

SQL Example	MongoDB Example	Description
SELECT COUNT(*) AS count FROM orders	db.orders.aggregate([{ \$group: { _id: null, count: { \$sum: 1 } } }])	Count all records from orders
SELECT SUM(price) AS total FROM orders	db.orders.aggregate([{ \$group: { _id: null, total: { \$sum: "\$price" } } }])	Sum the price field from orders
SELECT cust_id, SUM(price) AS total FROM orders GROUP BY cust_id	db.orders.aggregate([{ \$group: { _id: "\$cust_id", total: { \$sum: "\$price" } } }])	For each unique cust_id, sum the price field.
SELECT cust_id, SUM(price) AS total FROM orders GROUP BY cust_id ORDER BY total	db.orders.aggregate([{ \$group: { _id: "\$cust_id", total: { \$sum: "\$price" } } }, { \$sort: { total: 1 } }])	For each unique cust_id, sum the price field, results sorted by sum.
SELECT cust_id, ord_date, SUM(price) AS total FROM orders GROUP BY cust_id, ord_date	db.orders.aggregate([{ \$group: { _id: { cust_id: "\$cust_id", ord_date: "\$ord_date" }, total: { \$sum: "\$price" } } }])	For each unique cust_id, ord_date grouping, sum the price field.
SELECT cust_id, count(*) FROM orders GROUP BY cust_id HAVING count(*) > 1	db.orders.aggregate([{ \$group: { _id: "\$cust_id", count: { \$sum: 1 } } }, { \$match: { count: { \$gt: 1 } } }])	For cust_id with multiple records, return the cust_id and the corresponding record count.
SELECT cust_id, ord_date, SUM(price) AS total FROM orders GROUP BY cust_id, ord_date HAVING total > 250	db.orders.aggregate([{ \$group: { _id: { cust_id: "\$cust_id", ord_date: "\$ord_date" }, total: { \$sum: "\$price" } } }, { \$match: { total: { \$gt: 250 } } }])	For each unique cust_id, ord_date grouping, sum the price field and return only where the sum is greater than 250.
SELECT cust_id, SUM(price) as total FROM orders WHERE status = 'A' GROUP BY cust_id	db.orders.aggregate([{ \$match: { status: 'A' } }, { \$group: { _id: "\$cust_id", total: { \$sum: "\$price" } } }])	For each unique cust_id with status A, sum the price field.
SELECT cust_id, SUM(price) as total FROM orders WHERE status = 'A' GROUP BY cust_id HAVING total > 250	db.orders.aggregate([{ \$match: { status: 'A' } }, { \$group: { _id: "\$cust_id", total: { \$sum: "\$price" } } }, { \$match: { total: { \$gt: 250 } } }])	For each unique cust_id with status A, sum the price field and return only where the sum is greater than 250.
6.4. Aggregation Reference		311
SELECT cust_id,	db.orders.aggregate([For each unique cust_id, sum the