1. Find the price per night of the first record in the listingsAndReviews collection.

```
 + ListingandReview> db.lsr.findOne({},{price: Decimal128('317.00')})
```

2. Retrieve the cleaning fee of the first record in the listingsAndReviews collection

```
 + db.lsr.findOne(
  {}, { "cleaning_fee": 1, _id: 0 } )
```

3. Find the host_name, host_location, host_about of the first record in the listingsAndReviews collection.

```
 + ListingandReview> db.lsr.aggregate([ { $limit: 1 }, { $project: {"host.host_name":1,"host.host_location":-
1,"host.host_about":1 } }] )
```

4. Retrieve the number of bedrooms in the first record in the listingsAndReviews collection.

```
 + ListingandReview> db.lsr.aggregate([
... {$limit:1},
... {$project:{bedrooms:1}}
... ])
```

5. Retrieve the number of guests are included in the first record in the listingsAndReviews collection.

```
 + ListingandReview> db.lsr.aggregate([
... {$limit:1},
... {$project:{guests_included:1}}
... ])
```

6. Write a MongoDB query to check whether the host have a profile picture in the first record in the listingsAndReviews collection.

```
 + ListingandReview> db.lsr.findOne({},{"host.host_has_profile_pic":1,_id:0})
```

7. Write a MongoDB query to check whether the host's identity have been verified in the first record in the listingsAndReviews collection.

```
 + ListingandReview> db.lsr.findOne({},{"host.host_identity_verified":1,_id:0})
```

8. Write a MongoDB query to find how many listings does the host have in the first records in the listingsAndReviews collection.

```
 + ListingandReview> db.lsr.findOne(
  {},{ "host.host_listings_count": 1, _id: 0 } )
```

9. Write a MongoDB query to find the street address of the first record in the listingsAndReviews collection.

```
 + ListingandReview> db.lsr.findOne({},{"address.street":1})
```

10. Find all the listings in the listingsAndReviews collection where the property_type field is set to "House".

```
 + ListingandReview> db.lsr.findOne({property_type:"House"})
```

11. Find all the listings in the listingsAndReviews collection with listing_url, name, host_name, host_location, reviewer_name and price that have a nightly price greater than $500.

```
 + db.lsr.find(
  { price: { $gt: 500 } },  // Filter for nightly price greater than 500
  { listing_url: 1, name: 1, host_name: 1, host_location: 1, reviewer_name: 1, price: 1, _id: 0 })
```

12. Find all the listings in the listingsAndReviews collection that are located in Brazil and have a review score rating of at least 9. Return name, address, and review_scores_rating.

```
 + ListingandReview> db.lsr.find( { "address.country": "Brazil", "review_scores.review_scores_rating": { $gte:
9 } }, { name: 1, address: 1, "review_scores.review_scores_rating": 1, _id: 0 } )
```

13. Find all the listings with name, address, reviewer_name, and review_scores_rating in the listingsAndReviews collection that have a "hot tub" amenity and are located in the United States.

```
 + ListingandReview> db.lsr.find({},{country:"United States",amenities:['hot tub']})
```

14. Find all the listings with name, amenities and price in the listingsAndReviews collection that have a "pool" amenity and a nightly price between $200 and $400.

```
 + db.lsr.find( { amenities: "Pool", price: { $gte: 200, $lte: 400 } }, { name: 1, amenities: 1, price: 1,
_id: 0 } )
```

15. Find all the listings with name, amenities and address in the listingsAndReviews collection that have a "Washer" amenity and are located in either Canada or Mexico.

```
 +  db.lsr.find( { amenities: "Washer", "address.country": { $in: ["Canada", "Mexico"] } },{name:1,address:-
1,amenities:1})
```

16. Find the top 10 most reviewed listings with listing_url, name, country, review_scoresin the listingsAndReviews collection.

```
 + ListingandReview> db.lsr.aggregate([
... {$sort:{"review_scores.review_scores_rating": -1}},
... {$limit:10},
... {$project:{listing_url:1,name:1,country:1,"review_scores_rating":1}}
... ])
```

17. Find all the listings with listing_url, name, address and review_scores in the listingsAndReviews collection that have a "fireplace" amenity and a review score rating of at least 8.

```
 + skip
```

18. Find all the listings with listing_url, name, address and amenities, review scores in the listingsAndReviews collection that have a "washer" amenity and are located in either Italy or Spain.

```
 + ListingandReview> db.lsr.find( { amenities: "Washer", "address.country": { $in: ["Italy", "Spain"] } },
{ name: 1, address: 1, listing_url: 1, amenities: 1, review_scores: 1 } )
```

19. Find the listings with listing_url, name, address and amenities, price, review scores in the listingsAndReviews collection that have the highest nightly prices.

```
  + ListingandReview> db.lsr.find( {}).sort({ price: -1 }).limit(1)
```

20. Find the listings with listing_url, name, address and amenities,price,review scores in the listingsAndReviews collection that have the lowest nightly prices.

```
 + ListingandReview> db.lsr.find( {}).sort({ price: 1 }).limit(1)
```

21. Retrieve all documents with name, address, reviewer_name, review_scores_ratingin the listingsAndReviewscollection that have a number_of_reviews field is equal to 0.

```
 + db.lsr.find({number_of_reviews:0},{name:1,address:1,"reviews.reviewer.name":-
1,"review_scores.review_scores_rating":1,_id:0})
```

22. Retrieve all documents with name, address, host, reviewer_name, review_scores_ratingin the listingsAndReviews collection where the host_is_superhost field is equal to true.

```
 + db.lsr.find(
  { "host.host_is_superhost": true },
  { name: 1, address: 1, "host.host_is_superhost": 1, "reviews.reviewer_name": 1,
"review_scores.review_scores_rating": 1 })
```

23. Retrieve all documents with name, address, host, reviewer_name, review_scores_ratingin the listingsAndReviews collection where the coordinates field is not null.

```
 + ListingandReview> db.lsr.find({"address.location.coordinates":{$ne:null}},{"address.location.coordinates":-1,_id:0})
```

24. Retrieve all documents with name, address, host, bed_type, bed, review_scores_ratingfrom the listingsAndReviewscollection where the beds field is greater than or equal to 2.

```
 + ListingandReview> db.lsr.find({beds:{$gte:2}},{name:1,address:1,host:1,bed_type:1,beds:-1,review_scores_rating:1})
```

25. Find all listings with name, address, host in the listingsAndReviews collection that have a host with a host_name containing the word "Livia".

```
 + ListingandReview> db.lsr.find({"host.host_name":/Livia/},{_id:1,"host.host_name":1,name:1,address:1})
```

26. Find all listings with name, address, host in the listingsAndReviews collection that have a host with a host_location of "Brazil".

```
 + ListingandReview> db.lsr.find({"host.host_location":"Brazil"},{name:1,address:1,"host.host_location":1})
```

27. Retrieve all documents with name, address, host, availability in the listingsAndReviews collection where the availability_365 field is greater than 300.

```
 + ListingandReview> db.lsr.find( { "availability.availability_365": { $gt: 300 } }, { name: 1, address: 1, host: 1, availability: 1 })
```

28. Retrieve all documents with listing_url, name, bedrooms, pricein the listingsAndReviews collectionwhere the bedrooms field is equal to 1.

```
 + ListingandReview> db.lsr.find({"bedrooms":1},{listing_url:1,name:1,bedrooms:1,price:1})
```

29. Retrieve all documents with listing_url, name, bedrooms, cleaning_fee, and price in the listingsAndReviews collectionwhere the cleaning_fee field is not null.

```
 + db.lsr.find(
  { "cleaning_fee": { $ne: null } },
  { listing_url: 1, name: 1, bedrooms: 1, cleaning_fee: 1, price: 1,_id: 0 }
)
```

30. Retrieve all documents with listing_url, name, bedrooms, pricein the listingsAndReviews collection where the price field is between 600 and 900.

```
 + ListingandReview> db.lsr.find({price:{$gte:600,$lte:900}},{listing_url:1,name:1,bedrooms:1,price:1})
```

31. Retrieve all documents with listing_url, name, host, price in the listingsAndReviews collection where the host_verifications array contains "email".

```
 + ListingandReview> db.lsr.find({"host.host_verifications":"email"},{listing_url:1,name:1,host:1,price:1})
```

32. Retrieve all documents with listing_url, name, amenity, host in the listingsAndReviews collection where the amenities array contains both "TV" and "Wifi".

```
 + ListingandReview> db.lsr.find({$and:[{amenities:"TV"},{amenities:"Wifi"}]},{listing_url:1,name:1,amenities:-1,host:1})
```

33. Find all listings with listing_url, name, amenities, host in the listingsAndReviewscollection that have a host with a Jumio verification and a about section.

```
 + ListingandReview> db.lsr.find({$and:[{"host.host_verifications":{$in:["jumio"]}},{"host.host_about":-
{$ne:""}}]},{listing_url:1, name:1, amenities:1, host:1})
```

34. Retrieve all documents with listing_url, name, host, price in the listingsAndReviews collection where the host_total_listings_count field is greater than 1.

```
 + ListingandReview>  db.lsr.find({ "host.host_total_listings_count": { $gt: 1 } }, { listing_url: 1, name: 1,
host: 1, price: 1 })
```

35. Retrieve all documents with listing_url, name, property_type, bed, price in the listingsAndReviewscollectionwhere the property_type field is equal to "Apartment" and the beds field is greater than or equal to 2.

```
 + db.lsr.find(
  { property_type: "Apartment",  beds: { $gte: 2 }},
  { listing_url: 1, name: 1, property_type: 1, beds: 1, price: 1 })
```

36. Find all listings with listing_url, name, property_type, bed, bathrooms, price in the listingsAndReviews collection that have a minimum of 2 bathrooms.

```
 + ListingandReview> db.lsr.find(
  { bathrooms: { $gte: 2 } },
  { listing_url: 1, name: 1, property_type: 1, beds: 1, bathrooms: 1, price: 1
  }
)
```

37. Find all listings with listing_url, name, property_type, bed, price, guests_included in the listingsAndReviews collection that have a maximum of 5 guests included in the price.

```
 + ListingandReview> db.lsr.find(
  { guests_included: { $lte: 5 }  },
  { listing_url: 1, name: 1,property_type: 1, beds: 1, price: 1, guests_included: 1 }
)
```

38. Find all listings with listing_url, name, property_type, bed, price, security_deposit in the listingsAndReviews collection that have a price greater than $500 and a security deposit of $1000 or more.

```
 + ListingandReview> db.lsr.find({$and:[{price:{$gt:500}},{security_deposit:{$gte:1000}}]}, {listing_url: 1,
name: 1,property_type: 1, beds: 1, price: 1,cancellation_policy:1})
```

39. Find all listings with listing_url, name, property_type, bed, price, cancellation_policy in the listingsAndReviews collection that have a cancellation policy of "flexible".

```
 + ListingandReview> db.lsr.find( {cancellation_policy:"flexible"}, {listing_url: 1, name: 1,property_type: 1,
beds: 1, price: 1,cancellation_policy:1})
```

40. Find all listings with listing_url, name, property_type, bed_type, amenities, price in the listingsAndReviews collection that have a real bed as the bed type and a kitchen amenity.

```
 + ListingandReview> db.lsr.find(
  { bed_type: "Real Bed",amenities: { $in: ["Kitchen"] } },
  { listing_url: 1, name: 1, property_type: 1, bed_type: 1, amenities: 1, price: 1 }
)
```

41. Find all listings with listing_url, name, address, amenities in the listingsAndReviews collection that have a 24-hour check-in amenity and are located in Brazil.

```
 + ListingandReview> db.lsr.find({"amenities":"24-hour check-in","address.country":"Brazil"},{listing_url:-
1,name:1,address:1,amenities:1,_id:0})
```

42. Find all listings with listing_url, name, address, reviews in the listingsAndReviews collection that have at least one review.

```
 + ListingandReview> db.lsr.find({"reviews.comments":{$ne:""}},{listing_url:1,name:1,address:1,reviews:1})
```

43. Find the number of documents that have a blank medium picture url in the istingsAndReviews collection.

```
 + ListingandReview> db.lsr.find({"images.medium_url":""}).count()
```

44. Find all listings with listing_url, name, address, availability_30 in the istingsAndReviews collection that have an availability of at least 30 days.

```
 + ListingandReview> db.lsr.find( { "availability.availability_30": { $gte: 30 } }, { listing_url: 1, name: 1,
address: 1,"availability. availability_30": 1 })
```

45. Find all listings with listing_url, name, address in the listingsAndReviews collection that have a suburb of "Lagoa".

```
 + db.lsr.find(
  { "address.suburb": "Lagoa" },
  { listing_url: 1,  name: 1, address: 1 }
)
```

46. Find all listings with listing_url, name, address, host in the listingsAndReviews collection that have a host who is a superhost and has at least 2 listings.

```
 + ListingandReview> db.lsr.find({"host.host_is_superhost":true,"host.host_listings_count":{$gte:2}},-
{listing_url:1, name:1, address:1, host:1})
```

47. Find all listings with listing_url, name, address, host in the listingsAndReviews collection that have a host who has a profile pic and has been identity verified.

```
 + ListingandReview> db.lsr.find({ $and:[{ "host.host_has_profile_pic": true },
{ "host.host_identity_verified": true }]}, { listing_url: 1, name: 1, address: 1, host: 1 })
```

48. Write a mongodb query to find the listing_url, name, address, host_verifications, and size of host_verification under the host subdocument in the listingsAndReviews collection.

```
 + ListingandReview> db.lsr.aggregate([{$project:{_id:0,size:{$size : "$host.host_verifications"},_id:-
0,listing_url:1}}])
```

49. Find all listings with listing_url, name, address, host_verificationand size of host_verification array in the listingsAndReviews collection that have a host with at least 3 verifications.

```
 + ListingandReview> db.lsr.aggregate([{$project:{_id:0,size:{$size : "$host.host_verifications"},_id:-
0,listing_url:1,name:1,address:1,"host.host_verifications":1}},{$match:{size:{$gte:3}}}])
```

50. Find all listings with listing_url, name, address, host_picture_url in the listingsAndReviews collection that have a host with a picture url.

```
 + ListingandReview> db.lsr.find({ "host.picture_url": { $ne: "" } },{"listing_url": 1,"name": 1,"address":
1,"host.picture_url": 1, "_id": 0})
```