MONGO DB NOTES

mongoDB.

Basic Database Commands:

Command to show all databases:
show dbs

Create or switch databases:
use <databaseName>

View the name of the current database:
db

Delete database:
db.dropDatabase();

Collection Commands:

Q) What is a collection in MongoDB?
Ans) Collection is a collection of documents (A document is simply just a single data record or row). A collection is similar to a table in SQL.

Command to show all collections:
show collections

Command to create a collection:
db.createCollection("Products")

Command to delete a collection:
db.products.drop()

Command to show all rows in a collection:
db.products.find()

Command to show all rows in a collection (prettified):
db.products.find().pretty()

Command to insert a single value in a collection:
db.products.insertOne({_id: 1, name: "Pen", price: 1.20})

Command to show all rows in a collection:
db.products.insertMany([{_id: 2, name: "Pencil", price: 0.80}, {_id: 3, name: "Notebook", price: 5.25}])

Find Commands:

Command to search in a database:
db.products.find({name: 'Pen'})

Command to count the number of matching results:
db.products.find({name: 'Pen'}).count()

Command to limit the number of rows in output:
db.products.find().limit()

Commands using queries:
db.products.find({price: {$lt: 1.0}}) -> Less than

db.products.find({price: {$lte: 1.0}}) -> Less than equals

db.products.find({price: {$gt: 1.0}}) -> Greater than

db.products.find({price: {$gte: 1.0}}) -> Greater than equals

db.products.find({price: {$eq: 1.0}}) -> Equal to

Commands using projections:
db.products.find({price: {$lt: 1.0}}, {name: 1}) -> Displays only the name and id (By default)

db.products.find({price: {$lt: 1.0}}, {name: 1, _id: 0}) -> Displays only the name and prevents the default display of id as well.

Update and Delete Commands:

Command to update one row and set a value:
db.products.updateOne({_id: 1}, {$set: {stock: 32}})

Command to delete a record:
db.products.deleteOne({_id: 2})

Relationships in MongoDB:

We can create relationships in mongo by embedding documents in other documents as follows:

```
db.products.insert(
   {
      _id: 3,
      name: "Rubber",
      price: 1.30,
      stock: 43,
      reviews: [
         {
            authorName: "Sally",
            rating: 5,
            review: "Best rubber ever!"
         },
         {
            authorName: "John",
            rating: 5,
            review: "Awesome Rubber!"
         }
      ]
   }
)
```