

7a. Find all listings with listing\_url, name, address, host\_picture\_url in the listings And Reviews collection that have a host with a picture url.

use vacationRentals

```
db.listingsAndReviews.insertMany([
  {
    listing_url: "http://www.example.com/listing/123456",
    name: "Beautiful Apartment",
    address: {
      street: "123 Main Street",
      suburb: "Central",
      city: "Metropolis",
      country: "Wonderland"
    },
    host: {
      name: "Alice",
      picture_url: "http://www.example.com/images/host/host123.jpg"
    }
  },
  {
    listing_url: "http://www.example.com/listing/654321",
    name: "Cozy Cottage",
    address: {
      street: "456 Another St",
      suburb: "North",
      city: "Smallville",
      country: "Wonderland"
    },
    host: {
      name: "Bob",
      picture_url: ""
    }
  },
  {
    listing_url: "http://www.example.com/listing/789012",
    name: "Modern Condo",
    address: {
      street: "789 Side Road",
      suburb: "East",
      city: "Gotham",
      country: "Wonderland"
    },
  },
])
```

```

    host: {
      name: "Charlie",
      picture_url: "http://www.example.com/images/host/host789.jpg"
    }
  }
])

```

### Query to Find Listings with Host Picture URLs:

Note: Now that the collection is set up, you can run the query to find all listings with listing\_url, name, address, and host\_picture\_url where the host has a picture URL.

```

db.listingsAndReviews.find(
  {
    "host.picture_url": { $exists: true, $ne: "" }
  },
  {
    listing_url: 1,
    name: 1,
    address: 1,
    "host.picture_url": 1
  }
).pretty()

```

### OUTPUT:

```

[
  {
    _id: ObjectId('666c40ce85a7615d27cdcdfb'),
    listing_url: 'http://www.example.com/listing/123456',
    name: 'Beautiful Apartment',
    address: {
      street: '123 Main Street',
      suburb: 'Central',
      city: 'Metropolis',
      country: 'Wonderland'
    },
    host: { picture_url: 'http://www.example.com/images/host/host123.jpg' }
  },
  {

```

```

    _id: ObjectId('666c40ce85a7615d27cdcdfd'),
    listing_url: 'http://www.example.com/listing/789012',
    name: 'Modern Condo',
    address: {
      street: '789 Side Road',
      suburb: 'East',
      city: 'Gotham',
      country: 'Wonderland'
    },
    host: { picture_url: 'http://www.example.com/images/host/host789.jpg' }
  }
]

```

7b. Using E-commerce collection write a query to display reviews summary.

use ecommerce

```

db.products.insertMany([
  {
    product_id: 1,
    name: "Laptop",
    category: "Electronics",
    price: 1200,
    reviews: [
      { user: "Alice", rating: 5, comment: "Excellent!" },
      { user: "Bob", rating: 4, comment: "Very good" },
      { user: "Charlie", rating: 3, comment: "Average" }
    ]
  },
  {
    product_id: 2,
    name: "Smartphone",
    category: "Electronics",
    price: 800,
    reviews: [
      { user: "Dave", rating: 4, comment: "Good phone" },
      { user: "Eve", rating: 2, comment: "Not satisfied" },
      { user: "Frank", rating: 5, comment: "Amazing!" }
    ]
  }
])

```

```

},
{
  product_id: 3,
  name: "Headphones",
  category: "Accessories",
  price: 150,
  reviews: [
    { user: "Grace", rating: 5, comment: "Great sound" },
    { user: "Heidi", rating: 3, comment: "Okay" }
  ]
}
])

```

### Query to Find E-commerce collection:

Note: To display a summary of reviews in an e-commerce collection, we can assume the ecommerce database contains a products collection with documents structured to include reviews. Each product document could have a reviews array with review details such as rating, comment, and user.

```

db.products.aggregate([
  {
    $unwind: "$reviews"
  },
  {
    $group: {
      _id: "$name",
      totalReviews: { $sum: 1 },
      averageRating: { $avg: "$reviews.rating" },
      comments: { $push: "$reviews.comment" }
    }
  },
  {
    $project: {
      _id: 0,
      product: "$_id",
      totalReviews: 1,
      averageRating: 1,

```

```
        comments: 1
    }
}
]).pretty()
```

## OUTPUT:

```
[
  {
    totalReviews: 3,
    averageRating: 4,
    comments: [ 'Excellent!', 'Very good', 'Average' ],
    product: 'Laptop'
  },
  {
    totalReviews: 3,
    averageRating: 3.6666666666666665,
    comments: [ 'Good phone', 'Not satisfied', 'Amazing!' ],
    product: 'Smartphone'
  },
  {
    totalReviews: 2,
    averageRating: 4,
    comments: [ 'Great sound', 'Okay' ],
    product: 'Headphones'
  }
]
```