



University of Exeter
Medical School

Assignment Brief - Coursework

Module Title:	Computational Skills for Health and Life Sciences
Module Code:	HPDM172
Module credits	15
% of Module Mark	100%

Module Convenor:	Neil Vaughan
Summative/Formative assessment:	Summative
Assignment Type:	Group Coding Assignment
Date Set	Monday 10 th November 2025

DATE DUE:	Friday 12 th December 2025
------------------	---------------------------------------

Feedback and marks released:	
-------------------------------------	--

1. Required Task

Two parts of this assignment

This assessment is worth 100% of the marks for the module. Within this assessment:

- Part 1 will be based on your coding project GitHub repository (Worth 90%).
- Part 2 will be a portfolio of evidence of effective teamworking (Worth 10%).

Overall Summary

Work as part of a team of around 4-5 students to:

- Develop a hospital database. Each team member should contribute to developing the database. The database should include planning with entity relationship diagrams, tables for the data and coding SQL queries. The database should be created in MySQL.
- Develop a portfolio of teamwork evidence including team meetings.
- Create one new GitHub repository per group, to store your group's project and portfolio, with each team member being listed as contributor to the GitHub repository with access to commit changes. When complete, include all files in your GitHub repository.

Hospital Database Requirements Description

Create a database with tables, relationships and queries to capture the descriptions below:

Table for Hospitals

Create a table with details of 40 hospitals. Each hospital should have a unique name, address, size (number of beds), type, accreditation status. Each of the doctors should work at one hospital and each hospital can have 0 or more doctors working at it.

Tables for Doctors and Patients

The database should contain a list of 100 doctors and 600 patients. You should randomly generate realistic names for the patients. Each patient should be assigned to only one doctor. Doctors should each be assigned a roughly equal number of patients. Additional related information should be stored about each doctor and patient at least including: name, date of birth, Address.

Table for medications

The database should contain a list of at least 30 medications. Generate data for the names of medications. A medication can be prescribed to multiple patients and each patient could be prescribed multiple medications.

Table for Prescriptions

The database should also contain a table which stores all prescriptions. Each prescription should be for a single medication to a single patient and should be prescribed by a single doctor on a date within the past 2 years. Generate 500 prescriptions fitting this criteria.

SQL Queries to extract various data

Create SQL Queries, so that the database has functionality to complete these tasks:

- Print a list of all doctors based at a particular hospital.
- Print a list of all prescriptions for a particular patient, ordered by the prescription date.
- Print a list of all prescriptions that a particular doctor has prescribed.
- Print a table showing all prescriptions ordered by the patient name alphabetically.

Table for Diseases

Create Diseases table, and define which disease is treated with which medication, and which doctor is a specialist for which disease.

Table for Appointments

A table should be created for appointments, to allow storing of individual appointments that are currently scheduled.

Lab Results table

A table should be created for storing the results from lab tests, which are for individual patients and each lab test should be requested by a specific doctor, for a specific type of test.

SQL Queries

Create SQL Queries, so that the database has functionality to complete the tasks below. Give your queries names that make it clear what the query does:

- Add a new customer to the database, including being registered with one of the doctors.
- Modify address details of an existing customer.
- Print a list of all patient names and addresses for patients registered to doctors based at one particular hospital - that could be used for posting information mail to all of one hospital's registered patients.
- Print a list of all doctors based at Teaching hospitals which were accredited between 2015-2024.
- List all patients who may have a particular disease based on which medication they have been prescribed.
- List all doctors based at who specialize in a particular disease.
- List all lab results for all patients over the age of 60.
- Print a list of all appointments for a given patient.
- Print a list of all appointments for a given doctor.
- Print all prescriptions made from a particular hospital ordered alphabetically by the name of the medication being prescribed - The output of this SQL query should include only these 4 columns: the medication name, the name of doctor who prescribed it, the name of the patient, and the name of hospital.
- Print a list of all lab results from all hospitals that were accredited between 2013-2020.
- Identify which doctor has made the most prescriptions.
- Print a list of all doctors at the hospital with biggest size (number of beds).
- A list of all hospital names which were accredited prior to 2015 and do have Emergency Service facilities.
- A list of patients registered with doctors who are based at hospital with <400 beds.

Entity Relationship Diagrams and Planning

Create planning documents such as any Entity Relationship Diagrams (ERDs), flow charts, or pseudocode. This could help to plan how the database tables will be organized, the relationships, the fields within each table, the datatypes planned to use for each field. This

will help to break down the task and plan what will be done. These documents should be included in the group's GitHub repository.

Developing within MySQL

In this assessment, the database should be developed within MySQL database. The MySQL version of your group's database should be included in your group's submission.

To submit your MySQL database, when your MySQL database is ready, you should export your MySQL database into a .sql file using mysqldump command. Then your .sql file can be pushed onto your group's GitHub repository. Instructions for exporting mysql databases and committing to GitHub are in the HPDM172 GitHub workshop.

Portfolio of evidence of collaborative working (10%)

As this is a group assignment, we want to see evidence that the group operated in a manner akin to what would be expected in a professional environment. Therefore you will be required to submit a portfolio of documents to showcase teamworking.

Compulsory documents to include in the portfolio:

- Meeting agendas (x2) (*template available on ELE page*)
- Meeting minutes (x2) (*template available on ELE page*)
- Action plan with milestones (*template available on ELE page*)

Please note that this portfolio will not be marked for content but compliance only. Therefore, if you submit all the documents and they are fit for purpose you will receive the full 10% for this component. These documents should be within a folder named TeamPortfolio within the repository and submitted zip file.

Dividing tasks between the team

Your GitHub submission will be made as a group, however it is up to you to how you distribute the workload, whilst developing the code and project. It is up to your team to decide which team members work on which parts. For example you might all contribute to multiple parts. Alternatively, one person may focus on the MySQL development, another person may focus on managing the GitHub repository or documentation. You can decide this yourselves within your team.

GitHub Repository contents

Each group will be required to create one shared GitHub repository to contain all their project database, files and code into. Through this repository we will be assessing to what extent you have met the assessment brief, so be sure to include all the required components and present a coherent and complete project repository. Any member of the group can be the initial creator of the repository. Every member of your group should be invited as a collaborator within the repository and accept the invitation. Ensure that the repository is set to Public to enable

Assignment Brief

markers to access the repository, and add a readme file to the project. Instructions for inviting team collaborators to a GitHub repository are in the HPDM172 GitHub workshop.

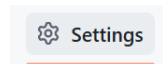
Readme Page

Your group's GitHub repository should have a Readme page that describes your database project as a short summary. For example this could include describing what the main files are, description of the database, tables, queries and how to use the files. Also describe what is the purpose of each SQL query.

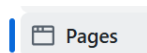
GitHub Pages

Your group's GitHub repository should have a 'GitHub Pages' webpage with short description of your project. For example the webpage can describe the project, or explain some of it's features. To set up a webpage:

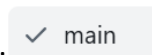
Enable GitHub Pages for the repository. GitHub Pages enables the repository to be viewed as a website. This will help the project to be viewed as a website, that will provide more information about your project. To enable GitHub Pages, click the gear settings icon:



Go to the Pages section:



Select the Main branch to be used for the web pages:



The website should now be public, on the URL in this format:

`https://{userid}.github.io/{reponame}` . Note that the repository name is case-sensitive.

Verify that your website can be accessed. By default, the webpage will show the project's readme file. However if you create a new file called index.htm then it will show this instead.

Checklist of Files to Submit on ELE and GitHub

Your group's GitHub repository should contain all parts of the developed project, including:

- Your group's planning documents such as any Entity Relationship Diagrams (ERDs), flow charts, or pseudocode.
- Your group's MySQL database exported as a .sql file.
- Teamwork portfolio of evidence including the meeting agendas and notes.
- The GitHub readme file containing brief description of the files in the repository and any files for the GitHub webpage.
- Any code you use, such as for generating random names and data.

Also on ELE each individual team member must submit the project as a single zip file including:

- A final copy of your group's whole repository contents.
- A URL hyperlink to your group's GitHub repository.
- Student Generative AI declaration form found on the module ELE page.

Use of GenAI:

This assessment falls under the category of **AI-Assisted** in the University's Guidance on use of GenAI in Assessment.

This is because using AI tools in specific ways can assist with your learning and will not inhibit fair assessment of your achievement of the module's intended learning outcomes.

This means: You may use GenAI tools ethically and responsibly to assist in the development of an assessment in accordance with the boxes checked below.

- ☒ To develop ideas
- ☒ To assist with research or information gathering
- ☒ To help you understand key concepts and theories
- ☒ To identify trends and themes as part of data analysis
- ☒ To provide feedback on a draft
- ☒ To improve the plan or structure of my assessment
- ☒ To generate images, figures or diagrams
- ☒ To proofread and correct spelling or grammar errors
- ☐ To format citations or references
- ☐ To translate material into English in line with guidance from your module lead. Unless this box is checked, you must never use AI translate more than a word or short phrase into English.
- ☒ OTHER [To generate data that will be inserted or appended into database tables]

When *writing* your assessment, you must never use AI tools:

1. For uses other than those represented by checked boxes in the list above.
2. To translate more than a word or short phrase into English unless agreed above.
3. To upload sensitive or identifying material to an AI tool
4. To present material that has been generated by AI as your own work or the work of someone else.

In assessments you should treat the AI tool like a citation from any other source. You must **include a list of all AI prompts and, where possible, hyperlinks to their output with your references**, at the end of your work. You do not need to include the outputs themselves, just the links.

Retain the full outputs generated by the prompts you have used during your assignment. These outputs should be accessible at the hyperlinks which you have submitted with your assignment. You may be asked to produce this material in the event of an academic conduct inquiry.

Submission

Your assignment must be submitted to ELE2 by the stated deadline. For instructions on how to do this, please see the Online Submission Student Handbook (<https://www.exeter.ac.uk/students/infopoints/yourinfopointservices/assessments/>)

If you submit your work late, but are within one hour of the deadline, a penalty of 5% will be deducted from your mark. For any work submitted more than an hour late your mark will be capped at 40% (up to 24hrs after the original deadline; after which a score of zero will be awarded). This is unless you have valid mitigation (please visit the [Mitigation SharePoint page](#) for more information.)

Please ensure that your name DOES NOT appear anywhere in the document, but that your personal ID number is included. You should submit your work as a PDF file.

There are no requirements for naming of your files. Your file will be renamed by ELE2 to a random submission number once you submit your work. However, it is recommended that you use a logical file naming system to help ensure that you submit the correct file for your assessment.

Please note that submitting an incorrect file, or submitting the correct file to the wrong location, will not be considered as grounds for mitigation. Late penalties will be applied if you submit the correct file to the right location after the submission deadline.

In submitting this you are declaring the work is your own independent piece of work, not produced in collusion with a fellow student or plagiarized from a fellow student, or a web site, or a textbook, or any other information source. You must demonstrate good referencing practise and ensure you have sufficiently paraphrased all sources of information. The direct copying of AI-generated content is also included under plagiarism, misrepresentation and contract cheating under definitions and offences in [TQA Manual](#) Chapter 12.3. During the submission process **you will be asked to check a box**, that confirms you have adhered to the university's academic conduct policy and the expectations on use of GenAI in your assessment brief.

Failure to do so may result in being referred to an academic misconduct panel which has the power to grant penalties based on specific criteria. For more details, please revisit the [Academic Honesty and Plagiarism](#) information on ELE.

2. Acquired skills

The skills this assignment develop are related to the specific goals identified in the relevant module ILOs being assessed by the assignment.

Intended learning outcomes (ILOs):

ILO: Module-specific skills

1. Demonstrate understanding and competence in fundamental skills for health data science.

2. Highlight the differences between computational tools and how to combine them to analyse health data.

ILO: Discipline-specific skills

3. Version control for reproducible analysis pipelines.

4. Using Openstack and the Linux command line to perform advanced computing tasks.

ILO: Personal and key skills

5. Working collaboratively on developing software.

6. Dynamically learning new computing skills.

3. Sources of support

What lectures/workshops provide support for this assessment?

*What previous assessments have used relevant skills (core or prerequisite modules only)?
Direct to feedback from these.*

International students and those who have English as a second language can make use of the [English Language Skills Development programme](#). This programme provides face-to-face workshops and courses as well as one-to-one assignment writing tutorials and an extensive range of online resources via their [Guided Independent Learning site](#). Find out more by browsing the timetables:

<https://www.exeter.ac.uk/into/englishlanguage/howtoparticipate/timetablesupport/>

or emailing inseasonal@exeter.ac.uk

4. Marking Criteria

- Part 1 will be based on your coding project GitHub repository (Worth 90%).
- Part 2 will be a portfolio of evidence of effective teamworking (Worth 10%).

This assessment will be marked using the standard assessment criteria as detailed on SharePoint

(<https://universityofexeteruk.sharepoint.com/sites/MedicalSciencesHomePage/SitePages/Assessments.aspx>), using the following criteria.

- Planning the task such as with ERDs adequately used, pseudocode or flowcharts.
- Interaction and contribution from each team member on contributions to the team meetings, planning, coding and GitHub repository evidenced from team portfolio.
- Functionality of the code SQL developed.
- Clarity and organisation including naming conventions of tables and queries.
- Database relationships adequately set up between tables based on ERD.
- Working SQL queries used to achieve the desired functionality.
- Appropriate data types, field names and correctly importing external data files.

Grade boundaries:

Assignment Brief

For selecting the overall score of a piece of work notched marking is used. This table gives a reminder the grade classifications.

Overall Grade	<i>Exceptional 1st</i>	<i>1st</i>	<i>2:1</i>	<i>2:2</i>	<i>3rd</i>	<i>Fail</i>	<i>Clear Fail</i>
Mark(%)	>85	82,78,75,72	68,65,62	58,55,52	48,45,42	38,35,32	<30

For more information about how coursework is marked and moderated please see the 'Marking of Coursework' heading in the Assessment tab on the Medical Sciences homepage. This can be accessed [here](#).

Marking Rubric 1

Task	Excellent	Good	Acceptable	Poor
Importing data from files and setting the IDs between tables.	Clear evidence provided of importing data from files and setting clear IDs to link between tables.	Some examples provided of importing data from files and setting IDs to link between tables.	Limited evidence of importing data from files and setting IDs to link between tables.	Little or no importing data and changes would be required to the IDs to link between tables.
Clarity and structure of the developed SQL queries.	Good clear evidence of clear SQL queries organized consistently with good naming convention.	Some good evidence of SQL queries structured and organized with naming convention.	Some evidence of SQL queries organized.	Improvements could be of benefit to the structure and organization of the SQL queries.
Generating data to import using code or other methods.	Clear evidence of methods or code developed for tasks including generating the data and tasks required for the database.	Good evidence of code developed for tasks including generating data and tasks required for the task.	Attempts demonstrated to develop some of the code for various parts of the task described in the brief.	Little or no progress demonstrated on the development of code based on the assignment brief.
Database relationships adequately set up between	Clear evidence of the planning using ERD, good implementation	Good evidence of some planning using ERD, and good	Some planning using ERD, and attempts at several parts of	Little or no planning using ERD, and only minimal parts of

Assignment Brief

tables based on ERD.	of the one-many relationships using the database software.	attempt at a majority of implementation of relationships using database software.	implementation of relationships using database software.	implementation of relationships using database software.
SQL queries used to achieve the desired functionality.	Clear evidence of good grasp of various SQL keywords and methods for selecting and inserting data from specific tables in particular specific cases.	Good evidence of grasp of various SQL keywords and methods for selecting and inserting data but could improve specific aspects of detail in some tables.	Some evidence of using various SQL keywords and methods for selecting and inserting data but could improve specific aspects of detail in some tables.	Little or poor evidence of using SQL keywords and methods for selecting and inserting data and substantial improvement required in details within developed SQL.
Database developed in MySQL.	Clear evidence of good grasp of implementing the database on MySQL.	Good evidence of grasp of implementing the database on MySQL.	Some evidence of using both platforms for development but may be some missing elements.	Little or poor evidence of using MySQL platform or database developed in one platform.
Use of joins and SQL syntax to select specific queries combining data from multiple tables	Clear and good use of joins in SQL to select specific queries combining data from multiple tables	Good use of joins in SQL to select specific queries combining data from tables	Some use of SQL to select specific queries combining data from tables	Limited use of SQL to select queries from tables

Marking Rubric 2

	Exceptional - highly effective (5 marks)	Excellent - very effective (4 marks)	Limited - Some Weaknesses (3 marks)	Good - Successful (2 marks)	Poor - Significant weaknesses (1 mark)	Absent (0 marks)
Organisation of group project submission						
Setup of						

Assignment Brief

appropriate tables and fields						
Relationships and connections between tables						
Code or method for generating random data						
The coding of the SQL statements for the tasks						
Functionality of SQL queries for the various tasks						
Planning such as with Entity Relationship Diagrams						