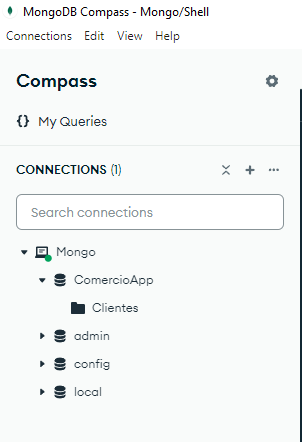
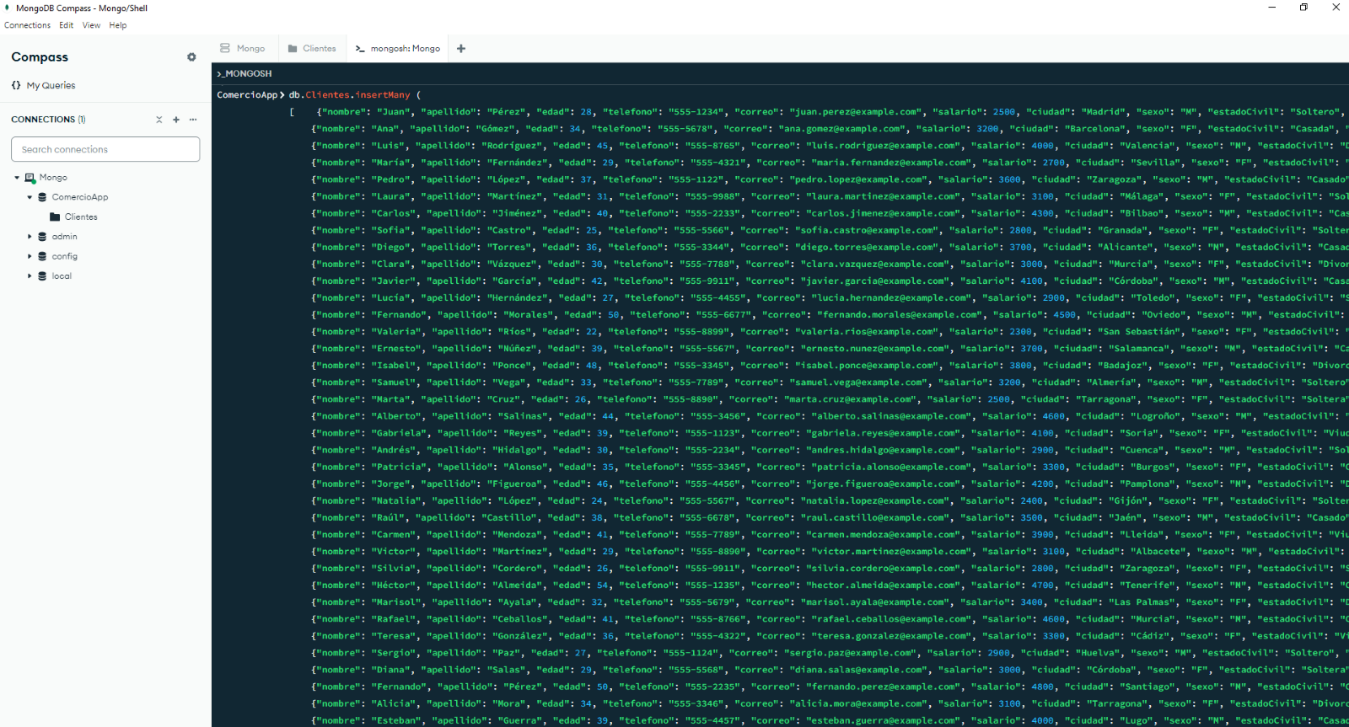
**Fase 1: MongoDB**

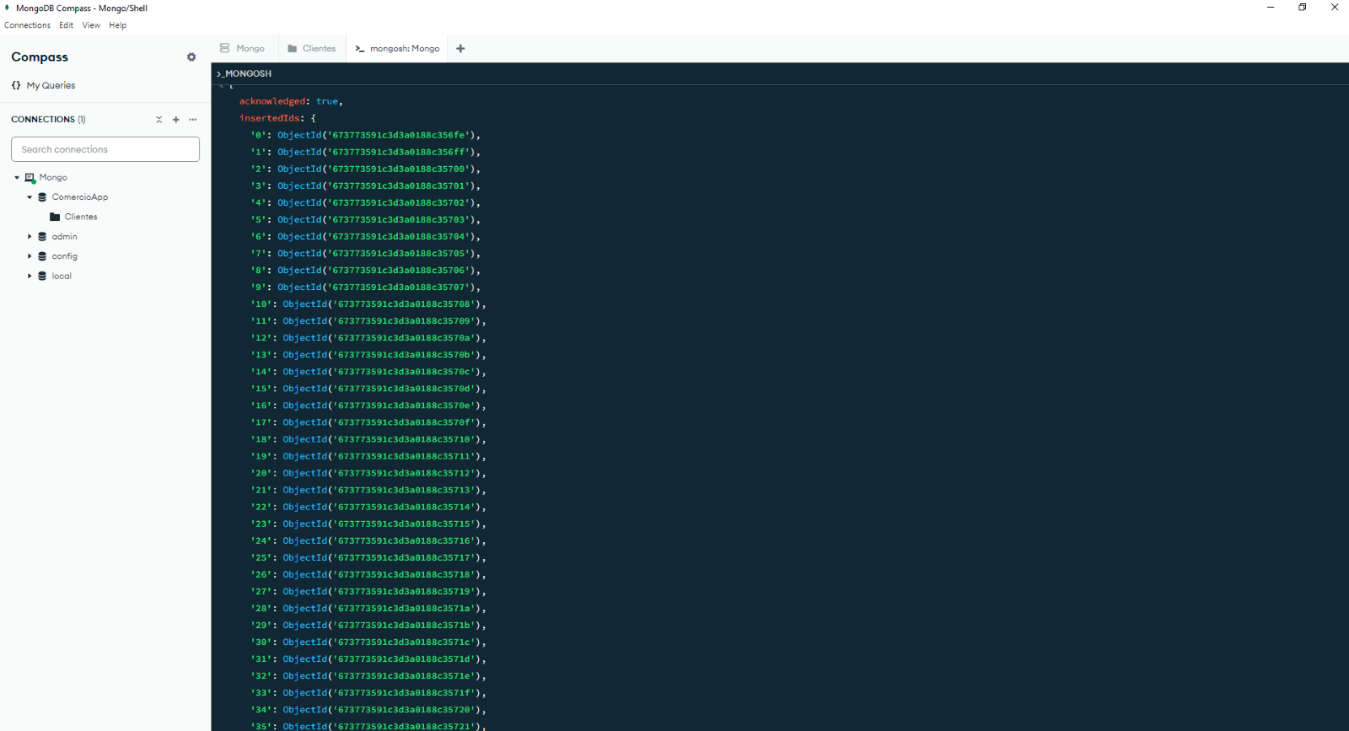
Creación base de datos



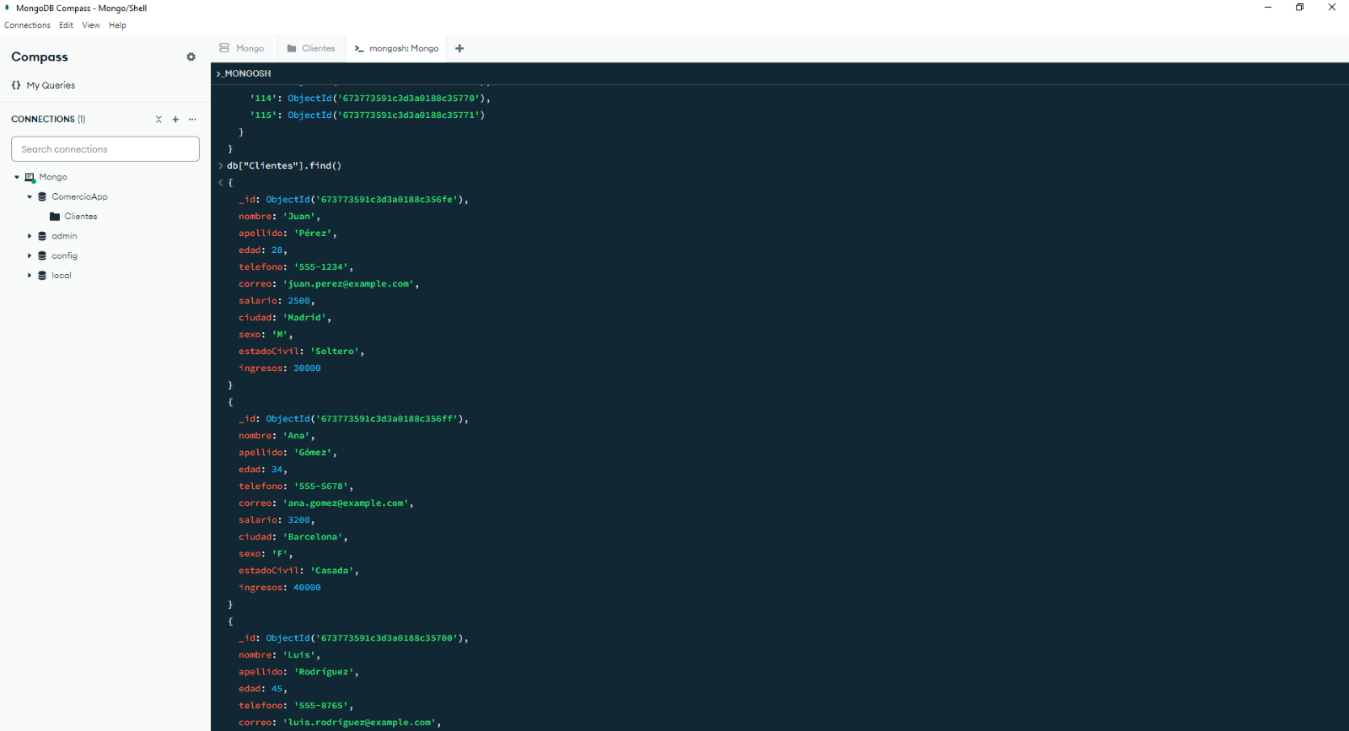
Colección de información a insertar en tabla **Clientes**



Se visualiza que se insertaron los registros



Se consultan los registros insertados previamente



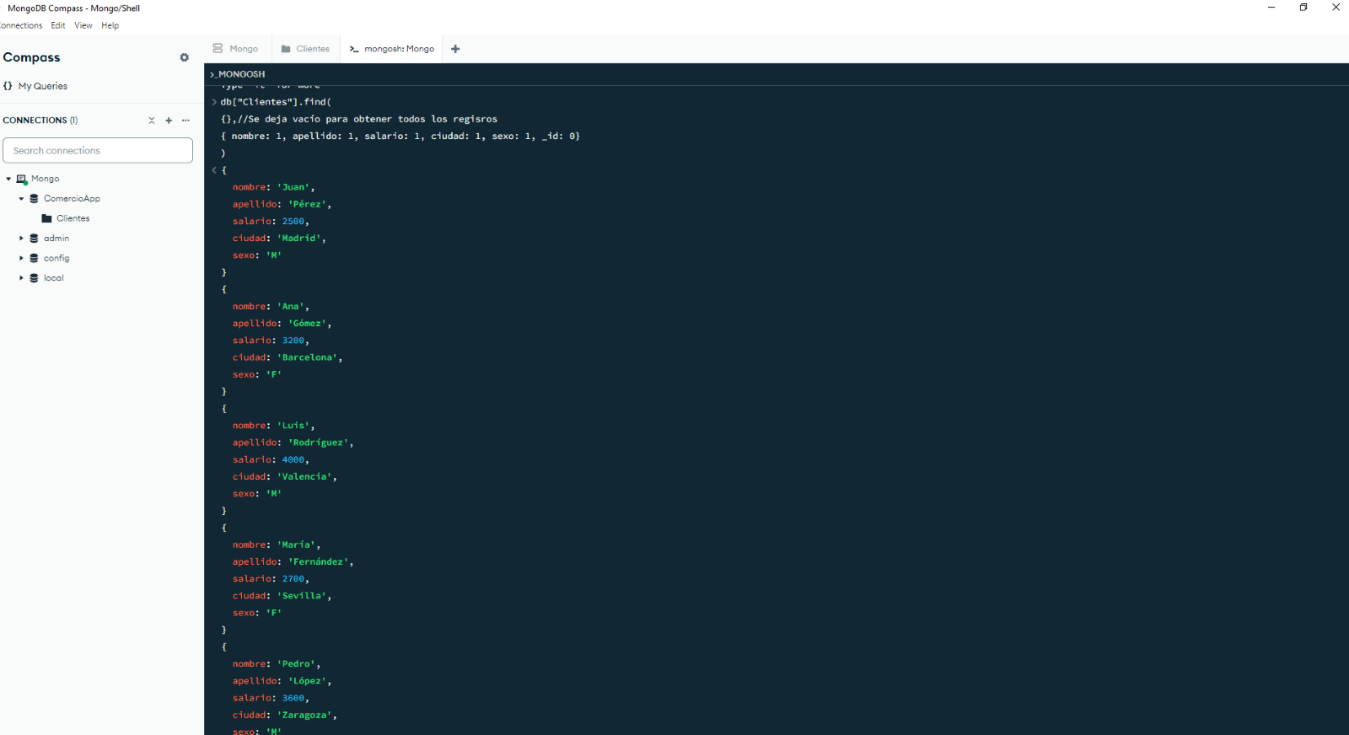
Filtrar ciertas columnas de los registros que se insertaron anteriormente

db["Clientes"].find(

{},//Se deja vacío para obtener todos los regisros

{ nombre: 1, apellido: 1, salario: 1, ciudad: 1, sexo: 1, \_id: 0}

)

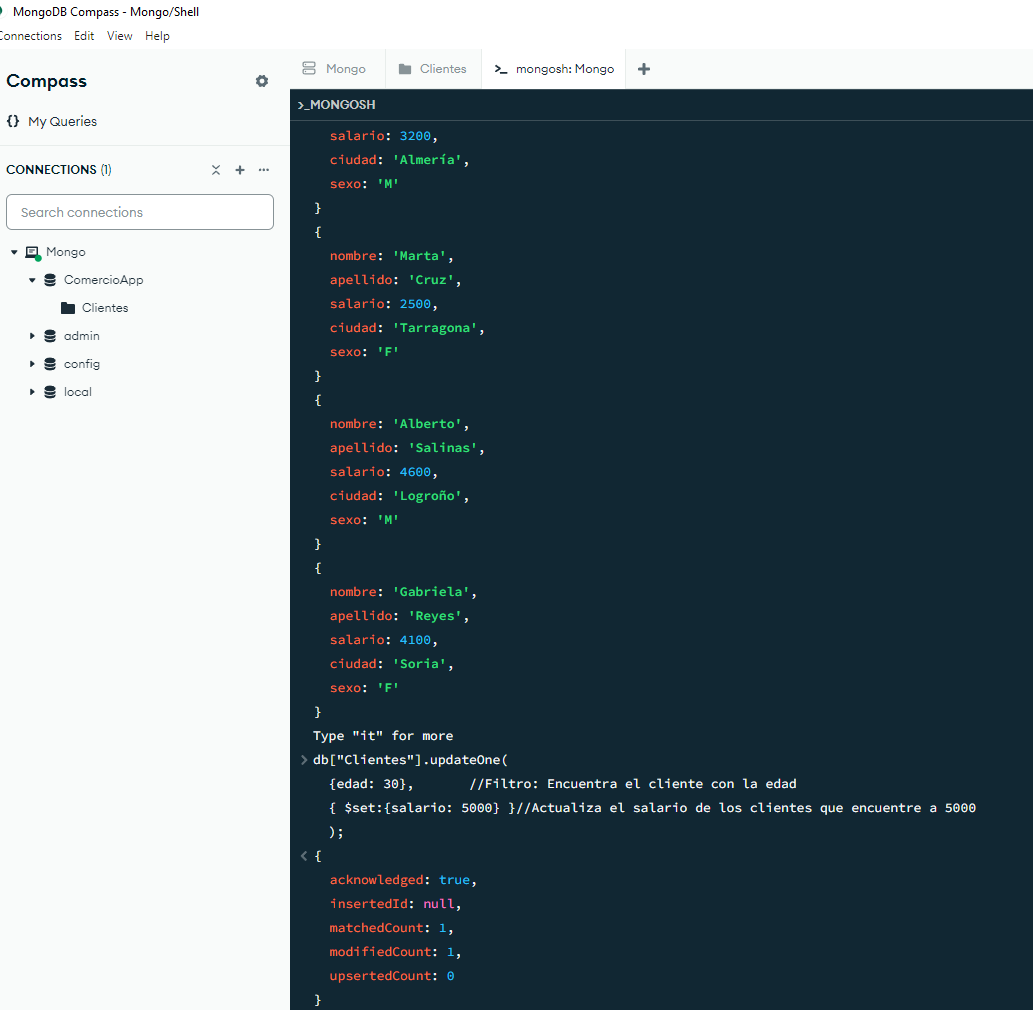


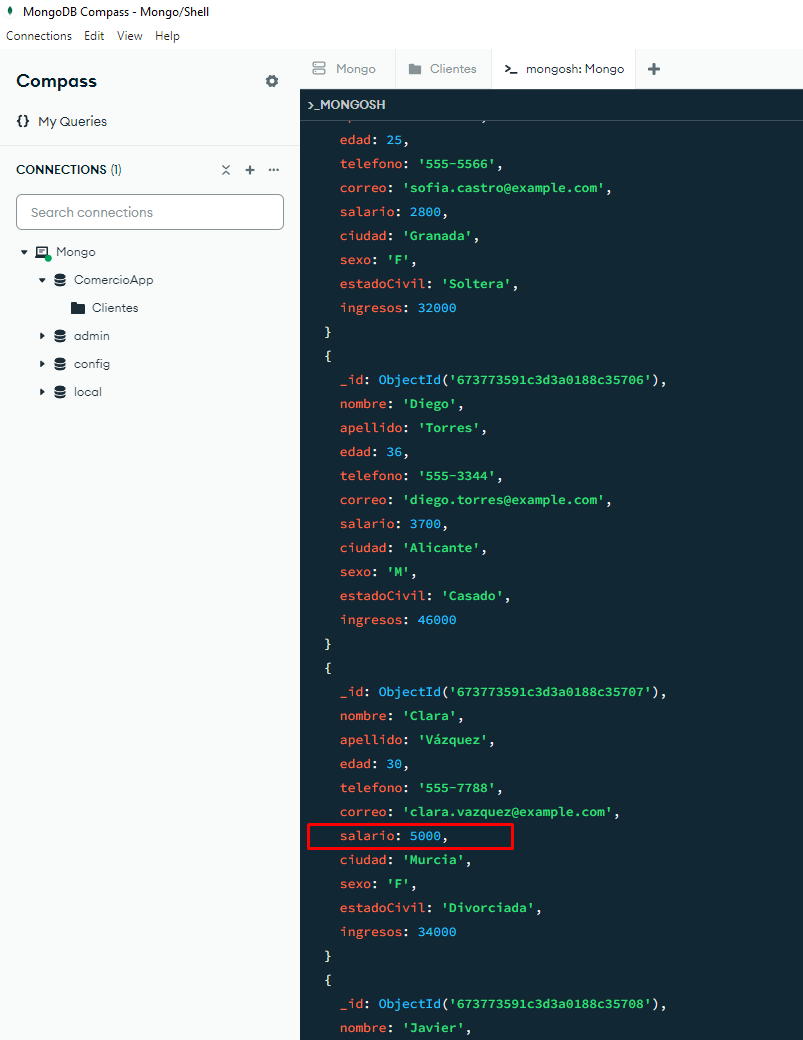
Actualiza un registro que cumpla la siguiente condición

db["Clientes"].updateOne(

{edad: 30}, //Filtro: Encuentra el cliente con la edad

{ $set:{salario: 5000} }//Actualiza el salario del cliente que encuentre a 5000

); 



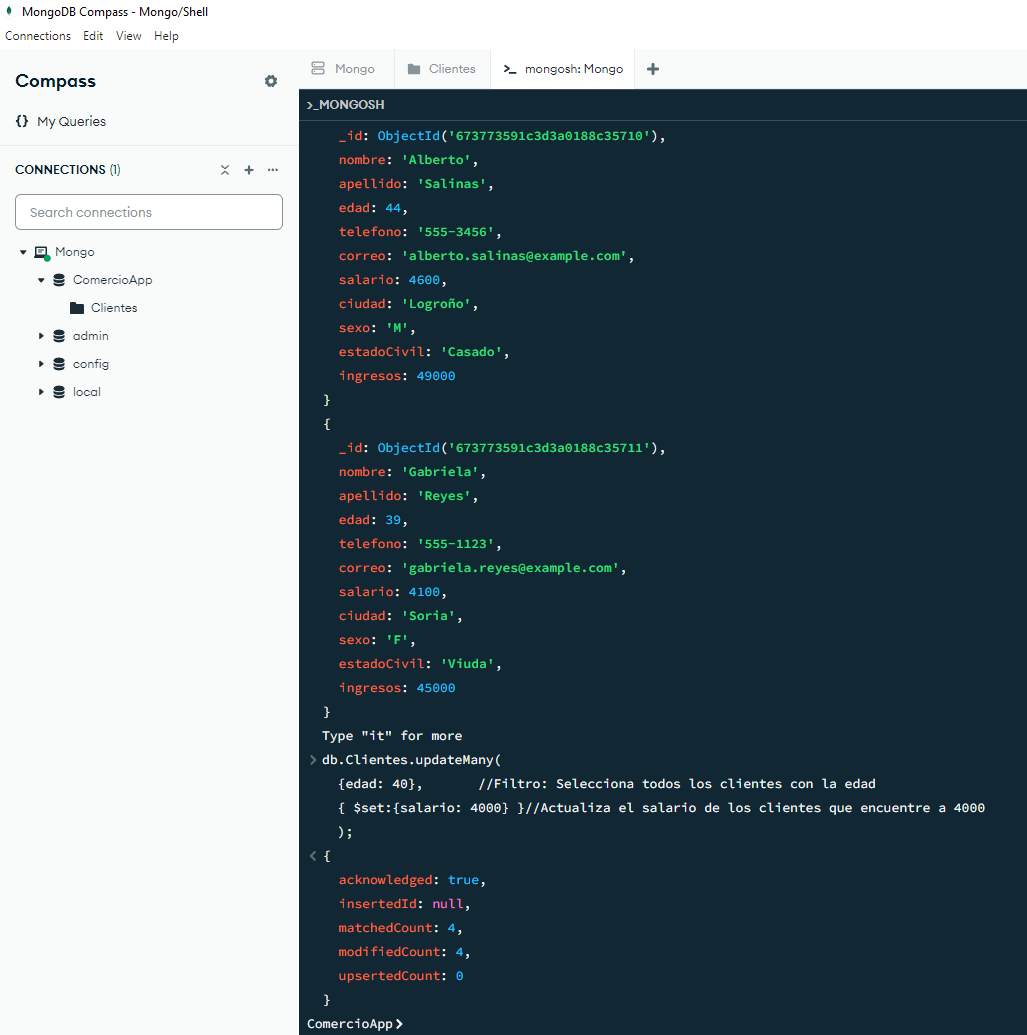
Se aplica otro filtro para que actualice todos los clientes que apliquen la condición

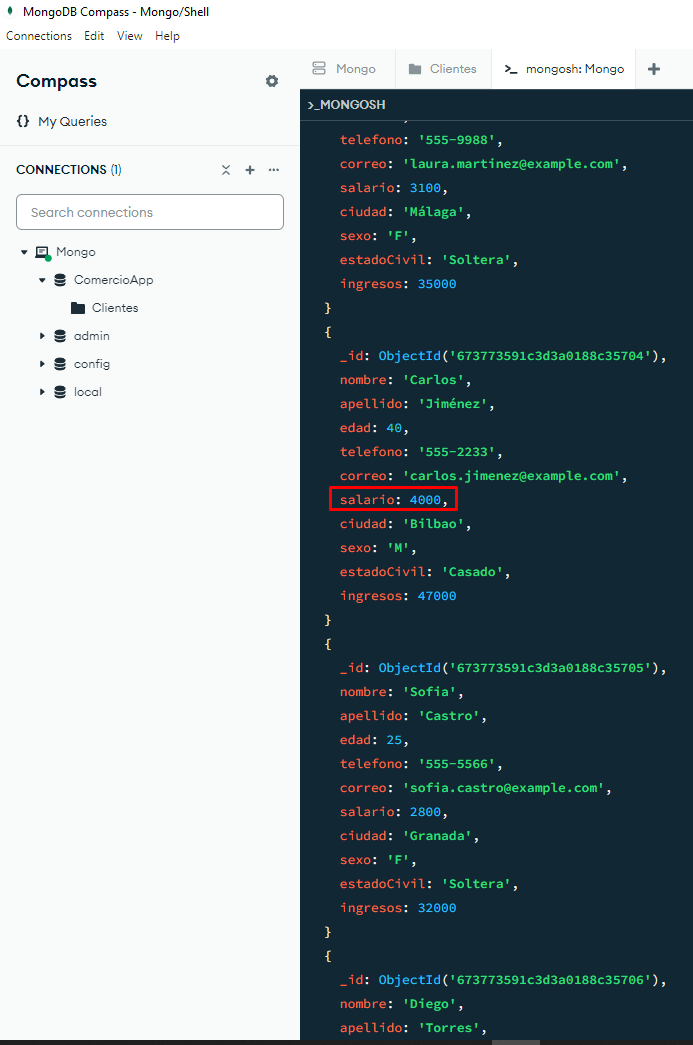
db.Clientes.updateMany(

{edad: 40}, //Filtro: Selecciona todos los clientes con la edad

{ $set:{salario: 4000} }//Actualiza el salario de los clientes que encuentre a 4000

);



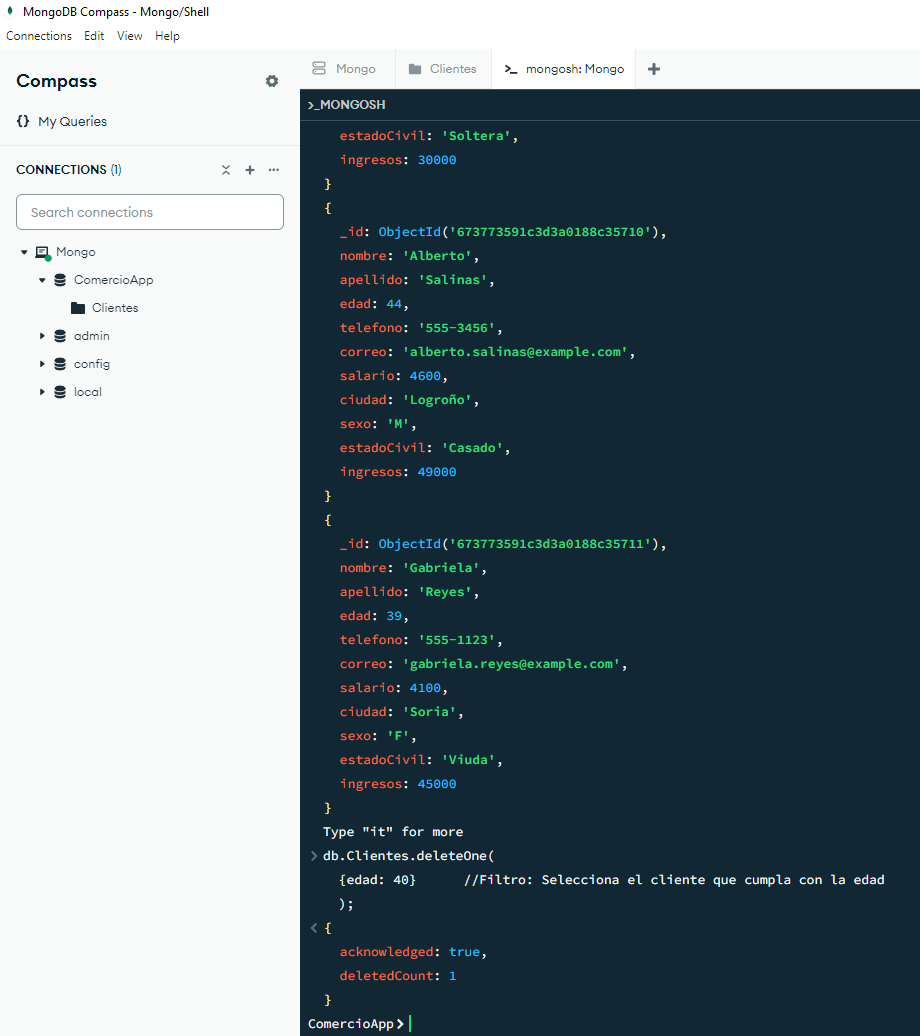


Se elimina un registro que aplique la condición de edad

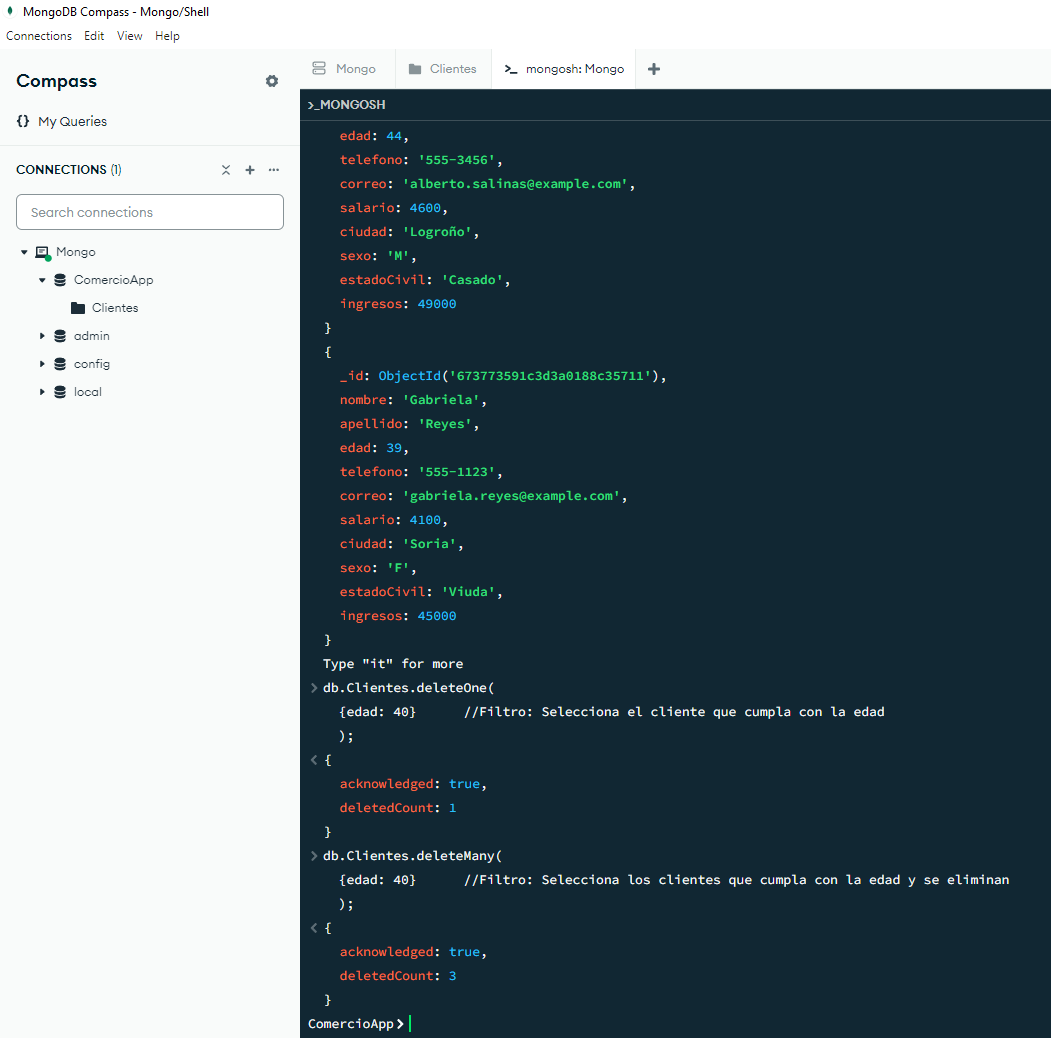
db.Clientes.deleteOne(

{edad: 40} //Filtro: Selecciona el cliente que cumpla con la edad

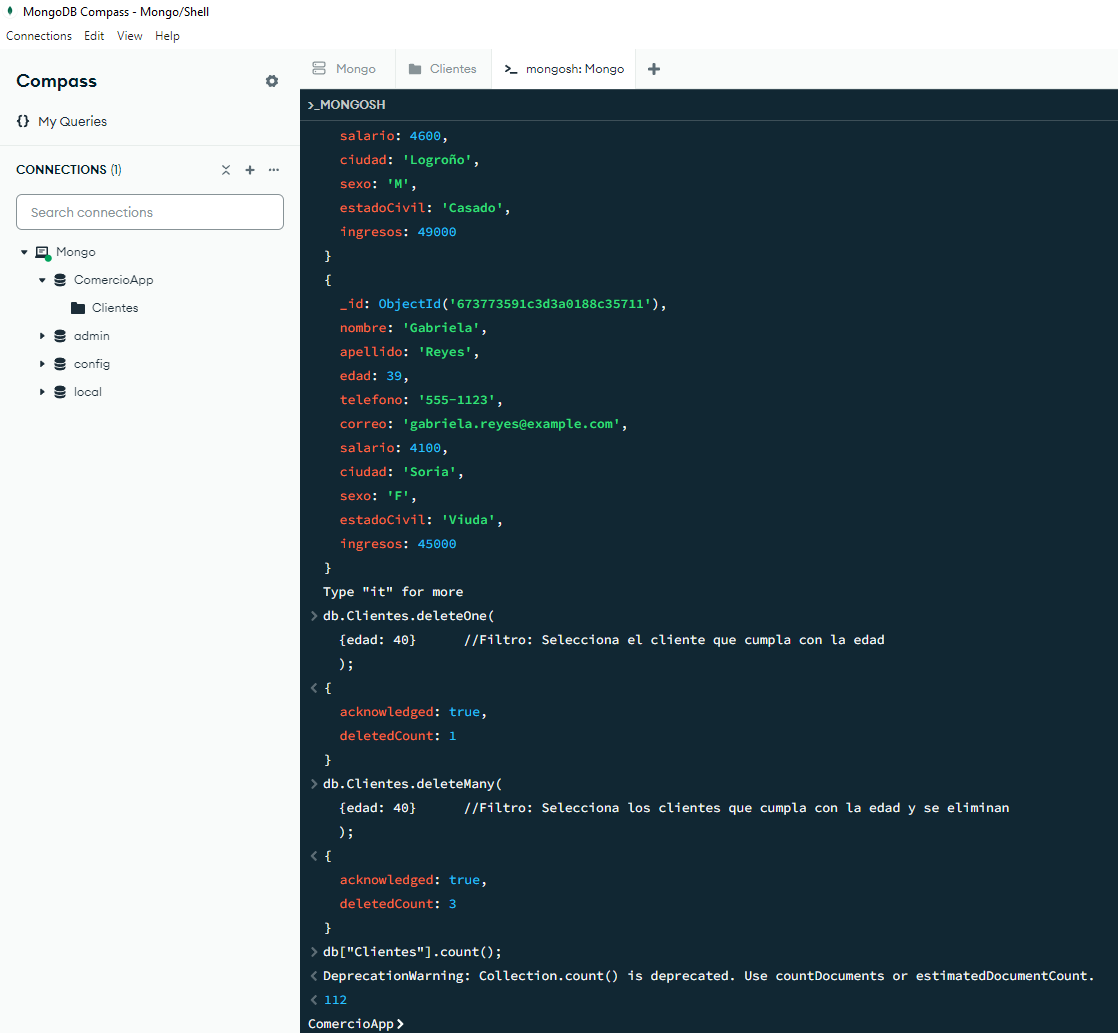
);



Se eliminan todos los registros que apliquen la condición



Se cuentan los registros, inicialmente habían 116, al borrar 4 quedan actualmente 112



Agrupación por ciudad

db.Clientes.aggregate ([

{

$match: { edad: { $gte: 40 } }// Selecciona clientes donde su edad sea mayor o igual a 40

},

{

$project: {//Calcula el total de ingresos para cada ciudad

ciudad: 1,

totalIngreso: { $multiply: [ 1, "$ingresos" ] }

}

},

{

$group: {//Agrupa los resultados por ciudad

\_id: "$ciudad",

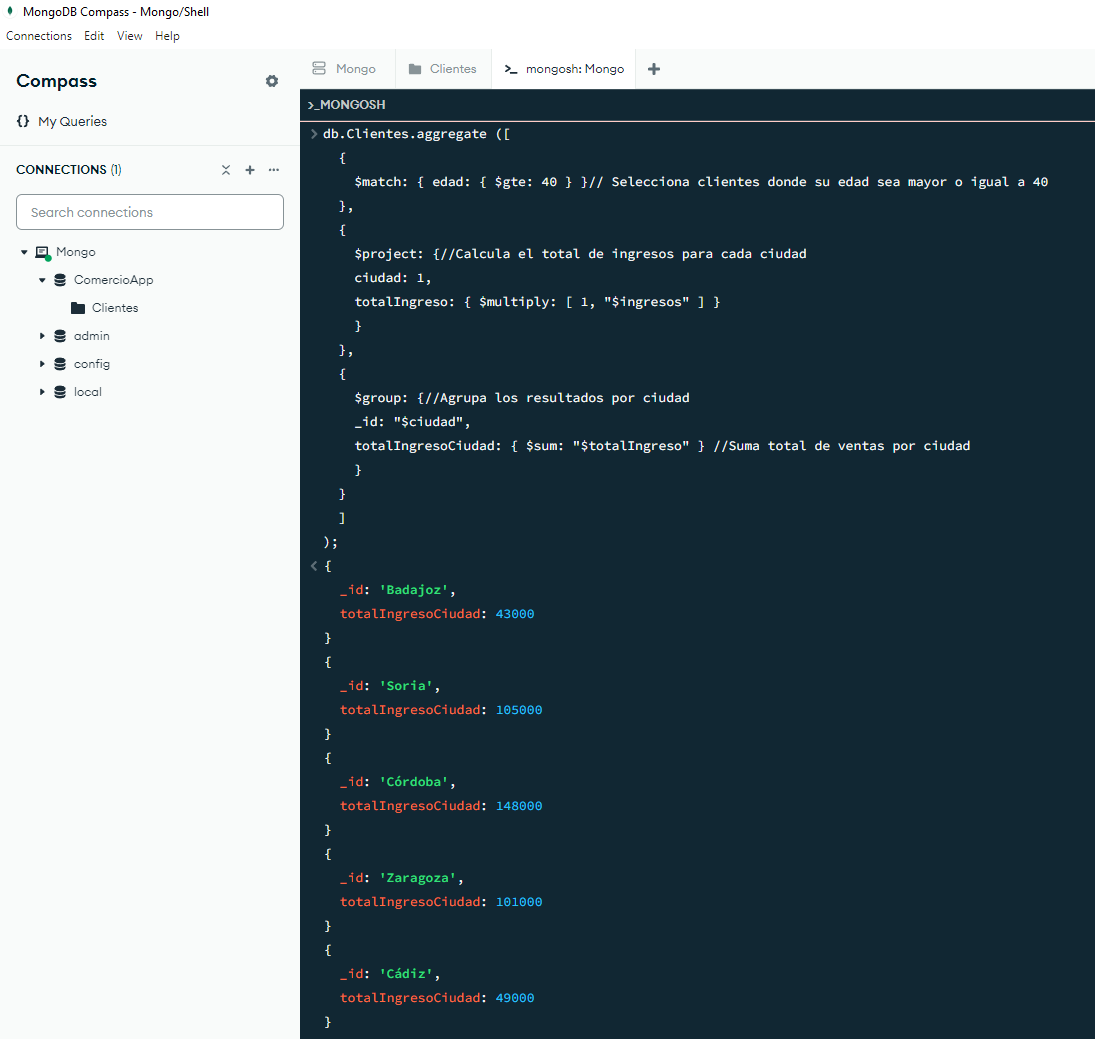
totalIngresoCiudad: { $sum: "$totalIngreso" } //Suma total de ventas por ciudad

}

}

]

);



Filtrar por edad menor o igual a 40

db.Clientes.find ( {edad: {$lte: 40} } );

