Cardiff School of Computer Science and Informatics

Coursework Assessment for CM3104 Large Scale Databases

Module Code: CM3104
Module Title: Large Scale Databases
Lecturer: C.B. Jones, A.I. Abdelmoty, J. Shao
Assessment Title: Coursework 1
Assessment Number: 1
Date Set: Thursday 1st November 2018
Submission Date and Time: Wednesday 5th December 2018 at 9:30am.
Return Date: Week 12, Friday 11th January 2019.

This assignment is worth 30 % of the total marks available for this module. The penalty for late or non-submission is an award of zero marks.

Your submission must include the official Coursework Submission Cover sheet, which can be found here:

https://docs.cs.cf.ac.uk/downloads/coursework/Coversheet.pdf

Submission Instructions

Description		Type	Name
Cover sheet	**Compulsory**	One PDF (.pdf) file	[student number].pdf
PART A	**Compulsory**	One PDF (.pdf) file comprising your answer to all questions with snapshots of the mongodb shell as explained below.	PartA_[student number].pdf
	Compulsory	The javascript file with your answer to all the questions in Task 2.	PartA_[student number].js
PART B	**Compulsory**	One PDF (.pdf) file that contains answers to all questions.	PartB_[student number].pdf
PART C	**Compulsory**	One PDF (.pdf) file that includes **for each of the five answers** to Part C: 1 - The Oracle SQL query; 2 – The answer to the query; 3 - A **screen shot** of: the query in Oracle followed by the Oracle output from the query.	PartC_[student number].pdf

Any deviation from the submission instructions above (including the number and types of files submitted) may result in a mark of zero for the assessment or question part.

Coursework PART A: NoSQL Databases [worth 10 marks]

In this part of the coursework you will make use of two data sets: a restaurants data set (in the file, restaurants.js) and a zipcodes data set (in the file zipcodes.js). You are building an application with MongoDB that will use both data sets to find information about restaurants in different cities.

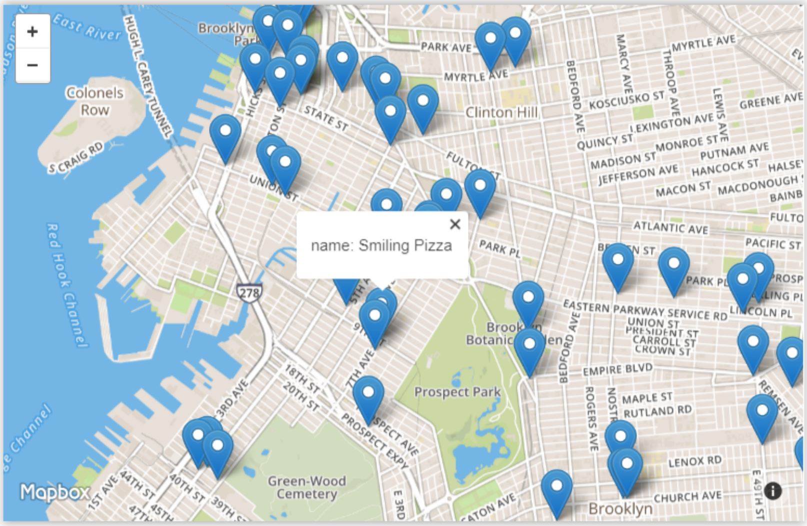
An example record from the restaurants dataset is:

```
{
    "_id" : ObjectId("55cba2476c522cafdb053ae8"),
    "location" : {
        "coordinates" : [
            -73.9973325,
            40.61174889999999
        ],
        "type" : "Point"
    },
    "name" : "C & C Catering Service"
}
```

An example record from the zipcodes dataset is:

```
{
    "_id" : "01002",
    "city" : "CUSHMAN",
    "loc" : [
        -72.51565,
        42.377017
    ],
    "pop" : 36963,
    "state" : "MA"
}
```

Here is a map of a sample from the restaurants dataset.



Task 1

Import both datasets into MongoDB in ONE database. Each dataset should be stored in a separate collection. For example, you could build a “services” database with two collections: “restaurants” and “zipcodes”.

(1 mark)

Task 2

1. Find all the Pizza restaurants in the dataset and count their number.
   A Pizza restaurant is any restaurant which has the word “Pizza” in its title.
   (1 mark)

2. For each Pizza restaurant from the above set, find the city that it is located in.
   You can assume that a restaurant is located in the nearest city to its location.
   (2 marks)

3. In your application you will need to retrieve the restaurants and their associated cities frequently.
   Create a new collection (“geodb”) that contains all the data in the restaurant collection, but for every restaurant document in the new collection, add an additional field “city” whose value is the closest city to the restaurant’s location. Use the same method of finding the city as you have done in question 2.

   Documents in your new “geodb” collection should be similar to the following:

```
{
    "_id" : ObjectId("55cba2476c522cafdb053ae8"),
    "location" : {
        "coordinates" : [
            -73.9973325,
            40.61174889999999
        ],
        "type" : "Point"
    },
    "name" : "C & C Catering Service",
    "city" : "BROOKLYN"
}
```

   (3 marks)

4. Query the new “geodb” collection to find restaurants, grouped by city. Your answer should include: city name, number of restaurants in the city and a list of all the names of the restaurants in the city.

   An example document of the results of this query is as follows:

```
{
    "City": " MOBILE",
    "No of Restaurants": 3,
    "Restaurants": [
        "Island Soft Pretzel Stop",
        "Dairy Queen Grill & Chill",
        "Statue Of Liberty Deli"
      ]
}
```

(2 marks)

5. A different method of storing the restaurant and city information together is by modifying the original “zipcodes” collection by adding a field “restaurantsInCity” whose value is an array of all the restaurants that are located in the city. An example of a document in the modified “zipcodes” collection is as follows:

```
{
    "_id" : "36607",
    "city" : "MOBILE",
    "loc" : [
        -88.1029,
        30.697486
    ],
    "pop" : 8610,
    "state" : "AL"
    “restaurantsInCity” : [“Island Soft Pretzel Stop”, “Dairy Queen Grill & Chill”, "Statue
Of Liberty Deli"]
}
```

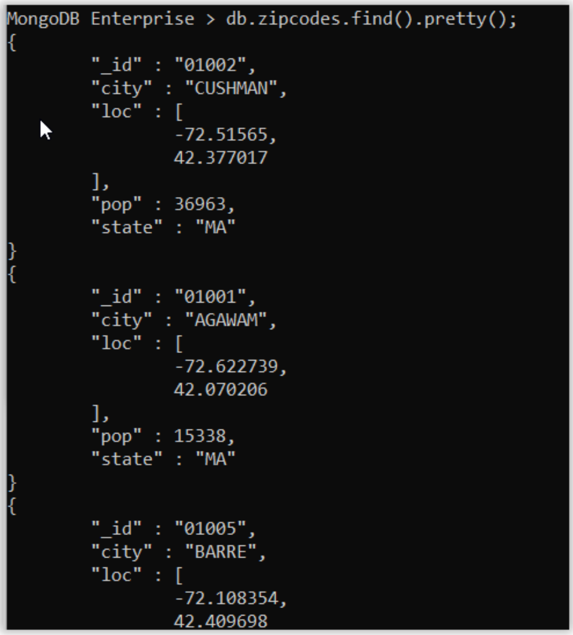
   Explain whether this is a sensible modelling option in MongoDB, by referring to its effectiveness for <u>storing</u> and <u>retrieving</u> the information about restaurants and associated cities.

(1 mark)

UPLOADS FOR PART A

1. Save your answer to the questions in Task 2 in a javascript file with the name: PartA_[student number].js

2. Save your answer to all questions in Task 1 and Task 2 in a pdf file: PartA_[student number].pdf. You can take a snapshot of the question being executed in the MongoDB shell, clearly demonstrating the answer to the question (include snapshots of all intermediate steps as appropriate). A sample of the results is sufficient – this should be the first 5 documents. An example of a sufficient snapshot of the answer to a query to find all records in the “zipcodes” collection is as follows.



```
MongoDB Enterprise > db.zipcodes.find().pretty();
{
        "_id" : "01002",
        "city" : "CUSHMAN",
        "loc" : [
                -72.51565,
                42.377017
        ],
        "pop" : 36963,
        "state" : "MA"
}
{
        "_id" : "01001",
        "city" : "AGAWAM",
        "loc" : [
                -72.622739,
                42.070206
        ],
        "pop" : 15338,
        "state" : "MA"
}
{
        "_id" : "01005",
        "city" : "BARRE",
        "loc" : [
                -72.108354,
                42.409698
```

PART B: Data Mining [worth 10 marks]

We have the following table of instances (cases) recorded for contact lens prescriptions. The same dataset will be made available on Learning Central in the Assessment document and its attributes and values are self-explanatory.

D	age	spectacle-prescrip	astigma tism	tear-prod-rate	contact-lenses
1	young	myope	no	reduced	none
2	young	myope	no	normal	soft
3	young	myope	yes	reduced	none
4	young	myope	yes	normal	hard
5	young	hypermetrope	no	reduced	none
6	young	hypermetrope	no	normal	soft
7	young	hypermetrope	yes	reduced	none
8	young	hypermetrope	yes	normal	hard
9	pre-presbyopic	myope	no	reduced	none
10	pre-presbyopic	myope	no	normal	soft
11	pre-presbyopic	myope	yes	reduced	none
12	pre-presbyopic	myope	yes	normal	hard
13	pre-presbyopic	hypermetrope	no	reduced	none
14	pre-presbyopic	hypermetrope	no	normal	soft
15	pre-presbyopic	hypermetrope	yes	reduced	none
16	pre-presbyopic	hypermetrope	yes	normal	none
17	presbyopic	myope	no	reduced	none
18	presbyopic	myope	no	normal	none
19	presbyopic	myope	yes	reduced	none
20	presbyopic	myope	yes	normal	hard
21	presbyopic	hypermetrope	no	reduced	none
22	presbyopic	hypermetrope	no	normal	soft
23	presbyopic	hypermetrope	yes	reduced	none
24	presbyopic	hypermetrope	yes	normal	none

You are required to do the following in this part of the coursework:

1. Derive a decision tree for the class attribute *Contact-lenses* using the ID3 method available within the Weka package, using Weka’s default setting. Note that you will need to use package manager to install ID3 first.

(2 Marks)

2. In the default setting, there is a setting “Cross-Validation Folds 10” in the test options. Briefly explain what *Cross Validation* means in this context and why we use cross validation in testing.

(4 Marks)

3. Now perform the following tests: you vary “fold” from 2 to 10, run ID3 and observe classification accuracy for each setting. You then change the test options setting to “Use training set” and run ID3 and observe classification accuracy. You can present these test results as a table or a bar chart. Comment on your test results: which method (cross validation or using training set) is better for testing your derived tree and why?

(4 Marks)

Uploads for Part B

One PDF (.pdf) file that contain answers to all questions. The file name should be formatted as PartB_[student number].pdf

Coursework PART C: Spatial Databases [worth 10 marks]

Please note that instructions on accessing the data that are required for this section follow the questions below.

For each of the following questions: write the Oracle SQL query that answers the question and provide the answer. Follow that with a screen shot that shows **both** the SQL query and the output of the query in your SQL interface (SQL Developer is recommended).

1. How many districts does the M4 motorway pass through?
   Note: the motorway is represented by a sequence of short segments, so be careful that each district is only counted once.
   (2 marks)

2. What is the total number railway stations in the districts that have a coastal boundary?
   Note: the coastline is represented by a sequence of segments, so be careful that each railway station is only counted once.
   The railway stations are in stationssouthwales.
   (2 marks)

3. Name the populated places are within 80 metres of the M4 Motorway.
   Note: The motorway is represented by a sequence of short segments, so ensure that each populated place is only counted once;
   Use the cardiffpopulatedplaces table to obtain the populated places and check that the attribute "classifica" has the value 'Populated Place'. The attribute for the name of the motorway is "number_mb".
   (2 marks)

4. Which railway station is furthest from the coast and what is the distance?
   The output of your SQL code should be only those two items of information.
   Note: the SQL code that you provide should return just the name of the single settlement that is furthest from the coast.
   The railway stations are in the file called stationssouthwales.
   (2 marks)

5. Which district has the longest length of the M4 motorway running through it and what is its length?
   The output of your SQL code should be only those two items of information.
   Note: the motorway is represented by a sequence of short segments and all such segments must be considered when answering this question (i.e. the measurement of length must be based on all segments in the respective district).
   (2 marks)

Instructions on accessing data required for Part 3 (Spatial Databases)

The file on Learning Central called LSD-SpatialData2018Coursework.zip contains some Ordnance Survey digital map data in shape file format relating to the area around Cardiff in South Wales. You must import them to tables in Oracle using the Mapbuilder application. [Note also that the same shapefile data can be used to add layers to a project in a GIS such as ArcGIS or QuantumGIS (QGIS).]

On Learning Central you will also find the program mapbuilder as a jar file, i.e. map_builder-12.2.1.3.0.jar. The purpose of this program is to access the shape files and transform them into tables in the Oracle database.

To run the mapbuilder program, open a terminal and set the directory to the location where you have saved map_builder-12.2.1.3.0.jar and type:

```
java –jar map_builder-12.2.1.3.0.jar
```

The application should then open.
Click on the add connection icon (below the file menu item)
In the connection dialogue box:
- Create a name of your choice for the connection
- User and password are your Oracle username (probably same as normal user name) and password which is distinct for Oracle.

Set Host to : csoracle.cs.cf.ac.uk
Set Service Name to : csora12edu.cs.cf.ac.uk

In the Mapbuilder >tools menu, select >Import Shapefile
In the dialogue box, click on the shapefile button and navigate to and select a shape file (i.e. with extension .shp) in the directory LSD-SpatialData2018Coursework
There are six .shp files in the directory:
CardiffPopulatedPlaces
coastlineSouthWales
CardiffDistricts
stationsSouthWales
MotorwaysSouthWales
RailwaysSouthWales [This last file is not required for the SQL queries but you might find it of use if you plot the data in a package such as QGIS]

The geometry table field tells you the name of the table that is created (it is the name of the shape file). Click next.

In the next dialogue, click on the SRID button to tell Oracle which spatial reference system (SRS) the data uses. Scroll down to and select “British National Grid”. For future reference note that in the dialogue that prompts for SRID you can just type in the numeric code for the SRS, which in this case is 81989. The Create/update spatial index checkbox needs to be checked (which it is by default). Click next for the following dialogue on Theme name. It is not necessary to enter anything, just click Finish. The program should then go ahead and create the table. Repeat this process for all six shape files, resulting in six tables in Oracle. To see the fields of the created tables, go to SQL Developer and click on the table name in the Tables sub-menu for your connection.

In each table there is a column called “geometry” that contains the spatial data, which in the case of CardiffDistricts is a set of polygons; for stationsSouthWales and CardiffPopulatedPlaces it is points, and for MotorwaysSouthWales and coastlineSouthWales it consists of lines.

Uploads for Part C

One PDF (.pdf) file the name of which is in the form: PartC_[student number].pdf
that includes for each of the five answers to Part C:
1 - The SQL query (using Oracle syntax) – typed out by you.
2 – The output to the query (copied from the Oracle output) – typed out by you;
3 - A screen shot of:
the query in Oracle followed by the
output from the query.
The screen output should match precisely the content of Items 1 and 2

Learning Outcomes Assessed

1. Demonstrate an appreciation of applications of large-scale databases in a variety of commercial, scientific and professional contexts;
2. Understand how relational databases are extended with object-relational technologies to support management of spatial information;
3. Describe non-relational database approaches to support access to very large databases;
5. Exhibit a sound understanding of data mining and show familiarity with data mining algorithms;

Criteria for assessment

Credit will be awarded against the following criteria.

Part A

- Ability to create and populate a MongoDB database
- Ability to query and analyse data stored in the database
- Quality of reflection on modelling options using the document data model

Feedback on your performance will address each of these criteria.

Allocated marks are specified against each specific questions, to a total of **[10 marks]**

Part B

- Correct output from Weka for Q1 (you should cut and paste the output from Weka into your coursework). **[2 marks]**
- Clarity (not the length) in explaining the concept of cross validation, how it works, and why we use it for Q2. [**4 marks**]
- Correct output from Weka for Q3 (i.e. presenting the correct test results), and convincing argument on which method is better for testing the derived tree. **[4 marks]**

Part C

- The two marks allocated for each question are awarded according to: formulation of Oracle spatial SQL queries with appropriate use of spatial operators and functions, along with correct corresponding output from the queries. **[10 marks]**

Feedback and suggestion for future learning

Feedback on your coursework will address the above criteria. Feedback and marks will be returned by email at your @cardiff.ac.uk address in Week 12 on Friday 11th January 2019. It will also be possible to request face to face feedback with the lecturers by appointment.

Feedback from this assignment will be useful for the final exam and potentially for your future learning more broadly.