Instructions for this assignment.

This assignment requires you to take a dataset of your choice, to **analyse** and **model** it, and then to **implement** it as a database with a simple associated **web application**. The assignment consists of four stages:

- Find a dataset and critique it, identifying some questions you would like to ask it;
- Build an E/R model for the data structure and adjust it for relational implementation;
- Build a MySQL database for the structure and put some or all of the dataset into it as instance data (there must be enough data that all tables and fields are used multiple times);
- Construct a simple web application in node to illustrate the data and implement asking some of the questions you identified in the first step. You will need to install packages as you did in your previous work – no package files have been provided for you.

You will submit a **report in .pdf format**, the **URL for your data source(s)**, and the **link to your lab environment** with the MySQL database and the web app in it (There should be a big button for a shareable link on page that launches the lab).

Please make sure that all original code (that you've written yourself and without assistance) is clearly and precisely labelled, or you will not get full credit for your work.

There is no page limit for your report, but we recommend that you aim for **less than 20 pages** (excluding any code, data and screenshots). In practice, reports far shorter than that often get top marks.

Don't forget to include the link to your lab and data, and your pdf – we will not be able to mark work that we can't access. We may also use the data source link you provide to compare with previous and current submissions based on the same data.

More information on each of the four stages is given in their sections below.

Stage 1. Find and critique a dataset

1. **Choose a source of open data**. This can be one that you've used earlier in this module, or it can be a new source. You may also combine two simpler datasets, provided there is benefit in linking them. The data must be **open** and **real** (not generated artificially for learning or research), and a normalised, relational model for the data can't have been published already.
2. **Assess the dataset in your report**. You should use the criteria used in discussion 1.104 (**Quality**, **level of detail**, **documentation**, **interrelation**, **use**, **discoverability**) and assess the **terms of use** (as in discussion 1.206)
3. **Explain your interest in your report.** Why is this an interesting dataset? Give some questions you would like to ask of the dataset that a database application could help with.

Stage 2. Model your data

When you draw your E/R diagram, follow the rules from **this module** and its set reading. Lots of other things get called E/R diagrams (you may have met other ones for other modules). Only diagrams that match the ones in this module will get full credit.

1. **Draw a complete E/R model of the data.** If you are only implementing a subset of the data structure, justify your decision in the report. All diagrams must be in the report pdf.
2. **Add cardinality to your E/R diagram.** If there are any structures that are not compatible with the Relational model, draw a second diagram showing the modified structure.
3. **List database tables and fields.** Evaluate the tables against the normal forms and adjust to ensure that your database will be **at least in 3NF**. Include an evaluation of which normal forms your database is in. If it isn't in higher forms, such as BCNF or 4NF, justify your decision not to normalise further.

Stage 3. Create the database

1. **Build the database structure in MySQL** in the lab environment. Record all *CREATE* commands used in your report.
2. **Enter instance data.** This can be a usable sample of the dataset or all the dataset. Detail how you added the data in your report.
3. **Reflect on how well the database reflects the data.** In your report note one or two points of elements that do or don't work well.
4. **List SQL commands** that answer questions identified in Stage 1/Step 3. If any can't be answered, explain why.

Stage 4. Create a simple web application

1. **Write a node.js module to present a web application that queries the database.** This can be quite simple, but should address some of the motivation and questions identified in Stage 1/Step 3. The user account used to connect should have appropriate privileges.
2. Take **screenshots** of the main screens from your web application and include them in your report.

Markers may choose to give up to 15% for students who go beyond the basic requirements, following their own ideas or making an extra effort. This can also be used to give credit to aspects of your work that are not recognised in the marking scheme (such as data cleaning or dataset alignment). You will be asked to highlight aspects of your work that are original or exceptional to help ensure that these are rewarded.

Referencing and good academic practice

All sources of information, data and code should be labelled. Your code should make explicit everything that is created solely by you and explain the origins of everything else.

You will be required to give the URL of your data source when you submit. If you are using a source used by current or former students, the submissions may be checked for evidence of collusion or plagiarism.

Summary of review criteria.

Each stage has equal weight in the marking scheme. The evaluation criteria for each stage are given below.

Stage 1. Find and critique a dataset (20%)

1. **Dataset is appropriate** (5%)– the data should be open, real (not artificial) and not too simple. It should fit the theme.
2. **Dataset assessment** (5%)– all criteria should be addressed
3. **Interest in the dataset** (5%)– should be clear and well justified
4. **Research questions** (5%) – questions are identified and justify a database approach (rather than either just sorting a spreadsheet or, on the other hand, using a statistical or machine learning approach)

Stage 2. Model your data (20%)

1. **E/R model** (5%) – identifies all fields and entities to be modelled
2. **E/R diagram** (5%) – diagram is clear, legal and uses ellipse/rhombus/rectangle notation
3. **E/R to relational mapping** (5%) – modelling is clear and sensible; issues with the relational model are resolved
4. **Relational model normalisation** (5%) – analysis is clear, explicit and accurate, resulting in an appropriate normalisation

Stage 3. Create the database (20%)

1. **Accurately implement the model** (5%) – the MySQL database reflects the models
2. **Implementation is sensible** (5%) – including column types, primary and foreign keys, and other constraints
3. **Critical reflection** (5%) – issues with data and implementation are noted, along with how well these relate to interest and research questions
4. **Queries** – queries are correct and accurately reflect the identified questions

Stage 4. Create a simple web application (20%)

1. **Application runs** (5%)
2. **Database interaction works** (5%) – Connections and queries are appropriately handled
3. **Application works** (5%) – Data is presented appropriately; valid and accessible HTML
4. **Goals are satisfied** (5%) – Motivation and (some) queries are addressed

Clear referencing (5%)

1. **Referencing includes data, literature and code labelling** (5%)– All external sources of data, code, techniques and research are clearly included in the report (5 marks)

Discretionary extra credit (15%)