code>`{ \$dateToString: { format: "%Y-%m-%d", date: "\$dateField" } }`</code>

操作符	說明	語法和示例
<code>`\$year`</code>	返回日期的年份	<code>`{ \$year: <date> }`</code> 例如： <code>`{ year: { \$year: "\$dateField" } }`</code>
<code>`\$month`</code>	返回日期的月份	<code>`{ \$month: <date> }`</code> 例如： <code>`{ month: { \$month: "\$dateField" } }`</code>
<code>`\$dayOfMonth`</code>	返回日期的日	<code>`{ \$dayOfMonth: <date> }`</code> 例如： <code>`{ day: { \$dayOfMonth: "\$dateField" } }`</code>
<code>`\$hour`</code>	返回日期的小時	<code>`{ \$hour: <date> }`</code> 例如： <code>`{ hour: { \$hour: "\$dateField" } }`</code>
<code>`\$minute`</code>	返回日期的分鐘	<code>`{ \$minute: <date> }`</code> 例如： <code>`{ minute: { \$minute: "\$dateField" } }`</code>
<code>`\$second`</code>	返回日期的秒	<code>`{ \$second: <date> }`</code> 例如： <code>`{ second: { \$second: "\$dateField" } }`</code>

7. 條件操作符 (Conditional Operators)

操作符	說明	語法和示例
<code>`\$cond`</code>	基於條件返回不同的值	<code>`{ \$cond: { if: <expression>, then: <expression>, else: <expression> } }`</code> 例如： <code>`{ \$cond: { if: { \$gt: ["\$quantity", 10] }, then: "big", else: "small" } }`</code>
<code>`\$ifNull`</code>	如果第一個參數為 null，則返回第二個參數	<code>`{ \$ifNull: [<expression>, <replacement-if-null>] }`</code> 例如： <code>`{ name: { \$ifNull: ["\$username", "unknown"] } }`</code>

8. 類型操作符 (Type Operators)

操作符	說明	語法和示例
<code>`\$type`</code>	返回字段的 BSON 類型	<code>`{ \$type: <field> }`</code> 例如： <code>`{ fieldType: { \$type: "\$age" } }`</code>
<code>`\$exists`</code>	測試字段是否存在	<code>`{ <field>: { \$exists: <boolean> } }`</code> 例如： <code>`{ age: { \$exists: true } }`</code>

9. 聚合管道階段操作符 (Pipeline Stage Operators)

操作符	說明	語法和示例
<code>`\$match`</code>	過濾文件，只傳遞符合指定條件的文件	<code>`{ \$match: { status: "A" } }`</code>
<code>`\$group`</code>	將輸入文件按指定的 <code>`_id`</code> 表達式分組	<code>`{ \$group: { _id: "\$user_id", totalAmount: { \$sum: "\$amount" } } }`</code>
<code>`\$project`</code>	改變輸出文件的結構，只保留指定的字段	<code>`{ \$project: { item: 1, totalAmount: 1 } }`</code>
<code>`\$sort`</code>	根據一個或多個字段對輸出文件進行排序	<code>`{ \$sort: { totalAmount: -1 } }`</code>

操作符	說明	語法和示例
<code>`\$limit`</code>	限制傳遞到下一階段的文件數量	<code>`{ \$limit: 5 }`</code>
<code>`\$skip`</code>	跳過指定數量的文件，傳遞剩餘的文件	<code>`{ \$skip: 10 }`</code>
<code>`\$lookup`</code>	執行左外連接 (left outer join) 以從另一個集合中引入文件	<code>`{ \$lookup: { from: "orders", localField: "user_id", foreignField: "user_id", as: "orders" } }`</code>
<code>`\$unwind`</code>	將每個輸入文件中的某個數組字段展開為多個文件	<code>`{ \$unwind: "\$items" }`</code>
<code>`\$addFields`</code>	添加新字段或重命名現有字段	<code>`{ \$addFields: { totalWithTax: { \$multiply: ["\$total", 1.1] } } }`</code>
<code>`\$out`</code>	將聚合管道的結果寫入指定的集合	<code>`{ \$out: "aggregated_results" }`</code>
<code>`\$merge`</code>	將管道的結果合併到現有集合中	<code>`{ \$merge: { into: "myCollection", on: "_id", whenMatched: "merge", whenNotMatched: "insert" } }`</code>

10. 其他操作符

操作符	說明	語法和示例
<code>`\$geoNear`</code>	返回按距離排序的文件	<code>`{ \$geoNear: { near: { type: "Point", coordinates: [<lng>, <lat>] }, distanceField: "dist.calculated", spherical: true } }`</code>
<code>`\$text`</code>	執行文本搜索	<code>`{ \$text: { \$search: "keyword" } }`</code>
<code>`\$redact`</code>	基於文檔級安全性來限制返回的字段	<code>`{ \$redact: { \$cond: { if: <condition>, then: <action>, else: <action> } } }`</code>

這些操作符涵蓋了 MongoDB 聚合框架中的所有常用操作符，並提供了簡要的語法和示例說明。你可以根據具體需求使用這些操作符來構建複雜的數據查詢和分析邏輯。

