

MongoDB Compass - localhost:27017/mongodbproject.room

Connect Edit View Collection Help

localhost:27017

Documents  
mongodbproject...

My Queries

Databases

Search

- admin
- config
- local
- mongodbproject
  - booking
  - guests
  - hotel
  - room**
- mydb

### mongodbproject.room

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' }

EXPLAIN RESET FIND OPTIMIZE

1 - 12 of 12

```
type: "double"
price: 120

_id: ObjectId('64daac346f2f7c717baff7d6')
roomNo: "R10"
hotelNo: "H4"
type: "family"
price: 140

_id: ObjectId('64daac686f2f7c717baff7d7')
roomNo: "R9"
hotelNo: "H4"
type: "double"
price: 120

_id: ObjectId('64daac686f2f7c717baff7d8')
roomNo: "R10"
hotelNo: "H4"
type: "family"
price: 140
```

My Queries

Databases

Search

- admin
- config
- local
- mongodbproject
  - booking
  - guests
  - hotel**
  - room
- mydb

### mongodbproject.hotel

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' }

ADD DATA EXPORT DATA

```
_id: ObjectId('64da5a6e5f8a92b4a723d7b8')
hotelNo: 3
hotelName: "Marriott"
city: "Brighton"

_id: ObjectId('64da80df5f8a92b4a723d7bc')
hotelNo: "H1"
hotelName: "Hilton"
city: "London"

_id: ObjectId('64daa9eb6f2f7c717baff7d4')
hotelNo: "H4"
hotelName: "Grosvenor Hotel"
city: "Grosvenor"
```

Search

admin

config

local

mongodbproject

booking

guests

hotel

room

mydb

Documents

Aggregations

Schema

Indexes

Validation

Filter

Type a query: { field: 'value' }

Explain

Reset

Find

ADD DATA

EXPORT DATA

1 - 4 of 4

\_id: ObjectId('64da5ef65f8a92b4a723d7b9')

guestNo: 1

guestName: "Frank"

guestAddress: "110 Maple Des Moines, IA."

\_id: ObjectId('64da5f275f8a92b4a723d7ba')

guestNo: 2

guestName: "Sue"

guestAddress: "800 Oak St. Lois, MO."

\_id: ObjectId('64da5f465f8a92b4a723d7bb')

guestNo: 3

guestName: "Mark"

guestAddress: "666 Main Fairfield, Ia."

\_id: ObjectId('64dab27c8c618c8f2e57e102')

guestNo: "G6"

guestName: "John Doe"

guestAddress: "123 New Street"

My Queries

Databases

Search

admin

config

local

mongodbproject

booking

guests

hotel

room

mydb

mongodbproject.booking

Documents

Aggregations

Schema

Indexes

Validation

Filter

Type a query: { field: 'value' }

Explain

ADD DATA

EXPORT DATA

1 -

\_id: ObjectId('64d5ab677dc338dae67fd7fd')

hotelNo: "H1"

guestNo: "G1"

dateFrom: "2022-02-15T00:00:00Z"

dateTo: "2022-02-20T00:00:00Z"

roomNo: "R1"

\_id: ObjectId('64d5ab677dc338dae67fd7fe')

hotelNo: "H2"

guestNo: "G2"

dateFrom: "2022-03-01T00:00:00Z"

dateTo: "2022-03-05T00:00:00Z"

roomNo: "R3"

\_id: ObjectId('64d5ab677dc338dae67fd7ff')

hotelNo: "H2"

guestNo: "G3"

dateFrom: "2022-03-10T00:00:00Z"

dateTo: "2022-03-15T00:00:00Z"

roomNo: "R4"

/\*6.7 List full details of all hotels.\*/

```
mongodbproject> db.hotel.find({});
[
  {
    _id: ObjectId("64da5a6e5f8a92b4a723d7b8"),
    hotelNO: 3,
    hotelName: 'Marriott',
    city: 'Brighton'
  },
  {
    _id: ObjectId("64da80df5f8a92b4a723d7bc"),
    hotelNo: 'H1',
    hotelName: 'Hilton',
    city: 'London'
  }
]
mongodbproject>
```

/\* 6.8 List full details of all hotels in London.\*/

```
mongodbproject> db.hotel.find({ city: 'London' });
[
  {
    _id: ObjectId("64da80df5f8a92b4a723d7bc"),
    hotelNo: 'H1',
    hotelName: 'Hilton',
    city: 'London'
  }
]
mongodbproject>
```

**/\* 6.9 List the names and addresses of all guests living in London, alphabetically ordered by name.\*/**

```
mongodbproject> db.guests.find(
...   { guestAddress: "London" },
...   { guestName: 1, guestAddress: 1, _id: 0 }
... ).sort({ guestName: 1 }).forEach(printjson);

mongodbproject> db.guests.find(
...   { guestAddress: "110 Maple Des Moines, IA." },
...   { guestName: 1, guestAddress: 1, _id: 0 }
... ).sort({ guestName: 1 });
[ { guestName: 'Frank', guestAddress: '110 Maple Des Moines, IA.' } ]
mongodbproject>
```

**/\*6.10 List all double or family rooms with a price below £40.00 per night, in ascending order of price.\*/**

```
mongodbproject> db.room.find( { $and: [{ type: { $in: ['double', 'family'] } }, { price: { $lt: 40.00 } } ] }).sort({ price: 1 });
[
  {
    _id: ObjectId("64da84b75f8a92b4a723d7bd"),
    roomNo: 'R7',
    hotelNo: 'H4',
    type: 'double',
    price: 30
  },
  {
    _id: ObjectId("64da84b75f8a92b4a723d7be"),
    roomNo: 'R8',
    hotelNo: 'H5',
    type: 'family',
    price: 35
  }
]
mongodbproject>
```

**/\* 6.11 List the bookings for which no dateTo has been specified.\*/**

```
mongodbproject> db.booking.find({ dateTo: null });
[
  {
    _id: ObjectId("64da859d5f8a92b4a723d7bf"),
    hotelNo: 'H1',
    guestNo: 'G1',
    dateFrom: ISODate("2022-02-15T00:00:00.000Z"),
    dateTo: null,
    roomNo: 'R1'
  }
]
mongodbproject>
```

**/\*6.12 How many hotels are there?\*/**

```
mongodbproject> db.hotel.aggregate([
...   {
...     $group: {
...       _id: null,
...       num_hotels: { $sum: 1 }
...     }
...   },
...   {
...     $project: {
...       _id: 0,
...       num_hotels: 1
...     }
...   }
... ]);
[ { num_hotels: 2 } ]
mongodbproject>
```

**/\*6.13 What is the average price of a room?\*/**

```
mongodbproject> db.room.aggregate([
...   {
...     $group: {
...       _id: null,
...       average_price: { $avg: "$price" }
...     }
...   },
...   {
...     $project: {
...       _id: 0,
...       average_price: 1
...     }
...   }
... ]);
[ { average_price: 125.625 } ]
```

**/\*6.14 What is the total revenue per night from all double rooms?\*/**

```

mongodbproject> db.room.aggregate([
...   {
...     $match: { type: "double" }
...   },
...   {
...     $group: {
...       _id: null,
...       total_revenue: { $sum: "$price" }
...     }
...   },
...   {
...     $project: {
...       _id: 0,
...       total_revenue: 1
...     }
...   }
... ]);
[ { total_revenue: 30 } ]
mongodbproject>

```

**/\*6.15 How many different guests have made bookings for August?\*/**

```

mongodbproject> const augustBookings = db.booking.aggregate([
...   {
...     $match: {
...       dateFrom: {
...         $gte: new Date("2023-08-01T00:00:00Z"),
...         $lt: new Date("2023-09-01T00:00:00Z")
...       }
...     }
...   },
...   {
...     $group: {
...       _id: "$guestNo"
...     }
...   }
... ]);

mongodbproject>

mongodbproject> const numAugustGuests = augustBookings.toArray().length;

mongodbproject>

mongodbproject> print("Number of different guests who made bookings for August:", numAugustGuests);
Number of different guests who made bookings for August: 0

mongodbproject>

```

**/\*6.16 List the price and type of all rooms at the Grosvenor Hotel.\*/**

```

mongodbproject> const hotelInfo = db.hotel.findOne({ hotelName: 'Grosvenor Hotel' });
;

    if (grosvenorRooms.length > 0) {
        print("Price and type of rooms at the Grosvenor Hotel:");
    }

mongodbproject>

mongodbproject> if (hotelInfo) {
...   const hotelNoGrosvenor = hotelInfo.hotelNo;
...
...   const grosvenorRooms = db.room.find(
...     { hotelNo: hotelNoGrosvenor },
...     { type: 1, price: 1, _id: 0 }
...   ).toArray()
...   if (grosvenorRooms.length > 0) {
...     grosvenorRooms.forEach(room => {
...       print(`Type: ${room.type}, Price: ${room.price}`);
...     });
...   } else {
...     print("No rooms found for the Grosvenor Hotel.");
...   }
... } else {
...   print("Hotel 'Grosvenor Hotel' not found.");
... }
Type: double, Price: 30

mongodbproject>

```

**/\*6.17 List all guests currently staying at the Grosvenor Hotel\*/**

```

mongodbproject> const guestsAtGrosvenor = db.booking.aggregate([
...   {
...     $match: {
...       hotelNo: "H4", // Assuming "H4" is the hotelNo for the Grosvenor Hotel
...       dateFrom: { $lte: currentDate },
...       dateTo: { $gt: currentDate }
...     } },
...   {
...     $lookup: {
...       from: "Guests",
...       localField: "guestNo",
...       foreignField: "guestNo",
...       as: "guestInfo"
...     }
...   },
...   {
...     $unwind: "$guestInfo"
...   },
...   {
...     $project: {
...       guestName: "$guestInfo.guestName",
...       guestAddress: "$guestInfo.guestAddress"
...     }
...   }
... ]);

mongodbproject>

mongodbproject> if (guestsAtGrosvenor.hasNext()) {
...   print("Guests currently staying at the Grosvenor Hotel:");
...   guestsAtGrosvenor.forEach(guest => {
...     print(`Name: ${guest.guestName}, Address: ${guest.guestAddress}`);
...   });
... } else {
...   print("No guests currently staying at the Grosvenor Hotel.");
... }
No guests currently staying at the Grosvenor Hotel.

```

*/\* 6.18 List the details of all rooms at the Grosvenor Hotel, including the name of the guest staying in the room, if the room is occupied.\*/*



```

> // 6.18: List the details of all rooms at the Grosvenor Hotel, including the name of the guest staying in the room, if the room is booked
// Query to list the details of all rooms at the Grosvenor Hotel, including the name of the guest staying in the room, if the room is booked
db.bookings.find({
  "hotel.name": "Grosvenor Hotel",
  date_from: { $lte: new Date() },
  date_to: { $gte: new Date() }
}, { "room.type": 1, "room.price": 1, "guest.name": 1 }).pretty();

```

```

< {
  _id: ObjectId("64d6b3bd976ad998579edf94"),
  guest: {
    name: 'Jane Smith'
  },
  room: {
    type: 'single',
    price: 100
  }
}

```

**/\* 6.19 What is the total income from bookings for the Grosvenor Hotel today?\*/**

```

test> use mongodbproject
switched to db mongodbproject
mongodbproject> const currentDate = new Date();
;

mongodbproject>

mongodbproject> const totalIncome = db.booking.aggregate([
...   {
...     $match: {
...       hotelNo: "H4",
...       dateFrom: { $lte: currentDate },
...       dateTo: { $gt: currentDate }
...     }
...   },
...   {
...     $lookup: {
...       from: "room",
...       localField: "roomNo",
...       foreignField: "roomNo",
...       as: "roomInfo"
...     }
...   },
...   {
...     $unwind: "$roomInfo"
...   },
...   {
...     $project: {
...       income: "$roomInfo.price"
...     }
...   },
...   {
...     $group: {
...       _id: null,
...       totalIncome: { $sum: "$income" }
...     }
...   }
... ])

mongodbproject> const totalIncomeResult = totalIncome.toArray();

mongodbproject>

mongodbproject> if (totalIncomeResult.length > 0) {
...   const result = totalIncomeResult[0];
...   print(`Total income from bookings for the Grosvenor Hotel today: ${result.totalIncome}`);
... } else {
...   print("No income from bookings for the Grosvenor Hotel today.");
... }
No income from bookings for the Grosvenor Hotel today.

mongodbproject>

```

/\*6.22 List the number of rooms in each hotel\*/

```

mongodbproject> const roomsPerHotel = db.room.aggregate([
...   {
...     $group: {
...       _id: "$hotelNo",
...       numberOfRooms: { $sum: 1 }
...     }
...   },
...   {
...     $lookup: {
...       from: "hotels",
...       localField: "_id",
...       foreignField: "hotelNo",
...       as: "hotelInfo"
...     }
...   }, {
...     $project: {
...       hotelName: { $arrayElemAt: ["$hotelInfo.hotelName", 0] },
...       numberOfRooms: 1
...     }
...   }
... ]);

mongodbproject>

mongodbproject> print("Number of rooms in each hotel:");
Number of rooms in each hotel:

mongodbproject> roomsPerHotel.forEach(hotel => {
...   print(`Hotel: ${hotel.hotelName}, Number of Rooms: ${hotel.numberOfRooms}`);
... });
Hotel: undefined, Number of Rooms: 2
Hotel: undefined, Number of Rooms: 5
Hotel: undefined, Number of Rooms: 1
Hotel: undefined, Number of Rooms: 2
Hotel: undefined, Number of Rooms: 2

mongodbproject>

```

*/\*6.23 List the number of rooms in each hotel in London\*/*

```

// 6.23: List the number of rooms in each hotel in London.
// Query to list the number of rooms in each hotel in London.
db.rooms.aggregate([
  {
    $lookup: {
      from: "hotels",
      localField: "hotel_id",
      foreignField: "_id",
      as: "hotel"
    }
  },
  {
    $unwind: "$hotel"
  },
  {
    $match: {
      "hotel.city": "London"
    }
  },
  {
    $group: {
      _id: "$hotel._id",
      hotelName: { $first: "$hotel.name" },
      roomCount: { $sum: 1 }
    }
  }
]).pretty();
{
  _id: ObjectId("64d3b17dc5ac2dee471560d3"),
  hotelName: 'Grosvenor Hotel',
  roomCount: 2
}

```

*/\*6.24 What is the average number of bookings for each hotel in August?\*/*

```
// 6.24: What is the average number of bookings for each hotel in August?
// Query to calculate the average number of bookings for each hotel in August.
db.bookings.aggregate([
  {
    $match: {
      date_from: {
        $gte: ISODate("2023-08-01"),
        $lte: ISODate("2023-08-31")
      }
    }
  },
  {
    $group: {
      _id: "$hotel._id",
      hotelName: { $first: "$hotel.name" },
      bookingsCount: { $sum: 1 }
    }
  },
  {
    $lookup: {
      from: "rooms",
      localField: "_id",
      foreignField: "hotel_id",
      as: "rooms"
    }
  },
  {
    $unwind: "$rooms"
  },
  {
```

```
    $group: {
      _id: "$_id",
      hotelName: { $first: "$hotelName" },
      bookingsCount: { $first: "$bookingsCount" },
      totalRooms: { $sum: 1 }
    }
  },
  {
    $project: {
      hotelName: 1,
      averageBookings: { $divide: ["$bookingsCount", "$totalRooms"] }
    }
  }
]).pretty();
< {
```

```
< {
  _id: ObjectId("64d3b17dc5ac2dee471560d3"),
  hotelName: 'Grosvenor Hotel',
  averageBookings: 1
}
{
  _id: null,
  hotelName: null,
  averageBookings: 1.1666666666666667
}
```

**/\*6.27 Insert rows into each of these tables\*/**

```
mongodbproject> db.hotels.insertOne({
...   hotelNo: "H6",
...   hotelName: "New Hotel",
...   city: "New City"
... });
{
  acknowledged: true,
  insertedId: ObjectId("64dab2a88c618c8f2e57e103")
}
mongodbproject>
```

```
mongodbproject> db.room.insertMany([
...   {
...     roomNo: "R11",
...     hotelNo: "H6",
...     type: "Standard",
...     price: 150
...   },
...   {
...     roomNo: "R12",
...     hotelNo: "H6",
...     type: "Suite",
...     price: 250
...   }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64dab23e8c618c8f2e57e100"),
    '1': ObjectId("64dab23e8c618c8f2e57e101")
  }
}
mongodbproject>
```

```

mongodbproject> db.guests.insertOne({
...   guestNo: "G6",
...   guestName: "John Doe",
...   guestAddress: "123 New Street"
... });
{
  acknowledged: true,
  insertedId: ObjectId("64dab27c8c618c8f2e57e102")
}
mongodbproject>
```

```

mongodbproject> db.hotels.insertOne({
...   hotelNo: "H6",
...   hotelName: "New Hotel",
...   city: "New City"
... });
{
  acknowledged: true,
  insertedId: ObjectId("64dab2a88c618c8f2e57e103")
}
mongodbproject> db.booking.insertOne({
...   bookingNo: "B6",
...   hotelNo: "H6",
...   roomNo: "R11",
...   guestNo: "G6",
...   dateFrom: ISODate("2023-08-15"),
...   dateTo: ISODate("2023-08-20")
... });
{
  acknowledged: true,
  insertedId: ObjectId("64dab2cc8c618c8f2e57e104")
}
mongodbproject>

```

/\* 6.28 Update the price of all rooms by 5%.\*/

```

mongodbproject> db.room.updateMany({}, { $mul: { price: 1.05 } });
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 14,
  modifiedCount: 14,
  upsertedCount: 0
}
mongodbproject>

```



