**Database Systems**

# Assignment specification 2024

**Originality**: Your submission must be prepared by yourself and for this module; you may not work with others or re-use their work. **All Submission made to the Module will be checked against Plagiarism.**

## Overview

In this assignment, you will:

* consider a scenario that has been provided for a database design, review and adapt it for your assignment.
* identify a report (standard query) that can be run against your database design.
* plan how to test your report and identify the necessary test data.
* create a corresponding relational scheme, and verify it meets first, second and third normal forms.
* implement your database schema using MariaDB and load the test data.
* then write, run and test the report you identified.

The Assignment is broken down into two Parts so that you can use any feedback from the first part to help when completing the second. Separate links are provided for submitting each part of the assignment on the Moodle.

Submit by 17th of July 2024 [40]

Instructions for Part 1 Submission

Part 1 should be submitted in .pdf or .doc format to the appropriate Moodle upload link by the stated deadline. Please ensure all submissions are clearly labelled with your matriculation number. Also ensure all diagrams/screenshots and labelled are very clear and detailed.

The purpose of part 1 is to ensure that you have a good plan for the technical implementation in part 2. For guidance, you may want to allocate around 4 – 6 hours to this part.

**Task A**

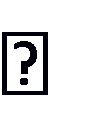
1. Frequent fliers at the Edinburgh Airport International has been complaining to the Airport management officials about the poor organization at the airport. As a result, the officials have decided that all information related to the airport should be organized using a DBMS, and you’ve been hired to design the database. Your first task is to organize the information about all the airplanes that are stationed and maintained at the airport. The relevant information is as follows:
   * Every airplane has a registration number, and each airplane is of a specific model.
   * The airport accommodates a number of airplane models, and each model is identified by a model number (e.g., DC-10) and has a capacity and a weight.
   * A number of technicians work at the airport. You need to store the name, SSN, address, phone number, and salary of each technician.
   * Each technician is an expert on one or more plane model(s), and his or her expertise may overlap with that of other technicians. This information about technicians must also be recorded.
   * Traffic controllers must have an annual medical examination. For each traffic controller, you must store the date of the most recent exam.
   * All airport employees (including technicians) belong to a union. You must store the union membership number of each employee. You can assume that each employee is uniquely identified by the social security number.
   * The airport has several tests that are used periodically to ensure that airplanes are still airworthy. Each test has a Federal Aviation Administration (FAA) test number, a name, and a maximum possible score.
   * The FAA requires the airport to keep track of each time that a given airplane is tested by a given technician using a given test. For each testing event, the information needed is the date, the number of hours the technician spent doing the test, and the score that the airplane received on the test.

1. You should construct an ER diagram for the airport database. Be sure to indicate the various attributes of each entity and relationship set; also specify the key and participation constraints for each relationship set. Specify any necessary overlap and covering constraints as well (in English). (The number of tables may be higher as some tables represent m:n relationships and lookups, not only entities). Each entity should normally have three or more attributes. Explain the reasons for any modifications you have made in constructing the schema.

You may us[e draw.io](https://www.draw.io/) [,STARUML](https://staruml.io/) or any suitable tool to author the diagram. Ensure you use the UML-type notation with clear relationship mappings.

1. You need to clearly justify as appropriate, your decision for all the keys and constraints you have included in the modified schema.

**Task B**

 Propose for a realistic report[[1]](#footnote-1) that might be performed on your database schema. Note that no SQL or DB query is required at this point. However, when implemented, your report query should:

* + - use data from at least three tables (not counting lookup tables).
    - summarise at least one set of attribute values (typically as a sum or count).
    - Include a scenario that allows you to perform correlated subqueries to retrieve any four attributes selected from a minimum of two tables.
    - Include a brief test plan with relevant sample test data to support the proposed testing.

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **Marking Criteria** | **Marks** |
| ER Diagram | Structure and labelling in diagram (10)  Correct use of ER conventions & keys (6)  Correct Justifications for key constraints (8)  Accurate representation of design (4) | 30 |
| Report and  Test Plan | Realistic suggestion, clear description (2)  Requires data from >= 4 tables (2)  Data aggregation, correlated subqueries (2)  Inclusion of Appropriate Test case with relevant test data (4) | 10 |
| **TOTAL** |  | 40 |

Submit by 14th of Aug 2024

Instructions for Part 2 Submission [60]

Part 2 should be submitted in .pdf or .doc format to the appropriate Moodle upload link by the stated deadline. Please ensure all submissions are clearly labelled with your matriculation number. Also ensure all diagrams/screenshots and labelled are very clear and detailed. Please ensure all SQL code is included as text (not only as screenshots), either incorporated into the document or in an appendix.

You should submit all SQL code together with the output of your report. For guidance, you may wish to allocate around 6 – 10 hours for this part.

You may wish to amend your database design based on feedback from Part 1 before continuing; in that case, explain your modifications. You may use any appropriate tools to interact with MariaDB; you should include screenshots in your report showing your work.

### Task D

* Derive an appropriate relational schema corresponding to your ER diagram of part 1.
* Demonstrate that all your tables and attributes eliminate all possible anomalies as much as possible. All assumptions should be properly documented separately, and for each form, either (a) demonstrate that your schema already meets the normal form, or (b) amend it so that it does.

### Task E

* Create a MariaDB database on the server soc-web-liv-11 implementing your database design.
* Document the following:

o DDL statements to create all database objects

o DML statements to insert test data into all tables

* You should insert sufficient test data to give a good indication of your report in use, and to support your test plan from Stage 1. This will normally require 10 or more rows of data per entity.

### Task F

* Implement the report query proposed in part 1 – write and run the corresponding SQL and present the output from your test data.
* Review the output in terms of your test plan; is the SQL correct?
* (Note that I should get the same result you have documented when I run your query against your database on soc-web-liv-11).

1. For example, if you chose a banking database design, your report could be “Show all customers in Wales who have a loan of over £2000 but have not made any repayment this month”. Of course, make the example appropriate to the database design you have chosen. [↑](#footnote-ref-1)