Querying

task: write a query to find all the RAs under 40 years old. Hint: use $lt for less than.

db.hwuPeople.find({role: "ra", age: {$lt: 40}})

task: Sort the entire list of people by age; use ascending order.

db.hwuPeople.find().sort({age: 1})

Inserting, Updating & Removing

task: add yourself, specifying your \_id manually. Hint: treat “\_id” as just another name/value pair.

db.hwuPeople.insert({\_id : "somethingUnique", first\_name : "joe", last\_name : "blogs", age : 21, role : "msc", id : "jb33"})

task: update your information to provide your email address and your title (e.g., mr, ms etc.).

db.hwuPeople.update({\_id : "somethingUnique"}, {$set: {email: "jb33@hw.ac.uk", title: "mr"}})

task: Write an upsert query to properly insert Andy Proudlove into the database.

There are many possibilities, but something like this works:

db.hwuPeople.update({first\_name : "andy", last\_name : "proudlove"}, {first\_name : "andy", last\_name : "proudlove", role : "ra", age: 47}, {upsert: true})

task: Use the *remove* command to delete Joe Bloggs from the database.

db.hwuPeople.remove({id : "jb33"})

Final Exercise

Create a new collection (called “exercise”) that includes the following information:

db.exercise.insert({\_id : "ab1", name: "albert burger", role: "supervisor"})

db.exercise.insert({\_id : "ag1", name: "alasdair grey", role: "supervisor"})

db.exercise.insert({\_id : "iw1", name: "iain wiles", role: "phd", supervisor: ["ag1"]})

db.exercise.insert({\_id : "ss1", name: "steve smith", role: "phd", supervisor: ["ag1", "ab1"]})

db.exercise.insert({\_id : "hg1", name: "hugh dollar", role: "phd", supervisor: ["ab1"]})

Now construct an optimized query to list all of Dr Grey’s students.

db.exercise.ensureIndex({supervisor: 1})

db.exercise.find({supervisor: "ag1"})

Write a second query to find all the students with 2 supervisors.

Hint: use the condition *$size: 2*.

db.exercise.find({supervisor : {$size : 2}})