1. **USRES Collection:**
   1. User count by their role:

db.users.aggregate([

{ $group: { \_id: "$role", count: { $sum: 1 } } }

])

* 1. Average reward points:

db.users.aggregate([

{ $group: { \_id: null, avgPoints: { $avg: "$rewards.points" } } }

])

* 1. Recent Users:

db.users.aggregate([

{ $match: { createdAt: { $gt: new Date(new Date()-30\*24\*3600\*1000) } } },

{ $count: "newUsers" }

])

1. **VOLUNTEER Collection:**
   1. Count available and unavalibale:

db.volunteers.aggregate([

{ $group: { \_id: "$availability", count: { $sum: 1 } } }

])

* 1. Volunteers with GPS enabled:

db.volunteers.aggregate([

{ $match: { gpsTrackingEnabled: true } },

{ $count: "gpsEnabledCount" }

])

* 1. Request per volunteer:

db.volunteers.aggregate([

{ $project: { requestCount: { $size: "$assignedRequests" } } }

])

1. **DONATIONS Collection:**
   1. Total donated Amount:

db.donations.aggregate([

{ $group: { \_id: "$donorId", totalAmount: { $sum: "$amount" } } }

])

* 1. Average donation value:

db.donations.aggregate([

{ $group: { \_id: null, avgDonation: { $avg: "$amount" } } }

])

1. **ALERTS Collection:**
   1. Alerts by type:

db.alerts.aggregate([

{ $group: { \_id: "$type", count: { $sum: 1 } } }

])

* 1. Severity Distribution:

db.alerts.aggregate([

{ $group: { \_id: "$severity", total: { $sum: 1 } } }

])

1. **HELPREQUEST Collection:**
   1. Count by category:

db.helpRequests.aggregate([

{ $group: { \_id: "$category", count: { $sum: 1 } } }

])

* 1. Hight priority request count:

db.helpRequests.aggregate([

{ $match: { priority: "high" } },

{ $count: "highPriority" }

])

* 1. Request per status:

db.helpRequests.aggregate([

{ $group: { \_id: "$status", count: { $sum: 1 } } }

])

1. **RESOURCE Collection:**
   1. Total quantity by category:

db.resources.aggregate([

{ $group: { \_id: "$category", totalQty: { $sum: "$quantityAvailable" } } }

])

* 1. Items with less quantity:

db.resources.aggregate([

{ $match: { quantityAvailable: { $lt: 50 } } }

])

* 1. Count per manager:

db.resources.aggregate([

{ $group: { \_id: "$managedBy", count: { $sum: 1 } } }

])

1. **MESSAGE Collection:** 
   1. Sender unread count::

db.messages.aggregate([

{ $match: { read: false } },

{ $group: { \_id: "$senderId", unread: { $sum: 1 } } }

])

* 1. Message per use pair:

db.messages.aggregate([

{ $group: { \_id: { from: "$senderId", to: "$receiverId" }, count: { $sum: 1 } } }

])

* 1. Dialy message volume:

db.messages.aggregate([

{ $group: { \_id: { day: { $dayOfMonth: "$timestamp" } }, total: { $sum: 1 } } }

])

1. **FEEDBACK Collection:**
   1. Average rating per module:

db.feedback.aggregate([

{ $group: { \_id: "$forModule", avgRating: { $avg: "$rating" } } }

])

* 1. Low rating counts:

db.feedback.aggregate([

{ $match: { rating: { $lt: 3 } } },

{ $count: "lowRatings" }

])

* 1. Rating distribution:

db.feedback.aggregate([

{ $group: { \_id: "$rating", count: { $sum: 1 } } },

{ $sort: { \_id: -1 } }

])

1. **ANALYTICS Collection:**
   1. Averagr help request:

db.analytics.aggregate([

{ $group: { \_id: null, avgRequests: { $avg: "$totalHelpRequests" } } }

])

* 1. Active zone count:

db.analytics.aggregate([

{ $unwind: "$mostAffectedZones" },

{ $group: { \_id: "$mostAffectedZones.priority", count: { $sum: 1 } } }

])

* 1. Max volunteersActive:

db.analytics.aggregate([

{ $group: { \_id: null, maxVol: { $max: "$volunteersActive" } } }

])

1. **UPDATES Collection:**
   1. Count of updates by disaster types:

db.updates.aggregate([

{$group: {\_id: "$disasterType",

totalUpdates: { $sum: 1 }} }

])

* 1. Verified and Unverified update count:

db.updates.aggregate([

{$group: {\_id: "$verified",

count: { $sum: 1 }}}

])