

```

// Lab Solution
const
  mongoose = require('mongoose');

// Models used in this solution
var Person, Currency;

// First lets open connection to the DB
mongoose.connect('mongodb://127.0.0.1/lab_db', {
  useNewUrlParser: true
});

/**
lab 01 - Define the relevant schema

Field Name | Type
-----|-----
id         | Number
first_name | String
last_name  | String
age        | Number
email      | String
credit_card| String
gender     | String
ip_address | String
currency   | String
*/

function lab01() {

  Person = mongoose.model('Person', {
    "id": Number,
    "first_name": String,
    "last_name": String,
    "age": Number,
    "email": String,
    "credit_card": String,
    "gender": String,
    "ip_address": String,
    "currency": String,
  });
}

/**
lab 02 - Add multiple records
*/
//Person.insertMany([...])

/** lab 03 - Find records */

function lab03A() {
  // Use `
find ` to count the number of records in the table
  Person.countDocuments({},
    function (err, count) {
      console.log("Number of persons:", count);
    });
}

function lab03B() {
  // Use `
find ` to find specific record ["age":79]
  Person.find({
    age: 54
  },
    function (err, records) {
      console.log("Number of persons with age of 54:", records.length);
    });
}

/** lab 04 - Aggregations */
// Group by credit card type (how many records per each card type)
function lab04A() {

  Person.aggregate([
    $group: {
      _id: "$credit_card",
      total: {
        "$sum": 1
      }
    }
  ])
}

```

```

    }
  },
  function (err, reply) {
    console.log("The average age is:", reply);
  });
}

function lab04B() {

  // Sort the results from the previous lab
  Person.aggregate([
    {
      $group: {
        _id: "$credit_card",
        total: {
          "$sum": 1
        }
      }
    },
    {
      $sort: {
        total: -1 // can be 1 or -1
      }
    }
  ],
  function (err, reply) {
    console.log("Sorted credit_cards by total:", reply);
  });
}

// lab 05 - Populate
/*
Create new table for the currency rates [named: currencies]
Field Name | Type
-----|-----
currency   | String
rate       | Number
**/
function lab05A() {

  Currency = mongoose.model('currencies', {
    "currency": String,
    "rate": Number
  }).insertMany([
    {
      "currency": "COP",
      "rate": 1.1
    },
    {
      "currency": "USD",
      "rate": 3.6
    },
    {
      "currency": "PEN",
      "rate": 6.2
    },
    {
      "currency": "SEK",
      "rate": 1.34
    }
  ]);
}

function lab05B() {
  Person.aggregate([
    {
      $match: {}
    },
    {
      $lookup: {
        from: "currencies",
        localField: "currency",
        foreignField: "currency",
        as: "rate"
      }
    },
    {
      $limit: 2
    },
  ],
  function (err, reply) {
    console.log("Sorted credit_cards by total:", reply);
    process.exit();
  });
}

```