|  |  |
| --- | --- |
| 1) Devuelve el listado de clientes indicando el nombre del cliente y cuántos pedidos ha realizado. Tenga en cuenta que pueden existir clientes que no han realizado ningún pedido. | 2 |
| SELECT c.nombrecliente, COUNT(p.codigocliente) AS 'Cantidad de pedidos realizados' FROM cliente c LEFT JOIN pedido p ON c.idcliente = p.idcliente GROUP BY c.nombrecliente  db.cliente.aggregate([  {  $lookup: {  from: "pedido",  localField: "codigocliente",  foreignField: "idcliente",  as: "pedidos"  }  },  {  $unwind: {  path: "$pedidos",  preserveNullAndEmptyArrays: true  }  },  {  $group: {  \_id: "$nombrecliente",  cantidadPedidos: { $sum: 1 }  }  },  {  $project: {  \_id: 1,  nombrecliente: "$id",  cantidadPedidos: 1  }  } ]); | SELECT o.ciudad, COUNT(e.codigooficina) AS 'Cantidad de empleados'  FROM oficina o  INNER JOIN empleado e on e.idoficina=o.idoficina  GROUP BY o.ciudad  db.oficina.aggregate([  {  $lookup: {  from: "empleado",  localField: "idoficina",  foreignField: "idoficina",  as: "empleados"  }  },  {  $unwind: "$empleados"  },  {  $group: {  \_id: "$ciudad",  cantidadEmpleados: { $sum: 1 }  }  },  {  $project: {  \_id: 0,  ciudad: "$\_id",  cantidadEmpleados: 1  }  }  ]); |

|  |  |
| --- | --- |
| 3.Devuelve el listado de clientes donde aparezca el nombre del cliente, el nombre y primer apellido de su representante de ventas y la ciudad donde está su oficina. | 4.Devuelve el nombre de los clientes que hayan hecho pedidos en 2008 ordenados alfabéticamente de menor a mayor. |
| SELECT c.nombrecliente, e.nombre, e.apellido1, o.ciudad  FROM cliente c  INNER JOIN empleado e ON c. idcodigoempleadoventas= e.idempleado  INNER JOIN oficina o ON o.idoficina = e.idoficina  db.cliente.aggregate([  {  $lookup: {  from: "empleado",  localField: "idcodigoempleadoventas",  foreignField: "idempleado",  as: "empleado\_info"  }  },  {  $unwind: "$empleado\_info"  },  {  $lookup: {  from: "oficina",  localField: "empleado\_info.idoficina",  foreignField: "idoficina",  as: "oficina\_info"  }  },  {  $unwind: "$oficina\_info"  },  {  $project: {  \_id: 0,  nombrecliente: "$nombrecliente",  nombre: "$empleado\_info.nombre",  apellido1: "$empleado\_info.apellido1",  ciudad: "$oficina\_info.ciudad"  }  }  ]); | SELECT DISTINCT c.nombrecliente  FROM cliente c  INNER JOIN pedido p on p.idcliente=c.idcliente  WHERE YEAR(p.fechapedido) = 2008  ORDER BY c.nombrecliente ASC  db.cliente.aggregate([  {  $lookup: {  from: "pedido",  localField: "idcliente",  foreignField: "idcliente",  as: "pedidos"  }  },  {  $unwind: "$pedidos"  },  {  $match: {  "pedidos.fechapedido": {  $gte: new Date("2008-01-01T00:00:00Z"),  $lt: new Date("2009-01-01T00:00:00Z")  }  }  },  {  $group: {  \_id: "$nombrecliente"  }  },  {  $sort: {  \_id: 1  }  },  {  $project: {  \_id: 0,  nombrecliente: "$\_id"  }  }  ]); |
| **Subconsultas con EXISTS y NOT EXISTS** | |
| 5. SELECT \*  FROM cliente c  WHERE NOT EXISTS (SELECT \*  FROM pago p  WHERE c.idcliente = p.idcliente)  db.cliente.aggregate([  {  $lookup: {  from: "pago",  localField: "idcliente",  foreignField: "idcliente",  as: "pagos"  }  },  {  $match: {  "pagos": { $eq: [] }  }  },  {  $project: {  \_id: 0,  idcliente: 1,  nombrecliente: 1,  direccion: 1,  telefono: 1,  email: 1,  // Agrega aquí los demás campos de la colección `cliente`  }  }  ]); | 6. Devuelve un listado que muestre solamente los clientes que sí han realizado algún pago.  SELECT \*  FROM cliente c  WHERE EXISTS (SELECT \*  FROM pago p  WHERE c.idcliente = p.idcliente)  db.cliente.aggregate([  {  $lookup: {  from: "pago",  localField: "idcliente",  foreignField: "idcliente",  as: "pagos"  }  },  {  $match: {  "pagos": { $ne: [] }  }  },  {  $project: {  \_id: 0,  idcliente: 1,  nombrecliente: 1,  direccion: 1,  telefono: 1,  email: 1,  // Agrega aquí los demás campos de la colección `cliente`  }  }  ]); |
| 7 Muestra el nombre de los clientes que hayan realizado pagos junto con el nombre de sus representantes de ventas. | 8. Lista la dirección de las oficinas que tengan clientes en Fuenlabrada. |
| SELECT c.nombrecliente, e.nombre, e.apellido1, e.apellido2  FROM cliente c  INNER JOIN pago p ON c.codigocliente = p.codigocliente  INNER JOIN empleado e ON c.codigoempleadorepventas = e.codigoempleado;  db.cliente.aggregate([  {  $lookup: {  from: "pago",  localField: "idcliente",  foreignField: "idcliente",  as: "pagos"  }  },  {  $unwind: "$pagos"  },  {  $lookup: {  from: "empleado",  localField: "idcodigoempleadoventas",  foreignField: "idempleado",  as: "empleado"  }  },  {  $unwind: "$empleado"  },  {  $project: {  \_id: 0,  nombrecliente: 1,  "empleado.nombre": 1,  "empleado.apellido1": 1,  "empleado.apellido2": 1  }  }  ]); | SELECT DISTINCT o.lineadireccion1, o.lineadireccion2  FROM cliente c  INNER JOIN pago p ON c.idcliente = p.idcliente  INNER JOIN empleado e ON c.idcodigoempleadoventas = e.idempleado  INNER JOIN oficina o ON e.idoficina = o.idoficina  WHERE c.ciudad = 'Fuenlabrada';  db.cliente.aggregate([  {  $match: {  ciudad: 'Fuenlabrada'  }  },  {  $lookup: {  from: "pago",  localField: "idcliente",  foreignField: "idcliente",  as: "pagos"  }  },  {  $unwind: "$pagos"  },  {  $lookup: {  from: "empleado",  localField: "idcodigoempleadoventas",  foreignField: "idempleado",  as: "empleado"  }  },  {  $unwind: "$empleado"  },  {  $lookup: {  from: "oficina",  localField: "empleado.idoficina",  foreignField: "idoficina",  as: "oficina"  }  },  {  $unwind: "$oficina"  },  {  $group: {  \_id: {  lineadireccion1: "$oficina.lineadireccion1",  lineadireccion2: "$oficina.lineadireccion2"  }  }  },  {  $project: {  \_id: 0,  lineadireccion1: "$\_id.lineadireccion1",  lineadireccion2: "$\_id.lineadireccion2"  }  }  ]); |
| Devuelve un listado que muestre el nombre de cada empleados, el nombre de su jefe y el nombre del jefe de sus jefe. | Devuelve un listado de los productos que nunca han aparecido en un pedido. |
| SELECT CONCAT(e.nombre, ' ', e.apellido1, ' ', e.apellido2) AS 'Empleado',  CONCAT(j.nombre, ' ', j.apellido1, ' ', j.apellido2) AS 'Jefe',  CONCAT(m.nombre, ' ', m.apellido1, ' ', m.apellido2) AS 'Jefe del Jefe'  FROM empleado e INNER JOIN empleado j ON e.idempleadojefe = j.idempleado  INNER JOIN empleado m ON j.idempleadojefe = m.idempleado  db.empleado.aggregate([  {  $lookup: {  from: "empleado",  localField: "idempleadojefe",  foreignField: "idempleado",  as: "jefe"  }  },  {  $unwind: "$jefe"  },  {  $lookup: {  from: "empleado",  localField: "jefe.idempleadojefe",  foreignField: "idempleado",  as: "jefeDelJefe"  }  },  {  $unwind: "$jefeDelJefe"  },  {  $project: {  \_id: 0,  "Empleado": { $concat: ["$nombre", " ", "$apellido1", " ", "$apellido2"] },  "Jefe": { $concat: ["$jefe.nombre", " ", "$jefe.apellido1", " ", "$jefe.apellido2"] },  "Jefe del Jefe": { $concat: ["$jefeDelJefe.nombre", " ", "$jefeDelJefe.apellido1", " ", "$jefeDelJefe.apellido2"] }  }  }  ]); | **SELECT** DISTINCT *p*.nombre  **FROM** producto *p* **LEFT** **JOIN** detallepedido *dp* **ON** *p*.idproducto = *dp*.idproducto  **LEFT** **JOIN** pedido *pe* **ON** *dp*.idpedido = *pe*.idpedido  **WHERE** *dp*.idproducto **IS** **NULL**   * Agrupa los documentos por el campo **nombre** del producto, utilizando **$group** para asegurarse de que cada nombre de producto sea único (similar a **DISTINCT** en SQL).   db.producto.aggregate([  {  $lookup: {  from: "detallepedido",  localField: "idproducto",  foreignField: "idproducto",  as: "detallepedido"  }  },  {  $unwind: {  path: "$detallepedido",  preserveNullAndEmptyArrays: true  }  },  {  $lookup: {  from: "pedido",  localField: "detallepedido.idpedido",  foreignField: "idpedido",  as: "pedido"  }  },  {  $unwind: {  path: "$pedido",  preserveNullAndEmptyArrays: true  }  },  {  $match: {  "detallepedido.idproducto": { $eq: null }  }  },  {  $group: {  \_id: "$nombre"  }  },  {  $project: {  \_id: 0,  nombre: "$\_id"  }  }  ]); |
| Calcula la suma de la cantidad total de todos los productos que aparecen en cada uno de los pedidos. | |
| **SELECT** *pe*.idpedido,*c*.nombrecliente , **SUM**(*pr*.precioventa) **AS** *Suma\_de\_total\_de\_productos*  **FROM** (producto *pr* **INNER** **JOIN** detallepedido *dp* **ON** *pr*.idproducto = *dp*.idproducto)  **INNER** **JOIN** pedido *pe* **ON** *dp*.idpedido = *pe*.idpedido  **inner** **join** cliente *c* **on** *c*.idcliente =*pe*.idcliente  **GROUP** **BY** *pe*.idpedido,*c*.nombrecliente | db.producto.aggregate([  {  $lookup: {  from: "detallepedido",  localField: "idproducto",  foreignField: "idproducto",  as: "detallepedido"  }  },  {  $unwind: "$detallepedido"  },  {  $lookup: {  from: "pedido",  localField: "detallepedido.idpedido",  foreignField: "idpedido",  as: "pedido"  }  },  {  $unwind: "$pedido"  },  {  $lookup: {  from: "cliente",  localField: "pedido.idcliente",  foreignField: "idcliente",  as: "cliente"  }  },  {  $unwind: "$cliente"  },  {  $group: {  \_id: {  idpedido: "$pedido.idpedido",  nombrecliente: "$cliente.nombrecliente"  },  Suma\_de\_total\_de\_productos: { $sum: "$precioventa" }  }  },  {  $project: {  \_id: 0,  idpedido: "$\_id.idpedido",  nombrecliente: "$\_id.nombrecliente",  Suma\_de\_total\_de\_productos: 1  }  }  ]); |
| La facturación que ha tenido la empresa en toda la historia, indicando la base imponible, el IVA y el total facturado. La base imponible se calcula sumando el coste del producto por el número de unidades vendidas de la tabla detalle\_pedido. El IVA es el 13% de la base imponible, y el total la suma de los dos campos anteriores. | |
| **SELECT** **SUM**(*dp*.preciounidad \* *dp*.cantidad) **AS** *Base\_Imponible*,  **SUM**((*dp*.preciounidad \* *dp*.cantidad) \* 0.13) **AS** *IVA*,  **SUM**(*dp*.preciounidad \* *dp*.cantidad + ((*dp*.preciounidad \* *dp*.cantidad) \* 0.13)) **AS** *Total*  **FROM** producto *pr* **INNER** **JOIN** detallepedido *dp* **ON** *pr*.idproducto = *dp*.idproducto  **INNER** **JOIN** pedido *pe* **ON** *dp*.idpedido = *pe*.idpedido  **INNER** **JOIN** cliente *c* **ON** *c*.idcliente = *pe*.idcliente  **INNER** **JOIN** pago *pa* **ON** *pa*.idcliente = *c*.idcliente  Lista las ventas totales de los productos que hayan facturado más de 3000 bs. Se mostrará el nombre, unidades vendidas, total facturado y total facturado con impuestos (13% IVA).  Explicación de la Consulta  $lookup (detallepedido):  Unir producto con detallepedido en el campo idproducto y almacenar los resultados en el campo detallepedidos.  $unwind (detallepedidos):  Desanidar el array detallepedidos para trabajar con documentos individuales.  $lookup (pedido):  Unir detallepedido con pedido en el campo idpedido y almacenar los resultados en el campo pedidos.  $unwind (pedidos):  Desanidar el array pedidos para trabajar con documentos individuales.  $lookup (cliente):  Unir pedido con cliente en el campo idcliente y almacenar los resultados en el campo clientes.  $unwind (clientes):  Desanidar el array clientes para trabajar con documentos individuales.  $lookup (pago):  Unir cliente con pago en el campo idcliente y almacenar los resultados en el campo pagos.  $unwind (pagos):  Desanidar el array pagos para trabajar con documentos individuales.  $group:  Agrupar los documentos para calcular Base\_Imponible, IVA y Total utilizando $sum, $multiply, y $add.  $project:  Proyectar los campos Base\_Imponible, IVA y Total en el resultado final, excluyendo el campo \_id. | db.producto.aggregate([  {  $lookup: {  from: "detallepedido",  localField: "idproducto",  foreignField: "idproducto",  as: "detallepedidos"  }  },  {  $unwind: "$detallepedidos"  },  {  $lookup: {  from: "pedido",  localField: "detallepedidos.idpedido",  foreignField: "idpedido",  as: "pedidos"  }  },  {  $unwind: "$pedidos"  },  {  $lookup: {  from: "cliente",  localField: "pedidos.idcliente",  foreignField: "idcliente",  as: "clientes"  }  },  {  $unwind: "$clientes"  },  {  $lookup: {  from: "pago",  localField: "clientes.idcliente",  foreignField: "idcliente",  as: "pagos"  }  },  {  $unwind: "$pagos"  },  {  $group: {  \_id: null,  Base\_Imponible: { $sum: { $multiply: ["$detallepedidos.preciounidad", "$detallepedidos.cantidad"] } },  IVA: { $sum: { $multiply: [{ $multiply: ["$detallepedidos.preciounidad", "$detallepedidos.cantidad"] }, 0.13] } },  Total: {  $sum: {  $add: [  { $multiply: ["$detallepedidos.preciounidad", "$detallepedidos.cantidad"] },  { $multiply: [{ $multiply: ["$detallepedidos.preciounidad", "$detallepedidos.cantidad"] }, 0.13] }  ]  }  }  }  },  {  $project: {  \_id: 0,  Base\_Imponible: 1,  IVA: 1,  Total: 1  }  }  ]); |
| Lista las ventas totales de los productos que hayan facturado más de 3000 bs. Se mostrará el nombre, unidades vendidas, total facturado y total facturado con impuestos (13% IVA). | |
| **SELECT** *pr*.codigo\_producto,  *dp*.cantidad **AS** *Unidades\_Vendidas*,  **SUM**(*dp*.preciounidad \* *dp*.cantidad) **AS** *Total*,  **SUM**(*dp*.preciounidad \* *dp*.cantidad + ((*dp*.preciounidad \* *dp*.cantidad) \* 0.13)) **AS** *Total\_con\_impuestos*  **FROM** producto *pr* **INNER** **JOIN** detallepedido *dp* **ON** *pr*.idproducto = *dp*.idproducto  **INNER** **JOIN** pedido *pe* **ON** *dp*.idpedido = *pe*.idpedido  **INNER** **JOIN** cliente *c* **ON** *c*.idcliente = *pe*.idcliente  **INNER** **JOIN** pago *pa* **ON** *pa*.idcliente = *c*.idcliente  **GROUP** **BY** *pr*.codigo\_producto,*dp*.cantidad, *pa*.total  **HAVING** Total > 3000 | db.producto.aggregate([  {  $lookup: {  from: "detallepedido",  localField: "idproducto",  foreignField: "idproducto",  as: "detallepedidos"  }  },  {  $unwind: "$detallepedidos"  },  {  $lookup: {  from: "pedido",  localField: "detallepedidos.idpedido",  foreignField: "idpedido",  as: "pedidos"  }  },  { $unwind: "$pedidos" },  {  $lookup: {  from: "cliente",  localField: "pedidos.idcliente",  foreignField: "idcliente",  as: "clientes"  }  },  {  $unwind: "$clientes"  },  {  $lookup: {  from: "pago",  localField: "clientes.idcliente",  foreignField: "idcliente",  as: "pagos"  }  },  {  $unwind: "$pagos"  },  {  $group: {  \_id: {  codigo\_producto: "$codigo\_producto",  cantidad: "$detallepedidos.cantidad"  },  Total: { $sum: { $multiply: ["$detallepedidos.preciounidad", "$detallepedidos.cantidad"] } },  Total\_con\_impuestos: {  $sum: {  $add: [  { $multiply: ["$detallepedidos.preciounidad", "$detallepedidos.cantidad"] },  { $multiply: [{ $multiply: ["$detallepedidos.preciounidad", "$detallepedidos.cantidad"] }, 0.13] }  ]  }  }  }  },  {  $match: { Total: { $gt: 3000 } }  },  {  $project: {  \_id: 0,  codigo\_producto: "$\_id.codigo\_producto",  Unidades\_Vendidas: "$\_id.cantidad",  Total: 1,  Total\_con\_impuestos: 1  }  }  ]); |