



COMP1850 Programming

School of Computer Science, University of Leeds

Worksheet – Semester 2 Week 2

This worksheet contains a combination of formative activities and activities that contribute towards your portfolio.

- Portfolio activities are indicated in the title of the task.
- Activities marked by (*) are advanced activities and may take more time to complete.

Expectations:

1. **Timeliness:** You should complete all the tasks in the order provided and submit your portfolio task to Gradescope before the deadline.
2. **Presentation:** You should present all your work clearly and concisely following any additional guidance provided by the module staff in the module handbook.
3. **Integrity:** You are responsible that the evidence you submit as part of your portfolio evidence is **entirely your own work**. You can find out more about Academic integrity on the Skills@library website. All work you submit for assessment is subject to the academic integrity policy.

Feedback: Feedback on formative activities will be provided via Lab classes and tutorials. Feedback on evidence submitted as part of the portfolio will be available on Gradescope.

Support opportunities: Support with the activity sheet is available in the Lab classes and tutorials. Individual support is available via office hours.

Expected complete date: Friday 13th February

All portfolio tasks are rated AMBER by the university: AI tools can be used in an assistive role. You are permitted to use AI tools for specific defined processes within the assessment.

Within this assessment you **may** use Generative AI to:

- help you start to understand more complex ideas by providing accessible summaries
- test your knowledge against a piece of content
- help identify and correct spelling mistakes and grammatical errors in your work
- provide feedback and advice on your overall coding style

You must **not** use Gen AI to:

- produce written content (code, comments or any other content) which you then submit as your work, regardless of whether you make changes to it.
- rewrite or make changes to any of your work.

Learning Outcomes:

- Compose SQL queries to retrieve data from a database
- Articulate the components of a relational data model
- Develop programs using functions with parameter passing.

Task 1 – Exam Style Question

You are working with a database used by a university library system.

The system tracks library members, books, and borrowing activity.

The database contains the following tables:

members(member_id, member_name, join_date)

books(book_id, title, author)

loans(loan_id, member_id, book_id, loan_date, return_date)

- a) For each pair of tables below, **state** the type of relationship (one-to-one, one-to-many, many-to-many) and briefly explain your reasoning.
 - i. Members and loans [2]
 - ii. Books and loans [2]
 - iii. Members and books [2]
- b) A query joins members to loans using an INNER JOIN.
 - i. Explain what happens to members who have never borrowed a book. [2]
 - ii. Explain how the results of the query would change if a LEFT JOIN were used instead. [2]
- c) The head librarian would like to see how many books have been borrowed by each library member.
 - i. Write an SQL query which would show the name of each library member and how many loans they have taken out. [5]
- d) The head librarian asks ‘Why don’t you store the book title with the loan? Wouldn’t that make it easier to see the data?’
 - i. Explain using appropriate non-technical language why this would be bad database design. [5]

You can answer these questions on paper for your own notes, or you can answer them digitally in the weekly repository if you prefer.

Task 2 – Portfolio Question

This task will be completed on Github, where you are asked to complete three functions to get specific data out of a database.

The full task is given in the task_2 folder, where you will find:

- cinema.py – this is where you will fill out the functions and should be submitted to Gradescope
- diagram.png – the database diagram
- example_output.txt – an example of the output you should expect when you test your code
- task.md – explains the full task and gives some tips – you should refer to this for full information about what you need to do.
- test.py – this program will import and use the functions you write, and is how you should test your code works. It will give you a menu so you can test each function independently
- tickets.db – the database file.

You need to ensure that you're returning the data in the right format, and that you have read the full instructions for each task as each query has several lines. Make sure you think about what SQL features you need to use, such as aggregators, groupings and order.

When you submit cinema.py to gradescope it will give you some feedback on mismatches, but you should rely on running your own code using test.py before you submit your work.

You **must not** edit the function names or argument, or hard-code values – this will cause your program to fail to run, as the tests run the functions exactly as defined in cinema.py.

Task 3 - Stretch

This week, you have been given significant amounts of more independent work to complete – if you want to continue working on your SQL then you should look at what additional features you could add to the session 2 task.

For example, you could add:

- Making a new order
- Editing order details
- Different data which management may want to be outputted to them
- A way of exporting data into a excel-friendly format