Group_01

Group Member Names:

Name	BITS ID	Contribution
Movva Manaswi	2021sc04180	100%
Ajay Saxena	2021sc04160	100%
Aditi	2021sc04050	100%
Prarita Arora	2021sc04049	100%

// Connect to MongoDB Shell through MongoDB Atlas

//Q1 to Q3 by Movva Manaswi(2021sc04180)

```
mongosh "mongodb+srv://cluster0.nh1h2hm.mongodb.net/" --apiVersion 1 --username 2021sc04180
```

password: MongoDBM

// Use the appropriate database

```
use Roadaccidents
```

Collection: Roadacc

1. What is the total number of road accidents in Maharashtra during 2018?

Query:

```
db.Roadacc.aggregate([
 {
  $match: {
   State: "Maharashtra"
  }
 },
 {
  $group: {
   _id: null,
   "total number of road accidents in Maharashtra during 2018": {
    $sum: "$Road Accidents  during 2018"
   }
  }
 }
])
```

Output: 'total number of road accidents in Maharashtra during 2018': 35717

2.Which state had the highest number of road accidents in 2019?

//To find which state had the highest number of road accidents in 2019 from the "Roadacc" collection in the "Roadaccidents" database, we can use the MongoDB aggregation framework.

Query:

```
db.Roadacc.aggregate(
   { $project:
    {"State":1, _id:0," Road Accidents  during 2019":1 }
  },
  { $sort:
    {' Road Accidents  during 2019':-1} }, { $skip:1}, { $limit:1}
)
```

Output: [{ State: 'Tamil Nadu', ' Road Accidents during 2019': 57228 }]

3. Find the difference in road accidents in 2018 and 2020 in karnataka?(use $substract).

//we can use the MongoDB find method to filter the data for Karnataka, and then simply subtract the number of accidents in 2020 from the number in 2018.

```
db.Roadacc.find(
   { "State": "Karnataka" },
   {
      "_id": 0,
      "State": 1,
      "Difference in Accidents (2018-2020)": { $subtract: [ "$Road Accidents  during 2018", "$Road Accidents  during 
2020" ] }
   }
)
```

Output: [{ State: 'Karnataka', 'Difference in Accidents (2018-2020)': 7529 }]

Explanation:

In this query:

db.Roadacc.find() is the method used to query the "Roadacc" collection.

The first argument { "State": "Karnataka" } is the query criteria to filter the data for Karnataka.

The second argument is an object specifying which fields to include in the result. Here, _id is excluded (with 0), State is included (with 1), and Difference in Accidents (2018-2020) is a computed field resulting from the subtraction of "Road Accidents during 2020" from "Road Accidents during 2018".

//Q4 to Q6 by Ajay Saxena(2021sc04160)

// Connect to MongoDB Shell through MongoDB Atlas

mongosh "mongodb+srv://cluster0.nh1h2hm.mongodb.net/" --apiVersion 1 --username user1

password: user1

// Use the appropriate database

use BDS_BITS

Collection: Accidents

4. Which state had the lowest number of persons killed in 2019?

We can use MongoDB's aggregation framework to find the state that had the lowest number of persons killed in 2019. The aggregation pipeline would first sort the documents in the "Accidents" collection in ascending order of "Persons Killed 2019", then limit the result to one document - the one with the lowest number of persons killed.

Query:

```
db.Accidents.aggregate([
 { $sort: { "Persons Killed 2019": 1 } },
 { $limit: 1 },
 { $project: { "State": 1, "Persons Killed 2019": 1, "_id": 0 } }
])
```

Output : State: 'Lakshadweep', 'Persons Killed 2019': 0

5. Calculate the average number of road accidents per lakh population in Goa for the years 2018-2020.

To calculate the average number of road accidents per lakh population in Goa for the years 2018-2020, we would use the MongoDB aggregation framework to find the document corresponding to Goa and then perform an operation to calculate the average.

Query:

```
db.Accidents.aggregate([
 { $match: { "State": "Goa" } },
 { $project: {
    "Average Accidents Per Lakh Population 2018-2020": {
       $avg: [
          "$Total Number of Accidents  Per Lakh Population - 2018",
          "$Total Number of Accidents  Per Lakh Population - 2019",
          "$Total Number of Accidents  Per Lakh Population - 2020"
       ]
    },
    "_id": 0
  }
 }
])
```

Output: 'Average Accidents Per Lakh Population 2018-2020': 147.70000000000002

6. What is the total number of road accidents during 2018 to 2020 in Bihar?(use $add)

In order to calculate the total number of road accidents during 2018 to 2020 in Bihar, we can use MongoDB's aggregate function to match the "State" with "Bihar", and then use the $add operator in a $project stage to sum the accidents of the years 2018, 2019, and 2020.

Query:

```
db.Accidents.aggregate([
 { $match: { "State": "Bihar" } },
 { $project: {
    "Total Accidents 2018-2020": {
       $add: [
          "$Road Accidents  during 2018",
          "$Road Accidents  during 2019",
          "$Road Accidents  during 2020"
       ]
    },
    "_id": 0
  }
 }
])
```

Output: 'Total Accidents 2018-2020': 28246 }]

7. How many road accidents per 10,000 vehicles were reported in Madhya Pradesh during 2017?

To find the number of road accidents per 10,000 vehicles in Madhya Pradesh during 2017, you can use MongoDB's find() method and specify the state and the year.

Query:

```
db.Accidents.find(
 { "State": "Madhya Pradesh" },
 { "_id": 0, "Road Accidents per 10,000 Vehicles- 2017": 1 }
)
```

Output: 'Road Accidents per 10,000 Vehicles- 2017': 41.41414509

8. Calculate the average number of persons injured per lakh population in Jharkhand for the years 2017-2019.

To calculate the average number of persons injured per lakh population in Jharkhand for the years 2017-2019, we would need to fetch the data for those years and then calculate the average.

In MongoDB, we can't directly calculate averages of multiple fields. Instead, we need to use the aggregation pipeline to calculate the average. Unfortunately, MongoDB's aggregation framework doesn't have an operator for averaging across multiple fields of a single document, so we'd need to manually add the fields together and then divide by the count of fields.

Query:

```
db.Accidents.aggregate([
 {
  $match: {
   "State": "Jharkhand"
  }
 },
 {
  $project: {
   _id: 0,
   average_injured_per_lakh: {
    $divide: [
     { $add: ["$Injury Per Lakh Population - 2017", "$Injury Per Lakh Population - 2018", "$Injury Per Lakh
Population - 2019"] },
     3
    ]
   }
  }
 }
])
```

Output: average_injured_per_lakh: 11.323838236666667

9. What is the total number of road accidents per 10,000 km of roads in Chhattisgarh for the year 2018?

To find the total number of road accidents per 10,000 km of roads in Chhattisgarh for the year 2018, we would use the find() function to query for the specific state and year.

Query:

```
db.Accidents.find(
 { "State": "Chhattisgarh" },
 { "_id": 0, "Accidents per 10,000 Km of Roads - 2018": 1 }
)
```

Output: 'Accidents per 10,000 Km of Roads - 2018': 1351.6

10. Which state has the highest share in death in the year 2020?

//To find which state has the highest share in death in the year 2020, from the "Accidents" collection in the "BDS_BITS" database, we can use the MongoDB aggregation framework.

Query:

```
db.Accidents.aggregate(
   { $project:
    {"State": 1, _id: 0,"Share in Death- 2020" : 1 }
  },
  { $sort:
    {'Share in Death- 2020' : -1} }, { $skip : 1}, { $limit : 1}
)
```

Output: [{ State: 'Uttar Pradesh', 'Share in Death- 2020': 12.7 }]

11. Find the total number of road accidents during 2018 all over the country?

Query:

```
db.Accidents.aggregate([
 {
   $group: {
     _id: null,
     totalAccidents2018: { $sum: "$Road Accidents  during 2018" }
   }
 }
])
```

Output: id: null, totalAccidents2018: 934088

12. What are the top five states with lowest accidents per lakh population during 2017?

Query:

```
db.Accidents.aggregate([
 {
   $match: { "Accidents  Per Lakh Population - 2017": { $ne: null } }
 },
 {
   $sort: { "Accidents  Per Lakh Population - 2017": 1 }
 },
 {
   $limit: 5
```

```
  },
  {
   $project: { _id: 0, State: 1, Accidents2017: "$Accidents  Per Lakh Population - 2017" }
  }
])
```

Explanation:

This query uses the $match stage to filter out the documents where the "Accidents Per Lakh Population - 2017" field is not null. Then, it uses the $sort stage to sort the documents in ascending order based on the "Accidents Per Lakh Population - 2017" field. The $limit stage is used to limit the result to only the top five documents.

Finally, the $project stage is used to reshape the output and include only the "State" and "Accidents Per Lakh Population - 2017" fields in the result while excluding the "_id" field.

Output:

```
  { State: 'Lakshadweep', Accidents2017: 1.2 },
  { State: 'Mizoram', Accidents2017: 6.3 },
  { State: 'Bihar', Accidents2017: 8.3 },
  { State: 'West Bengal', Accidents2017: 12.2 },
  { State: 'Tripura', Accidents2017: 12.9 }
]
```