

# QUERIES AND TRANSACTIONS

**Cassandra Chang ST125002:**

## **Transactions:**

### **1. Rental Rate Adjustment**

```
db.space.updateOne(
{
  "id": 2,
  "Owner.id": 3
},
{
  $set: {
    "hourlyRate": 15,
    "halfdayRate": 120,
    "fulldayRate": 220
  },
  $currentDate: { "updatedAt": true }
}
);
```

### **2. Feedback Submission**

```
db.feedback.updateOne(
{
  "customer.id": 20,
  "space.id": 20
},
{
  $set: {
    "review": "Superb place for presentations!"
  },
  $currentDate: { "updatedAt": true }
}
);
```

### **3. User Profile Updates**

```
db.user.updateOne (
{
  {id:5},
  {
    $set: {
      "email": "ethan_wright@example.com",
```

```
"phone": "5612357896",
"preferredRange": "5 - 10"
},
$currentDate: { "updatedAt": true }}
);
```

## Query:

### 1. Feedback Report

```
db.feedback.aggregate([
{
  $group: {
    _id: "$space.location",
    avgRating: { $avg: "$rating" },
    spaceName: { $first: "$space.name" },
    topReview: { $first: "$review" }
  }
},
{
  $project: {
    _id: 0,
    location: "$_id",
    spaceName: 1,
    avgRating: { $round: ["$avgRating", 2]},
    topReview: 1
  }
}
]);
```

### 2. Customer Booking History

```
db.booking.aggregate([
{
  $match: {
    "customer.id": 8
  }
},
{
  $project: {
    _id: 0,
    bookingId: "$id",
    spaceName: "$space.name",
    location: "$space.location",
```

```

    bookingDate: "$bookingDate",
    startTime: "$startFrom",
    endTime: "$endAt",
    totalCharges: "$total"
  }
}
]);

```

### 3. Space Owner Report

```

db.booking.aggregate([
{
  $lookup: {
    from: "space",
    localField: "space.id",
    foreignField: "id",
    as: "spaceDetails"
  }
},
{
  $unwind: "$spaceDetails"
},
{
  $match: {
    "spaceDetails.Owner.id": 5
  }
},
{
  $group: {

    _id: {
      spaceName: "$spaceDetails.name",
      ownerName: "$spaceDetails.Owner.name"
      address: "$spaceDetails.address"
    },
    totalBookings: { $sum: 1},
    totalEarnings: { $sum: "$spacePrice"},
    platformFees: { $sum: "$sysFee" }
  }
},
{
  $project: {
    _id: 0,
    spaceName: "$_id.spaceName",
    ownerName: "$_id.ownerName",

```

```

address: "$_id.address",
totalBookings: 1,
totalEarnings: 1,
platformFees: 1
}
}
]);

```

#### 4. Space Utilization Report

```

db.booking.aggregate([
{
$group: {
_id: {
id: "$space.id",
name: "$space.name",
location: "$space.location"
},
totalBookings: { $sum: 1},
totalHours: {
$sum: {
$divide: [
{ $subtract: ["$endAt", "$startFrom"]},
3600000 // Convert milliseconds to hours
]}]},
{
$project: {
_id: 0,
spaceName: "$_id.name",
location: "$_id.location",
totalBookings: 1,
totalHours: 1,
utilizationRate: {
$multiply: [
{
$divide: [
"$totalHours",
{ $multiply: ["$totalBookings", 24]
}],
100 // Convert to percentage
}] } }
}
}
]);

```

## 5. Cancelled Bookings Report

```
db.booking.aggregate([
{
$match: {
status: "cancelled"
}
},
{
$project: {
_id: 0,
bookingID: "$id",
customerName: "$customer.name"
spaceName: "$space.name",
cancellationDate: "$updatedAt"
}
}
]);
```

---

## Kaung Nyo Lwin ST125066: Transactions:

### 1. User Registration

```
db.user.insertOne({
id: 21,
name: "John Doe",
email: "johndoe1@gmail.com",
phone: "1234567890",
address: "1234 Main St",
preferredRange: "10 - 15",
isCustomer: true,
isOwner: false,
rating: 4.5,
createdAt: ISODate("2020-01-01T00:00:00Z"),
updatedAt: ISODate("2020-01-01T00:00:00Z")
});
```

### 2. Booking Spaces

```
db.booking.insertOne({
id: 1,
customer: {
```

```

    id: 14,
    name: "Natalie Carter"
  },
  space: {
    id: 1,
    name: "Space A",
    location: "Downtown",
    hourlyRate: 10,
    halfdayRate: 100,
    fulldayRate: 180,
    rating: 4.5,
    address: "1234 Main St"
  },
  bookingDate: ISODate("2025-01-01T00:00:00Z"),
  startFrom: ISODate("2025-01-01T10:00:00Z"),
  endAt: ISODate("2025-01-01T12:00:00Z"),
  status: "booked",
  createdAt: ISODate("2025-01-01T01:00:00Z"),
  updatedAt: ISODate("2025-01-01T01:00:00Z"),
  payment: null
});

```

```

db.booking.updateOne(
  { id: 1 }, // Search condition
  [
    {
      $set: {
        spacePrice: {
          $let: {
            vars: {
              hours: {
                $dateDiff: {
                  startDate: "$startFrom",
                  endDate: "$endAt",
                  unit: "hour"
                }
              }
            }
          }
        },
      },
      in: {
        $switch: {
          branches: [
            {
              case: { $lt: [ "$$hours", 12 ] },

```

```

        then: { $multiply: [ "$space.hourlyRate", "$$hours" ] }
      },
      {
        case: { $eq: [ "$$hours", 12 ] },
        then: "$space.halfdayRate"
      },
      {
        case: { $eq: [ "$$hours", 24 ] },
        then: "$space.fulldayRate"
      },
      {
        case: { $and: [ { $gt: [ "$$hours", 12 ] }, { $lt: [ "$$hours", 24 ] } ] },
        then: {
          $add: [
            "$space.halfdayRate",
            { $multiply: [ "$space.hourlyRate", { $subtract: [ "$$hours", 12 ] } ] }
          ]
        }
      },
    ],
    default: 0
  }
}
}
}
},
{
  $set: {
    sysFee: { $multiply: [ "$spacePrice", 0.03 ] },
    total: { $add: [ "$spacePrice", { $multiply: [ "$spacePrice", 0.03 ] } ] },
    updatedAt: new Date() // Updates timestamp
  }
}
],
{ upsert: true } // Inserts if not found
);

```

### 3. Pay the charges

```

db.booking.updateOne(
  { id: 1 },

```

```

{
  $set: {
    status: "rent",
    payment: {
      id: 1,
      amountToPay: 20.6,
      paid: 20.6,
      unpaid: 0,
      status: "paid",
      transacId: "1234567890",
      bank: "Chase",
      createdAt: ISODate("2025-01-01T01:00:00Z"),
      updatedAt: ISODate("2025-01-01T01:00:00Z")
    }
  }
}
);

```

## Query:

### 1. Most Spender Report

```

db.booking.aggregate([
  {
    $addFields: {
      daysOnPlatform: {
        $floor: {
          $divide: [
            { $subtract: [new Date(), "$customer.createdAt"] }, // Subtract current date from
customer.createdAt
            1000 * 60 * 60 * 24 // Convert milliseconds to days
          ]
        }
      }
    }
  },
  {
    $group: {
      _id: "$customer.id", // Group by customer id
      username: { $first: "$customer.name" }, // Get the first customer's name
      totalAmount: { $sum: "$total" }, // Sum of all total amounts
      daysOnPlatform: { $first: "$daysOnPlatform" } // Get the number of days on the platform for
this customer
    }
  }
]);

```



```

    }
  },
  {
    $project: {
      _id: 0,
      username: 1, // Customer name
      daysOnPlatform: 1, // Number of days on the platform
      totalAmount: 1, // Total amount spent
      avgSpendingPerDay: {
        $cond: {
          if: { $gt: ["$daysOnPlatform", 0] },
          then: { $divide: ["$totalAmount", "$daysOnPlatform"] },
          else: 0
        }
      } // Average spending per day, conditionally handle division by 0
    }
  }
];

```

## 2. Monthly Revenue Report

```

db.booking.aggregate([
  {
    $match: {
      // Filter documents within the given date range (start date to end date)
      endDate: {
        $gte: ISODate("2023-01-20T17:00:00.000Z"), // Replace <Start Date> with your start date
        (e.g., "2025-01-01T00:00:00Z")
        $lte: ISODate("2025-12-20T17:00:00.000Z") // Replace <End Date> with your end date
        (e.g., "2025-12-31T23:59:59Z")
      }
    }
  },
  {
    $project: {
      // Extract month and year from bookingDate for grouping
      month: { $month: "$endDate" },
      year: { $year: "$endDate" },
      spacePrice: 1,
      sysFee: 1,
      total: 1
    }
  },

```

```

{
  $group: {
    _id: { month: "$month", year: "$year" }, // Group by year and month
    spacePriceTotal: { $sum: "$spacePrice" }, // Sum of spacePrice for the month
    sysFeeTotal: { $sum: "$sysFee" }, // Sum of sysFee for the month
    totalAmount: { $sum: "$total" } // Sum of total for the month
  }
},
{
  $sort: { "_id.year": 1, "_id.month": 1 } // Sort by year and month in ascending order
},
{
  $project: {
    _id: 0, // Exclude _id from the output
    year: "$_id.year", // Project year
    month: "$_id.month", // Project month
    spacePrice: "$spacePriceTotal", // Project spacePriceTotal as spacePrice
    sysFee: "$sysFeeTotal", // Project sysFeeTotal as sysFee
    total: "$totalAmount" // Project totalAmount as total
  }
}
]);

```

### 3. Popular Day of the Week

```

db.booking.aggregate([
  {
    $match: {
      // Filter documents based on endAt date range (start date to end date)
      endAt: {
        $gte: ISODate("2023-01-20T17:00:00.000Z"), // Replace <Start Date> with your start date
        (e.g., "2025-01-01T00:00:00Z")
        $lte: ISODate("2025-12-20T17:00:00.000Z") // Replace <End Date> with your end date
        (e.g., "2025-12-31T23:59:59Z")
      }
    }
  },
  {
    $project: {
      // Extract day of the week from endAt (1 = Sunday, 2 = Monday, ..., 7 = Saturday)
      dayOfWeek: { $dayOfWeek: "$endAt" },
      spacePrice: 1,
      sysFee: 1,
      total: 1
    }
  }
]);

```

```

    }
  },
  {
    $addFields: {
      // Map numeric dayOfWeek to actual day namesa
      dayOfWeekName: {
        $switch: {
          branches: [
            { case: { $eq: ["$dayOfWeek", 1] }, then: "Sunday" },
            { case: { $eq: ["$dayOfWeek", 2] }, then: "Monday" },
            { case: { $eq: ["$dayOfWeek", 3] }, then: "Tuesday" },
            { case: { $eq: ["$dayOfWeek", 4] }, then: "Wednesday" },
            { case: { $eq: ["$dayOfWeek", 5] }, then: "Thursday" },
            { case: { $eq: ["$dayOfWeek", 6] }, then: "Friday" },
            { case: { $eq: ["$dayOfWeek", 7] }, then: "Saturday" }
          ],
          default: "Unknown"
        }
      }
    }
  },
  {
    $group: {
      _id: "$dayOfWeekName", // Group by the day name
      totalBookings: { $count: {} }, // Count the total number of bookings for each day
      spacePriceTotal: { $sum: "$spacePrice" }, // Sum of spacePrice for the day
      sysFeeTotal: { $sum: "$sysFee" }, // Sum of sysFee for the day
      totalAmount: { $sum: "$total" } // Sum of total for the day
    }
  },
  {
    $sort: { "_id": 1 } // Sort by day of the week in ascending order (Monday first, Sunday last)
  },
  {
    $project: {
      _id: 0, // Exclude _id from the output
      dayOfWeek: "$_id", // Project dayOfWeek as dayOfWeek
      totalBookings: 1, // Project totalBookings
      spacePrice: "$spacePriceTotal", // Project spacePriceTotal as spacePrice
      sysFee: "$sysFeeTotal", // Project sysFeeTotal as sysFee
      total: "$totalAmount" // Project totalAmount as total
    }
  }
]);

```

#### 4. Rental report by price ranges

```
// Define the input variables (you can change these based on your use case)
const startHourlyRate = 0; // Start Hourly Rate
const endHourlyRate = 100; // End Hourly Rate
const step = 5;           // Step for hourly rate ranges

// Generate the ranges dynamically based on the input
let ranges = [];
for (let i = startHourlyRate; i < endHourlyRate; i += step) {
  ranges.push({
    start: i,
    end: i + step
  });
}

db.booking.aggregate([
  {
    $addFields: {
      // Determine the range for each booking based on the space's hourlyRate
      range: {
        $let: {
          vars: {
            hourlyRate: "$space.hourlyRate" // Use the space's hourlyRate
          },
          in: {
            $arrayElemAt: [
              // Find the range the hourlyRate falls into
              {
                $filter: {
                  input: ranges, // Use the ranges array we defined earlier
                  as: "range",
                  cond: {
                    $and: [
                      { $gte: ["$$range.start", "$$hourlyRate"] },
                      { $lt: ["$$hourlyRate", "$$range.end"] }
                    ]
                  }
                }
              ],
              0 // Take the first matching range
            ]
          }
        }
      }
    }
  }
])
```

```

    }
  }
}
},
{
  $group: {
    _id: {
      start: "$range.start", // Group by the start of the range
      end: "$range.end"      // Group by the end of the range
    },
    totalBookings: { $count: {} }, // Count the total number of bookings in this range
    spacePriceTotal: { $sum: "$spacePrice" }, // Sum of spacePrice for this range
    sysFeeTotal: { $sum: "$sysFee" }, // Sum of sysFee for this range
    totalAmount: { $sum: "$total" } // Sum of total for this range
  }
},
{
  $project: {
    _id: 0, // Exclude _id field
    range: { $concat: [{ $toString: "$_id.start" }, " - ", { $toString: "$_id.end" }] }, // Concatenate
start and end to form a range string
    totalBookings: 1,
    spacePrice: "$spacePriceTotal",
    sysFee: "$sysFeeTotal",
    total: "$totalAmount"
  }
},
{
  $sort: { "range": 1 } // Sort the results by range in ascending order
}
]);

```

## 5. Space recommendation query

```

db.space.aggregate([
// Step 1: Left join with booking to get the number of bookings by input user id
{
  $lookup: {
    from: "booking",
    let: { spaceId: "$id" },

```

```

    pipeline: [
      { $match: { "customer.id": 20 } }, // Replace with input user ID
      { $match: { $expr: { $eq: ["$space.id", "$$spaceId"] } } },
      { $group: { _id: "$space.id", numBookings: { $sum: 1 } } }
    ],
    as: "userBookings"
  }
},

```

// Step 2: Left join with booking to get total booking hours in search period

```

{
  $lookup: {
    from: "booking",
    let: { spaceId: "$id" },
    pipeline: [
      {
        $match: {
          startFrom: { $gte: ISODate("2025-01-01T10:00:00Z") }, // Replace with input startTime
          endAt: { $lte: ISODate("2025-01-01T12:00:00Z") }, // Replace with input endTime
        }
      },
      { $match: { $expr: { $eq: ["$space.id", "$$spaceId"] } } },
      {
        $group: {
          _id: "$space.id",
          totalBookedHours: {
            $sum: { $divide: [{ $subtract: ["$endAt", "$startFrom"] }, 3600000] } // Convert ms to
hours
          }
        }
      }
    ],
    as: "bookingHours"
  }
},

```

// Step 3: Unwind lookup results

```

{
  $addFields: {
    numBookings: { $ifNull: [{ $arrayElemAt: ["$userBookings.numBookings", 0] }, 0] },
    totalBookedHours: { $ifNull: [{ $arrayElemAt: ["$bookingHours.totalBookedHours", 0] }, 0] }
  }
},

```

```

// Step 4: Filter out fully booked spaces
{
  $match: {
    $expr: {
      $lt: [
        "$totalBookedHours",
        { $divide: [{ $subtract: [ISODate("2025-01-01T12:00:00Z"),
          ISODate("2025-01-01T10:00:00Z")] }, 3600000] } // Search hours
      ]
    }
  }
},

// Step 5: Filter by location
{
  $match: { location: "Uptown" } // Replace with input location
},

// Step 6: Project required fields
{
  $project: {
    _id: 0,
    spaceName: "$name",
    hourlyRate: 1,
    halfdayRate: 1,
    fulldayRate: 1,
    rating: 1,
    numBookings: 1
  }
},

// Step 7: Sort by number of bookings and rating
{
  $sort: { numBookings: -1, rating: -1 }
},

// Step 8: Limit to first 10 documents
{ $limit: 10 }
]);

```

---

**Truong Vuong st124689:**  
**Transactions:**

### 1. Space Status Update

```
db.space.updateOne(  
  { id: 8 },  
  {  
    $set: {  
      status: "unavailable",  
      remark: "Under maintenance"  
    }  
  }  
)
```

### 2. Space Removal

```
db.space.deleteOne({ id: 10 })
```

### 3. Change Number of Facilities

```
db.space.updateOne(  
  { id: 8 },  
  {  
    $set: { "facilitiy.chair": 15 }  
  }  
)
```

## Queries \$ Reports:

### 1. Customer Retention Report

```
db.booking.aggregate([  
  {  
    "$group": {  
      "_id": "$customer.id",  
      "CustomerName": { "$first": "$customer.name" },  
      "TotalBookings": { "$sum": 1 },  
      "TotalSpend": { "$sum": "$total" },  
      "LastBookingDate": { "$max": "$bookingDate" }  
    }  
  },  
  {
```



```

"$addFields": {
  "RetentionStatus": {
    "$cond": { "if": { "$gte": ["$TotalBookings", 3] }, "then": "Loyal", "else":
"Casual" }
  }
},
{
  "$project": {
    "_id": 0,
    "CustomerName": 1,
    "TotalBookings": 1,
    "TotalSpend": 1,
    "LastBookingDate": 1,
    "RetentionStatus": 1
  }
}
])

```

## 2. Total System Fee

```

db.booking.aggregate([
  {
    "$group": {
      "_id": { "$dateToString": { "format": "%Y-%m", "date": { "$toDate":
"$bookingDate" } } },
      "TotalSystemFee": { "$sum": "$sysFee" },
      "TotalBookings": { "$sum": 1 }
    }
  },
  { "$sort": { "_id": 1 } },
  {
    "$project": {
      "_id": 0,
      "Month": "$_id",
      "TotalSystemFee": 1,

```

```

    "TotalBookings": 1
  }
}
])

```

### 3. Booking Trends by Season Report

```

db.booking.aggregate([
  {
    "$project": {
      "_id": 0,
      "BookingID": "$id",
      "BookingDate": "$bookingDate",
      "TotalOwnerRevenue": { "$subtract": ["$total", "$sysFee"] },
      "Season": {
        "$switch": {
          "branches": [
            { "case": { "$lte": [{ "$month": { "$toDate": "$bookingDate" } }, 3] },
"then": "Winter" },
            { "case": { "$lte": [{ "$month": { "$toDate": "$bookingDate" } }, 6] },
"then": "Spring" },
            { "case": { "$lte": [{ "$month": { "$toDate": "$bookingDate" } }, 9] },
"then": "Summer" }
          ],
          "default": "Fall"
        }
      }
    }
  },
  {
    "$group": {
      "_id": "$Season",
      "TotalBookings": { "$sum": 1 },
      "TotalOwnerRevenue": { "$sum": "$TotalOwnerRevenue" }
    }
  }
]

```

])

#### 4. Customer Feedback Sentiment Report

```
db.feedback.aggregate([
  {
    $lookup: {
      from: "space",
      localField: "space.id",
      foreignField: "id",
      as: "space_info"
    }
  },
  {
    $unwind: "$space_info"
  },
  {
    $group: {
      _id: "$space_info.name",
      SpaceName: { $first: "$space_info.name" },
      Location: { $first: "$space_info.location" },
      TotalReviews: { $sum: 1 },
      PositiveReviews: {
        $sum: { $cond: [{ $gte: ["$rating", 4] }, 1, 0] }
      },
      NeutralReviews: {
        $sum: { $cond: [{ $gte: ["$rating", 3] }, { $cond: [{ $lt: ["$rating", 4] }, 1,
0] }, 0] }
      },
      NegativeReviews: {
        $sum: { $cond: [{ $lt: ["$rating", 3] }, 1, 0] }
      },
      AverageRating: { $avg: "$rating" }
    }
  },
  {
```

```

$project: {
  _id: 0,
  SpaceName: 1,
  Location: 1,
  TotalReviews: 1,
  PositiveSentimentPercentage: {
    $cond: [{ $eq: ["$TotalReviews", 0] }, 0, { $multiply: [{ $divide:
["$PositiveReviews", "$TotalReviews"] }, 100] }]
  },
  NeutralSentimentPercentage: {
    $cond: [{ $eq: ["$TotalReviews", 0] }, 0, { $multiply: [{ $divide:
["$NeutralReviews", "$TotalReviews"] }, 100] }]
  },
  NegativeSentimentPercentage: {
    $cond: [{ $eq: ["$TotalReviews", 0] }, 0, { $multiply: [{ $divide:
["$NegativeReviews", "$TotalReviews"] }, 100] }]
  },
  AverageRating: { $round: ["$AverageRating", 2] }
}
});

```

## 5. Space Demand and Performance Report

```

db.booking.aggregate([
{
  $lookup: {
    from: "space",
    localField: "space.id",
    foreignField: "id",
    as: "space_info"
  }
},
{
  $unwind: "$space_info"
},

```

```

{
  $group: {
    _id: "$space_info.name",
    SpaceName: { $first: "$space_info.name" },
    Location: { $first: "$space_info.location" },
    TotalRevenue: { $sum: "$total" }, // Summing the total booking amount
    TotalBookings: { $sum: 1 }, // Counting the number of bookings
    AverageBookingRate: { $avg: "$space_info.hourlyRate" }, // Averaging
the hourly rate for the space
  }
},
{
  $project: {
    _id: 0,
    SpaceName: 1,
    Location: 1,
    TotalRevenue: 1,
    TotalBookings: 1,
    AverageBookingRate: 1,
    PerformanceStatus: {
      $switch: {
        branches: [
          { case: { $gte: ["$TotalBookings", 5] }, then: "High Demand" },
          { case: { $gte: ["$TotalBookings", 2] }, then: "Moderate Demand" },
          { case: { $lt: ["$TotalBookings", 2] }, then: "Low Demand" }
        ],
        default: "Low Demand"
      }
    }
  }
}
});

```

---

**AYMEN ZUBAIR QURESHI St125405:**

**TRANSACTIONS:**

**1. New Space Listing**

```
db.space.insertOne({
  id: 22,
  name: "OurSpace",
  location: "Most",
  address: "1234 Main St",
  size: "208",
  numRooms: 5,
  hourlyRate: 15,
  halfdayRate: 120,
  fulldayRate: 220,
  rating: 4.7,
  rentFrom: "08:00:00",
  rentTo: "22:00:00",
  discount: 0.1,
  status: "Open",
  remark: "Modern space with premium amenities",
  createdAt: new Date(),
  updatedAt: new Date(),
  facility: {
    ac: true,
    wifi: true,
    projector: true,
    whiteboard: true,
    chair: 20,
    tables: 8,
    fan: 4,
    toilet: 3,
    kitchen: true,
    bed: false
  },
  Owner: {
    id: 3,
    name: "Charlie Brown"
  }
});
```

## 2. Feedback Review and Removal

```
db.feedback.find({"id":4})
db.feedback.deleteOne({ id: 4 });
db.feedback.find({ id: 4 });
```

## 3. Editing the Bookings

```
db.bookings.updateOne(
  { "id": 1 },
  {
    $set: {
      "startFrom": new Date("2025-03-01T10:00:00.000Z"),
      "endAt": new Date("2025-03-01T12:00:00.000Z")
    }
  }
);
```

### QUERIES:

#### 1. Owner Review Feedback

```
db.feedback.aggregate([
  { $lookup: { from: "space", localField: "space.spacedd", foreignField: "-id", as: "spaceDetails" } },
  { $unwind: "$spaceDetails" },
  { $match: { "spaceDetails.Owner.userid": ObjectId("67bca321a88c21f2dc64ec9f") } },
  { $project: { _id: 0, customerName: "$customer.name", spaceName: "$spaceDetails.name",
review: 1, rating: 1, createdAt: 1 } }
]);
```

#### 2. Recently Listed Space

```
db.space.aggregate([
  {
    $match: {}
  },
  {
    $project: {
      _id: 0,
      spacetime: "$name",
      ownerName: 1,
      Location: 1,

```

```

        rating: 1,
        createdAt: 1
      }
    },
    {
      $sort: { createdAt: -1 }
    },
    {
      $limit: 10
    }
  ]
});

```

### 3. Retrieve a Specific Booking Details

```

db.booking.aggregate([
  {
    $match: { "id": 2 }
  },
  {
    $project: {
      _id: 0,
      customerName: "$customer.name",
      bookingDate: 1,
      startFrom: 1,
      endAt: 1,
      status: 1,
      paymentAmount: "$payment.amountToPay",
      createdAt: 1,
      updatedAt: 1
    }
  }
]);

```

### 4. Popular Booking Time Slots

```

db.booking.aggregate([
  {
    $project: {
      spaceName: "$space.name",
      timeSlot: {
        $cond: [
          { $lt: [{ $hour: { $toDate: "$startFrom" } }, 12] },
          "Morning",
          {

```



```

        $cond: [
          { $lt: [{ $hour: { $toDate: "$startFrom" } }, 18] },
          "Afternoon",
          "Evening"
        ]
      }
    ]
  }
},
{
  $group: {
    _id: { spaceName: "$spaceName", timeSlot: "$timeSlot" },
    count: { $sum: 1 }
  }
}
]);

```

## 5. Most Popular Spaces by Booking Frequency

```

db.booking.aggregate([
  {
    $group: {
      _id: "$space.spacedd",
      spaceName: { $first: "$space.name" },
      bookingCount: { $sum: 1 }
    }
  },
  {
    $sort: { bookingCount: -1 }
  },
  {
    $project: {
      _id: 0,
      spacedd: "$_id",
      spaceName: 1,
      bookingCount: 1
    }
  }
]);

```