COMP1204/Coursework2

From NotesWiki
< COMP1204

Contents

Assessment Criteria

This coursework will be marked out of 40. There are 2 marks available for every exercise, of which there are 20.

Marks will be awarded as follows:

- **0 marks**: Not attempted/wrong
- **1 mark**: Partially attempted/correct
- **2 marks**: Complete and correct

This Coursework counts for 15% of this module. The deadline for this coursework is Wednesday 27th May.

Failure to submit by the deadline will incur a 10% penalty per working day. Submissions later by more than five working days will not be accepted.

Your Task

Your task is to create a SQlite 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 normalising a database to hold the data from a dataset and constructing queries against the data once it is in a suitable form.

There are 4 parts, along with an optional extension, for this coursework:

1. The Relational Model (20%)
2. Normalisation (25%)
3. Modelling (20%)
4. Querying (30%)
5. Extension (5%)

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 any answers. You will also need to construct a SQLite database, and produce a set of queries that can be run against that database.

Dataset

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

Please use the snapshot of this dataset, taken on the 27th of April, available here for this coursework:
http://secure.ecs.soton.ac.uk/notes/comp1204/1920/ofb/cw2/dataset.csv

1. The Relational Model

EX1: Write down the relation directly represented in the dataset file. Assign relevant data types to each column.

EX2: List all of the functional dependencies that exist in the dataset

- Note: You do not need to (but will not be penalised for) including overlapping dependencies, but you should aim for the tightest dependency (e.g. if A -> B, it adds nothing that A, C -> 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 do not need to include trivial dependencies

EX3: List the potential candidate keys

EX4: Identify a suitable primary key, and justify your decision

2. Normalisation

Keys: Where possible, you should only introduce new Surrogate Keys where they are necessary. If and where they are necessary, 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 remove any of the attributes in the original relation (such as the dateRepresentation, even though it is otherwise broken up)

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 normalisation 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, for example.

EX5: List any partial-key dependencies in the relation as it stands and any resulting additional relations you should create as part of the decomposition.

EX6: 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.

EX7: List any transitive dependencies in your new relations

EX8: 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.

EX9: Is your relation in Boyce-Codd Normal Form? Justify your answer.

3. Modelling

Note: Where possible, you should only introduce new Surrogate Keys where they are necessary. If and where they are necessary, to avoid anomalies, you should explain this in your report with a justification.

Attribute Names: All original attribute names must remain as they were in the original dataset, you may not rename them - this ensures consistency with the original dataset and easy importing of new data.

EX10: Using the CSV import function, import the raw dataset into *SQLite* into a single table called 'dataset' in an SQLite database called *coronavirus.db*.

- Tip: You can import a CSV in SQlite using .mode csv then .import *file table*
- Note: You may not change the CSV file - it must be the original provided dataset file

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

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

EX11: Write the SQL to create the full normalised representation, including all additional tables, with no data. The SQL should contain CREATE statements to create any new tables. You should include indexes where appropriate, and list and justify these in your answer.

If you have introduced any surrogate keys, please list and justify them as part of this answer.

- The SQL statements to create the tables should be saved as *ex11.sql*
- The entire database at this point should be dumped as *dataset2.sql*

EX12: Write INSERT statements using SELECT to populate the new tables from the 'dataset' table

- The SQL statements to populate the tables from the dataset table should be saved as *ex12.sql*
- The entire database at this point should be dumped as *dataset3.sql*

EX13: Test and ensure that on a clean SQLite database, you can execute dataset.sql followed by ex11.sql followed by ex12.sql to successfully populate your database.

4. Querying

For each exercise in this question, you will need to write an SQL query (against your newly created normalised tables in your database). Each SQL statement should be written in the report, as well as saved as *ex<number>.sql* (e.g. ex14.sql for the first) which can be run against your database, as it stands at the end of EX12.

IMPORTANT: When run, the query **MUST return the results** (such that sqlite3 coronavirus.db < ex14.sql will return the results)

Ordering: Remember, you can order by things not in the SELECT part of the query (e.g. ORDER BY year,month,day)

Write an SQL statement for each of the following:

EX14: The worldwide total number of cases and deaths (with total cases and total deaths as columns)

EX15: The number of cases and the date, by increasing date order, for the United Kingdom (with date and number of cases as columns)

EX16: The number of cases, deaths and the date, by increasing date order, for each continent (with continent, date, number of cases and number of deaths as columns)

EX17: The number of cases and deaths as a percentage of the population, for each country (with country, % cases of population, % deaths of population as columns)

EX18: A descending list of the the top 10 countries, by percentage deaths out of the cases in that country (with country name and % deaths of country cases as columns)

EX19: The date against a cumulative running total of the number of deaths by day and cases by day for the united kingdom (with date, cumulative UK deaths and cumulative UK cases as columns)

- Tip: You will want to use the Window Functions (https://www.sqlite.org/windowfunctions.html) in SQLite to achieve this. However, these require a newer version of SQLite, which you can download from the SQLite website (https://www.sqlite.org/download.html) . I would suggest either using the autoconf amalgamated version and using the configure script with a --prefix, or using the precompiled Linux binaries.

5. Extension

EX20: Using GnuPlot, write a small script (plot.sh) which will, using the data in the SQLite database (called coronavirus.db), produce a graph named graph.png with the date on the horizontal axis and the cumulative number of deaths by country on the vertical axis.

- Tip: You may need to create temporary files or folders as part of your script - please ensure you use *mktemp* to create these (you can create an entire temporary folder with mktemp and then write inside it if you wish).
- Note: You must run 'sqlite3' not a different path or copy. If you have updated sqlite3 and need to run it explicitly, please add it to your path not the script.
- Note: Assume that the coronavirus.db file is in the same folder as the script.
- Note: There will be a lot of countries if you represent them all, so if possible, limit to the *top 10* (in terms of cumulative deaths) countries

Include an explanation of your script in the report. The full script and resulting graph should be included as an appendix in the report (not counting towards the page limit) and in the archive itself.

Submission

Report

You must write your report in LaTeX and produce a report PDF. The name of the generated file should be report.pdf, the source report.tex. Your report should not be more than 5 pages long excluding the cover (first) page. Your report should contain:

A title, your name, your username and your student ID.

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

1. The Relational Model
2. Normalisation
3. Modelling
4. Querying
5. Extension (if applicable)

Each section **must** have a **/subsection** for each exercise labelled EX#.

Files

Your submission should be a single *cw2.tar.gz* tar.gz file. Your submission *must* include the following in this structure:

1. *report.tex*: The full source code to the LaTeX report
2. *report.pdf*: The generated result of the LaTeX report
3. *dataset.sql*: The full dump of the database after importing the dataset
4. *dataset2.sql*: The full dump of the database after creating the normalised tables
5. *dataset3.sql*: The full dump of the database after the normalised tables have been populated
6. *ex11.sql*: The SQL to create the normalised tables
7. *ex12.sql*: The SQL to populate the normalised tables from the dataset table
8. *ex14.sql*
9. *ex15.sql*
10. *ex16.sql*
11. *ex17.sql*

12. *ex18.sql*
13. *ex19.sql*
14. *plot.sh*: The script to generate a graph for the extension section (optional)
15. *graph.png*: The generated graph (optional)

Support

Any questions and answers will be added to the FAQ

Please email ob1a12@soton.ac.uk with any questions you may have or anything you would like us to go over. Please make sure you include COMP1204 in the subject.

Good luck!

Retrieved from "https://secure.ecs.soton.ac.uk/noteswiki/index.php?title=COMP1204/Coursework2&oldid=32310"

- This page was last modified on 29 June 2020, at 10:55.