**Assessment Overview**
The main component of assessment for this module is a programming assignment (70% weighting). Assessment of work is based partly on the software artifact submitted (does it work as advertised, is it well-structured and readable) and partly on a summary report which should demonstrate reasoning, reflective and critical thought. It is therefore important to record thoughts, decisions, and designs in some form of personal journal; writing a suitable report is then much simpler.

The task is: design, implement, populate, and query a database on the basis of the "Company Database" specification provided (See attached)

The supplementary materials provided as part of this course (eg .csv files) may be useful for populating parts of your database.  It is expected that the database will contain several tables (or collections), some of which must be populated with generated test data or test data loaded from external files.

Once your database is working as required the next step will be to reflect upon how it might be improved and verified. Can more data be added (loaded) easily?  What kind of file format would you use for importing/exporting data from the database?

The assessment is divided into two main artefacts:

1. A working database allowing you to demonstrate a non-trivial operation on the data (JOINs, UNIONs, VIEWs, using procedures and/or functions), documented with a diagram (ERD and/or Relational Model), implemented with appropriate SQL (DDL) code, including queries (views) and PL/SQL code.
2. A brief report describing how you have included the requirements below, and your findings from processing the data.

**The populated database**
You should develop:
I) A model of the database (ERD and/or Relational Model).
II) Code for implementing the database (DDL code) and PL/SQL subprograms.
III) Code for populating the database (DML code)
IV) Code for querying the database (queries, views)
V) Include a front-end, in form of an (online) database application.

Use of Generative AI Tools such as ChatGPT, DALL-E, Bard etc. is explicitly prohibited for this part of the assessment (“The populated database”). The output gleaned or generated from
the tools mentioned and others that are similar relates to sources already published or available. Consequently, using these tools without correct referencing is deemed plagiarism.
Also, be aware that the information obtained can be inaccurate or incomplete. Thus, all submitted work should be your own. If the assessment is found to have been plagiarised or
to have used unauthorised AI tools, you will be referred to the Student Disciplinary Officer within the School and this may result in an Academic Misconduct charge.
You may use specific tools, such as Grammarly for aiding Grammar and Structure.
 a. It is recommended that you keep a record of your interaction with these tools.
 b. It is recommended that you have evidence of drafts of your work.

**The Report**
Your report should be brief (approximately 1500 words) and describe the goal of your application, the dataset used and the generated test data, a description and reasoning of the methods and choices for data processing and a critical reflection and evaluation based on your results.

**Grading the Assessment**
The report (40% weighting)
Should be well written and structured in a style suitable for a technical report (Introduction, Method: Database design and coding with reasoning, Results: Reflection & Evaluation and Conclusion etc.).

The database code (60% weighting)
The database should be coded efficiently and properly documented. You will need to submit the DDL code, the code for populating the database, any DML code, and PL/SQL code, and screen shots of the database application.

**Submission**
You must submit your written report and database code separately, along with all dataset files in a ZIP file
\* Database code: SQL (DDL and DML code), PL/SQL code.
\* Report: approximately 1500 words (+/-10%) of body text (excluding annotations, code fragments, references, appendices etc.) including images illustrating the software in use.