Ejemplo:

db.createCollection(“SampleDataCollection”)  
db.SampleDataCollection.insert([  
{  \_id : 1, episodeName : “Dragonstone”, directedBy : “Jeremy Podeswa”, writtenBy : “David Benioff, D.B.Weiss”, airedDate : “July2017”},  
{  \_id : 2, episodeName: “Stormborn”, directedBy : “Mark Mylod”, writtenBy : “Bryan Cogman”, airedDate : “July2017”},  
{ \_id : 3, episodeName: “The Queen’s Justice”, directedBy : “Mark Mylod”, writtenBy : “David Benioff, D.B.Weiss”, airedDate : “July2017”},  
{  \_id : 4, episodeName: “The Spoils of War”, directedBy : “Matt Shakman”, writtenBy : “David Benioff, D.B.Weiss”, airedDate: “August2017”},  
{  \_id : 5, episodeName: “Eastwatch”, directedBy : “Matt Shakman”, writtenBy : “Bryan Cogman”, airedDate: ”August2017”}  
])

|  |  |
| --- | --- |
| **Operación** | **Descripción & Ejemplo** |
| db.collection.find() | Esto es equivalente a un query READ con una cláusula especificada WHERE con una condición    Query: db.SampleDataCollection.find()  RDMS Equivalent: SELECT \* FROM SampleDataCollection  Result:  El query anterior va a devolver todos los datos de la table como resultado, ya que no hay ninguna condición que satisfaga al READ  Query:  db.SampleDataCollection.find( \_id : 5)    RDBMS Equivalent:  SELECT \* FROM SampleDataCollection WHERE \_id = 5  Result:  {  "\_id" : 5,  "episodeName" : "Eastwatch",  "directedBy" : "Matt Shakman",  "writtenBy" : "Bryan Cogman",  "airedDate" : "August2017"  } |
| db.collection.find(, ) | Esto es equivalente al query READ con un WHERE especifcado y con las columnas a ser recuperadas desde una tabla    Query : db.SampleDataCollection.find( { \_id : 5}, { episodeName : 1, writtenBy : 1})  RDBMS Equivalent :  SELECT \_id, episodeName, writtenBy  FROM SampleDataCollection  WHERE \_id = 5  Result:  {  "\_id" : 5,  "directedBy" : "Matt Shakman",  "writtenBy" : "Bryan Cogman"  } |
| db.collection.find().sort() | Esto es equivalente a un SELECT ordenado con ORDER BY por una cierta columna  Query: db.SampleDataCollection.find().sort( { \_id : -1 })  RDBMS Equivalent:  SELECT \* FROM SampleDataCollection ORDER BY \_id DESC  Result:  /\* 1 \*/  {  "\_id" : 5,  "episodeName" : "Eastwatch",  "directedBy" : "Matt Shakman",  "writtenBy" : "Bryan Cogman",  "airedDate" : "August2017"  },  /\* 2 \*/  {  "\_id" : 4,  "episodeName" : "The Spoils of War",  "directedBy" : "Matt Shakman",  "writtenBy" : "David Benioff, D.B.Weiss",  "airedDate" : "August2017"  },  /\* 3 \*/  {  "\_id" : 3,  "episodeName" : "The Queen’s Justice",  "directedBy" : "Mark Mylod",  "writtenBy" : "David Benioff, D.B.Weiss",  "airedDate" : "July2017"  },  /\* 4 \*/  {  "\_id" : 2,  "episodeName" : "Stormborn",  "directedBy" : "Mark Mylod",  "writtenBy" : "Bryan Cogman",  "airedDate" : "July2017"  },  /\* 5 \*/  {  "\_id" : 1,  "episodeName" : "Dragonstone",  "directedBy" : "Jeremy Podeswa",  "writtenBy" : "David Benioff, D.B.Weiss",  "airedDate" : "July2017"  } |
| db.collection.find().sort() | Esto es equivalente a un SELECT con un WHERE y luego ORDERED BY una columna  Query : db.SampleDataCollection.find({ directedBy : “Mark Mylod”}).sort( { \_id : -1 })  RDBMS Equivalent:  SELECT \* FROM SampleDataCollection  WHERE directedBY = “Mark Mylod”  ORDER BY \_id DESC  Result:  /\* 1 \*/  {  "\_id" : 3,  "episodeName" : "The Queen’s Justice",  "directedBy" : "Mark Mylod",  "writtenBy" : "David Benioff, D.B.Weiss",  "airedDate" : "July2017"  },  /\* 2 \*/  {  "\_id" : 2,  "episodeName" : "Stormborn",  "directedBy" : "Mark Mylod",  "writtenBy" : "Bryan Cogman",  "airedDate" : "July2017"  } |
| db.collection.find().limit( ) | Esto es equivalente aun SELECT con una condición TOP o LIMIT  Query : db.SampleDataCollection.find().limit(2)  RDBMS Equivalent :  SELECT \* FROM SampleDataCollection LIMIT 2  Result :  /\* 1 \*/  {  "\_id" : 1,  "episodeName" : "Dragonstone",  "directedBy" : "Jeremy Podeswa",  "writtenBy" : "David Benioff, D.B.Weiss",  "airedDate" : "July2017"  },  /\* 2 \*/  {  "\_id" : 2,  "episodeName" : "Stormborn",  "directedBy" : "Mark Mylod",  "writtenBy" : "Bryan Cogman",  "airedDate" : "July2017"  } |
| db.collection.find( ).limit( ) | Esto es equivalente a un SELECT con un LIMIT con un WHERE sobre un campo especifico  Query : db.SampleDataCollection.find( {airedDate : “July2017”} ).limit(2)  RDBMS Equivalent :  SELECT \* FROM SampleDataCollection  WHERE airedDate = ‘July2017’  LIMIT 2  Result :  /\* 1 \*/  {  "\_id" : 1,  "episodeName" : "Dragonstone",  "directedBy" : "Jeremy Podeswa",  "writtenBy" : "David Benioff, D.B.Weiss",  "airedDate" : "July2017"  },  /\* 2 \*/  {  "\_id" : 2,  "episodeName" : "Stormborn",  "directedBy" : "Mark Mylod",  "writtenBy" : "Bryan Cogman",  "airedDate" : "July2017"  } |
| db.collection.find().skip( ) | Esto es equivalente a un SELECT con un WHERE sobre la clave primaria evitando las primeras 4 filas del principio de la tabla  Query : db.SampleDataCollection.find().skip(4)  RDBMS Equivalent :  SELECT \* FROM SampleDataCollection WHERE \_id > 4  Result :  {  "\_id" : 5,  "episodeName" : "Eastwatch",  "directedBy" : "Matt Shakman",  "writtenBy" : "Bryan Cogman",  "airedDate" : "August2017"  } |
| db.collection.find( ).skip( ) | Esto es equivalente a un SELECT con un WHERE sobre una columna específca y evitar un número N de filas del conjunto resultante    Query : db.SampleDataCollection.find( { directedBy : “Matt Shakman” } ).skip(1)  RDBMS Equivalent :  SELECT \* FROM SampleDataCollection  WHERE directedBy = ‘Matt Shakman’  Result :  {  "\_id" : 5,  "episodeName" : "Eastwatch",  "directedBy" : "Matt Shakman",  "writtenBy" : "Bryan Cogman",  "airedDate" : "August2017"  } |
| db.collection.count() | Esto es equivalente a un SELECT COUNT(\*) de una tabla  Query:db.SampleDataCollection.count()  RDBMS Equivalent:  SELECT COUNT(\*) FROM SampleDataCollection  Result : 5 |
| db.collection.find( ).count() | Esto es equivalente a un SELECT COUNT(\*) de una tabla con un WHERE  Query : db.SampleDataCollection.find({ airedDate : “August2017” }).count()  RDBMS Equivalent:  SELECT COUNT(\*)  FROM SampleDataCollection  WHERE airedDate = ‘August2017’  Result : 2 |
| db.collection.findOne( ) | Esto es equivalente a buscar y retornar solo un registro del conjunto resultante  Query : db.SampleDataCollection.findOne({ airedDate : “August2017” })  RDBMS Equivalent :  SELECT TOP 1 \*  FROM SampleDataCollection  WHERE airedDate = ‘August2017’  Result :  {  "\_id" : 4,  "episodeName" : "The Spoils of War",  "directedBy" : "Matt Shakman",  "writtenBy" : "David Benioff, D.B.Weiss",  "airedDate" : "August2017"  } |