Assessment Information/Brief 2024-25

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. Surbhi Bhatia Khan** **Dr Azadeh Mohammadi** **Dr. Muhammad Hammad Saleem**
Submission deadline date and time	**Friday 25th 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 issuing the e-boarding number for the pre-booked flights at the airport. They are currently in the process of developing a new database system which they require for storing information on the flight details, including passengers, reservation, employees, tickets, and baggage information. In your initial consultation with the airport, you have gathered the information below. Please read the below carefully and continue to the task description.

Client Requirements

An employee begins their shift by logging into the airport's secure ticketing system using their unique username and password (role can be classified as Ticketing Staff or Ticketing Supervisor with Employee ID, email and name). Once authenticated, they access the ticket reservation module to manage passenger bookings. If not authenticated, then Ticketing Supervisor will intervene by granting permission. A passenger who previously reserved a flight now requires ticket issuance. The employee retrieves the passenger's reservation by searching for the Passenger Name Record (PNR) containing the traveller's itinerary, booking details and status (confirmed, pending, cancelled). After confirming the reservation details such as flight number, departure and arrival times, and seat assignment (if preferred seat not available, will be set out to NULL), the employee proceeds to issue the ticket. This process involves generating an e-boarding number, which is then associated with the passenger's PNR in the system, the Employee ID of the employee issuing the boarding number will be stored. Additionally, the ticketing system calculates the additional fare, including any taxes or fees, is they have taken any additional services like extra baggage (100 GBP per kg), upgraded meal (20 GBP per person) or preferred seat (30 GBP per person). Each flight, identified by a unique FlightID, flight number, departure time, arrival time, origin, destination, is associated with multiple tickets (Ticket ID, reservation ID, issuedate, issuetime, fare, seatnumber, class (which can be business, firstclass or economy), linking passengers to specific flights and seat assignments. Passengers, identified by PNR including PassengerID, email, meal (vegetarian or non-vegetarian) DoB, first name, and last name (where passenger emergency contact number is optional), can hold multiple tickets, reflecting their travel itineraries. The system tracks baggage information by associating each flight with its corresponding ticket, weight, status (checkedin or loaded), and additional services like baggage fee. This process ensures that passengers with prior reservations are efficiently issued tickets, maintaining the airport's commitment to operational excellence and customer satisfaction.

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. If you have made any additional assumptions aside from the information above when designing your database, you should clearly state these. Create tables according to the scenario explained above which should include details on flights, reservations, employees, baggage, passengers, and tickets. Populate (Insert) the tables with the appropriate number of records (at least 7). You should also ensure the data you input allows you to adequately test that all the following queries.

2. Add the constraint to check that the reservation date is not in the past.
3. Identify Passengers with Pending Reservations and Passengers with age more than 40 years.
4. The ticketing system also requires stored procedures or user-defined functions to do the following things:
   a) Search the database of the ticketing system for matching character strings by last name of passenger. Results should be sorted with most recent issued ticket first.
   b) Return a full list of passengers and his/her specific meal requirement in business class who has a reservation today (i.e., the system date when the query is run)
   c) Insert a new employee into the database.
   d) Update the details for a passenger that has booked a flight before.

5. The ticketing system wants to be able to view all e-boarding numbers Issued by a Specific Employee showing the overall revenue generated by that employee on a particular flight. It should include details of the fare, additional baggage fees, upgraded meal or preferred seat. You should create a view containing all the required information.

6. Create a trigger so that the current seat allotment of a passenger automatically updates to reserved when the ticket is issued.

7. You should provide a function or view which allows the ticketing system to identify the total number of baggage’s (which are checkedin) made on a specified date for a specific flight.

8. 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 airport scenario given the brief above, you will obtain higher marks for providing these along with an explanation of their functionality. (Give 2 additional queries).

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

NB: All functions and stored procedures in task1, should be called as well (You should define, call them and show the results) Your report for Task 1 should be no more than 4,000 words (code snippets are not included in the wordcount). In the first part you should provide a description of the database design and normalization, then you should answer each question, explain and show the results and then you should explain what you have considered in your database for Data integrity and concurrency, Database security and database backup and recovery. 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. You should include the screenshots of all the codes and their outputs/results with explanation in your report You should include one SQL file for Task 1 in the zip file provided in the code area. Name the SQL file "studentid_task1.sql" and make sure to separate the answers to different questions by commenting very clearly.

Task 2 (35 marks)

For the second task, you should use the five csv files provided for you on Blackboard. These files are an excerpt from a larger file which is a real-world dataset from an online shopping site. The file provides the information on customers on products and their orders, order items with payments. The data includes **five** related tables, which are provided in five csv files:

- The **Customer.csv** file has 120 records and provides the customer_id, name, email, phone and country. Customer_id is a unique key identifier.
- The **Product.csv** file provides details of the 5 different products that can be purchased. This includes the product_id, product_name, category and price. The product_id column provides a unique identifier for each product.
- The **Orders.csv** file provides a total of 120 records with order_id, customer_id and order_date. The order_id column provides a unique identifier for each order.
- **Order_items.csv** lists the total 120 rows including order_item_id, order_id, product_id, quantity, price_each, Total_price and Total_amount. The order_item_id column provides a unique identifier for each order-item. Total_amount is calculated based on the multiple order_id placed for different products.
- **Payments.csv** includes 120 rows including payment_id, order_id, payment_date, payment_method and Amount_paid. The payment_id column provides a unique identifier for this table.

1. For this task, imagine you work as a database consultant for an Online shopping site. The first stage of your task is to create a database and import the five 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 five tables and their relationships. You should create the database with the name OnlineShoppingDB and the tables with the following names:
    a. Customers
    b. Products
    c. Orders
    d. Orders_items
    e. Payments

    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 the names and countries of customers who made orders with a total amount between £500 and £1000.

3. Get the total amount paid by customers belonging to UK who bought at least more than three products in an order.

4. Write a query that returns the highest and second highest amount_paid from UK or Australia – this is calculated after applying VAT as 12.2% multiplied by the amount_paid. Some of the results are not integer values and your client has asked you to round the result to the nearest integer value.

5. Write a query that returns a list of the distinct product_name and the total quantity purchased for each product called as total_quantity. Sort by total_quantity.

6. Write a stored procedure for the query given as: Update the amount_paid of customers who purchased either laptop or smartphone as products and amount_paid>=£17000 of all orders to the discount of 5%.

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.

NB: You should include the screenshots of all the codes and their outputs/results with explanation in your report. You should include one SQL file for Task 2 in the zip file provided in the code area.

Name the SQL file "studentid_task2.sql" and make sure to separate the answers to different questions by commenting very clearly.

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 s.khan138@salford.ac.uk, m.h.saleem@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