|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **ASSIGNMENT COVER PAGE** | | | C:\Users\hoching.tay\Desktop\Lincoln_UK_06092017-01.png |
| **Programme** | | | **Course Code and Title** | | |
| Bachelor of Computer Science (Hons) | | | CDB3033N (Database Programming) | | |
| **Student’s name / student’s id** | | | **Lecturer’s name** | | |
|  | | | Ts. Chng Chern Wei | | |
| **Date issued** | **Submission Deadline** | | | **Indicative Weighting** | |
| Week 8 - 30/10/2023 | Week 11 - 20/11/2023 | | | 30% | |
| **Assignment [2]** | Cursor Processing and Physical Database Design (Group Assignment) | | | | |
| This assessment assesses the following course learning outcomes | | | | | |
| **# as in Course Guide** | **UOWM KDU Penang University College Learning Outcome** | | | | |
| **CLO3** | Develop database applications implementing business rules with stored procedures. (C5, PLO3) | | | | |
| **CLO4** | Apply access methods to improve database performance and to build cost models. (C3, PLO7) | | | | |
| **# as in Course Guide** | **University of Lincoln Learning Outcome** | | | | |
| **CLO1** | Use appropriate tools and techniques to design a database | | | | |
| **CLO2** | Appraise the structure of a database design using standard evaluation mechanisms | | | | |
| **CLO3** |  | | | | |
| **CLO4** |  | | | | |
| **Student’s declaration** | | | | | |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.  Student’s signature: Submission Date: | | | | | |

|  |  |
| --- | --- |
| **Dates and Mechanisms for Assessment Submission and Feedback** | |
| **Mechanism for handout to students** | Open Learning |
| **Mechanism for submission of work by student** | *Softcopy online submission via Open Learning* |
| **Date by which work, feedback and marks will be returned to students** | 30th Nov 2023 |
| **Mechanism for return of assignment work, feedback and marks to students** | Feedback will be provided by a marking template. This will be available to students via Open Learning. The discussions at the walkthroughs will also provide informal feedback |

# COURSEWORK SUBMISSION GENERAL INFORMATION

# Academic Integrity Statement

You must adhere to the university college regulations on academic conduct. Formal inquiry proceedings will be instigated if there is any suspicion of plagiarism or any other form of misconduct in your work. Students must **NOT** collude with other groups of students or plagiarize their work.

**We practice zero tolerance towards plagiarism, and we use Turnitin to evaluate the similarity index. Your similarity index score must not exceed 20%.**

**Your tasks must be your own work. Unless the use of Artificial Intelligence (AI) is permitted in your assessment task, using AI to complete your assignment is a form of plagiarism.**

# Nature of the submission required

A softcopy of your assignment in **PDF version** should be submitted to lecturer, no later than the date and time stipulated on the cover sheet. In addition, an electronic copy of your work must be submitted to Turnitin. The first page of your report, immediately after the cover page, must be a page from Turnitin clearly showing your name and your Originality Score (Please refer to [submission arrangement](#_Submission_arrangement)).

Diagrams may be used where they are helpful to support your arguments or description. If they are not your own work, the source must be referenced. Please help us to handle and mark your work efficiently.

Please take note for group submission, only **one submission per group**. This will contain both the group and individual elements. The individual element must be clearly labelled to indicate which group member completed the task.

# Documentation guidelines

Student is required to submit a **SOFTCOPY** of the report and ensure that it use the following formatted styles: 1) Font type: **ARIAL**, 2) Font size: **11** **pt**., 3) Line spacing: **Single spacing** and 4) Page layouts: **Justify**. Please make sure you have proper format alignment for all paragraphs, following standard writing style and use **HARVARD CITATION STYLE** for citation. Please include a **HEADER** with the following information: **Student ID, Student name, Course code and Assignment type**. Please also include a proper cover page for your submission which contains information about the students, assignment, course, and department with KDU and University of Lincoln (UoL) logos on top. Also include page number and list of references, which is shown in the last page.

# Penalties for Late Submission

For late submission of this Assignment, a penalty of a reduction by 10% of the maximum mark may be applicable for each Calendar Day or part thereof that the submission is late. An Assignment submitted more than **TEN** Calendar Days after the deadline will have a mark of zero recorded for this Assignment.

# Submission arrangement

1. Cover page
2. Turnitin similarity report
3. Table of Content
4. Main Report
5. Reference List or Bibliography List (whichever applicable)

Marking Rubric (in landscape orientation)

# Assignment instructions/Background (2-3 students per group)

In this group assignment, you are required to use the new database called ***Publication***. The database consists of seven schemes namely author, publication, wrote, proceedings, journal, article and book.

**Task 1 (40%)**

Write a MYSQL procedure called *print\_publication* that print a list of publication records for each publication by an author name as parameter input. You should sort the name of the author for the publication in ascending order and follow by the year of publication (earliest first). The sample output format is as follows:

Publication ID | Publication Title | Publication Year

-------------- | ---------------------------- | --------------

1 | Sample Publication 1 | 2010

2 | Sample Publication 2 | 2012

3 | Sample Publication 3 | 2012

4 | Sample Publication 4 | 2013

**Task 2 (40%)**

1. Create a new table called *publication\_master*. This table is used to consolidate the all 5 relations namely publication, proceedings, journal, book and article. Introduce a new column called *type* to specify the category of publication (either journal, book, etc.).

Each category of publication is having common or different types of details such as a book has publisher and year and on the other hand, the journal has volume, number and year. Your new table should hold these values with common column names such as *detail1*, *detail2*, *detail3*, and *detail4*.

Write a MYSQL stored procedure called *merge\_publication* to transfer all the data from five respective tables into the *publication\_master* table. This procedure should indicate how many records successfully posted into the master table upon execution to standard output. Besides, give an appropriate error message for any unsuccessful cases such as no publication details found in proceedings, journal, article, and book tables.

1. Based on the *merge\_publication* table create in part 2a, create another MYSQL stored procedure called *print\_article* to print a list of publication records for each article appearing in a proceeding, journal or book which identified by a *pubid* supplied as an input parameter. The articles should be displayed in ascending order on starting page number.

Call the procedures from an anonymous block and capture the results. Appropriate exception handling should be performed for the above cases by sending the error message to the standard output

**Task 3 (20%)**

You have been asked by the client to investigate the physical design of the **publication** database. Prepare a simple report (300-400 words) to review the design of the database and provide appropriate approaches/ways to improve database performance in accessing or searching the publication records. The report should contain a fully justified set of recommendations proposing an appropriate solution (e.g. using appropriate index in particular search column).

# Submitting your work

You should submit the following:

* A complete source code (MYSQL) in softcopy and hardcopy format. Save your solution according to the question number (e.g. q1.sql, q2a.sql …etc)
* Report:
  + Title page. Include the names and ID's.
  + Appropriate screenshots for all the test cases of question 1, question 2a and question 2b. A review report for question 3.
  + Individual Report
    - A review (500+ words). This review should include two parts:
      * Part1: Discuss how the course affected you, reflecting on what you have achieved. Please include in your report what went well; any design/implementation problems encountered and how you solved them.
      * Part2: Teamwork is an integral part of the Software Development Project course and Personal Development. ***The teamwork mark is awarded to each team member individually***. In this review, you should write on your personal experience of teamwork and how the group dynamics worked in the group coursework. You are asked to assess yourself and each of the other members of your team in terms of:
      1. Contributing to the planning of the work.
      2. Contributing to the leadership and management of the work of the team as it progressed.
      3. Contributing to the final products.
  + Conclusion
  + References
  + A completed copy of the group member contribution form on the last page of this document as agreed by all members of the group.
  + An appendix with
  + Program listings for question 1, question 2a, and question 2b.
  + The code should be well structured and self-documentation features such as sensible variable names and comments should be used.
* Upload the following items through MS Teams:
  + All the query files and source code for Task1 & Task2.
  + A Report in softcopy format
  + Turnitin report in softcopy format.

It is important to include citation and references in the assignment. You are required to upload your document to turnitin.com for plagiarism checks.

**Task 3**

Database performance can also be improved by optimizing query techniques. It can be achieved by following best database querying practices that were already cemented through community trial and errors. This includes minimizing the use of wildcard characters, avoiding subquery usages, using row limiters such as LIMIT or TOP, using EXISTS instead of IN, and using stored procedures (Sarang, 2023), as doing the opposite may cause inefficient or unnecessary CPU utilization. To gauge query performances, developers can use SQL EXPLAIN statements to analyze a query’s execution plan (MySQL, n.d.).

Finally, a periodical data defragmentation maintenance should be enacted to achieve optimal database performance. The process of data undergoing constant rewriting and removal from a storage device will inevitably cause its data to become fragmented in the storage space, which bottlenecks the data retrieval process (Buchanan, n.d.). Data defragmentation helps by rearranging data on the storage medium and group relevant data together, eliminating holes between the data storage space (Yasar, 2023). Ultimately, this allows I/O related operations to be more responsive and efficient, including database processes (Buchanan, n.d.).

**Group members work contribution form**

**In percentage, please indicate the work contribution** of each member. This should be agreed by all group members**. The total of all members work must add to 100%**

You must **submit this form in your final report.** Put your initials in the signature columns. This copy must be signed by all members.

**Group/Team Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_­­­­­­­­­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |  |  |
| --- | --- | --- | --- |
| Team member name | Student ID | Individual overall work contribution (%) | Signature |
| Student: |  |  |  |
| Student: |  |  |  |
| Student: |  |  |  |
| Student: |  |  |  |
| **Total 100%** | | |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CDB3033N DATABSE PROGRAMMING**  **MARKING RUBRIC**  **ASSIGNMENT [2]**  **STORED PROCEDURE, CURSOR, & ACCESS METHODS** | | | | | | | | | |
| **Section (1)-80%** | | | | | | | | | |
| **LEARNING OUTCOME** | **MARKING CRITERIA** | **SCALE** | | | | | | | |
|  | **Fail**  **(0-49)** | **3rd Class**  **(50-59)** | **2nd Lower Class**  **(60-69)** | **2nd Upper Class**  **(70-79)** | **1st Class**  **(80-100)** | **YOUR MARKS/COMMENTS** | | |
| **100%** | **Weightage** | **Actual Marks** |
| CLO3 | **Task 1 (40%)**   * **Store procedure implementation (15%)** * **Cursors Implementation (15%)** * **Correct output (10%)** | Little or no attempt to implement the feature correctly | A partial implementation of the feature, but some aspects are incorrect and not particularly well coded. May give rise to run-time errors. | Appropriate implementation of the features but with some minor flaws and some aspects are not well implemented. | A mostly complete implementation of the feature which works correctly, although the coding could be clearer. | A complete implementation of the feature, clearly coded. |  | 0.4 |  |
| **Task 2a (25%)**   * **Correct table creation (5%)** * **Insertion procedure with business rules (20%)** | Little or no attempt to implement the feature correctly | A partial implementation of the feature, but some aspects are incorrect and not particularly well coded. May give rise to run-time errors. | Appropriate implementation of the features but with some minor flaws and some aspects are not well implemented. | A mostly complete implementation of the feature which works correctly, although the coding could be clearer. | A complete implementation of the feature, clearly coded. |  | 0.25 |  |  |  |
| **Task 2b (15%)**   * **Display procedure (10%)** * **Correct output (5%)** | Little or no attempt to implement the feature correctly | A partial implementation of the feature, but some aspects are incorrect and not particularly well coded. May give rise to run-time errors. | Appropriate implementation of the features but with some minor flaws and some aspects are not well implemented. | A mostly complete implementation of the feature which works correctly, although the coding could be clearer. | A complete implementation of the feature, clearly coded. |  | 0.15 |  |
| **Total (80%)** | | | | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CDB3033N DATABSE PROGRAMMING**  **MARKING RUBRIC**  **ASSIGNMENT [2]**  **STORED PROCEDURE, CURSOR, & ACCESS METHODS** | | | | | | | | | |
| **Section (1)-80%** | | | | | | | | | |
| **LEARNING OUTCOME** | **MARKING CRITERIA** | **SCALE** | | | | | | | |
|  | **Fail**  **(0-49)** | **3rd Class**  **(50-59)** | **2nd Lower Class**  **(60-69)** | **2nd Upper Class**  **(70-79)** | **1st Class**  **(80-100)** | **YOUR MARKS/COMMENTS** | | |
| **100%** | **Weightage** | **Actual Marks** |
| CLO4 | **Task 3 (20%)**   * **Physical database review (10%)** * **Improvement (10%)** | A poor piece of documentation has been produced. The coverage is unclear, and/or there are significant omissions | A fair piece of documentation has been produced, although there are some weaknesses—either the coverage is not particularly clear, or some aspects have been omitted | An appropriate piece of documentation has been produced but with minor flaws or some aspects are not well elaborated. | A good piece of documentation has been produced, providing detailed and clear coverage of the aspect concerned, although there may be a number of minor flaws which prevent it being regarded as excellent | An excellent piece of documentation has been produced, providing full and clear coverage of the aspect concerned |  |  |  |
| **Total (20%)** | | | | | | |  | | |
|  | | | | | | | | | |
| **Overall Score (100%)** | | | | | | |  | | |