**Assignment1)**

**Full Code Of ASSIGFNMENT 1 :**

Microsoft Windows [Version 10.0.19042.1052]

(c) Microsoft Corporation. All rights reserved.

C:\Users\PAVAN SAI RISHI>mongo "mongodb+srv://cluster0.jliw6.mongodb.net/myFirstDatabase" --username admin

MongoDB shell version v4.4.6

Enter password:

connecting to: mongodb://cluster0-shard-00-02.jliw6.mongodb.net:27017,cluster0-shard-00-00.jliw6.mongodb.net:27017,cluster0-shard-00-01.jliw6.mongodb.net:27017/myFirstDatabase?authSource=admin&compressors=disabled&gssapiServiceName=mongodb&replicaSet=atlas-3hnxpg-shard-0&ssl=true

Implicit session: session { "id" : UUID("efdf8928-2d33-47ca-95e3-ec7e96bde3ac") }

MongoDB server version: 4.4.6

Error while trying to show server startup warnings: user is not allowed to do action [getLog] on [admin.]

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> show dbs

admin 0.000GB

local 5.656GB

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> use mongo\_practice

switched to db mongo\_practice

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.insert({title:"Fight Club", writer: "Chuck Palahniuk", year: "1999", actors:["Brad Pitt", "Edward Norton"]})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.insert({title:"Pulp Fiction", writer:"Quentin Tarantino", year:"2009", actors:["John Travolta", "Uma Thurman"]})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.insert({title:"Inglorious Basterds", writer:"Quentin Tarantino", year:"2009", actors:["Brad Pitt", "Diane Kruger", "Eli Roth"]})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.insert({title:"The Hobbit: An unexpected Journey", writer:"J.R.R. Tolkein", year:"2012",franchise:"The Hobbit"})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.insert({title:"The Hobbit: The Desolation of Smaug", writer:"J.R.R Tolkien", year:"2013", franchise:"The Hobbit"})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.insert({title:"The Hobbit: The Battle of the Five Armies", writer:"J.R.R Tolkien", year:"2002", franchise:"The Hobbit", synopsis:"Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness."})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.insert({title:"Pee Wee Herman's Big Adventures"})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.insert({title:"Avatar"})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.find()

{ "\_id" : ObjectId("60e5b3fe93d5772f67e1dd5d"), "title" : "Fight Club", "writer" : "Chuck Palahniuk", "year" : "1999", "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : ObjectId("60e5b41093d5772f67e1dd5e"), "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : "2009", "actors" : [ "John Travolta", "Uma Thurman" ] }

{ "\_id" : ObjectId("60e5b43893d5772f67e1dd5f"), "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : "2009", "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }

{ "\_id" : ObjectId("60e5b48a93d5772f67e1dd60"), "title" : "The Hobbit: An unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : "2012", "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60e5b4bd93d5772f67e1dd61"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R Tolkien", "year" : "2013", "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60e5b4ec93d5772f67e1dd62"), "title" : "The Hobbit: The Battle of the Five Armies", "writer" : "J.R.R Tolkien", "year" : "2002", "franchise" : "The Hobbit", "synopsis" : "Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness." }

{ "\_id" : ObjectId("60e5b50b93d5772f67e1dd63"), "title" : "Pee Wee Herman's Big Adventures" }

{ "\_id" : ObjectId("60e5b52293d5772f67e1dd64"), "title" : "Avatar" }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.find({writer:"Quentin Tarantino"})

{ "\_id" : ObjectId("60e5b41093d5772f67e1dd5e"), "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : "2009", "actors" : [ "John Travolta", "Uma Thurman" ] }

{ "\_id" : ObjectId("60e5b43893d5772f67e1dd5f"), "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : "2009", "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.find({actors:"Brad Pitt"})

{ "\_id" : ObjectId("60e5b3fe93d5772f67e1dd5d"), "title" : "Fight Club", "writer" : "Chuck Palahniuk", "year" : "1999", "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : ObjectId("60e5b43893d5772f67e1dd5f"), "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : "2009", "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.find({franchise:"The Hobbit"})

{ "\_id" : ObjectId("60e5b48a93d5772f67e1dd60"), "title" : "The Hobbit: An unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : "2012", "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60e5b4bd93d5772f67e1dd61"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R Tolkien", "year" : "2013", "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60e5b4ec93d5772f67e1dd62"), "title" : "The Hobbit: The Battle of the Five Armies", "writer" : "J.R.R Tolkien", "year" : "2002", "franchise" : "The Hobbit", "synopsis" : "Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness." }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.find({year:{$gt:"1990", $lt:"2000"}})

{ "\_id" : ObjectId("60e5b3fe93d5772f67e1dd5d"), "title" : "Fight Club", "writer" : "Chuck Palahniuk", "year" : "1999", "actors" : [ "Brad Pitt", "Edward Norton" ] }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.find({$or:[{year:{$gt:"2010"}},{year: {$lt:"2000"}}]})

{ "\_id" : ObjectId("60e5b3fe93d5772f67e1dd5d"), "title" : "Fight Club", "writer" : "Chuck Palahniuk", "year" : "1999", "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : ObjectId("60e5b48a93d5772f67e1dd60"), "title" : "The Hobbit: An unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : "2012", "franchise" : "The Hobbit" }

{ "\_id" : ObjectId("60e5b4bd93d5772f67e1dd61"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R Tolkien", "year" : "2013", "franchise" : "The Hobbit" }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.update({\_id:ObjectId("60e5b48a93d5772f67e1dd60")}, {$set:{synopsis:"A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.update({\_id:ObjectId("60e5b4bd93d5772f67e1dd61")}, {$set:{synopsis:"The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.update({\_id:ObjectId("60e5b41093d5772f67e1dd5e")}, {$push:{actors:"Samuel L. Jackson"}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.find({synopsis:{$regex:"Bilbo"}})

{ "\_id" : ObjectId("60e5b48a93d5772f67e1dd60"), "title" : "The Hobbit: An unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : "2012", "franchise" : "The Hobbit", "synopsis" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug." }

{ "\_id" : ObjectId("60e5b4bd93d5772f67e1dd61"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R Tolkien", "year" : "2013", "franchise" : "The Hobbit", "synopsis" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring." }

{ "\_id" : ObjectId("60e5b4ec93d5772f67e1dd62"), "title" : "The Hobbit: The Battle of the Five Armies", "writer" : "J.R.R Tolkien", "year" : "2002", "franchise" : "The Hobbit", "synopsis" : "Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness." }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.find({synopsis:{$regex:"Gandalf"}})

{ "\_id" : ObjectId("60e5b4bd93d5772f67e1dd61"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R Tolkien", "year" : "2013", "franchise" : "The Hobbit", "synopsis" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring." }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.find({$and:[{synopsis:{$regex:"Bilbo"}}, {synopsis:{$not:/Gandalf/}}]})

{ "\_id" : ObjectId("60e5b48a93d5772f67e1dd60"), "title" : "The Hobbit: An unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : "2012", "franchise" : "The Hobbit", "synopsis" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug." }

{ "\_id" : ObjectId("60e5b4ec93d5772f67e1dd62"), "title" : "The Hobbit: The Battle of the Five Armies", "writer" : "J.R.R Tolkien", "year" : "2002", "franchise" : "The Hobbit", "synopsis" : "Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness." }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.find({$or:[{synopsis:{$regex:"dwarves"}}, {synopsis:{$regex:"hobbit"}}]})

{ "\_id" : ObjectId("60e5b48a93d5772f67e1dd60"), "title" : "The Hobbit: An unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : "2012", "franchise" : "The Hobbit", "synopsis" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug." }

{ "\_id" : ObjectId("60e5b4bd93d5772f67e1dd61"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R Tolkien", "year" : "2013", "franchise" : "The Hobbit", "synopsis" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring." }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.find({$and:[{synopsis:{$regex:"gold"}}, {synopsis:{$regex:"dragon"}}]})

{ "\_id" : ObjectId("60e5b48a93d5772f67e1dd60"), "title" : "The Hobbit: An unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : "2012", "franchise" : "The Hobbit", "synopsis" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug." }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.find()

{ "\_id" : ObjectId("60e5b3fe93d5772f67e1dd5d"), "title" : "Fight Club", "writer" : "Chuck Palahniuk", "year" : "1999", "actors" : [ "Brad Pitt", "Edward Norton" ] }

{ "\_id" : ObjectId("60e5b41093d5772f67e1dd5e"), "title" : "Pulp Fiction", "writer" : "Quentin Tarantino", "year" : "2009", "actors" : [ "John Travolta", "Uma Thurman", "Samuel L. Jackson" ] }

{ "\_id" : ObjectId("60e5b43893d5772f67e1dd5f"), "title" : "Inglorious Basterds", "writer" : "Quentin Tarantino", "year" : "2009", "actors" : [ "Brad Pitt", "Diane Kruger", "Eli Roth" ] }

{ "\_id" : ObjectId("60e5b48a93d5772f67e1dd60"), "title" : "The Hobbit: An unexpected Journey", "writer" : "J.R.R. Tolkein", "year" : "2012", "franchise" : "The Hobbit", "synopsis" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug." }

{ "\_id" : ObjectId("60e5b4bd93d5772f67e1dd61"), "title" : "The Hobbit: The Desolation of Smaug", "writer" : "J.R.R Tolkien", "year" : "2013", "franchise" : "The Hobbit", "synopsis" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring." }

{ "\_id" : ObjectId("60e5b4ec93d5772f67e1dd62"), "title" : "The Hobbit: The Battle of the Five Armies", "writer" : "J.R.R Tolkien", "year" : "2002", "franchise" : "The Hobbit", "synopsis" : "Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness." }

{ "\_id" : ObjectId("60e5b50b93d5772f67e1dd63"), "title" : "Pee Wee Herman's Big Adventures" }

{ "\_id" : ObjectId("60e5b52293d5772f67e1dd64"), "title" : "Avatar" }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.remove({\_id:ObjectId("60e5b50b93d5772f67e1dd63")})

WriteResult({ "nRemoved" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.movies.remove({\_id:ObjectId("60e5b52293d5772f67e1dd64")})

WriteResult({ "nRemoved" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.users.insert({username:"GoodGuyGreg", first\_name:"Good Guy", last\_name:"Greg"})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.users.insert({ username:"ScumbagSteve", fullname:{first: "Scumbag", last:"Steve"}})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.posts.insert({username:"GoodGuyGreg", title:"Passes out at Party", body:"Raises your credit score"})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.posts.insert({ username:"GoodGuyGreg", title:"Steals your identity", body:"Raises your credit score"})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.posts.insert({username:"GoodGuyGreg", title:"Reports a bug in your code", body:"Sends you a pull request"})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.posts.insert({ username:"ScumbagSteve", title:"Borrows something", body:"Sells it"})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.posts.insert({ username:"ScumbagSteve", title:"Borrows everything", body:"The end"})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.posts.insert({username:"ScumbagSteve", title:"Forks your repo on github", body:"Sets to private"}

...

... )

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.posts.find()

{ "\_id" : ObjectId("60e5bc8f93d5772f67e1dd67"), "username" : "GoodGuyGreg", "title" : "Passes out at Party", "body" : "Raises your credit score" }

{ "\_id" : ObjectId("60e5bc9793d5772f67e1dd68"), "username" : "GoodGuyGreg", "title" : "Steals your identity", "body" : "Raises your credit score" }

{ "\_id" : ObjectId("60e5bca393d5772f67e1dd69"), "username" : "GoodGuyGreg", "title" : "Reports a bug in your code", "body" : "Sends you a pull request" }

{ "\_id" : ObjectId("60e5bcaa93d5772f67e1dd6a"), "username" : "ScumbagSteve", "title" : "Borrows something", "body" : "Sells it" }

{ "\_id" : ObjectId("60e5bcb493d5772f67e1dd6b"), "username" : "ScumbagSteve", "title" : "Borrows everything", "body" : "The end" }

{ "\_id" : ObjectId("60e5bce293d5772f67e1dd6c"), "username" : "ScumbagSteve", "title" : "Forks your repo on github", "body" : "Sets to private" }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db\_obj\_id

uncaught exception: ReferenceError: db\_obj\_id is not defined :

@(shell):1:1

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.posts.obj.id

mongo\_practice.posts.obj.id

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.posts.find()

{ "\_id" : ObjectId("60e5bc8f93d5772f67e1dd67"), "username" : "GoodGuyGreg", "title" : "Passes out at Party", "body" : "Raises your credit score" }

{ "\_id" : ObjectId("60e5bc9793d5772f67e1dd68"), "username" : "GoodGuyGreg", "title" : "Steals your identity", "body" : "Raises your credit score" }

{ "\_id" : ObjectId("60e5bca393d5772f67e1dd69"), "username" : "GoodGuyGreg", "title" : "Reports a bug in your code", "body" : "Sends you a pull request" }

{ "\_id" : ObjectId("60e5bcaa93d5772f67e1dd6a"), "username" : "ScumbagSteve", "title" : "Borrows something", "body" : "Sells it" }

{ "\_id" : ObjectId("60e5bcb493d5772f67e1dd6b"), "username" : "ScumbagSteve", "title" : "Borrows everything", "body" : "The end" }

{ "\_id" : ObjectId("60e5bce293d5772f67e1dd6c"), "username" : "ScumbagSteve", "title" : "Forks your repo on github", "body" : "Sets to private" }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.comments.insert({ username:"GoodGuyGreg", comment:"Hope you got a good deal!", post:ObjectId("60e5bcaa93d5772f67e1dd6a")})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.comments.insert({username:"GoodGuyGreg", comment:"What's mine is yours!", post:ObjectId("60e5bcb493d5772f67e1dd6b")})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.comments.insert({username:"GoodGuyGreg", comment:"Don't violate the licensing agreement!", post:ObjectId("60e5bce293d5772f67e1dd6c")})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.comments.insert({username:"ScumbagSteve", comment:"It still isn't clean", post:ObjectId("60e5bc8f93d5772f67e1dd67")})

WriteResult({ "nInserted" : 1 })

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.users.find().pretty()

{

"\_id" : ObjectId("60e5bc4c93d5772f67e1dd65"),

"username" : "GoodGuyGreg",

"first\_name" : "Good Guy",

"last\_name" : "Greg"

}

{

"\_id" : ObjectId("60e5bc6893d5772f67e1dd66"),

"username" : "ScumbagSteve",

"fullname" : {

"first" : "Scumbag",

"last" : "Steve"

}

}

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.posts.find().pretty()

{

"\_id" : ObjectId("60e5bc8f93d5772f67e1dd67"),

"username" : "GoodGuyGreg",

"title" : "Passes out at Party",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("60e5bc9793d5772f67e1dd68"),

"username" : "GoodGuyGreg",

"title" : "Steals your identity",

"body" : "Raises your credit score"

}

{

"\_id" : ObjectId("60e5bca393d5772f67e1dd69"),

"username" : "GoodGuyGreg",

"title" : "Reports a bug in your code",

"body" : "Sends you a pull request"

}

{

"\_id" : ObjectId("60e5bcaa93d5772f67e1dd6a"),

"username" : "ScumbagSteve",

"title" : "Borrows something",

"body" : "Sells it"

}

{

"\_id" : ObjectId("60e5bcb493d5772f67e1dd6b"),

"username" : "ScumbagSteve",

"title" : "Borrows everything",

"body" : "The end"

}

{

"\_id" : ObjectId("60e5bce293d5772f67e1dd6c"),

"username" : "ScumbagSteve",

"title" : "Forks your repo on github",

"body" : "Sets to private"

}

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.posts.find({username:"GoodGuyGreg"})

{ "\_id" : ObjectId("60e5bc8f93d5772f67e1dd67"), "username" : "GoodGuyGreg", "title" : "Passes out at Party", "body" : "Raises your credit score" }

{ "\_id" : ObjectId("60e5bc9793d5772f67e1dd68"), "username" : "GoodGuyGreg", "title" : "Steals your identity", "body" : "Raises your credit score" }

{ "\_id" : ObjectId("60e5bca393d5772f67e1dd69"), "username" : "GoodGuyGreg", "title" : "Reports a bug in your code", "body" : "Sends you a pull request" }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.posts.find({username:"ScumbagSteve"})

{ "\_id" : ObjectId("60e5bcaa93d5772f67e1dd6a"), "username" : "ScumbagSteve", "title" : "Borrows something", "body" : "Sells it" }

{ "\_id" : ObjectId("60e5bcb493d5772f67e1dd6b"), "username" : "ScumbagSteve", "title" : "Borrows everything", "body" : "The end" }

{ "\_id" : ObjectId("60e5bce293d5772f67e1dd6c"), "username" : "ScumbagSteve", "title" : "Forks your repo on github", "body" : "Sets to private" }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.comments.find().pretty()

{

"\_id" : ObjectId("60e5c26393d5772f67e1dd6d"),

"username" : "GoodGuyGreg",

"comment" : "Hope you got a good deal!",

"post" : ObjectId("60e5bcaa93d5772f67e1dd6a")

}

{

"\_id" : ObjectId("60e5c28e93d5772f67e1dd6e"),

"username" : "GoodGuyGreg",

"comment" : "What's mine is yours!",

"post" : ObjectId("60e5bcb493d5772f67e1dd6b")

}

{

"\_id" : ObjectId("60e5c29593d5772f67e1dd6f"),

"username" : "GoodGuyGreg",

"comment" : "Don't violate the licensing agreement!",

"post" : ObjectId("60e5bce293d5772f67e1dd6c")

}

{

"\_id" : ObjectId("60e5c29e93d5772f67e1dd70"),

"username" : "ScumbagSteve",

"comment" : "It still isn't clean",

"post" : ObjectId("60e5bc8f93d5772f67e1dd67")

}

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.comments.find({username:"GoodGuyGreg"})

{ "\_id" : ObjectId("60e5c26393d5772f67e1dd6d"), "username" : "GoodGuyGreg", "comment" : "Hope you got a good deal!", "post" : ObjectId("60e5bcaa93d5772f67e1dd6a") }

{ "\_id" : ObjectId("60e5c28e93d5772f67e1dd6e"), "username" : "GoodGuyGreg", "comment" : "What's mine is yours!", "post" : ObjectId("60e5bcb493d5772f67e1dd6b") }

{ "\_id" : ObjectId("60e5c29593d5772f67e1dd6f"), "username" : "GoodGuyGreg", "comment" : "Don't violate the licensing agreement!", "post" : ObjectId("60e5bce293d5772f67e1dd6c") }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.comments.find({username:"ScumbagSteve"})

{ "\_id" : ObjectId("60e5c29e93d5772f67e1dd70"), "username" : "ScumbagSteve", "comment" : "It still isn't clean", "post" : ObjectId("60e5bc8f93d5772f67e1dd67") }

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.comments.find({post: db.posts.findOne({title: "Reports a bug in your code"}).\_id})

MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY> db.posts.aggregate([

... {

... $match: { title: 'Reports a bug in your code' }

... },

... {

... $lookup: {

... from: 'comments',

... localField: '\_id',

... foreignField: 'post',

... as: 'comments'

... }

... }

... ]).pretty();

{

"\_id" : ObjectId("60e5bca393d5772f67e1dd69"),

"username" : "GoodGuyGreg",

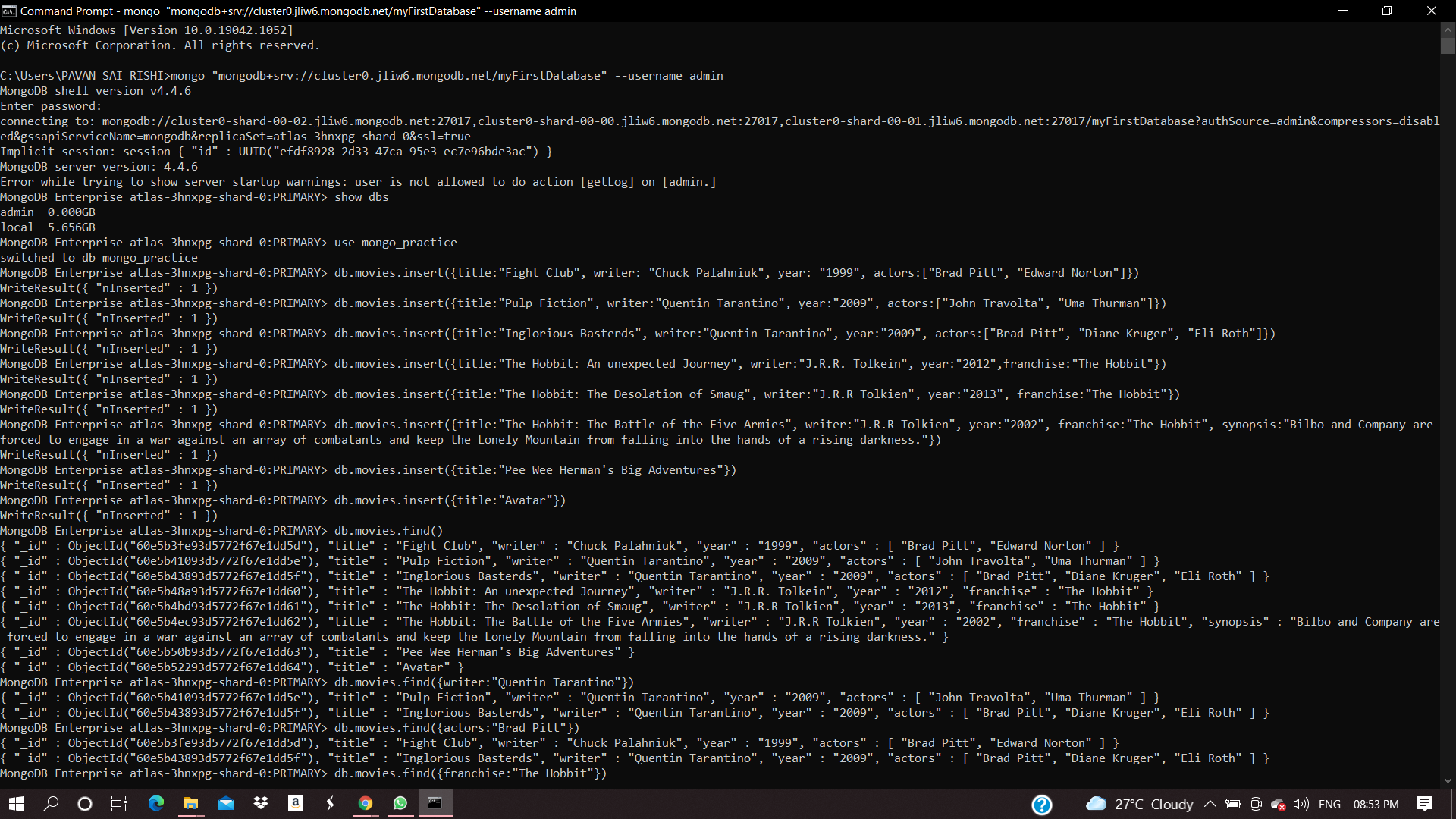
"title" : "Reports a bug in your code",

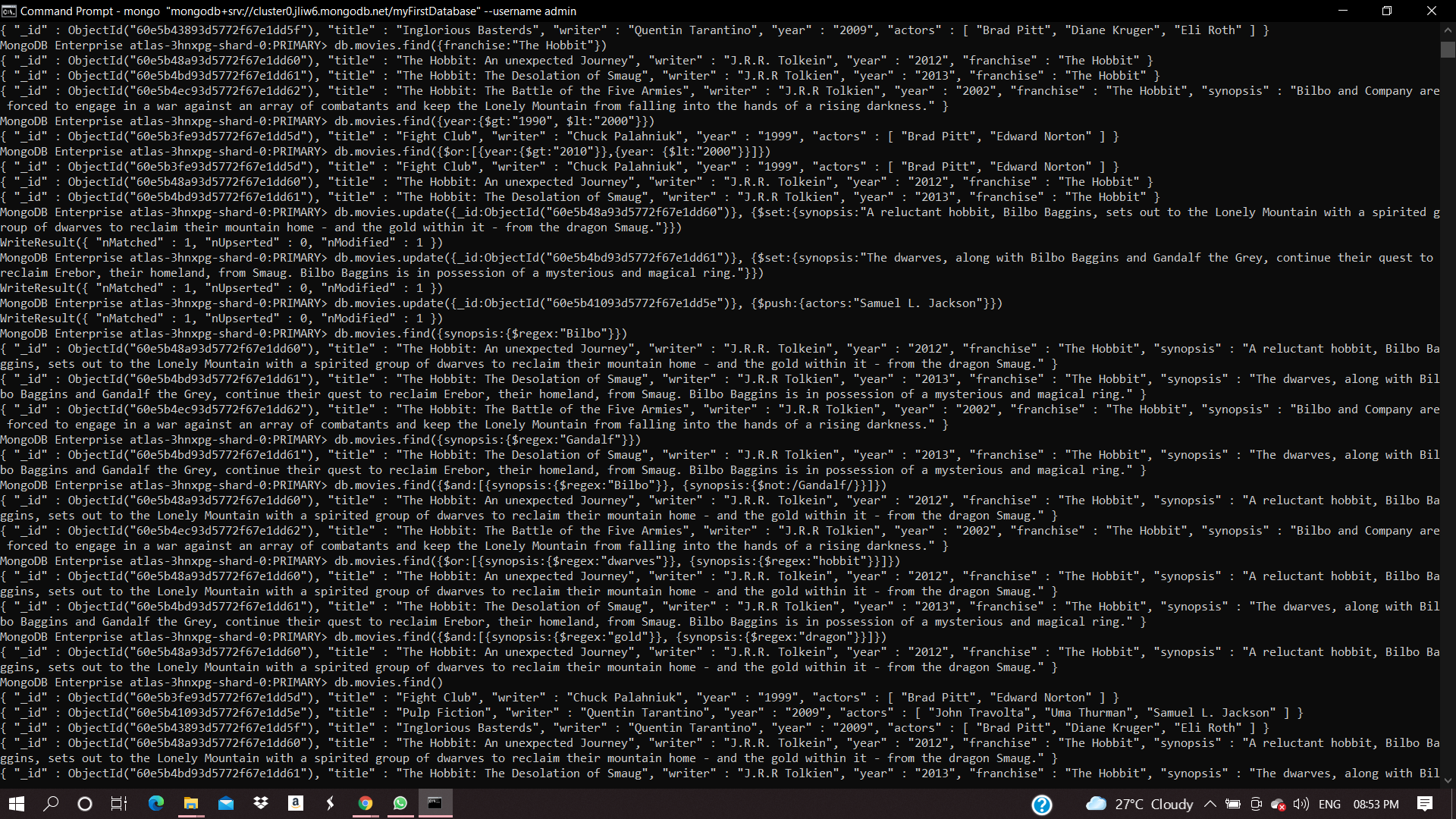
"body" : "Sends you a pull request",

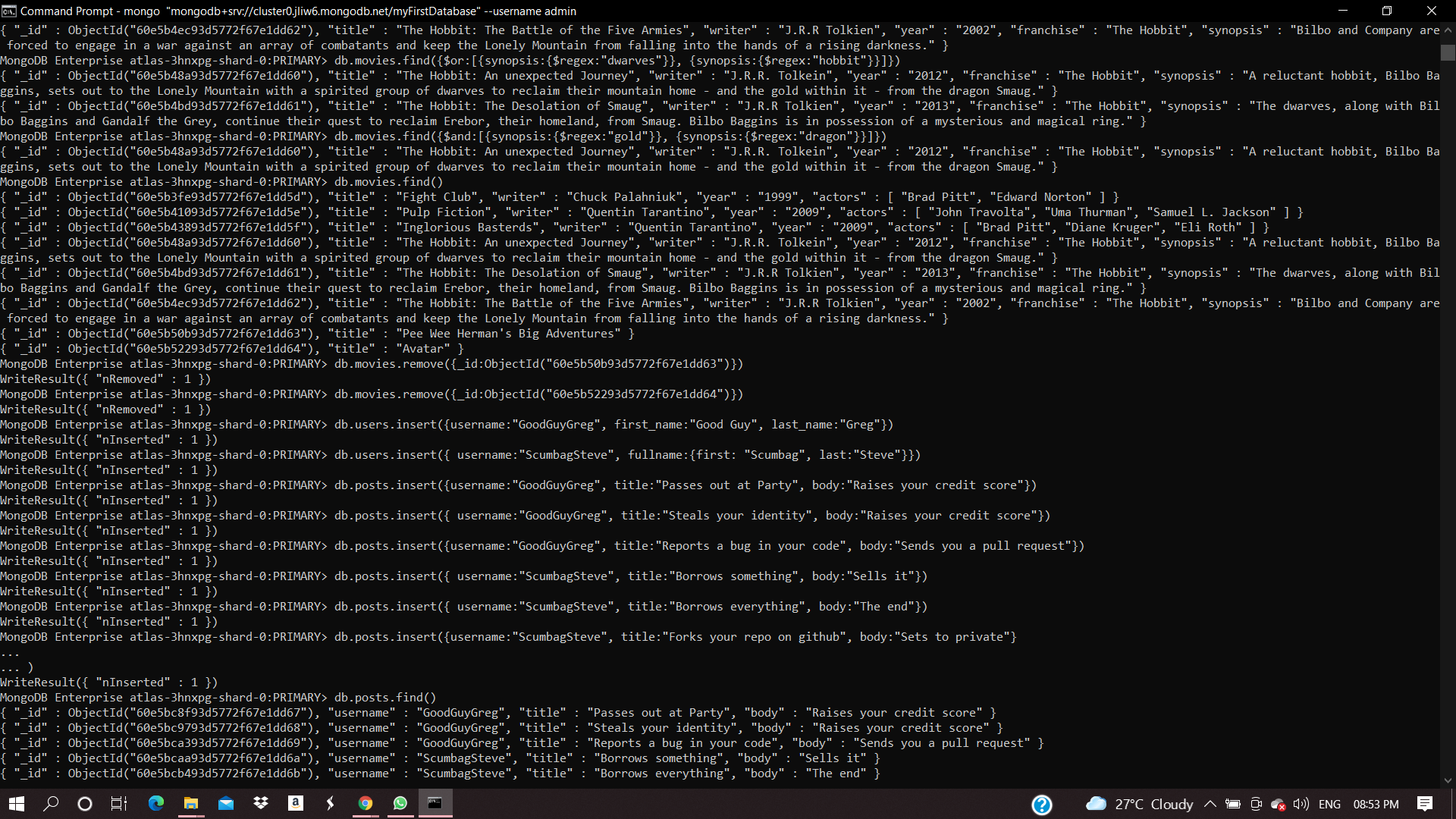
"comments" : [ ]

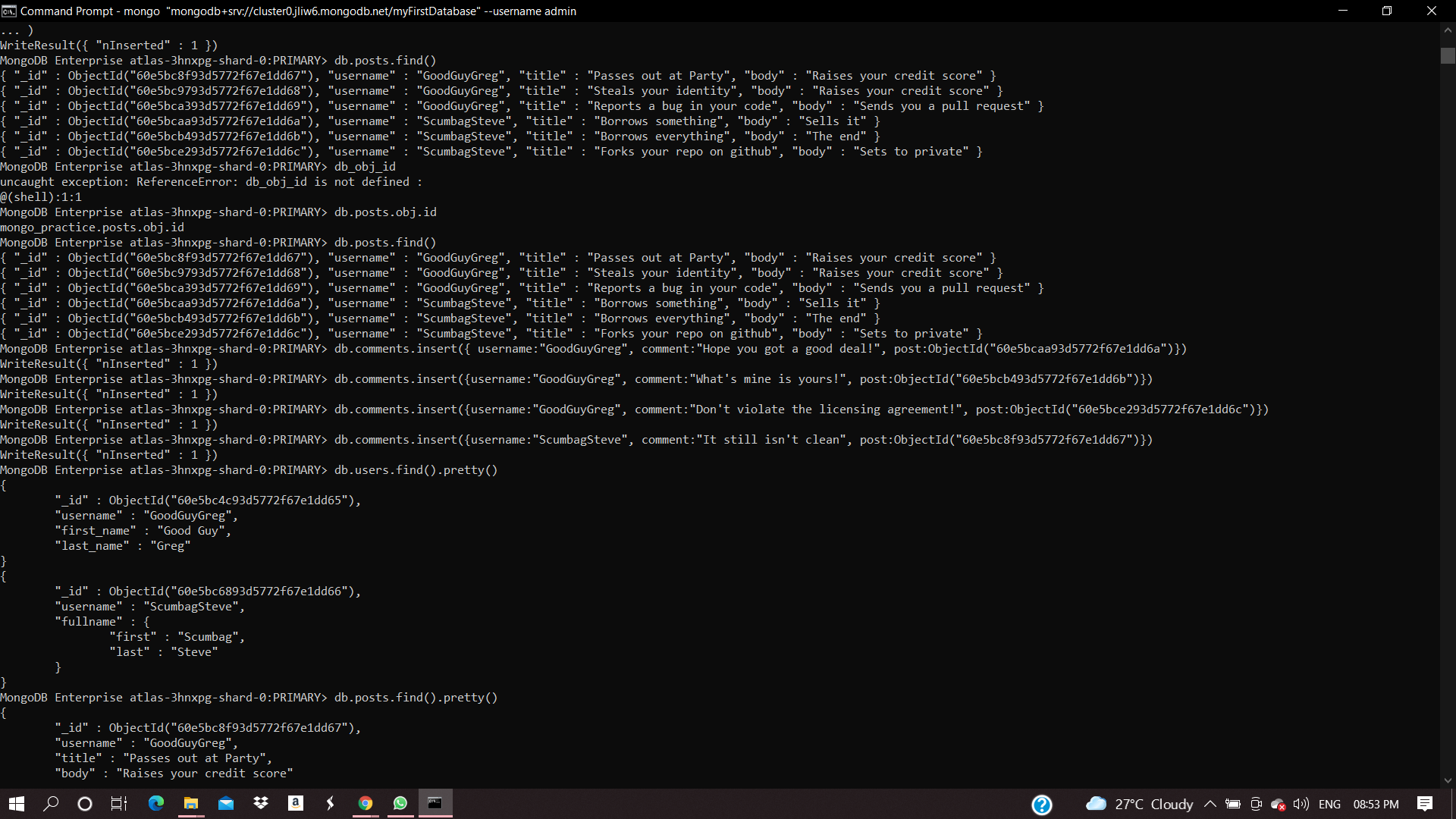
}

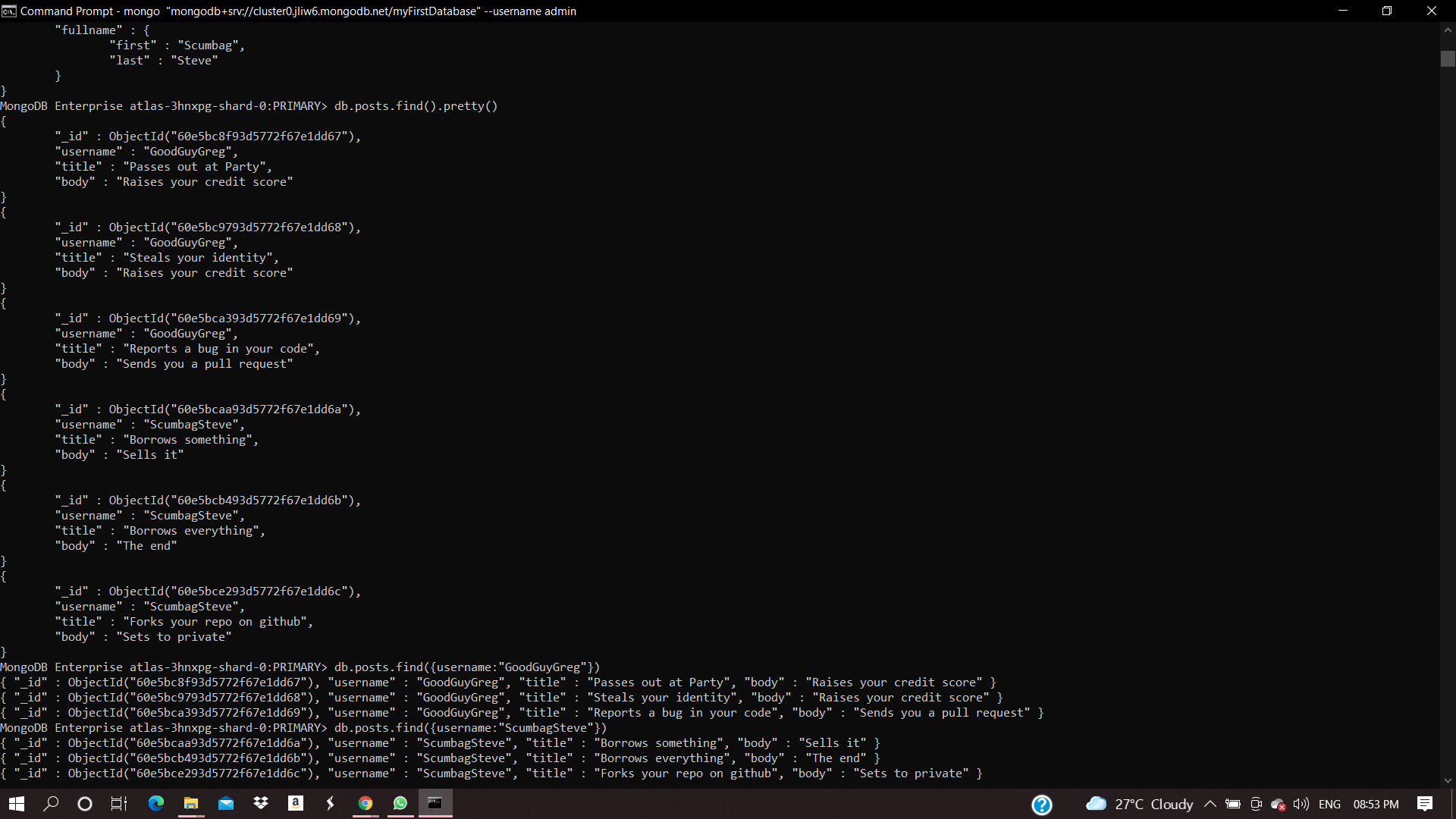
MongoDB Enterprise atlas-3hnxpg-shard-0:PRIMARY>

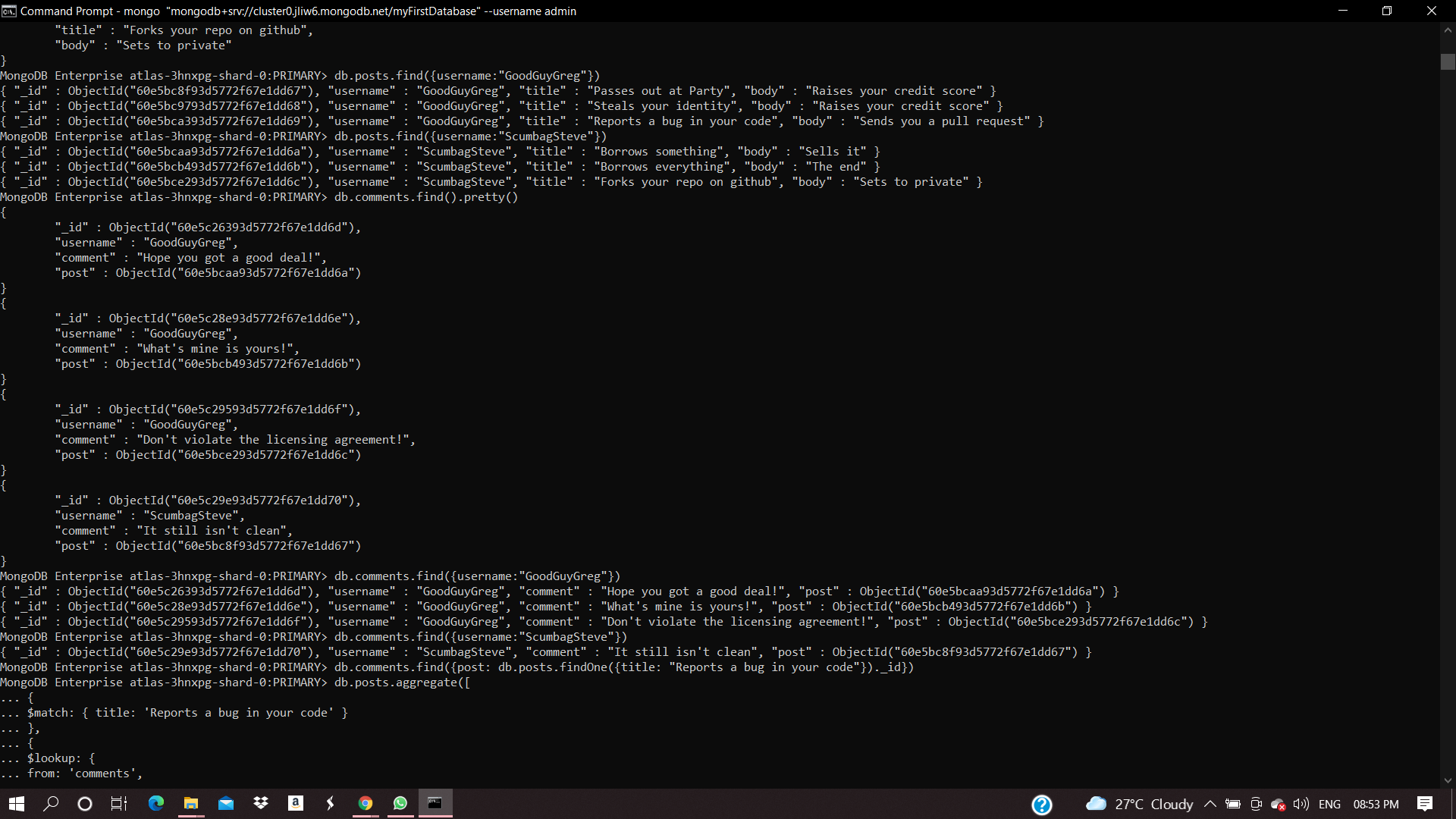
****

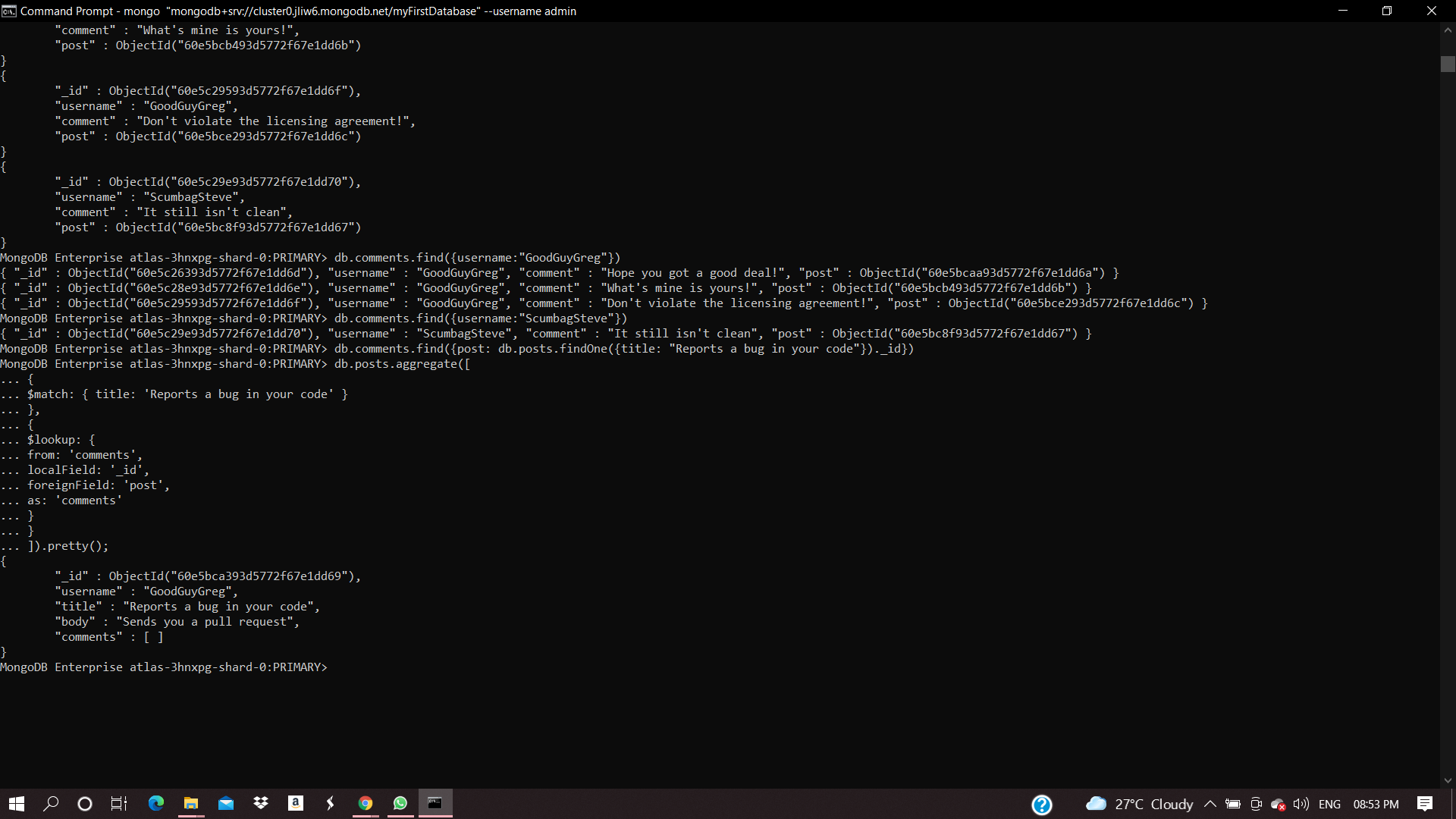
****

****

****

****

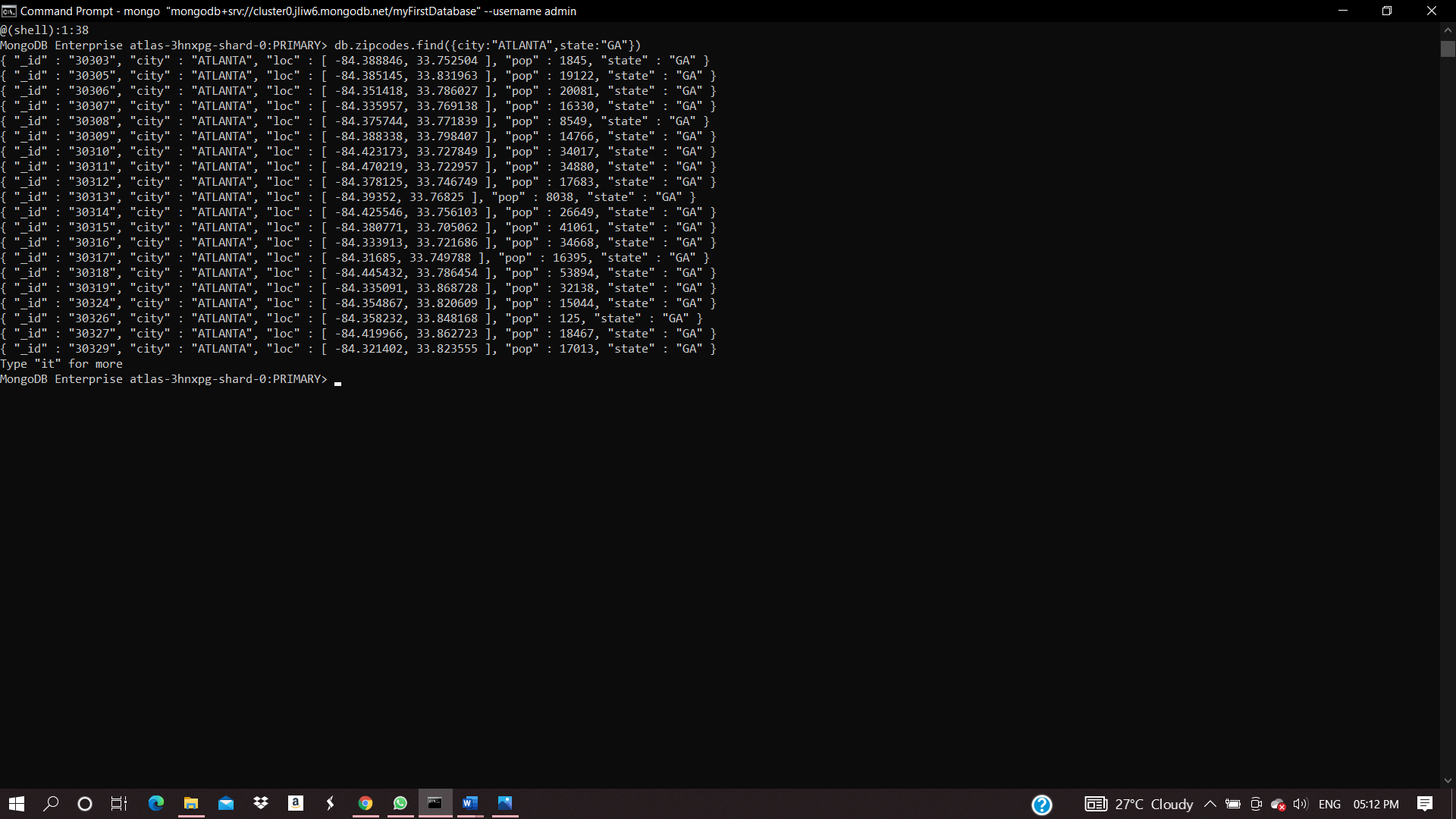
****

****

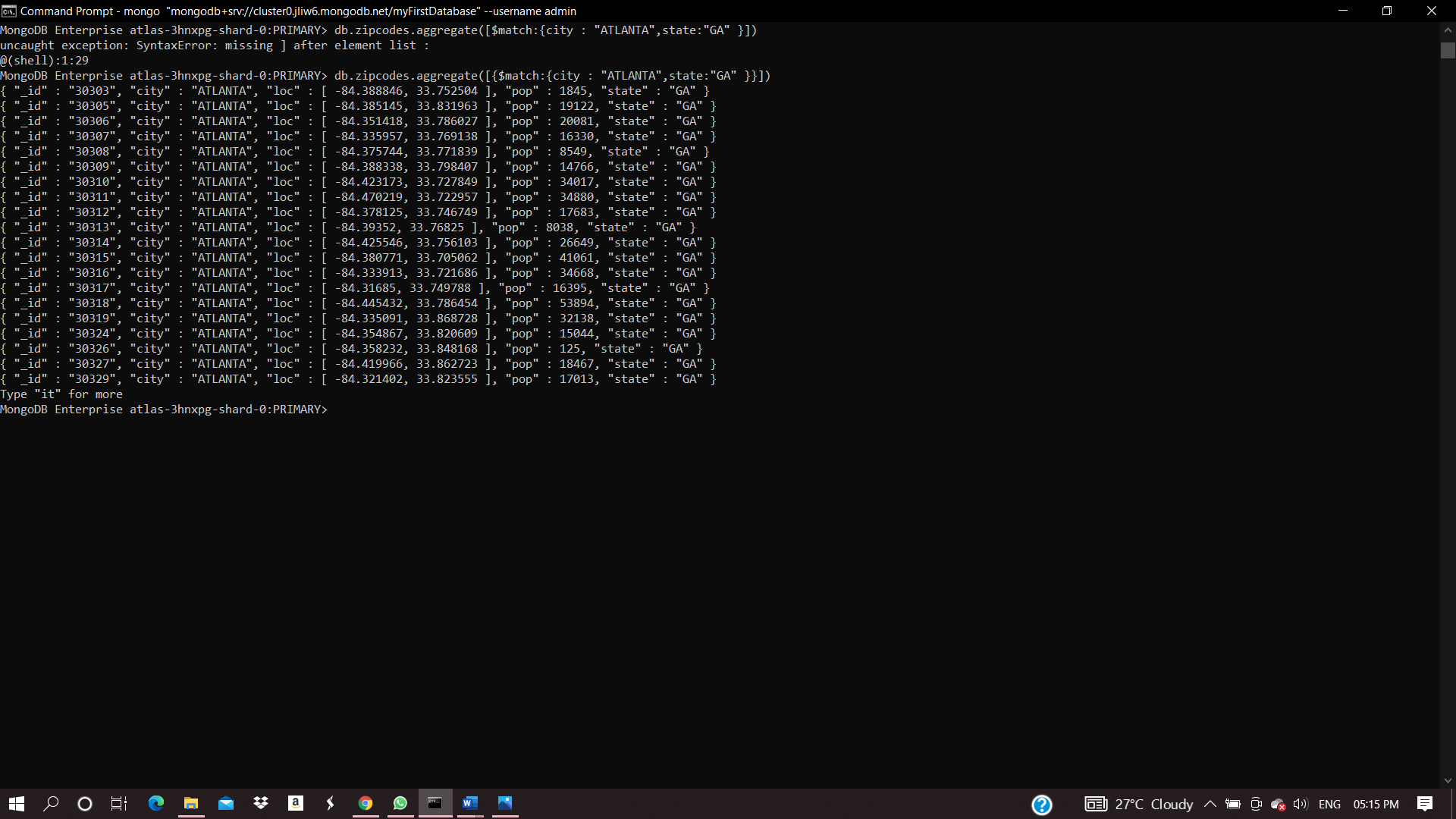
**Assignment-2)**

**Atlanta Population**

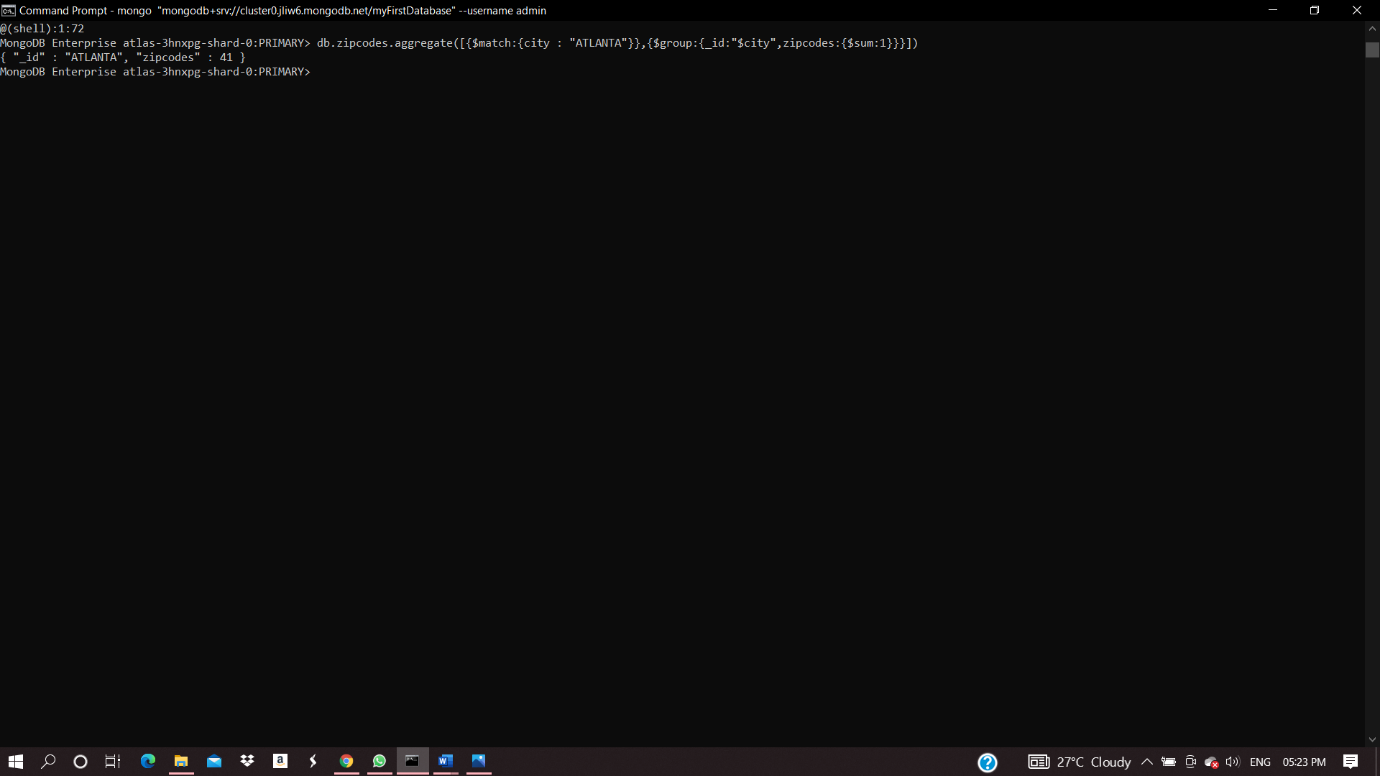
1. db.zipcodes.find({city:"ATLANTA",state:"GA"})



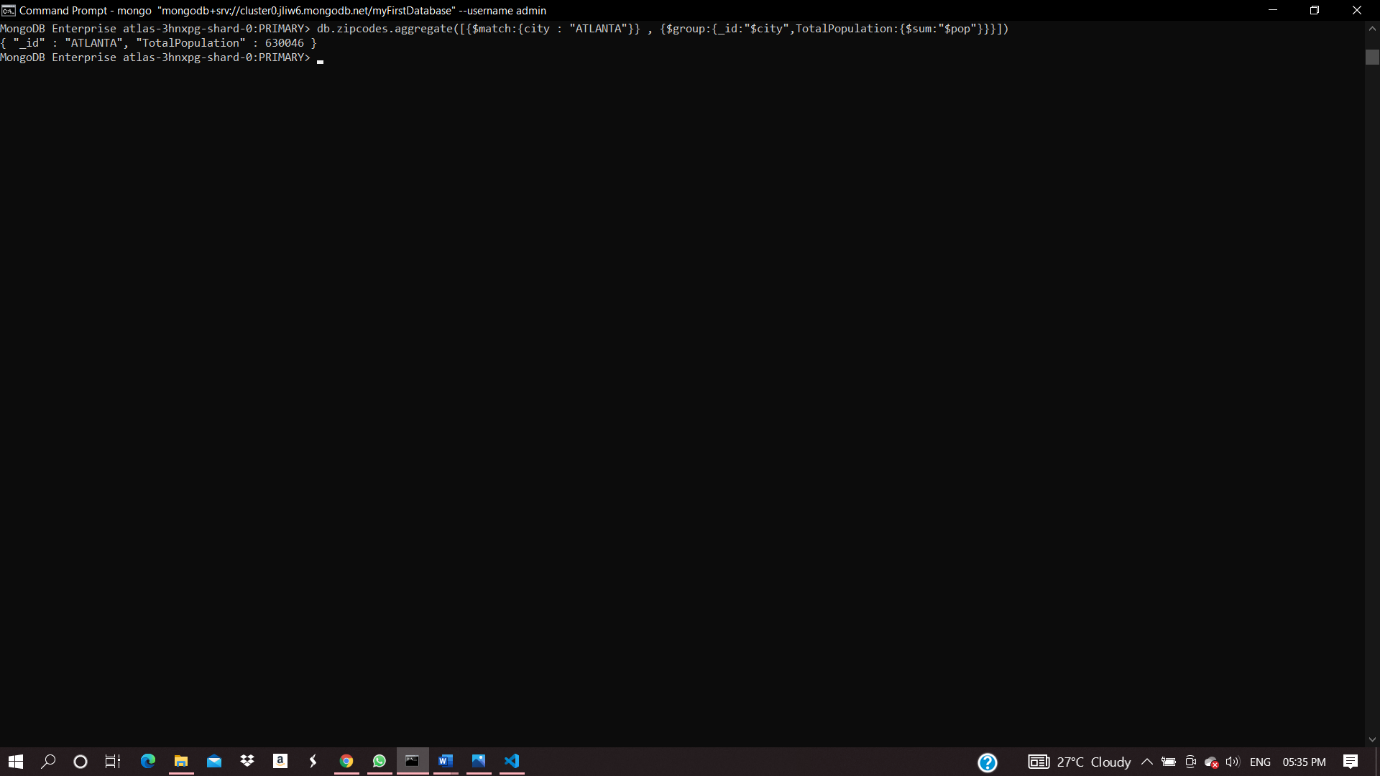
1. db.zipcodes.aggregate([{$match:{city : "ATLANTA",state:"GA" }}])



1. db.zipcodes.aggregate([{$match:{city : "ATLANTA"}},{$group:{\_id:"$city",zipcodes:{$sum:1}}}])

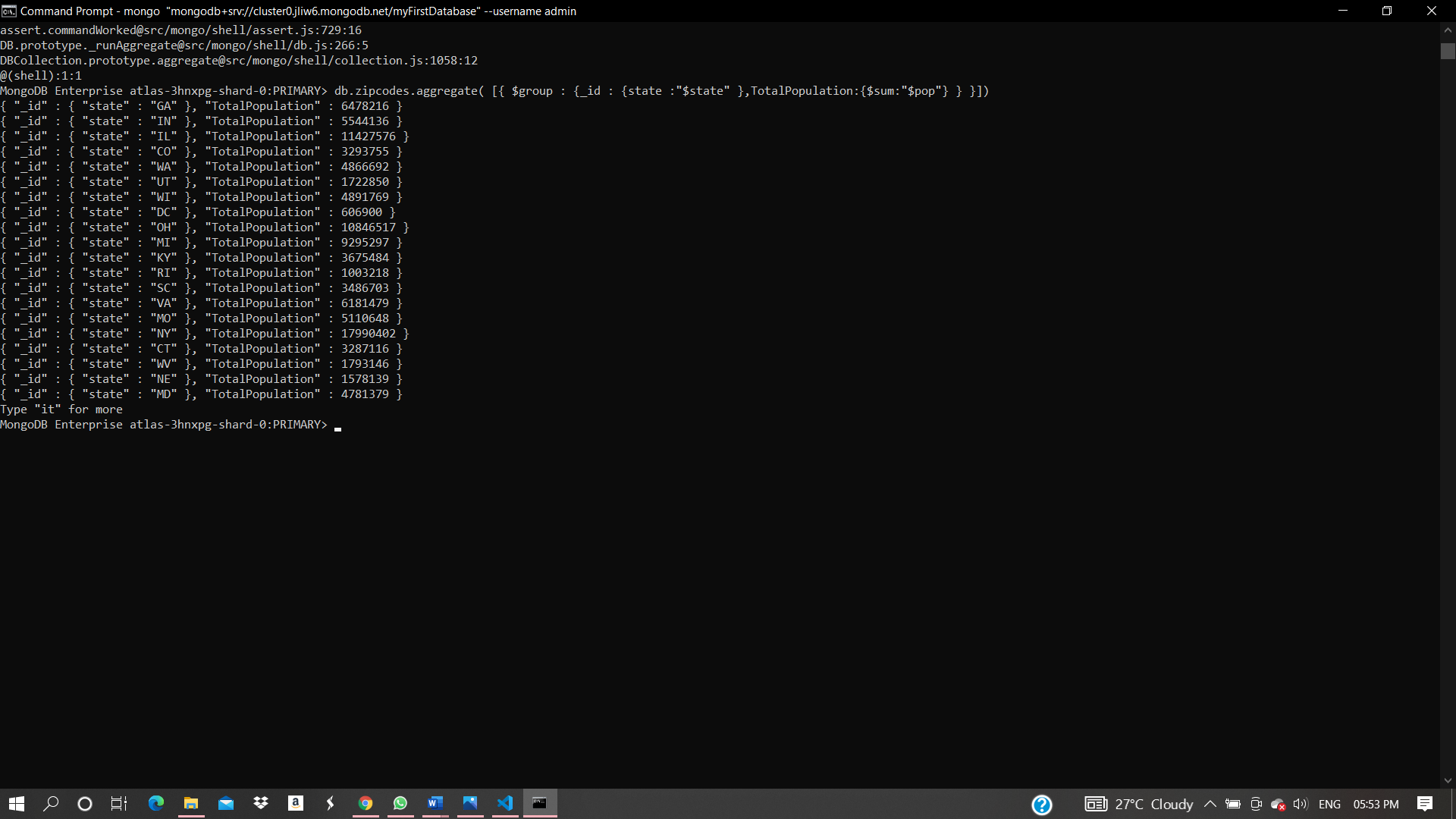


1. db.zipcodes.aggregate([{$match:{city : "ATLANTA"}} , {$group:{\_id:"$city",TotalPopulation:{$sum:"$pop"}}}])

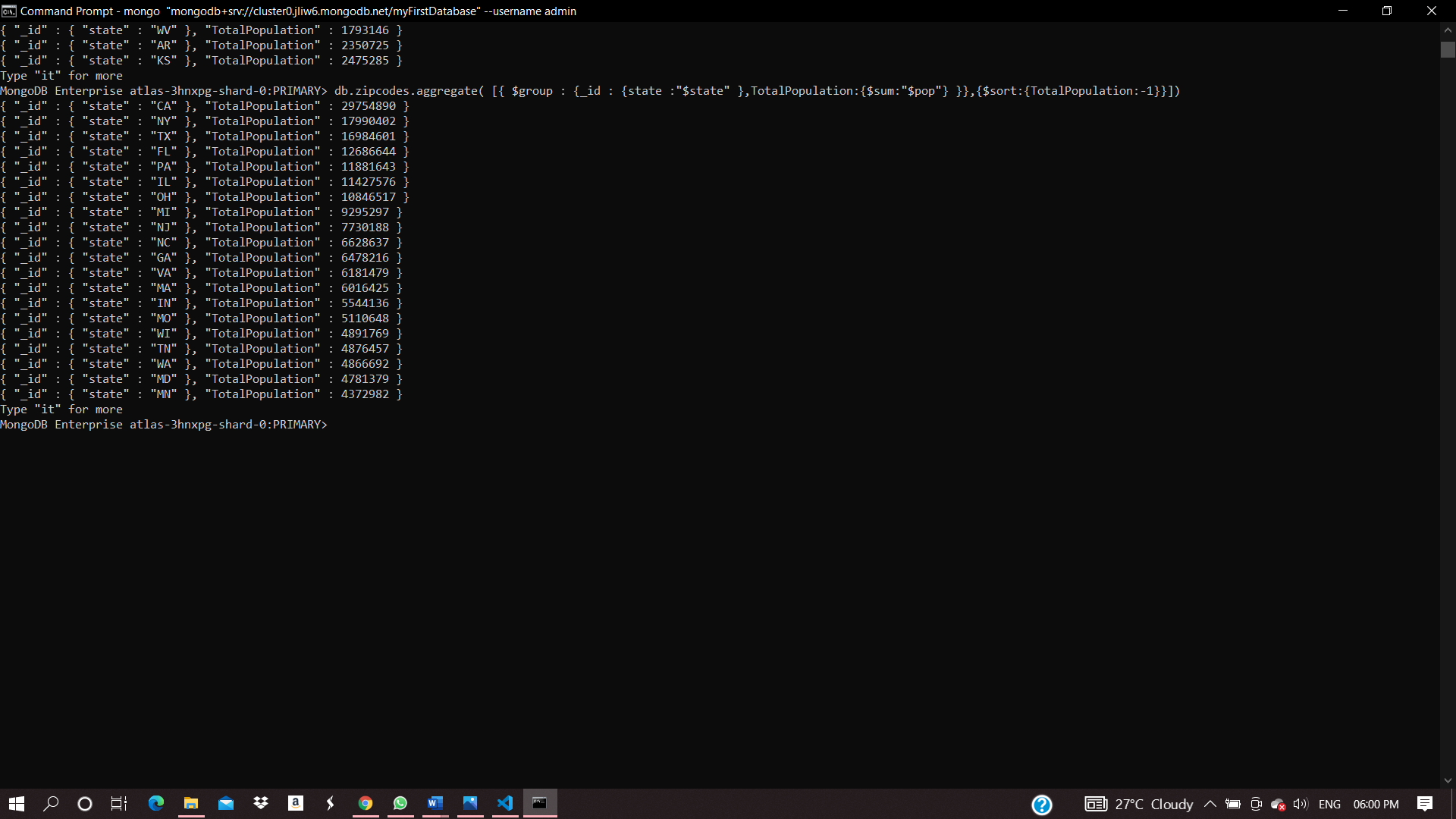


**Populations By State :**

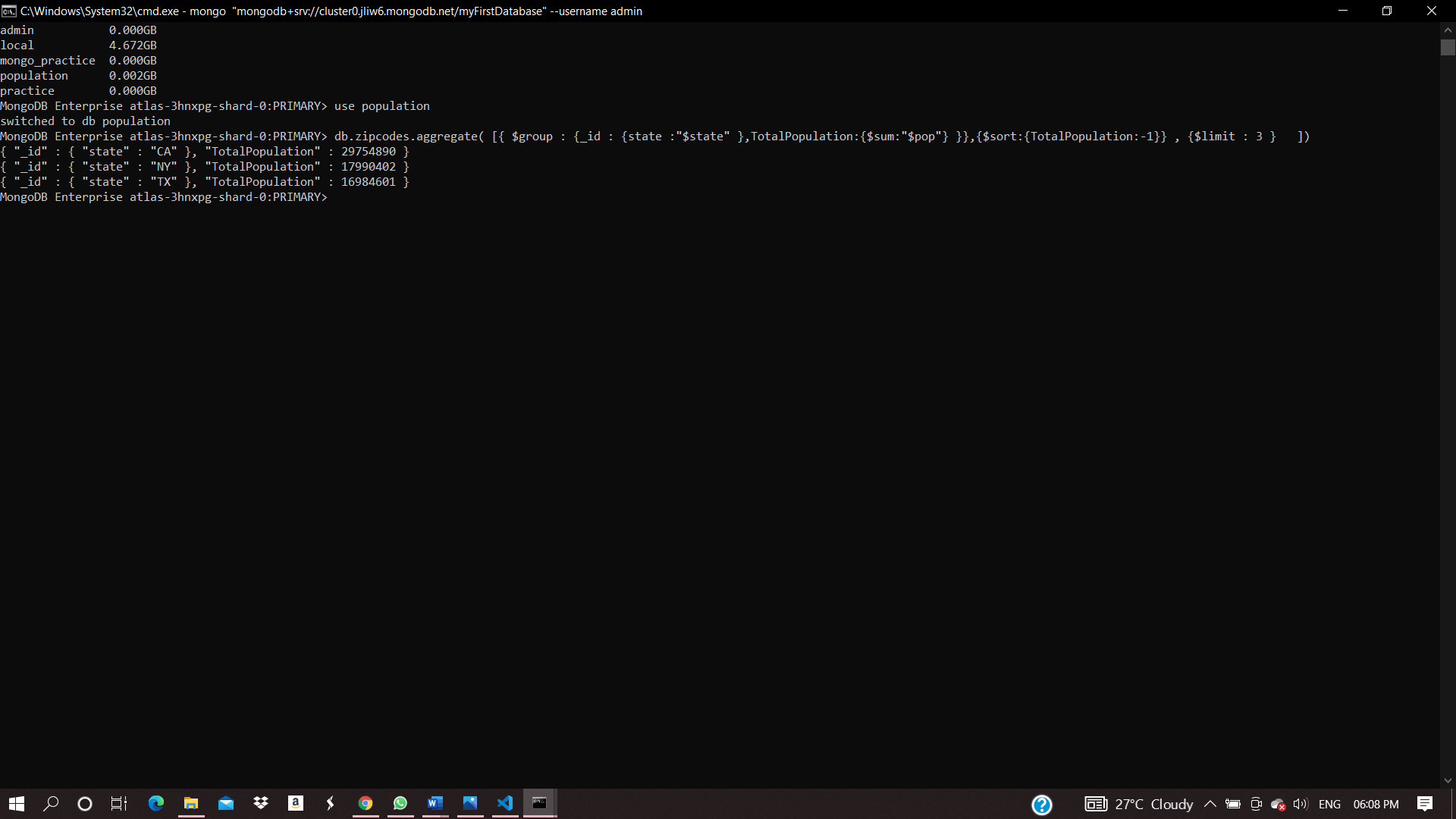
1) db.zipcodes.aggregate( [{ $group : {\_id : {state :"$state"},TotalPopulation:{$sum:"$pop"} } }])



2) db.zipcodes.aggregate( [{ $group : {\_id : {state :"$state" },TotalPopulation:{$sum:"$pop"} }},{$sort:{TotalPopulation:-1}}])



3) db.zipcodes.aggregate( [{ $group : {\_id : {state :"$state" },TotalPopulation:{$sum:"$pop"} }},{$sort:{TotalPopulation:-1}} , {$limit : 3 } ])

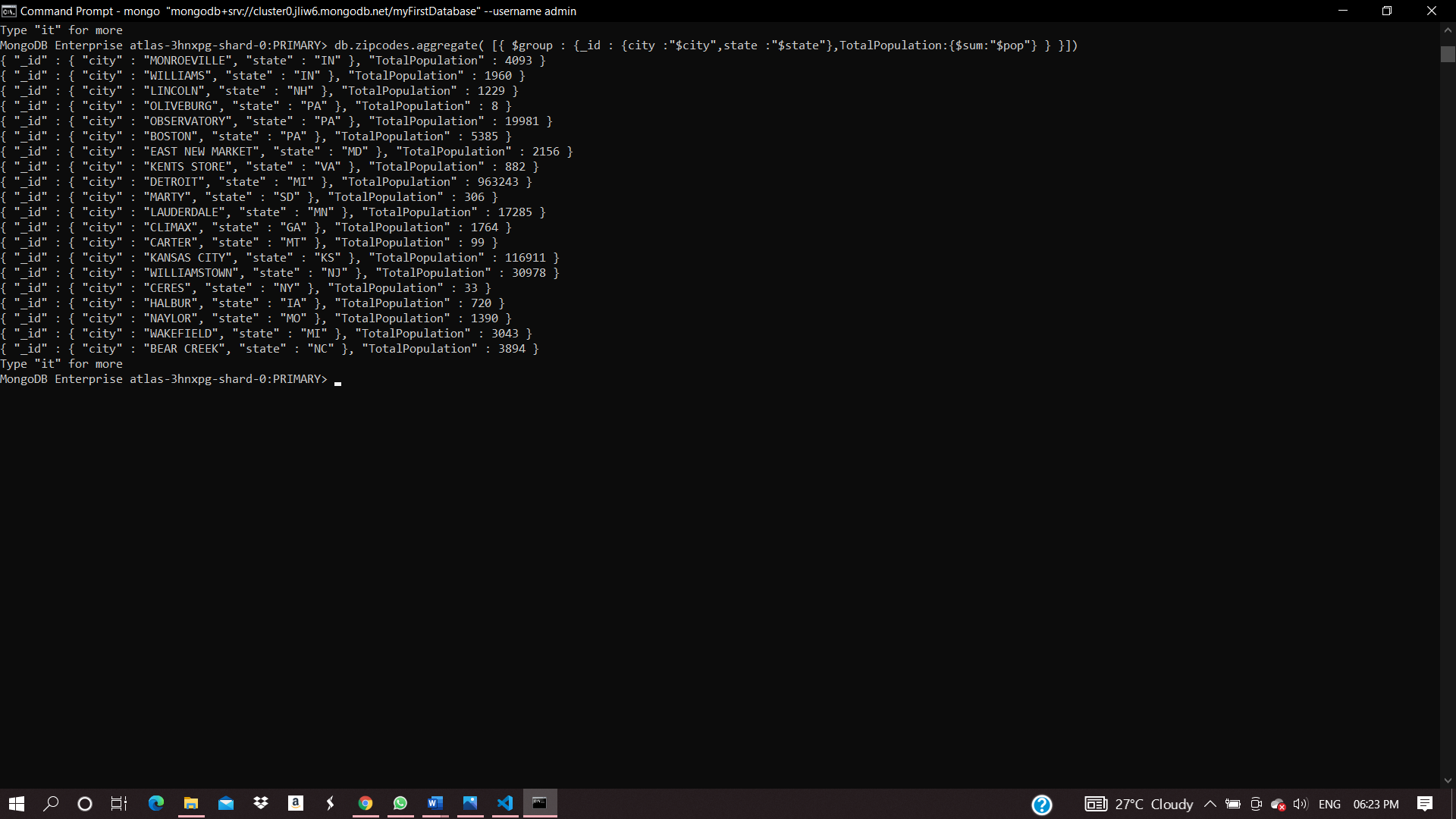


**Populations by city :**

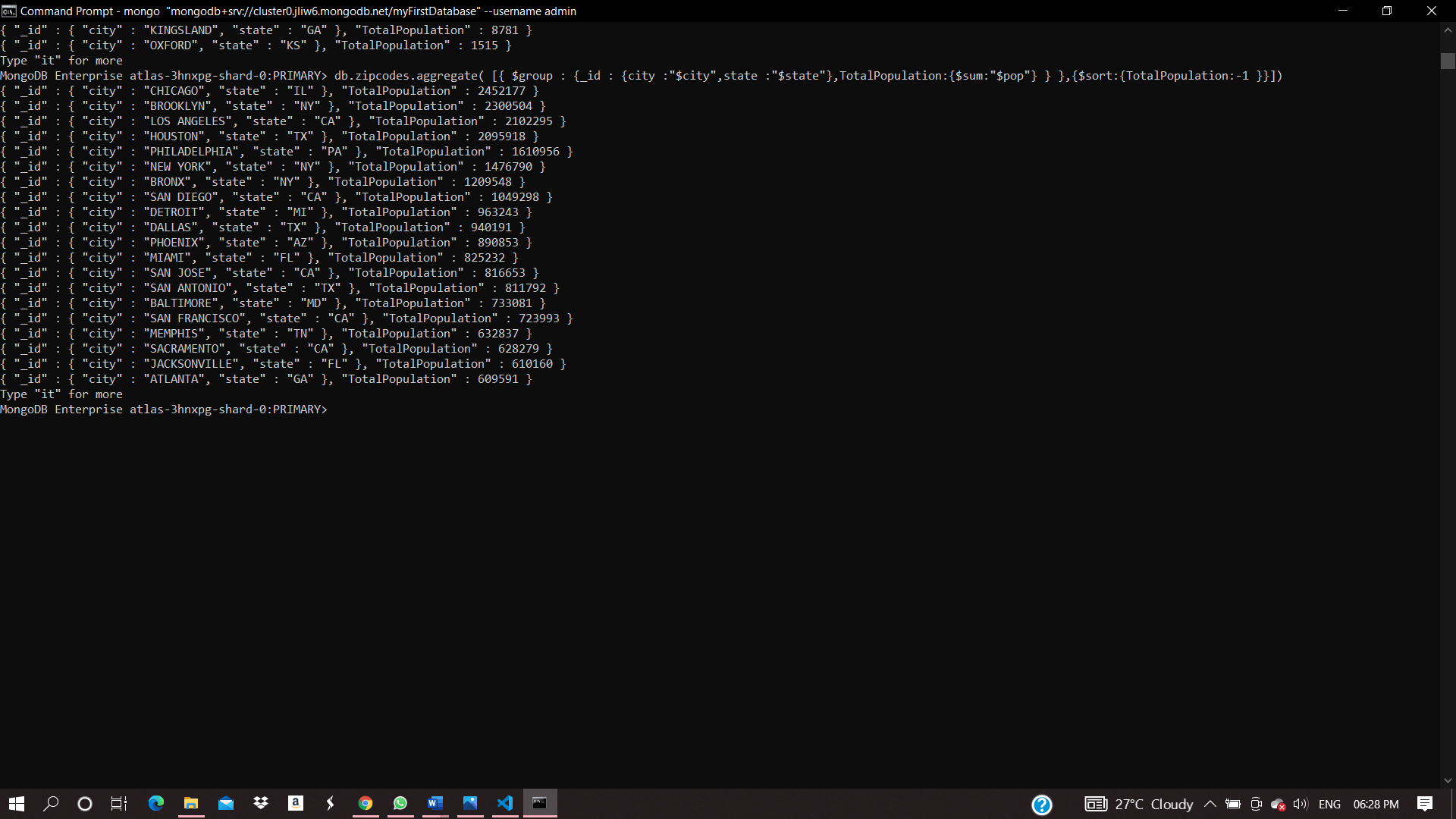
1) db.zipcodes.aggregate( [{

$group : {\_id : {city :"$city",state :"$state"},

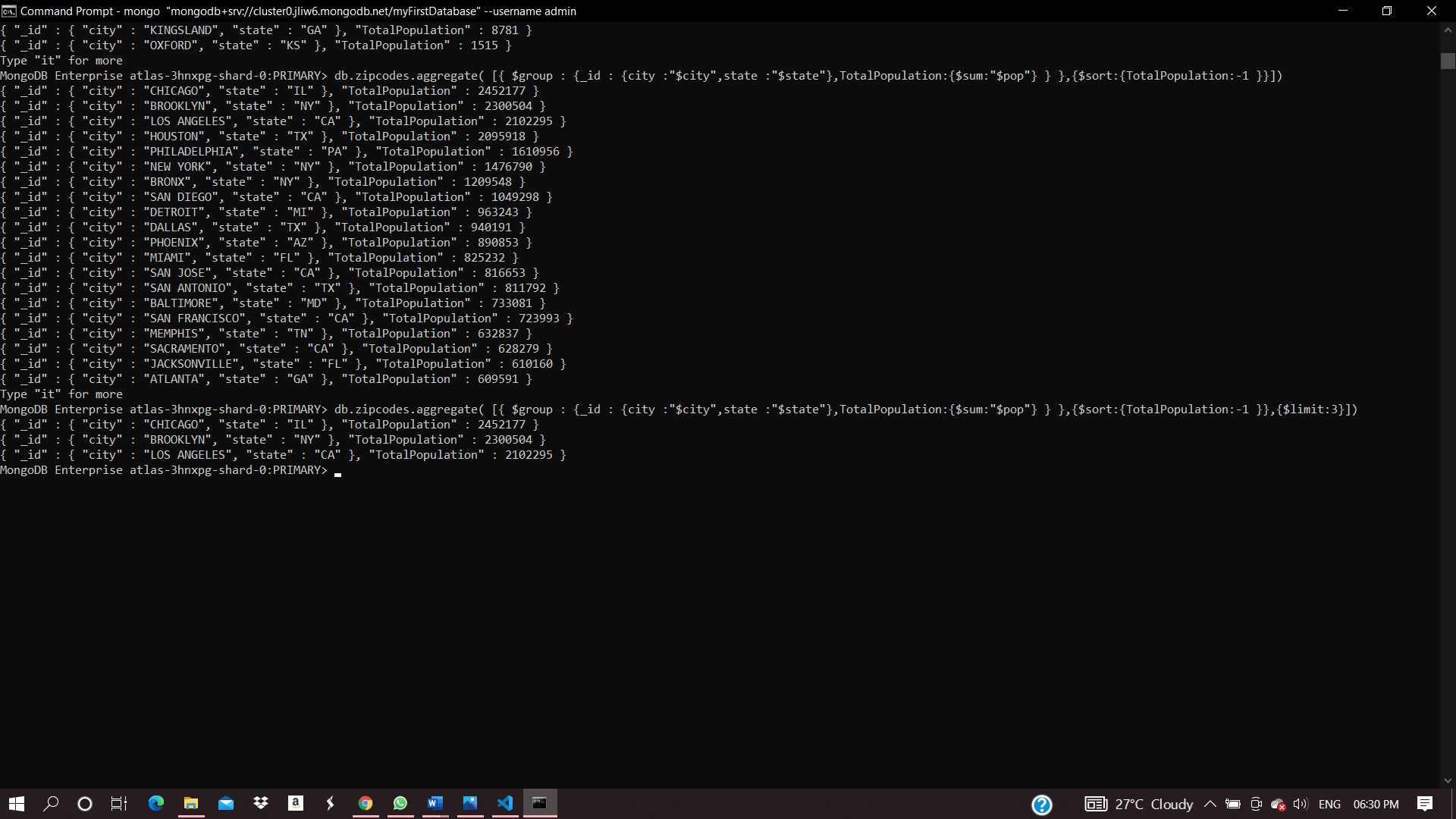
TotalPopulation:{$sum:"$pop"} } }])

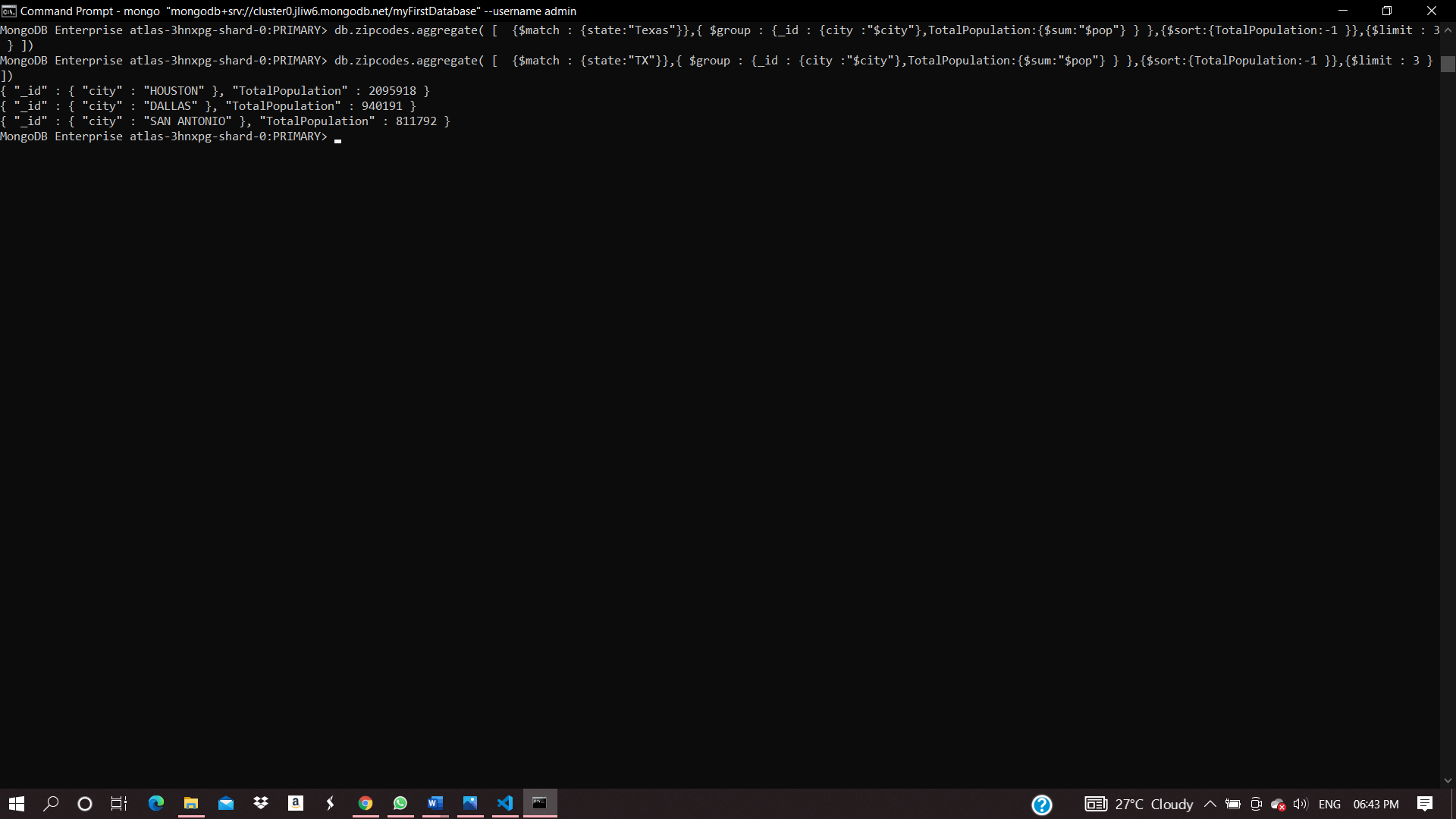


2) db.zipcodes.aggregate( [{ $group : {\_id : {city :"$city",state :"$state"},TotalPopulation:{$sum:"$pop"} } },{$sort:{TotalPopulation:-1 }}])



3) db.zipcodes.aggregate( [{ $group : {\_id : {city :"$city",state :"$state"},TotalPopulation:{$sum:"$pop"} } },{$sort:{TotalPopulation:-1 }},{$limit : 3 } ])

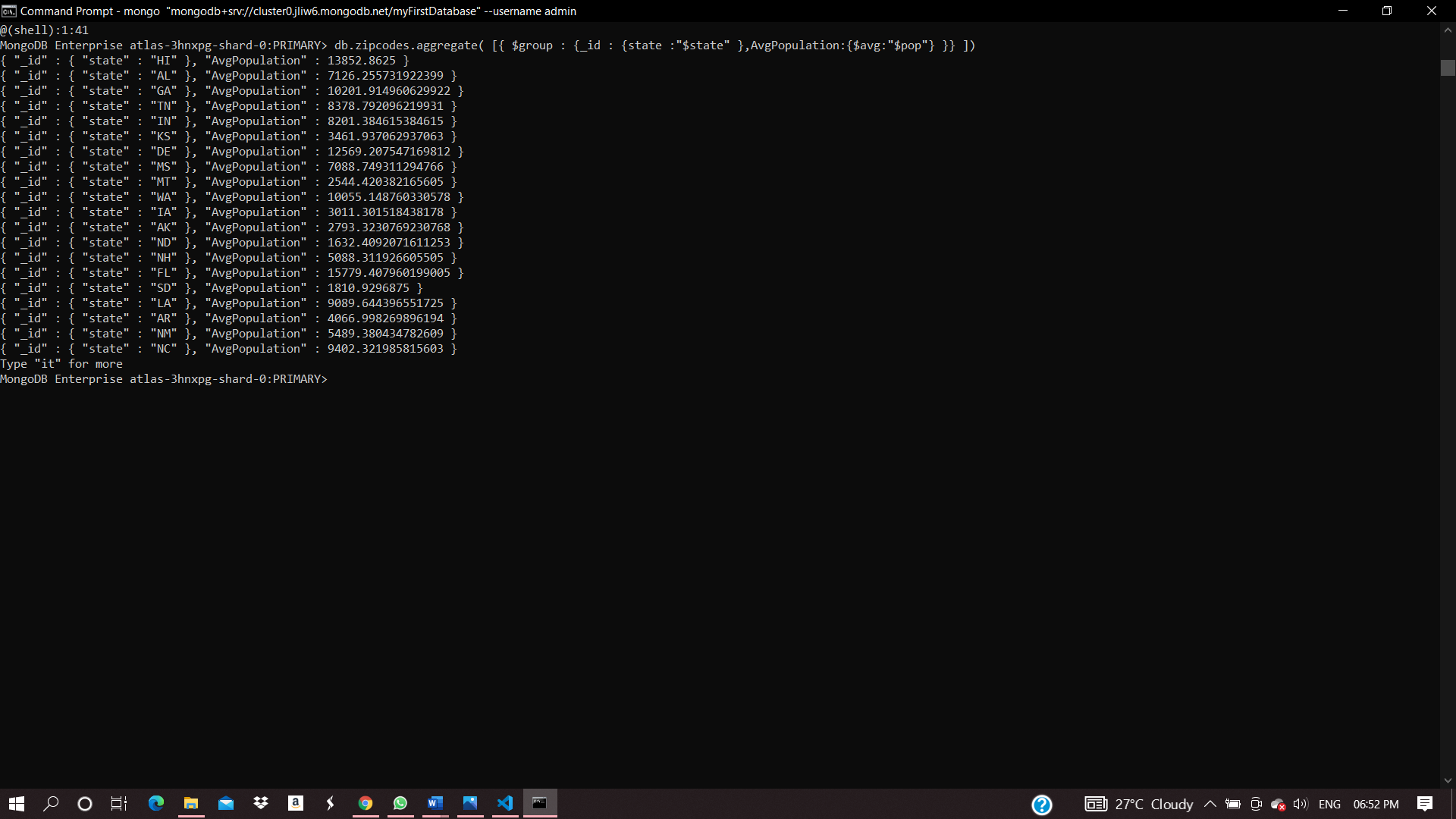


4)db.zipcodes.aggregate( [ {$match : {state:"TX"}},{ $group : {\_id : {city :"$city"},TotalPopulation:{$sum:"$pop"} } },{$sort:{TotalPopulation:-1 }},{$limit : 3 } ])

**Bonus**

1)

db.zipcodes.aggregate( [{ $group : {\_id : {state :"$state" },AvgPopulation:{$avg:"$pop"} }} ])



2) db.zipcodes.aggregate( [{ $group : {\_id : {state :"$state" },AvgPopulation:{$avg:"$pop"} }},{$sort:{AvgPopulation:-1}} , {$limit : 3 } ])

