# 

# Prácticas Avanzadas de MongoDB – Agregación

## 🧪 PRÁCTICA 1 – Tienda Online: Análisis de Ventas

### 📦 InsertMany:

db.ventas.insertMany([  
 { producto: "Auriculares Bluetooth", categoria: "Electrónica", cliente: { nombre: "Ana", pais: "Argentina" }, cantidad: 2, precioUnitario: 5000, fecha: ISODate("2024-01-15") },  
 { producto: "Teclado Mecánico", categoria: "Electrónica", cliente: { nombre: "Luis", pais: "Uruguay" }, cantidad: 1, precioUnitario: 8000, fecha: ISODate("2024-02-10") },  
 { producto: "Zapatillas Deportivas", categoria: "Indumentaria", cliente: { nombre: "Lucía", pais: "Argentina" }, cantidad: 3, precioUnitario: 10000, fecha: ISODate("2023-12-01") },  
 { producto: "Remera", categoria: "Indumentaria", cliente: { nombre: "Marcos", pais: "Chile" }, cantidad: 2, precioUnitario: 4000, fecha: ISODate("2023-11-22") },  
 { producto: "Notebook", categoria: "Tecnología", cliente: { nombre: "Sofía", pais: "Argentina" }, cantidad: 1, precioUnitario: 250000, fecha: ISODate("2024-03-20") },  
 { producto: "Mouse", categoria: "Electrónica", cliente: { nombre: "Pedro", pais: "Brasil" }, cantidad: 4, precioUnitario: 3000, fecha: ISODate("2024-04-05") },  
 { producto: "Smartwatch", categoria: "Tecnología", cliente: { nombre: "Julieta", pais: "Chile" }, cantidad: 1, precioUnitario: 40000, fecha: ISODate("2023-09-12") },  
 { producto: "Campera", categoria: "Indumentaria", cliente: { nombre: "Ricardo", pais: "Uruguay" }, cantidad: 1, precioUnitario: 18000, fecha: ISODate("2024-03-05") },  
 { producto: "Tablet", categoria: "Tecnología", cliente: { nombre: "Elena", pais: "Argentina" }, cantidad: 2, precioUnitario: 120000, fecha: ISODate("2024-01-28") },  
 { producto: "Cargador Portátil", categoria: "Electrónica", cliente: { nombre: "Martín", pais: "Chile" }, cantidad: 5, precioUnitario: 3500, fecha: ISODate("2024-02-17") }  
]);

### 🧠 Preguntas:

1. Mostrar todas las ventas realizadas en 2024.

2. Calcular el total vendido por país (cantidad \* precioUnitario).

3. Agrupar por categoría y calcular el total de productos vendidos.

4. Promedio del valor total por venta.

5. Mostrar el top 3 de productos más vendidos por cantidad.

6. Ordenar las ventas por fecha descendente.

7. Mostrar las ventas realizadas en Argentina mayores a $10.000.

8. Calcular el total recaudado por categoría.

9. Mostrar los productos que se vendieron en más de un país.

10. Contar cuántas ventas hubo por cliente.

### ✅ Respuestas esperadas (MongoDB Shell):

// 1  
db.ventas.aggregate([{ $match: { fecha: { $gte: ISODate("2024-01-01") } } }]);

// 2  
db.ventas.aggregate([ { $project: { pais: "$cliente.pais", total: { $multiply: ["$cantidad", "$precioUnitario"] } } }, { $group: { \_id: "$pais", totalPais: { $sum: "$total" } } } ]);

// 3  
db.ventas.aggregate([ { $group: { \_id: "$categoria", totalProductos: { $sum: "$cantidad" } } } ]);

// 4  
db.ventas.aggregate([ { $project: { totalVenta: { $multiply: ["$cantidad", "$precioUnitario"] } } }, { $group: { \_id: null, promedio: { $avg: "$totalVenta" } } } ]);

// 5  
db.ventas.aggregate([ { $group: { \_id: "$producto", cantidadTotal: { $sum: "$cantidad" } } }, { $sort: { cantidadTotal: -1 } }, { $limit: 3 } ]);

// 6  
db.ventas.aggregate([{ $sort: { fecha: -1 } }]);

// 7  
db.ventas.aggregate([ { $match: { "cliente.pais": "Argentina" } }, { $project: { producto: 1, total: { $multiply: ["$cantidad", "$precioUnitario"] } } }, { $match: { total: { $gt: 10000 } } } ]);

// 8  
db.ventas.aggregate([ { $project: { categoria: 1, total: { $multiply: ["$cantidad", "$precioUnitario"] } } }, { $group: { \_id: "$categoria", totalRecaudado: { $sum: "$total" } } } ]);

// 9  
db.ventas.aggregate([ { $group: { \_id: { producto: "$producto", pais: "$cliente.pais" } } }, { $group: { \_id: "$\_id.producto", paises: { $addToSet: "$\_id.pais" } } }, { $match: { "paises.1": { $exists: true } } } ]);

// 10  
db.ventas.aggregate([ { $group: { \_id: "$cliente.nombre", cantidadVentas: { $sum: 1 } } } ]);

## 🧪 PRÁCTICA 2 – Plataforma de Cursos Online

### 📦 InsertMany:

db.cursos.insertMany([  
 {  
 titulo: "MongoDB Avanzado",  
 duracionHoras: 20,  
 categorias: ["Bases de Datos", "Backend"],  
 docente: "Juan Pérez",  
 estudiantes: [  
 { nombre: "Lucía", nota: 8 },  
 { nombre: "Martín", nota: 9 },  
 { nombre: "Sofía", nota: 7 }  
 ]  
 },  
 {  
 titulo: "Python para Principiantes",  
 duracionHoras: 15,  
 categorias: ["Programación", "Backend"],  
 docente: "Ana Torres",  
 estudiantes: [  
 { nombre: "Carlos", nota: 6 },  
 { nombre: "Julieta", nota: 8 }  
 ]  
 },  
 {  
 titulo: "Diseño UX/UI",  
 duracionHoras: 18,  
 categorias: ["Diseño", "Frontend"],  
 docente: "Diego López",  
 estudiantes: [  
 { nombre: "Lucía", nota: 9 },  
 { nombre: "Matías", nota: 10 }  
 ]  
 },  
 {  
 titulo: "JavaScript Intermedio",  
 duracionHoras: 22,  
 categorias: ["Programación", "Frontend"],  
 docente: "Ana Torres",  
 estudiantes: [  
 { nombre: "Lucía", nota: 7 },  
 { nombre: "Carlos", nota: 7 },  
 { nombre: "Elena", nota: 8 }  
 ]  
 }  
]);

### 🧠 Preguntas:

1. Mostrar los cursos con más de 15 horas de duración.

2. Descomponer estudiantes y mostrar su nombre, nota y curso.

3. Calcular el promedio de nota por docente.

4. Contar cuántos cursos tiene cada categoría.

5. Mostrar los cursos con más de 2 estudiantes, incluyendo su cantidad.

6. Ordenar los cursos por duración descendente.

7. Mostrar los cursos que contienen la categoría 'Backend'.

8. Obtener la cantidad total de estudiantes por curso.

9. Listar todos los estudiantes únicos (sin repetir) que cursaron algo.

10. Agrupar cursos por docente y mostrar duración total impartida por cada uno.

### ✅ Respuestas esperadas (MongoDB Shell):

// 1  
db.cursos.aggregate([{ $match: { duracionHoras: { $gt: 15 } } }]);

// 2  
db.cursos.aggregate([ { $unwind: "$estudiantes" }, { $project: { curso: "$titulo", estudiante: "$estudiantes.nombre", nota: "$estudiantes.nota", \_id: 0 } } ]);

// 3  
db.cursos.aggregate([ { $unwind: "$estudiantes" }, { $group: { \_id: "$docente", promedioNota: { $avg: "$estudiantes.nota" } } } ]);

// 4  
db.cursos.aggregate([ { $unwind: "$categorias" }, { $sortByCount: "$categorias" } ]);

// 5  
db.cursos.aggregate([ { $project: { titulo: 1, cantidadEstudiantes: { $size: "$estudiantes" } } }, { $match: { cantidadEstudiantes: { $gt: 2 } } } ]);

// 6  
db.cursos.aggregate([ { $sort: { duracionHoras: -1 } } ]);

// 7  
db.cursos.aggregate([ { $match: { categorias: "Backend" } } ]);

// 8  
db.cursos.aggregate([ { $project: { titulo: 1, totalEstudiantes: { $size: "$estudiantes" } } } ]);

// 9  
db.cursos.aggregate([ { $unwind: "$estudiantes" }, { $group: { \_id: "$estudiantes.nombre" } } ]);

// 10  
db.cursos.aggregate([ { $group: { \_id: "$docente", horasTotales: { $sum: "$duracionHoras" } } } ]);

## 🧪 PRÁCTICA 3 – Aplicación de Seguimiento de Hábitos

### 📦 InsertMany:

db.habitos.insertMany([  
 {  
 usuario: "Carlos",  
 habito: "Caminar",  
 frecuencia: "Diaria",  
 registros: [  
 { fecha: ISODate("2024-04-01"), duracionMin: 30 },  
 { fecha: ISODate("2024-04-02"), duracionMin: 25 }  
 ]  
 },  
 {  
 usuario: "Ana",  
 habito: "Meditación",  
 frecuencia: "Semanal",  
 registros: [  
 { fecha: ISODate("2024-04-03"), duracionMin: 15 },  
 { fecha: ISODate("2024-04-10"), duracionMin: 20 }  
 ]  
 },  
 {  
 usuario: "Martín",  
 habito: "Leer",  
 frecuencia: "Diaria",  
 registros: [  
 { fecha: ISODate("2024-04-01"), duracionMin: 40 },  
 { fecha: ISODate("2024-04-02"), duracionMin: 35 },  
 { fecha: ISODate("2024-04-03"), duracionMin: 50 }  
 ]  
 },  
 {  
 usuario: "Lucía",  
 habito: "Correr",  
 frecuencia: "Semanal",  
 registros: [  
 { fecha: ISODate("2024-04-02"), duracionMin: 45 }  
 ]  
 }  
]);

### 🧠 Preguntas:

1. Mostrar hábitos registrados en abril 2024.

2. Descomponer registros y mostrar usuario, hábito, fecha y duración.

3. Calcular el promedio de minutos dedicados por hábito.

4. Agrupar por frecuencia y contar cuántos usuarios hay por tipo.

5. Mostrar los 3 hábitos con mayor tiempo total registrado.

6. Listar los hábitos cuya duración promedio supere los 30 minutos.

7. Calcular el total de registros por usuario.

8. Mostrar todos los registros del 2 de abril.

9. Agrupar por usuario y listar sus hábitos únicos.

10. Ordenar hábitos por cantidad total de minutos, descendente.

### ✅ Respuestas esperadas (MongoDB Shell):

// 1  
db.habitos.find({ "registros.fecha": { $gte: ISODate("2024-04-01"), $lt: ISODate("2024-05-01") } });

// 2  
db.habitos.aggregate([ { $unwind: "$registros" }, { $project: { usuario: 1, habito: 1, fecha: "$registros.fecha", duracionMin: "$registros.duracionMin" } } ]);

// 3  
db.habitos.aggregate([ { $unwind: "$registros" }, { $group: { \_id: "$habito", promedioMin: { $avg: "$registros.duracionMin" } } } ]);

// 4  
db.habitos.aggregate([ { $group: { \_id: "$frecuencia", usuarios: { $addToSet: "$usuario" } } }, { $project: { frecuencia: "$\_id", cantidad: { $size: "$usuarios" }, \_id: 0 } } ]);

// 5  
db.habitos.aggregate([ { $unwind: "$registros" }, { $group: { \_id: "$habito", totalTiempo: { $sum: "$registros.duracionMin" } } }, { $sort: { totalTiempo: -1 } }, { $limit: 3 } ]);

// 6  
db.habitos.aggregate([ { $unwind: "$registros" }, { $group: { \_id: "$habito", promedio: { $avg: "$registros.duracionMin" } } }, { $match: { promedio: { $gt: 30 } } } ]);

// 7  
db.habitos.aggregate([ { $project: { usuario: 1, totalRegistros: { $size: "$registros" } } } ]);

// 8  
db.habitos.aggregate([ { $unwind: "$registros" }, { $match: { "registros.fecha": ISODate("2024-04-02") } }, { $project: { usuario: 1, habito: 1, duracion: "$registros.duracionMin" } } ]);

// 9  
db.habitos.aggregate([ { $group: { \_id: "$usuario", habitos: { $addToSet: "$habito" } } } ]);

// 10  
db.habitos.aggregate([ { $unwind: "$registros" }, { $group: { \_id: "$habito", minutosTotales: { $sum: "$registros.duracionMin" } } }, { $sort: { minutosTotales: -1 } } ]);