

MONGODB ASSIGNMENT

1. Create a collection called sales. Create multiple documents with keys like store (id, name etc.), product (type, name etc.), quantity, region, items, amount, price, sales, customer age etc. with arrays and other data types wherever applicable. Please go through the queries first and you would know how many documents to create with what key value pairs.

1. Write a MongoDB query to display all the documents in the collection.

```
test> use SALES
switched to db SALES
SALES> show collections
sales
SALES> db.sales.find()
[
  {
    _id: ObjectId("655a2ac3d337a7f96e12e432"),
    store: { id: 7, name: 'Store A' },
    product: { type: 'Clothing', name: 'T-Shirt' },
    quantity: 8,
    region: 'Delhi',
    items: [ 'item60', 'item61' ],
    amount: 1200,
    price: 150,
    sales: true,
    customer: { age: 28, name: 'Michael Johnson' },
    customer_rating: 85,
    sales_agent: { id: 101, name: 'Sara Thomas' }
  },
  {
    _id: ObjectId("655a2b10d337a7f96e12e434"),
    store: { id: 2, name: 'Store B' },
    product: { type: 'Electronics', name: 'Smartwatch' },
    quantity: 15,
    region: 'Pune',
    items: [ 'item70', 'item71' ],
    amount: 3500,
    price: 230,
    sales: true,
    customer: { age: 35, name: 'Jessy David' },
    customer_rating: 95,
    sales_agent: { id: 102, name: 'Christy' }
  },
]
```

```
_id: ObjectId("655a2b4bd337a7f96e12e436"),
SALES>

SALES>
  price: 200,
  sales: true,
  customer: { age: 30, name: 'Ryan Wilson' },
  customer_rating: 92,
  sales_agent: { id: 103, name: 'Emily' }
},
{
  _id: ObjectId("655a2bc4d337a7f96e12e438"),
  store: { id: 4, name: 'Store D' },
  product: { type: 'Clothing', name: 'Jeans' },
  quantity: 12,
SALES>
  items: [ 'item90', 'item91' ],
  amount: 1800,
  price: 150,
  sales: true,
  customer: { age: 22, name: 'Martin' },
  customer_rating: 88,
  sales_agent: { id: 104, name: 'Daniel' }
},
{
  _id: ObjectId("655a2bf1d337a7f96e12e43a"),
  store: { id: 5, name: 'Store E' },
  product: { type: 'Merchandise', name: 'T-Shirt' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: 'Sophie ' },
  customer_rating: 80,
  sales_agent: { id: 105, name: 'Jason' }
},
```

```

{
  _id: ObjectId("655a2dadbc0d8f834969dc33"),
  store: { id: 6, name: 'Store F' },
  product: { type: 'Electronics', name: 'Washing Machine' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: ' Wilson' },
  customer_rating: 80,
  sales_agent: { id: 105, name: 'Tony' }
},
{
  _id: ObjectId("655a35638bc0915ea35e300f"),
  store: { id: 1, name: 'Store G' },
  product: { type: 'Electronics', name: 'Television' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: ' Wilson' },
  customer_rating: 80,
  sales_agent: { id: 105, name: 'Tony' }
},
{
  _id: ObjectId("655a35638bc0915ea35e3010"),
  store: { id: 8, name: 'Store F' },
  product: { type: 'Electronics', name: 'Mixer' },
  quantity: 10,
  region: 'Delhi',
  items: [ 'ite45', 'item47' ],
  amount: 1500,
  price: 150,
  sales: true,
  customer: { age: 55, name: ' Sanjay' },
  customer_rating: 75,
  sales_agent: { id: 110, name: 'Lizzy' }
},

```

```

{
  _id: ObjectId("655a35638bc0915ea35e3011"),
  store: { id: 9, name: 'Store H' },
  product: { type: 'Book', name: 'Novel' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: 'Reena' },
  customer_rating: 88,
  sales_agent: { id: 125, name: 'Bony' }
},
{
  _id: ObjectId("655a35638bc0915ea35e3012"),
  store: { id: 10, name: 'Store I' },
  product: { type: 'Appliances', name: 'Refrigerator' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item70', 'item71' ],
  amount: 2000,
  price: 150,
  sales: true,
  customer: { age: 45, name: 'Joseph' },
  customer_rating: 95,
  sales_agent: { id: 109, name: 'Mary' }
}
]

```

2. Write a MongoDB query to display the fields; products and price for all the documents in the collection.

```

SALES> db.sales.find({}, { product: 1, price: 1, _id: 0 })
[
  { product: { type: 'Clothing', name: 'T-Shirt' }, price: 150 },
  { product: { type: 'Electronics', name: 'Smartwatch' }, price: 230 },
  {
    product: { type: 'Books', name: 'Science Fiction Novel' },
    price: 200
  },
  { product: { type: 'Clothing', name: 'Jeans' }, price: 150 },
  { product: { type: 'Merchandise', name: 'T-Shirt' }, price: 150 },
  {
    product: { type: 'Electronics', name: 'Washing Machine' },
    price: 150
  },
  { product: { type: 'Electronics', name: 'Television' }, price: 150 },
  { product: { type: 'Electronics', name: 'Mixer' }, price: 150 },
  { product: { type: 'Book', name: 'Novel' }, price: 150 },
  { product: { type: 'Appliances', name: 'Refrigerator' }, price: 150 }
]

```

3. Write a MongoDB query to display all the fields but exclude the field `_id` for all the documents in the collection.

```
SALES> db.sales.find({}, { _id: 0 })
[
  {
    store: { id: 7, name: 'Store A' },
    product: { type: 'Clothing', name: 'T-Shirt' },
    quantity: 8,
    region: 'Delhi',
    items: [ 'item60', 'item61' ],
    amount: 1200,
    price: 150,
    sales: true,
    customer: { age: 28, name: 'Michael Johnson' },
    customer_rating: 85,
    sales_agent: { id: 101, name: 'Sara Thomas' }
  },
  {
    store: { id: 2, name: 'Store B' },
    product: { type: 'Electronics', name: 'Smartwatch' },
    quantity: 15,
    region: 'Pune',
    items: [ 'item70', 'item71' ],
    amount: 3500,
    price: 230,
    sales: true,
    customer: { age: 35, name: 'Jessy David' },
    customer_rating: 95,
    sales_agent: { id: 102, name: 'Christy' }
  },
  {
    store: { id: 3, name: 'Store C' },
    product: { type: 'Books', name: 'Science Fiction Novel' },
    quantity: 10,
    region: 'Jaipur',
    items: [ 'item80', 'item81' ],
    amount: 2000,
    price: 200,
    sales: true,
    customer: { age: 30, name: 'Ryan Wilson' },
    customer_rating: 92,
    sales_agent: { id: 103, name: 'Emily' }
  },
  —
]
```

```
{
  store: { id: 4, name: 'Store D' },
  product: { type: 'Clothing', name: 'Jeans' },
  quantity: 12,
  region: 'Delhi',
  items: [ 'item90', 'item91' ],
  amount: 1800,
  price: 150,
  sales: true,
  customer: { age: 22, name: 'Martin' },
  customer_rating: 88,
  sales_agent: { id: 104, name: 'Daniel' }
},
{
  store: { id: 5, name: 'Store E' },
  product: { type: 'Merchandise', name: 'T-Shirt' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: 'Sophie ' },
  customer_rating: 80,
  sales_agent: { id: 105, name: 'Jason' }
},
{
  store: { id: 6, name: 'Store F' },
  product: { type: 'Electronics', name: 'Washing Machine' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: ' Wilson' },
  customer_rating: 80,
  sales_agent: { id: 105, name: 'Tony' }
},
{
```

```

{
  store: { id: 1, name: 'Store G' },
  product: { type: 'Electronics', name: 'Television' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: 'Wilson' },
  customer_rating: 80,
  sales_agent: { id: 105, name: 'Tony' }
},
{
  store: { id: 8, name: 'Store F' },
  product: { type: 'Electronics', name: 'Mixer' },
  quantity: 10,
  region: 'Delhi',
  items: [ 'ite45', 'item47' ],
  amount: 1500,
  price: 150,
  sales: true,
  customer: { age: 55, name: 'Sanjay' },
  customer_rating: 75,
  sales_agent: { id: 110, name: 'Lizzy' }
},
{
  store: { id: 9, name: 'Store H' },
  product: { type: 'Book', name: 'Novel' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: 'Reena' },
  customer_rating: 88,
  sales_agent: { id: 125, name: 'Bony' }
},

```

```

{
  store: { id: 10, name: 'Store I' },
  product: { type: 'Appliances', name: 'Refrigerator' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item70', 'item71' ],
  amount: 2000,
  price: 150,
  sales: true,
  customer: { age: 45, name: 'Joseph' },
  customer_rating: 95,
  sales_agent: { id: 109, name: 'Mary' }
}
]

```

4. Write a MongoDB query to display the fields product and quantity, but exclude the field _id for all the documents in the collection.

```
SALES> db.sales.find({}, { product: 1, quantity: 1, _id: 0 })
[
  { product: { type: 'Clothing', name: 'T-Shirt' }, quantity: 8 },
  {
    product: { type: 'Electronics', name: 'Smartwatch' },
    quantity: 15
  },
  {
    product: { type: 'Books', name: 'Science Fiction Novel' },
    quantity: 10
  },
  { product: { type: 'Clothing', name: 'Jeans' }, quantity: 12 },
  { product: { type: 'Merchandise', name: 'T-Shirt' }, quantity: 20 },
  {
    product: { type: 'Electronics', name: 'Washing Machine' },
    quantity: 20
  },
  {
    product: { type: 'Electronics', name: 'Television' },
    quantity: 20
  },
  { product: { type: 'Electronics', name: 'Mixer' }, quantity: 10 },
  { product: { type: 'Book', name: 'Novel' }, quantity: 20 },
  {
    product: { type: 'Appliances', name: 'Refrigerator' },
    quantity: 20
  }
]
```

5. Write a MongoDB query to display all the sales that happened in Delhi.


```
SALES> db.sales.find({ region: "Delhi" })
[
  {
    _id: ObjectId("655a2ac3d337a7f96e12e432"),
    store: { id: 7, name: 'Store A' },
    product: { type: 'Clothing', name: 'T-Shirt' },
    quantity: 8,
    region: 'Delhi',
    items: [ 'item60', 'item61' ],
    amount: 1200,
    price: 150,
    sales: true,
    customer: { age: 28, name: 'Michael Johnson' },
    customer_rating: 85,
    sales_agent: { id: 101, name: 'Sara Thomas' }
  },
  {
    _id: ObjectId("655a2bc4d337a7f96e12e438"),
    store: { id: 4, name: 'Store D' },
    product: { type: 'Clothing', name: 'Jeans' },
    quantity: 12,
    region: 'Delhi',
    items: [ 'item90', 'item91' ],
    amount: 1800,
    price: 150,
    sales: true,
    customer: { age: 22, name: 'Martin' },
    customer_rating: 88,
```

SALES>

```
  },
  {
    _id: ObjectId("655a2bf1d337a7f96e12e43a"),
    store: { id: 5, name: 'Store E' },
    product: { type: 'Merchandise', name: 'T-Shirt' },
    quantity: 20,
    region: 'Delhi',
    items: [ 'item100', 'item101' ],
    amount: 3000,
    price: 150,
    sales: true,
    customer: { age: 25, name: 'Sophie ' },
    customer_rating: 80,
    sales_agent: { id: 105, name: 'Jason' }
  },
  {
    _id: ObjectId("655a2dadbc0d8f834969dc33"),
    store: { id: 6, name: 'Store F' },
    product: { type: 'Electronics', name: 'Washing Machine' },
    quantity: 20,
    region: 'Delhi',
    items: [ 'item100', 'item101' ],
    amount: 3000,
    price: 150,
    sales: true,
    customer: { age: 25, name: ' Wilson' },
    customer_rating: 80,
    sales_agent: { id: 105, name: 'Tony' }
  },
  {
```

```
{
  _id: ObjectId("655a35638bc0915ea35e300f"),
  store: { id: 1, name: 'Store G' },
  product: { type: 'Electronics', name: 'Television' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: 'Wilson' },
  customer_rating: 80,
  sales_agent: { id: 105, name: 'Tony' }
},
{
  _id: ObjectId("655a35638bc0915ea35e3010"),
  store: { id: 8, name: 'Store F' },
  product: { type: 'Electronics', name: 'Mixer' },
  quantity: 10,
  region: 'Delhi',
  items: [ 'ite45', 'item47' ],
  amount: 1500,
  price: 150,
  sales: true,
  customer: { age: 55, name: 'Sanjay' },
  customer_rating: 75,
  sales_agent: { id: 110, name: 'Lizzy' }
},
{
  _id: ObjectId("655a35638bc0915ea35e3011"),
  store: { id: 9, name: 'Store H' },
  product: { type: 'Book', name: 'Novel' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: 'Reena' },
  customer_rating: 88,
  sales_agent: { id: 125, name: 'Bony' }
},
```

```

{
  _id: ObjectId("655a35638bc0915ea35e3012"),
  store: { id: 10, name: 'Store I' },
  product: { type: 'Appliances', name: 'Refrigerator' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item70', 'item71' ],
  amount: 2000,
  price: 150,
  sales: true,
  customer: { age: 45, name: 'Joseph' },
  customer_rating: 95,
  sales_agent: { id: 109, name: 'Mary' }
}
]

```

6. Write a MongoDB query to display the first 5 sales happened in Delhi.

```

SALES> db.sales.find({ region: "Delhi" }).limit(5)
[
  {
    _id: ObjectId("655a2ac3d337a7f96e12e432"),
    store: { id: 7, name: 'Store A' },
    product: { type: 'Clothing', name: 'T-Shirt' },
    quantity: 8,
    region: 'Delhi',
    items: [ 'item60', 'item61' ],
    amount: 1200,
    price: 150,
    sales: true,
    customer: { age: 28, name: 'Michael Johnson' },
    customer_rating: 85,
    sales_agent: { id: 101, name: 'Sara Thomas' }
  },
  {
    _id: ObjectId("655a2bc4d337a7f96e12e438"),
    store: { id: 4, name: 'Store D' },
    product: { type: 'Clothing', name: 'Jeans' },
    quantity: 12,
    region: 'Delhi',
    items: [ 'item90', 'item91' ],
    amount: 1800,
    price: 150,
    sales: true,
    customer: { age: 22, name: 'Martin' },
    customer_rating: 88,
    sales_agent: { id: 104, name: 'Daniel' }
  },
  {
    _id: ObjectId("655a2c4d337a7f96e12e439"),
    store: { id: 5, name: 'Store B' },
    product: { type: 'Appliances', name: 'Washing Machine' },
    quantity: 10,
    region: 'Delhi',
    items: [ 'item80', 'item81' ],
    amount: 1500,
    price: 150,
    sales: true,
    customer: { age: 35, name: 'John Doe' },
    customer_rating: 90,
    sales_agent: { id: 102, name: 'Alice' }
  },
  {
    _id: ObjectId("655a2d4d337a7f96e12e43a"),
    store: { id: 6, name: 'Store C' },
    product: { type: 'Clothing', name: 'Socks' },
    quantity: 50,
    region: 'Delhi',
    items: [ 'item70', 'item71' ],
    amount: 500,
    price: 10,
    sales: true,
    customer: { age: 40, name: 'Emily White' },
    customer_rating: 80,
    sales_agent: { id: 103, name: 'Bob' }
  },
  {
    _id: ObjectId("655a2e4d337a7f96e12e43b"),
    store: { id: 8, name: 'Store E' },
    product: { type: 'Appliances', name: 'Microwave' },
    quantity: 15,
    region: 'Delhi',
    items: [ 'item90', 'item91' ],
    amount: 2250,
    price: 150,
    sales: true,
    customer: { age: 30, name: 'David Brown' },
    customer_rating: 85,
    sales_agent: { id: 105, name: 'Charlie' }
  }
]

```

```

    _id: ObjectId("655a2bf1d337a7f96e12e43a"),
    store: { id: 5, name: 'Store E' },
    product: { type: 'Merchandise', name: 'T-Shirt' },
    quantity: 20,
    region: 'Delhi',
    items: [ 'item100', 'item101' ],
    amount: 3000,
    price: 150,
    sales: true,
    customer: { age: 25, name: 'Sophie ' },
    customer_rating: 80,
    sales_agent: { id: 105, name: 'Jason' }
SALES>
{
  _id: ObjectId("655a2dadbc0d8f834969dc33"),
  store: { id: 6, name: 'Store F' },
  product: { type: 'Electronics', name: 'Washing Machine' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: ' Wilson' },
  customer_rating: 80,
  sales_agent: { id: 105, name: 'Tony' }
},
{
  _id: ObjectId("655a35638bc0915ea35e300f"),
  store: { id: 1, name: 'Store G' },
  product: { type: 'Electronics', name: 'Television' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: ' Wilson' },
  customer_rating: 80,
  sales_agent: { id: 105, name: 'Tony' }
}
]

```

7. Write a MongoDB query to display the next 5 sales after skipping first 5 which happened in Delhi.

```
SALES> db.sales.find({ region: "Delhi" }).skip(5).limit(5)
[
  {
    _id: ObjectId("655a35638bc0915ea35e3010"),
    store: { id: 8, name: 'Store F' },
    product: { type: 'Electronics', name: 'Mixer' },
    quantity: 10,
    region: 'Delhi',
    items: [ 'ite45', 'item47' ],
    amount: 1500,
    price: 150,
    sales: true,
    customer: { age: 55, name: 'Sanjay' },
    customer_rating: 75,
    sales_agent: { id: 110, name: 'Lizzy' }
  },
  {
    _id: ObjectId("655a35638bc0915ea35e3011"),
    store: { id: 9, name: 'Store H' },
    product: { type: 'Book', name: 'Novel' },
    quantity: 20,
    region: 'Delhi',
    items: [ 'item100', 'item101' ],
    amount: 3000,
    price: 150,
    sales: true,
    customer: { age: 25, name: 'Reena' },
    customer_rating: 88,
    sales_agent: { id: 125, name: 'Bony' }
  },
  {
    _id: ObjectId("655a35638bc0915ea35e3012"),
    store: { id: 10, name: 'Store I' },
    product: { type: 'Appliances', name: 'Refrigerator' },
    quantity: 20,
    region: 'Delhi',
    items: [ 'item70', 'item71' ],
    amount: 2000,
    price: 150,
    sales: true,
    customer: { age: 45, name: 'Joseph' },
    customer_rating: 95,
    sales_agent: { id: 130, name: 'Ravi' }
  }
]
```

8. Write a MongoDB query to find the sales agents who achieved a score more than 90.

```
SALES> db.sales.find({ "sales_agent.score": { $gt: 80, $lt: 100 } })
[
  {cknowledged: true,
    _id: ObjectId("655a2ac3d337a7f96e12e432"),
    store: { id: 7, name: 'Store A' },
    product: { type: 'Clothing', name: 'T-Shirt' },
    quantity: 8, 0
    region: 'Delhi',
    items: [ 'item60', 'item61' ],
    amount: 1200,
    price: 150,
    sales: true,
    customer: { age: 28, name: 'Michael Johnson' },
    customer_rating: 85,
    sales_agent: { id: 101, name: 'Sara Thomas', score: 95 }
  },
  {
    _id: ObjectId("655a2b10d337a7f96e12e434"),
    store: { id: 2, name: 'Store B' },
    product: { type: 'Electronics', name: 'Smartwatch' },
    quantity: 15,
    region: 'Pune',
    items: [ 'item70', 'item71' ],
    amount: 3500,
    price: 230,
    sales: true,
    customer: { age: 35, name: 'Jessy David' },
    customer_rating: 95,
    sales_agent: { id: 102, name: 'Christy', score: 95 }
  },
]
```

```

{
  _id: ObjectId("655a2b4bd337a7f96e12e436"),
  store: { id: 3, name: 'Store C' },
  product: { type: 'Books', name: 'Science Fiction Novel' },
  quantity: 10,
  region: 'Jaipur',
  items: [ 'item80', 'item81' ],
  amount: 2000,
  price: 200,
  sales: true,
  customer: { age: 30, name: 'Ryan Wilson' },
  customer_rating: 92,
  sales_agent: { id: 103, name: 'Emily', score: 95 }
},
{
  _id: ObjectId("655a2bc4d337a7f96e12e438"),
  store: { id: 4, name: 'Store D' },
  product: { type: 'Clothing', name: 'Jeans' },
  quantity: 12,
  region: 'Delhi',
  items: [ 'item90', 'item91' ],
  amount: 1800,
  price: 150,
  sales: true,
  customer: { age: 22, name: 'Martin' },
  customer_rating: 88,
  sales_agent: { id: 104, name: 'Daniel', score: 95 }
},
{
  _id: ObjectId("655a2bf1d337a7f96e12e43a"),
  store: { id: 5, name: 'Store E' },
  product: { type: 'Merchandise', name: 'T-Shirt' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: 'Sophie ' },
  customer_rating: 80,
  sales_agent: { id: 105, name: 'Jason', score: 95 }
},
{

```

9. Write a MongoDB query to find the sales agents that achieved a score, more than 80 but less than 100.


```
SALES> db.sales.find({ "sales_agent.score": { $gt: 80, $lt: 100 } })
[
  {cknowledged: true,
    _id: ObjectId("655a2ac3d337a7f96e12e432"),
    store: { id: 7, name: 'Store A' },
    product: { type: 'Clothing', name: 'T-Shirt' },
    quantity: 8, 0
    region: 'Delhi',
    items: [ 'item60', 'item61' ],
    amount: 1200,
    price: 150,
    sales: true,
    customer: { age: 28, name: 'Michael Johnson' },
    customer_rating: 85,
    sales_agent: { id: 101, name: 'Sara Thomas', score: 95 }
  },
  {
    _id: ObjectId("655a2b10d337a7f96e12e434"),
    store: { id: 2, name: 'Store B' },
    product: { type: 'Electronics', name: 'Smartwatch' },
    quantity: 15,
    region: 'Pune',
    items: [ 'item70', 'item71' ],
    amount: 3500,
    price: 230,
    sales: true,
    customer: { age: 35, name: 'Jessy David' },
    customer_rating: 95,
    sales_agent: { id: 102, name: 'Christy', score: 95 }
  },
]
```

```

{
  _id: ObjectId("655a2b4bd337a7f96e12e436"),
  store: { id: 3, name: 'Store C' },
  product: { type: 'Books', name: 'Science Fiction Novel' },
  quantity: 10,
  region: 'Jaipur',
  items: [ 'item80', 'item81' ],
  amount: 2000,
  price: 200,
  sales: true,
  customer: { age: 30, name: 'Ryan Wilson' },
  customer_rating: 92,
  sales_agent: { id: 103, name: 'Emily', score: 95 }
},
{
  _id: ObjectId("655a2bc4d337a7f96e12e438"),
  store: { id: 4, name: 'Store D' },
  product: { type: 'Clothing', name: 'Jeans' },
  quantity: 12,
  region: 'Delhi',
  items: [ 'item90', 'item91' ],
  amount: 1800,
  price: 150,
  sales: true,
  customer: { age: 22, name: 'Martin' },
  customer_rating: 88,
  sales_agent: { id: 104, name: 'Daniel', score: 95 }
},
{
  _id: ObjectId("655a2bf1d337a7f96e12e43a"),
  store: { id: 5, name: 'Store E' },
  product: { type: 'Merchandise', name: 'T-Shirt' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: 'Sophie ' },
  customer_rating: 80,
  sales_agent: { id: 105, name: 'Jason', score: 95 }
},
{

```

10. Write a MongoDB query to find the outlets which are located in Jaipur.

```

SALES> db.sales.updateMany({ "store.city": "Jaipur" }, { $set: { "is_in_jaipur": true } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
SALES> db.sales.find({ "is_in_jaipur": true })
SALES> db.sales.find({ "store.city": "Jaipur" })

```

11. Write a MongoDB query to find the stores that do not sell merchandise and the user rating is more than 70.

```
SALES> db.sales.find({
...   $and: [
...     { "product.type": { $ne: "Merchandise" } },
...     { "customer_rating": { $gt: 70 } }
...   ]
... })
[
  {
    _id: ObjectId("655a2ac3d337a7f96e12e432"),
    store: { id: 7, name: 'Store A' },
    product: { type: 'Clothing', name: 'T-Shirt' },
    quantity: 8,
    region: 'Delhi',
    items: [ 'item60', 'item61' ],
    amount: 1200,
    price: 150,
    sales: true,
    customer: { age: 28, name: 'Michael Johnson' },
    customer_rating: 85,
    sales_agent: { id: 101, name: 'Sara Thomas', score: 95 }
  },
  {
    _id: ObjectId("655a2b10d337a7f96e12e434"),
    store: { id: 2, name: 'Store B' },
    product: { type: 'Electronics', name: 'Smartwatch' },
    quantity: 15,
    region: 'Pune',
    items: [ 'item70', 'item71' ],
    amount: 3500,
    price: 230,
    sales: true,
    customer: { age: 35, name: 'Jessy David' },
    customer_rating: 95,
    sales_agent: { id: 102, name: 'Christy', score: 95 }
  },
]
```

```

    {
      _id: ObjectId("655a2b4bd337a7f96e12e436"),
      store: { id: 3, name: 'Store C' },
      product: { type: 'Books', name: 'Science Fiction Novel' },
      quantity: 10,
      region: 'Jaipur',
      items: [ 'item80', 'item81' ],
      amount: 2000,
      price: 200,
      sales: true,
      customer: { age: 30, name: 'Ryan Wilson' },
      customer_rating: 92,
      sales_agent: { id: 103, name: 'Emily', score: 95 }
    },
    {
      _id: ObjectId("655a2bc4d337a7f96e12e438"),
      store: { id: 4, name: 'Store D' },
      product: { type: 'Clothing', name: 'Jeans' },
      quantity: 12,
      region: 'Delhi',
      items: [ 'item90', 'item91' ],
      amount: 1800,
      price: 150,
      sales: true,
      customer: { age: 22, name: 'Martin' },
      customer_rating: 88,
      sales_agent: { id: 104, name: 'Daniel', score: 95 }
    },
    {
      _id: ObjectId("655a2dadbc0d8f834969dc33"),
      store: { id: 6, name: 'Store F' },
      product: { type: 'Electronics', name: 'Washing Machine' },
      quantity: 20,
      region: 'Delhi',
      items: [ 'item100', 'item101' ],
      amount: 3000,
      price: 150,
      sales: true,
      customer: { age: 25, name: 'Wilson' },
      customer_rating: 80,
      sales_agent: { id: 105, name: 'Tony', score: 95 }
    },
  ],
  {

```

```

},
{
  _id: ObjectId("655a35638bc0915ea35e300f"),
  store: { id: 1, name: 'Store G' },
  product: { type: 'Electronics', name: 'Television' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: 'Wilson' },
  customer_rating: 80,
  sales_agent: { id: 105, name: 'Tony', score: 95 }
},
{
  _id: ObjectId("655a35638bc0915ea35e3010"),
  store: { id: 8, name: 'Store F' },
  product: { type: 'Electronics', name: 'Mixer' },
  quantity: 10,
  region: 'Delhi',
  items: [ 'ite45', 'item47' ],
  amount: 1500,
  price: 150,
  sales: true,
  customer: { age: 55, name: 'Sanjay' },
  customer_rating: 75,
  sales_agent: { id: 110, name: 'Lizzy', score: 95 }
},
{
  _id: ObjectId("655a35638bc0915ea35e3011"),
  store: { id: 9, name: 'Store H' },
  product: { type: 'Book', name: 'Novel' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: 'Reena' },
  customer_rating: 88,
  sales_agent: { id: 125, name: 'Bony', score: 95 }
},

```

12. Do the same query without using \$and operator.

```
SALES> db.sales.find({
...   "product.type": { $ne: "Merchandise" },
...   "customer_rating": { $gt: 70 }
... })
[
  {
    _id: ObjectId("655a2ac3d337a7f96e12e432"),
    store: { id: 7, name: 'Store A' },
    product: { type: 'Clothing', name: 'T-Shirt' },
    quantity: 8,
    region: 'Delhi',
    items: [ 'item60', 'item61' ],
    amount: 1200,
    price: 150,
    sales: true,
    customer: { age: 28, name: 'Michael Johnson' },
    customer_rating: 85,
    sales_agent: { id: 101, name: 'Sara Thomas', score: 95 }
  },
  {
    _id: ObjectId("655a2b10d337a7f96e12e434"),
    store: { id: 2, name: 'Store B' },
    product: { type: 'Electronics', name: 'Smartwatch' },
    quantity: 15,
    region: 'Pune',
    items: [ 'item70', 'item71' ],
    amount: 3500,
    price: 230,
    sales: true,
    customer: { age: 35, name: 'Jessy David' },
    customer_rating: 95,
    sales_agent: { id: 102, name: 'Christy', score: 95 }
  },
]
```

```

},
{
  _id: ObjectId("655a2b4bd337a7f96e12e436"),
  store: { id: 3, name: 'Store C' },
  product: { type: 'Books', name: 'Science Fiction Novel' },
  quantity: 10,
  region: 'Jaipur',
  items: [ 'item80', 'item81' ],
  amount: 2000,
  price: 200,
  sales: true,
  customer: { age: 30, name: 'Ryan Wilson' },
  customer_rating: 92,
  sales_agent: { id: 103, name: 'Emily', score: 95 }
},
{
  _id: ObjectId("655a2bc4d337a7f96e12e438"),
  store: { id: 4, name: 'Store D' },
  product: { type: 'Clothing', name: 'Jeans' },
  quantity: 12,
  region: 'Delhi',
  items: [ 'item90', 'item91' ],
  amount: 1800,
  price: 150,
  sales: true,
  customer: { age: 22, name: 'Martin' },
  customer_rating: 88,
  sales_agent: { id: 104, name: 'Daniel', score: 95 }
},
{
  _id: ObjectId("655a2dadbc0d8f834969dc33"),
  store: { id: 6, name: 'Store F' },
  product: { type: 'Electronics', name: 'Washing Machine' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: 'Wilson' },
  customer_rating: 80,
  sales_agent: { id: 105, name: 'Tony', score: 95 }
},
{

```

```

},
{
  _id: ObjectId("655a35638bc0915ea35e300f"),
  store: { id: 1, name: 'Store G' },
  product: { type: 'Electronics', name: 'Television' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: 'Wilson' },
  customer_rating: 80,
  sales_agent: { id: 105, name: 'Tony', score: 95 }
},
{
  _id: ObjectId("655a35638bc0915ea35e3010"),
  store: { id: 8, name: 'Store F' },
  product: { type: 'Electronics', name: 'Mixer' },
  quantity: 10,
  region: 'Delhi',
  items: [ 'ite45', 'item47' ],
  amount: 1500,
  price: 150,
  sales: true,
  customer: { age: 55, name: 'Sanjay' },
  customer_rating: 75,
  sales_agent: { id: 110, name: 'Lizzy', score: 95 }
},
{
  _id: ObjectId("655a35638bc0915ea35e3011"),
  store: { id: 9, name: 'Store H' },
  product: { type: 'Book', name: 'Novel' },
  quantity: 20,
  region: 'Delhi',
  items: [ 'item100', 'item101' ],
  amount: 3000,
  price: 150,
  sales: true,
  customer: { age: 25, name: 'Reena' },
  customer_rating: 88,
  sales_agent: { id: 125, name: 'Bony', score: 95 }
},

```

13. Write a MongoDB query to find the stores which do not sell any merchandise and achieved a user rating of 90+ and do not belong to Delhi. The documents must be displayed according to the products in descending order.


```

...   "product.type": { $ne: "Merchandise" },
...   "customer_rating": { $gte: 90 },
...   "store.city": { $ne: "Delhi" }
... }).sort({ "product.name": -1 })
[
  {
    _id: ObjectId("655a2b10d337a7f96e12e434"),
    store: { id: 2, name: 'Store B' },
    product: { type: 'Electronics', name: 'Smartwatch' },
    quantity: 15,
    region: 'Pune',
    items: [ 'item70', 'item71' ],
    amount: 3500,
    price: 230,
    sales: true,
    customer: { age: 35, name: 'Jessy David' },
    customer_rating: 95,
    sales_agent: { id: 102, name: 'Christy', score: 95 }
  },
  {
    _id: ObjectId("655a2b4bd337a7f96e12e436"),
    store: { id: 3, name: 'Store C' },
    product: { type: 'Books', name: 'Science Fiction Novel' },
    quantity: 10,
    region: 'Jaipur',
    items: [ 'item80', 'item81' ],
    amount: 2000,
    price: 200,
    sales: true,
    customer: { age: 30, name: 'Ryan Wilson' },
    customer_rating: 92,
    sales_agent: { id: 103, name: 'Emily', score: 95 }
  },
]

```

14. Write a MongoDB query to find the store Id, name, and total sales for those stores which contain 'A' as first letter for the name of the city.

```

SALES> db.sales.updateOne(
...   { "store.id": 10 },
...   { $set: { "store.name": "New Store Name" } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
SALES> db.sales.updateOne(
...   { "store.id": 7 },
...   { $set: { "store.name": "Azure" } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
SALES> db.sales.aggregate([
...   {
...     $match: {
...       "store.city": { $regex: /^A/i }
...     }
...   },
...   {
...     $group: {
...       _id: "$store.id",
...       name: { $first: "$store.name" },
...       totalSales: { $sum: "$amount" }
...     }
...   },
...   {
...     $project: {
...       _id: 0,
...       storeId: "$_id",
...       name: 1,
...       totalSales: 1
...     }
...   }
... ])

```

15. Write a MongoDB query to find the store Id, name, and total sales for those stores which contain 'I' as the last letter for the name of the city.

```

SALES> db.sales.updateOne( { "store.id": 6 }, { $set: { "store.name": "Sundari" } } )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
... {
...   $match: {
...     "store.city": { $regex: /I$/i }
...   },
... },
... {
...   $group: {
...     _id: "$store.id",
...     name: { $first: "$store.name" },
...     totalSales: { $sum: "$amount" }
...   }
... },
... {
...   $project: {
...     _id: 0,
...     storeId: "$_id",
...     name: 1,
...     totalSales: 1
...   }
... }
... ] )

```

16. Write a MongoDB query to know whether all the addresses contain the pin code or not.


```
SALES> db.sales.aggregate([
...   {
...     $match: {
...       "store.customer_rating": { $exists: true }
...     }
...   },
...   {
...     $project: {
...       storeId: "$store.id",
...       name: "$store.name",
...       customerRating: "$store.customer_rating"
...     }
...   },
...   {
...     $match: {
...       $expr: {
...         $eq: [{ $mod: ["$customerRating", 5] }, 0]
...       }
...     }
...   }
... ])
```