



# MODULAR PROGRAMME

## COURSEWORK ASSESSMENT

### SPECIFICATION

#### Module Details

|  |                           |   |
|--|---------------------------|---|
| <b>Module Code</b><br>UFCFR5-15-3  | <b>Run</b><br>21SEP/1     | <b>Module Title</b><br>Data Management Fundamentals     |
| <b>Module Leader</b><br>Prakash Chatterjee                                     | <b>Module Coordinator</b> | <b>Module Tutors</b><br>P Chatterjee                    |
| <b>Component and Element Number</b><br>B: CW1                                  |                           | <b>Weighting: (% of the Module's assessment)</b><br>50% |
| <b>Element Description</b><br>Model, cleanse, normalize, map, & query big data |                           | <b>Total Assignment time</b><br>36 hours                |

#### Dates

|  |  |
|--|--|
| <b>Date Issued to Students</b><br>01 Nov 2021    | <b>Date to be Returned to Students</b><br>4 working weeks after hand-in. |
| <b>Submission Place</b><br><br><b>Blackboard</b> | <b>Submission Date</b><br>20 Jan 2022                                    |
|  | <b>Submission Time</b><br><b>2.00 pm</b>                                 |

## Deliverables

A ZIP file submitted to BB called *dmf.zip* containing all attached code & reports (in MARKDOWN format).

## Module Leader Signature




## UFCFLR-15-M Data Management Fundamentals

### Assignment Specification 2021-22

#### Learning Goals & Outcomes

- Learn to model, cleanse, normalize, shard, map, query and analyze substantial real-world big data (230mb+);
- Understand the data cleansing, normalization and sharding processes by writing PYTHON scripts to process and convert the data to first (cleansed) CSV and then (normalized) SQL;
- Design and implement a relational (MySQL) database and then write a PYTHON script to pipe (import) the SQL into the appropriate tables ensuring all referential integrity constraints are met.
- Construct and implement a set of SQL queries to extract data using various filters and constraints.
- Map (forward engineer) the data to a NoSQL database of your choice (MongoDB, BaseX, CouchBase, ArangoDB etc.)

- Write a short, reflective report on the learning outcomes you have achieved.
- Get exposure to and learn the use of a range of data oriented technologies (databases, python & sql.)
- Learn and use the [MARKDOWN](#)  markup syntax.

### Context: Measuring Air Quality

Levels of various air borne pollutants such as Nitrogen Monoxide (NO), Nitrogen Dioxide (NO<sub>2</sub>) and particulate matter (also called particle pollution) are all major contributors to the measure of overall air quality.

For instance, NO<sub>2</sub> is measured using micrograms in each cubic metre of air ( $\mu\text{g m}^3$ ). A microgram ( $\mu\text{g}$ ) is one millionth of a gram. A concentration of  $1 \mu\text{g m}^3$  means that one cubic metre of air contains one microgram of pollutant.

To protect our health, the UK Government sets two air quality objectives for NO<sub>2</sub> in their [Air Quality Strategy](#)

1. The hourly objective, which is the concentration of NO<sub>2</sub> in the air, averaged over a period of one hour.
2. The annual objective, which is the concentration of NO<sub>2</sub> in the air, averaged over a period of a year.

The following table shows the colour encoding and the levels for Objective 1 above, the mean hourly ratio, adopted in the UK.

| Index             | 1    | 2      | 3       | 4        | 5        | 6        | 7       | 8       | 9       | 10          |
|-------------------|------|--------|---------|----------|----------|----------|---------|---------|---------|-------------|
| Band              | Low  | Low    | Low     | Moderate | Moderate | Moderate | High    | High    | High    | Very High   |
| $\mu\text{g/m}^3$ | 0-67 | 68-134 | 135-200 | 201-267  | 268-334  | 335-400  | 401-467 | 468-534 | 535-600 | 601 or more |

Further details of colour encodings and health warnings can be found at the [DEFRA Site](#).

## The Input Data

The following ZIP file provides data ranging from 2004 to 03 February 2021 taken from 18 monitoring stations in and around Bristol.

Monitors come and go and may suffer down times, so the data isn't complete for all stations at all times.

Download & save the data file: [bristol-air-quality-data.zip](#)

Shown here is the first 8 lines of the file (cropped):

```
1 Date Time;NOx;NO2;NO;SiteID;PM10;NVPM10;VPM10;NVPM2.5;PM2.5;VPM2.5;CO;O3;SO2;Temperature;RH;Air Pressure;I
2 2013-08-23T07:00:00+00:00;51.54044;30.50055;13.72186;452;27.8;23.2;4.6;16.4;19.454;2.9;;20.40603;;;AURN
3 2013-08-23T08:00:00+00:00;94.5;44.25;33.0;203;;;;;;;;;;Brislington Depot;51.4417471802,-2.55995583224;2
4 2013-08-23T10:00:00+00:00;242.75;59.75;119.5;206;;;;;;;;;;Rupert Street;51.4554331987,-2.59626237324;2
5 2013-08-23T14:00:00+00:00;197.75;73.25;81.25;270;;;;;;;;;;Wells Road;51.4278638883,-2.56374153315;2003-
6 2013-08-23T18:00:00+00:00;81.0;55.5;16.5;203;;;;;;;;;;Brislington Depot;51.4417471802,-2.55995583224;2
7 2013-08-23T18:00:00+00:00;46.65544;42.06448;2.99414;452;31.1;26.5;4.6;19.7;24.595;4.7;;42.30884;;;AURN
8 2013-08-23T19:00:00+00:00;95.75;65.5;19.75;215;;;;;;;;;;Parson Street School;51.432675707,-2.604956656;
```

Note the following:

### There are 18 stations (monitors):

188 => 'AURN Bristol Centre',  
 203 => 'Brislington Depot',  
 206 => 'Rupert Street',  
 209 => 'IKEA M32',  
 213 => 'Old Market',  
 215 => 'Parson Street School',  
 228 => 'Temple Meads Station',  
 270 => 'Wells Road',  
 271 => 'Trailer Portway P&R',  
 375 => 'Newfoundland Road Police Station',  
 395 => 'Shiner's Garage',  
 452 => 'AURN St Pauls',  
 447 => 'Bath Road',  
 459 => 'Cheltenham Road \ Station Road',  
 463 => 'Fishponds Road',  
 481 => 'CREATE Centre Roof',  
 500 => 'Temple Way',  
 501 => 'Colston Avenue'

Each line represents one reading from a specific detector. Detectors take one reading every hour. If you examine the file using a programming editor, Notepad++ can handle the job, you can see that the first row gives headers and there are another 1408379 (1.4 million+) rows (lines). There are 23 data items (columns) per line.

The schema is given below:

| <b>measure</b> | <b>desc</b>  | <b>unit</b> |
|----------------|--|-------------|
| Date Time      | Date and time of measurement   | datetime    |
| NOx            | Concentration of oxides of nitrogen                                    | µg/m3       |
| NO2            | Concentration of nitrogen dioxide                                      | µg/m3       |
| NO             | Concentration of nitric oxide  | µg/m3       |
| SiteID         | Site ID for the station  | integer     |
| PM10           | Concentration of particulate matter <10 micron diameter                | µg/m3       |
| NVPM10         | Concentration of non - volatile particulate matter <10 micron diameter | µg/m3       |
| VPM10          | Concentration of volatile particulate matter <10 micron diameter       | µg/m3       |
| NVPM2.5        | Concentration of non volatile particulate matter <2.5 micron diameter  | µg/m3       |
| PM2.5          | Concentration of particulate matter <2.5 micron diameter               | µg/m3       |
| VPM2.5         | Concentration of volatile particulate matter <2.5 micron diameter      | µg/m3       |
| CO             | Concentration of carbon monoxide                                       | mg/m3       |
| O3             | Concentration of ozone   | µg/m3       |
| SO2            | Concentration of sulphur dioxide                                       | µg/m3       |
| Temperature    | Air temperature  | °C          |
| RH             | Relative Humidity  | %           |
| Air Pressure   | Air Pressure   | mbar        |

| measure         | desc                               | unit      |
|-----------------|------------------------------------|-----------|
| Location        | Text description of location       | text      |
| geo_point_2d    | Latitude and longitude             | geo point |
| DateStart       | The date monitoring started        | datetime  |
| DateEnd         | The date monitoring ended          | datetime  |
| Current         | Is the monitor currently operating | text      |
| Instrument Type | Classification of the instrument   | text      |

### Task 1: Crop, Cleanse and Refactor the Data (16 marks)

Design & write appropriate PYTHON scripts to carry out the following.

- Crop the file to delete any records before 00:00 1 Jan 2010 (1262304000).
- Filter for and remove any dud records where there is no value for SiteID or there is a mismatch between SiteID and Location. (This script should print the lines number and mismatch field values for each dud record.)

**Submission files:** Two Python scripts: *crop.py* & *clean.py* that generate cropped & cleaned CSV files.

### Task 2: Create and Implement a Normalized Database. (12 marks)

- Use MySQL Workbench or any other tool to create a ER model in the third-normal form to hold the given data.

- b. Use the forward engineer feature of MySQL Workbench to generate the SQL schema and implement the database (pollution-db).  
(If this does not work for you, e.g. MYSQL Worbench configuration issues, you can use PHPMyAdmin within XAMPP to create the tables by hand. You can then use the export feature to extract the SQL.)

**Submission files:** A ER diagram (*pollution\_er.gif* or *pollution\_er.png*) and a SQL file (*pollution.sql*) holding table definitions.

---

### Task 3: Write Python scripts to populate the database & generate SQL. (20 marks)

- a. Design and write a PYTHON script (populate.py) that takes the cleaned CSV file as input and creates a new database instance (pollution-db2) and populates it.
- b. Create a PYTHON script (insert-100.py) that generates a SQL file (insert-100.sql) that holds the first 100 inserts to the readings table.

**Submission files:** Two Python scripts - *populate.py* & *insert-100.py*.

---

### Task 4: Design, Write and Run SQL Queries. (12 marks)

Write and implement (test run) the following four SQL queries:

- a. Return the date/time, station name and the highest recorded value of nitrogen oxide (NOx) found in the dataset for the year 2019.
  - b. Return the mean values of PM2.5 (particulate matter <2.5 micron diameter) & VPM2.5 (volatile particulate matter <2.5 micron diameter) by each station for the year 2019 for readings taken on or near 08:00 hours (peak traffic intensity).
  - c. Extend the previous query to show these values for all stations in the years 2010 to 2019.
-



**Submission file:** Code listing of the three SQL queries (*query-a.sql*, *query-b.sql*, *query-c.sql*)

---

### Task 5: Model, implement and query a selected NoSQL database. (30 marks)

Model the data for a specific monitor (station) to a NoSQL data model (key-value, xml or graph) to implement the selected database type/product & pipe or import the data.

You can select from any of the seven databases listed below but if you want, you can select one not currently on the list (after confirmation from the tutor).



**Submission file:** A short report in Markdown format (<1200 words) called *nosql.md* describing the data models used & relevant implementation details.

---

### Task 6: Reflective Report. (10 marks)

A short report in Markdown format (<800 words) reflecting on the assignment, the problems encountered and the solutions found.

In addition you should discuss and outline some of the Python tools and libraries that could be used to visualize this data. What maps / charts with which content?

You should also briefly outline the Learning Outcomes you have managed to achieve in undertaking this Assignment.

**Submission file:** A report in Markdown format called *report.md*.

---

## Assessment Criteria and Marks Allocation

### Task 1: Crop, Cleanse and Refactor the Data (16%)

- all scripts are well designed, structured and commented;
- scripts make use of dataframes, chunking and other techniques as appropriate;
- cropped and cleansed data is correctly formatted and complete.

### Task 2: Create and Implement a Normalized Database. (12%)

- a normalised ER diagram showing all entities, keys, attributes & relationships;
- a implemented database structure with all required tables, fields and keys;

### Task 3: Write a Python script to generate the required SQL. (20%)

- the scripts are well designed, structured and commented;
- the script makes use of objects and/or functions as required;
- the script generates valid SQL matching the database schema.
- the database is populated with the base data

### Task 4: Design, Write and Run SQL Queries. (12%)

- queries are valid and return the required results;

### Task 5: Model, implement and query a selected NoSQL database. (30%)

- database is one chosen from the list provided (unless explicitly agreed with the tutor);
- an adequate data model is developed and realized (implemented);

- all data is imported into the selected database type/product;
- evidence of example query implementation and result output;

#### **Task 6: Reflective Report. (10%)**

- a clear and concise report describing the problems, solutions and possible visualizations;
  - some reflection on the Learning Outcomes achieved.
- 

#### **Tutor support**

This coursework is seen as providing a learning experience in the tools & technologies used on this module. Support will be provided in workshops and via email.

Tutor help can be requested for any aspect of the coursework such as the overall design, Python coding problems or data structuring. Please ask for assistance after a bit of an effort with the problem rather than get stuck.

---

#### **Assessment Offences**

This assignment should be your own work. Allowing others to do the work for you, or sharing significant portions of code with others will be considered an assessment offence and may lead to your mark being reduced to 0. Part of the marking process will include similarity checks and we may ask you to explain your code in detail to verify that it is your own. Please refer to the [assessment offences policy document](#) for more information.

---

#### **References**

[Air Pollution - Wikipedia](#)

[UK Government Air Quality Strategy](#)

[Markdown Tutorial](#)

---

url: <http://fetstudy.uwe.ac.uk/~p-chatterjee/2021-22/modules/dmf/assignment/>

