

ZOMATO CASESTUDY

1. No of orders in each location :

```
db.restaurant.aggregate([
  {$group:{_id:"$location",orderCount:{$sum:1}}},
  {$sort:{orderCount:1}}])
```

```
zomato> db.restaurant.aggregate([
...  {$group:{_id:"$location",orderCount:{$sum:1}}},
...  {$sort:{orderCount:1}}])
[
  { _id: 'Peenya', orderCount: 1 },
  { _id: 'Rajarajeshwari Nagar', orderCount: 2 },
  { _id: 'Jakkur', orderCount: 3 },
  { _id: 'Yelahanka', orderCount: 6 },
  { _id: 'West Bangalore', orderCount: 6 },
  { _id: 'Central Bangalore', orderCount: 8 },
  { _id: 'Nagarbhavi', orderCount: 9 },
  { _id: 'Kengeri', orderCount: 9 },
  { _id: 'North Bangalore', orderCount: 14 },
  { _id: 'Hebbal', orderCount: 14 },
  { _id: 'Uttarahalli', orderCount: 17 },
  { _id: 'KR Puram', orderCount: 18 },
  { _id: 'Kanakapura Road', orderCount: 19 },
  { _id: null, orderCount: 21 },
  { _id: 'Old Madras Road', orderCount: 22 },
  { _id: 'Mysore Road', orderCount: 22 },
  { _id: 'Langford Town', orderCount: 27 },
  { _id: 'Sankey Road', orderCount: 27 },
  { _id: 'Rammurthy Nagar', orderCount: 32 },
  { _id: 'Magadi Road', orderCount: 34 }
]
Type "it" for more
zomato> _
```

2.Find total no of hotels in Bangalore and list top 5:

```
db.restaurant.find({}, {name:1,rate:1,votes:1}).limit(5)
```

```
zomato> db.restaurant.find({}, {name:1,rate:1,votes:1}).limit(5)
[
  {
    _id: ObjectId('6673c0c3e8b9dde76e4afa5c'),
    name: 'Jaiss',
    rate: '4.1/5',
    votes: 775
  },
  {
    _id: ObjectId('6673c0c3e8b9dde76e4afa5d'),
    name: 'Spice Elephant',
    rate: '4.1/5',
    votes: 787
  },
  {
    _id: ObjectId('6673c0c3e8b9dde76e4afa5e'),
    name: 'San Churro Cafe',
    rate: '3.8/5',
    votes: 918
  },
  {
    _id: ObjectId('6673c0c3e8b9dde76e4afa5f'),
    name: 'Addhuri Udupi Bhojana',
    rate: '3.7/5',
    votes: 88
  },
  {
    _id: ObjectId('6673c0c3e8b9dde76e4afa60'),
    name: 'Grand Village',
    rate: '3.8/5',
    votes: 166
  }
]
zomato> _
```

3. Find the most liked dishes by the customer

```
db.restaurant.aggregate([
  {$unwind:"$dish_liked"},
  {$group: {_id:"$dish_liked",count:{$sum:1}}},
  {$sort:{count:-1}},
  {$limit:10}])
```

```
zomato> db.restaurant.aggregate([
... {$unwind:"$dish_liked"},
... {$group: {_id:"$dish_liked",count:{$sum:1}}},
... {$sort:{count:-1}},
... {$limit:10}])
[
  { _id: 'Biryani', count: 182 },
  { _id: 'Chicken Biryani', count: 73 },
  { _id: 'Friendly Staff', count: 69 },
  { _id: 'Waffles', count: 68 },
  { _id: 'Paratha', count: 57 },
  { _id: 'Masala Dosa', count: 56 },
  { _id: 'Rooftop Ambience', count: 42 },
  { _id: 'Coffee', count: 42 },
  { _id: 'Pizza', count: 38 },
  { _id: 'Burgers', count: 33 }
]
zomato> _
```

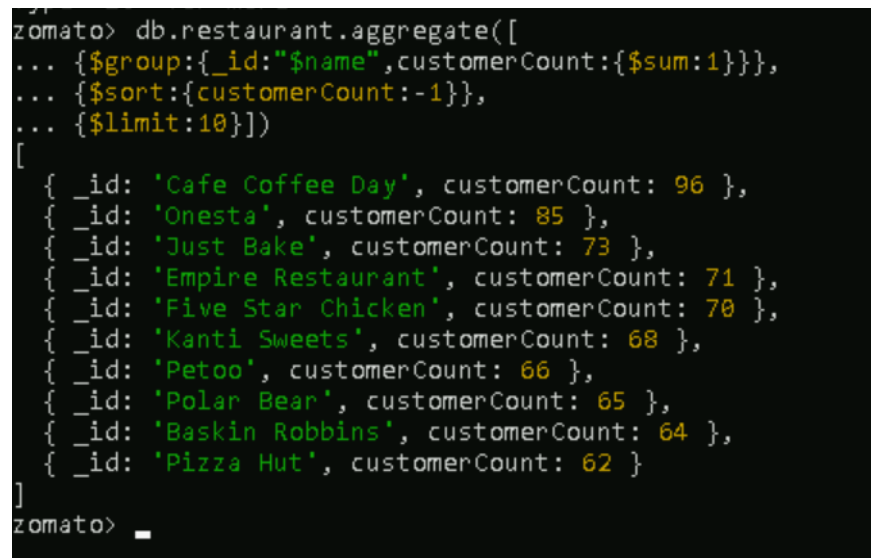
4. Find total sales in each hotel:

```
db.restaurant.aggregate([
  {$group: {_id:"$name",totalSales:{$sum:"$approx_cost(for two people)"}},
  {$sort:{totalSales:-1}}])
```

```
zomato> db.restaurant.aggregate([
... {$group: {_id:"$name",totalSales:{$sum:"$approx_cost(for two people)"}},
... {$sort:{totalSales:-1}}])
[
  { _id: 'Cafe Coffee Day', totalSales: 81100 },
  { _id: 'Smooor', totalSales: 51100 },
  { _id: 'Onesta', totalSales: 51000 },
  { _id: 'Empire Restaurant', totalSales: 48650 },
  { _id: 'Pizza Hut', totalSales: 45650 },
  { _id: 'Beijing Bites', totalSales: 45300 },
  { _id: 'Petoo', totalSales: 43550 },
  { _id: 'Truffles', totalSales: 38700 },
  { _id: 'Mani's Dum Biryani', totalSales: 35250 },
  { _id: 'Smally's Resto Cafe', totalSales: 35100 },
  { _id: 'Subway', totalSales: 33700 },
  { _id: 'Burger King', totalSales: 32700 },
  { _id: 'Mudpipe Cafe', totalSales: 31950 },
  { _id: 'Chung Wah', totalSales: 30450 },
  { _id: 'The Chocolate Heaven', totalSales: 30400 },
  { _id: 'Sweet Truth', totalSales: 30000 },
  { _id: 'New Prashanth Hotel', totalSales: 29900 },
  { _id: 'Domino's Pizza', totalSales: 29600 },
  { _id: 'McDonald's', totalSales: 29500 },
  { _id: 'Pizza Stop', totalSales: 29500 }
]
Type "it" for more
zomato> _
```

5. Find which hotels has the highest number of customers :

```
db.restaurant.aggregate([
  {$group:{_id:"$name",customerCount:{$sum:1}}},
  {$sort:{customerCount:-1}},
  {$limit:10}])
```



```
zomato> db.restaurant.aggregate([
...  {$group:{_id:"$name",customerCount:{$sum:1}}},
...  {$sort:{customerCount:-1}},
...  {$limit:10}])
[
  { _id: 'Cafe Coffee Day', customerCount: 96 },
  { _id: 'Onesta', customerCount: 85 },
  { _id: 'Just Bake', customerCount: 73 },
  { _id: 'Empire Restaurant', customerCount: 71 },
  { _id: 'Five Star Chicken', customerCount: 70 },
  { _id: 'Kanti Sweets', customerCount: 68 },
  { _id: 'Petoo', customerCount: 66 },
  { _id: 'Polar Bear', customerCount: 65 },
  { _id: 'Baskin Robbins', customerCount: 64 },
  { _id: 'Pizza Hut', customerCount: 62 }
]
zomato> _
```

6. Find total sales for each dish in each hotel :

```
db.restaurant.aggregate([
  {$match: {
    "approx_cost(for two people)": { $exists: true, $ne: null },
    dish_liked: { $exists: true, $ne: [] } }},
  {$addFields: {
    approx_cost_num: {
      $convert: {
        input: {
          $replaceAll: {
            input: { $toString: "$approx_cost(for two people)" },
            find: ",",
            replacement: ""
          }
        }
      }
    }
  }
})
```

```

    },
    to: "double",
    onError: 0, // Handle conversion errors by setting value to 0
    onNull: 0
  }},
  {$unwind: "$dish_liked" // Unwind the dish_liked array},
  {$group: {
    _id: { hotel: "$name", dish: "$dish_liked" }, // Group by hotel and dish
    totalSales: { $sum: "$approx_cost_num" } // Sum the sales}},
  {$sort: { "_id.hotel": 1, totalSales: -1 } // Sort by hotel name and total sales in
  descending order},
  {$limit: 10 // Limit the results to the top 10}
])

```

7. Find the hotels which have menu and don't have menu

```

db.restaurant.aggregate([
  {$project: {name: 1, hasMenu:
  {$cond: {if: {$gt: [{ $type: "$menu_item"}, "missing"]},
  then: true, else: false}}}},
  {$group: { _id: "$hasMenu", count: { $sum: 1 }}}})

```

```

zomato> db.restaurant.aggregate([
...  {$project: {name: 1, hasMenu:
...  {$cond: {if: {$gt: [{ $type: "$menu_item"}, "missing"]},
...  then: true, else: false}}}},
...  {$group: { _id: "$hasMenu", count: { $sum: 1 }}}})
[ { _id: false, count: 39617 }, { _id: true, count: 12100 } ]
zomato>

```

8. Find the average sales in each town in Bangalore and sort the result locationwise

```

db.restaurant.aggregate([
  {$match: {location: { $regex: /Bangalore/i }}},
  {$addFields: {approx_cost_num: {$toDouble: {
  $replaceAll: {input: { $toString: "$approx_cost(for two people)" },
  find: ",", replacement: ""}}}}},

```

```
{ $group: { _id: "$location", averageSales: { $avg: "$approx_cost_num" } } },
{ $sort: { _id: 1 } } } );
```

```
zomato> db.restaurant.aggregate([
... { $match: { location: { $regex: /Bangalore/i } } },
... { $addFields: { approx_cost_num: { $toDouble: {
... $replaceAll: { input: { $toString: "$approx_cost(for two people)" },
... find: ",", replacement: "" } } } } },
... { $group: { _id: "$location", averageSales: { $avg: "$approx_cost_num" } } },
... { $sort: { _id: 1 } } } ]
[
  { _id: 'Central Bangalore', averageSales: 387.5 },
  { _id: 'East Bangalore', averageSales: 401.16279069767444 },
  { _id: 'North Bangalore', averageSales: 332.14285714285717 },
  { _id: 'South Bangalore', averageSales: 330.1869158878505 },
  { _id: 'West Bangalore', averageSales: 350 }
]
zomato> _
```

9. Find the hotels which has rating > 4.5 and cost < 500

```
db.restaurant.aggregate([
  $addFields: { approx_cost_num: { $convert: {
    input: { $replaceAll: { input: { $toString: "$approx_cost(for two people)" },
    find: ",", replacement: "" } }, to: "double", onError: null, onNull: null } },
    rate_num: { $convert: { input: { $trim:
    { input: { $arrayElemAt: [ { $split: ["$rate", "/" ] }, 0 ] } },
    to: "double", onError: null, onNull: null } } },
    { $match: { rate_num: { $gt: 4.5 }, approx_cost_num: { $lt: 500 } } },
    { $project: { name: 1, rate_num: 1, approx_cost_num: 1 } } ] );
```

```
zomato> db.restaurant.aggregate([
... { $addFields: { approx_cost_num: { $convert: {
... input: { $replaceAll: { input: { $toString: "$approx_cost(for two people)" },
... find: ",", replacement: "" } }, to: "double", onError: null, onNull: null } },
... rate_num: { $convert: { input: { $trim:
... { input: { $arrayElemAt: [ { $split: ["$rate", "/" ] }, 0 ] } },
... to: "double", onError: null, onNull: null } } },
... { $match: { rate_num: { $gt: 4.5 }, approx_cost_num: { $lt: 500 } } },
... { $project: { name: 1, rate_num: 1, approx_cost_num: 1 } } ] );
[
  {
    _id: ObjectId('6673c0c3e8b9dde76e4afb00'),
    name: 'The Blue Wagon - Kitchen',
    approx_cost_num: 400,
    rate_num: 4.6
  },
  {
    _id: ObjectId('6673c0c3e8b9dde76e4afc4'),
    name: 'Taza Thindi',
    approx_cost_num: 100,
    rate_num: 4.7
  },
  {
    _id: ObjectId('6673c0c4e8b9dde76e4b0430'),
    name: 'The Blue Wagon - Kitchen',
    approx_cost_num: 400,
    rate_num: 4.6
  },
  {
    _id: ObjectId('6673c0c4e8b9dde76e4b05aa'),
    name: 'The Blue Wagon - Kitchen',
    approx_cost_num: 400,
    rate_num: 4.6
  },
  {
    _id: ObjectId('6673c0c4e8b9dde76e4b0766'),
    name: 'BrahmIn's Coffee Bar',
    approx_cost_num: 100,
    rate_num: 4.8
  }
]
```

```

    _id: ObjectId('6673c0c4e8b9dde76e4b05aa'),
    name: 'The Blue Wagon - Kitchen',
    approx_cost_num: 400,
    rate_num: 4.6
  },
  {
    _id: ObjectId('6673c0c4e8b9dde76e4b0766'),
    name: 'Brahmin's Coffee Bar',
    approx_cost_num: 100,
    rate_num: 4.8
  },
  {
    _id: ObjectId('6673c0c4e8b9dde76e4b07fe'),
    name: 'The Blue Wagon - Kitchen',
    approx_cost_num: 400,
    rate_num: 4.6
  },
  {
    _id: ObjectId('6673c0c5e8b9dde76e4b0e7d'),
    name: 'Belgian Waffle Factory',
    approx_cost_num: 400,
    rate_num: 4.9
  },
  {
    _id: ObjectId('6673c0c5e8b9dde76e4b110d'),
    name: 'Belgian Waffle Factory',
    approx_cost_num: 400,
    rate_num: 4.9
  },
  {
    _id: ObjectId('6673c0c5e8b9dde76e4b154f'),
    name: 'Dock Frost'd',
    approx_cost_num: 400,
    rate_num: 4.6
  },
  {
    _id: ObjectId('6673c0c6e8b9dde76e4b1bcd'),
    name: 'The Pancake Story',
    approx_cost_num: 300,
    rate_num: 4.6
  },

```

10. Find the best rated hotels, calculate the total votes and sort it in descending order of votes :

```

db.restaurant.aggregate([
  {$group: {_id: "$name",
    totalVotes: { $sum: "$votes" },
    averageRating: { $avg: "$rate" } }},
  {$sort: { totalVotes: -1 }}]);

```

```

zomato> db.restaurant.aggregate([
...  {$group: {
...    _id: "$name",
...    totalVotes: { $sum: "$votes" },
...    averageRating: { $avg: "$rate" }  }},
...  {$sort: { totalVotes: -1 }}]);
[
  { _id: 'Onesta', totalVotes: 347520, averageRating: null },
  { _id: 'Truffles', totalVotes: 301059, averageRating: null },
  { _id: 'Empire Restaurant', totalVotes: 229808, averageRating: null },
  { _id: 'Hammered', totalVotes: 180602, averageRating: null },
  { _id: 'The Black Pearl', totalVotes: 172122, averageRating: null },
  { _id: 'Meghana Foods', totalVotes: 129557, averageRating: null },
  { _id: 'Barbeque Nation', totalVotes: 108425, averageRating: null },
  {
    _id: "Smally's Resto Cafe",
    totalVotes: 102877,
    averageRating: null
  },
  {
    _id: 'Byg Brewski Brewing Company',
    totalVotes: 99531,
    averageRating: null
  },
  { _id: "Gilly's Restobar", totalVotes: 98808, averageRating: null },
  {
    _id: 'Arbor Brewing Company',
    totalVotes: 92362,
    averageRating: null
  },
  { _id: 'House Of Commons', totalVotes: 90573, averageRating: null },
  { _id: 'Chutney Chang', totalVotes: 89910, averageRating: null },
  {
    _id: "Fenny's Lounge And Kitchen",
    totalVotes: 89183,
    averageRating: null
  },
]

```

```

{
  _id: "AB's - Absolute Barbecues",
  totalVotes: 86418,
  averageRating: null
},
{
  _id: 'Church Street Social',
  totalVotes: 83179,
  averageRating: null
},
{ _id: 'Prost Brew Pub', totalVotes: 78609, averageRating: null },
{ _id: 'The Biere Club', totalVotes: 76649, averageRating: null },
{ _id: 'Stoner', totalVotes: 75194, averageRating: null },
{ _id: 'Koramangala Social', totalVotes: 75021, averageRating: null }
]
Type "it" for more
zomato>

```

11.Delete the column dish_liked from the collection :

```

db.restaurant.updateMany({},
{ $unset: { dish_liked: "" } } )

```

```

zomato> db.restaurant.updateMany(
...  {}, // Match all documents
...  { $unset: { dish_liked: "" } } // Unset the dish_liked field
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 51717,
  modifiedCount: 0,
  upsertedCount: 0
}
zomato>

```

12.find the same named restaurants using \$text and \$Search :

```
db.restaurant.createIndex({ name: "text" });
```

```
db.restaurant.find({ $text: { $search: "\"ABC Restaurant\"" } });
```

```
zomato> db.restaurant.createIndex({ name: "text" });  
name_text  
zomato> db.restaurant.find({ $text: { $search: "\"ABC Restaurant\"" } });  
zomato>
```