**XJTLU Entrepreneur College (Taicang) Cover Sheet**

|  |  |  |
| --- | --- | --- |
| Module code and Title | CPT103TC: Introduction to Database | |
| School Title | School of AI and Advanced Computing | |
| Assignment Title | Assessment Task 002 (CW) | |
| Submission Deadline | **2nd June 2022 at 5:00 PM** | |
| Final Word Count | NA | |
| If you agree to let the university use your work anonymously for teaching and learning purposes, please type **“yes”** here. | | **Yes** |

I certify that I have read and understood the University’s Policy for dealing with Plagiarism, Collusion and the Fabrication of Data (available on Learning Mall Online). With reference to this policy I certify that:

* My work does not contain any instances of plagiarism and/or collusion.  
  My work does not contain any fabricated data.

**By uploading my assignment onto Learning Mall Online, I formally declare that all of the above information is true to the best of my knowledge and belief.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Scoring – For Tutor Use** | | | | | | | | |
| **Student ID** | | | |  | | | | |
|  | | | | | | | | |
| **Stage of Marking** | | **Marker**  **Code** | **Learning Outcomes Achieved （F/P/M/D）**  **(please modify as appropriate)** | | | | | **Final**  **Score** |
| **1** | **2** | **3** | **4** | **5** |
| 1st Marker – red pen | |  |  |  |  |  |  |  |
| Moderation  – green pen | | **IM**  **Initials** | The original mark has been accepted by the moderator (please circle as appropriate): | | | | | Y / N |
|  | Data entry and score calculation have been checked by another tutor (please circle): | | | | | Y |
| 2nd Marker if needed – green pen | |  |  |  |  | | |  |
| **For Academic Office Use** | | | **Possible Academic Infringement (please tick as appropriate)** | | | | | |
| **Date**  **Received** | **Days late** | **Late Penalty** | **Category A** | | Total Academic Infringement Penalty (A,B, C, D, E, Please modify where necessary) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | |
|  |  |  | **Category B** | |
| **Category C** | |
| **Category D** | |
| **Category E** | |

**Students**

**(Please modify where necessary)**

The assignment must be typed in an MS Word document and submitted as pdf via Learning Mall Online. Only electronic submission is accepted and no hard copy submission.

All students must download their file and check that it is viewable after submission. Documents may become corrupted during the uploading process (e.g. due to slow internet connections). However, students themselves are responsible for submitting a functional and correct file for assessments.

**Assessment 002: Coursework**

**Due:** June 2nd 2022 @ 17:00

**Weight:** This Coursework counts for 40% of this module.

**Maximum Marks:** 100

The course work will be assessed for the following learning outcomes:

1. Demonstrate a basic understanding of the design of databases.
2. Show a fundamental grounding in the operation and usage of database management systems including "hands-on" experience of a basic database management system.
3. Demonstrate in-depth knowledge of the database language, SQL.

**Assessment Tasks:**

Your task is to create a database to represent current Coronavirus data from an Open Data Source, in order to be able to answer questions and perform simple analysis. This will involve creating and normalizing a database to hold the data from a dataset and constructing queries against the data once it is in a suitable form.

There are the following 4 parts of this coursework

The Relational Model (20%)

Normalization (25%)

Database construction (15%)

Querying (30%)

Visualization (10%)

You will be required to write a brief report to answer the questions and note down your process, thoughts, and assumptions made as well as answers. You will also need to construct a database using Oracle Application Express (APEX) or SQLite and produce a set of queries that can be run against that database.

**Marking Criteria**

This coursework will be marked out of 100. There are 20 questions and 5 marks available for every question. Marks will be awarded as follows:

5 marks: Complete and correct answers with rationale

4 marks: An answer is correct but rationale/justification is partially correct

3 mark: Partially attempted/correct

2 and 1 marks, based on the level of attempt/correctness

0 marks: Not attempted/wrong

**Dataset**

The dataset to be used for this exercise is COVID-19 Coronavirus data downloaded from the EU Open Data Portal. The dataset contains the latest available public data on COVID-19.

The dataset and data dictionary can be downloaded from LMO under the Assessment 002-Coursework.

**Setup**

You can complete this coursework either using Oracle Application Express (APEX) or SQLite. All you need to submit SQL scripts for constructing and querying the database along with the report.

**Q1: The Relational Model (4\*5=20 Marks)**

**Q1(a):** Write down the relations that directly map the dataset file into a mega table/relation. You don’t have to create SQL code for transforming the dataset into a table, just list down relations and attributes with appropriate data types, here is an example:

*R1 (A1 [data type], A2[data type]…An[data type])*

*Where R1 refers to relation and A1 to An are set of attributes*

***Note:*** *You must use the attribute names exactly as they appear in the dataset.*

**Types:** To ease the modeling process, you can either use SQLite data types (INTEGER, TEXT, BLOB, REAL, and NUMERIC) <https://www.sqlite.org/datatype3.html>

OR Oracle data types (NUMBER, VARCHAR2, CHAR, CLOB, BLOB, etc.) <https://docs.oracle.com/cd/B12037_01/server.101/b10759/sql_elements001.htm>

**Q1(b):** List the minimal set of Functional Dependencies (FDs)

* Every FD must be minimal on its LHS (left-hand side)
  + There can be more than one attribute on the LHS, but there should be no attributes on the LHS that add no further information (e.g. If A -> C, then A, B -> C would not be minimal as the B is not adding anything further)
* There should be no trivial FDs (a -> a adds nothing of value)
* There must be no redundant FDs (those which are already implied by other FDs)
* Tip: If A -> B and C -> B and C -> A then C -> B is redundant (as this is implied by C -> A -> B)

***Tip:*** *Explain any assumptions you make applying what you know of the domain to the data and consider future data and the impact it may have as well.*

***Tip:*** *You will need to think and determine whether values are 'blank' (a known value of blank) or null (an as yet unknown value) as this may have an impact on your dependencies. Explain any assumptions and decisions you make in the report.*

**Q1(c):** List all potential candidate keys with justification

***Tip****: Remember, a candidate key contains no redundant attributes*

**Q1(d):** Identify a suitable primary key, and justify your decision

**Q2: Normalization (5\*5=25 Marks)**

In this exercise, you will decompose the mega relation into a set of normalized schemas and explain the role of each schema in the database.

You should apply step by step procedure to transform the unnormalized tables created in Q1 to create a normalized database. Each schema must be in the Third Normal Form. Please consider the following guidelines while answering the below questions:

**Keys:** Where possible, you should only introduce new Surrogate Keys to avoid anomalies, you should explain this in your report with a justification.

**Attributes:** While you are able to introduce new attributes if you wish; you must not rename or remove or change the values of any of the attributes in the original relation. All must appear as originally named in your decomposed relations.

**NULL values**: NULL values are not values in themselves, but represent unknown values in the dataset

(You cannot treat all NULL values as the same 'null' value). NULL values can be present throughout the

normalization process, you do not need to remove them. However, you may find you need to introduce

surrogate keys in the case where a NULL could or is present in something you would want to be a key or

split into a relation.

**Q2(a):** List any partial-key dependencies from the mega relation as it stands in the dataset file and any resulting additional relations you should create as part of the decomposition.

**Q2(b):** Convert the relation into 2nd Normal Form using your answer to the above. List the new relations and their fields, types, and keys. Explain the process you took.

**Q2(c):** List any transitive dependencies, if any, in your new relations and justify your answer

**Q2(d):** Convert the relation into 3rd Normal Form using your answers to the above. List the new relations and their fields, types, and keys. Explain the process you took.

***Note:*** Depending on assumptions you have made in earlier processes, tables may already be in 3NF - in

that case, demonstrate why a table is in 3NF.

**Q2(e):** Is your relation in Boyce-Codd Normal Form? Justify your answer.

**Q3: Database Construction (3\*5=15 Marks)**

**Q3(a):** Import the raw dataset into a single table called 'dataset'.

***Tip:*** *To import the dataset in SQLite DB Browser, you should create a database called coronavirus.db and then use File->import-> Table from CSV file.*

***Note:*** *You should not change the CSV file - it must be the original provided dataset file. The attribute names must be the same as in the original file.*

Export this table as dataset.sql (including CREATE and INSERT statements), such that running it will import the full dataset into a fresh database.

The entire database at this point should be saved as **dataset.sql.**

In the report write down the process you took and submit **dataset.sql** along with the report.

**Q3(b)**: Write the SQL to create the full normalized representation, including all tables as a result of the decomposition process you took for Q2, excluding the mega dataset table.

The SQL should contain CREATE statements to create all new tables. You should include foreign keys where appropriate, and list and justify these in your answer.

**Q3(c):** Write INSERT statements using SELECT to populate the new tables from the 'dataset' table.

Save both *(CREATE and INSERT statements in a single file)* as Normalized\_*Data.sql and run* them to populate the normalized database.

Each SQL statement should be written in the report, as well as saved. You should explain the steps for each of the above questions. Briefly write down the process you went through to go with it - enough that a person with just your report could reproduce what you have done.

Submit dataset.sql (containing original dataset) and Normalized\_Data.sql.sql along with your report.

**Q4: Querying (6\*5=30 Marks)**

For each exercise in this question, you will need to write a SQL query against your newly created normalized tables in your database. The queries created over the unnormalized mega dataset table will be considered wrong.

Each SQL statement should be written in the report, as well as all queries should be saved in a single file with numbers such as 4a, 4b, and so on, as sqlScript.sql which can be run against your normalized database, as it stands at the end of Q3. You should also briefly describe your approach for each in the report.

Write an SQL statement for each of the following:

**Q4(a):** The worldwide total number of cases and deaths with total cases and total deaths as columns.

**Q4(b):** The total number of cases and deaths for each country (with country, total cases, and total deaths as columns)

**Q4(c):** The number of cases and deaths by date, in increasing date order, for each continent with the continent, date, number of cases, and number of deaths as columns)

Tip: You may choose whether to organize by date first or by continent first

**Q4(d):** The total number of cases and deaths as a percentage of the population, for each country with

country, % cases of population, % deaths of population as columns.

**Q4(e):** A descending list of the top 10 countries, by percentage total deaths out of total cases in that country (with country name and % deaths of country cases as columns)

**Q4(f):** The date against a cumulative running total of the number of deaths by day and cases by day for the United Kingdom (date, UK\_daily\_cases, UK\_cumulative\_cases, UK\_daily\_deaths, and UK\_cumulative\_Deaths as columns)

***Note:*** *Cumulative deaths refer to the addition of all the deaths to a specific date. For example, Deaths: 1, 6, 10, 20, Cumulative Deaths would be: 1, 7, 17, 37.*

**Q5: Visualization (2\*5=10 Marks)**

**Q5(a)**: Write a SQL script to plot the top 10 countries in terms of overall cumulative deaths only. Include an explanation of your script in the report.

**Q5(b):** Produce agraph that has a date on the horizontal axis and the cumulative number of deaths by country on the vertical axis. You should represent the top 10 countries in terms of overall cumulative deaths only.

The full script and resulting screenshot of the graph should be included in the report.

**Submission**

**Report**

You must write your report in MS word and submit it as a PDF. The name of the pdf file should be

yourID\_report.pdf.

Your report should contain:

The first page of this document with your ID for grading purpose

You must have a section for each part of the coursework, and they must be named as follows:

1. The Relational Model

2. Normalization

3. Database construction

4. Querying

5. Visualization

**Files**

Your submission should be a single zip file. Your submission must include the following in this structure:

1. *Report.pdf*
2. *dataset.sql:* the create and insert statements of the unnormalized dataset after importing the dataset
3. *Normalized*\_*Data.sql*: The full database after creating and populating the normalized tables (CREATE and INSERT statements of the normalized dataset
4. *sqlScrip.sql:* SQL statements to run queries against normalized tables

**Support**

Any questions and answers will be added to the FAQ. Please use the coursework discussion channel at LMO as the first point of call for any questions or problems or clarifications or anything you would like us to go over. If you prefer to ask privately, please send an email to the relevant instructor.

**Good luck!**