

ISYS2095 Database Concepts

Assessment 3: Database Design Project



Assessment type: Take-home assessment

Word limit: N/A



Due Date: Sunday 26 February (Melbourne Time)



Weighting: 35%, 35 Marks

Overview

This is a practical and real-world project that puts the knowledge you gained into practice. You are required to investigate and understand a publicly available dataset, design a conceptual model for storing the dataset in a relational database, apply normalisation techniques to improve the model, build the database according to your design, and import the data into your database, and develop several SQL queries for a set of requirements.

An overview of the sections and grades is as follows.

- Part A: Understanding the Data (0 Marks, Preliminary Work)
- Part B: Designing the Database (10%)
- Part C: Creating the Database and Importing Data (10%)
- Part D: Data Retrieval and Visualisation (15%)

Assessment criteria

This assessment will measure your ability to:

- Analyse the requirements outlined in the problem description.
- Develop a conceptual model for the design of a database backend required for the system.
- Use an industry-standard ER modelling tool to draw the ER model.
- Use the mapping process to create relational database schema.
- Use normalisation process to evaluate the schema and make sure that all the relations are at least 3NF.
- Create tables on SQLite Studio and populate them with data available from the specified sources.

- Write SQL statements required for CRUD (create, read, update, and delete) operations on the database you built.
- Develop your knowledge further to represent data in a meaningful way using data visualisation.

Course learning outcomes

This assessment is relevant to the following course learning outcomes.

CLO1	Describe various data modelling and database system technologies.
CLO2	Explain the main concepts for data modelling and characteristics of database systems.
CLO3	Identify issues with, compare and justify relational database designs using the functional dependency concepts.
CLO4	Apply SQL as a programming language to define database schemas and update database contents.
CLO5	Apply SQL as programming language to extract data from databases for specific users' information needs.
CLO6	Design a database schema using conceptual modelling mechanisms such as entity-relationship diagrams.

Assessment details

Part A: Understanding the Data

In this assignment, we are working with a publicly available dataset: **A Global Database of COVID-19 Vaccinations**. Further details about this dataset are available in the article at: <https://www.nature.com/articles/s41562-021-01122-8>. The abstract of the article is as follows.

An effective rollout of vaccinations against COVID-19 offers the most promising prospect of bringing the pandemic to an end. We present the Our World in Data COVID-19 vaccination dataset, a global public dataset that tracks the scale and rate of the vaccine rollout across the world. This dataset is updated regularly and includes data on the total number of vaccinations administered, first and second doses administered, daily vaccination rates and population-adjusted coverage for all countries for which data are available (169 countries as of 7 April 2021). It will be maintained as the global vaccination campaign continues to progress. This resource aids policymakers and researchers in understanding the rate of current and potential vaccine rollout; the interactions with non-vaccination policy responses; the potential impact of vaccinations on pandemic outcomes such as transmission, morbidity and mortality; and global inequalities in vaccine access.

A live version of the vaccination dataset and its documentations are available in a public GitHub repository at <https://github.com/owid/covid-19-data/tree/master/public/data/vaccinations>. There are several data files that can be downloaded in CSV and JSON formats.

For the purposes of completing this assignment, we are using the following files. You are required to review and analyse the dataset available in these files. You will find that reviewing the rest of the files, even if not listed below, will help you to form a better understanding about the big picture.

	FILE NAME	DESCRIPTION
1	locations.csv	Country names and the type of vaccines administered. Each line represents the last observation in a specific country. Refer to README.md for the details.
2	us_state_vaccinations.csv	History of observations for various locations in the US.
3	vaccinations-by-age-group.csv	History of observations for vaccinations of various age groups in each country.
4	vaccinations-by-manufacturer.csv	History of observations for various types of vaccines used in each country.
5	vaccinations.csv	Country-by-country data on global COVID-19 vaccinations. Each line represents an observation date. Refer to README.md for the details.
6	country_data/China.csv	Daily observations of vaccination in China.
7	country_data/Egypt.csv	Daily observations of vaccination in Egypt.

8	country_data/Israel.csv		Daily observations of vaccination in Iseael.
9	country_data/United Emirates.csv	Arab	Daily observations of vaccination in UAE.

Table 1: List of data files with brief descriptions

Part B: Designing the Database (10%)

Task B.1 Produce an ER diagram for a relational database that will be able to store the given dataset.

It is important to note that the given CSV files are not representing a good design for a relational database. It is your task to design a database that will adhere to good design principles that were taught throughout the course. If you are doing this for the first time, it is likely that, at some stage, you will find yourself struggling to understand the relationship between various data files and their overall design. This is natural and is a part of the process. Figuring out how the data is structured requires persistence and attention to details, which are required in this discipline. This also means your database schema will not match the structure of the CSV files and, therefore, you will require to manipulate the structure of the dataset (and not the data itself) to be able to import it into your database. Importing the data is required to complete Task C.2.

The ER diagram must be produced by [Lucidchart](#) similar to the exercises that were completed in in the course. UML notation is expected, and using other notations will not be acceptable and will result in lost marks. Including a high-quality image representing of your model is important, which can be achieved using the Export function of Lucidchart.

You are also required to transform the ER diagram into a database schema that will be used in the next part of the assignment.

Creating a good database design typically involves some database normalisation activities. You should document your normalisation activities and support them with good reasoning. This typically involves explaining what the initial design was, what the problem was, and what changes have been made to rectify the issue.

The expected outcome of completing this task is one PDF file named design.pdf containing the following sections.

1. Database ER diagram and, if needed, a set of reasonable assumptions
2. Explanation of normalisation challenges and the resulting changes
3. Database schema

Part C: Creating the Database and Importing Data (10%)

Task C.1 Produce one SQL script file named database.sql. This file should contain all the SQL statements necessary to create all the database relations and their corresponding integrity constraints as per your proposed design in Part B. The script file must run without any errors in SQLite Studio and contain necessary commenting to separate various relations. Note that this script is *not* supposed to store/import any data into the relations.

The expected outcome of completing this task is one valid script file named database.sql.

Task C.2 Create a database file named Vaccinations.db. Import the given dataset from the CSV files into your database.

To complete this task, you will need to change the format of the CSV files to match the structure of your designed database. This can be done using Microsoft Excel.

The next step is to *import* the (modified) spreadsheets into the database you created in SQLite Studio in Task C.1. Use the menu option *Tools – Import* in SQLite Studio.

The expected outcome of completing this task is one database file named Vaccinations.db, which must contain all the data that is stored in the CSV files named in Table 1.

Part D: Data Retrieval and Visualisation (15%)

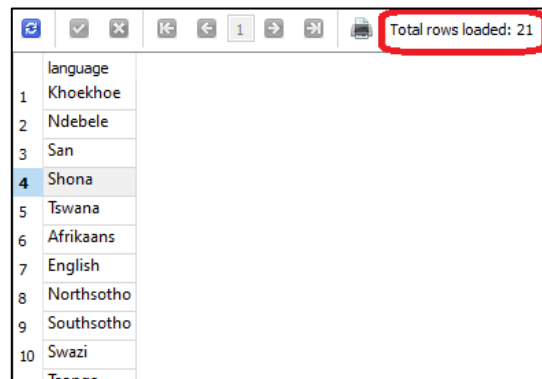
Now that you have created and populated a database, it is time to create some queries to investigate the data in various ways. In addition to writing the required queries, you are also asked to produce data visualisation for the results of your queries.

The tasks in this section represent the queries that must be supported. Each query must consist of one SQL statement. It would be acceptable to use several nested queries, combine several SELECT statements with various operators etc. However, it would not be acceptable to have multiple and separated queries for each task.

After you have written each query, you are expected to produce a data visualisation for each result set. You have the freedom to choose the tool for creating your visuals (e.g., Excel, Google Charts, Tableau) as well as the visualisation techniques (e.g., charts, plots, diagrams, maps). Completing this portion of the work will require that you understand the nature of the results of each query, undertake research to choose a visualisation tool you are comfortable with, decide about the best technique to visually represent each result set, and produce the data visualisation. **Answers to tasks in Part D that are not supported by a visualisation can achieve up to 80% of the grade associated with each task.**

The expected outcome of completing this task is as follows.

1. One SQL script file named Queries.sql containing *all* the queries developed for the tasks in this section. It is important that the queries are in simple text format and not presented as snapshots. To improve readability, add comments to separate the queries and indicate the task each query belongs to. Note that valid SQL comments must not generate errors in SQLite Studio. The marker of your work will use this file to verify *your* queries against *your* database.
2. A PDF file named QueryResults.pdf containing the following elements for each task.
 - a. The SQL query
 - b. A snapshot of the first 10 results of the query, which should also show the total number of rows loaded. A sample snapshot is shown in Figure 1 below for your reference.
 - c. Data visualisation



	language
1	Khoekhoe
2	Ndebele
3	San
4	Shona
5	Tswana
6	Afrikaans
7	English
8	Northsotho
9	Southsotho
10	Swazi
11	Tsonga

Figure 1: Sample results snapshot with total rows

List of Tasks

Task D.1 For any two given countries (i.e., you can assume any two countries, e.g., China and Israel), list the name of the countries, the total number of vaccines administered in each observation date in each country, and the difference between the administered vaccines. Each row in the result set must have the following structure.

Observation Date (OD)	Country 1 Name (CN1)	Administered Vaccine in Country 1 (TVC1)	Country 2 Name (CN2)	Administered Vaccine in Country 2 (TVC2)	Difference of totals (TVC1-TVC2)
--------------------------	----------------------------	--	-------------------------	--	--

Figure 2: Column Headers in the Result Set for Task D.1

Task D.2 Produces a result set containing the cumulative number of COVID-19 vaccine doses administered by each country in each year. For each pair of country and year available in the dataset, the result set must have a separate row with the following structure.

Country Name	Year	Cumulative Doses in the Year
--------------	------	------------------------------

Figure 3: Column Headers in the Result Set for Task D.2

Task D.3 Produce a list of all countries with the type of vaccines (e.g., Oxford/AstraZeneca, Pfizer/BioNTech) administered in each country. Each row in the result set must have the following structure where, for a given country, Vaccine Type Count represents how many different types of vaccine are administered in that country, and Vaccine Type Names contains a comma-separated set of vaccine names administered in that country.

Country Name	Vaccine Type Count	Vaccine Type Names
--------------	--------------------	--------------------

Figure 4: Column Headers in the Result Set for Task D.3

Task D.4 There are different data sources used to produce the dataset. Produce a report showing the total number of vaccines administered according to each data source (i.e., each unique URL). Order the result set by source name and URL. Each row in the result set must have the following structure.

Source Name	Total Administered Vaccines	Source URL
-------------	-----------------------------	------------

Figure 5: Column Headers in the Result Set for Task D.4

Task D.5 How do various countries compare in the speed of their vaccine administration? Produce a report that lists all the observation dates and, for each date, list the total number of people *fully vaccinated* in each one of the 4 countries used in this assignment. Each row in the result set must have the following structure.

Date	China	Egypt	Israel	UAE
------	-------	-------	--------	-----

Figure 6: Column Headers in the Result Set for Task D.5

Submission Format

The five files required for your submission must be named in a specific way. Assuming your student number is 1234567, the name of your files must be as below. Replace the first 7 digits with your student number.

- 1234567_1_Design.pdf
- 1234567_2_Database.sql
- 1234567_3_Vaccinations.db
- 1234567_4_Queries.sql
- 1234567_5_QueryResults.pdf

The content of each file is detailed in the previous sections of this document.

Referencing guidelines

Use [RMIT Harvard](#) referencing style for this assessment.

You must acknowledge all the courses of information you have used in your assessments.

Refer to the [RMIT Easy Cite](#) referencing tool to see examples and tips on how to reference in the appropriated style. You can also refer to the library referencing page for more tools such as EndNote, referencing tutorials and referencing guides for printing.

Academic integrity and plagiarism

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge, and ideas.

You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e., directly copied), summarised, paraphrased, discussed, or mentioned in your assessment through the appropriate referencing methods.
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.

RMIT University treats plagiarism as a very serious offence constituting misconduct.

Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to the [University website](#).

Assessment declaration

When you submit work electronically, you agree to the [assessment declaration](#).