Materia Bases de datos avanzadas

PROFESOR: WILLIAM RUIZ

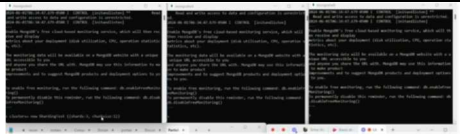
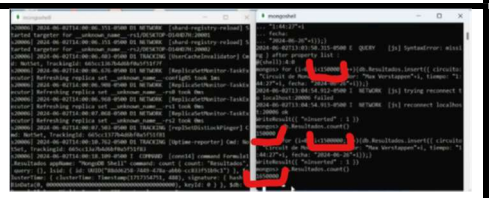
Universidad Iberoamericana

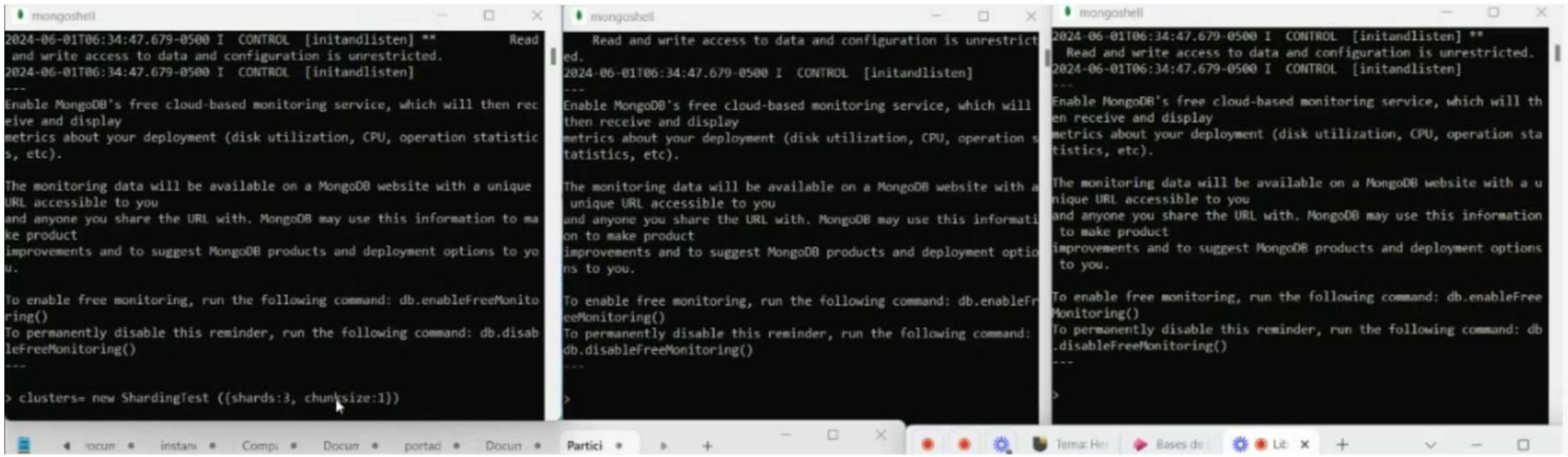
Ingeniería en Ciencia de Datos

Actividad 4 - Pruebas de particionamiento de bases de datos No SQL

Nelson Fernando Nopssa Castro

ID 100169351

ID	CASO DE PRUEBA	DESCRIPCION	FUNCIONALIDAD	RESULTADO ESPERADO	SOPORTE
CASE001	**Verificación de la Creación del Clúster de Sharding**	Confirmar que el clúster de Sharding se ha creado correctamente con al menos tres shards	• Iniciar el clúster de Sharding con tres shards y un tamaño de chunk de 1 MB en la consola base o primaria	El clúster de Sharding debe ser inicializado correctamente con tres shards	
CASE002	**Verificación de la Inserción de Datos y Distribución en Shards**	Comprobar que los datos se insertan correctamente y se distribuyen entre los shards	• Conectar al clúster y realizar la inserción masiva de datos en la consola cliente	Se deben insertar 1,650,000 documentos y el conteo debe reflejarlo. Para este caso se hizo una prueba inicial con 150.000 y después con 1500.000	
CASE003	**Verificación de Operaciones Sin Contratiempos**	Asegurar que las operaciones de inserción, consulta y modificación de datos se realizan sin interrupciones.	• Realizar operaciones de inserción, consulta y modificación en la base de datos. • Verificar que todas las operaciones se completan con éxito y sin errores.	Todas las operaciones deben completarse sin errores y con un tiempo de respuesta adecuado.	


mongoshell
2024-06-01T06:34:47.679-0500 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2024-06-01T06:34:47.679-0500 I CONTROL [initandlisten]

Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).
The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.
To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

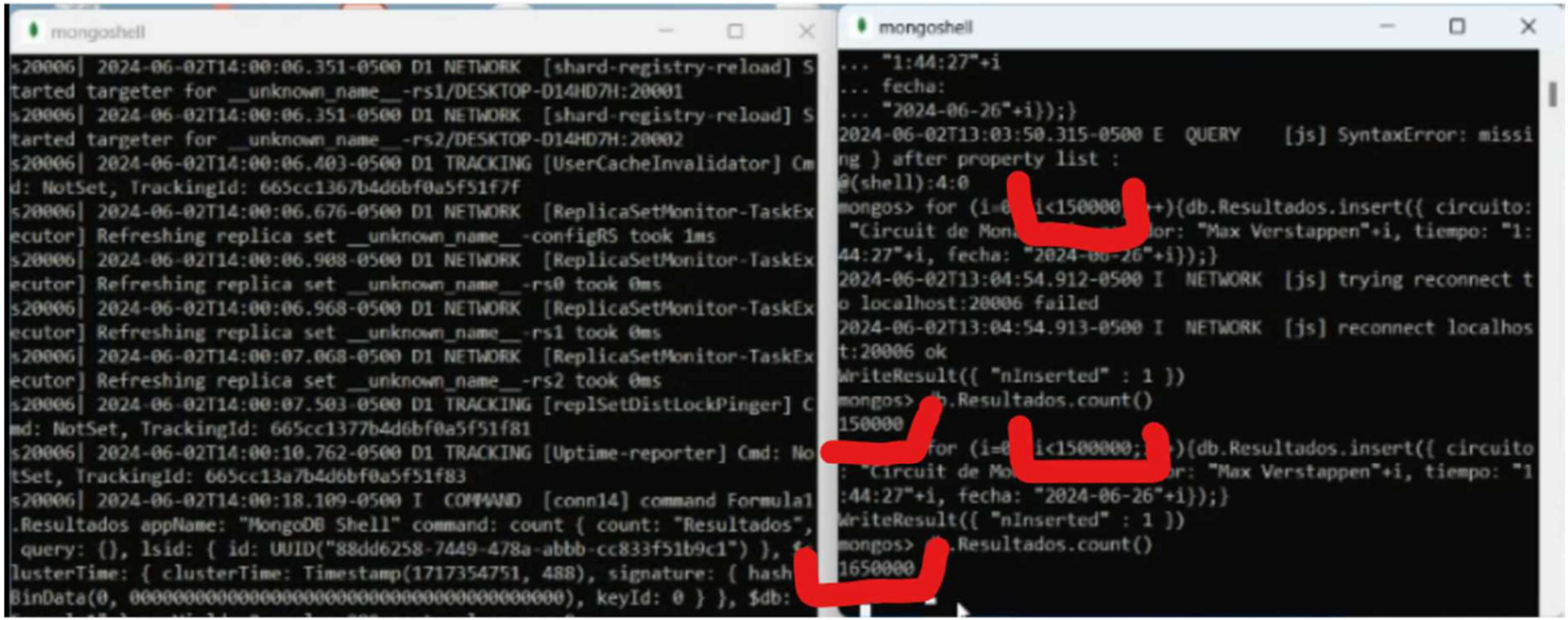
> clusters= new ShardingTest ({shards:3, chunksize:1})
mongoshell
Read and write access to data and configuration is unrestricted.
2024-06-01T06:34:47.679-0500 I CONTROL [initandlisten]

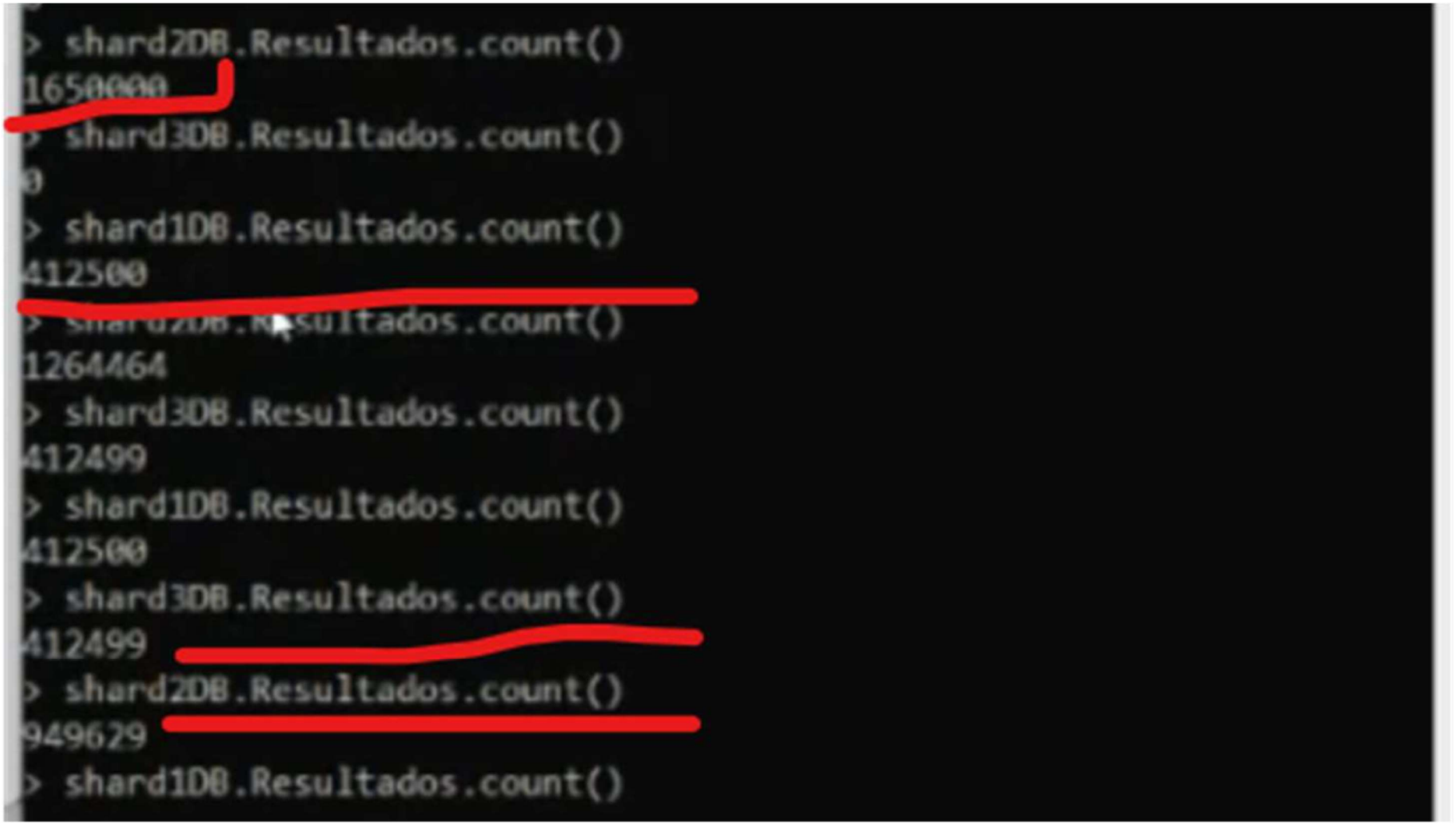
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).
The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.
To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

>
mongoshell
2024-06-01T06:34:47.679-0500 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2024-06-01T06:34:47.679-0500 I CONTROL [initandlisten]

Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).
The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.
To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

>


mongoshell
s20006| 2024-06-02T14:00:06.351-0500 D1 NETWORK [shard-registry-reload] S
tarted targeter for __unknown_name__-rs1/DESKTOP-D14HD7H:20001
s20006| 2024-06-02T14:00:06.351-0500 D1 NETWORK [shard-registry-reload] S
tarted targeter for __unknown_name__-rs2/DESKTOP-D14HD7H:20002
s20006| 2024-06-02T14:00:06.403-0500 D1 TRACKING [UserCacheInvalidator] Cm
d: NotSet, TrackingId: 665cc1367b4d6bf0a5f51f7f
s20006| 2024-06-02T14:00:06.676-0500 D1 NETWORK [ReplicaSetMonitor-TaskEx
ecutor] Refreshing replica set __unknown_name__-configRS took 1ms
s20006| 2024-06-02T14:00:06.908-0500 D1 NETWORK [ReplicaSetMonitor-TaskEx
ecutor] Refreshing replica set __unknown_name__-rs0 took 0ms
s20006| 2024-06-02T14:00:06.968-0500 D1 NETWORK [ReplicaSetMonitor-TaskEx
ecutor] Refreshing replica set __unknown_name__-rs1 took 0ms
s20006| 2024-06-02T14:00:07.068-0500 D1 NETWORK [ReplicaSetMonitor-TaskEx
ecutor] Refreshing replica set __unknown_name__-rs2 took 0ms
s20006| 2024-06-02T14:00:07.503-0500 D1 TRACKING [replSetDistLockPinger] C
md: NotSet, TrackingId: 665cc1377b4d6bf0a5f51f81
s20006| 2024-06-02T14:00:10.762-0500 D1 TRACKING [Uptime-reporter] Cmd: No
tSet, TrackingId: 665cc13a7b4d6bf0a5f51f83
s20006| 2024-06-02T14:00:18.109-0500 I COMMAND [conn14] command Formula1
.Resultados appName: "MongoDB Shell" command: count { count: "Resultados",
query: {}, lsid: { id: UUID("88dd6258-7449-478a-abbb-cc833f51b9c1") }, $c
lusterTime: { clusterTime: Timestamp(1717354751, 488), signature: { hash
BinData(0, 0000000000000000000000000000000000000000), keyId: 0 } }, $db:
mongoshell
... "1:44:27"+i
... fecha:
... "2024-06-26"+i});}
2024-06-02T13:03:50.315-0500 E QUERY [js] SyntaxError: missi
ng) after property list :
@(shell):4:0
mongos> for (i=0; i<150000; i++){db.Resultados.insert({ circuito:
"Circuit de Mona... dor: "Max Verstappen"+i, tiempo: "1:
44:27"+i, fecha: "2024-06-26"+i});}
2024-06-02T13:04:54.912-0500 I NETWORK [js] trying reconnect t
o localhost:20006 failed
2024-06-02T13:04:54.913-0500 I NETWORK [js] reconnect localhos
t:20006 ok
WriteResult({ "nInserted" : 1 })
mongos> db.Resultados.count()
150000
mongos> for (i=0; i<1500000; i++){db.Resultados.insert({ circuito
: "Circuit de Mo... dor: "Max Verstappen"+i, tiempo: "1
:44:27"+i, fecha: "2024-06-26"+i});}
WriteResult({ "nInserted" : 1 })
mongos> db.Resultados.count()
1650000


> shard2DB.Resultados.count()
1650000
> shard3DB.Resultados.count()
0
> shard1DB.Resultados.count()
412500
> shard2DB.Resultados.count()
1264464
> shard3DB.Resultados.count()
412499
> shard1DB.Resultados.count()
412500
> shard3DB.Resultados.count()
412499
> shard2DB.Resultados.count()
949629
> shard1DB.Resultados.count()