Assessment Information/Brief 2022-23

To be used for all types of assessment and provided to students at the start of the module. Information provided should be compatible with the detail contained in the approved module specification although may contain more information for clarity.

Module title	**Advanced Databases**
CRN	33386, 34902
Level	7
Assessment title	**Database Design & SQL for Data Analysis**
Weighting within module	This assessment is worth 100% of the overall module mark.
Module Leader/Assessment set by	**Professor Mo Saraee** **Dr Azadeh Mohammadi** **Nathan Topping**
Submission deadline date and time	**Friday 28th April 4pm** For coursework assessments only: students with a Reasonable Adjustment Plan (RAP) or Carer Support Plan should check your plan to see if an extension to this submission date has been agreed.
How to submit	You should submit your written report in the format of a word of PDF document uploaded to Blackboard via the Turnitin submission area, with the name **<<student-name>>_report.** You should also take a backup of your database on completion of the work and provide the backup file and a SQL script including all T-SQL statements used in the completion of this assignment. The .bak and .sql files should be provided in one zip file and uploaded to the 'Code & Dashboard' submission area on Blackboard.

Assessment task details and instructions

Task 1 (65 marks)

Imagine you are employed as a database developer consultant for a library. They are currently in the process of developing a new database system which they require for storing information on their members, their library catalogue, loan history and overdue fine repayments. In your initial consultation with the library, you have gathered the information below. Please read the below carefully and continue to the task description.

Client Requirements

When a member joins the library, they need to provide their full name, address, date of birth and they must create a username and password to allow them to sign into the member portal. Optionally, they can also provide an email address and telephone number. Members are charged a fine if they have overdue books and the library has to keep track of the total overdue fines owed by an individual, how much they have repaid and the outstanding balance. When a member leaves, the library wants to retain their information on the system so they can continue marketing to them, but they should keep a record of the date the membership ended. Members can sign up and login online.

When a member has overdue fines, they can repay some or all the overdue fines. Each repayment needs to be recorded, along with the date / time of the repayment, the amount repaid and the repayment method (cash or card).

The library has a catalogue of items. For each they have an item title, item type (which is classified as either a Book, Journal, DVD or Other Media), author, year of publication, date the item was added to the collection and current status (which is either On Loan, Overdue, Available or Lost/Removed). If the item is identified as being lost or removed from the collection, the library will record the date this was identified. If the item is a book, they will also record the ISBN.

The library wants to also keep a record of all current and past loans. Each loan should specify the member, the item, the date the item was taken out, the date the item is due back and the date the item was actually returned (this will be NULL if the item is still out). If the item is overdue, an overdue fee needs to be calculated at a rate of 10p per day.

The library processes hundreds of loans a day and details of the items on loan are business critical for them, so they need to avoid data loss if their systems go down. However, if their systems are down for a couple of hours, they think this isn't too much of an issue for them.

Task Details

As the database consultant, you are required to design the database system based on the information provided above, along with a number of associated database objects, such as stored procedures, user-defined functions, views and triggers. Your submission will take the form of working T-SQL statements required for the steps outlined below, a backup of the

database created, and a report explaining and justifying your design decisions, and the process you followed to complete the tasks. You should include screenshots and the T-SQL statements within the report itself.

1. You should design and normalise your proposed database into 3NF, fully explaining and justifying your database design decisions and documenting the process you have gone through to implement this design using T-SQL statements in Microsoft SQL Server Management Studio, using screenshots to support your explanation. All tables and views must be created using T-SQL statements, which should be included in your report. Clearly highlight which column(s) are primary keys or foreign keys. You should also explain the data type used for each column and justify the reason for choosing this. You should also consider using constraints when creating your database to help ensure data integrity. You **must** include a database diagram as part of your submission. <u>If you have made any additional assumptions aside from the information above when designing your database, you should clearly state these.</u>

2. The library also requires stored procedures or user-defined functions to do the following things:
    a) Search the catalogue for matching character strings by title. Results should be sorted with most recent publication date first. This will allow them to query the catalogue looking for a specific item.
    b) Return a full list of all items currently on loan which have a due date of less than five days from the current date (i.e., the system date when the query is run)
    c) Insert a new member into the database
    d) Update the details for an existing member

3. The library wants be able to view the loan history, showing all previous and current loans, and including details of the item borrowed, borrowed date, due date and any associated fines for each loan. You should create a view containing all the required information.

4. Create a trigger so that the current status of an item automatically updates to Available when the book is returned.

5. You should provide a function, view, or SELECT query which allows the library to identify the total number of loans made on a specified date.

6. So that you can demonstrate the database to the client you should insert some records into each of the tables (you only need to add a small number of rows to each, however, you should also ensure the data you input allows you to adequately test that all SELECT queries, user-defined functions, stored procedures, and triggers are working as you expect).

7. If there are any other database objects such as views, stored procedures, user-defined functions, or triggers which you think would be relevant to the library given the brief above, you will obtain higher marks for providing these along with an explanation of their functionality.

Within your report, you will also need to provide your client with advice and guidance on:

- Data integrity and concurrency
- Database security
- Database backup and recovery

Generic information on these topics, which is not applied to the given scenario, is likely to score poorly.

To get more than a satisfactory mark, you **must** use all of the below at least once in your database:

- Views
- Stored procedures
- System functions and user defined functions
- Triggers
- SELECT queries which make use of joins and sub-queries

Your report for this task should be **no more than 4,000 words**, excluding code snippets.

Your report should also be structured and numbered so that we can easily identify which part of the report relates to each of the above numbered steps above.

Task 2 (35 marks)

For the second task, you should use the three csv files provided for you on Blackboard. These files are an excerpt from a larger file which is a real-world dataset released every month by the National Health Service (NHS) in England. The file provides a information on prescriptions which have been issued in England, although the extract we have provided focusses specifically on Bolton.

The data includes **three** related tables, which are provided in three csv files:

- The **Medical_Practice.csv** file has 60 records and provides the names and addresses of the medical practices which have prescribed medication within Bolton. The PRACTICE_CODE column provides a unique identifier for each practice.
- The **Drugs.csv** file provides details of the different drugs that can be prescribed. This includes the chemical substance, and the product description. The BNF_CHAPTER_PLUS_CODE column provides a way of categorising the drugs based on the British National Formulatory (BNF) Chapter that includes the prescribed product. For example, an antibiotic such as Amoxicillin is categorised under '05: Infections'. The

BNF_CODE column provides a unique identifier for each drug.

- The **Prescriptions.csv** file provides a breakdown of each prescription. Each row corresponds to an individual prescription, and each prescription is linked to a practice via the PRACTICE_CODE and the drug via the BNF_CODE. It also specifies the quantity (the number of items in a pack) and the items (the number of packs). The PRESCRIPTION_CODE column provides a unique identifier for each prescription.

For this task, imagine you work as a database consultant for a pharmaceutical company. They want to analyse the prescribing data to understand more about the types of medication being prescribed, the organisations doing the prescribing, and the quantities prescribed.

1. The first stage of your task is to create a database and import the three tables from the csv file. You should also add the necessary primary and foreign key constraints to the tables and provide a database diagram in your report which shows the three tables and their relationships. You should create the database with the name PrescriptionsDB and the tables with the following names:
    a. Medical_Practice
    b. Drugs
    c. Prescriptions

    You should also leave the column names as they appear in the csv file. This is so we can re-run your code.

2. Write a query that returns details of all drugs which are in the form of tablets or capsules. You can assume that all drugs in this form will have one of these words in the BNF_DESCRIPTION column.

3. Write a query that returns the total quantity for each of prescriptions – this is given by the number of items multiplied by the quantity. Some of the quantities are not integer values and your client has asked you to round the result to the nearest integer value.

4. Write a query that returns a list of the distinct chemical substances which appear in the Drugs table (the chemical substance is listed in the CHEMICAL_SUBSTANCE_BNF_DESCR column)

5. Write a query that returns the number of prescriptions for each BNF_CHAPTER_PLUS_CODE, along with the average cost for that chapter code, and the minimum and maximum prescription costs for that chapter code.

6. Write a query that returns the most expensive prescription prescribed by each practice, sorted in descending order by prescription cost (the ACTUAL_COST column in the prescription table.) Return only those rows where the most expensive prescription is more than £4000. You should include the practice name in your result.

7. You should also write **at least five queries** of your own and provide a brief explanation

of the results which each query returns. You should make use of all of the following at least once:

- Nested query including use of EXISTS or IN
- Joins
- System functions
- Use of GROUP BY, HAVING and ORDER BY clauses

Your report for Task 2 should be no more than 2,000 words and should provide a brief description of the data import steps you followed (with screenshots as needed), along with your T-SQL statements, a brief explanation of each one and the full result set (where it is feasible to include this). As with Task 1, code snippets are **not** included in the wordcount.

Your report should also be structured and numbered so that we can easily identify which part of the report relates to each of the above numbered steps above.

Assessment Criteria	Information on the assessment criteria of this assignment is provided in the rubric at the end of this document.
Knowledge and Understanding	**Assessed intended learning outcomes** On successful completion of this assessment, you will be able to: 1. Design a relational database, following best practice principles, including database normalisation and entity-relationship modelling 2. Implement a database design using T-SQL to create database objects, including tables, views, stored procedures, triggers and user defined functions 3. Apply T-SQL SELECT statements to query data for data analysis purposes 4. Use your knowledge of database security, database recovery and transaction management to make recommendations in a real-world scenario
Word count	Your assessment should be no more than 6,000 words in total. The report for Task 1 should be no more than 4,000 words and the report for Task 2 should be no more than 2,000 words.
Academic Integrity and Referencing	Students are expected to learn and demonstrate skills associated with good academic conduct (academic integrity). Good academic conduct includes the use of clear and correct referencing of source materials. Here is a link to where you can find out more about the skills which students need: Academic integrity & referencing Referencing

	Academic Misconduct is an action which may give you an unfair advantage in your academic work. This includes plagiarism, asking someone else to write your assessment for you or taking notes into an exam. The University takes all forms of academic misconduct seriously.
Assessment Information and Support	**Support for this Assessment** You can obtain support for this assessment by contacting the module team via email on N.J.Topping@salford.ac.uk or A.Mohammadi1@salford.ac.uk . You can find more information about understanding your assessment brief and assessment tips for success here. **Assessment Rules and Processes** You can find information about assessment rules and processes in Blackboard in the Assessment Support module. **Develop your Academic and Digital Skills** Find resources to help you develop your skills here. **Concerns about Studies or Progress** If you have any concerns about your studies, contact your Academic Progress Review Tutor/Personal Tutor or your Student Progression Administrator (SPA). **askUS Services** The University offers a range of support services for students through askUS including Disability and Learner Support, Wellbeing and Counselling Services. **Personal Mitigating Circumstances (PMCs)** If personal mitigating circumstances (e.g. illness or other personal circumstances) may have affected your ability to complete this assessment, you can find more information about the Personal Mitigating Circumstances Procedure here. Independent advice is available from the Students' Union Advice Centre about this process. Click here for an appointment to speak to an adviser or email advicecentre-ussu@salford.ac.uk.
Reassessment	If you fail your assessment, and are eligible for reassessment. For students with accepted personal mitigating circumstances for absence/non submission, this will be your replacement assessment attempt. Explain what happens if a student needs to be reassessed, will the reassessment be the same, what is the submission date?

We know that having to undergo a reassessment can be challenging however support is available. Have a look at all the sources of support outlined earlier in this brief and refer to the Personal Effectiveness resources.

Level	Descriptor
Outstanding (90-100%)	• All T-SQL statements are correct, well-written and concise, with good commenting and demonstrate an excellent grasp of T-SQL. • Evidence in the T-SQL statements that all of the following have all been considered: concurrency, performance, data integrity, database security (Task 1 only) • All basic requirements of the brief have been met and the student has gone beyond these requirements to provide additional functionality • Excellent use of all required database objects, including stored procedures, user-defined functions, triggers, and views (Task 1 only) • SELECT queries make use of all requested clauses, subqueries, and joins in queries which are appropriately selected and justified in terms of a proposed use case. Clear explanation of the query and result set. Queries go beyond the basic requirements and show a high level of complexity while demonstrating knowledge of query optimisation considerations (Task 2) • Database is correctly normalised in 3NF, and schema provides flexibility to meet future client requirements. Inclusion of database diagram with full explanation of schema. Rigorous, clear, and concise documentation of the database design process which reflects best practice principles (Task 1 only) • In-depth and detailed demonstration of knowledge of database security, recovery and transaction management which shows clear application to the specifics of the given scenario (Task 1 only)
Excellent (80-89%)	• All T-SQL statements are correct, well-written and concise, with good commenting and demonstrate an excellent grasp of T-SQL. • Evidence in the T-SQL statements that the following have mostly been considered: concurrency, performance, data integrity, database security (Task 1 only) • All basic requirements of the brief have been met and the student has tried to go beyond these requirements to provide additional functionality • Excellent use of all required database objects, including stored procedures, user-defined functions, triggers, and views (Task 1 only) • SELECT queries make use of all requested clauses, subqueries, and joins in queries which are appropriately selected and justified in terms of a proposed use case. Clear explanation of the query and result set. Queries go beyond the basic requirements and show a high level of complexity while demonstrating knowledge of query optimisation considerations (Task 2) • Database is correctly normalised in 3NF. Inclusion of database diagram with full explanation of schema. Rigorous, clear, and concise documentation of the database design process which reflects best practice principles (Task 1 only) • In-depth and detailed demonstration of knowledge of database security, recovery and transaction management which shows clear application to the specifics of the given scenario (Task 1 only)
Very Good (70-79%)	• All T-SQL statements are correct, well-written and concise, with good commenting and demonstrate an excellent grasp of T-SQL. • Some evidence in the T-SQL statements of an attempt to consider concurrency, performance, data integrity or database security (Task 1 only) • All basic requirements of the brief have been met and the student has tried to go beyond these requirements to provide additional functionality • Good use of all required database objects, including stored procedures, user-defined functions, triggers, and views (Task 1 only) • SELECT queries make use of all requested clauses, subqueries, and joins in queries which are appropriately selected and justified in terms of a proposed use case. Clear explanation of the query and result set. Queries go beyond the basic requirements and show a reasonable level of complexity while

	demonstrating knowledge of query optimisation considerations (Task 2) • Database is correctly normalised in 3NF. Inclusion of database diagram with full explanation of schema. Clear documentation of the database design process which reflects best practice principles (Task 1 only) • Demonstration of knowledge of database security, recovery and transaction management which shows clear application to the specifics of the given scenario (Task 1 only)
Good (60-69%)	• All T-SQL statements are correct and demonstrate a strong grasp of T-SQL. • Some limited evidence in the T-SQL statements of an attempt to consider some of the following: concurrency, performance, data integrity or database security (Task 1 only) • All basic requirements of the brief have been met • Good use of most required database objects, including stored procedures, user-defined functions, triggers, and views (Task 1 only) • SELECT queries make use of most of the requested clauses, subqueries, and joins in queries which are appropriately selected and with some justification in terms of a proposed use case. Clear explanation of the query and result set. Queries show a reasonable level of complexity (Task 2) • Database is correctly normalised in 3NF. Inclusion of database diagram with full explanation of schema. Partial documentation of the database design process which reflects best practice principles (Task 1 only) • Some demonstration of knowledge of database security, recovery and transaction management which shows an attempt to apply to the specifics of the given scenario (Task 1 only)
Satisfactory (50-59%)	• Most T-SQL statements are correct • All basic requirements of the brief have been met • Satisfactory use of some required database objects, including stored procedures, user-defined functions, triggers, and views (Task 1 only) • SELECT queries make use of some of the requested clauses, subqueries, and joins in queries which are appropriately selected and with basic justification in terms of a proposed use case. (Task 2) • An attempt to correctly normalise the database in 3NF and to document these decisions. Inclusion of database diagram with partial explanation of schema. (Task 1 only) • Limited demonstration of knowledge of database security, recovery and transaction management which shows an attempt to apply to the specifics of the given scenario (Task 1 only)
Unsatis-factory (40-49%)	• Some T-SQL statements are correct • Most basic requirements of the brief have been met • An attempt to correctly normalise the database in 3NF and to document these decisions (Task 1 only)
Inadequate (30-39%)	• A few T-SQL statements are correct • Some basic requirements of the brief have been met • Database not correctly normalised (Task 1 only)
Poor (20-29%)	• Limited correct T-SQL statements • A few basic requirements of the brief have been met • Database not correctly normalised (Task 1 only)
Very Poor (10-19%)	• No correct T-SQL statements • No basic requirements of the brief have been met • Database not correctly normalised (Task 1 only)
Extremely Poor (0-9%)	• No attempt to correctly answer questions